

REPUBLIQUE TUNISIENNE
Ministère de l'Education

INBMI

Institut National de Bureautique
et de Micro-Informatique

AFDI

Association Francophone de la
Didactique de l'Informatique

LES ACTES

DE LA

*5ème Rencontre Francophone
sur la Didactique de
l'Informatique*

10-11 et 12 avril 1996

Monastir - TUNISIE

Comité international de l'AFDI

Tahar Hafaied	Président de l'AFDI, INBMI 18 bis, rue de l'Inde 1002 Tunis -Tunisie -
Mohamed Ben Ahmed	Secrétaire d'Etat auprès du Premier Ministre chargé de l'Informatique - Tunisie -
Charles DUCHATEAU	CEFIS, Facultés Universitaires N-D de la paix, rue de Bruxelles / 61 B-5000 Namur -Belgique-
Alain BRON	SSIE-AFDI / case 588-CH 1401Yverdon- Les-Bains - Suisse -
Georges-Louis BARON	Institut National de Recherche Pédagogique Département Technologies Nouvelles et Education. 91, rue Gabrielle Péri, 92120 Montrouge - FRANCE .
Jacques BAUDE Jacques LUCY Catherine COUDERC	Enseignement Public et Informatique 13, rue de Jura 75013 Paris - FRANCE
Régine RAYNAUD	Université Paul Sabatier .118 route de Narbonne, 31062 Toulouse Cedex -FRANCE -
Chantal RICHARD	Institut Universitaire de Technologie de Villetaneuse .Université de Paris XIII Avenue Jean-Baptiste Clément 93430 Villetaneuse -FRANCE -
Philippe CORNU	ENA 1, rue Sainte Marguerite 67000 Strasbourg - FRANCE
Mary Thérèse REY	Petits Epinays 29 ch 1920 Martigny - Suisse -
Mohamed Miled	INSE-Tunisie
Farouk Kamoun	ENSI-Tunisie
Abdelhafidh ABIDI	Inspecteur d'informatique - Tunisiè -

A. Ben

REPUBLIQUE TUNISIENNE
Ministère de l'Education

INBMI
Institut National de Bureautique
et de Micro-Informatique

AFDI
Association Francophone de la
Didactique de l'Informatique

LES ACTES

DE LA

*5ème Rencontre Francophone
sur la Didactique de
l'Informatique*

10-11 et 12 avril 1996

Monastir - TUNISIE

S O M M A I R E

I. Ouverture du Colloque

- L'allocution de M. le Ministre de l'Education 5
- L'allocution de M. le S.E. auprès du Premier Ministre chargé de l'Informatique 7

II. Les Conférences

- L'informatique: de la science à la pédagogie (*Tahar. HAFAIED*) 9
- L'informatique: science humaine ou inhumaine (*Claude PAIR*) 15
- (R) Evolutions et tendances (*Raymond MOREL*) 33

III. Les Communications

Thème1 : Informatique au Primaire

- Approche de l'informatique au niveau préscolaire (*Francis LOWENTHAL*) 47
- Les apports du «jeu de l'enfant-robot» à la didactique de l'informatique (*Eric GREFF*) 67

Thème2 : Informatique au Secondaire

- Utilisation raisonnée des instruments micro-informatiques dans les disciplines de l'enseignement secondaire (*Jean-François LEVY*) 89

Thème3 : Informatique au Supérieur (Sciences Humaines)

- La didactique de l'informatique à l'université des Lettres et des Arts: Tunisie (*Husein HEBAILI*) 101

Thème4 : Informatique au Supérieur (Sciences Techniques)

- Entre informatique-outil et informatique-science: la question de l'usage: le cas de l'enseignement de la programmation (*Eric BATARD*) 117
- Evolution des langages de programmation, évolution des méthodes d'enseignement (*Jean VICARD*) 133
- L'enseignement de la conception logicielle (*Patrice RICHARD*) 147

Thème5 : La Formation Continue

- Formation des enseignants en informatique dans le secondaire: ingrédients et recettes pour le fonctionnement d'un centre de formation (*Etienne VANDEPUT*) 161

Thème6 : L'Enseignement Professionnel

- Intégration du pilotage de micro-robots pédagogiques à un environnement de programmation (*Pascal LEROUX*) 183
- Contribution à l'étude des représentations sociales de l'informatique chez les lycéens (*Hervé DAGUET*) 195

Thème7 : Informatique Générale

- Quels concepts opératoires en didactique générale peut-on interroger pour la constitution d'une didactique de l'informatique (*Mohamed MILED*) 207
- L'apprentissage de l'informatique dans les milieux scolaire, universitaire et professionnel: impacts méthodologies d'apprentissage et environnements technologiques (*Maha KHEMAJA*) 219
- Connaissances des étudiants en informatique avant le cours de traitement de l'information (*Marie-Thérèse REY*) 229
- Une pédagogie d'apprentissage d'outils informatiques (*Mohamed Kamel BEN RHOUMA, Mohamed BEN AHMED*) 239

IV. Les Ateliers

Thème1 : Informatique au Primaire

- Instruments informatiques pour l'enseignement préélémentaire
(*Théo PRINIOTAKIS*) 253
- Symboles écrits et symboles graphiques: plus qu'une querelle d'école ?
un exemple issu de la robotique pédagogique (*Paul DELANNOY*) 263

Thème2 : Informatique au Secondaire

- G.B.D.: Les requêtes, méthodes et exercices progressifs pour découvrir
les concepts fondamentaux et préparer à la création de nouvelles bases
(*E.VANDEPUT*) 273
- La compression de données:prétexte pour la programmation (*idem*) 275
- Robotique Virtuelle (*Rupert-Meurice de DORMALE*) 277
- Le projet Minerva (*Rui João-Baptista. SOARES*) 287
- Banques de données pédagogiques en maths .. - Analyse de Q.C.M.
(*Jean-Claude RODRIGUEZ, Jean-Pierre MOUSSETTE*) 295

Thème3 : Informatique au Supérieur (Sciences Humaines)

- Mieux les connaître pour mieux les initier
(*Georges ANTONIADIS, Taghi BARUMENDZADEH*) 307
- Utilisation des analogies multiples dans l'apprentissage de la
programmation destiné à des étudiants de filières non-scientifiques:
étude d'opportunité (*Gabriel MICHEL*) 319
- Connaissance des étudiants en informatique avant le cours de traitement
de l'information (*Caroline de SCHAETZEN*) 333
- Quelle informatique pour une université de lettres et sciences humaines ?
L'expérience de l'université Stendhal, dix ans après
(*Georges ANTONIADIS, Marie-Christine JANDARD*) 377
- Une première expérience de brai-storming en T.D. d'informatique
(*Patrice de MARCONNAY*) 387

Thème4 : Informatique au Supérieur (Sciences Techniques)

- Au ras des paquerettes (*Guy CHATY*) 403
- Une expérience d'enseignement optimal de CAML à l'Université
(*Daniel BOURGET*) 413
- Apprendre avec Alliance
(*M-M. PAGET, M.M.ROSSET-LOUERAT, A. DERIEUX*) 427

Thème6 : L'Enseignement Professionnel

- L'enseignement de l'informatique-outil dans l'enseignement
technique supérieur (*Robert WALD*) 445
- Les outils de l'informatique: didactique pour l'enseignement secondaire,
didactique de l'utilisation des logiciels en milieu professionnel,
quelles influences ? (*Sass FERENC*) 449

V- Les Recommandations Finales du Colloque

VI-La Table Ronde de Clôture

Epilogue

- Conférence de M. Jacques Arzac: « De l'ordinateur à l'informatique » 475
- La situation dans les pays francophones 489
 - En Tunisie
 - En Belgique
 - En France
 - En Suisse

L'allocution de Monsieur le Ministre de l'Education

Mesdames et Messieurs;

Honorables Invités;

La Tunisie, qui a pris l'engagement d'entrer de plain-pied dans son époque, a fait de l'enseignement le fondement de son essor et de l'ouverture sur les autres cultures un moyen d'interaction et d'enrichissement permanent.

Parce que de telles orientations sont tributaires, pour leur concrétisation, d'efforts constants et de mutations continues, la réforme du système éducatif, initiée par le Président Ben Ali au lendemain du changement, s'est fixé comme objectif, à l'instar de la réforme globale dont elle fait partie intégrante, de contribuer à l'édification d'une société moderne, aux solides ancrages et résolument tournée vers l'avenir.

La profusion de connaissance que connaît le monde en cette fin du siècle et la diversité des savoirs que nous diffusent les médias placent notre école devant la nécessité de s'adapter rapidement afin que les élèves soient dotés des compétences et des habiletés requises. Il ne suffit plus aujourd'hui d'acquérir des connaissances et de les accumuler, il importe surtout de les mettre en pratique dans des situations authentiques d'apprentissage.

C'est à cet effet que nous avons institué un enseignement de base obligatoire, assuré l'encadrement des jeunes, offert à tous les mêmes opportunités, rénové les programmes, multiplié les stages et ouvert des horizons nouveaux par le biais du recyclage et de la formation continue.

Et s'il est vrai que nous sommes en droit d'être fiers aujourd'hui des résultats obtenus, il n'en est pas moins vrai que nous devons redoubler d'efforts afin de gagner la bataille de la mise à niveau globale et du transfert de la technologie.

Depuis le changement du 7 novembre 1987, la Tunisie est persuadée, que dans un monde en perpétuel devenir, il n'y a pas de place pour ceux qui ne s'y adaptent pas. Et comme les Sciences constituent le fondement du progrès, nous nous sommes fait un devoir de les maîtriser. La technologie constituant le principe même de toute évolution, nous avons focalisé notre intérêt sur une formation des jeunes dans ce domaine.

Le développement du savoir nous interpellant sans cesse, nous nous sommes attachés à rénover nos méthodes d'enseignement, à adapter nos outils de travail et à approfondir nos connaissances en tirant profit des techniques les plus avancées dans le domaine de la pédagogie et de l'informatique.

Ainsi, peut-on mesurer l'importance que nous accordons à toute manifestation qui, comme la vôtre, réunit d'éminents chercheurs et pédagogues,

pour débattre de sujets actuels qui nous préoccupent tous et auxquels il ne peut y avoir d'ailleurs de solutions individuelles et isolées.

L'informatique, Mesdames et Messieurs, nous introduit certes dans un monde magique et merveilleux, mais elle doit être appréhendée sur le mode du réalisme, de la rationalité et de l'efficacité. En outre, loin d'être conçue comme une matière d'enseignement autonome, elle doit constituer un moyen au service d'un enseignement efficient de toutes les disciplines.

L'informatique qui gagne peu à peu nos établissements scolaires, comme outil de gestion administrative et comme outil didactique, constitue un excellent moyen permettant de réaliser la mise à niveau et facilitant son adaptation à un contexte marqué par des mutations rapides. En effet, dans un monde régi par les nouvelles technologies, sillonné par les autoroutes de l'information, notre école doit être suffisamment armée pour relever les défis.

Nos efforts portent aussi bien sur l'équipement des établissements scolaires en matériel informatique, que sur la formation des jeunes et des formateurs. Permettez-moi, Mesdames et Messieurs, de vous donner à ce propos quelques indications chiffrées :

1. Les établissements équipés de laboratoire d'informatique sont aujourd'hui au nombre de 183; en l'an 2000, nous en aurons 480.
2. Les élèves bénéficiant d'une formation en informatique étaient en 92/93 au nombre de 9000; ils sont 34500 aujourd'hui. En l'an 2000, il y en aura 108 000.
3. Nous avons actuellement 230 enseignants; en l'an 2000 ce nombre passera à 688.

Les professeurs chargés de l'enseignement de l'informatique sont soit des spécialistes de la maîtrise en Sciences de l'informatique, soit des enseignants appartenant à d'autres disciplines et qui ont bénéficié d'une formation spéciale en informatique. Nous ne cessons d'encourager cette deuxième option, parce qu'elle représente, à nos yeux, le meilleur moyen de permettre aux enseignants des différentes disciplines de tirer profit de l'apport de l'informatique. C'est d'ailleurs dans ce cadre que s'inscrivent les recherches réalisées par l'Institut National de Bureautique et de Micro-Informatique pour l'enseignement de certaines disciplines telles que l'arabe, le français, l'histoire, les mathématiques, la physique... qui gagnent en efficacité du fait du caractère interdisciplinaire de la matière informatique.

Mesdames et Messieurs,

En vous remerciant de votre attention, je souhaite la bienvenue à nos invités et plein succès à vos travaux.

Hatem BEN OTHMAN

Ministre de l'Education

L'allocation de Monsieur le Secrétaire d'Etat auprès du Premier Ministre chargé de l'Informatique

Mesdames et Messieurs,

La Tunisie a reconnu, depuis le début des années 70, l'importance du rôle des nouvelles technologies de l'information et de la communication dans les processus de développement économique, social et culturel. Notre pays n'a cessé dès lors d'investir et de miser sur la formation des hommes pour nous permettre de participer à la mesure de nos moyens et en fonction de nos objectifs à la révolution technologique que vit le Monde, tant au plan de l'utilisation de la haute technologie qu'au plan de la recherche tant académique que finalisée.

Cette orientation a été largement renforcée après le changement du 7 novembre, car soucieux de donner à l'informatique la place qui lui revient, le Président de la République a pris plusieurs décisions visant le développement du secteur et la diffusion de la culture informatique. Parmi ces décisions on peut citer: *la libéralisation de l'importation du matériel informatique, l'annulation des droits de douane sur le matériel informatique, la promulgation de la loi pour la promotion du secteur des services et la promulgation de la loi sur la propriété intellectuelle.*

Dans ce contexte l'administration et les pouvoirs de décision joueront un rôle important pour soutenir la mise à niveau globale. Là aussi l'informatique sera présente avec force pour permettre d'abord à l'administration d'être plus efficace, d'offrir des services de qualité aux citoyens et d'assurer surtout une plus grande transparence dans ses activités mais aussi pour doter les pouvoirs de décision d'un système de pilotage assurant une plus grande visibilité, une possibilité de réactivité et une pertinence des informations.

Nous vivons aujourd'hui une mutation profonde vers la société postindustrielle: la société de l'information qui marquera sans nul doute la fin d'une époque, la mutation d'une civilisation qui après avoir misé sur la matière et l'énergie prend désormais compte de l'information et du temps comme fondements essentiels dans la gestion de l'activité humaine qu'elle soit physique ou intellectuelle.

Pour se préparer à toutes ces mutations (Mondialisation de l'économie, société planétaire de l'information), nous avons commencé à mettre en place une politique globale de formation pour la constitution d'un vivier de compétences dans tous les domaines. La politique de formation en Informatique fait l'objet d'une étude détaillée dans le cadre de la préparation du 9ème Plan Socio-Economique.

Cette politique traite de plusieurs axes :

Le 1er axe: la mise en place d'une stratégie homogène et coordonnée pour la généralisation de l'enseignement de l'informatique de l'Ecole de base jusqu'au premier cycle de l'enseignement supérieur.

Le 2ème axe: la mise en place d'une stratégie pour la formation de spécialistes en informatique, telles que: L'Informatique industrielle, les Réseaux et le Multimédia, la Productique

Le 3ème axe: la formation des doctorants en Informatique.

Le 4ème axe: la formation continue et continuée.

Le 5ème axe: la formation professionnelle et privée.

Le 6ème axe: la diffusion de la culture informatique.

Il s'agit pour la Tunisie d'assurer une 2ème alphabétisation qui permettra d'utiliser l'ordinateur et l'informatique d'une manière quasi-naturelle, car il faut bien avoir à l'esprit que l'analphabète du troisième millénaire sera celui qui ne saura pas utiliser l'ordinateur et les logiciels comme on utilise, aujourd'hui, le stylo, le cahier et les livres.

Mesdames et Messieurs,

La formation en informatique occupe une place de choix dans notre stratégie nationale. Mais il reste un travail de longue haleine qu'il faudrait entamer pour toucher tous les publics cibles.

A cet effet la Tunisie va s'employer à intensifier la formation et la diffusion de la culture informatique en utilisant toutes ses ressources, toutes les méthodes et technologies d'enseignement direct et à distance, l'informatique sera à la fois un support et un outil pour l'enseignement mais aussi une discipline privilégiée.

Mesdames et Messieurs

Toutes les forces vives de la Tunisie de l'ère nouvelle ne ménageront aucun effort pour atteindre les nobles objectifs fixés par le Président de la République.

Nous sommes certains que la réussite couronnera nos efforts d'autant plus que la Tunisie pourra compter sur la coopération avec les pays frères et amis et particulièrement avec les pays méditerranéens.

Je vous remercie pour votre aimable attention.

Mohamed BEN AHMED

Secrétaire d'Etat auprès du Premier
Ministre chargé de l'Informatique

Conférence 1

L'INFORMATIQUE : DE LA SCIENCE A LA PEDAGOGIE

Tahar Hafaied
Président de l'AFDI

présentée le mercredi 12 avril 1996 à 10h00

INTRODUCTION

Les découvertes scientifiques et les mutations technologiques se font à un rythme que l'histoire de l'humanité n'a jamais connu auparavant. Après avoir vécu l'ère de l'automatisation et l'ère de l'intégration des systèmes, on affronte l'ère de la mondialisation où l'information circule sans frontière et l'accès aux savoirs devient une nécessité démocratique et un devoir moral de la part de ceux qui le peuvent.

Pour y faire face, l'école s'est donnée à une solide formation qui se baserait sur les fondements de chaque discipline et qui insisterait davantage sur les méthodes qui permettent d'inculquer aux jeunes certaines qualités intellectuelles fondamentales et intrinsèques caractérisées par :

-une rigueur de raisonnement, de synthèse et de déduction logique permettant à cet esprit une grande faculté d'adaptation aux situations nouvelles et imprévisibles.

- une curiosité culturelle et scientifique et un désir ardent du savoir qui développerait chez les apprenants une volonté permanente de s'informer sur les nouveautés.

L'INFORMATIQUE A L'ECOLE

Je ne veux pas évoquer les arguments narratifs pour défendre les apports de l'utilisation de l'ordinateur dans les mécanismes d'apprentissage des différentes disciplines car les écrits sont suffisamment riches et abondants à ce propos. Il faut cependant remarquer que ces nombreuses expériences et réalisations ont été effectuées dans un cadre expérimental et d'une manière cavalière ce qui lui a permis d'échapper aux critiques qui ont frappé la discipline informatique, devenue vulnérable, par le fait qu'elle doit avoir un statut officiel pour exister même à titre optionnel.

Depuis les années 60 l'informatique a pris forme en tant que discipline scientifique grâce, entre autres, aux réflexions de Dreyfus, de Jean Lois Regal de Perlis et Chomsky mettant bien l'accent sur le fait qu'il ne s'agit pas de science de l'ordinateur ni celle de calcul mais une science qui sous-tend l'ordinateur et comme l'a définie Jacques Arsac "L'informatique est la science du traitement de l'information considérée comme le support formel des connaissances".

L'informatique a été introduite à l'école il y a déjà 25 ans et l'évaluation de sa rentabilité a suscité des polémiques entre adeptes et opposants et souvent les

pays ont été mis en demeure de choisir. Souvent les avis cachent une indécision de part et d'autre et l'on assiste à des mesures parfois contradictoires au sein d'un même pays. Combien de fois l'on décide d'instituer l'informatique comme discipline et quelques années après on la supprime pour la réinstaurer de nouveau plus tard.

La discipline informatique est en perpétuelle constitution et il est normal qu'il y'ait des hésitations et des balbutiements. Elle ne s'est pas dotée véritablement d'un cadre didactique conceptuel lui garantissant une transposition réfléchie des savoirs savants aux savoirs scolaires afin d'éviter de tomber dans les travers de l'algorithmique et lui préservant ses vertus d'ordre cognitifs afin d'éviter de faire de son apprentissage de simples procédés mécanisés et réduits à des recettes. Parfois quand un tel cadre existe, il ne tient pas compte des possibilités de soutien qu'elle doit aux autres disciplines. Mr Mohamed Miled évoquera, dans son intervention sur les concepts opératoires pour la constitution d'une didactique, l'ampleur de cette dimension.

Il faut aussi remarquer que durant de longues années la communauté scientifique internationale a observé un silence à l'égard du statut de cette discipline ce qui a insité certains à la prudence . En 1994 ce silence a été fort heureusement rompu et l'UNESCO avait chargé l'Organisation Internationale de Traitement de l'Information (l'IFIP) pour élaborer un programme scolaire d'informatique pour l'enseignement secondaire des pays industrialisés ainsi que des pays en voie de développement et un autre pour le supérieur.

L'Association Francophone sur la Didactique de l'Informatique (AFDI) n'a pas cessé durant cinq rencontres internationales biennales de mener des travaux de réflexion sur le statut de l'informatique dans les institutions de l'enseignement et de formation sur la base d'une concertation continue entre ses membres. Durant sa rencontre de Monastir en Tunisie 1996 les participants vont, certes, préciser davantage les objectifs bien connus par tous en matière de savoirs et des savoirs-faire moyennant des mesures concrètes réalisables. Ils seront surtout amenés à envisager éventuellement des voies nouvelles possibles pour pratiquer les concepts fondamentaux de l'informatique et à dégager davantage le soutien qu'elle peut apporter aux autres disciplines et peut être de lever les hésitations actuelles.

L'EVOLUTION DE LA PEDAGOGIE

L'histoire a montré que l'évolution des concepts informatiques a été souvent suivie d'une évolution parallèle des concepts pédagogiques en matière de présentation et de traitement des connaissances objet d'apprentissage. En effet, Depuis Socrate et Platon et jusqu'au début du 20^è siècle, la pédagogie a toujours été une affaire d'intuition. Cela n'empêchait pas les pédagogues, dans certains milieux, d'être très performants et même d'exceller dans l'art du savoir-enseigner. Avec le développement du concept de l'école de masse et les problèmes d'effectifs d'élèves que cela suppose, la notion de pédagogie devait de plus en plus se poser en termes d'efficacité. MONTAIGNE, au 16^è siècle déjà, insistait sur l'importance de la "tête bien faite" par rapport à la "tête bien pleine". ROUSSEAU, plus tard introduisit la

notion de psychologie en éducation en insistant sur l'importance de "connaître" l'apprenant avant de lui inculquer un quelconque savoir. "Commencez par étudier vos élèves, car assurément, vous ne les connaissez point", disait-il.

Depuis le début du 20^e siècle, de nombreuses théories en matière d'apprentissage ont été avancées. En Europe, les travaux de FREINET¹, de DECROLY et de MONTESSORI ont porté sur des tentatives d'améliorer la qualité de l'enseignement (soit du côté de l'enseignant soit du côté de l'apprenant) mais sont malheureusement restés insuffisants et/ou restreints à certaines expériences.

Entre la conception aristotélicienne de l'esprit² (qui suppose que l'esprit humain est une donation du ciel et celle des "béhavioristes"³ qui considèrent que l'esprit (et les acquis cognitifs) peut être formé par des mécanismes d'apprentissage⁴ finement étudiés et appliqués, il y a une voie intermédiaire qui semble accorder beaucoup d'importance à l'idée que chez l'enfant, c'est l'environnement qui lui permet de développer son esprit (ou facultés intellectuelles).

En effet, d'après Jean PIAGET⁵ et Seymour PAPERT⁶, l'enfant- en interagissant avec son environnement-, forge son esprit par des activités d'apprentissage sur l'apprentissage.

L'environnement socio-éducatif qui semble en cette fin de siècle de plus en plus se caractériser par un envahissement des nouvelles technologies (TV, lecteurs CD-Audio et Vidéo, jeux électroniques, ordinateur, communications, etc.), pourrait bénéficier de cette conjoncture pour servir de champ d'expérience à de nouvelles approches pédagogiques.

Partant de cette constatation et en se basant sur les principes d'interaction de l'enfant dans sa phase de formation avec son environnement le plus direct et le plus immédiat, il devient opportun de réfléchir une nouvelle adéquation entre cette nouvelle réalité et une pédagogie susceptible de mieux servir la cause de la réforme de notre système éducatif.

Au début de son introduction à l'école, l'ordinateur a suscité d'énormes espoirs. Cependant ces espoirs ont été très souvent déçus car on avait voulu continuer à pratiquer la même pédagogie avec de nouveaux outils et face à de nouvelles difficultés. Or l'ordinateur de par ses performances, sa conception et les

1. Celestin FREINET,

2 ARISTOTE

3 "Behaviorisme": théorie des comportements (cf WATSON)

4 SKINNER

5 Jean PIAGET: " L'épistémologie génétique"

6 Seymour PAPERT: MIT et le laboratoire de LOGO

concepts que beaucoup de ses outils offrent (hypertextualité, hypermédia) est capable de mieux adapter la didactique au mode du raisonnement humain qui fonctionne plutôt par association d'idées que par linéarité et séquentialité. En effet, contrairement aux technologies anciennes, les nouvelles technologies semblent être plus capables de permettre à l'outil de verbaliser la pensée humaine.

"La science des ordinateurs, disait Herbert Simon en 1975 est une science expérimentale dans le but principal est la description du modèle de fonctionnement de l'esprit humain" D'ailleurs, l'ambition suprême des constructeurs d'ordinateurs semble être d'en faire un outil à l'image du cerveau humain. Et la terminologie adoptée à propos de cette technologie (Le cerveau électronique, la mémoire de l'ordinateur, l'intelligence artificielle) n'est qu'une succession de métaphores qui renforcent l'idée de parallélisme constant (depuis déjà 1945 et Van Newmann) qu'il y a entre d'une part, le fonctionnement de l'ordinateur et d'autre part, le fonctionnement du cerveau humain.

A l'école, cette analogie (ou correspondance) entre l'ordinateur et le cerveau humain va évoluer en faisant appliquer à la pédagogie comme dans d'autres domaines, l'évolution des concepts de l'informatique.

La phase d'automatisation et de mécanisation

Jusqu'aux années 70, l'informatique était limitée essentiellement aux langages de programmation et permettait peu de soutien aux utilisateurs. Par soucis d'automatisation et de mécanisation des tâches, des programmes ont été développés mais d'une manière figée et linéaire caractérisée par une "centration sur l'enseignant". Parallèlement, à l'école s'est développé l'enseignement assisté par ordinateur (EAO) basé sur des approches pédagogiques linéaristes, séquentielles et de juxtaposition tirée de la théorie Béhavoriste héritée de Skinner et de l'enseignement programmé.

La phase des systèmes intégrés et de communication

Depuis 1980, apparaît l'ère des concepts des systèmes intégrés (bases de données) qui permettent une meilleure cohérence et une disponibilité de l'information et ce grâce à la délocalisation de l'information et la possibilité de fusion intelligente et surtout sans redondance des différents modules constituants.

En pédagogie, ces concepts se sont traduits par une démarche dite activiste basée sur une intégration pédagogique des technologies de l'information selon les mêmes principes intégrateurs à travers lesquels l'apprenant s'érige en facteur agissant dans l'itinéraire de son apprentissage et la construction de son savoir faisant cette fois-ci une "centration" sur l'apprenant

Ces concepts favorisent un exercice d'apprentissage solitaire faisant appel à des ressources parfois distantes mais disponibles. Même si les résultats concrets durant cette phase n'ont pas été à la hauteur des espérances, il n'en reste pas moins vrai

que ce type d'apprentissage favorise nettement les aptitudes du savoir-faire chez les apprenants longuement revendiqué par les adeptes de la pédagogie active.

La phase de mondialisation

A partir de 1990, les possibilités d'intégration entre les activités de traitement et de manipulation des informations et celles de son transfert par les moyens de communications surtout ceux à large bande, ont permis de mettre en place des autoroutes d'informations qui assurent des transferts d'information sous toutes ses formes. Ceci a débouché sur les traitements distribués et dont les effets se traduisent par une forte demande d'accès venant de tous les secteurs par une gamme élargie de services et de produits informatiques. La mondialisation de l'information a supprimé les frontières et généré l'interopérabilité entre les systèmes qui a favorisé une participation collective entre individus anonymes interagissant grâce à un formalisme basé sur des aptitudes de savoir faire et de curiosités culturelles et scientifiques.

Dans ce contexte le savoir acquis ne résulte plus exclusivement de l'effort personnel de l'apprenant mais devient un concours pluriel, participatif et coopératif faisant éclater ainsi le huis clos de l'environnement d'apprentissage traditionnel faisant ainsi une "centration" sur le processus d'apprentissage. Ceci est de nature à élever le niveau de l'apprentissage vers le savoir être et même le savoir devenir

Conclusion

Cette évolution parallèle de la science vers la pédagogie ne fait que confirmer davantage le potentiel pédagogique apporté par l'informatique pour soutenir les différentes disciplines et qui permet de mettre l'école au diapason des nouveautés technologiques réduisant ainsi l'écart qui l'a longtemps frappée. Ainsi se justifie encore une fois l'apprentissage de l'informatique .

La question qui reste posée est de savoir si l'informatique devrait rester en tant qu'entité séparée ou si elle être intégrée transversalement dans les différentes disciplines. Je crois que la solution ne peut être tranchante car il est bien connu que l'informatique est un processus de mutation progressive plutôt que de changement. Tout dépend du degré d'intégration des technologies d'information dans la société en général et dans l'école en particulier. Aussi l'évolution de la didactique d'une discipline passe d'abord par une affirmation de son autonomie d'où son aspect discipline. Il est peut-être peu commode d'envisager la délocalisation des concepts de la discipline informatique dans les autres disciplines surtout quand elles se trouvent frappées par un cloisonnement. Avec l'évolution didactique que les disciplines sont amenées à faire, l'informatique pourrait constituer un facteur d'osmose interdisciplinaire et perdra son aspect de discipline à part.

Tahar Hafaied

Président de l'AFDI

Institut National de Bureautique et de Micro Informatique

18,bis rue de l'Inde, Tunis.Tunisie. Tél.: 834 476 Fax: 832 034

Conférence 2

L'informatique, science humaine ou inhumaine ?

Claude PAIR

présentée le mercredi 12 avril 1996 à 11h45

Introduction

Cette conférence inaugurale de la cinquième *Rencontre francophone sur la Didactique de l'Informatique* ne va pas parler de didactique, ni même d'informatique au sens où on peut le faire lorsque l'on se trouve au centre de cette science; il reste qu'elle sera francophone, et pour certains aspects centrée sur la France.

Ce n'est pourtant pas la première fois que je participe à un tel colloque. J'étais à Paris en septembre 1988, et même à un double titre: comme représentant du ministre de l'Education Nationale, et comme intervenant, sur un sujet tout à fait didactique: "*je ne sais (toujours) pas enseigner la programmation*"⁷. À l'époque, je ne le savais donc pas, malgré plusieurs années de réflexion "dure" sur les méthodes puis une teinture de psychologie cognitive⁸: j'ai d'ailleurs retrouvé ce dernier terme dans l'annonce de ce colloque. Je ne sais pas davantage aujourd'hui, peut-être parce que j'ai depuis abandonné ce champ d'activité: je vais sans doute l'apprendre ici. Je ne sais même plus s'il faut vraiment le faire: comme le demande le programme de cette rencontre, "*quelle informatique pour quel milieu ?*". D'ailleurs l'une de mes dernières communications concernant l'informatique posait la question sur un domaine plus large, qui me préoccupait fortement depuis quelques années et qui continue à le faire, celui de l'échec scolaire: "*can computer help combat school failure ?*"⁹.

J'ai complètement quitté l'informatique depuis cette année 88, et même depuis 1981 pour ce qui concerne son "noyau dur". Je suis devenu alors un de ces administrateurs de l'éducation qui parfois ne donnent pas les moyens nécessaires, ou peuvent même aller jusqu'à mettre en doute la nécessité de l'informatique à

⁷ Actes publiés par l'EPI, p. 75-86 et 301-303.

⁸ A. Ducrin, *Programmation*, Paris, Dunod, 1984.

Programmation, langages et méthodes de programmation, Le Travail humain, 51, no 4, 1988.

Programming. Programming Languages et Programming Methods, in *Psychology of Programming* (Hoc, Green, Samurçay, Gilmore, ed.), London, Academic Press, 1990.

⁹ avec D. Chartier, A. Flieller, M. Galloy, J.M. Gebler, M. Pillot, M. Quéré, *European Conference on Computers in Education*, Lausanne, juillet 1998, actes, Amsterdam, North Holland, p. 113-118.

l'école. J'ai donc beaucoup hésité avant d'accepter de vous parler aujourd'hui, et je ne me suis rendu qu'à l'insistance de Georges-Louis Baron qui a travaillé avec moi entre 1981 et 1985 pour mettre en place l'option informatique dans les lycées français.

Cette prise de distance peut expliquer mon titre: science humaine ou inhumaine? Je ne veux pas dire science dure ou science molle- même si, en passant, ces adjectifs, relativement récents dans ce sens, manifestent l'influence de l'informatique (*hard* et *soft*)- ni m'interroger pour savoir si l'informatique fait partie du secteur économique tertiaire ou du secteur secondaire, comme on l'a fait par exemple vers 1967 au moment de la création des IUT. Il ne s'agit pas de se demander si elle est une science *de* l'Homme, comme on dit au CNRS, mais si elle est une science *pour* l'Homme ou *contre* lui (au CNRS, elle est une science "pour l'ingénieur", ce qui n'est pas une réponse).

En fait, il existe une interaction entre l'évolution du monde, de la société, et celle de l'informatique depuis sa naissance il y a une cinquantaine d'années. Et une autre interaction entre l'évolution du monde et celle de l'éducation, qui doit nous intéresser puisque nous sommes ici des enseignants. Quant à l'informatique "pédagogique", dont la vie a duré à peu près la moitié de celle de l'informatique, elle est bien entendu liée à tout cela. Voilà de quoi je voudrais parler. De manière un peu arbitraire, je séparerai en deux parties: évolution du monde, avec la place qu'y a prise l'informatique; évolution de l'école, et informatique à l'école. Puis, en conclusion, je tenterai un coup d'oeil sur l'avenir.

L'évolution du monde depuis 50 ans

Les trente glorieuses

Il y a 50 ans, la seconde guerre mondiale venait de se terminer. Les besoins de la reconstruction exigeaient une relance de la production. Une période de croissance économique s'amorçait; elle allait durer 30 années, les "trente glorieuses". Il est intéressant de se demander pourquoi la croissance a alors été plus forte et plus durable qu'après la première guerre mondiale. On peut dire que le caractère global du conflit avait été plus marqué, provoquant un brassage d'hommes et une mondialisation des problèmes et des consciences, confirmée ensuite en Europe de l'ouest par le Plan Marshall, alors que 1918 avait au contraire été suivi par une période d'isolationnisme américain.

Mais en outre *l'évolution technologique*, également accélérée par la guerre, était d'une autre nature. Au départ, l'informatique n'y avait qu'une place bien modeste, mais déjà le progrès technique était fondé sur le développement des outils de communication et d'information. C'est pourquoi il a constitué un support pour *la mondialisation* - pensons à la télévision - et qu'en retour celle-ci a accéléré la diffusion de ses applications.

La période des "trente glorieuses" a été l'époque des convergences entre ces deux phénomènes -mutation technologique et mondialisation- et d'autres, de nature diverse, en une interaction poussant à une évolution cohérente dans un "cercle vertueux", au moins dans les pays développés puis dans certains pays en développement. Citons rapidement (voir schéma 1): croissance et automatisation de la production; enrichissement; augmentation de la consommation, qui prend une place de plus en plus grande dans la vie, se transforme en consumérisme et petit à petit fait passer au second plan le souci de produire, devant la préoccupation de soi, du bien-être, du corps; développement de la protection sociale; accroissement du rôle des pouvoirs publics; complexification des organisations, entreprises et états notamment; augmentation et diversification de l'emploi, avec dans un second temps recul du nombre des ouvriers -producteurs directs- mais développement du secteur tertiaire; augmentation de l'activité féminine; développement aussi des emplois intermédiaires, donc du point de vue social, des classes moyennes; diminution des inégalités de toute sorte; démocratisation; montée de l'éducation et de la formation.

Ce n'est pas que cette période de croissance ait été sans problèmes. Mais ce qu'on peut dire avec le recul, c'est qu'ils ont été absorbés sans trop de difficulté. Il en est ainsi pour la régression de l'emploi agricole - fruit de l'automatisation - grâce à la croissance générale de l'emploi; ou de l'urbanisation et des difficultés de logement qu'elle a créées. Il en est encore ainsi de l'émancipation coloniale - conséquence de la mondialisation - et des guerres qui l'ont accompagnée: elle a conduit à une égalité formelle entre les pays plus qu'à une diminution des inégalités économiques; car elle a été accompagnée de nouveaux moyens de domination des pays développés grâce au processus de mondialisation de l'économie.

Et si la crise pétrolière a marqué la fin de la période de croissance, elle a finalement été bien absorbée, elle aussi, par le développement de l'énergie nucléaire, les économies d'énergie, le recyclage des pétrodollars, l'action diplomatique, politique et militaire, tous ces éléments étant fondés sur la mondialisation et sur les techniques de l'information et de la communication: là, l'informatique et ses applications se sont bien trouvées au premier plan. Quant au vieil obstacle que constituait la séparation du monde en deux blocs, cela a mis un peu plus de temps, jusqu'en 1989, mais il n'a pas non plus résisté à la mondialisation et à tout ce qui l'accompagne, en particulier l'appétit de consommation et de liberté. On pourrait encore parler des événements de 1968. Eux ont manifesté la contradiction entre la volonté de consommation, l'exaltation de l'individu, le souci du corps, et des organisations anciennement établies ou devenues complexes, considérées comme répressives (état, armée, églises, université, entreprises). Cela aussi a été absorbé par une recrudescence de la consommation et un développement de la liberté.

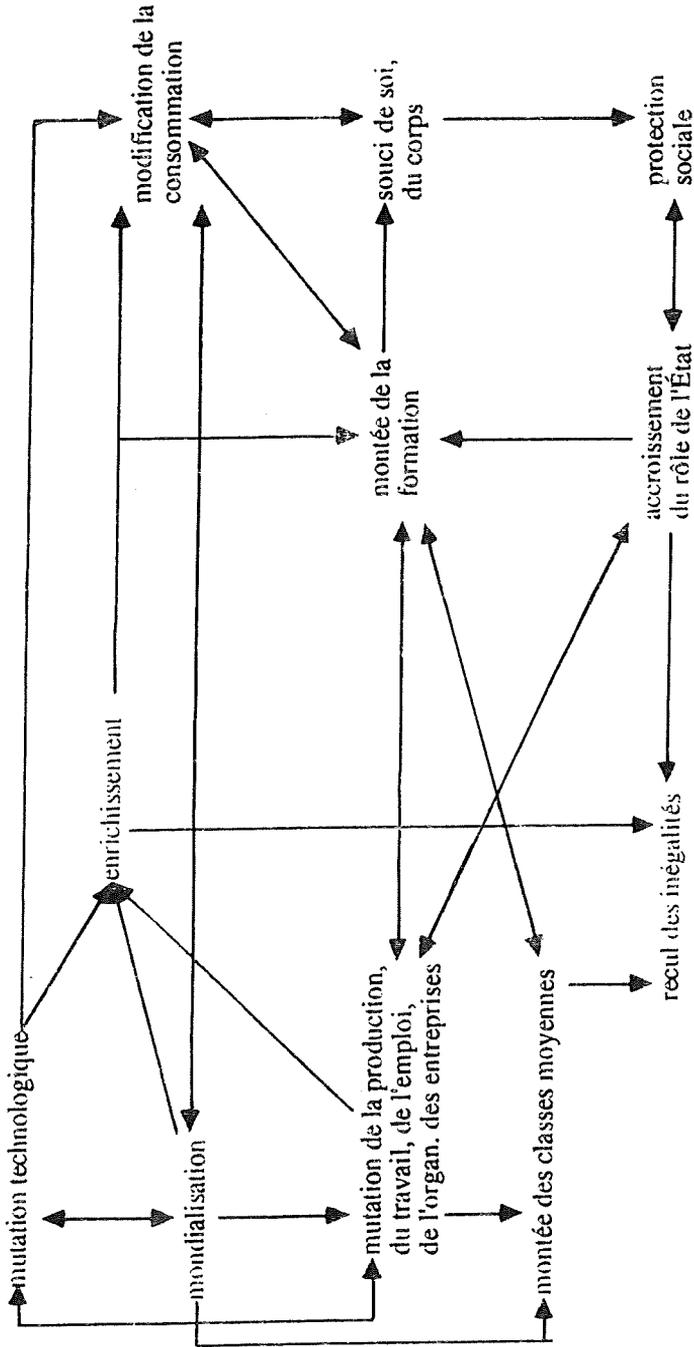


schéma 1

Un desserrement des contraintes

Au fond, ce développement de la liberté sous diverses formes, au sens d'un relâchement des contraintes permis par l'évolution technologique, est sans doute au cœur de tous les phénomènes qui se sont produits et renforcés mutuellement au cours de la période de croissance. Liberté des mœurs, avec les techniques de contraception; liberté dans la consommation dont un symbole est le "libre-service"; déplacement des contraintes dans le travail par une relation moins directe avec la production, grâce à l'automatisation où progressivement entre de plus en plus d'informatique: suppression de l'effort physique, passage de l'application de normes à la réaction à l'imprévu; liberté de temps et d'occupation puisque moins de temps est nécessaire pour produire, d'où la civilisation des loisirs mais aussi le développement du secteur tertiaire, c'est-à-dire de l'action sur les hommes et les groupes humains.

En effet, cette liberté plus grande ne s'applique pas seulement aux individus, mais aussi aux organisations. En particulier, l'organisation des entreprises se transforme considérablement au cours de cette période. Dans un premier temps, un traitement de l'information devenu plus facile et plus économique permet un accroissement de taille et de complexité, répondant à la croissance de la production: l'emploi se diversifie, mais l'organisation demeure hiérarchique et taylorienne, selon le modèle né de la révolution industrielle au dix-neuvième siècle et au début du vingtième. Cependant l'avantage de cette organisation, qui est d'être la plus économe en transmission de l'information, diminue avec la baisse des coûts de transmission et d'autres possibilités deviennent possibles. En outre, les organisations peuvent se déployer plus facilement dans l'espace grâce à la rapidité des communications physiques et électroniques: la localisation des activités économiques n'est plus liée à la proximité des matières premières et de l'énergie, mais bien plus à l'existence, au coût, à la qualité de la main-d'oeuvre et à sa formation; on assiste à une mondialisation croissante de l'économie. C'est donc bien un desserrement des contraintes dans les choix d'organisation qui apparaît.

On voit aussi l'influence de l'informatique, ou plutôt l'interaction avec elle. Ses premiers pas sont issus de recherches visant des applications militaires et demandant des calculs mathématiques: son influence sociale est alors limitée. Mais très vite, dès les années 50, elle sort des laboratoires pour commencer à intervenir dans l'administration et la gestion d'organisations qui deviennent plus complexes, ce qui incite à des transformations techniques, notamment sur la taille et la diversité des mémoires ... à moins que ce ne soit l'inverse et que les progrès techniques aient suscité ces nouvelles applications. Mais l'exécution des programmes en "traitement par lots" - c'est-à-dire de manière centralisée avec un différé de plusieurs heures entre le dépôt d'un paquet de cartes et l'obtention du résultat⁴ - ne suffira pas longtemps: ici aussi s'opérera un premier desserrement des contraintes, vers 1970, sous la forme d'une interactivité avec la machine. Et cette nouvelle innovation technique ouvrira la porte au "temps réel" qui permettra

⁴ Assez souvent, voire la plupart du temps, "fatal error".

l'introduction de l'informatique dans une automatisation naissante de la production. Un peu plus tard, vers 1980, s'achèvera la période de la centralisation informatique, avec l'apparition du micro-ordinateur, la baisse accélérée des prix, une facilité croissante d'utilisation; d'où une diffusion considérable et l'ouverture au grand public, donc la fin de l'exclusivité des informaticiens.

Mais le changement n'est pas seulement technique. Il s'agit aussi, et c'est peut-être plus important, d'un changement dans les idées, dans les têtes. Du côté des sciences de base, si au moment de la révolution industrielle la science dominante était la physique, science du matériel concevant la matière comme formée d'unités hiérarchisées et indifférenciées, pendant la période de croissance va se développer la biologie; or, contrairement à la physique, la biologie met l'accent sur la variété, l'environnement et l'incertitude, donc sur l'adaptation. Mais le développement industriel incite au développement d'autres sciences, nées des techniques, des sciences de l'artificiel visant l'action et pas seulement la compréhension: mécanique, électrotechnique, électronique, automatique; dans la liste précédente, qui suit en gros l'ordre historique, on peut constater que l'aspect matériel a tendance à s'affaiblir. Quant à l'informatique, née de la technique de l'ordinateur, nous savons bien qu'elle est d'abord une science de l'information; elle porte moins sur des matériels, comme la mécanique, que, comme la mathématique, sur des êtres de raison mais qui pour elle s'incarnent dans des programmes; la notion est neuve et, pour qualifier cet immatériel incarné, on a même dû inventer un nouveau mot: "logiciel". Ce passage du matériel au logiciel est aussi une forme d'affaiblissement des contraintes.

La crise

Cependant, l'accroissement des degrés de liberté va remettre en cause les convergences entre certains des phénomènes envisagés plus haut et conduire à ce que l'on a appelé la crise, qui est bien loin d'être seulement économique (voir schéma 2).

Pour les entreprises, un premier degré de liberté consiste à remplacer l'homme par la machine, ce qui accroît la productivité, et ceci bien au-delà de la production proprement dite, dans la surveillance des processus de production et le secteur tertiaire, c'est-à-dire là où les emplois s'étaient déplacés. Ainsi, la croissance du nombre d'emplois, forte dans les années cinquante et soixante caractérisées par le taylorisme, se ralentit, puis devient plus faible que celle de la population active, le chômage apparaît donc, on assiste de plus en plus à des suppressions de postes de travail, jusqu'à une diminution nette qui dans un certain nombre de pays marque le début des années 90.

L'informatique, d'ailleurs, conduit à créer des "modèles" des réalités et, parmi les réalités modélisées, figurent les organisations humaines, voire l'Homme lui-même avec l'intelligence "artificielle": c'est en ce sens une science de l'Homme. Mais les pionniers de l'informatique, à cause de ses premières origines, viennent des sciences "dures", mathématiques et physique, et ce ne sera pas sans influence

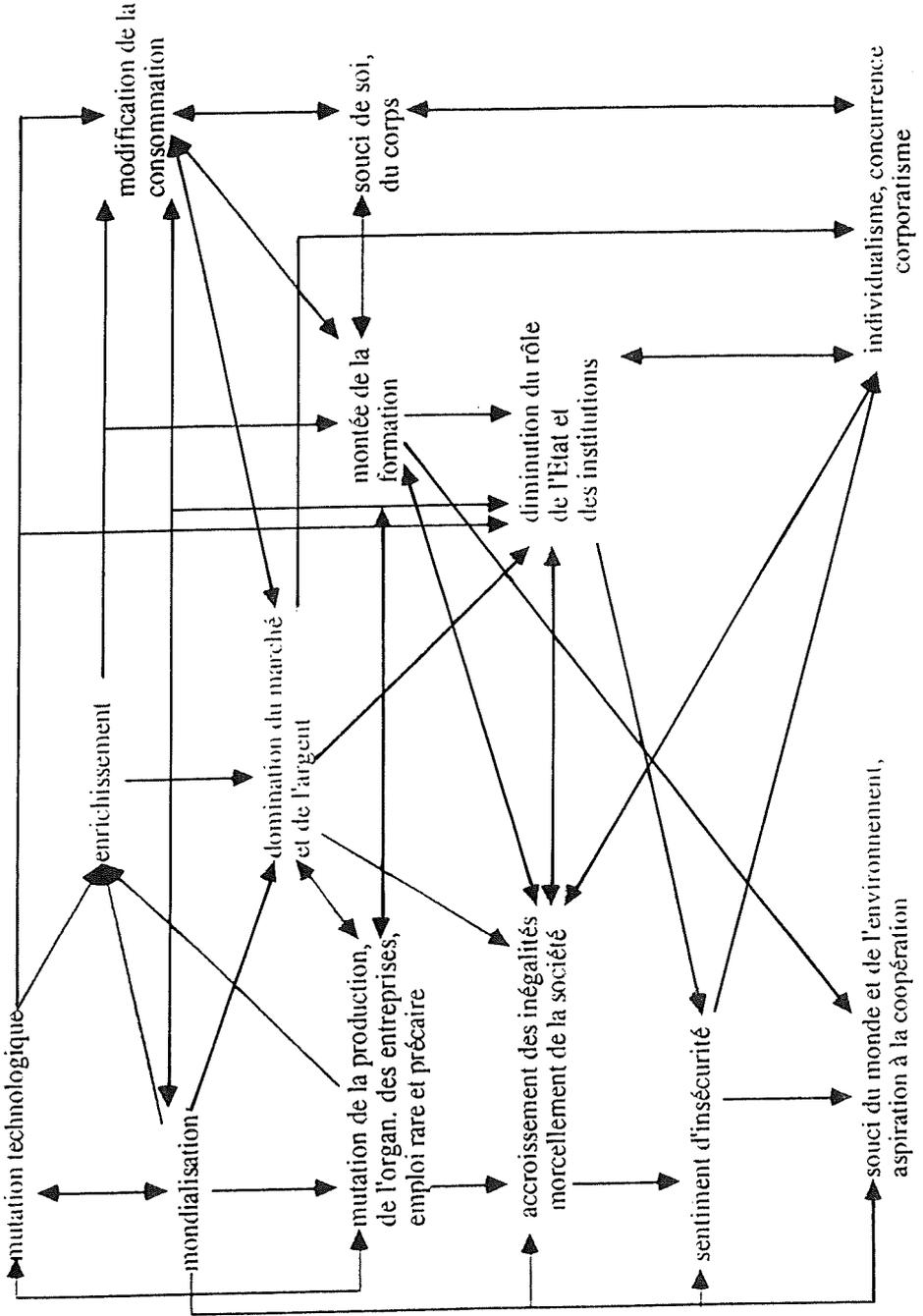


schéma 2

sur son développement. D'ailleurs modélisation implique rationalisation et simplification des problèmes au service d'un objectif. Pour les entreprises, dans une période de concurrence internationale croissante, il s'agit de la baisse des coûts de production.

En outre, la perte des contraintes crée l'incertitude, surtout lorsqu'elle s'exerce au plan mondial. Et, dans un second temps, autour de 1990, les entreprises sont amenées à utiliser leur liberté pour rechercher l'adaptabilité, la souplesse, la flexibilité; d'où une remise en cause de la complexité et de l'intégration, par un éclatement en cercles concentriques -cœur, filiales, sous-traitants- et un éclatement parallèle de l'emploi: contrats à durée indéterminée, à temps complet mais aussi partiel, contrats à durée déterminée, intérim, stages, travail indépendant, voire travail au noir. Le travail à temps plein, à statut, à durée indéterminée perd son caractère dominant.

Tout cela conduit, non plus à une montée des classes moyennes comme à l'époque des trente glorieuses, mais à un accroissement des inégalités dans l'usage de la technique, l'emploi, la richesse, la consommation, l'habitat, la protection, avec divers degrés selon le domaine. Très partagée est l'information télévisée, mais mondialisée (ou américanisée) et aussi source de pouvoir mais pour un nombre restreint de personnes; inégalement partagés, selon le pays, la génération et le milieu social, sont le travail, la certitude du lendemain, l'argent, la consommation, la mobilité géographique; inégalement partagé aussi, notamment selon les pays, est l'accès à la technique, en particulier pour l'informatique malgré sa diffusion; peu partagée est l'information interactive (Internet ...) à l'échelle mondiale, source d'un pouvoir qui, lui, est encore moins partagé bien que là encore on puisse distinguer des degrés. Un morcellement de la société se dessine ainsi, avec au sommet une "super-élite" qui vit au plan mondial (la "jet society", mais sans doute aussi les scientifiques; si, grâce à elle, les frontières s'atténuent entre les pays, elles se recréent dans chacun d'eux et les décideurs -mais aussi les scientifiques- sont parmi ceux qui sont certains de leur avenir ou au moins les mieux protégés. En réaction se développent l'individualisme, lié au souci de soi et à la recherche de la liberté, et le corporatisme.

La question se pose alors des règles qui limiteraient le pouvoir des puissants. Or, les Etats, bien qu'ils se soient renforcés pendant la période de croissance, n'ont pas la possibilité de réguler des phénomènes mondiaux, et l'ont d'autant moins que leur territoire et leur zone d'influence sont plus restreints. Le seul régulateur possible devient alors le marché, c'est-à-dire l'argent et notamment le marché financier; l'argent n'est plus là seulement pour permettre les échanges, mais comme bien en soi, qui se produit lui-même et qui produit la puissance. Le néolibéralisme -puisque c'est de cela qu'il s'agit -a en Europe submergé la Grande-Bretagne et les pays de l'Est, mais il atteint aussi les autres pays; dans le monde, il triomphe presque partout, en commençant par les pays dont l'économie est faible. Il conduit au recul de l'état en matière économique, puis à celui de la décision démocratique,

prise entre les corporatismes et le sentiment qu'ont les élites de connaître l'unique solution imposée par les marchés, enfin au désenchantement des citoyens.

Cette diminution du rôle de l'état accroît encore l'incertitude, d'autant plus qu'elle s'accompagne de celle des autres institutions qui étaient facteurs de stabilité et d'intégration sociale. Partis politiques, syndicats, armée, entreprises voient reculer leur influence; il en est de même pour les religions: consommation, souci de soi et du corps, individualisme, abondance d'informations et mondialisation leur font une rude concurrence et battent en brèche leur vocation à l'universel; beaucoup de leurs membres les abandonnent et d'autres se replient sur des règles formelles considérées comme sûres et intangibles -l'intégrisme- ou la proximité chaleureuse -les sectes.

D'ailleurs, plus généralement, l'incertitude, l'appel constant au changement, la mondialisation de l'information, la montée des inégalités, le recul de l'état, nourrissent un sentiment d'insécurité, notamment dans des classes moyennes qui s'inquiètent de leur avenir: autre rupture dans l'évolution de la période précédente. Il est souvent lié à un sentiment d'impuissance, mais heureusement il crée aussi chez un certain nombre de gens un souci du monde, de l'environnement, de l'exclusion, et une aspiration à la coopération qui entrent en tension avec l'individualisme et le corporatisme.

L'informatique pour ou contre l'Homme ?

Au terme de ce parcours, on peut se poser la question. L'informatique se préoccupe du travail de l'Homme, comme toutes les sciences de l'artificiel nées des techniques. Son objet est de faire réaliser par la machine des tâches qu'il effectuait autrefois, ou au moins de l'assister dans son travail, de le partager avec lui: nous parlions autrefois, avec G.L. Baron, de l'ordinateur "auxiliaire de pensée et d'action"; l'association des deux termes différencie l'informatique des autres sciences de l'artificiel. L'informatique est donc bien au départ une science *pour* l'Homme au sens où elle est en grande partie motivée par une volonté de diminuer l'effort, le temps passé, la fatigue, l'ennui né de la répétition, et aussi d'étendre le rayon d'action de l'Homme dans l'espace et dans le temps.

Mais quel est le résultat ? Il s'agit d'abord du déclin de la notion de métier, si on la caractérise par: force, adresse, répétition, démarche largement implicite. Certes, ce phénomène n'est pas nouveau: dès l'époque de la révolution industrielle au dix-neuvième siècle, les hommes de métier se sont trouvés en concurrence avec les machines. Ce qui est nouveau, c'est qu'on a maintenant cessé de passer du métier et de la qualification aux postes d'exécutants non qualifiés des systèmes tayloriens. Au contraire, on est allé vers la professionnalisation et l'expertise, avec comme caractéristiques: réaction à l'imprévu et à l'indéterminisme, compétence, décision, adaptation, création, liberté, responsabilité, relations humaines.

Mais, même sur l'expertise, il y a maintenant concurrence, comme en témoigne le perfectionnement continu des "systèmes-experts". Ce qui reste à

L'Homme se situe à un niveau de complexité ou d'imprévu de plus en plus élevé : il existe une course entre Homme et machine, ou plutôt entre les hommes de divers secteurs, en particulier des producteurs au sens classique, dans le secteur industriel et dans le secteur tertiaire, avec les producteurs de logiciel et de communication. Or l'idéologie dominante en termes de recherche dans les sciences de l'artificiel et en termes d'organisation reste d'économiser le travail de l'Homme, alors que le travail passe de la malédiction au privilège. C'est en ce sens qu'on peut dire que l'informatique est inhumaine par ses conséquences.

Car cette idéologie est dévastatrice pour l'emploi parce que nous ne sommes plus au temps de Jacquard ou de Ford. L'évolution scientifique et technique fait que la production de richesse est de moins en moins liée linéairement à la quantité de main d'oeuvre; la notion de productivité -qui exprime cette linéarité- s'évanouit avec le changement du rôle de l'Homme: de la production à la présence, à la surveillance, à la création d'informations. Il n'est pas étonnant que, dans beaucoup de secteurs, l'évolution technique conduise à une raréfaction du travail, contrairement à ce qui se produisait lorsque production, emploi, consommation étaient plus fortement liés. Et que les conséquences des suppressions d'emploi ne se voient que par seuil, à terme ou par une rétroaction du niveau macro-économique (par exemple à travers des coûts induits par le chômage).

C'est que la richesse "information" est de nature bien différente des autres. Un logiciel n'est pas une plaque de chocolat ou un réfrigérateur car sa reproduction ne coûte pratiquement rien. Souvenez-vous de l'aphorisme: "si nous avons chacun un dollar et que nous les échangeons, nous avons toujours chacun un dollar; mais si nous avons chacun une idée et que nous les échangeons, alors nous avons chacun deux idées". A la place de "dollar", je peux mettre "plaque de chocolat" ou "réfrigérateur"; à la place de "idée", je ne peux pas mettre "dollar" ou "euro", mais je peux mettre "logiciel" ou "fichier".

En outre, cette différence de nature -qui s'est manifestée par exemple sur les problèmes de copie de logiciel- rend difficile le choix entre dissémination et raréfaction de l'information, ce qui peut opposer technique -une certaine rareté pour faire payer- et science, qui se nourrit de l'échange d'idées: ce qui s'est passé à propos du Sida entre les laboratoires Montagnier et Gallo en est une illustration. De la même façon, elle peut poser des problèmes à l'éducation, traditionnellement gratuite.

En parallèle, l'évolution de l'école

Ici, je vais me fonder sur le cas français, mais je crois que l'on peut transposer pour d'autres pays.

L'école "républicaine", dans une société stable

À la fin de la guerre, l'école est encore celle de la Troisième République. Le système scolaire calque la dichotomie d'une société partagée entre une élite peu nombreuse, largement héréditaire, et une masse de paysans, d'artisans, de petits

commerçants, d'ouvriers, d'employés. Il se compose pour l'essentiel de deux écoles parallèles: l'enseignement primaire destiné à la masse, avec son prolongement primaire supérieur; l'enseignement secondaire pour l'élite sociale, commençant dès les classes maternelles. L'école primaire s'est construite pour cimenter la République en dépassant les particularismes, par la transmission de connaissances et de comportements; l'enseignement secondaire a pour objectifs de faire acquérir des savoirs établis et les signes distinctifs de l'élite. Cela correspond au modèle, à l'idéologie, que J.L. Derouet⁵ nomme *civique*, fondé sur la *transmission de savoirs* considérés comme universels voire éternels, dans cette transmission sur la primauté de disciplines bien établies et plutôt tournées vers le passé, sur la clôture de l'école, sur l'uniformité, sur la méritocratie qui justifie la séparation des deux ordres d'enseignement, avec l'alibi que constitue la passerelle du primaire vers le secondaire: on peut parler de "l'élitisme républicain".

Ecole de la croissance et croissance de l'école

Les transformations économiques et sociales des trente glorieuses conduisent à un allongement de la scolarité, avec notamment un accès à l'enseignement secondaire pour tous, la mise en place progressive du Collège, décidée complètement en fin de période, la complexification du "système éducatif" (l'expression apparaît à cette époque), la naissance de la formation continue. Inversement, cette montée de la formation retentit sur les changements sociaux, et elle sera déterminante pour provoquer les mouvements de 1968 qui partent des universités. Le modèle éducatif qui naît alors est suscité par la démocratisation qui fait prendre en compte le phénomène de l'échec scolaire et les observations des sociologues (Bourdieu...) mettant en évidence l'aspect social de la méritocratie: c'est celui que J.L. Derouet nomme *domestique*, qui se centre sur l'élève comme individu, sur ses particularités, sur son épanouissement, sur l'apprentissage de la vie avec les autres. Le mot *éducation*, plutôt qu'*instruction*, est bien caractéristique de ce second modèle⁶.

La concurrence est dure avec le modèle "civique" qui reste très fort voire dominant chez les enseignants, avec le changement que ce sont les disciplines scientifiques qui prennent le pouvoir; cette concurrence explique les résistances au changement qui se manifestent alors et qui se poursuivent encore; mais sur le point de la relation avec l'environnement social et surtout économique, ces deux modèles convergent dans l'indifférence ou même le rejet. Pourtant à l'extérieur de l'école apparaissent de nouveaux signes: une demande accrue d'éducation de la part d'une partie des milieux économiques, soucieux de disposer d'un réservoir de main-d'oeuvre qui leur permette croissance et flexibilité; la recherche par les pouvoirs publics, à travers des techniques de planification, d'une adéquation de l'éducation aux besoins de l'emploi.

⁵ *La profession enseignante comme montage composite : les enseignants face à un système de justification complexe*, Education permanente no 96, 1988.

⁶ même si le changement de nom du ministère date de 1934.

Pour l'informatique, elle se développe au cours des années 60 dans l'enseignement supérieur pour former des spécialistes. En revanche, malgré quelques tentatives dans l'enseignement technique, elle n'apparaît que très peu dans les établissements scolaires jusqu'à que les mini-ordinateurs et l'interactivité la rendent directement accessible. Les caractéristiques du modèle civique ne le rendent pas favorables à son introduction à l'école. Le modèle domestique lui est moins opposé, pour autant qu'il s'agisse de favoriser le développement et l'épanouissement des jeunes. C'est cet aspect qui ressort en 1970 du premier colloque international organisé par l'OCDE à Sèvres⁷, intitulé "*l'informatique à l'école secondaire*". La phrase qui a été relevée le plus souvent dans le rapport du colloque est: "*introduire l'informatique dans l'enseignement permettrait de développer chez les élèves des aptitudes algorithmiques, opérationnelles, organisatrices*"⁸. Il s'agit donc bien d'une vue transversale et culturelle de l'informatique à l'école, et en même temps d'une attitude un peu "missionnaire", visant à apporter quelque chose de nouveau à l'éducation. En France, la balle est reprise au bond et il est décidé, par W. Mercouroff conseillé par J. Hebenstreit, d'expérimenter l'introduction de l'informatique dans les lycées généraux, non pas comme une nouvelle discipline qui transmettrait des connaissances nouvelles, mais à travers toutes les disciplines pour faire acquérir des attitudes d'esprit, des capacités transversales.

Crise économique et accélération de la croissance scolaire

La crise économique renforce les attentes vis-à-vis de l'école. De tous les côtés, un appel explicite lui est lancé: elle est en effet perçue par les employeurs comme permettant une compétitivité accrue et comme nécessaire pour rendre possibles les transformations des entreprises, par les individus comme une assurance contre le chômage, par les pouvoirs publics comme pouvant retarder l'entrée sur un marché de l'emploi difficile; il y a là pour les enseignants et les responsables de l'éducation une divine surprise qu'ils sont tout prêts à ratifier. La convergence des causes se renforce pour aboutir à une croissance, voire une explosion, de la scolarisation, qui contraste avec la stagnation de l'économie. Cependant les exigences de compétitivité et de qualité, comme l'intérêt croissant porté aux entreprises par la société dans son ensemble, imprègnent l'Education Nationale, ce qui fait naître un nouveau modèle que J.L. Derouet nomme *industriel*, ouvert sur l'extérieur, qui se préoccupe de *l'insertion professionnelle et sociale* des jeunes et dont les maîtres-mots sont objectif, projet, travail d'équipe, évaluation, efficacité, ouverture, rapprochement avec les entreprises,... et formation, notamment formation permanente. La visée est une amélioration de la compétitivité des entreprises et une baisse du chômage et il paraît légitime que la formation s'adapte à ces objectifs.

⁷ La même année, l'IFIP organisait sa première *World Conference on Computer Education*, à Amsterdam.

⁸ Rapprocher de l'annonce de ce colloque : "*l'apprentissage de l'informatique favorise (...) l'acquisition d'outils intellectuels et le développement d'habiletés cognitives*".

En particulier, l'opinion et les décideurs s'inquiètent de ce que les "nouvelles technologies", et au premier rang l'informatique, soient si peu présentes à l'école. Pour la France, l'expérience d'introduction dans les lycées avait été suspendue en 1976: le vent tournait avec le début de la crise qui rendait plus difficiles les investissements et moins attirante la recherche pédagogique; il était d'ailleurs impossible d'envisager une généralisation étant donné le coût de l'équipement à cette époque; mais en outre apparaissait une incertitude sur les objectifs, avec une dérive par rapport aux buts initiaux, vers un EAO de qualité médiocre. Cependant, la technique crée le micro-ordinateur, moins coûteux et plus facile d'emploi. Très vite de nouvelles idées se font jour avec le rapport Nora-Minc, *l'informatisation de la société* (1978), et celui de J.C. Simon, *l'éducation et l'informatique dans la société* (1980)⁹. Mais le point de vue est devenu plus économique que culturel, donc bien conforme au modèle "industriel" qui naît pour l'éducation. Un signe: l'introduction, décidée en 1979, de 10 000 micro-ordinateurs dans les établissements scolaires -chiffre considérable à l'époque- est pilotée par le ministère de l'industrie, sans grandes préoccupations pédagogiques. On oscille entre l'informatique-outil d'enseignement, marquée par le modèle industriel, et l'informatique-discipline (point de vue de J.C. Simon), qui pourrait se rattacher à une modernisation du modèle "civique": j'ai remarqué qu'un écho de cette distinction va se retrouver ici.

Un compromis est trouvé en 1981, après l'étude que le ministre A. Savary confie à Y. Le Corre et à moi-même: on définit un développement progressif pour tous les niveaux d'enseignement, concerté entre le ministère, les académies, les établissements; une option informatique est mise en place dans les lycées à titre expérimental. Cependant cette prudence ne satisfait pas les zélateurs de la modernisation. D'ailleurs l'informatique domestique continue à se développer et il devient possible d'annoncer en 1983 un plan de mise en place de 100 000 ordinateurs et de formation de 100 000 enseignants: ce sera réalisé dès 1986 par le plan "Informatique pour Tous", même si le matériel -d'ailleurs assez peu fiable- a, cette fois encore, pris le dessus sur la réflexion et la formation.

Le temps des divergences

Les trois "modèles d'école" dont nous avons parlé -"civique", "domestique", "industriel" - se combinent chez les enseignants et dans l'institution pour construire une identité. La Loi d'Orientation du 10 juillet 1989 en fait une synthèse, mais le dernier y est bien présent et on peut considérer qu'elle conclut une époque. Car dès l'année suivante se produisent des troubles dans les lycées qui, comme ceux de 1968, ont valeur d'alerte: la belle convergence dont nous avons parlé entre les employeurs, les jeunes et leurs familles, les pouvoirs publics est en train de disparaître. C'est que des deux objectifs d'amélioration de la compétitivité et de baisse du chômage, seul le premier est atteint, même si on peut soutenir que, pour le second, la situation serait pire si la formation ne s'était pas développée comme elle l'a fait.

⁹ tous deux publiés par la Documentation Française.

Autour de 1990, beaucoup d'employeurs affirment ne pas trouver les personnes qui pourraient satisfaire leurs offres; aujourd'hui, ils considèrent volontiers que les jeunes sortant des établissements scolaires ont besoin d'être reformés avant d'être utilisables : visiblement, une élévation générale du niveau de formation ne suffit pas, c'est une notion trop vague. Les jeunes, eux, constatent que la mobilité sociale n'existe plus guère et que la formation n'est plus une garantie contre le chômage: beaucoup ont le sentiment d'être pris dans un piège. Les enseignants voient les phénomènes d'exclusion sociale s'aggraver: une partie de la population scolaire est dans une situation qui lui rend difficile de trouver un sens à l'école, et une meilleure réussite de la majorité creuse encore davantage le gouffre avec ces "exclus". Et devant ces divergences, les pouvoirs publics sont incertains dans leur action: on pourrait citer de multiples exemples, comme la "rénovation" des lycées en 1991-93, les hésitations sur le collège, l'affaire du contrat d'insertion professionnelle en 1994, celle de la poursuite d'études des titulaires des IUT en 1995 ...

Pour l'informatique pédagogique, les incertitudes ont commencé auparavant. Les résistances internes s'appuient sur des interrogations concernant son intérêt pour le développement cognitif, toujours difficile à appréhender, comme pour la préparation à la vie professionnelle dans un contexte d'évolution technologique accélérée où on recherche plutôt des compétences personnelles et relationnelles. En France, dès 1986, il est question de réduire son domaine à la documentation et à la simulation, puis la rénovation des lycées voit la suppression de l'option. Plus généralement, du congrès organisé par l'UNESCO en avril 1989 ressort un certain désenchantement: un peu partout, il n'existe plus guère d'impulsion forte et on constate un manque de visibilité sur ce qui se passe réellement dans les établissements scolaires.

Ces difficultés peuvent avoir deux causes. La première est liée à l'évolution de l'informatique. Son principal moteur est de simplifier l'usage de la technique de l'ordinateur ; de sorte qu'elle-même détruit la motivation à ce qu'on l'apprenne : elle ne dévore pas seulement l'emploi, elle se dévore elle-même. Aujourd'hui, l'informatique n'est plus un langage, comme elle pouvait le paraître en 1970, et comme le sont d'autres disciplines, mathématiques ou langues vivantes. Elle n'est pas (encore ?) non plus un recueil de savoirs bien identifiés, comme la physique ou la géographie. Et son caractère de démarche de pensée n'est plus tellement apparent. Bien sûr, la construction d'un programme demeurerait formatrice, comme on le proclamait dans les années 70, mais quelle motivation reste-t-il pour la pratiquer alors qu'elle ne permet au débutant que des usages de l'ordinateur rudimentaires et peu attrayants au regard de tout ce qui est aujourd'hui possible avec facilité ?

La seconde cause renvoie à un phénomène plus général: c'est le recul du rôle de l'état dont nous avons parlé plus haut, accompagné de son appauvrissement. En France, le pilotage ministériel avait joué un rôle important jusqu'en 1986: son

affaiblissement prend prétexte de la décentralisation qui confie les investissements aux collectivités territoriales; il achève donc une dissociation, qui était déjà apparue à plusieurs reprises, entre pédagogie et équipement.

Les trois modèles à l'épreuve

En fait, les trois phénomènes majeurs que nous avons mis en évidence plus haut -précarité de l'emploi et chômage, accroissement des inégalités, recul de l'état- heurtent chacun l'un des trois modèles qui se combinent aujourd'hui pour constituer l'identité des enseignants et de l'école, tout au moins en France.

Le recul de l'état et le libéralisme heurtent le modèle "civique": on n'en a pas encore trop pris conscience en France, parce qu'on pas véritablement vu de conséquences financières, la part de la dépense d'éducation dans le PIB ayant continué à croître. Mais il existe des signes prémonitoires: pas seulement la décentralisation, qui reste modeste et ne fait que modifier l'équilibre entre les pouvoirs publics, que les difficultés à prendre des décisions face aux corporatismes, ou encore la volonté récurrente de développer l'apprentissage. Et puis, il est intéressant de regarder ce qui se passe dans d'autres pays. De plus, toutes les institutions sont aussi remises en question et l'école en est une, même si elle est jusqu'à présent moins remise en cause que d'autres, sauf peut-être sous la forme du consumérisme.

Les inégalités qui se creusent posent problème au modèle "domestique", étant donné son origine. Certes on peut dire qu'il a préparé l'école à prendre en compte les diversités, ce qui explique qu'elle tienne mieux que d'autres institutions là où les inégalités éclatent le plus durement. Mais les difficultés deviennent aiguës à cause du morcellement social et elles sont renforcées par la domination de l'argent, la consommation, l'individualisme, le recul des autres institutions éducatives et la coupure entre jeunes et adultes liée en particulier à l'augmentation de la durée de la scolarisation.

Le défi créé par l'emploi est peut-être plus fort encore parce qu'il heurte le modèle le plus récent, celui sur lequel l'école a le plus fonctionné dans la dernière décennie. Il n'est plus possible de s'organiser comme si était encore valable l'idée, née pendant la période de croissance, d'une correspondance formation -examen-diplôme -classification- emploi, avec embauche en fin d'études et emploi pour la vie même si on ajoute qu'il faudra s'adapter quelque peu par la formation continue. Ce défi met en cause la formation professionnelle initiale, à l'école comme en apprentissage. Surtout, il casse le ressort sur lequel a fini par se construire l'école: "travaille pour avoir un bon métier et une vie agréable", devenu ensuite: "si tu ne travailles pas, tu n'auras rien dans la vie". Plus profondément encore, la valeur "travail", qui reste à la base de l'école, est en chute libre dans la société ... telle que l'informatique a contribué à la transformer.

Et demain ?

Tendances et scénarios

Des phénomènes aussi imbriqués ne s'arrêtent pas facilement, sauf catastrophe (au sens de R. Thom comme au sens banal) peut-être pas si improbable. L'évolution technologique se poursuivra, comme le fera la mondialisation, élément dominant, comme encore la rareté du travail qu'elles entraînent et qui a toujours été le lot de la majorité des habitants de la planète. Le scénario tendanciel, renforcé par la difficulté de faire évoluer la société par consensus, est celui de la domination du "marché", le rôle de l'état continuant à se réduire et les inégalités à croître. L'évolution peut cependant être plus brutale et conduire à un second scénario, celui de la rupture, sous l'effet des tensions provoquées par les inégalités et le morcellement de la société, conduisant à des événements graves, avec remise en question de la démocratie, terrorisme généralisé ou guerre mondiale.

On peut pourtant espérer un troisième scénario, où nous apprendrions à vivre ensemble dans le nouveau contexte. Il s'agit d'abord de résister au libéralisme en dépit de la mondialisation, mais on doit prendre conscience que c'est difficile, et même impossible pour un pays isolé. Un objectif serait, face à la rareté de l'emploi, de lutter contre la précarité, et plus généralement de maîtriser les inégalités, sur le plan national et international. Elles sont en effet facteur de rupture, mais il existe aussi d'autres raisons, liées à la croissance économique et à l'emploi: aujourd'hui, l'enrichissement, qui se poursuit dans notre monde, ne crée guère de consommation mais plutôt de la spéculation, non pas faute de besoins mais faute d'une bonne répartition des ressources; la création d'emplois "de proximité" se heurte à leur caractère considéré comme subalterne et le partage du travail à l'idée que le pouvoir ne saurait se diviser; quant aux décisions de suppression d'emplois, elle sont prises par une hiérarchie qui ignore de quoi elle se prive. Or l'organisation hiérarchique que nous connaissons est née du coût des communications et de la rareté de la formation, deux éléments aujourd'hui largement dépassés. Ne peut-on envisager des hiérarchies multidimensionnelles, où personne ne serait dominé dans tous ses rôles sociaux, en profitant notamment du recul de la place du travail contraint dans la vie? La possibilité d'infléchir les tendances porte sur l'organisation de la société, celle qu'on baptise parfois "civile" pour la distinguer de l'état. Mais l'état devrait retrouver un rôle pour impulser la transformation : moins un rôle de décision, que d'écoute; de formulation, d'entraînement. Et le point d'appui, certes fragile, pour une telle transformation est le souci du monde et l'aspiration à la coopération dont nous avons parlé plus haut. L'école peut là entrer en jeu.

Quelques pistes pour l'école

Nous avons dit que l'école était affrontés à trois défis. Le plus rude est la rareté du travail, qui devrait se poursuivre dans tous les scénarios, et il sera demandé à l'école d'aider les jeunes à s'y adapter. Cela ne devrait pas conduire à renoncer à toute formation professionnelle car le métier fait partie d'une identité personnelle aujourd'hui malmenée par l'incertitude, mais à en faire davantage un atout parmi d'autres, pour évoluer à partir de là, en fonction de possibilités locales

au développement desquelles devraient s'associer les établissements scolaires et universitaires. Un autre problème est celui de l'utilisation du temps libéré et de l'acquisition par là d'une position sociale: il s'agit d'apprendre à créer des activités, avec d'autres. Et d'ailleurs la création, avec insertion dans un tissu local, est peut-être le nouveau ressort que pourrait trouver l'école pour donner du sens au travail scolaire, à la place du plaisir de la connaissance ou de la recherche du bon métier, qui ont perdu de leur réalisme. Tout cela rejoint une préparation au "vivre ensemble", avec une insistance, une fois de plus, sur le niveau local qui, en dépit du développement des communications, reste bien le premier lieu d'insertion pour la grande majorité des jeunes. De même, le changement de rôle de l'état, sinon son recul - second défi - donne de l'importance à un ancrage local de l'école.

Cela ne signifie pas que l'école n'ait pas à dépasser ce niveau local, pour s'intégrer dans des ensembles plus vastes et préparer les jeunes à le faire. C'est vrai à cause de la mondialisation des problèmes. Ce l'est plus encore pour le troisième défi, le plus important pour aller vers le troisième scénario: lutter contre les inégalités et éviter le morcellement de la société. Une coordination des actions à un niveau qui dépasse le local est ici nécessaire. On retrouve là un rôle pour les pouvoirs publics, notamment l'état, plus animateur et régulateur qu'organisateur et directeur.

Et l'informatique ?

La place que peut jouer l'informatique dans l'avènement d'un scénario favorable n'est pas très claire : comment en faire une science "humaine"? Nous retrouvons la vieille question de l'usage qui est fait de la technique et de la science, et celle de la responsabilité des scientifiques. Peut-on dire que cette responsabilité, celle des universitaires, est de mettre la science à la disposition de tous, et pas seulement des plus puissants ?

Pour l'informatique à l'école, il me semble qu'elle devrait trouver assez facilement une place dans les pistes tracées ci-dessus: aider à développer les compétences demandées aujourd'hui par l'emploi et la nécessité de s'adapter, qui sont de nature transversale plus que technique; contribuer à une création qui donne du sens au travail scolaire; participer à l'insertion des établissements scolaires dans le tissu local; mais aussi permettre de dépasser le niveau local et de s'ouvrir sur le monde.

Cependant la relation entre formation et changement est complexe. La formation est un processus qui provoque un changement des personnes et de leurs représentations du monde. Le changement des représentations individuelles provoque un changement de la société. Mais la formation n'est pas seulement un processus individuel (on ne se forme pas seul), c'est aussi un processus social. Et le changement agit donc sur elle en retour. Bref, lorsqu'on tente d'étudier cette relation entre formation et changement, on est en face d'un phénomène récursif. Et nous savons bien que la récursivité peut conduire à des contradictions. C'est la faute au changement : il agit sur un état de la société, mais la représentation de cet état,

et aussi celle du changement, font partie de l'état même. Alors, quand la situation change, le changement change, et la représentation du changement aussi ; de sorte que le changement prévu ou souhaité, sur lequel a pu se fonder la formation, n'est finalement pas celui qui se produit.

C'est la faute aussi à la nature de l'Homme qui est un être conscient, donc autoréférent : d'où le caractère central qu'occupent dans le processus de changement ses représentations, ses modèles du monde. Qui est aussi un être social, et les modèles sont des objets sociaux, des systèmes de référence. Nous avons déjà dit que l'informatique n'est pas sans influence sur ces modèles.

Or, la manière de sortir des contradictions que peut créer la récursivité, c'est sans doute de passer à un autre niveau, et en l'occurrence d'introduire une éthique, éthique pour l'école, éthique à développer chez les jeunes. Ce que nous avons vu, c'est que sa base devrait être le refus des inégalités. Alors finalement, la question qu'appelle le titre de cette conférence, c'est: l'informatique a-t-elle quelque chose à voir avec l'éthique ?¹⁰

Claude PAIR

6, avenue Saint-Sébastien

54600 Villers Les Nancy

Tél. : 83 90 29 42

¹⁰ Certaines parties de cette conférence sont extraites d'un article de la revue *Administration et Education*, 1996, no 1: "déchiffrer le présent pour défricher l'avenir".

Conférence 3

(R)Evolution et Tendances

Raymond Morel

présentée le jeudi 13 avril 1996 à 09h00

Introduction

A partir d'un ensemble d'observations liées à l'évolution de l'informatique et rassemblées ces derniers mois en dehors du monde éducatif, il est prévu, lors de la présentation de les placer tel un écho face aux mutations lentes et aux problèmes constants (non résolus) des NTIC (Nouvelles technologies de l'information et de la communication) dans les systèmes de formation.

1. La notion de document change et s'étoffe

Sans partir dans la direction de Joël de Rosnay qui, dans son dernier livre ("L'homme symbiotique, Regards sur le troisième millénaire"), propose le terme d'unimédia (cf. p. 83) plutôt que celui de multimédia pour mettre en évidence la **convergence de quatre domaines** (l'écrit, l'informatique, les télécommunications et l'audiovisuel) vers les réseaux numériques, chacun ressent confusément que le terme "document" ne recouvre plus uniquement le livre, la revue ou une brochure. Les utilisateurs de l'informatique parlent fréquemment de leurs documents pour leurs fichiers, les infographistes ou les musiciens également pour leurs images, leurs animations ou leurs créations sonores. Une tendance qui se dégage de plus en plus est qu'un **document** au sens large est une production qui possède plusieurs des **cinq attributs** suivants :

- le **texte**
- l'**image**
- l'**animation**
- le **son**
- le **script de diffusion** et son support (cf. fig. 1)

Les quatre premiers attributs sont les caractéristiques fondamentales du **multimédia** où l'on manie les composantes **statiques** (textes et images) comme par exemple dans la PAO (publication assistée par ordinateur) et/ou les composantes **dynamiques** (sons et animations faisant intervenir le temps).

Le cinquième attribut (le script de diffusion) est intrinsèquement lié au **mode de distribution** du produit et se fait de plus en plus électroniquement et avec notamment la **télématique**.

Essayez de penser à quelques documents (journal, livre, message, séquence vidéo, album de photos, etc.) sous leur forme traditionnelle et à la version électronique. Il est intéressant de constater que l'environnement du World Wide

Web, dont il est abondamment question partout depuis des mois avec Internet, est justement basé sur cette nouvelle problématique avec une intégration complète des cinq attributs.

2. Performance n'est pas synonyme d'intégration

En tirant le bilan de l'évolution en 10 ans (1984-1994) de l'augmentation des performances des principales caractéristiques d'une des deux marques de machines utilisées dans notre enseignement, un animateur d'un des derniers séminaires organisés le printemps dernier remarquait, sous forme de boutade, qu'en une décennie la puissance avait augmenté de plus de 30 millions de fois !

Comment est-ce possible ?

- les processeurs sont 20 fois plus rapides (par ex. 68000 → Power PC)
- les mémoires vives sont 20 fois plus grandes (1 Méga → 20 Méga)
- les mémoires mortes sont 20 fois plus importantes (64K → 4 Méga)
- les espaces disque sont 200 fois plus étendus (2 x 800K → 500 Méga)
- les informations sur l'écran sont 20 fois plus nombreuses (noir / blanc → couleurs)

et $20 * 20 * 20 * 200 * 20 = 32 * 106^3$ Trente millions. Même si le fait de multiplier les différents facteurs entre eux est des plus douteux, cette plaisanterie met bien en évidence **une certaine évolution**.

Quand au raisonnement que l'on serait tenté de faire **par rapport aux logiciels** pendant ces 10 dernières années, il est visible qu'ils ont évolué (pensez aux traitements de texte, aux tableurs, aux bases de données, aux logiciels graphiques, etc.), mais certainement pas dans les mêmes proportions (à propos, quelles pourraient être les unités de comparaison ?).

Par rapport à la **qualité et à l'ergonomie des logiciels**, là les avis sont partagés, les uns voyant une sensible amélioration, les autres ayant déjà repéré que la situation a déjà passé par un optimum et pointant le doigt accusateur en direction de la course effrénée pour les nouvelles versions de logiciels (cf. également le paragraphe 4 ci-dessous).

En pensant aux utilisateurs et aux usages, on peut remarquer pendant ces 10 dernières années une **lente progression de l'intégration des NTIC dans l'éducation**, mais il faut du temps pour se former, absorber, s'approprier, continuer à se former, digérer, avant de pouvoir utiliser ces outils sereinement jusqu'à leur banalisation, stade où on ne parlera plus d'intégration.

3. De l'importance de la veille technologique

La mise en œuvre de projets suite à des **décisions stratégiques** obéit à des objectifs visant le **long terme** (souvent plus de 10 ans).

Les **moyens** de réalisation ont recours à des **technologies** qui se situent dans le **moyen terme** (5 à 10 ans) et obéissent à des **normes**.

Les **outils** qui rendent finalement possible ces applications sont des **techniques et/ou produits** dont la durée de vie est dans le **court terme** (à peine 1 an). Ils font l'objet de **standardisation**.

La course systématique à la dernière version d'un logiciel est une démarche malheureusement trop fréquente, peu rentable, car très coûteuse (... en ressources humaines surtout), pour des améliorations aléatoires avec des perturbations assurées qui se terminent souvent en cul-de-sac (cf. fig. 2). Il faut donc **se resituer au niveau des technologies** et de leur évolution et non pas au niveau des produits pour atteindre les objectifs fixés.

Par ailleurs, il serait bon de distinguer entre caractéristiques d'une norme, d'un standard du marché, d'un standard de force, d'un système ouvert/ fermé, etc., et ce tant pour le matériel et le logiciel que pour les réseaux ou les outils de développement.

La nécessité de préciser **des normes et des standards** est incontournable; mais encore faut-il qu'il existe une **veille technologique** solide, réelle et complète pour procéder à une mise à jour continue. Il semble à l'heure actuelle que cette problématique n'ait pas échappé aux responsables d'entreprises et d'institutions puisque quelques **Observatoires Technologiques** sont, soit en gestation, soit mis en oeuvre depuis quelques mois.

4. L'interface utilisateur évolue lentement, mais...

Au niveau des interfaces et de la navigation pour les utilisateurs des progrès sensibles ont été réalisés depuis des années (passage du terminal alphanumérique à la ligne de commande, puis au mode graphique). Mais des progrès importants sont encore à réaliser rapidement (par ex. l'aide en ligne n'est souvent qu'une description plus ou moins compréhensible d'une fonctionnalité alors que l'utilisateur dans une telle situation souhaiterait qu'on lui "prenne la main" afin de l'aider réellement en lui montrant le "comment"). Le schéma ci-dessous (cf. fig.3) ébauche ces axes d'évolution.

Le potentiel d'amélioration est encore à découvrir pour l'interface utilisateur (reconnaissance de l'écriture, de la voix, etc.).

La marge de progression est énorme avant qu'on puisse imaginer l'adéquation de slogans comme "Anytime/Anywhere" ou "Useful for Most Things" ou "Easy to Access", même si on devine que l'on se dirige inexorablement vers un "interface unique pour toutes formes de communication" selon le vœu de Jim Clark, PDG de Netscape.

A ce propos, ne pas oublier de lire le dernier livre de Yves Lasfargue dont le titre est "ROBOTISES, REBELLES, REJETES - Maîtriser les nouvelles technologies".

5. De l'explosion des services... à la convergence et la cohérence des usages

Le schéma ci-dessous (cf. fig. 4), tiré également de l'exposé de Joe Nervin, s'efforce de mettre en évidence que les nouvelles technologies **ne se réduisent pas** à des fonctions de calcul ou de bureautique traditionnelles (traitements de textes, tableurs, bases de données, etc.). En effet, **d'autres technologies et services émergent** (cf. fig.4) et sont tout aussi importants:

- le courrier électronique (**e-mail**)
- le **groupware** (travail collaboratif/ coopératif)
- le traitement de la voix (**voice**) en général et des applications liées au téléphone (fax, serveurs vocaux, etc.)
- **Internet** et les autoroutes de l'information avec une série de services
- le **multimédia** en général (vidéoconférence, etc.: le tout numérique, cf. le paragraphe 1 ci-dessus).

Comme le suggère le schéma ci-dessous (cf. fig.4), on est en face d'une **véritable explosion des services**, mais dans une **vision globale qui converge** et doit garder une **cohérence** pour les utilisateurs. Une pièce de puzzle ne devrait pas imposer un produit, au sens défini plus haut (cf. paragraphe 3 ci-dessus), qui fixe des contraintes péjorant les concepts de base de tout système d'information qui est au service des utilisateurs en premier lieu.

6. Une dérive qui traduit une (des) évolution(s)

Avec le temps qui s'écoule, il est possible de repérer une **mutation assez rapide** des conceptions et des points forts liés à l'informatique. Jusqu'à la fin des années 70 et même au-delà, ce qui était le moteur de l'action, et donc vital, était le **matériel** (hardware) avec les **gros ordinateurs**, puis les **minis** (mainframe, minicomputers).

Depuis 10 ans, on se rend bien compte que le dynamisme repose sur le **logiciel** (software) et les **micro-ordinateurs** (PC: personal computer) et bientôt les machines à communiquer (NC: networking computer et/ou PDA: personal digital assistant).

On parlait de systèmes **fermés**, de systèmes **propriétaires**, d'unité **centrale**, alors que maintenant on dit systèmes **ouverts**, systèmes **multiplateformes**, **réseaux**.

Sur une période plus restreinte (ces 10 dernières années) on est passé de l'informatique **personnelle** à l'informatique de **groupe**.

On remarque aussi que le **constructeur** qui avait un **statut d'intégrateur et d'assembleur** depuis R&D (Recherche et Développement) jusqu'à la production, tant pour le matériel que pour le logiciel, est condamné à **se spécialiser** sur un **créneau** et ne devient plus qu'un **partenaire** parmi d'autres par rapport à un **équipement** qui est toujours plus modulaire et multifonction.

Ce rôle d'intégration est repris par l'arrivée de nombreux nouveaux services et acteurs pour encadrer et aider les utilisateurs (développements d'applications spécifiques, formation, etc.).

7. Miser sur l'avantage concurrentiel

Par rapport aux différents créneaux cités ci-dessus, au paragraphe 5, on constate ces derniers mois une montée en puissance de certaines technologies (**groupware, Internet et multimédia**), où l'on passe des concepts à une série de produits en quittant le stade des expérimentations pour arriver sur le terrain de la bagarre industrielle et commerciale. On débouche alors sur la notion **d'avantage concurrentiel**...

"Face à la concurrence mondiale, aux exigences croissantes des clients, les managers américains misent avec succès sur les nouvelles technologies de l'information et de la communication tel que le groupware, la messagerie, Internet, le multimédia, ... Ce sont des opportunités à saisir rapidement pour bénéficier des marchés émergents et s'assurer des avantages concurrentiels décisifs.

Les technologies de l'information se développent à une vitesse remarquable. Ce qui était totalement impossible hier devient possible aujourd'hui et sera demain complètement banalisé. Internet, le groupware, et le multimédia d'entreprise connaissent une expansion explosive. Le nombre et la variété des innovations sont tout à fait étonnants. Et tout ce que l'on en sait, montre que leur rythme de croissance n'est pas près de se ralentir dans les années à venir.

Face à ces évolutions, le décideur est de plus en plus perplexe. Faut-il investir dans ces nouvelles technologies ou est-il préférable d'attendre davantage? Une fois la décision prise, comment y aller dans de bonnes conditions et réussir la mise en œuvre? Quels vont être les résultats de tous les investissements effectués?

Pour tout décideur, il est aujourd'hui fondamental d'obtenir des réponses précises et claires à ces questions. Pour cela, il est nécessaire de dégager des expériences un certain nombre de clés permettant de mieux cerner l'apport présent et futur de ces nouvelles technologies aux objectifs stratégiques des entreprises.

L'intérêt porté aux notions de groupware, d'Internet et au multimédia d'entreprise est fondamentalement lié à la contribution de ces différentes technologies, à l'amélioration de la profitabilité des sociétés et à l'élargissement de leurs marchés. L'expérience des entreprises les plus novatrices montre qu'il existe de très nombreuses opportunités dans ces domaines. Faut-il encore savoir les saisir à temps! Il est évident que les entreprises qui négligeraient ces opportunités se mettraient en difficulté et on peut craindre que certaines ne disparaissent."...

“Pour être efficace dans le domaine des nouvelles technologies, il est impératif d’anticiper pour mieux comprendre et prendre à temps les bonnes décisions.”...

*“La règle du jeu est simple. Elle s’appelle **avantage concurrentiel**, gains de parts de marché, profitabilité, amélioration de la valeur ajoutée, “best practices”, ... Grâce à la maîtrise des nouvelles technologies, votre entreprise s’assurera des avantages concurrentiels considérables.*

Tel est votre enjeu pour demain!”

(extraits de la présentation de la conférence BUSINESS’96, janvier 96)

8. Ce n’est qu’un début...!

Daniel Borel, PDG de Logitech (la célèbre firme qui a mis à la mode la souris pour interagir avec un équipement), signalait au début janvier dans la revue Bilan que “pour la première fois de l’histoire il **s’est vendu dans le monde**, cette année (1995), **plus de PC que de postes de télévision**”. Si les ventes de TV représentaient surtout des renouvellements de matériel sur un marché saturé, les 57,7 millions de PC vendus étaient, dans leur grande majorité, des investissements dans un contexte en pleine expansion.

Ce croisement inexorable et prévisible de ces deux courbes est irréversible, car les ventes annuelles de PC vont plutôt atteindre 100 millions de pièces par an d’ici la fin du siècle. La gestion et les méthodes mises en œuvre pour “digérer” cette avalanche de technologies sont parfois un peu primaires et des dérapages peu souhaitables ne sont pas exclus sur plusieurs plans (cf. l’article de A.-Y. Portnoff “Mourir compétitifs ou innover ensemble”). Autant dire qu’en reprenant ensemble les différentes observations décrites dans cet article, nous avons derrière nous quelques évolutions lentes et/ou spectaculaires, mais que si l’on veut parler de **révolution**, elle est plutôt à venir et ce surtout au niveau des innovations à trouver dans les systèmes organisationnels sans oublier l’impact et les différents aspects de l’activité de la société.

9. Vers la généralisation ... et la banalisation

Dès 1979 certains experts affirmaient qu’il faudrait une quinzaine d’années avant d’avoir les conditions nécessaires à une généralisation des expériences pilotes d’introduction des NTIC dans l’éducation. Plus colorée fut la remarque d’Allan Kay en 1990 lors de la conférence mondiale de l’IFIP sur éducation et informatique devant plusieurs milliers de personnes concernées : “Si dans une école de 700 élèves vous n’avez que 10 stylos rassemblés dans une seule salle, vous n’oserez quand même pas affirmer que tous vos élèves écrivent en même temps plusieurs fois par jour” (sous-entendu l’intégration des NTIC dans l’éducation ne sera possible qu’à condition d’une généralisation de l’outil pour pouvoir le banaliser comme un crayon ou un stylo – “On ne va pas dans la salle des stylos si on a besoin d’écrire !” ajoutait-il).

Dans le contexte économique de ces dernières années, la solution au problème posé n'était pas dans des achats massifs supplémentaires, alors qu'on se trouve dans tous les pays avec des difficultés à renouveler les équipements de départ. Une des possibilités envisagées a toujours été d'équiper tous les élèves d'une "grosse" calculatrice de "poche". C'est pourquoi les participants au dernier séminaire sur Cabri-Géomètre à Genève ont été particulièrement intrigués de pouvoir "toucher" et "jouer" avec une petite merveille pour un enseignement de sciences : la TI-92 avec entre autres CABRI II, DERIVE, un tableur, des représentations graphiques à 2 et 3 dimensions, des analyses statistiques, du calcul matriciel, etc. Munie d'un clavier QWERTY et à peine plus grosse qu'une calculatrice de poche traditionnelle, cet outil est disponible pour moins de SFr.300.- (cf. fig. 5 et 6)

Cette innovation inaugure une des pistes réelles qui mérite d'être suivie de près quand on pense généralisation.

10. ... et dans les systèmes de formation ?

En échos à ces 9 observations, il semble judicieux de placer maintenant sous la loupe quelques domaines liés aux NTIC dans l'éducation et voir si quelques tendances ou évolutions se dessinent, notamment pour :

- mise à jour des plans d'études pour les élèves (cf. références a,b,c,d,i,n)
- curriculum pour la formation des enseignants (cf. références a,c,d,m,n)
- centre de ressources - encadrement (cf. références c,d,e,m)
- place des NTIC dans l'enseignement (discipline à part entière vs intégrées dans les autres disciplines) (cf. références a,b,c,d,g,h,i,j,k,l,m,n,p)
- rôle de l'enseignant (NTIC pour enseigner ou pour apprendre ?) (cf. références c,d,g,i,m,n)
- intégration vs banalisation (cf. références d,e,g,i,j,k,l,m,n,o,p,q)
- travail collaboratif / coopératif - communiquer et apprendre à réfléchir ensemble (cf. références c,d,h,i,k,l,m,n,o,p,q)
- le marché des logiciels éducatifs multimédia (cf. références c,d,e,h,n,q)
- processus d'innovation (cf. références c,d,f,g,h,i,m,n,p,q,r)
- "alphabétisation" et prise de conscience des décideurs (cf. références b,c,d,e,i,r,s)
- apprentissage de base - Invariants - Mutations des mises en oeuvre.(cf. références c,d,i,m,n,p,q)

Raymond Morel

Directeur du Centre informatique pédagogique (CIP)

Département de l'Instruction publique

2-4, rue Théodore-de-Bèze Case postale 3144 CH 1211

Genève 3 (Suisse)

Tél. + 41.22.318.05.30 / Fax + 41.22.318.05.35 / e-mail : morel@uni2a.unige.ch

Annexes

12. Annexes

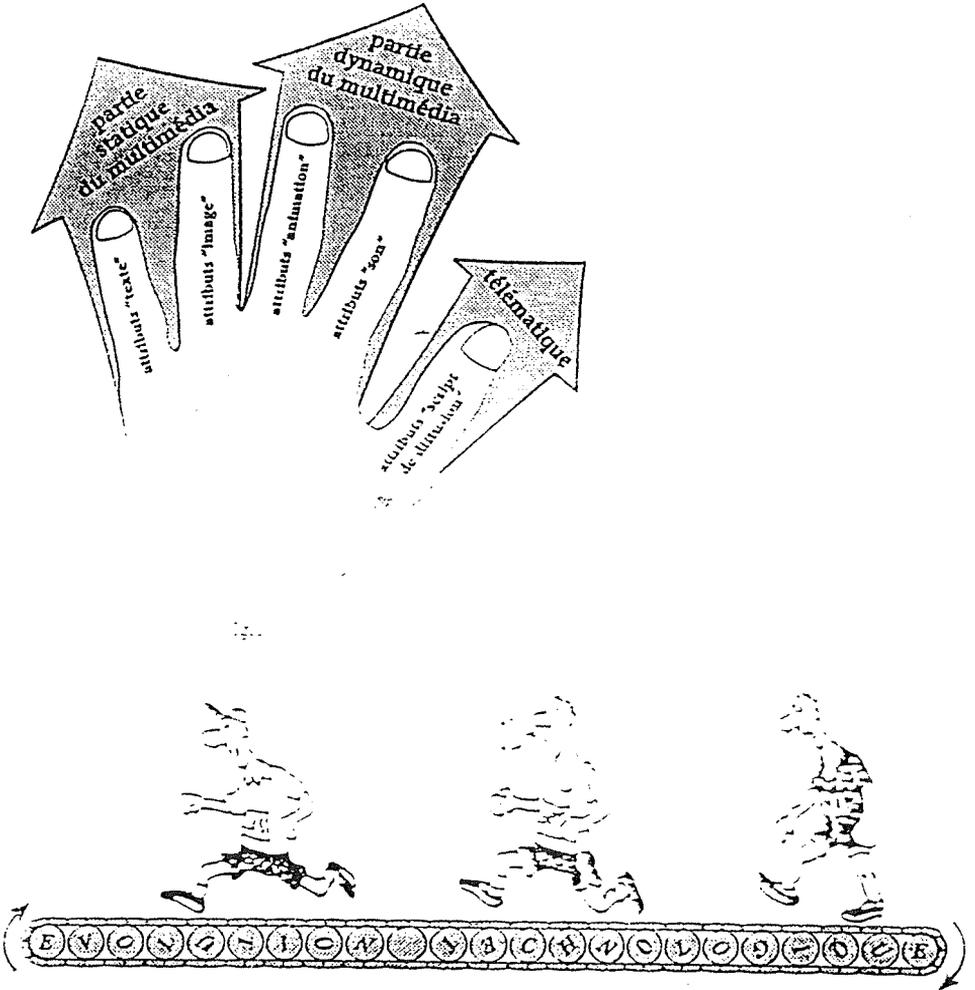
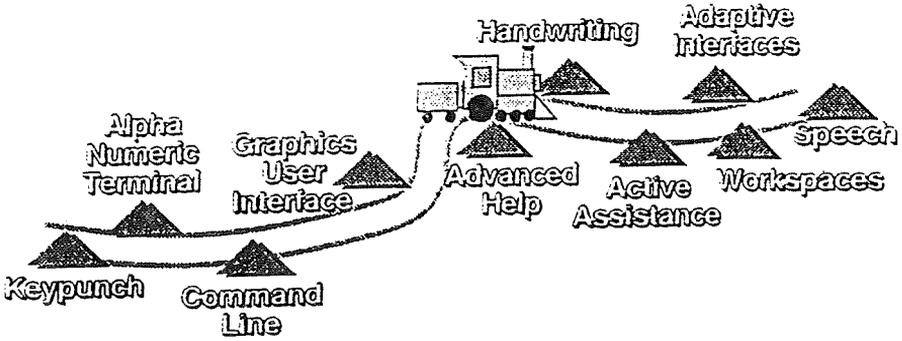


fig.2

Natural Interface

Human Interface Strategy



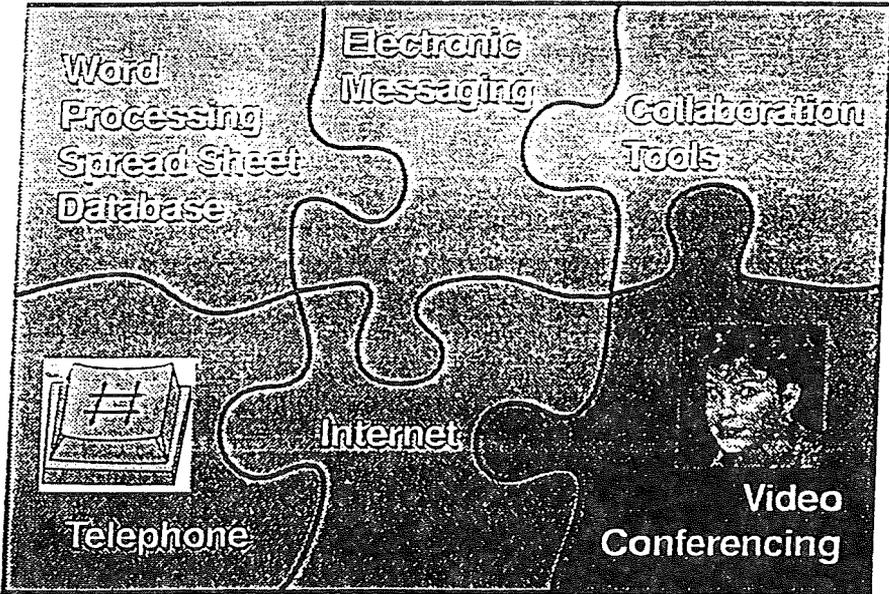
Making Complexity Simple

fig.3

Schéma tiré de l'exposé de Joe Nervi à Genève, octobre 95

Personal Communications

Moving Beyond Computation



cf. fig.4

Schéma tiré de l'exposé de Joe Nervi à Genève, octobre 95

101146.1

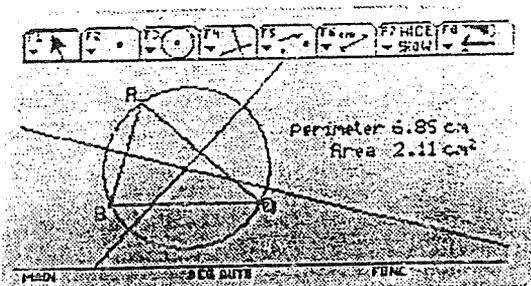


fig.5

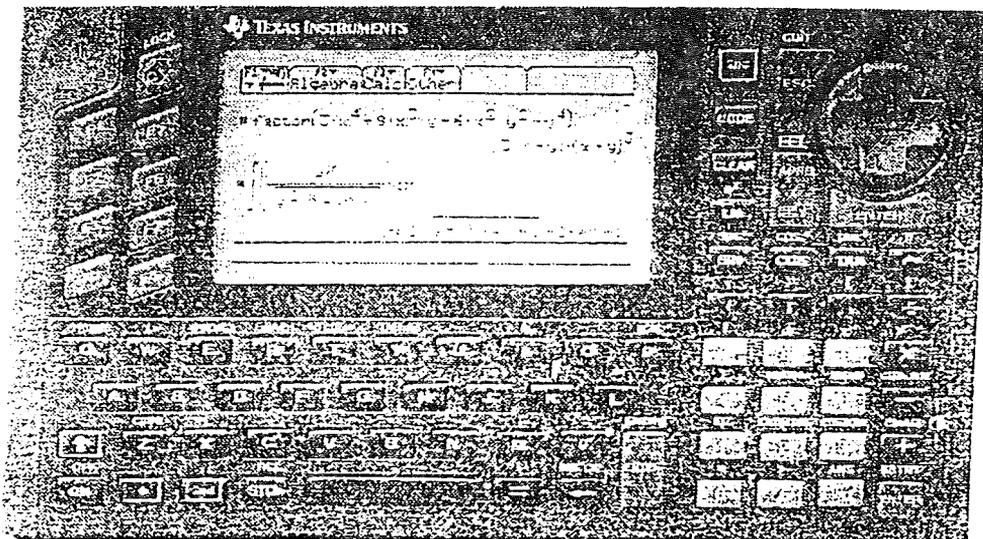


fig.6

11. Références

- [a] Informatics for Secondary Education - A Curriculum for Schools, UNESCO, (1994)
- [b] Superhighways for Education - The Way Forward (The UK policy), HMSO publication center,(1995).
- [c] Mémoires soumis à la Commission des Etats généraux sur l'éducation et allocation du Directeur Général de la CECM - AQUOPS, Québec, (août 1995).
- [d] World Conference on Computers in Education VI (IFIP WCCE'95) - Liberating the Learner, Chapman & Hall (1995).
- [e] Logiciels éducatifs et multimédia (rapport intermédiaire) - Task Force de la Commission Européenne, (1996).
- [f] Robotisés, Rebelles, Rejetés - maîtriser les nouvelles technologies, Yves Lasfargue, (1993) No ISBN : 2-7082-3048-4.
- [g] L'intégration de l'EAO à l'école, Commission EAO du DIP à Genève, (1991).
- [h] Projet pilote du G7 "Information and Society", Thème 3 : Transcultural education and training - Feasibility study "GETALL" (Global communication platform for european transcultural and language learning and the european language industry).
- [i] Systèmes de formation et nouvelles technologies - Commission EAO du DIP à Genève, (1995).
- [j] Opération "Success Stories" version 1.0 - monographie no 3 du CIP (1994), version 2.0 - pour la conférence IFIP WCCE'95,(1995).
- [k] Télématicque pédagogique - monographie no 2 du CIP,(1994).
- [l] Opération 4ème de couverture du CIP - encouragement à la lecture, (éditions 1994, 1995, 1996).
- [m] Programme SOCRATES de la Commission Européenne : projet Fétiche (Formation des Enseignants aux nouvelles Technologies de l'Information et de la communication : Changements et évolutions), (1996).
- [n] Travaux de l'AFDI depuis 1988.
- [o] Versions successives de la didacthèque du CIP à Genève (1987 - 1996).
- [p] Conférence internationale à Genève sur le thème "Accès à la formation à distance : clés pour un développement durable", (1994).
- [q] Revue Informatique-Information du DIP à Genève (Numéros 1 à 29)
- [r] Article d'André-Yves Portnoff intitulé "Mourir compétitif ou innover ensemble" revue Futuribles,(1995).
- [s] Article de Nicole S. Morgan intitulé "L'utopie de Thomas More revisitée", revue Futuribles (1996).

LES COMMUNICATIONS

Thème2 : Informatique au Secondaire

Approche de l'informatique au niveau préscolaire

Francis LOWENTHAL

présentée le jeudi 11 avril 1996 à 11h15

Résumé: Nous présentons ici une manière nouvelle d'utiliser un jouet classique pour introduire des éléments de langage informatique de type Logo auprès de très jeunes enfants, ou auprès d'enfants handicapés qui ne savent ni lire ni écrire. Après avoir précisé le cadre théorique, nous décrivons le matériel utilisé et la manière de l'employer (en proposant entre autre une hiérarchie d'exercices). Nous décrivons le rôle du sujet qui utilise ce matériel simple de la manière proposée, nous envisageons des transferts vers divers types d'apprentissage: lecture, mathématique, raisonnements, etc. Dans une dernière partie, nous comparons le langage Logo et la méthode que nous décrivons dans le présent article, à la lumière de recherches concernant les stratégies cognitives.

A. Introduction

Le langage Logo est le résultat des travaux d'une équipe de chercheurs comprenant, entre autres, des informaticiens soucieux d'adapter le LISP (le premier langage créé pour l'Intelligence Artificielle) aux besoins et aux possibilités de jeunes enfants mais aussi des spécialistes formés à l'approche piagétienne en psychologie et en éducation. Parmi ces spécialistes, le plus important semble être S. Papert qui est à la fois un spécialiste de l'Intelligence Artificielle et un chercheur qui s'est formé auprès de Piaget. Dans son livre "Mindstorm. Children, computers and powerful ideas" (1980) Papert présente de nombreuses possibilités offertes par le langage Logo. Depuis lors de nombreux auteurs, tels Gurtner et Retschitzki (1991), ont tenté de montrer que confronter l'apprenant aux problèmes de programmation dans le cadre d'un langage souple et extensible favorise leur développement cognitif. Ces auteurs ont pourtant négligé un facteur important: seul un enfant qui sait lire et écrire est capable d'utiliser de manière autonome le langage Logo, ceci exclut donc les jeunes enfants de 5 ou 6 ans qui sont capables de raisonner correctement, mais pas de communiquer leur raisonnement à une machine. Papert, dans l'introduction de "Mindstorms", avait pourtant présenté un autre dispositif qui avait favorisé son apprentissage de l'algèbre: un jeu d'engrenages qu'il avait reçu très jeune et qui pouvait être utilisé par un enfant sans qu'une consigne verbale (écrite ou orale) ne soit nécessaire. Depuis 1970, nous avons choisi d'utiliser ce type d'outils que nous appelons "représentations concrètes de systèmes formels".

B. Les représentations concrètes de systèmes formels

Une représentation concrète d'un système formel n'est rien d'autre qu'un ensemble d'objets munis de contraintes techniques. Ces contraintes techniques rendent certaines actions possibles et d'autres impossibles, sans qu'il soit nécessaire de parler au sujet. Tout ceci permet donc d'installer une structure logique, soit pour

observer le sujet dans un cadre précis, soit pour favoriser le développement cognitif de ce sujet (Lowenthal, 1978).

Les briques LEGO constituent un excellent exemple d'un tel dispositif: dès que la première brique est placée sur la plaque de base, la suivante ne peut plus être placée (dans le cas d'un chemin à construire) que dans le prolongement de la précédente ou perpendiculairement à celle-ci; il est techniquement impossible de placer des briques simples en formant un angle autre que 90° ou 180° . Ceci traduit de manière très concrète les éléments de base de ce que les mathématiciens appellent la "géométrie de la distance absolue", distance qui est fort utilisée dans un domaine de pointe en mathématique: L'analyse.

Il est facile de montrer (Lowenthal, 1986a) qu'une représentation concrète d'un système formel offre de nombreux avantages: les axiomes et les lois d'inférences du système sont imposés par les contraintes techniques du matériel. Un tel dispositif peut être utilisé de manière quasi non-verbale et les objets qui le constituent n'ont pas de signification autre que celle imposée par les contraintes techniques-axiomes: le dispositif est donc non ambigu. Les petits éléments d'un tel matériel sont simples et faciles à manipuler par de jeunes sujets, ils permettent généralement de construire plusieurs solutions différentes (ou d'arriver de plusieurs manières différentes à la même solution) pour un même problème. Toutes ces manipulations, accessibles aux très jeunes enfants, peuvent avoir lieu dans le cadre de jeux; ces manipulations permettent en outre à l'observateur d'étudier, pas à pas, les processus de réponse utilisés puisque les éléments qui constituent cette représentation concrète doivent être placés l'un après l'autre.

Le recours aux manipulations de ces représentations concrètes de systèmes formels, pour proposer aux apprenants des problèmes qu'ils ne parviennent pas à résoudre lorsque ceux-ci leur sont présentés verbalement, permet de mieux observer chaque enfant: on voit ce que le sujet fait; ces manipulations concrètes permettent de lui présenter un problème logique, d'observer comment il construit pas à pas une solution et d'obtenir ainsi une représentation de ce qui se passe dans son esprit. Ceci permet ensuite de préparer des séquences d'exercices adaptés aux difficultés de chaque apprenant pour préparer ceux-ci à formuler des hypothèses, à les tester et à les valider en les confrontant à la réalité.

De telles représentations concrètes de systèmes formels peuvent ainsi être utilisées pour observer ou pour favoriser le développement cognitif de jeunes sujets, normaux handicapés (Lowenthal, 1986b, 1987b, 1992; Ledru & Lowenthal, 1986).

C. La mosaïque: une représentation concrète d'un système formel

Parmi les représentations concrètes que notre équipe a utilisées, celle appelée "mosaïque" permet d'aborder très tôt des problèmes de programmation. Le matériel se compose d'une plaque blanche perforée dans laquelle on peut enfoncer des clous en plastique. Ces clous sont définis à l'aide de deux variables: la couleur et la forme. Il y a des clous de cinq couleurs différentes: bleu, rouge, orange, vert et

jaune; il y a aussi deux formes possibles: la tête d'un clou peut être un carré ou un quart de cercle. Dans le texte qui suit, les clous à tête carrée sont appelés "carrés" et ceux à tête en quart de cercle sont appelés "pointes", les perforations de la plaque sont appelées "trous". Pour des raisons pratiques évidentes, les couleurs des pièces utilisées sont représentées par des nombres dans les figures concernant ce matériel: bleu est codé par 1, rouge par 2, vert par 3, orange par 4 et jaune par 5.

Ce matériel a en fait été inventé par des concepteurs de jouets dans un but très différent de celui que nous poursuivons. Leur idée de départ est la suivante: fournir un matériel qui permette à l'enfant de réaliser, dans un cadre uniquement ludique, un beau dessin, très coloré, à l'aide de clous de formes et couleurs variées. La structure du matériel permet néanmoins de faire mieux: il est possible de placer des clous l'un à côté de l'autre "joliment et sans espace perdu" à condition de respecter certaines contraintes techniques qui obligent le sujet à n'utiliser que certains "trous" et à en négliger d'autres. Il est possible de communiquer avec le sujet de manière quasi non verbale en utilisant les clous: il est possible, grâce à ce matériel, de faire réaliser par le sujet des programmes et d'autres exercices de type informatique, sans ordinateur.

Nos premiers travaux avec ce matériel concernent uniquement des exercices destinés à installer chez certains enfants handicapés des notions de rythme, de correspondance biunivoque, de reconnaissance de forme et de couleur ainsi que certaines notions d'orientation dans l'espace de la plaque perforée. Ces activités sont décrites dans un rapport concernant les activités logiques entreprises pour remédier aux troubles de la communication chez un enfant aphasique (Lowenthal & Saerens, 1986).

Une approche nouvelle, inspirée du langage Logo, a été imaginée par notre équipe (Saerens, 1984; Lowenthal, 1984, 1985): dans ce cadre un "carré" représente une action de base ou primitive tandis qu'une "pointe" représente un nom pour une séquence d'actions de base: une "pointe" représente donc le nom d'une procédure, sauf la "pointe" de couleur jaune qui est exclusivement utilisée pour indiquer une direction. Une fois cette convention établie, il devient facile de proposer plusieurs types d'exercices préparant à la programmation. Nous avons travaillé avec de nombreux enfants pour mettre au point une gradation d'exercices, divisée en 5 grandes parties que nous présentons ci-dessous.

Il faut souligner ici que la résolution de chacun des types d'exercices décrits ci-dessous implique la prise en considération simultanée de deux informations distinctes pour produire une réponse. On peut faire le parallèle entre ce type d'activités et des exercices en Logo où l'apprenant doit tenir compte d'une part des définitions des procédures utilisées et d'autre part de l'ordre dans lequel il faut les utiliser. Piaget a montré que de jeunes enfants éprouvent des difficultés lorsqu'ils doivent ainsi tenir compte simultanément de deux types d'informations (un gobelet large et bas contient-il autant de liquide qu'un gobelet haut et étroit ?). Nos observations prouvent que ce n'est pas le cas dans le cadre de la mosaïque.

Pour la clarté de l'exposé nous distinguons trois zones sur la plaque: près du bord gauche se trouve la zone des procédures, au centre, la zone du programme et près du bord droit, la zone d'effectuation.

1. L'exécution du programme.

L'enfant doit construire dans la zone d'effectuation la suite de carrés correspondant aux procédures et à la suite de pointes placées par l'enseignant dans les zones prévues à cet effet.

Exemple

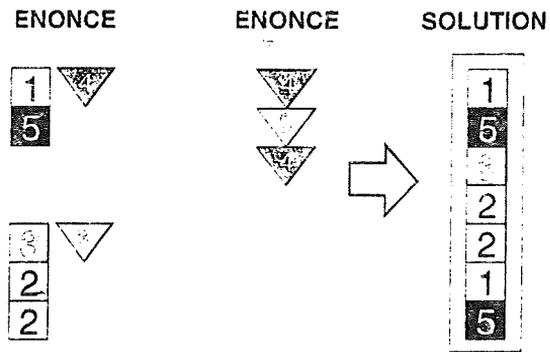


Figure 1.

L'enseignant place le long du bord gauche de la plaque la séquence de carrés suivante: un bleu suivi d'un jaune, et il associe à cette séquence la pointe orange: cette pointe devient ainsi un nom de procédure; il place ensuite en dessous la séquence de carrés suivante: un vert suivi de deux rouges, et il associe à cette séquence la pointe verte¹. L'enseignant place ensuite, verticalement, au milieu de la plaque, la séquence de pointes suivante: une orange, une verte puis une orange. Il montre alors la séquence de pointes et dit à l'enfant "Fais la même chose rien que avec des carrés". L'enfant peut alors construire la séquence codée par ce petit

¹ Dans cet énoncé-ci les couleurs sont choisies de manière tout à fait arbitraire, seule la pointe jaune a une signification particulière (elle est réservée aux changements de directions) et elle doit donc être évitée dans les 4 premiers types d'exercices.

programme et créer le long du bord droit de la plaque la séquence de carrés suivante: bleu, jaune, vert, rouge, bleu, jaune.

2. La recherche du programme.

L'enfant doit ici construire dans la zone du programme la suite de pointes qui permet de réaliser la séquence de carrés (placée dans la zone d'effectuation) à partir des procédures proposées (dans la zone des procédures).

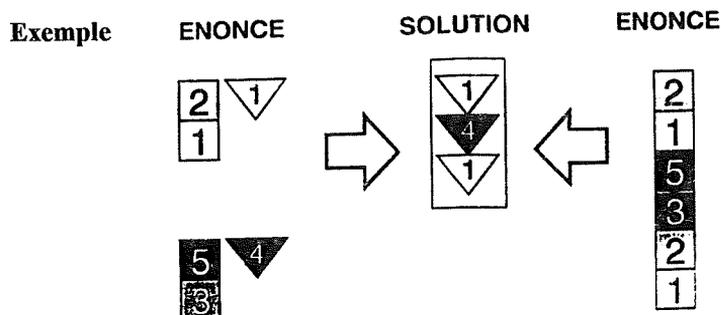


Figure 2.

L'enseignant place, le long du bord gauche de la plaque, deux séquences de carrés: la séquence carré rouge + carré bleu, associée à la pointe bleue, et la séquence carré jaune + carré vert, associée à la pointe orange. Le long du bord droit de la plaque, l'enfant voit la séquence de carrés suivante: rouge, bleu, jaune, vert, rouge, bleu et il doit construire au milieu de la plaque la séquence de pointes qui aurait produit ce résultat (il s'agit de la séquence de pointes suivante: bleue, orange, bleue).

Certains exercices de ce type permettent de confronter l'apprenant, de manière concrète, à des situations qui l'amènent à découvrir le raisonnement par l'absurde. D'autres exercices permettent à l'enfant de proposer plusieurs solutions différentes mais toutes correctes.

3. La recherche des procédures.

L'enfant doit construire (dans la zone des procédures) la séquence de carrés associée à chacune des pointes imposées, de manière à pouvoir réaliser avec ces pointes la séquence de carrés placée dans la zone d'effectuation, en respectant le programme donné dans la zone du programme. L'enfant doit ici recréer à partir de primitives, les procédures que l'expérimentateur (ou l'enseignant) n'a pas décrites mais qu'il a utilisées, pour obtenir un résultat qui lui est bien visible sur la plaque.

Exemple

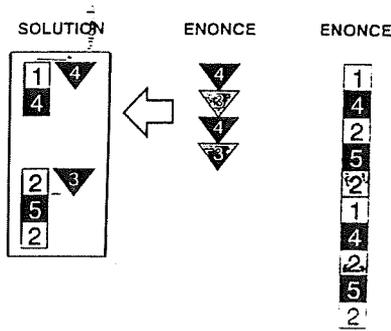


Figure 3.

L'enseignant place au milieu de la plaque la séquence de pointes suivante: orange, vert, orange; il place aussi le long du bord droit de la plaque la séquence de carrés suivante: bleu, orange, rouge, jaune, rouge, bleu, orange. Le sujet doit construire le long du bord gauche de la plaque la définition de chacune des pointes employées: pointe orange est ici le nom de la séquence carré bleu + carré orange, pointe verte est le nom de la séquence carré rouge + carré jaune + carré rouge.

Les exercices de ce type conduisent l'apprenant à rechercher les régularités et les symétries qu'il peut observer dans la séquence de carrés placés à sa droite.

4. La notion de procédures imbriquées.

L'enfant doit construire dans la zone d'effectuation la séquence de carrés correspondant aux procédures et à la suite de pointes données: la démarche demandée à l'enfant ressemble fort à celle qui lui est demandée lors de la première phase (l'exécution du programme), néanmoins les procédures proposées sont définies, cette fois, à l'aide de carrés et de pointes.

Exemple

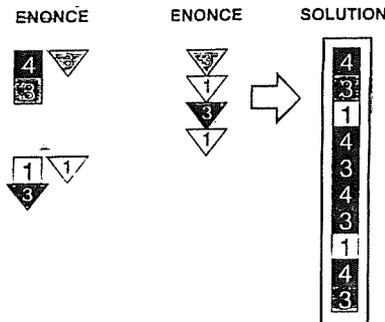


Figure 4.

L'enseignant propose à l'enfant deux pointes. La première est la pointe verte définie comme le nom de la séquence suivante: carré orange + carré vert. La seconde est la pointe bleue et est définie comme suit: c'est le nom de la séquence

carré bleu + pointe verte. Au milieu de la plaque, l'enfant voit une séquence de quatre pointes: pointe verte, pointe bleue, pointe verte et pointe bleue. L'apprenant doit alors réaliser la séquence de carrés correspondant à cette séquence de pointes. Au début, l'enfant n'a pas de problème: il place un carré orange puis un carré vert. La situation se complique lorsque le sujet aborde la deuxième pointe, une pointe bleue: il faut d'abord placer un carré bleu, ensuite il faut placer ce qui est représenté par la pointe verte, c'est-à-dire une séquence de carrés comprenant le carré orange suivi du carré vert. L'enfant doit alors continuer de cette manière pour les autres pointes du programme proposé.

5. L'introduction de la pointe directionnelle.

Au cours des exercices précédents, nous n'avons jamais utilisé la pointe jaune pour nommer une procédure: cette pointe est réservée pour introduire la notion de direction. L'enseignant la présente aux enfants comme étant une petite souris dont le museau (l'angle de la pointe) indique la direction à suivre pour placer les carrés des procédures lors de l'exécution. Ceci peut se faire de deux manières:

a) dans le programme.

La pointe directionnelle (pointe jaune, codée 5) est d'abord introduite dans le programme. Précédant chaque fois une pointe de procédure, elle a pour but d'indiquer dans quelle direction l'enfant doit placer les carrés correspondant à la procédure lors de l'exécution du programme. Cette technique peut être comparée à l'emploi du repère absolu pour la tortue Logo. Nos observations montrent qu'il faut introduire ce type d'exercices en premier lieu lorsque l'on s'adresse aux très jeunes enfants.

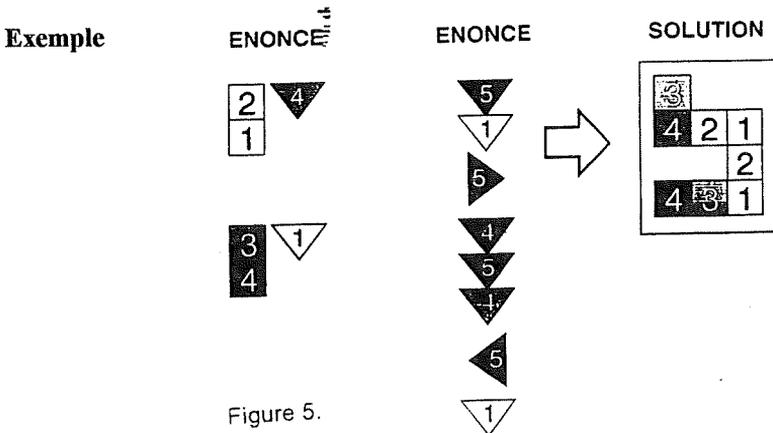


Figure 5.

Employons deux procédures définies comme suit: pointe orange est le nom de la séquence "carré rouge + carré bleu", pointe bleue est le nom de la séquence "carré vert + carré orange". Le "programme" proposé est le suivant: une pointe jaune 'pointant son museau' vers le bas, pointe bleue, pointe jaune vers la droite, pointe orange, pointe jaune vers le bas, pointe orange, pointe jaune vers la gauche,

pointe bleue. L'enfant doit alors placer les carrés comme suit: un carré vert, en dessous de celui-ci un carré orange, à droite de ce dernier un carré rouge, à droite de ce dernier un carré bleu, en dessous de ce dernier un carré rouge, etc. ...

Dans notre cas, à cause des contraintes techniques du matériel employé, il s'agit toujours d'un angle de 90° ou de 180° et d'un repère absolu. En effet, l'enfant doit placer les carrés dans la direction que lui indique la pointe jaune du programme: vers le haut, le bas, la gauche ou la droite de la plaque.

b) dans les procédures.

La pointe directionnelle est, cette fois, placée à la fin de chacune des procédures. Elle indique cette fois la direction à suivre "pour continuer" l'exécution: le repère utilisé ici est le repère relatif puisque la direction à suivre "pour continuer" dépendra de la position de la procédure dans l'exécution.

Exemple

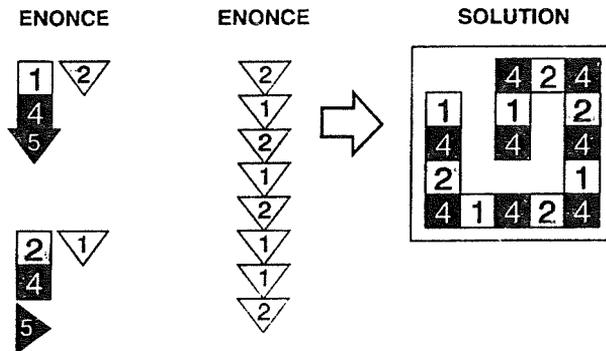


Figure 6.

L'enseignant peut proposer les deux procédures suivantes: pointe rouge et pointe bleue. Pointe rouge est ici le nom de la séquence suivante: carré bleu, carré rouge et une pointe jaune 'dont le museau indique continuer tout droit'; pointe bleue est par contre le nom de la séquence suivante: carré rouge, carré orange et une pointe jaune 'dont le museau indique quart de tour à droite'. Le programme proposé est simple: pointe rouge, pointe bleue, pointe rouge, pointe bleue, pointe rouge, pointe bleue, pointe bleue, pointe rouge. Pour réaliser ce programme, l'enfant doit placer les deux carrés de la pointe rouge l'un au-dessus de l'autre, puis continuer en plaçant immédiatement en dessous les deux carrés de la pointe bleue l'un au-dessus de l'autre, ensuite il doit placer à la droite du dernier carré placé les deux carrés de la pointe rouge, etc. ...

Cet emploi de la pointe directionnelle se rapproche plus de celui qui est le plus souvent employé en Logo avec les primitives GAUCHE et DROITE. Dans ce cas, l'enfant doit non seulement tenir compte de l'indication donnée par la pointe jaune qui figure dans chacune des procédures, mais aussi de la position et de l'orientation qui précèdent immédiatement. Les exercices les plus fréquents en Logo ont habitué des enfants qui savent lire et écrire à tenir compte de la position et de l'orientation de la tortue à chaque moment. Dans le cas de la mosaïque, un exercice similaire peut être fait de manière non-verbale avec des enfants de 4 et 5 ans.

6. Remarques.

Cette approche permet de proposer à de très jeunes enfants des exercices assez complexes: une des procédures est définie à l'aide d'une pointe alors que cette pointe représente elle aussi une procédure définie à l'aide d'une autre pointe.

Exemple

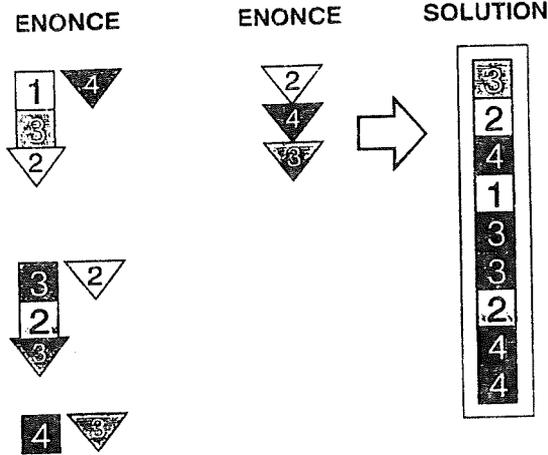


Figure 7.

La démarche utilisée par l'enfant, dans le cas de procédures imbriquées, se rapproche de celle utilisée en Logo lorsque pour créer une nouvelle procédure (procédure appelante), on utilise, dans le programme, d'autres procédures déjà définies ou que l'on va définir (procédures appelées).

Exemple

Soit les définitions suivantes: On obtient la figure ci-dessous:

```

Pour 4DRAPEAUX
  DRAPEAU
  DR 90 AV 50
  DRAPEAU
  DR 90 AV 50
  DRAPEAU
  DR 90 AV 50
  DRAPEAU
Fin

```

```

Pour DRAPEAU
  CARRE
  RE 50
Fin

```

```

Pour CARRE
  REPETE 4[AV 40 DR 90]
Fin

```

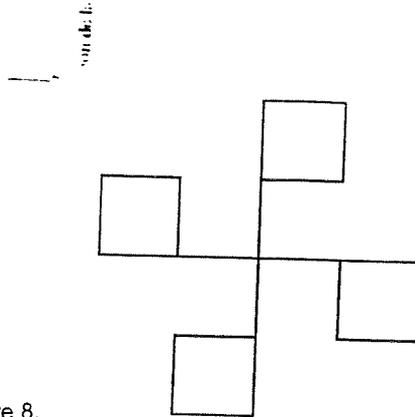


Figure 8.

Il est même possible d'aborder grâce à la mosaïque une pseudo-récursion en envisageant le cas où une procédure est définie à l'aide d'elle-même. Il ne s'agit pas ici de la vraie récursion car la mosaïque n'a pas de compteur.

Exemple

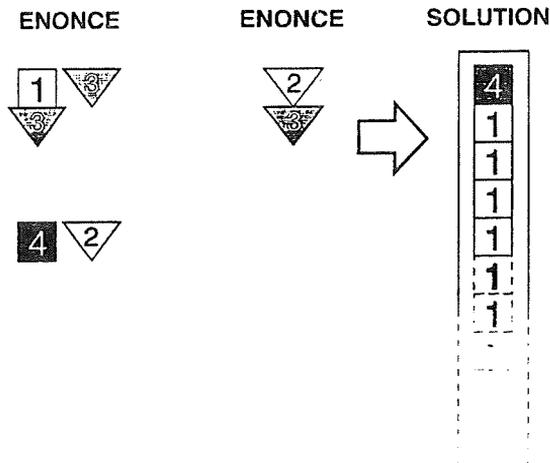


Figure 9.

On peut pourtant comparer de telles définitions à la définition suivante en Logo:

Pour CARRE
AV50 DR90
CARRE
Fin

Une telle définition de CARRE force la pauvre tortue Logo à parcourir sans fin les quatre côtés du même et éternel carré. Ce type d'exercice, avec la mosaïque, permet d'aborder avec les enfants de 4 et 5 ans la notion d'infini puisqu'un tel exercice ne se termine jamais.

Les relations entre les exercices avec pointe jaune directionnelle et le vrai Logo sont évidentes. Dans les deux cas les enfants apprennent à utiliser leur corps propre comme repère, ils apprennent dans le cas du repère relatif à se voir "comme la souris" et donc ils apprennent à tenir compte de l'existence et du point de vue de l'autre. Mais, dans le cas de la mosaïque ils n'ont que 4 ou 5 ans.

D. Rôle de l'enfant confronté à ce matériel.

Lorsque l'enfant utilise la mosaïque pour exécuter ou découvrir des programmes décrits en termes de pointes et de carrés, il accomplit en fait une activité informatique d'une manière inhabituelle: ici c'est le cerveau de l'enfant qui joue le rôle de processeur, les mains du sujet et les différents clous servent de clavier tandis que la plaque ne sert que d'écran.

E. Parallélisme entre la mosaïque et certains apprentissages.

a) lecture

On peut établir un parallèle entre certaines méthodes d'apprentissage de la lecture et les différents exercices proposés à l'aide de la mosaïque.

L'exécution du programme peut être comparée à certaines méthodes où l'enfant apprend d'abord la lettre et le son correspondant; l'enfant apprend ensuite à assembler les lettres en syllabes, les syllabes en mots et enfin les mots en phrases. Il s'agit de synthétiser des lettres pour former des sons et cela sans nécessairement faire appel au sens des mots.

La recherche du programme semble se rapprocher davantage de certaines méthodes où l'on part d'une phrase complète que l'on découpe en mots. On peut associer les procédures aux mots, le programme à découvrir à la syntaxe et l'exécution à la phrase.

La reconstruction des procédures peut être mise en rapport avec des exercices de lecture où l'enfant, partant d'une phrase complète, doit rechercher sur des étiquettes (les pointes données) les différents mots qui la composent.

b) techniques de raisonnement

Lors d'exercices de recherche du programme, l'enfant est souvent confronté à des difficultés inattendues qu'il surmonte pourtant aisément: certains exercices ne peuvent en effet être résolus qu'en commençant la construction demandée par la fin, dans d'autres cas l'enfant doit (dès 4 ou 5 ans !) faire un raisonnement par l'absurde. Certains exercices peuvent être résolus de plusieurs manières, d'autres ont plusieurs solutions. Rien de tout cela n'arrête nos jeunes élèves !

c) observation de l'image et perception d'indices pertinents

Il est tentant de faire un rapprochement entre la démarche de recherche des procédures utilisées et la démarche mentale utilisée par l'enfant lorsqu'il décompose un dessin qu'il doit réaliser en ses divers éléments.

Il faut néanmoins souligner qu'aucun de ces apprentissages ne peut être pris isolément; ils sont tous liés par un même élément: l'enfant doit apprendre à formuler des hypothèses (faire des prédictions), à les tester et à les valider. Or les différents types d'exercices qu'il est possible de proposer à l'aide de la mosaïque favorisent tous cet apprentissage d'une technique hypothético-déductive.

F. Stratégies cognitives, mosaïque et langage Logo.

Dans le cadre de nos recherches sur les stratégies cognitives utilisées par des enfants pour prendre de l'information, la traiter et produire une réponse, nous avons observé des sujets en train de résoudre des problèmes en Logo. Pour mieux les observer, nous avons créé un logiciel qui nous permet d'enregistrer toutes les interactions sujet-ordinateur, mais qui reste transparent pour le sujet (Lowenthal, Marcourt & Solimando, 1993a). Notre technique nous a permis de comparer les productions et les stratégies de sujets normaux et de sujets handicapés (Lowenthal, 1993). Elle nous a aussi permis de décrire une hiérarchie de stratégies (Lowenthal, Marcourt & Solimando, 1992, 1993b). Cette hiérarchie comprend 5 étapes principales et une étape parallèle. Il est important de souligner ici que ces 6 étapes apparaissent également dans le cadre d'une hiérarchie adaptée à la mosaïque. Il y a donc un parallèle supplémentaire entre l'emploi du Logo et celui d'exercices de type informatique présentés par la mosaïque. Ces étapes sont décrites ci-dessous.

PARALLELE ENTRE LA HIERARCHIE LOGO ET LA GRADATION D'EXERCICES DE MOSAIQUE:

Hiérarchie observée chez de jeunes enfants.

Nous présentons à chaque niveau d'abord une description de la stratégie, puis un exemple en Logo et un exemple réalisé à l'aide de la mosaïque.

1. Mode direct

L'utilisateur emploie exclusivement des instructions de base pour construire pas à pas un dessin qui apparaît au fur et à mesure sur l'écran (dans le cas du Logo) ou sur la grille (dans le cas de la mosaïque); aucun concept nouveau n'est défini.

Exemple Logo

L'utilisateur n'emploie que des instructions logo de base pour construire pas à pas un dessin qui apparaît au fur et à mesure sur l'écran de l'ordinateur. Il peut dessiner un carré comme suit:

AV 40 DR 90 AV 40 DR 90 AV 40
DR 90 AV 40 DR 90

Exemple mosaïque

Le mode direct en Logo correspond à la copie en mosaïque. Le principe de la copie est simple: l'animateur place des clous un à un sur une plaque et l'enfant doit "faire la même chose" sur sa plaque (correspondance biunivoque).

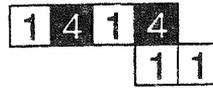


Figure 10.

2. Mode direct procéduralisé

L'utilisateur définit un concept nouveau en faisant appel uniquement à des instructions de base.

Exemple Logo

L'utilisateur peut définir **CARRE** et les **MAISON** comme suit:

Pour **CARRE**
AV 40 DR 90
AV40 DR90
AV40 DR90
AV40 DR90
Fin

Pour **MAISON** (définition de type 2)
REPETE 4[AV 40 DR 90]
AV 40 DR 30
REPETE 3[AV 40 DR 120]
Fin

Cet utilisateur n'a établi aucun lien entre son **CARRE** et le carré qui apparaît dans son dessin de **MAISON**.

Exemple mosaïque

Tous les exercices de la gradation où pointes sont définies uniquement à l'aide de carrés répondent à ce critère.

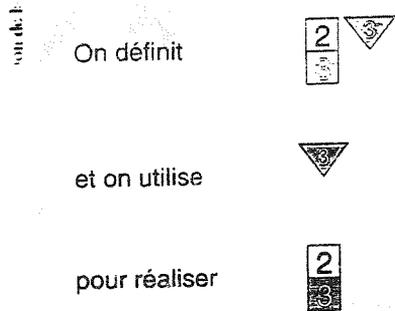


Figure 11.

3. Procédure simple

L'utilisateur définit un concept en faisant appel uniquement à des définitions de niveau 2 ou à des définitions de niveau 2 accompagnées d'instructions de base.

Exemple Logo

L'utilisateur peut définir d'abord **CARRE** comme ci-dessus, puis **TRIANGLE** d'une manière semblable, et enfin définir **MAISON** comme indiqué à la page suivante:

Pour **MAISON** (définition de type 3)
CARRE
AV 40 DR 30
TRIANGLE
Fin

Cet utilisateur a établi un lien entre son **CARRE**, son **TRIANGLE** et ceux qui apparaissent dans sa **MAISON**.

Exemple mosaïque

Ce niveau correspond à la notion de procédures imbriquées en mosaïque: les procédures proposées sont définies à l'aide de carrés et/ou de pointes et l'enfant devra exécuter le programme uniquement au moyen de carrés. Donc, pour définir une nouvelle procédure (procédure appelante), on utilise, dans cette procédure, d'autres procédures déjà définies (procédures appelées).

On définit



avec



Figure 12.

4. Procédure supérieure simple

L'utilisateur définit un concept en faisant appel soit à une ou plusieurs définition de type 3 exclusivement soit à des définitions de type 3 accompagnées de définitions de type inférieur et/ou d'instructions de base.

Exemple Logo

L'utilisateur peut définir d'abord **CARRE**, **TRIANGLE** et **MAISON** comme ci-dessus, il ajoute une définition de type 2 pour le concept **VITRE** et combine le tout pour créer une **VILLA**.

Exemple mosaïque

Cette étape correspond à la notion de procédures imbriquées utilisée à un niveau plus élevé: une des procédures proposées est définie à l'aide d'une pointe; cette pointe représentant elle aussi une procédure définie à l'aide d'une autre pointe.

Pour VITRE (définition de type 2)
AV5 DR9OAV10 DR90
AV5 DR9OAV 10 DR90
Fin

Pour VILLA (définition de type 4)
MAISON
LC GA 30 RE 40 DR 90 AV 15
GA9OAV20 BC
VITRE
Fin

Dans ce cas, la définition ne sera de type 4 que si l'utilisateur a défini sa MAISON en utilisant CARRE et TRIANGLE; si l'utilisateur a employé la définition de MAISON décrite au niveau 2, sa définition de VILLA ne sera que du niveau 3.

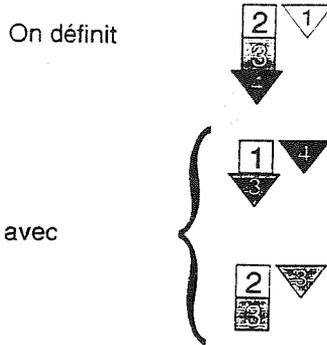


Figure 13.

5. Procédure supérieure complexe

L'utilisateur définit un concept en faisant appel uniquement à une ou plusieurs définitions de type 4 ou à au moins une définition de type 4 complétée par une ou plusieurs définitions de type 3 et par certaines instructions de base.

Exemple Logo

L'utilisateur définit à présent une RUE en utilisant plusieurs fois la définition de VILLA donnée ci-dessus (type 4) et quelques exemplaires de la MAISON définie juste avant (type 3) avec les instructions de base permettant de déplacer la tortue Logo.

Exemple mosaïque

Ce type d'exercice n'est pas prévu dans la gradation que nous avons utilisée avec des enfants de 5 et 6, ans mais il est tout à fait concevable.

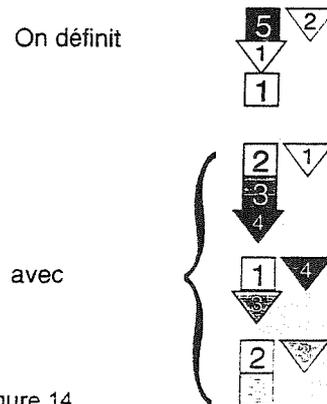


Figure 14.

6. Appel direct de procédure

L'utilisateur réalise un APPEL DIRECT lorsqu'il utilise des concepts définis antérieurement (quel que soit le type de définition utilisé pour eux) comme s'il s'agissait d'instructions de base: le sujet utilise donc ces définitions pour produire un dessin qu'il ne nomme pas et qu'il ne relie donc pas de manière à créer une nouvelle définition..

Exemple Logo

L'utilisateur dessine une rue, qu'il ne nomme pas, en donnant les instructions suivantes:

**VILLA LC ORIGINE DR 90 AV 40
GA 90 BC MAISON ...** (où les définitions de VILLA et MAISON sont respectivement de type 4 et de type 3).

Exemple mosaïque

Ce niveau correspond également à une forme de copie mais à un niveau de difficulté plus élevé: L'animateur place des séquences de clous que l'enfant doit reproduire sur sa plaque. Les procédures utilisées ne sont pas reliées en un "grand" programme.

L'utilisateur place des carrés sur sa grille en réalisant d'abord la séquence correspondant à la pointe 3 de la figure 11, puis juste en dessous la séquence correspondant à la pointe 4 de la figure 12; mais le sujet ne nomme pas son nouveau dessin, il ne lui attache aucune pointe et il ne relie donc pas les deux définitions qu'il a utilisées.

G. Résultats

L'emploi "informatique" du matériel décrit ci-dessus développe différents facteurs importants intervenant dans l'apprentissage de la lecture et de la mathématique (structuration spatiale, structuration temporelle, discrimination visuelle, correspondance terme à terme, ...), de sorte que les enfants qui ont manipulé les mosaïques sont mieux préparés pour aborder ces apprentissages. Ce matériel est tout indiqué pour proposer des exercices de remédiations aux enfants qui éprouvent des difficultés dans les apprentissages cités ci-dessus.

Les démarches utilisées dans la gradation d'exercices d'introduction à la programmation informatique et la confrontation des enfants avec des problèmes logiques qu'ils apprennent à résoudre, constituent, pour eux, un atout important lorsqu'ils commencent à travailler en Logo (Lowenthal, 1 987a, 1 988) .

Nous avons également constaté, après différentes expériences avec des enfants de 3ème maternelle et de 1ère primaire (âgés de 4, 5 et 6 ans), que ceux qui ont travaillé avec les mosaïques obtiennent généralement de meilleurs résultats scolaires dans le domaine de la résolution de problèmes.

Le travail accompli avec ce matériel pour intervenir auprès d'un enfant aphasique et auprès d'autres sujets souffrant de troubles de la communication et de la cognition, tels des enfants déficients sensoriels, montre clairement que ce matériel **concret, indépendant de tout arrière-plan cognitif, n'exigeant aucun prérequis**, permet de favoriser le développement de la logique, et ainsi de la communication chez certaines personnes handicapées. En effet, la manipulation de ce matériel leur permet d'établir les bases d'une structure logique indispensable à la communication mais qu'ils ne parviennent pas à établir de manière naturelle (verbale) à cause d'un obstacle affectif ou physiologique. La technique que nous proposons doit permettre à la personne handicapée d'acquérir le point de départ qui lui manque pour faire débiter un processus d'acquisition du langage personnalisé et efficace (Lowenthal & Saerens, 1986; Lowenthal, 1990).

H. Conclusions

L'utilisation de la mosaïque de la manière décrite ci-dessus constitue une étape importante dans un apprentissage informatique. Les manipulations indispensables sont à la portée de jeunes enfants qui ne savent ni lire ni écrire et pour qui l'accès à l'ordinateur est donc limité, voire impossible. En outre ces manipulations permettent aux apprenants de passer par des manipulations concrètes avant d'aborder le semi-concret et les raisonnements abstraits, respectant ainsi les étapes décrites par Piaget (1947); ces manipulations permettent surtout aux enfants d'utiliser d'abord une représentation par l'action avant de passer à une représentation par l'image et ensuite à une représentation symbolique. Bruner (1966) a montré l'importance de la succession de ces trois modes de représentation pour le développement de l'enfant.

De plus, l'emploi de la mosaïque suivant la gradation que nous avons établie permet d'installer (au niveau maternel) et de renforcer (au niveau primaire) bon nombre de capacités fondamentales indispensables à l'apprentissage de la lecture. En effet, que cet apprentissage se fasse par une méthode où l'enfant commence par identifier les lettres et les sons correspondant à chacune d'elles, puis les associations en syllabes, puis en mots (partant d'éléments simples pour en effectuer la synthèse) ou qu'il se fasse par une méthode qui privilégie le sens lors de l'apprentissage et qui, suivant un processus d'analyse, descend des formes globales vers des formes élémentaires, les différentes démarches utilisées lors de cette gradation développent les aptitudes nécessaires à cet apprentissage. De plus, ce matériel constitue un excellent support pour provoquer des raisonnements du même type que ceux utilisés lors de l'apprentissage de la lecture (prise d'indices, anticipation, émission et vérification d'hypothèses).

Au niveau de la **mathématique**, cette gradation initie l'enfant aux notions de rythme, d'espace, de temps, de correspondance terme à terme, de nombre, d'ensemble et à la résolution de problèmes.

Enfin, cet emploi de la mosaïque pour aborder des tâches de style informatique, sans utiliser d'ordinateur, permet de démythifier l'informatique.

L'ordinateur ne doit être qu'une seule chose: un outil rapide de traitement de l'information au service des humains qui l'utilisent. Certains pédagogues l'oublient parfois: en voyant certaines formes d'enseignement assisté par ordinateur on a parfois l'impression que le créateur du logiciel a vu l'enfant soumis à l'ordinateur au lieu de le voir dominant l'ordinateur comme acteur actif.

Francis LOWENTHAL

Université de Mons-Hainaut

Place du Parc 20, B-7000 Mons, Belgique

1. Bibliographie.

- Bruner J. S., 1966, On cognitive growth, in Studies in cognitive growth, Bruner J. S., Olver R. R. & Greenfield P. M. eds, John Wiley, 1-29 & 30-67.
- Gurtner J.-L. & Retschitzki J., 1991, Logo et apprentissages, Delachaux et Niestlé.
- Ledru L. et Lowenthal F., 1988, Utilisation des NVCD dans une optique de diagnostique ou de recherche clinique, Communication and Cognition, 21, 17-27.
- Lowenthal F., 1978, Logic of natural language and games at primary school, Revue de Phonétique Appliquée, vol 46-47, 133-140.
- Lowenthal F., 1984, Generative grammar, programming language and communication development, Proceedings du congrès AILA Brussels 84, 1700.
- Lowenthal F., 1985, Pegboard as basis for programming -in 5- and 6-year olds-, Psychology of Mathematics Education 9, 47-52.
- Lowenthal F., 1986a, Relevance of typically logico-mathematical formalisms for research in psychology, Logique et Analyse, vol 116, 501-508.
- Lowenthal F., 1986b, Non-verbal communication devices: their relevance, their use and the mental processes involved, in Pragmatics and education, Lowenthal F. & Vandamme F. eds., Plenum Press, 29-46.
- Lowenthal F., 1987a, Concrete representations of formal systems: an example of transfer in cognitive psychopedagogy, International congress Communication and cognition. applied epistemology, 198-200.
- Lowenthal F., 1987b, Représentation concrète de systèmes formels et structuration d'une communication, Revue de Phonétique Appliquée, vol 82-83-84, 231-245.
- Lowenthal F., 1988, Concrete introduction to programming languages and observation of Piagetian stages - clinical interviews -, Psychology of Mathematics Education 12, 479-486.
- Lowenthal F., 1990, Pegboard as structuring element for the verbal language, Revue de Phonétique Appliquée, vol 95/97, 255-262.
- Lowenthal F., 1992, Formalisms for clinical observations, Advances in nonverbal communication, Poyatos F. ed., John Benjamin, Amsterdam/Philadelphia, 125-143.
- Lowenthal F., 1993, Spatial and logical misconceptions observed by means of Logo sessions in normal and handicapped subjects, Actes du "Third International Seminar on Misconceptions and Educational Strategies in Science and Mathematics", Cornell University.

- Lowenthal F., Marcourt C. & Solimando C., 1992, Evolution of problem solving strategies in some handicapped children using Logo, International journal of Psychology: Abstracts du XXVème Congrès International de Psychologie, 151.
- Lowenthal F., Marcourt C. & Solimando C., 1993a, Observing computer for study of problem solving strategies, Actes de PEG 93: AI tools and the classroom - Theory into Practice, Moray House Institute of Education, Edinburgh, 375-388.
- Lowenthal F., Marcourt C. & Solimando C., 1993b, The Logo observer in special education, Actes de PEG 93: AI tools and the classroom - Theory into Practice, Moray House Institute of Education, Edinburgh, 731-737.
- Lowenthal F. & Saerens J., 1986, Evolution of an aphasic child after the introduction of NVCDs, in Praquatics and education, Lowenthal F. & Vandamme F. eds., Plenum Press, 301-330.
- Papert S., 1980, Mindstorm. Children. computers and powerful ideas, Basic books.
- Piaget J., 1947, La psychologie de l'intelligence, Colin.
- Saerens J., 1985, Semantic components and syntactic sequences introduced by means of multicoloured plastic pegs, Communication & Cognition, vol 17, 397-403.

Les apports du « jeu de l'enfant-robot » à la didactique de l'informatique

Eric GREFF

présentée le vendredi 12 avril 1996 à 9h00

• « Peut-on introduire une pensée algorithmique auprès des très jeunes enfants (4-6 ans) ? Peut-on, en tous cas, développer avec eux une série d'activités homogène dont on peut penser qu'elle les aidera, plus tard, dans leur approche de l'informatique ? »

Qu'est-ce que le jeu de l'enfant-robot ?

• Le jeu de l'enfant-robot est une méthode nous avons élaboré et qui vise à développer la pensée algorithmique et logique chez les très jeunes enfants (4-6 ans). Dans ce travail, l'élève peut être tour à tour programmeur ou programmé. Dans ce dernier cas, il agit à la manière d'un robot et ne se meut qu'en fonction des cartes qui lui sont montrées. Celles-ci sont constituées de flèches indiquant un déplacement précis auquel le robot doit se conformer (avance d'un pas, pivote à droite...). Un paquet de cartes détermine ainsi un parcours précis et constitue donc ce que nous nommerons un programme.

- A partir de cette base, se développent plusieurs types d'activités :
 - ◁ constituer des programmes.
 - ◁ les faire exécuter par un autre et veiller à leur bonne exécution.
 - ◁ exécuter un programme soit même en dépilant son paquet de cartes.
 - ◁ coder et décoder un parcours précis...

• Cette méthode aborde donc des domaines aussi variés que l'algorithmique, le jeu chez l'enfant, la robotique, la représentation, la construction de l'espace et du temps, la signalétique, le codage...

Les apports du « jeu de l'enfant-robot » à la didactique de l'informatique

Le jeu de l'enfant-robot

• En Janvier 1995, nous avons publié aux éditions « Nathan Pédagogie » un ouvrage intitulé « Logique et Algorithme avec les 5/6 ans ». Celui-ci, librement inspiré des expériences de la tortue de sol, se veut une méthode à destination des enseignants de Moyenne et Grande Section de l'Ecole Maternelle ainsi que de ceux du CP afin qu'ils puissent initier facilement leurs élèves au « jeu de l'enfant-robot ».

• Cette activité, destinée à de jeunes élèves ne sachant pas encore lire, est conçue pour fonctionner, à faible coût, dans la classe et sans utiliser ni ordinateur, ni machine d'aucune sorte. Partant du constat lucide que les Ecoles Maternelles sont dotées de peu d'argent, d'enseignants dynamiques mais peu ou pas formés à l'informatique et d'élèves pleins de bonne volonté mais en cours d'acquisition des

« apprentissages fondamentaux », nous nous sommes attaché à créer et à utiliser un environnement à la fois simple et homogène.

- Les réflexions concernant l'apprentissage de l'algorithmique et de la programmation ont été en permanence les moteurs et les référents de cette méthode. La programmation structurée constitue l'un de ses points d'ancrage théorique fort. Parmi les nombreuses questions ayant pu émerger, l'une des principales est la suivante : « Si l'on considère que l'art de programmer fait essentiellement appel à une structuration particulière de la pensée, peut-on, très tôt dans la scolarité de l'enfant, lui inculquer des notions l'aidant et/ou le préparant à acquérir ce type de structures mentales ? ».

- Il s'agissait donc de créer, non pas un langage informatique mais un environnement à la fois conceptuellement rigoureux et permettant l'expérimentation de l'utilisation des structures algorithmiques de base. Les possibilités d'activités développées à l'école Maternelle étant riches et variées, nous nous sommes attachés à relier au « jeu de l'enfant-robot » les autres notions essentielles habituellement abordées avec les très jeunes enfants dans le cadre scolaire.

Premiers programmes

- Seymour Papert désignait la tortue comme « un objet pour penser avec » auquel les enfants devaient fournir des instructions de mouvement afin qu'il puisse effectuer un parcours déterminé. Dans ce jeu, l'enfant est « un individu pour penser à la place ». En effet, c'est l'élève lui-même qui vit les déplacements avec son corps sans passer par l'intermédiaire d'un objet représentationnel tel que le robot de plancher. Ce faisant, il appréhende directement, puisqu'il en est l'acteur principal, ce que peut être une machine qui exécute précisément des instructions sans s'en écarter ni improviser. Il peut ainsi se rendre compte du rapport existant entre instruction et mouvement, entre programme et exécution. Comme cette activité utilise des ordres de déplacement du type « Avance d'un pas », « Recule d'un pas », « Pivote à droite », « Pivote à gauche », elle développe en outre des capacités ayant trait à la motricité et à la latéralisation qui sont essentielles à ce stade du développement de l'enfant.

- Non content d'être programmé, l'enfant joue également le rôle du programmeur, assemble lui-même les cartes-instructions qui lui permettront de constituer un programme qu'il présentera ensuite à l'un de ses camarades pour qu'il l'exécute. Il apprend ainsi combien il est important d'anticiper, de prévoir, d'être rigoureux, de « se mettre à la place de », de ranger ses instructions dans l'ordre choisi. Il appréhende également l'aspect séquentiel d'un programme informatique et pourra utiliser ses nouvelles connaissances en « planification des actions » dans de nombreux autres domaines tels que les tris, les classifications, les résolutions de problèmes simples qui sont des activités très présentes à l'Ecole Maternelle.

Le langage

- Les cartes-instructions utilisées dans cette méthode sont entièrement graphiques et constituées essentiellement de flèches représentant les mouvements à effectuer. Ce type d'images créées par nécessité pour des enfants ne sachant pas lire permet, en outre, de les préparer à assimiler l'association entre image et instruction qui est de plus en plus présente dans les logiciels et les systèmes d'exploitations modernes par l'intermédiaire des interfaces graphiques et des icônes. Au-delà du monde informatique, les logos et les pictogrammes font partie de notre univers ; il n'est probablement pas inutile d'enseigner à nos jeunes élèves que les images ont un sens et de les initier à ce nouveau type de communication.

- L'ensemble des cartes-instructions créées constitue un langage de commande à part entière avec des cartes ayant un rôle de délimiteur, c'est à dire des cartes servant à structurer le programme, d'autres engendrant uniquement des mouvements de l'enfant-robot. Ainsi, les programmes possèdent une en-tête, un corps et une fin et ils obéissent à des règles syntaxiques précises. Une fois les cartes-instructions choisies, l'enfant sait les empiler convenablement avant de les dépiler à l'intention de son camarade-robot qui les exécutera. Il se peut également que le même élève crée son programme et emporte avec lui sa pile d'instructions qu'il dépile et exécute, dans ce cas, lui-même. Il manipule un véritable langage dont il est alors l'interprète, sait qu'à une carte-instruction correspond une seule action et que celle-ci doit être entièrement exécutée avant d'aborder la carte suivante.

- Des structures de contrôle simplifiées existent dans le langage. Elles permettent d'introduire de manière douce les notions d'itération et d'alternative dont on peut espérer qu'une familiarisation précoce favorisera leur apprentissage ultérieur.

La représentation

- Afin de travailler également sur une représentation écrite des parcours effectués par l'enfant-robot, un petit personnage orienté nommé « Algor » a été imaginé. Celui-ci se déplace sur un quadrillage en utilisant exactement les mêmes cartes que celles auxquelles obéit l'enfant-robot. Outre qu'Algor permet de multiplier et de varier les exercices écrits, il incite les élèves à plonger dans le monde de la représentation de parcours. Il ne s'agit plus de se mouvoir dans la classe mais d'imaginer que ce petit personnage au nez pointu est une partie de nous-mêmes se déplaçant sur la feuille. Il faut alors se mettre « à la place de... », appréhender la droite et la gauche de l'autre, effectuer le passage du plan vertical de l'individu au plan horizontal de la figurine...

- Au-delà de ce travail spécifique, sont abordés les problèmes plus généraux de représentation de parcours (labyrinthes, cheminements...) très présents à l'école Maternelle ainsi que ceux concernant les codages et décodages de trajets. On pourra s'initier ainsi au code de la route ou à la lecture de plans. Ce type d'activité permet notamment de s'intéresser à l'explicitation de parcours qui fait intervenir les notions

de séquentialité et d'orientations relatives qui sont également étudiées dans la méthode.

Le robot

• Dans cette méthode, on demande à l'enfant d'agir à la manière d'un robot afin d'exécuter les instructions qui lui sont montrées. On inculque l'idée que le robot ne fait que ce que le programmeur lui demande de faire, qu'il n'agit pas à sa guise et ne connaît qu'un nombre limité de tâches. Outre les impératifs de rigueur que cela lui enseigne, l'élève se trouve confronté à la réalité de la robotique dont les médias lui donnent une idée plus que rocambolesque.

Les structures de contrôle

Des structures de contrôle simplifiées existent dans le langage. Elles permettent d'introduire de manière douce les notions d'itération et d'alternative dont on peut espérer qu'une familiarisation précoce favorisera leur apprentissage ultérieur.

L'itération

L'itération est liée à la notion de variable informatique, que celle-ci intervienne dans la condition où dans le compteur de boucle. Cette notion de variable est peu courante et difficile à établir auprès des très jeunes enfants auxquels nous nous adressons.

Dans notre langage, l'itération se limite à une répétition indicée s'appliquant dans un premier temps à une seule carte-action en y apposant un papillon collant de type Post-it indiquant le nombre de fois (inférieur ou égal à 4) où cette instruction doit être répétée. L'enfant-robot effectue alors le mouvement dessiné sur la carte autant de fois qu'il est précisé. Il doit donc se créer un « compteur mental » et mémoriser combien de fois il a déjà exécuté l'action et combien de mouvements il lui reste, en conséquence, à effectuer. Cette activité permet à l'élève de « s'imprégner » de la notion d'itération, de faire émerger la nécessité et les moyens de comptabiliser le nombre de cycles. On peut ensuite étendre cette notion de répétition à un bloc de 4-5 cartes qui seront alors réunies par un élastique.

L'alternative

L'alternative est une notion informatiquement difficile car elle n'existe pas formellement dans la vie courante. De plus, elle fait intervenir les expressions booléennes et « détruit » le caractère linéaire du programme.

Dans notre langage, l'alternative est restreinte à une structure de type « Si... Alors... ». Pour l'utiliser, on a créé des cartes-questions auxquelles l'enfant ne peut répondre que par oui ou par non. Les questions posées par celles-ci sont également graphiques et contiennent un point d'interrogation indiquant leur nature spécifique. Si la question posée à l'enfant a une réponse affirmative, alors le bloc d'instructions suivant (réuni par un élastique) sera exécuté. Dans le cas contraire, il sera ignoré. Cette activité permet de confronter l'enfant à cette structure d'alternative,

d'introduire l'idée de conséquence, d'entraîner l'élève à affirmer ses choix et à repérer les questions auxquelles on peut répondre par l'affirmative ou la négative.

L'après LOGO

- Il est flagrant, lorsqu'on lit les comptes-rendus d'expérimentation des années 86-87, de voir l'extraordinaire engouement qu'a suscité LOGO. A bien y réfléchir, cet enthousiasme n'est pas vraiment étonnant dans le contexte d'une micro-informatique naissante et par rapport à la richesse d'applications que pouvait apporter la tortue de Seymour Papert. Il est également frappant de constater combien ce langage est ensuite tombé en désuétude et combien il est désormais peu utilisé, voir renié.

- Les explications concernant l'échec de LOGO sont multiples. Certes le passage de ce langage de la tortue de sol au regretté (?) TO7, les applications trop mathématiques qui en ont été faites, le fait même que LOGO soit avant tout un langage de programmation limité servant essentiellement à « écrire » des programmes ayant trop souvent comme unique but de dessiner des maisons ont probablement concouru à sa disparition. LOGO est mort et enterré et il ne s'agit plus de le tuer, de le ressusciter ou de le réhabiliter mais d'essayer de comprendre, pour ne plus les reproduire, les mécanismes qui ont mené à sa disparition.

Conclusion

- Cette méthode n'a pas pour but l'apprentissage d'un langage informatique. Elle consiste à faire découvrir, à appréhender et à se familiariser avec des notions que le futur utilisateur averti de l'informatique, programmeur ou non, aura, plus tard, besoin de connaître. Pour ce faire, elle s'appuie sur un langage de commande graphique simplifié et sur les différents rôles que peut tenir l'enfant. En étant tour à tour programmeur et programmé, concepteur et exécutant, l'élève s'imprègne peu à peu de notions importantes. De manière plus large, il acquiert des connaissances dans des domaines aussi variés que la motricité, la latéralisation, la sémiologie de l'image, la communication, la rigueur, la résolution de problème, la représentation de parcours, le monde technologique... Certes, toutes ces connaissances pourraient être abordées par l'intermédiaire d'autres activités... qui ne développeraient pas forcément les concepts informatiques abordés par cette méthode et ne bénéficieraient pas de son caractère homogène.

Une Méthode Expérimentale

Un dispositif expérimental

Pour que ce projet devienne réalité, il fallait qu'il soit expérimenté dans les classes. C'est pourquoi, depuis 1993, 8 institutrices de la région de Versailles collaborent avec moi sur ce travail :

- Marilyn Buisson, institutrice en Grande Section de Maternelle au Chesnay, fut la première à expérimenter la méthode au cours de l'année scolaire 92/93 et co-signa donc l'ouvrage avec moi.

- En ce qui concerne l'année scolaire 93/94, l'expérimentation fut menée, au Chesnay, avec une classe de Moyenne Section (MS) de maternelle, une classe de Grande Section (GS) de maternelle (de la même école) et une classe de Cours Préparatoire (CP) dont une partie des élèves avaient déjà pris contact avec « l'enfant-robot » en 92/93 en Grande Section de maternelle. L'autre pôle se situait à Versailles dans une classe de CP et dans une classe d'adaptation décloisonnée qui regroupe de manière intermittente les élèves de CP en grande difficulté. Par commodité, je nommerai désormais cette classe d'Aide Individuelle Spécialisée : AIS.

- En ce qui concerne l'année scolaire 94/95, l'expérimentation s'est poursuivie, au Chesnay, avec les mêmes classes de Moyenne Section (MS) et de Grande Section (GS) de maternelle. A Versailles, outre la classe d'AIS précitée, trois nouvelles classes de CP se sont jointes au projet.

- En ce qui concerne l'année scolaire 95/96, l'expérimentation continue au Chesnay, avec les mêmes classes de Moyenne Section (MS) et de Grande Section (GS) de maternelle. A Versailles, outre la classe d'AIS précitée, une nouvelle classe de Grande Section (GS) avec un nombre important d'élèves en difficulté nous a rejoint. A Viroflay, une collègue se lance dans « l'aventure Algor » avec des enfants de Petite Section (PS).

La méthode constitue une progression pyramidale commune à ces 4 niveaux (MS, GS, CP, AIS). L'ordre dans lequel elle est abordée ainsi que son contenu demeurent identiques pour chaque niveau. Chaque étape doit être intégralement suivie, étant convenu que les enfants les plus âgés (ou ceux qui connaissent déjà « l'enfant-robot ») avancent plus vite dans la progression que les autres.

Les institutrices travaillant sur ce projet suivent régulièrement la progression. J'interviens ponctuellement dans les classes, soit pour observer, soit pour expérimenter avec les élèves différents aspects de la méthode. Cette activité évolue donc en permanence. La maquette finale du livre publié constitue la référence, datée de septembre 1994, sur laquelle je m'appuierai.

Depuis, ce travail est toujours expérimenté, annoté et rectifié au fur et à mesure de son utilisation afin de comprendre les erreurs commises, d'analyser les difficultés rencontrées, d'imaginer leurs remédiations et de peaufiner ainsi la méthode. Des aménagements sont encore apportés, des nouvelles questions se posent, la réflexion évolue, de nouveaux axes de recherches sont abordés.

Il est à noter que des étudiants de première ou deuxième année d'IUFM ainsi que des instituteurs en formation continue, ponctuellement en stage dans mes classes d'expérimentation, sont intervenus, en collaboration avec nous, sur le thème de l'enfant-robot. Ainsi, ils ont été amenés à élaborer des hypothèses et/ou des expériences. Leurs réflexions et leurs idées ont permis d'enrichir la méthode et ils sont, de ce fait, associés à ce travail.

Un matériel expérimental

Le travail d'expérimentation a pris plusieurs formes :

- Dans un premier temps, suivre la méthode, pas à pas, afin de vérifier, d'une part, si elle constituait une progression précise et suffisante pour l'enseignant et si, d'autre part, les enfants étaient aptes à la comprendre.
- élaborer des questionnaires ou des fiches d'évaluation permettant de répondre à des questions précises que nous nous posons par rapport à des s ou des techniques utilisées par les enfants, par rapport à des modifications, des aménagements ou des prolongements de la méthode.
- Organiser des tests collectifs et individuels de début et de fin d'année, avec des groupes témoins, afin de mesurer ce que la méthode pouvait apporter.

L'avis des enseignants sur la méthode

• Présentation du questionnaire sur la méthode

Expérimenter la méthode, c'est essentiellement l'utiliser, voir et noter les problèmes ou les questions qu'elle soulève, être à l'écoute des propositions d'aménagement ou de modification.

Les conditions d'expérimentation décrites ci-dessus nous ont permis de confier notre progression à des enseignantes expérimentées, impliquées dans ce projet qu'elles connaissaient bien et désiraient conduire à long terme. Nous avons également fait travailler des institutrices expérimentées de Maternelle et de CP auxquelles la méthode avait été brièvement exposée (3 heures) lors d'un stage de formation continue ainsi que des stagiaires en formation spécifique AIS. De jeunes professeurs d'école stagiaires ont aussi découvert et utilisé la méthode, certains en vue de la rédaction de leur mémoire professionnel. Tous ont saisi rapidement le sens de ce travail et s'y sont intéressés.

Nous avons rédigé un questionnaire afin d'évaluer comment notre méthode était perçue, au premier abord, par les enseignants après une brève présentation théorique et pratique (3 heures + 3 heures). Nous avons donc soumis ce formulaire à 2 groupes d'instituteurs en formation continue. L'un, de 23 personnes, constituait un « Stage Maternelle », l'autre de 24 institutrices désirait travailler sur la notion « Espace et temps en Cycle 2 ». Le troisième groupe sondé représentait une trentaine de jeunes professeurs d'école stagiaires en formation initiale. Nous ne rendrons pas compte ici des réponses fournies par les institutrices des « classes d'expérimentation Algor » qui sont des « inconditionnelles » du jeu de l'enfant-robot ni de celles des instituteurs stagiaires en formation spécifique AIS dont le nombre (4) n'a pas valeur statistique.

Pour plus de facilité dans l'analyse ci-dessous, nous nommerons FC1 les instituteurs du cycle 1 en Formation Continue, FC2 ceux du cycle 2 et nous utiliserons le sigle PE2 pour faire référence aux Professeurs d'Ecole stagiaires.

• Résultats du questionnaire

Malgré les craintes que certaines réflexions sur le terme « enfant-robot » avaient pu susciter, le sondage montre que seule une minorité (8 %) le considère choquant alors que 86 % le trouvent adapté. De plus, la majorité des collègues (59 %) a bien compris qu'il s'agissait d'une véritable progression ; ce chiffre monte même à 74 % pour les PE2 que cela rassure. 38 % des sondés estiment que la méthode est, en tous cas, un bon auxiliaire de progression adaptable à ses propres pratiques pédagogiques. Présenter ce travail sous forme d'une progression paraît utile (70 %), voir indispensable (29 %) à la quasi totalité de nos interlocuteurs.

La moitié (48 %) des instituteurs interrogés estime que la méthode est adaptée à la Grande Section de Maternelle et un tiers (34 %) au CP. Ils sont cependant 40 % à estimer qu'elle est adaptable en Moyenne Section. Plus de la moitié d'entre eux (53 %) se déclare prêt à tenter l'aventure de l'expérimentation en Grande Section et il sont 22 % tant en MS qu'en CP. Néanmoins, seuls 3 % d'intrépides se risqueraient en Petite Section. Le contact avec la méthode est jugé facile par plus des trois quarts des collègues (74 %).

Nous avons enfin demandé aux instituteurs de « Cocher, dans la liste suivante, uniquement les 4 (développées par cette méthode) qui vous apparaissent comme les plus **flagrantes** ». Force est de constater que nos enseignants ont encore quelques problèmes de lecture, de rigueur ou de décision puisque beaucoup d'entre eux biffent 5 voir 6 réponses (moyenne de 4,4 réponses par sondé). Nous avons donc calculé les pourcentages ci-dessous en fonction du nombre total de réponses fournies.

Les notions développées qui apparaissent comme les plus flagrantes à nos collègues sont la (20 %) puis la de l'espace (18 %) et sa (13 %). L'aide à la et à l' qui sont pour nous des objectifs importants n'émergent que très peu (respectivement 10 % et 5 %). Personne ne croit que la connaissance (fonctionnement du robot) est développée par ce travail.

Questions spécifiques

Lorsque nous désirons répondre à des questions spécifiques sur les pratiques ou les acquisitions de s des enfants, nous utilisons des grilles d'observation à l'usage des enseignants ou des questionnaires individuels à l'intention des enfants. Les exercices collectifs de déplacements ainsi que ceux relatifs aux permettent de collecter des informations sur la réussite du groupe-classe tout en constituant un bon indicateur de leur difficulté relative.

Pour que ce type d'enquête soit réalisable, il faut proposer au maître des questionnaires de synthèse, faciles à remplir, pour lesquels il aura à cocher des cases ou compter les élèves correspondant à tel ou tel critère. Les enseignants n'ont pas matériellement le temps de faire passer aux enfants des entretiens individuels. Ce type de tâches sera confié à des éléments extérieurs à la classe (stagiaires, par exemple).

En ce qui concerne l'analyse des séquences de la méthode elle-même, j'ai élaboré, pour les institutrices travaillant avec moi une fiche générale succincte d'évaluation rapide leur permettant de noter facilement les problèmes rencontrés (ou non) lors d'une séance donnée.

Tests d'acquisition

Ces tests ont pour but de mesurer si les enfants utilisant notre méthode ont (ou non) acquis (mieux ou plus vite) certaines par rapport à des élèves ne jouant pas à « l'enfant-robot ». Ce type de travail est toujours délicat car il fait appel à une population témoin (autre classe) dont on n'est jamais totalement sûr qu'elle est réellement comparable au groupe de travail qu'on étudie.

La méthode que nous utilisons n'est pas la seule à développer certains points du programme de l'école maternelle sur lesquels nous espérons que tout enseignant s'attarde d'une manière ou d'une autre. Les comparaisons deviennent alors bien hasardeuses car les mêmes notions peuvent être acquises de manières diverses, avec ou sans notre méthode.

Le jeu de l'enfant-robot

Ce qu'en disent les enfants

- **Présentation générale**

Les informations évoquées sont le résultat d'un travail d'interview intéressant réalisé sur 25 enfants par mes élèves de PE2 (Professeurs d'école stagiaires travaillant pour leur mémoire de 2^{ième} année sur le jeu de l'enfant-robot).

- **Synthèse**

La plupart des enfants (88 %) déclarent aimer le jeu de l'enfant-robot. (76 %) trouvent que c'est une activité « facile ». Un tiers des élèves (36 %) ne sait pas cependant formuler pourquoi il trouve ce travail aisé ou difficile. Ceux qui fournissent une explication trouvent ce jeu aisé parce « qu'on se déplace (36 %) » et « qu'on se conforme aux règles (20 %) ».

Lorsqu'ils jouent à l'enfant-robot, la moitié (50 %) des élèves déclare ne penser à rien de particulier. Un tiers (32 %) tente de « faire comme disent les cartes ». Seulement 15 % d'entre eux imaginent qu'ils jouent au robot alors que 4 % ont comme préoccupation essentielle de « faire attention aux copains quand on recule ».

Très majoritairement (92 %), les enfants savent que l'enfant-robot ne peut pas faire autre chose que ce qui est indiqué sur la carte. En effet, 61 % affirment qu'il ne fait que ce qu'ordonnent les cartes (son programme), puisque c'est une machine (17 %) et que « ce n'est pas permis » (9 %).

Un quart des enfants (25 %) ne sait pas dire quelle différence existe entre lui/enfant-robot et lui/enfant. Un autre quart avoue avoir conscience d'une distinction qu'il ne sait expliciter. 20 % expriment comme différence essentielle que le robot est dirigé par l'homme, 19 % que le robot est en fer et a des boutons... Enfin, 13 % se réfugient derrière le fait que « la maîtresse dit que c'est différent ».

Même si tous les enfants ne s'identifient pas à un robot dès qu'ils jouent, le jeu leur paraît facile et ils l'aiment. Ils ont, pour la plupart, compris que le robot est différent d'eux et ne peut pas fonctionner sans carte. Ils sont contents de voir le pouvoir que l'on peut avoir sur ce robot qui les fascine.

Exemple d'expérience ayant permis de peaufiner la méthode

Le trait : une représentation difficile

- **Présentation générale**
Référence de l'illustration : 5/MS-GS/94-95/1
Niveau : MS/GS
Classe de : Evelyne/Marylin
Date : Mai 1995

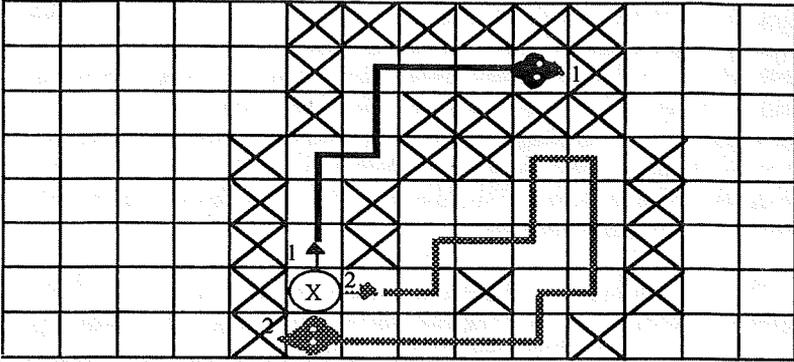
Objectif général de la séquence : Savoir passer d'une représentation de type « trait » au programme correspondant et inversement.

Conditions d'expérimentation : Ces expériences ont été menées en Mai 1995, dans 2 classes de Moyenne Section et de Grande Section connaissant Algor, par 3 petits groupes d'institutrices de Maternelle en stage de formation continue sur le thème « Espace et temps en cycle 2 ». Ces enseignantes n'avaient bénéficié que d'une rapide présentation de la méthode et ont mal évalué les difficultés des exercices qu'elles proposaient. Comme celles-ci sont du même ordre, nous les avons réunies dans ce paragraphe.

- **Synthèse**

- *1^{er} exercice*

Dans le premier groupe, les élèves doivent tracer 2 chemins sur un même quadrillage. La consigne, très discutable, est de tracer le « le plus long possible » allant de la base à chaque Algor. En effet, nous avons défini comme parcours « le plus court », celui correspondant au parcours utilisant le moins d'instructions. On voit bien qu'en ajoutant, par exemple, des tours complets (4 pivotements identiques), on peut agrandir, à loisir, le programme. Il aurait été plus judicieux de demander de tracer un qui passe dans le plus grand nombre possible de cases.



Une fois le chemin tracé, l'élève doit choisir, et étaler dans l', les cartes ayant permis à notre personnage d'atteindre son but. Il est à noter que l'enfant ne dispose d'aucun matériel auxiliaire (vignette Algor, par exemple) pour réaliser cet exercice.

Cet exercice pose un problème fondamental : une telle trace ne permet pas de déterminer, à coup sûr, quel programme l'a engendré. En effet, le trait dessiné n'indique pas si Algor avançait ou reculait, il ne précise pas non plus d'éventuels pivotements contradictoires ou autre demi-tour. Pour retrouver les instructions à partir d'une telle représentation, il convient donc de se restreindre au cas où notre petit personnage ne fait qu'avancer et n'opère pas de virages intempestifs.

Outre le problème précédemment évoqué, nos collègues ont connu quelques soucis d'ordre . La difficulté principale de cette activité ne réside pas dans la consigne elle-même (codage de parcours) qui ne présente habituellement pas de problème particulier mais dans ses conditions d'exécution. En effet, les élèves trouvent facilement les premières du programme mais n'arrivent plus, ensuite, à se rappeler quelle position d'Algor a déterminé leur choix précédent. Ils sont, en général, contraints de reprendre le programme depuis le départ afin de pouvoir le compléter. Ceux qui ne prennent pas cette précaution peuvent oublier une ou plusieurs positions (et cartes) intermédiaires.

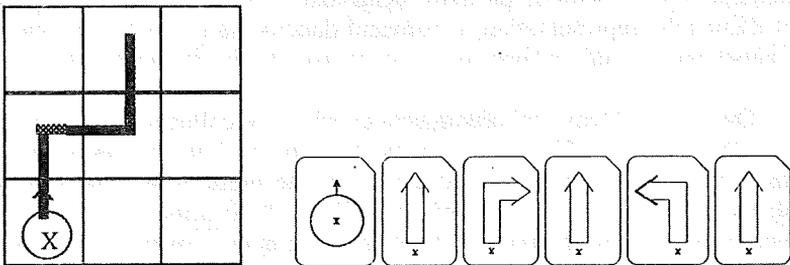
Cet exemple nous confirme, à nouveau, que les problèmes de représentation de parcours ne sont pas triviaux. L'enfant n'a pas osé « barrer » ou colorier différemment le au fur et à mesure qu'il progressait. Il aurait eu ainsi le sentiment de « détruire » le travail demandé lors de la première consigne (traçage du chemin). Une autre possibilité consistait dans l'utilisation d'un Algor mobile (non fourni ici) qui aurait sagement « attendu », sur le quadrillage, que l'élève ait choisi la carte-instruction correspondant à son dernier mouvement.

•2^{ème} exercice

Le second groupe a en charge de « dessiner avec un trait », sur un, le chemin emprunté par un Algor obéissant à une série d'instructions. L'équipe d'enseignantes proposant cette activité a tout d'abord accumulé les obstacles

pédagogiques à la réussite de cet exercice. En effet, le programme proposé est bien trop long puisque constitué d'une trentaine (!) de cartes. La bande des instructions est collective et affichée verticalement tandis que les élèves dessinent sur un quadrillage horizontal. Ces derniers ne peuvent pas voir la totalité du programme sans se déplacer. Bien entendu, la difficulté n'est pas la réalisation du parcours lui-même mais est liée au fait que lorsque l'enfant relève la tête, il ne se rappelle plus quelle est la prochaine carte à . Il n'a pas, en effet, la possibilité de barrer, au fur et à mesure les flèches déjà utilisées.

De plus, un obstacle didactique important réside dans la transcription des pivotements. Alors que les élèves interrogés connaissent parfaitement les mouvements de l'enfant-robot et tournent physiquement sur place sans changer de case, ils ne savent pas comment transcrire ce type de déplacement avec leur crayon et ont parfois tendance à tracer le pivotement puis avancer d'une case. Lorsqu'ils travaillent avec un Algor mobile (non fourni ici), cet embarras n'apparaît pas ; ils font pivoter le petit personnage sur sa propre case. Certains enfants réclament effectivement comme aide un « petit carré Algor ». On pourra également proposer aux enfants la technique consistant à amorcer le début du trait permettant de mémoriser un changement de direction.



•3ième exercice

Le troisième groupe doit représenter, également par un trait sur un quadrillage papier, le parcours qu'un de leur camarade enfant-robot est en train d'exécuter en dépilant son . On retrouve, dans cet exercice, les mêmes difficultés pédagogiques et didactiques que celles rencontrées dans les deux activités précédentes, à savoir un parcours trop long dont on « perd le fil » durant la transcription et une gêne réelle pour représenter les pivotements avec son crayon. Du reste, certains élèves ne suivent pas la consigne proposée et dessinent des petits personnages Algor sur leur feuille au lieu du trait demandé.

•Conclusion

Le travail de codage ou de décodage de parcours utilisant une représentation du type « trait sur un quadrillage » pose donc un problème important concernant le pivotement et l' du mobile au moment de l'exécution. Pour aider les enfants, il convient de leur fournir un Algor mobile ainsi que la possibilité de barrer ou de marquer d'une manière quelconque, au fur et à mesure de leur réalisation, les étapes

déjà effectuées. Il est également essentiel que les élèves aient eu la possibilité de vivre préalablement le parcours avec leur corps.

Résumé

Les expériences menées ci-dessus ont permis de peaufiner la méthode, de la modifier et de l'améliorer. Elles ont également amené les enfants à se familiariser avec ce travail et à se confronter à de véritables situations de consistant à décomposer un problème en une succession de tâches plus simples, à et à décoder des instructions... Ils se sont également heurtés aux difficultés de la représentation et ont découvert les intérêts ainsi que les limites de la notion de « trace ».

Exemples d'expérience concernant des objectifs didactiques associés

Connaissance du robot (tests de fin d'année 94/95)

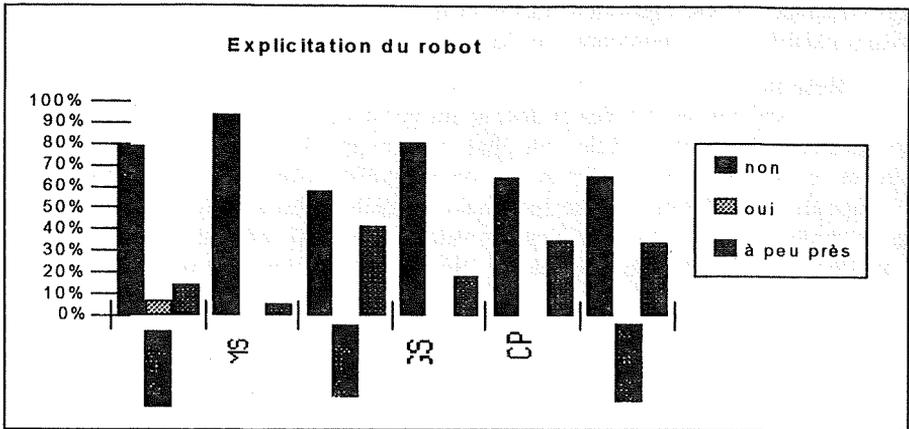
- **Présentation générale**

Ces tests de fin d'année 94/95 ont eu lieu les Jeudi 8 et Vendredi 9 Juin 1995. Ils ont été soumis à 170 élèves (60 de Grande Section, 60 de Moyenne section et 50 de CP) de Versailles et du Chesnay dont certains connaissaient le jeu de l'enfant-robot et d'autres pas. Une vingtaine de Professeurs d'école stagiaires de l'IUFM de Versailles ont fait passer les entretiens individuels.

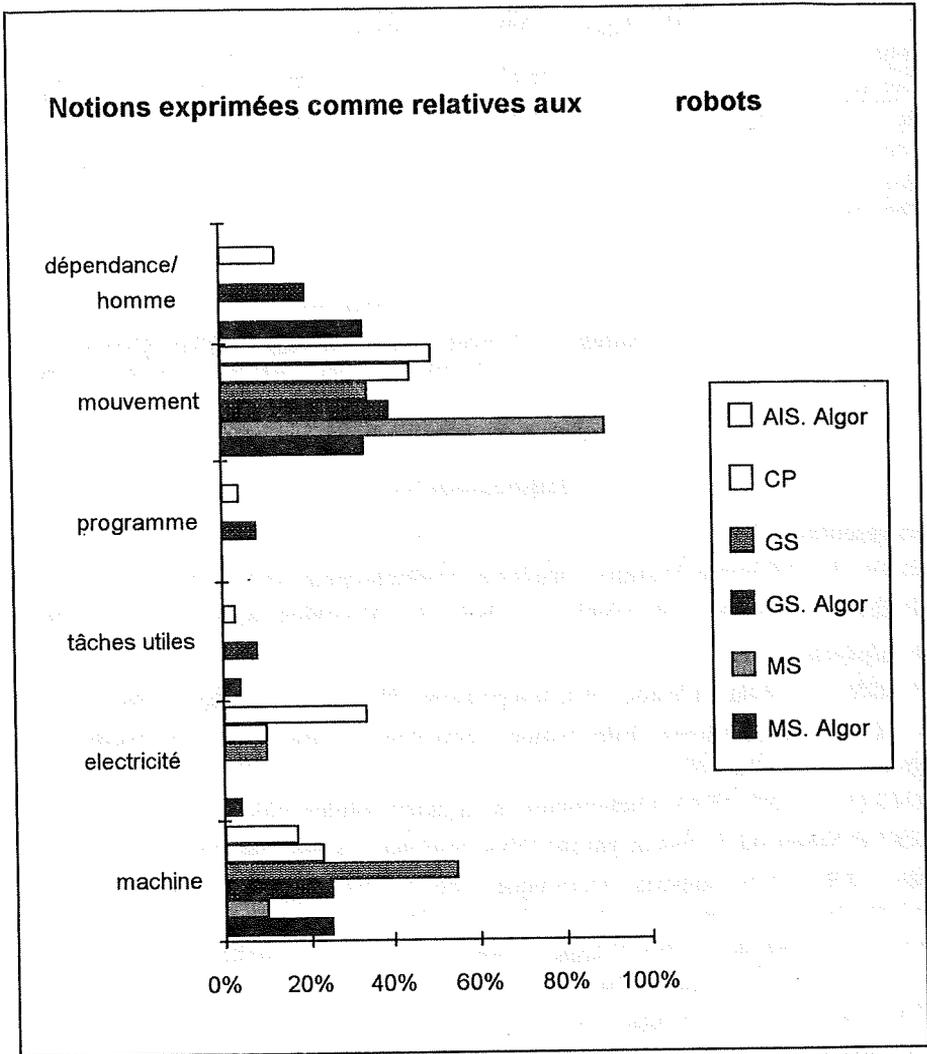
Nous nous étions entendu, dès le début de l'année, avec les institutrices expérimentant le jeu de l'enfant-robot dans leur classe sur la définition suivante : « Le robot est une machine, fonctionnant avec de l'électricité (secteur ou pile) à qui l'homme fournit un programme, éventuellement modifiable. Ce programme met le robot en mouvement et lui fait exécuter des tâches utiles, souvent répétitives. Nous pourrions introduire la notion de « robot de la maison », répondant à cette définition (lave-linge, lave-vaisselle, magnétoscope programmable...) en excluant les machines fonctionnant uniquement sur un mode marche/arrêt (télévision, fer à repasser...) ». Nous avons réuni sur un « imagier » des photos de robots de la maison, de robots-s et d'intrus (photocopies couleur pour plus de lisibilité). L'enfant devait y discriminer les différentes catégories de robots. Il avait également à justifier ses choix et à faire état de ses s sur la notion de robot.

Synthèse

La quasi totalité des enfants ne sait pas spontanément exprimer précisément ce qu'est un robot. Ils sont en moyenne 10 % à en donner une définition approximative en MS, 24 % en GS et 35 % en CP. Les enfants connaissant Algor obtiennent cependant des résultats bien supérieurs à ceux de leurs camarades (cf. tableau ci-dessous).



Dans les réponses naturelles fournies par les enfants, nous avons tenté d'analyser les rapports relatifs des qualités attribuées aux robots. C'est la notion de « machine en mouvement » qui revient le plus souvent. Il est également frappant de constater que les réponses des élèves connaissant Algor sont plus variées et réparties (cf. tableau ci-dessous) alors que, pour les autres, certains items ne sont jamais cités. La notion de « répétition » n'est, quant à elle, jamais exprimée.



Nous avons ensuite posé des questions précises aux enfants concernant les robots. Les résultats du groupe Algor sont bien meilleurs que ceux de leurs homologues. Environ 80 % savent que le robot fonctionne avec de l'électricité, ne fait pas ce qu'il veut et est programmé par l'homme (seulement 66 % pour ce dernier item en MS-Algor). Le prétexte du jeu de l'enfant-robot permet donc aux élèves d'avoir une bonne idée de la réalité du robot (cf. tableau ci-dessous).

Notion relative au robot	MS Algor	MS	GS Algor	GS	CP	AIS Algor
électricité	84%	39%	79%	61%	71%	100%
ne fait pas ce qu'il veut	76%	58%	83%	70%	71%	86%
programmé par l'homme	66%	37%	83%	65%	69%	100%

Eric GREFF

Adresse : 14, avenue Henri Régnauld 92310 Sèvres
Tél. : (1) 45.07.04.45 (Domicile) Fax : (1) 39.24.20.99 (IUFM)

Bibliographie

Représentations

ABRIC Jean-Claude, Pratique sociales et représentations, PUF, 1994

LIBEN L.S., Spatial representation and behavior, *New-York, Academic Press, 1981*

Philo/psycho

ALEGRIA J. et AL., L'espace et le temps aujourd'hui, *Point Sciences, 1983*

DUFOYER Jean-Pierre, Informatique, éducation et psychologie de l'enfant, *Le Psychologue, PUF, 1988*

FOULQUIE., ST JEAN, Dictionnaire de la langue philosophique, *PUF, 1969*

FREUD Sigmund, Essais de psychanalyse appliquée, *Idées, Gallimard, 1971*

HOC J.M., Les apports réciproques de la psychologie cognitive et de l'informatique, *Le journal des psychologues, 1985*

HOC Jean-Michel, Psychologie cognitive de la planification, *Presses Universitaires de Grenoble, 1987*

KLEIN Mélanie, Essais de psychanalyse, *Payot, 1967*

LINARD Monique, Des machines et des hommes, *Editions Universitaires, 1990*

MENDELSON P., L'analyse psychologique des activités de programmation chez l'enfant, *Enfance, 1985*

MONTESSORI Maria, L'enfant, *Desclée de Brouwer, 1936*

PIAGET Jean, Conférence de Genève, *Edition de la Baconnière, Neuchâtel, 1962*

PIAGET Jean, Construction du réel chez l'enfant, *Delachaux et Nieslé, 1977*

PIAGET Jean, La formation du symbole chez l'enfant, *Actualités pédagogiques et psychologiques, Delachaux et Nieslé, 1972*

PIAGET Jean, La géométrie spontanée chez l'enfant, *PUF, 1973*

PIAGET Jean, La prise de conscience, *Presses Universitaires de France, 1974*

PIAGET Jean, Six études de psychologie, *Médiations*, Gonthier, 1969

PIERON H., Vocabulaire de la psychologie, PUF, 1979

WALLON Henri, L'évolution psychologique de l'enfant, *Armand Colin*, 1991

WEIL-BARAIS Annick, L'homme cognitif, *Collection Premier Cycle*, PUF, 1993

Algorithmique. Programmation

ARSAC Jacques, Premières leçons de programmation, *Cedic*, Nathan, 1984

ARSAC Jacques, Préceptes pour programmer, *Dunod*, 1991

BIONDI J., CLAVEL G., Introduction à la programmation, *Masson*, 1981

DELANOY Paul, Les « Mathémotocrates » ont-ils tué le langage LOGO ? Les

« Technocrates le sauveront-ils ? », *Revue de l'EPI n°66 Juin 1992*, *EPI*, 1992

DELANOY Paul, Peut-on enseigner un langage que personne ne parle ?, *Actes du 4^{ième} Colloque Francophone de Didactique de l'Informatique*, Québec, 1994

DEMOUVEAUX J.P., *Revue de l'EPI n°6 Septembre 1984*, *EPI*, 1984

DESCARTES, Discours de la méthode, *Nathan*, 1994

DUCHATEAU Charles, Images pour programmer : un environnement pour l'apprentissage de l'algorithmique, *Actes du 2^{ième} Colloque Francophone de Didactique de l'Informatique*, Namur, 1990

GREFF Eric, Comment introduire la pensée algorithmique auprès de jeunes enfants à travers le jeu de l'enfant-robot, *La recherche et l'IUFM de l'Académie de Versailles*, 1995

HOC Jean-Michel, Etude de la formation à une méthode de programmation informatique, *Le travail humain*, 1978

HOC Jean-Michel, L'apprentissage de l'utilisation des dispositifs informatiques par analogie à des situations familières, *Psychologie française*, tome 32-4, 1987, page 217 à 226

HOC Jean-Michel, Prise de conscience et planification, *Psychologie française*, tome 32-4, 1987, page 247 à 252

LAGRANGE Jean-Baptiste, *Bulletin de l'EPI n°67*, *EPI*, 1992.

LEDGARD H.F., Proverbes de programmation, *Dunod Informatique*, 1983

MARCHAND Daniel, La robotique pédagogique ! ça existe ?, *Revue de l'EPI n°65*, Mars 1992, *EPI*, 1992

MEYER Bertrand, BAUDOIN Claude, Méthodes de programmation, *Eyrolles*, 1984

PAIR Claude, L'apprentissage de la programmation, *Actes du 1^{ier} Colloque Francophone de Didactique de l'Informatique*, Paris, 1988

PAPERT Seymour, Jaillissement de l'esprit, *Flammarion*, 1981

PEYRIN Jean-Pierre, Un jeu de rôle pour l'enseignement de la programmation, *Actes du 1^{er} Colloque Francophone de Didactique de l'Informatique*, Paris, 1988

PEYRIN Jean-Pierre, GUERAUD V., Le laboratoire Arcade : une assistance à l'enseignement de l'algorithmique, *Actes du 2^{ième} Colloque Francophone de Didactique de l'Informatique*, Namur, 1990

PEYRIN Jean-Pierre, Une variété d'expressions des algorithmes pour mieux apprendre à raisonner, *Actes du 3^{ième} Colloque Francophone de Didactique de l'Informatique*, Sion, 1992

PEYRIN Jean-Pierre, Enseigner la programmation : quoi ?, pourquoi ?, comment ?, *Actes du 4^{ième} Colloque Francophone de Didactique de l'Informatique*, Québec, 1994

RICHARD Chantal et Patrice, Initiation à l'algorithmique, *DIA, Belin*, 1983

RICHARD Chantal et Patrice, Programmation, Initiation à la programmation méthodique, *DIA, Belin*, 1984

ROGALSKY Janine, Méthode de programmation, *Actes du 1^{ier} Colloque Francophone de Didactique de l'Informatique*, Paris, 1988

WIRTH N., Introduction à la programmation systématique, traducteur *Lecarme O.*, Masson, 1977

Expériences informatiques à la maternelle

BASTIDE Pierre, LE TOUZE Jean-Claude, Prototype d'un dispositif autonome programmable par de jeunes enfants, *Revue Française de pédagogie n° 56, INRP*, 1981

BEAU DE MOULIN S., Tortue de sol et apprentissage de symboles en grande section de maternelle, *Colloque "l'enfant et l'ordinateur"*. Rouen, 1985

BOSSUET Gérard, L'accord LOGO. Vol. 2, *Université Paris VI*, 1986

BOSSUET Gérard, L'accord LOGO. Vol. 3, *Université Paris VI*, 1987

BOSSUET Gérard, L'ordinateur à l'école. / *L'éducateur*, PUF, 1983

BOSSUET Gérard, Sécante. Conséquence 1, *Université Paris VI*, 1987

BOSSUET Gérard, Sécante. Conséquence 2, *Université Paris VI*, 1988

BOULE François, L'informatique, l'enfant, l'école, *Armand Colin-Bourrelier*, 1988

CALMY-GUYOT Gisèle, Informaticiens en herbe, *Ecole La Fontaine, Meudon*, 1985

COMBES-TRITHARD Françoise, Enregistrer, lire, programmer à l'école maternelle, *Armand Colin-Bourrelier*, 1984

GREFF Eric, Une année de logique et algorithmes avec les 5/6 ans, *Nathan Education*, 1995

HENAFF Françoise, BASTIDE Anne, Informaticiens en herbe, *Ecole Maternelle Jean de la Fontaine, Meudon*, 1985

LE GUYADER-BOSSUET Christiane, La pédagogie LOGO en maternelle : camions et spatio-globe, *Université Paris VI*, 1985

LE TIRILLY Marc, Quelques visées éducatives : l'enfant programmeur, *CNDP, CRDP de Marseille, 1986*

PILLOT Jacqueline et Christian, L'ordinateur à l'école maternelle, *Armand Colin-Bourrelier, 1984*

REGINNI Henry, LOGO, des ailes pour l'esprit, *Cedic, Nathan, 1983*

Apprentissages logiques chez les jeunes enfants

BOULE François, Manipuler, organiser, représenter : prélude aux mathématiques, *Armand Colin-Bourrelier, 1989*

BOULE François, Mathématique et jeux, *CEDIC, Nathan, 1976*

BOURNEAU G., DANIAU J., MIAILLE M., Apprentissages mathématiques en grande section d'école maternelle, *L'Ecole, 1992*

CHAMPDAVOINE Lucette, Les mathématiques par les jeux. Grande section et C.P., *Vivre à la Maternelle, Nathan, 1986*

CHAMPDAVOINE Lucette, Les mathématiques par les jeux. Petite et moyenne section, *Vivre à la Maternelle, Nathan, 1985*

ERMEL, Apprentissages mathématiques à l'école élémentaire, *Sermap, Hatier, 1993*

IFRAH Georges, Les chiffres ou l'histoire d'une grande invention, *Robert Laffont, 1985*

Le signe

BAUDRILLARD Jean, Le système des objets, *Tel, Gallimard, 1988*

ECO Umberto, Le signe, *Média, Labor, 1990*

EVEART-DESMEDT Nicole, Le processus interprétatif, Introduction à la sémiotique de Ch.S. Peirce, *Philosophie et Langage, Liège, Pierre Mardaga, 1990*

FRUTIGER Adrian, Des signes et des hommes, *Delta et Spes, 1983*

MONNERAT Claude, LEFRANC Robert, PERRIAULT Jaques, L'enfant et l'image - 1879-1979, *Mémoires et documents scolaires, CNDP, 1979*

PRIETO L., Messages et signaux, *PUF, 1972*

PRIETO L., Pertinence et pratique, *Edition de Minuit, 1975*

SERRE-FLOERSHEIM Dominique, Quand les images vous prennent au mot, *Les éditions d'organisation, 1993*

Le jeu

CAILLOIS Roger, L'homme et le sacré, *Idées, Gallimard, 1972*

CAILLOIS Roger, Les jeux et les hommes, *Gallimard, 1958*

CHATEAU Jean, L'enfant et le jeu, *Scarabée, 1954*

CHATEAU Jean, Le jeu de l'enfant après 3 ans, sa nature, sa discipline, *Vrin, 1961*

CHAUVEL Denise, MICHEL Viviane, A la maternelle : des jeux avec des règles, *Retz, 1984*

- GAMOW G.**, M. Tompkins au pays des merveilles, *Dunod*, 1964
- GUTMACHER Francis**, Jeux d'un bord à l'autre, *APMEP n° 44*, 1976
- GUTTON Philippe**, Le jeu chez l'enfant, *Sciences humaines et sociales, Larousse*, 1972
- HENRIOT Jacques**, Le jeu, *Le philosophe, PUF*, 1976
- HUIZINGA J.**, Homo ludens, *Gallimard*, 1951
- LEE**, Play in education, *New-york, Macmillan*, 1916

Didactique

- CHEVALLARD Yves**, La transposition didactique : du savoir savant au savoir enseigné, *La pensée sauvage*, 1985
- PERES Jacques**, Recherches en didactique sur l'utilisation de la tortue de sol. Compte rendu d'une préexpérimentation, *Université de Bordeaux*, 1985
- PERES Jacques**, Recherches menées à l'IREM de Bordeaux sur l'utilisation de la tortue de sol LOGO à l'école Maternelle, *Université de Bordeaux*, 1987
- ROGALSKI Janine**, Acquisition et didactique des structures conditionnelles en programmation informatique, *Psychologie française, tome 32-4*, 1987, page 275 à 300
- ROGALSKI Janine, SAMURCAY René**, Approches cognitives de l'apprentissage de la programmation, *Actes du 2^{ième} Colloque Francophone de Didactique de l'Informatique, Namur*, 1990
- ROGALSKI Janine, VERGNAUD Gérard**, Didactique de l'informatique et acquisitions cognitives en programmation, *Psychologie française, tome 32-4*, 1987, page 267 à 273

L'enfant et l'espace

- DUTILLEUL M.B., GILABERT H., DU SAUSSOIS N.**, Les enfants de 4 à 6 ans à l'école maternelle, *Armand Colin*, 1990
- LAURENT-DELCHET Marguerite et al.**, Intuitions et construction de l'espace, *INRDP*, 1976
- LE ROCH Yves**, Vivre l'espace, *ISTRA*, 1975
- LURCAT Liliane**, L'enfant et l'espace : le rôle du corps, *PUF*, 1976
- LURAT François**, L'espace en Mathématiques et en Psychologie. Remarques d'un Physicien, *Cahiers de Psychologie, 19*, 1976
- LURAT Liliane**, Espace vécu et espace connu à l'école maternelle, *Sciences de l'éducation, ESF*, 1982
- PECHEUX Marie-Germaine**, Le développement des rapports de l'enfant à l'espace, *Psychologie, Nathan Université*, 1990.
- PIAGET Jean, INHELDER B.**, La représentation de l'espace chez l'enfant, *PUF*, 1948
- SAUVY Jean et Simone**, L'enfant à la découverte de l'espace, *E3, Casterman*, 1972

Thème1 : Informatique au Primaire

Utilisation raisonnée des instruments micro-informatiques dans les disciplines de l'enseignement secondaire

Jean François LEVY

présentée le mercredi 10 avril 1996 à 15h45

Résumé: *On s'intéresse aux problèmes posés par l'introduction et la mise en oeuvre des instruments micro-informatiques dans les disciplines de l'enseignement secondaire, pour les élèves, les enseignants et les formateurs (formation initiale et continue). On tente d'établir un bilan des difficultés sur les plans cognitif et didactique, de définir un ensemble minimal de concepts d'ordre informatique indispensables aux utilisateurs et d'esquisser des éléments de didactique des acquisitions de connaissances et de savoir-faire dans le domaine.*

Summary: *This paper deals with problems concerning the insertion and implementation of micro-computers in various subjects taught in French secondary schools, for the use of the students, teachers and trainers (initial and continuous training). We try to put forward an assessment of the main difficulties encountered - at cognitive and didactic levels, then we define a minimal set of the concepts relevant to informatics that are absolutely necessary to enable the users to work in good conditions. In the last part, we outline some basic didactic principles to complete knowledge and know-how in the field.*

Une recherche (conduite à l'INRP, Paris) s'était donnée plusieurs buts autour des difficultés rencontrées par les acteurs de l'enseignement secondaire (formateurs, enseignants et élèves) dans l'introduction et la mise en oeuvre des dispositifs micro-informatiques en tant qu'instruments d'aide aux disciplines :

- Bilan des problèmes de formation et d'utilisation sur les plans cognitif et didactique ;
- Tentative de définition d'un ensemble minimal de concepts d'ordre informatique indispensables aux utilisateurs ;
- Esquisse d'une didactique des acquisitions de connaissances et savoir-faire dans le domaine.

Ces objectifs pourraient contribuer à la réflexion sur l'intégration de l'informatique dans l'enseignement, qui semble être un souci dominant actuellement, créé en partie par l'offre de dispositifs de plus en plus sophistiqués pour la formation (initiale et continue). Les enseignants sont interpellés en permanence, et les multiples situations dans lesquelles ils se retrouvent face à l'informatique leur pose (et à leurs élèves) de nombreux problèmes.

Les décalages importants entre un discours social quasi-mythique à propos de l'informatique et les réalités accessibles, entre des potentialités réelles ou supposées de ces aides et les difficultés de leur mise en oeuvre ne clarifient pas les choses.

Une précédente étude [LEVY 93] avait permis de caractériser, sur les plans technique, cognitif et didactique, un certain nombre d'obstacles qui pouvaient conditionner les échecs dans les acquisitions des élèves, dans le cadre des situations d'apprentissages techniques et professionnels tertiaires initiaux. Nous relevions des difficultés de construction des concepts spécifiques à cet univers et des difficultés d'ordre didactique, dues à la contradiction entre une approche dans laquelle la théorie dominait et une approche plus expérimentale mais dont la spécificité du domaine (informatique) ne permettait pas à l'apprenant de construire les concepts par la pratique.

Nous nous posons maintenant la question de l'extension de ces conclusions à l'apprentissage et l'enseignement de l'usage de l'informatique dans les disciplines techniques et générales : existe-t-il des invariants transdisciplinaires qui permettraient à la fois de caractériser les difficultés inhérentes aux divers aspects de dans l'enseignement et d'y remédier ?

Les cadres théoriques dans lesquels nous avons ancré la recherche appartiennent aux théories sur l'évolution des objets techniques (épistémologie de la technique), sur les interactions entre homme et instruments, à la psychologie cognitive et à la didactique.

La méthodologie de l'étude a été fondée sur l'observation et l'expérimentation en situations réelles d'enseignement et de formation. Ces observations et expérimentations ont été effectuées avec la participation active de douze professeurs de lettres, philosophie, histoire, biologie, physique, technologie (collège), mathématiques et économie-gestion (lycée professionnel). La diversité des équipes ainsi constituées a garanti l'aspect pluridisciplinaire de la recherche, condition de sa validité. Cette démarche a assuré également une prise de conscience des phénomènes, des conditions d'émergence des erreurs et des interactions entre formateurs, formés, matériels, tâches et situations.

Les phases d'observation et de réflexion aboutissant à des propositions d'expérimentation sur le terrain et à des analyses en retour se sont succédées tout au long de l'étude. C'est ainsi que s'est élaboré un consensus sur les notions fondamentales à faire acquérir et sur les problèmes posés par un abord jugé par tous trop orienté vers les savoir-faire, au détriment des connaissances plus abstraites.

Principaux résultats

1 - Des difficultés disparaissent, d'autres émergent

Les observations ont confirmé l'ensemble des difficultés constatées de manière informelle depuis longtemps dans les usages de l'informatique :

- disparition de certaines lourdeurs de mise en oeuvre des matériels et des logiciels avec l'évolution technologique (heureux abandon du langage de Dos), mais survenue de nouveaux problèmes liés notamment aux outils d'interface récents: fenêtres, icônes, souris ;
- augmentation du degré d'abstraction dû aux systèmes de représentations graphiques, et dont le raisonnement par analogie ne permet pas toujours une bonne acquisition ;
- difficultés de l'utilisateur pour se situer à un niveau précis de fonctionnement (système d'exploitation, application) et pour spécifier les objets en jeu (données, programmes), difficultés augmentées par l'apparente facilité de manipulation directe des objets par les éléments graphiques (icônes).

Ainsi la richesse des outils entraîne leur complexité d'utilisation. Cette situation devient vite problématique, et se servir efficacement de nouvelles possibilités nécessiterait un abord plus fondamental, passant par des conceptualisations.

2 - Des invariants transdisciplinaires

L'observation des utilisations de l'ordinateur dans plusieurs disciplines a permis de repérer des éléments récurrents caractérisant les difficultés d'acquisitions (informatiques) au travers d'usages différents, compte tenu de la variété des situations et des conditions de leur mise en oeuvre. Ces éléments ne sont pas seulement caractéristiques des objets (abstrait) mais également des relations entre eux et des comportements des apprenants.

Des notions incontournables sont vite apparues : les fichiers, la mémoire, la différence données/programmes, etc. Mais les notions de base d'un système informatique se présentent plutôt comme un ensemble indissociable d'objets abstraits en interaction les uns des autres et inséparables - sur le plan de la conceptualisation - des opérations qui agissent sur eux. Nous aboutissons ainsi à la notion plus large de réseau conceptuel.

Les fichiers sont des objets informatiques fondamentaux dont la manipulation est difficile à acquérir par tous les utilisateurs. En effet, ce mot **fichier** est un terme dont le sens recouvre des objets différents selon le contexte, informatique ou hors informatique. En conséquence, les tentatives (nombreuses) d'enseignement de ces notions par analogie directe avec les sont susceptibles de semer une confusion peu propice à stabiliser des connaissances réellement opérationnelles sur les systèmes informatisés. Une première approche utilise trois niveaux d'analyse : la nature des objets (programmes /données et par extension système /application /document) et leur désignation, les opérations sur ces objets, et leurs lieux de mémorisation. Les observations de toutes les équipes ont montré des difficultés et des erreurs à ces trois niveaux.

La représentation par icônes des objets dossiers et documents présente une richesse qui dépasse largement les possibilités d'observation des débutants¹. Les opérations de réduction ou d'agrandissement ne sont pas davantage interprétées par les apprenants de manière satisfaisante, dans la mesure où ils ne peuvent pas établir un lien de sens entre les transformations purement graphiques et leurs conséquences sur le plan de la gestion des objets.

Les termes relatifs aux actions et supports de lecture /écriture, qui paraissent simples (simplement traduits de l'anglais...) sont facilement victimes des analogies trompeuses avec le support papier : a tendance à renvoyer à un déplacement de l'objet plutôt qu'à une recopie :

Erreur! Signet non défini.avons-nous déjà entendu lors de nos observations sur le traitement de texte. De la même manière, l'écriture ne fait pas prendre immédiatement conscience que le contenu précédent sera effacé. Il est vrai que les termes OUVRIER et ENREGISTRER sont plus proches de la réalité. Faut-il alors bannir du vocabulaire " lire " et " écrire " (en mémoire, sur disque ?).

Toutes ces questions renvoient à une problématique plus large, qu'on peut exprimer par la question suivante :

Erreur! Signet non défini.

(un enseignant de l'équipe au début de la recherche)

La notion de mémoire (sous toutes ses formes présentes dans le dispositif informatique) est centrale (ce qui n'est pas une originalité en soi) mais il a été observé que les apprenants peuvent mettre en oeuvre quasi-inconsciemment des démarches d'évitement de cette conceptualisation - pourtant indispensable - par des usages qui, s'ils n'empêchent pas certains résultats à un niveau de fonctionnement élémentaire, gênent considérablement l'utilisation de fonctionnalités plus complexes, telles que celles proposées maintenant de manière courante par les environnements évolués (changement d'applications et multi-fenêtrage sous WINDOWS...).

Si une notion très générale de mémoire peut être facilement construite par analogie avec la mémoire humaine, la distinction entre mémoire de stockage et mémoire de travail est loin d'être évidente. En effet, les mémoires de masse ont une réalité physique (les disquettes font maintenant partie de l'image socialisée des ordinateurs), tandis qu'il est difficile d'élaborer des images mentales substitués d'objets non donnés à la perception (la mémoire vive) et dont seule la fonction peut être observée, qui plus est indirectement. Ces évidences ont été confirmées par les observations.

¹ Les divers travaux d'observation de situations d'apprentissage ont permis de faire l'hypothèse qu'un sujet ne peut saisir par l'observation que des éléments auxquels il s'attend . Ainsi les petits détails des icônes qui spécifient les caractéristiques des fichiers échappent presque toujours aux débutants.

La conceptualisation des mémoires détachée de la localisation traditionnelle serait alors susceptible de renouveler les possibilités d'action des formateurs : envisager les transferts de fichiers en mettant l'accent sur des créations d'objets et des états de ces objets permettrait peut-être de faire bénéficier davantage aux apprenants des facilités offertes par les gestionnaires modernes basés sur les icônes et les représentations arborescentes.

De même, une description de l'écran comme simple organe de visualisation ne suffit pas à faire comprendre son rôle de dans l'ensemble d'un document mémorisé. La manipulation des fenêtres dans WINDOWS fait mettre en oeuvre implicitement un raisonnement par analogie assimilant ces objets à des feuilles de papier superposées. Or cette analogie, évidemment intentionnelle de la part des concepteurs du logiciel, a des limites pour l'utilisateur, pour qui la notion d'activation de fenêtre n'a pas vraiment son équivalent dans le domaine source.

La hiérarchie dossier/document, principe si simple et si utilisé dans le domaine ne fonctionne pas toujours aussi bien dans une structure informatisée [LEVY 93], car elle fait appel à l'abstraction de l'arborescence.

3 - Des comportements contradictoires des utilisateurs

D'autres éléments concernent le comportement général des utilisateurs. Les observations recueillies dans ce registre montrent que les apprenants peuvent avoir des conduites et des raisonnements paradoxaux, voire contradictoires avec ceux couramment attendus ; les sujets prennent également en compte, d'une manière souvent conflictuelle, la **logique de l'utilisateur** qui apparaît comme dominante et relativement claire, et la **logique de fonctionnement** [RICHARD 83] du dispositif, qui se présente comme une contrainte restrictive et plutôt mystérieuse. Ces facteurs pourraient être pris en considération tout au long de leur apprentissage, ce qui n'a pas toujours été le cas jusqu'à présent.

Il a été relevé des tendances à des apprentissages de types comportementaux, dus par exemple à l'usage d'interfaces qui s'uniformisent quel que soit le produit (ce dont on ne peut se plaindre, par ailleurs). Il s'installe alors des habitudes à long terme visant l'efficacité immédiate et faisant l'économie d'une compréhension plus profonde. A contrario, les apprenants savent souvent prendre en compte leurs expériences récentes en mettant en oeuvre des procédures de tâtonnement efficaces à partir du moment où la démarche habituelle a échoué et dans la mesure où le problème posé reste simple ; mais ces types d'actions les font retomber dans le comportementalisme dénoncé plus haut.

L'hypothèse d'une démarche d'évitement de la conceptualisation de la notion de mémoire est séduisante par son (le moindre coût cognitif est souvent rencontré dans d'autres domaines) ; elle demanderait certes des vérifications quantitatives, mais elle apporte d'ores et déjà une quasi-certitude sur la nécessité de faire porter les efforts de formation sur l'aspect conceptualisation, sans pour autant

supprimer tout apprentissage par la pratique. Ce manque de synthèse théorie-pratique a été souvent rencontré dans l'observation du domaine.

4 - L'hypothèse de la rupture technologique et ses conséquences

L'ensemble des nouveaux outils informatiques crée à la fois facilités et opacité. La distance croissante entre l'utilisateur et son matériel favorise la situation cognitive globale de l'abstraction. D'autre part, le changement d'échelle des composants matériels exclut toute observation directe des effets des actions, contrairement aux technologies plus classiques (non électroniques). Ces facteurs conduisent à émettre l'hypothèse selon laquelle le passage aux technologies de l'informatique induirait un phénomène de **rupture**, s'opposant aux continuités mises en évidence dans les autres domaines par plusieurs auteurs [DEFORGE 85], [PERRIN 91], [SIMONDON 69] :

a) les principes et les structures de fonctionnement spécifiques des ordinateurs (réalisation séquentielle d'instructions, architecture de mémorisation, structures de données arborescentes, etc.) constituent un élément fondamental de la rupture : on ne trouvait pas jusqu'à présent un tel nombre de dispositifs originaux réunis dans un ensemble cohérent (il existait certains éléments séparés, de mise en oeuvre plus aisée) ;

b) quand on passe du domaine des dispositifs mécaniques (par exemple) à celui des systèmes de traitement de l'information, les échelles de temps et d'espace varient considérablement (de l'ordre de 10^{-6} seconde et mm) ; cette variation, due à la technologie électronique, a pour conséquence de diminuer, médiatiser, voire supprimer les possibilités d'observation directe de l'utilisateur, qui ne peut plus en inférer des liens de causalité entre actions et effets ;

c) les processus techniques sont profondément modifiés dans leurs finalités et dans leurs structures par leur informatisation : un logiciel de traitement de texte, de CAO ou de DAO ouvre des possibilités *inimaginables* (au sens propre) pour un utilisateur débutant mis en sa présence, ce qui empêche le sujet de construire des stratégies d'action en rapport avec les possibilités de l'outil ;

d) une autre modification des processus pourrait être due à leur séparation en deux entités distinctes : le dispositif physique (le système matériel, ses capteurs et actionneurs) et le dispositif de traitement d'informations proprement dit ; cette séparation introduit une et le passage à l'abstraction, obligeant ainsi le sujet à raisonner dans un système de représentations.

Apprendre à utiliser un système micro-informatique, c'est construire des connaissances et des représentations mentales du dispositif, acquérir des savoir-faire, être capable d'effectuer des transferts et des généralisations nécessités par la diversité des situations pour réaliser des tâches de manière pertinente (puis optimale) à l'aide de cet instrument.

Un sujet dispose toujours de connaissances sur un domaine avant d'apprendre ; elles sont souvent erronées ou incomplètes et s'érigent en [BACHELARD 47]. Dans les connaissances informatiques des utilisateurs (et souvent des formateurs...), il circule à plusieurs niveaux beaucoup d'idées **Erreur! Signet non défini.**, dont bon nombre fonctionnent comme de véritables Ces représentations préexistantes sont en pleine évolution depuis la diffusion des micro-ordinateurs dans le grand public ; on en perçoit cependant des constantes, surtout chez les enfants (chez les adultes observés, elles sont plus rationalisées, mais il peut en découler des images assez imprécises).

La connaissance des représentations préexistantes des apprenants peut permettre de savoir sur quels points doivent porter les efforts initiaux des enseignants pour construire des éléments immédiatement compréhensibles par les élèves et utilisables dès les premiers exercices. Ces représentations ont été explorées dans la recherche et les résultats des observations dans le public scolaire mettent en évidence d'une part l'**aspect magique des dispositifs**, totalement inexplicable (au sens propre, ce qui pourrait bien être un frein aux acquisitions), et d'autre part l'**anthropomorphisme**, qui a des conséquences importantes : cette identification du système informatique à une personne humaine, dans toutes ses capacités de raisonnement et d'action, s'est par exemple manifestée, lors de l'utilisation d'un logiciel de comptabilité, par des confusions entre le rôle de l'ordinateur - en principe chargé uniquement de tester l'égalité des écritures comptables - et le rôle de l'utilisateur, qui est de saisir les écritures justes. Dans l'esprit de l'élève observé, dès que l'ordinateur avait accepté l'écriture en raison de son égalité comptable, celle-ci était forcément exacte au niveau du sens des valeurs ; l'ordinateur semblait ainsi dispenser l'élève du moindre contrôle de vraisemblance, ce dernier attribuant au premier une capacité de réflexion vraiment .

La rupture technologique amplifierait les difficultés d'apprentissage, notamment par une adéquation moins immédiate de ce domaine aux principaux modes d'acquisition, à savoir le **raisonnement par analogie** et l'**apprentissage par l'action**.

Le raisonnement par analogie pose en effet de nouveaux problèmes dans ce cadre: l'univers source n'est pas bien défini et l'univers cible est éloigné de la source par des différences importantes dont, par exemple, l'impossibilité d'imaginer des solutions absolument originales. De plus, les objets du réel que l'informatique modélise sont transformés par cette modélisation ; on ne peut plus appliquer les mêmes opérations aux objets des deux : on crée, on transforme et on supprime ; certains objets ne sont plus directement modélisables...

Dans les apprentissages des dispositifs **Erreur! Source du renvoi introuvable.**, les connaissances préexistantes se réfèrent à des éléments voisins des dispositifs étudiés. L'acquisition de [VERGNAUD 85] peut s'effectuer à partir des apprentissages de type , car ils s'appuient sur l'observations directe des effets des actions et s'opèrent principalement par des raisonnements par analogie. Dans ces

conditions, les transferts et la conceptualisation sont possibles à partir des modes opératoires. En effet, l'expérimentation, la répétition du cycle sont autant d'éléments qui alimentent la réflexion de l'apprenant et le conduisent à relier actions et effets prévisibles de ces actions par des lois de causalité aisément observables. Or dans l'univers de l'informatique, abstrait, les observables directs font défaut, et les apprenants ont beaucoup moins de possibilités de trouver des éléments d'anticipation qui permettraient à ces conceptualisations de s'élaborer par l'usage des modes opératoires. Il paraît nécessaire de davantage ce que l'on fait.

La nécessité de concevoir des contenus de formation associant de manière complémentaire connaissances et savoir-faire semble à nouveau s'imposer. Une des principales difficultés sera de synchroniser ces deux types de matériaux d'apprentissage de façon que l'assimilation des éléments théoriques soit facilitée et renforcée par les acquisitions de savoir-faire corrélés.

5 - Des propositions didactiques

Les didacticiens des disciplines à "références savantes" (mathématiques et une partie des sciences physiques) se réfèrent souvent avec profit à l'épistémologie de leur domaine pour élaborer des concepts fondamentaux à partir de leurs contenus. Nous en sommes partiellement privés, car l'histoire de l'informatique [BRETON 90] manque de recul pour prendre en compte les avatars récents des micro-ordinateurs, et le domaine proprement dit de la "science informatique" concerne des théories qui n'ont pas souvent de relations directes avec nos pratiques.

Nous ne pouvons donc pas utiliser le concept de , élaboré par CHEVALLARD [CHEVALLARD 85] en mathématiques. Le concept de semble mieux adapté à notre domaine : J.-L. MARTINAND [MARTINAND 86] définit les situations de **référence** dans le domaine professionnel, axées sur des **pratiques à caractère technique**, dans des secteurs **sociaux** (professionnels) bien spécifiés. Ce concept permet de mieux aborder les enseignements et formations technologiques et professionnels dispensés dans les établissements et lieux spécialisés.

Si ce concept se rapproche davantage de nos activités, il soulève cependant quelques problèmes :

- les lieux de pratique sont multiples : contrairement à l'exercice d'un métier (au sens) industriel ou tertiaire, socialement situé, l'utilisation du micro-ordinateur se diffuse dans toute la société ;

- les buts ne sont pas identiquement définis : il n'y a aucun point commun, par exemple, entre le sur traitement de texte (ce sont les connaissances - très empiriques - de l'usager qui définissent les buts accessibles) et l'utilisation par un personnel hautement qualifié d'une station de Publication Assistée par Ordinateur (PAO) chez un éditeur.

De plus, ces situations sont actuellement en évolution rapide et en partie imprévisible. Si nous pouvons miser sur une diffusion de plus en plus importante de

la micro-informatique dans l'ensemble de la société, nous ne savons pas comment le public s'adaptera concrètement, ni ce qu'il en fera - intellectuellement et pratiquement...

Une adaptation du de B.-M. BARTH [BARTH 87 et 93] a été expérimentée en collège à propos de l'acquisition des concepts de mémoire et d'information en relation avec l'écran. Les résultats sont jugés qualitativement positifs dans la mesure où les élèves du groupe expérimental ont mieux réagi par la suite devant un problème de gestion de fichiers. Cette expérience est intéressante pour la dynamique qu'elle apporte dans le fonctionnement du groupe-classe à propos de notions nouvelles qu'il s'agit d'élaborer, souvent à partir d'idées fausses préexistantes chez les apprenants. Un autre intérêt de cette séquence expérimentale est d'avoir montré que l'on pouvait adapter ce modèle, initialement conçu pour une taxinomie descriptive en sciences naturelles, aux situations visant à faire acquérir des savoir-faire, des opérations pratiques sur le dispositif. La démarche originale de B.-M. BARTH pourrait, à notre avis, être mieux portée à la connaissance des enseignants.

L'expérimentation d'une présentation aux apprenants de fonctions générales (manipulations de fichiers, ensemble du langage d'interface d'une application par l'usage des) et de certains logiciels (traitement de texte) à l'aide d'outils du type (objets, attributs, méthodes) mériterait d'être approfondie par des travaux plus systématiques, car elle semble apporter des clarifications intéressantes. Une analyse des actions de l'utilisateur en termes de *fonction /arguments /[paramètres]* a également été pratiquée. Nous voyons là un des rares exemples de liaison étroite et fructueuse entre le niveau théorique de la et les utilisations des dispositifs micro-informatique sur le terrain.

Enfin, la prise en compte d'une pluralité de facteurs ne peut être que positive. Insister par exemple sur la motivation, même si c'est un paramètre difficile à cerner précisément, permet au moins de ne pas se tromper sur ce qui pousse un apprenant à apprendre. Comprendre que cette motivation peut donner du sens à l'apprentissage permet de garder un fil conducteur fort dans les contenus et évite de s'égarer dans des détails superflus.

CONCLUSIONS

Malgré les difficultés rencontrées à tous les niveaux par tous les acteurs, ceux-ci ne remettent pas en cause la pertinence de l'utilisation de l'ordinateur dans leurs travaux, même s'ils ont conscience des efforts à fournir pour en tirer bénéfice ; leur motivation est solidement ancrée dans l'évolution technologique du monde, globalement acceptée.

L'intérêt d'avoir observé les utilisations de l'ordinateur dans plusieurs disciplines est que nous avons pu y repérer des éléments récurrents caractérisant les difficultés d'acquisitions (informatiques) au travers d'usages différents, ce dont

nous n'étions pas sûrs au départ, compte tenu de la variété des situations et des conditions de leur mise en oeuvre.

La durée de l'étude et surtout la rapidité de l'évolution technologique nous ont permis de suivre et de comparer des usages d'installations signant des époques, des de micro-ordinateurs assez différenciées. Ce panorama a fait ressortir à la fois la disparition des problèmes liés aux contraintes matérielles et aux logiciels de l'époque et l'apparition de nouvelles questions - selon nous fondamentales - relatives à la mise à la disposition de publics non préparés d'outils complexes, dont on ne peut attendre un résultat performant qu'avec l'aide d'une formation (bien) structurée.

Cette recherche nous a montré également la nécessité d'une réflexion qui dépasse le cadre psychologique et didactique fixé initialement : on doit s'interroger maintenant sur la manière dont fonctionnent les réseaux de constitution et de transmission du savoir pour l'usage des nouvelles technologies, ainsi que sur les conditions de la réussite de leur diffusion dans les établissements d'enseignement et de formation. Toutes ces questions nécessitent l'ouverture vers un large éventail de paramètres institutionnels.

Jean François Levy

Institut National de Recherche Pédagogique
Département TECNE 91, Rue Gabriel Péri,
92120 MONTROUGE France
Tél. : (1) 46.12.87.04

Bibliographie

- BACHELARD, G.** [1947]. *La formation de l'esprit scientifique*. Paris, (douzième édition 1983) Vrin.
- BARTH, B.-M.** [1987]. *L'apprentissage de l'abstraction*. Paris, Nathan.
- BARTH, B.-M.** [1993]. *Le savoir en construction*. Paris, Retz.
- BRETON, P.** [1990]. *Une histoire de l'informatique*. Paris, Seuil.
- CHEVALLARD, Y.** [1985]. *La transposition didactique, du savoir savant au savoir enseigné*. Grenoble, (deuxième édition 1991) La Pensée sauvage.
- DEFORGE, Y.** [1985]. *Technologie et génétique de l'objet industriel*. Paris, Maloine.
- LEVY, J.-F.** [1993]. *Traitement de texte et bureautique, observations et propositions pour la formation professionnelle*. Paris, INRP (Rencontres pédagogiques ; 32.)
- LEVY, J.-F.** [1995]. *Pour une utilisation raisonnée de l'ordinateur dans l'enseignement secondaire ; analyses de pratiques et propositions pour un meilleur usage des instruments micro-informatiques*. Paris, I.N.R.P.-E.P.I.

MALGLAIVE, G. [1990]. *Enseigner à des adultes*. Paris, PUF.

MARTINAND, J.-L. [1986]. *Connaître et transformer la matière, des objectifs pour l'initiation aux sciences et techniques*. Berne, Peter Lang.

PERRIN, J. (sous la direction de) [1991]. *Pour une science des techniques*. Limonest, L'Interdisciplinaire.

RICHARD, J.-F. [1983]. *Logique d'utilisation et logique de fonctionnement*. Rapport IRIA (ronéoté), Institut national de Recherche en Informatique et Automatique, Rocquencourt, 78.

SIMONDON, G. [1969]. *Du mode d'existence des objets techniques*. Paris, Montaigne.

VERGNAUD, G. [1985]. *L'enfant, la mathématique et la réalité*. Berne, Peter Lang.

**Thème3: Informatique au Supérieur
(Sciences Humaines)**

La didactique de l'informatique à l'Université des Lettres et des Arts (Tunis I)

HUSSEIN HABAILI

présentée le mercredi 10 avril 1996 à 14h45

1. Introduction

Le défi de la formation des étudiants des lettres, des arts et des sciences humaines dans le domaine des sciences informatiques sera dans les années à venir considérable. Les innovations technologiques ont eu pour effets le passage des sciences du langage à la linguistique computationnelle; de la bibliothéconomie à la documentique, de la lexicologie à la lexicomatique ; de la lexicographie à la dictionnaire; de la traduction à la traductique : assistée par ordinateur (**TAO**) et automatique (**TA**); de la géographie classique à la géomatique et aux systèmes d'information géographique (**SIG**); de l'enseignement des langues à la didactique (**EAO**) ...

Dès lors que les nouvelles technologies de l'information et de la communication continuent à progresser, il s'avère nécessaire d'innover en matière de didactique en vue de favoriser l'appropriation de ces nouvelles technologies par le plus grand nombre et notamment les étudiants quels que soient leurs cursus, leurs spécialités et les finalités de leur formation. S'il est vrai que les étudiants en arts, lettres et sciences humaines connaissent un certain retard dans ce domaine, il est utile de faire un **bilan** global, de saisir les **raisons des retards** accumulés et d'identifier les freins et de proposer des **solutions** possibles.

A cet égard, il est important d'évaluer les expériences de la dernière décennie en matière de didactique de l'informatique destinée aux linguistiques, documentalistes, bibliothécaires, archivistes, traducteurs, géographes ... Et surtout de poser le **problème** en terme d'**évaluation des objectifs**, des **méthodes**, du **contenu des formations**, des **formateurs** et des **moyens logistiques, humains et didactiques**.

2. La didactique de l'informatique documentaire: état de la question

2.1. Introduction

L'évolution technologique qui marque cette fin de siècle ne cesse de nous poser des problèmes aux professionnels de l'information documentaire. Les opérations traditionnelles de collecte, de traitement et de diffusion de l'information devant une inflation documentaire persistante, posent un véritable problème aux professionnels. Ainsi, de façon lente, mais progressive, nous avons vu les bibliothécaires, documentalistes et archivistes glisser vers les technologies dites nouvelles, faisant de l'ordinateur un nouveau allié.

Le changement opéré dans le milieu professionnel devrait avoir des prolongements en amont, au niveau des **institutions de formation**. Pourtant, dans la réalité des faits, l'évolution des mentalités a été plutôt lente, dans le milieu où le souci de conserver un certain académisme a freiné l'introduction des nouvelles technologies dans les **programmes d'enseignement**.

Avec le début de la présente décennie, la recherche d'un savoir-faire informatique a connu un engouement certain au niveau de la formation dans tous les pays. Le phénomène s'est d'ailleurs accéléré au point que certains ont été souvent tentés de parler de révolution informatique. Mais une prise en compte des différents **facteurs** expliquant le développement de la didactique de l'informatique dans la formation des professionnels de l'information documentaire permet de nuancer le terme et de le limiter à celui de révolution.

2.2. Facteurs justifiant l'introduction de l'informatique

Un certain nombre d'éléments objectifs justifient l'introduction de l'informatique documentaire dans les programmes de formation des professionnels de l'information documentaire:

- Un **premier facteur** est celui de l'évolution de la notion de sciences de l'information qui a progressivement élargi le champ des connaissances de cette discipline. Cette évolution a donné jour à une interdisciplinarité qui intègre dans le corpus des sciences de l'information d'autres disciplines dont justement l'informatique;

- Un **second facteur** est celui de la **transformation de l'environnement** des bibliothécaires, documentalistes et archivistes: l'évolution technologique actuelle a, en quelque sorte, imposé ses nouveaux outils de travail pour l'organisation et le traitement de l'information documentaire;

- Un **troisième facteur** est celui du **nouveau comportement des professionnels**: le scepticisme et la réticence qui ont initialement limité l'introduction de l'ordinateur dans les tâches documentaires ont cédé la place à un nouveau réalisme.

Face à ces nombreux facteurs de changement, les institutions de formation aux sciences de l'information ne pouvaient rester en dehors de la zone d'influence des "Nouvelles Techniques en Information et en Communication (NTIC)". La formation devait en effet tenir compte d'un environnement demandant sans cesse un minimum d'expertise dans le domaine des nouvelles technologies.

L'association internationale des écoles en sciences de l'information (AIESI), en consacrant en 1980 le thème des Actes de ses 2^e journées d'études à l'enseignement de l'informatique documentaire, traduisait déjà l'importance de cette discipline dans la formation de professionnels d'un type nouveau. Lors de ces journées de réflexion, un premier bilan a été fait, portant sur 19 écoles parmi 23 membres de l'AIESI, qui a permis de dégager une vision assez large de la didactique de l'informatique documentaire.

2.3. Bref historique

L'Institut Supérieur de Documentation (ISD) de Tunis, qui assure l'essentiel de la formation des professionnels de l'information documentaire en Tunisie et dont nous livrons ici l'expérience en matière d'enseignement informatique n'est pas resté insensible à ces changements. C'est ainsi, que dès 1983, il a organisé une première table ronde à l'échelle maghrébine sur le thème: "Problèmes et perspectives de l'enseignement de la DBA au Maghreb".

L'introduction de la didactique de l'informatique dans le secteur des sciences humaines a commencé en Tunisie vers les années 70 successivement à l'Institut Bach Hamba, à l'Ecole Nationale d'Administration (ENA), à l'Institut de Presse et des Sciences de l'Information (IPSI) et l'Institut Supérieur de Documentation (ISD).

Les programmes de l'Institut Bach Hamba qui se divisaient en cours de documentation et en cours de bibliothéconomie ne contenaient qu'un cycle intitulé: "la documentation à l'aide des moyens mécaniques et semi-automatiques".

A l'ENA , les cours d'informatique documentaire étaient de : 25 heures consacrées à la "Documentation automatisée" , à raison d'une heure seulement par semaine sur un total de 537,5 heures par an. Il n'y avait aucun cours réservé à l'informatique documentaire en première année du Cycle des DBA.

En deuxième année du Cycle des DBA 25 heures étaient réservées à la "Documentation automatisée" à raison d'une heure par semaine sur un total de 475 heures par an.

À l'IPSI, un programme détaillé d'informatique documentaire a été élaboré en 1982 pour les trois années DBA (2ème, 3ème et 4ème années). Ce programme comportait deux grandes parties:

- Informatique générale;
- Informatique documentaire.

Deux heures étaient réservées à cet enseignement pour les trois années de formation en DBA.

L'ISD qui relève de l'Université des Lettres, des Arts et des Sciences humaines , Tunis I, a été créé en 1981, pour dispenser un enseignement de deux ans en vue de former:

- des documentalistes adjoints;
- des bibliothécaires adjoints;
- des archivistes adjoints.

2.4. Objectifs et programmes

En 1981 lorsque l'ISD prenait le relais de l'Institut de Presse et des Sciences de l'Information (IPSI), les cours d'informatique générale et d'informatique documentaire n'étaient enseignés qu'à partir du deuxième semestre pour les deux années d'études.

En 1982-1983, les étudiants avaient 25 heures de cours dans cette matière sur un total de 525 heures par an. A l'ISD, la didactique de l'informatique est organisé à deux niveaux, suivant en cela la configuration de l'Institut qui compte deux cycles de formation:

2.4.1. Niveau premier cycle

A la suite de sa restructuration en 1988, l'ISD assure des études de maîtrise pour les meilleurs étudiants de premier cycle, en vue de les former aux professions de bibliothécaires, archivistes et documentalistes. Pour s'adapter encore plus au marché du travail, l'ISD a introduit un enseignement d'options en 4ème année.

Ce cycle assure la formation des bibliothèques, archivistes et documentalistes de niveau maîtrise (Bac. + 4 ans). L'objectif des cours d'informatique dans ce cycle est de bien préparer les étudiants formés au travail dans une bibliothèque ou un centre de documentation automatisé. Le programme est ainsi conçu pour leur fournir les connaissances théoriques de base en informatique et les connaissances pratiques permettant de s'adapter à leur contexte professionnel. L'enseignement comprend:

- Un cours semestriel d'informatique intitulé : "Informatique : principes généraux" en première année du premier cycle de 37h 30 par an, sur un total de 525 heures par an. Ce cours a pour objectif de donner à l'étudiant une connaissance générale de l'ordinateur et quelques notions de base de l'informatique. L'apprentissage de la machine est ici indispensable pour les étudiants qui arrivent à l'ISD sans "bagage" en informatique.

- Un cours d'informatique documentaire proprement dit prévu en deuxième année du premier cycle s'étale sur toute l'année universitaire. Il s'intitule : "Informatique: principes de l'automation documentaire. Nouvelles technologies de l'information". Il comprend 50 heures sur un total de 575 heures. Il a pour objectif d'initier les étudiants aux différentes possibilités d'utilisation de l'ordinateur pour l'organisation, la gestion et le traitement automatique de l'information documentaire.

2.4.2. Niveau second cycle

Mis en place depuis 1988 le second cycle de l'ISD qui forme des bibliothécaires, documentalistes et archivistes reçoit deux catégories d'étudiants. D'une part, des étudiants titulaires d'un premier grade universitaire (DUEL, Licence ou maîtrise) délivré par une université et qui subissent un concours d'entrée; d'autre part, des étudiants déjà titulaires du diplôme du 1er Cycle en DBA de l'ISD. Le programme d'informatique du second cycle comprend deux volets:

- Un cours intitulé : "Informatique documentaire et études de cas", de 75 heures par an, destiné à un auditoire plus réceptif du point de vue des connaissances théoriques et à leur application dans le domaine professionnel, par l'étude des cas, se déroule en première année du second cycle.

- Un second cours intitulée " Informatique documentaire (Gestion des banques de données) de 25 heures par an, est destiné aux étudiants en deuxième

année du second cycle. Au niveau du contenu ce cours est surtout orienté vers l'exploitation des bases de données.

2.5. LES MOYENS

2.5.1 Les ressources humaines

Depuis 1988 cependant, l'ISD, dans sa politique de développement de ses ressources humaines, a fait des efforts au niveau du recrutement. Actuellement 3 enseignants permanents (sur 14 enseignants permanents et 42 vacataires professionnels) dispensent les cours d'informatique documentaire.

2.5.2 Les ressources matérielles

Dans ce domaine, une évolution qualitative et quantitative a été faite à l'ISD qui, grâce à la coopération internationale (française et canadienne), a pu augmenter son parc. Cette évolution de la logistique a permis la mise en place d'un Laboratoire d'informatique générale et un second Laboratoire d'enseignement de traitement de textes devant appuyer le développement de la didactique de l'informatique documentaire. Ces deux laboratoires comprennent actuellement les équipements suivants:

Hardware :

- Le laboratoire d'informatique générale : 12 micro-ordinateurs;
- Le Laboratoire de traitement de texte : 12 micro-ordinateurs.

Software :

- Le Laboratoire d'Informatique générale dispose de plusieurs logiciels documentaires: (CDS- ISIS, Babylone documentation, Alexandria etc.)
- Le Laboratoire d'enseignement de traitement de texte dispose de plusieurs logiciels : (Word, Wintex, Ready Set Go etc.).

3. Problèmes

Malgré une infrastructure (institutions équipements, outils...) assez développée par rapport aux pays en développement, la Tunisie n'a pas connu une ascension harmonieuse en matière de didactique de l'informatique dans le secteur des sciences humaines et sociales. Jusqu'à nos jours une vision globale et une planification claire de la didactique de l'informatique dans ce secteur fait terriblement défaut.

En conséquence, le démarrage difficile et tardif de la didactique de l'informatique a eu une influence négative sur le système de formation de l'humaniste . Beaucoup de choses restent à faire . L'identité, les limites de la didactique informatique n'ont pas été définies . Les buts et les **objectifs** qui déterminent le **profil** de l'étudiant de demain n'ont pas été clairement formulés.

De plus, le **contenu des programmes** reste indigent en matière d'intégration des nouvelles technologies de l'information et de la communication. L'enseignement de l'informatique ne représente qu'un très faible pourcentage des cours dispensés. Les méthodes pédagogiques restent encore classiques. Les moyens didactiques modernes sont rarement utilisés . Il manque toujours un corps

professoral spécialisé dans la didactique de l'informatique dans le secteur des sciences humaines et sociales.

Les étudiants handicapés par une formation de base littéraire sont passifs et ne manipulent que rarement l'ordinateur. A l'orée du troisième millénaire, les structures et les fonctions de la didactique de l'informatique dans le secteur des humanités et des sciences sociales doivent subir une observation critique minutieuse. Pour entrer dans ce nouveau siècle, il est nécessaire de s'appuyer sur une réforme des programmes dans le sens d'élargissement aux nouvelles technologies de l'information et de l'approfondissement de leur didactique.

3.1. Les problèmes humains

En dépit des efforts accomplis dans le secteur des humanités, la didactique de l'informatique bute néanmoins sur un certain nombre de **problèmes**, dont la solution est urgente, eu égard aux objectifs fixés pour le développement de cette discipline.

Au moment du démarrage des enseignements d'informatique et d'informatique documentaire, l'Université Tunis I, ne disposait pas de ressources humaines suffisantes. Les enseignements étaient confiés à des enseignants vacataires ou à des professionnels n'ayant pas une expérience pratique dans les applications de l'informatique aux différents domaines des sciences humaines et sociales . C'est ainsi, que des conservateurs ont initié les étudiants aux techniques de l'automatisation.

Les insuffisances apparaissent au niveau du nombre d'enseignants chargés de la didactique de l'informatique et au niveau et de la spécialité des enseignants qui sont généralement des professionnels et non de véritables spécialistes en informatique.

La première difficulté est relative au profil des enseignants. Rappelons que le corps professoral comprend dans sa presque totalité des professionnels . C'est ainsi que leur niveau d'expertise théorique ne leur permet pas de prendre en charge certaines questions relevant de l'informatique générale .

Par ailleurs, dans le cadre de leurs cours, ils ne sont pas entièrement versés dans l'informatique, parce qu'ils sont impliqués parallèlement dans l'enseignement d'autres matières. Une telle situation, jointe au retard mis dans l'acquisition d'un matériel performant explique ainsi l'absence d'un programme de recherche en informatique .

3.2. Les problèmes pédagogiques

La didactique de l'informatique dans le domaine des humanités s'articule autour du découpage traditionnel distinguant la théorie et la pratique. Au niveau du second cycle d'enseignement, on trouve des cours comprenant des heures de théorie et autant d'heures pour la pratique. Au début , l'essentiel des enseignements était

surtout théorique , avec un peu de pratique, les institutions universitaires ne disposant pas de matériel. La mise en place de quelques laboratoires de micro-informatique a cependant permis de mettre un peu d'équilibre, avec plus de possibilités de manipulation.

La faiblesse du volume horaire des cours pose un autre problème pédagogique. D'une façon générale, le volume horaire prévu dans les différents cycles pour le cours d'informatique générale est insuffisant vu le ratio existant entre le nombre de postes de travail et l'effectif étudiant. La même remarque est valable pour la didactique de l'informatique documentaire dont le volume horaire est insuffisant au niveau des deux cycles d'enseignement.

3.3. Les problèmes logistiques

Il faut reconnaître que les problèmes rencontrés en didactique de l'informatique sont inhérents en quelque sorte à la nature de l'enseignement de cette discipline. La didactique de l'informatique exige en effet un équilibre entre la théorie et la pratique, nécessitant la réunion de conditions matérielles et humaines, difficile à réaliser la plupart du temps. D'une façon générale, les problèmes rencontrés ici sont d'ordre logistique et humain.

A ce niveau, le nombre de micro-ordinateurs est insuffisant pour le nombre d'étudiants. A cela il faut ajouter le problème de maintenance du matériel existant qui demeure une préoccupation majeure et permanente, les appareils étant souvent en panne.

4. Solutions

4.1. Une formation adéquate

Devant la diversification croissante des pratiques et des profils professionnels, il est nécessaire de déterminer précisément les compétences respectives des diverses catégories d'étudiants. Cette définition est indispensable , notamment, à l'appréciation des **savoirs** et des **savoir-faire** de ces étudiants. La mission fondamentale du didacticien de l'informatique est d'assurer la mise en relation entre des demandeurs de formation (les étudiants) et des fournisseurs de cette formation (les enseignants formateurs).

La diversité des situations et des objectifs de mise en relation, la variété des dispositifs mis en oeuvre et des traitements automatisés, la spécificité de certains fonds documentaires ou de certains usagers nécessitent une diversification croissante des pratiques professionnelles et des niveaux de compétence requis pour mener à bien cette rénovation de la didactique de l'informatique en général et de l'informatique documentaire d'une façon particulière.

Plusieurs catégories professionnelles sont apparues au fil des ans. Chaque catégorie fait l'objet d'une description des tâches avec les rubriques suivantes: appellations principales, définition, contribution économique, activités dominantes, compétences requises, contexte professionnel, etc. Cette description des tâches, si

elle est nécessaire pour clarifier les besoins et pour servir de référence dans les divers "répertoires de métiers" mis en place dans les organisations, n'est pas suffisante pour décrire le "profil de poste" d'une profession.

4.1.1. Les profils

Il semble important de situer ces trois catégories les unes par rapport aux autres afin de préciser leurs compétences respectives dans un tableau situant les principales catégories par rapport à deux critères: la technicité de la fonction et l'importance du contact avec le public:

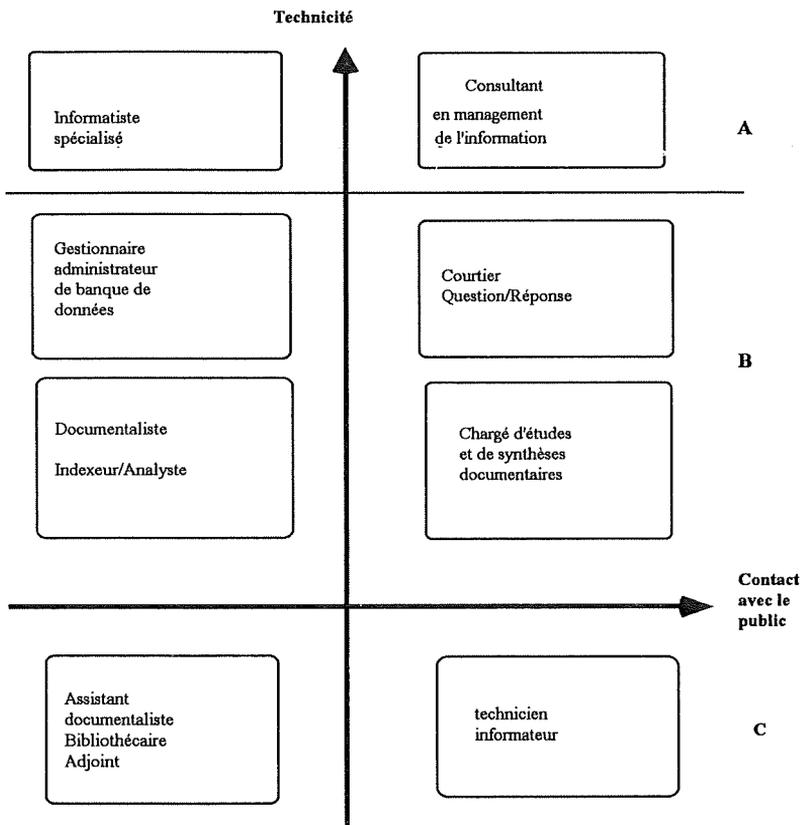


Fig. N° 1 : Les profils des professionnels de l'information

Cette grille commune d'analyse porte, d'une part, sur les connaissances (savoirs et savoir-faire) et, d'autre part, sur les aptitudes fondamentales:

Les **connaissances** sont à analyser à travers un nombre de domaines de compétence (ou familles de connaissances) ventilés en trois groupes :

- Groupe A

Profil: capacité de mettre opportunément en oeuvre des **savoirs universels**, reposant sur des connaissances théoriques mais applicables au domaine de l'informatique et de l'informatique documentaire. La possession de ces connaissances théoriques elles-mêmes, indépendamment de leurs applications, relève d'autres professions, comme celle du chercheur ou de l'enseignant.

- Groupe B

Profil : capacité de mettre en oeuvre des **savoir-faire généraux** nécessaires ou utiles à diverses professions, dont celles de l'informatique et de l'informatique documentaire. Ces savoir-faire ont donc leur place privilégiée dans un référentiel interprofessionnel. Mais dans le référentiel de l'information et de l'informatique, ils peuvent être définis de façon plus spécifique, pour tenir compte de leurs conditions d'utilisation.

- Groupe C

Profil: capacité de mettre en oeuvre des **savoirs et savoir-faire spécifiques** de l'informatique et de l'informatique documentaire indispensables à l'exercice de la profession . Ces savoir-faire peuvent naturellement figurer aussi dans un référentiel interprofessionnel s'ils s'avèrent également utiles à l'exercice d'autres professions; mais les niveaux de compétence peuvent alors être appréciés de façon plus différente.

4.1.2. Les niveaux

Chaque domaine est échelonné selon quatre **niveaux** significatifs, correspondant classiquement aux quatre niveaux d'objectifs pédagogiques retenus dans l'enseignement:

Niveau 1: Information sensibilisation

C'est le niveau de l'utilisateur . Une culture générale de base sur le domaine nécessaire (essentiellement connaissance du vocabulaire de base et capacité à exécuter quelques éléments pratiques ou concrets).

Niveau 2: Reconnaissance des concepts.

C'est le premier niveau du professionnel (usage de savoir-faire pratique). L'individu sait manipuler des outils de base, faire des travaux d'exécution spécialisés ou répétitifs, transmettre des consignes pratiques.

Niveau 3: Maîtrise des outils

L'individu est capable d'interpréter une situation et de porter un jugement impliquant une adaptation de la tâche, de créer un outil simple. Il peut choisir des actes élémentaires et les enchaîner dans des actions complexes.

Niveau 4: Maîtrise méthodologique

L'individu est capable de concevoir des outils ou des produits nouveaux, d'avoir une approche stratégique ou globale de son activité. Il perçoit la complexité pour éviter des erreurs et peut trouver des solutions originales adaptées à la situation.

4.1.3. Les composantes des programmes de formation

Nous assistons aujourd'hui à la construction d'un puzzle : télématique, micro-ordinateurs, stockage numérique des images et des sons, traitement de texte, édition électronique... Au fur et à mesure que les éléments de ce puzzle s'emboîtent, des sauts qualitatifs sont franchis.

La complexité et la dynamique souvent capricieuse et imprévisible des technologies de l'information, mêmes appliquées à des fonctions stables comme celle du transfert de l'information, compliquent l'élaboration des **programmes de formation** visant leur apprentissage et leur maîtrise dans les activités documentaires.

Initiation, savoir, savoir-faire: jusqu'où aller dans nos programmes de formation eu égard aux technologies de l'information? Gestionnaire, médiateur, concepteur, usager: que veut-on former? Qui veut-on former? Didactique des technologies de l'information ou exploitation des technologies de l'information dans la didactique des sciences informatiques.

Les composantes du puzzle nécessaires à l'élaboration de programmes et d'activités de formation aux technologies d'information se présentent de la façon suivante:

La **première composante** est celle des **sources d'information**. Il faut, de prime à bord, décider si le cours ou le programme se penchera sur chaque type de source d'information. Traditionnellement, certains programmes de formation excluait a priori toutes les sources d'information non textuelles. Certes, dans la grande majorité des programmes de formation actuels, les sources textuelles constituent encore le cœur des ressources documentaires. Toutefois, de plus en plus de programmes incluent, comme objet d'étude, des sources iconiques (banques d'images), des documents sonores, des films, vidéos, etc.

A cet égard, les supports et les technologies optiques sont susceptibles de changer sensiblement la dynamique des sources d'information (ex. indexation et repérage d'images, navigation dans un univers de documents composites, etc.).

La **seconde composante** est celle des **technologies** et correspond, dans les programmes de formation, aux **outils**. On pense, bien sûr, à **l'informatique** et surtout à la **micro-informatique**, mais aussi aux **techniques optiques (CD-ROM, CD-I, vidéo-CD, vidéodisque, etc.)**, aux technologies de communication réseaux, (**fibre optique, satellite, téléphonie, télécopie, etc.**), aux technologies audiovisuelles, aux technologies de micrographie, etc. Pour les programmes de formation, cette composante consistera surtout dans l'apprentissage et la maîtrise des outils de travail des futurs professionnels: équipements (micro-ordinateur, lecteur de CD-ROM, fax, etc.), logiciels d'exploitation et logiciels d'application générale (traitement de texte, chiffrier électronique, base de données, outils de communication, etc.).

La **troisième composante** est celle de la **fonction** ou de l'activité générale du **transfert de l'information** qui guidera, de façon immédiate, l'exploitation des technologies appropriées. Des activités telles que l'analyse documentaire et l'indexation sont des champs d'application pour des technologies spécialisées qui peuvent, elles aussi, faire l'objet d'activités de formation : courrier électronique, gestion automatisée de documents, indexation automatique, analyse automatique de textes, traduction automatique, télécatalogage, etc.

Ici, les programmes de formation se pencheront surtout sur les logiciels et produits d'application spécialisés correspondant à chaque fonction (ex. logiciels de gestion de bases de données, logiciels intégrés de bibliothèque, logiciels d'aide à l'interrogation de bases de données, bases de données sur CD-ROM, etc.).

La **quatrième composante** est celle des **intervenants**: "médiateurs du savoir" et apprenants. C'est à la fois la clientèle visée et l'orientation donnée à la formation. En effet, cette composante concerne la dimension de l'orientation à donner aux programmes de formation eu égard à la nécessaire intégration des technologies d'information dans les cursus: veut-on former surtout des bibliothécaires, documentalistes et archivistes :

- gestionnaires de technologies de l'information;
- concepteurs de technologies d'information;
- formateurs dans l'utilisation des technologies d'information;
- utilisateurs intensifs et intelligents de technologies de l'information.

La **cinquième composante** est celle des **niveaux de formation**. On peut distinguer trois grandes orientations en ce qui concerne le niveau de formation, c'est-à-dire le degré et le type d'apprentissage offert:

- Une **initiation** à l'informatique qui consiste à exposer les principales notions et connaissances générales de base sur une technologie; on vise à créer chez l'étudiant une représentation ou un modèle mental général de la technologie et de son utilisation en milieu documentaire.

- Un **savoir "éducation"** c'est-à-dire une connaissance théorique appliquée plus approfondie d'une technologie; on vise ici à créer chez l'étudiant un modèle

mental élaboré et complexe de la technologie et de son utilisation en milieu documentaire;

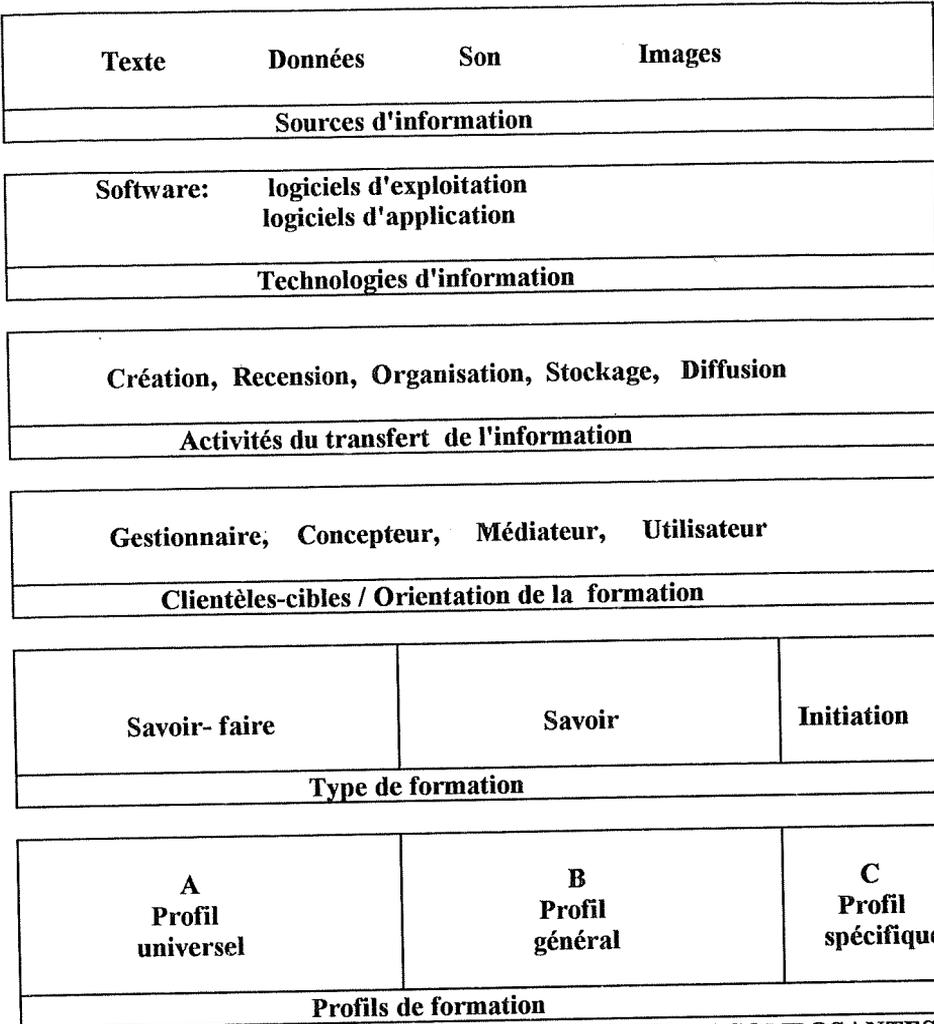
- Un **savoir-faire "training"**, une compétence, des habilités en un mot, c'est-à-dire une connaissance pratique, la maîtrise d'une technologie; on vise à développer chez l'étudiant une expertise concrète par des manipulations, des laboratoires, des travaux pratiques (télématique, micro-informatique, catalogage, analyse et indexation, etc.).

La réalité de nos programmes de formation, de même que toutes les contraintes de temps et de ressources auxquelles ils sont confrontés font que trop souvent, toute la formation informatique et technologique est concentrée, pour ne pas dire comprimée, dans un ou deux cours plus ou moins désincarnés du reste du programme.

L'orientation de la formation est alors davantage axée sur l'initiation que sur le savoir et le savoir-faire. Le danger de cette approche est que les étudiants ne font, généralement, pas le lien avec les autres savoirs et les autres disciplines. Ils finissent par croire que les technologies ne sont que des "gadgets" sans lien réel avec les fonctions documentaires, ou bien ils mettront toute leur énergie au détriment des fonctions documentaires. Dans un cas comme dans l'autre, il y a problème de formation pour acquérir une habilité technique.

Par ailleurs, les technologies de l'information ne doivent en aucun cas obnubiler les **programmes de formation** au point de leur faire perdre de vue leur **objet (le savoir)** et leur **objectif (le transfert du savoir)**. Les programmes de formation devraient donc viser d'abord et avant tout les **sources (d'information)**, le processus et les activités du transfert de l'information et les intervenants. C'est ce que l'on pourrait qualifier d'approche intégrée: un ou deux cours de base sur les technologies de l'information (informatique, autres technologies) où l'importance est donnée à l'apprentissage et à la maîtrise de l'outil (fonctionnement d'un micro-ordinateur, d'un lecteur CD-ROM) et exploitation intelligente et rationnelle des technologies de pointe dans des activités de formation aux différentes fonctions documentaires (interrogation de bases de données dans le cours de référence, de télé-référence, d'indexation automatique, etc.).

Cette réflexion s'applique parfaitement aux programmes de formation sur les technologies de l'information. Pour être vraiment efficaces eu égard aux technologies de l'information, les programmes de formation doivent être planifiés et conçus de façon intelligente, avec des objectifs précis et une ou des clientèles- cibles (médiateurs, concepteurs, gestionnaires, utilisateurs). Ces technologies de l'information et ces composantes des programmes de formation peuvent être présentées selon la schématique suivante:



**Fig. 2: TECHNOLOGIES DE L'INFORMATION / COMPOSANTES
DE PROGRAMMES DE FORMATION**

Programme standard modulaire d'enseignement des sciences de l'information de l'UNESCO

Brève description des modules (avec indication de la valeur en points)

Modules obligatoires	Nombre de points
C - 01	L'information dans son contexte social et communicationnel (2)
C - 02	Les utilisateurs de l'information (1)
C - 03	Méthodes quantitatives (2)
C - 04	Méthodes de recherche (1)
C - 05	Sources d'information (3)
C - 06	Système de stockage et de recherche de l'information (3)
C - 07	Services d'information (3)
C - 08	Traitement électronique de l'information (2)
C - 09	Application des technologies de l'information (4)
C - 10	Les télécommunications et la gestion des réseaux dans les systèmes d'information (1)
C - 11	Gestion des systèmes et services d'information (4)
C - 12	Economie et marketing de l'information (4)
	<hr/>
	Total des points des modules obligatoires : 28

Modules à option (4 modules de 2 points chacun) :

E - 13	Conception de systèmes d'information automatisées
E - 14	Systèmes et sources d'information orientés vers un secteur / un domaine particulier
E - 15	Conception et aménagement des bâtiments de bibliothèques de centres d'information /de documentation
E - 16	Recherche en ligne
E - 17	Services de bibliothèque et d'information spécialisés dans les domaines de la santé et de l'aide sociale
E - 18	Services de bibliothèque et d'information spécialisés dans le domaine de l'agriculture.
E - 19	Documents audiovisuels
E - 20	Impression, reliure et conservation.

5. CONCLUSION

Les efforts consentis dans la réflexion ont permis de dégager un ensemble de solutions pour le développement de l'enseignement de l'informatique documentaire et de sa parfaite intégration dans le programme d'enseignement et de recherche dans le secteur des sciences humaines et sociales.

Le bilan d'une dizaine d'années en matière de didactique de l'informatique générale et documentaire révèle deux types d'action à entreprendre pour la réalisation des objectifs précités, l'une en direction des hommes, l'autre en direction de la logistique:

- En ce qui concerne le matériel, conscience est prise de la nécessité de dégager des moyens d'acquisition et la maintenance tout en agrandissant le parc informatique. L'Université Tunis I pourrait offrir les services de ces laboratoires d'informatique et de traitement de texte par l'organisation de cours de formation continue et de séminaires pour autofinancer cet enseignement.

- Les domaines de la formation, de la spécialisation et de la pédagogie intègrent la donnée humaine. Une révision et une restructuration du programme d'enseignement de l'informatique générale et documentaire s'avère nécessaire à ce stade. Outre le renforcement du volume horaire, le recyclage des formateurs, la mise à niveau demeure un objectif prioritaire.

Hussein HEBAILI

Laboratoire de Recherche en Informatique
Arabisée et en Documentique Intégrée (RIADI)

BIBLIOGRAPHIE

[DES 92] DESCHATELETS, Gilles "L'enseignement des technologies de la DBA ou l'exploitation des technologies dans l'enseignement de la DBA" in *Actes du séminaire sur "L'enseignement de la documentation de la bibliothéconomie et de l'archivistique face aux nouvelles technologies"*, Tunis 24-27 novembre 1989. Tunis : ISD, Numéro spécial de la *Revue Maghrébine de Documentation* N° 6/7, pp.: 15- 28, Mai 1992.

[GAR 92] GHARIANI, Chaker et KEFI, Naceur "Les nouvelles technologies et la formation en DBA: Problèmes et perspectives" in *Actes du séminaire sur "L'enseignement de la documentation de la bibliothéconomie et de l'archivistique face aux nouvelles technologies"*, Tunis 24-27 novembre 1989. Tunis : ISD, Numéro spécial de la *Revue Maghrébine de Documentation* N° 6/7, pp.: 29- 46, Mai, 1992.

[GHA 84] GHANNOUCHI, L. "L'enseignement de l'Informatique documentaire en Tunisie" in *Actes de la première table ronde maghrébine sur "Problèmes et perspectives de l'enseignement de la DBA au Maghreb"*, Tunis 21-23 février 1983 Tunis : ISD, Numéro spécial de la *Revue Maghrébine de Documentation* N° 2, pp.:111- 116, 1984.

[KSI 95] KSIBI, Ahmed "L'évolution des technologies de l'information et l'enseignement des sciences de l'information en Tunisie" in *Actes de la cinquième réunion organisée par l'AFLI, La FTESI et le CDN sur "Etat des études sur la bibliothéconomie et sciences de l'information dans le monde arabe"*, Zaghouan, 1995, pp.: 362-402.

**Thème4 : Informatique au Supérieur
(Sciences Techniques)**

Entre informatique-outil et informatique-science: la question de l'usage: le cas de l'enseignement de programmation

Eric BATARD

présentée le mercredi 10 avril 1996 à 12h30

Mots-clés: enseignement de la programmation, épistémologie, pragmatique, sémiotique, théorie de l'informatique, usage.

Résumé : La programmation permet de développer des habiletés cognitives. Nous nous intéressons à la façon dont l'enseignement de la programmation peut contribuer à cet objectif. La variété des contenus et des stratégies sous-jacentes peut s'articuler autour de la dichotomie informatique-outil vs. informatique-science. Nous proposons un ensemble de concepts pour renouveler le débat en donnant à la question de l'usage un statut fondamental (et non pas périphérique) et pour systématiser l'intégration de la dimension contextuelle de l'usage à l'enseignement de la programmation.

Introduction

L'informatique n'est-elle qu'un outil technique, ou doit-elle être vue comme une science de l'ingénieur, voire une discipline scientifique à part entière ? La question est certainement aussi ancienne que l'informatique elle-même. Pour éviter de tomber dans l'ornière d'un tel débat, on peut tenter d'objecter que l'informatique, qu'on la voie comme outil ou comme discipline, n'est de toute façon guère âgée (un demi-siècle tout au plus), ce qui laisse à penser que ce débat s'éteindra sans doute de lui-même au fil des années. L'existence de débats réellement très anciens dans d'autres disciplines d'âge mûr — que l'on songe aux controverses sur la nature des objets mathématiques en mathématiques ou sur celle des phénomènes naturels dans les sciences physiques — tempère la foi que l'on peut mettre dans tout éclaircissement à venir. Et, indépendamment d'un éventuel pari sur l'avenir, cette question se pose actuellement, de façon quotidienne, avec acuité même, dans les situations d'enseignement de l'informatique, notamment en matière de programmation.

Dans ce contexte, elle peut se reformuler ainsi : ne faut-il voir dans l'enseignement de la programmation que l'objectif utilitaire, certes profitable mais cantonné au domaine technique, ou bien peut-on élargir ce point de vue assez restrictif ? On considère couramment que la programmation aide au développement des capacités intellectuelles (*intellectual skills*) et des habiletés cognitives, notamment concernant la résolution de problèmes. Ce qui nous intéresse ici, c'est la façon dont l'enseignement de la programmation peut contribuer à atteindre ce but, non du point de vue des stratégies pédagogiques, mais de celui des *contenus*.

La tendance, parfois clairement affichée, est de parvenir à un niveau suffisamment abstrait où se situerait l'apport recherché de l'enseignement de la programmation, ce qui permettrait de dépasser les objectifs purement utilitaires. Cette approche a pour conséquence d'exclure par principe la question de l'*usage*. Nous proposons de reconsidérer cette exclusion de principe. Selon nous, la dimension de l'usage est essentielle en informatique. Nous ferons ressortir ce point dans une première section consacrée à quelques conceptions courantes. Dans des travaux antérieurs (cf. bibliographie), nous avons systématisé la dimension de l'usage au moyen de quelques concepts (implémentation, traitement, fonctionnement) et nous y consacrerons une deuxième section où nous verrons en quoi ils permettent d'appréhender, sur un plan conceptuel, la question de l'usage et, par suite, d'élargir l'apport de l'enseignement de la programmation sur le plan des habiletés cognitives.

Cette communication s'appuie non seulement sur notre travail de recherche en informatique, mais aussi sur notre expérience de plus de sept années de formation en informatique devant des publics divers (école d'ingénieur, classe préparatoire, formation continue dans l'industrie et les services, université).

1. quel objectif pour l'enseignement de la programmation ?

Remarquons pour commencer que l'enseignement de la programmation impose sans détour possible le recours à une notation pour définir les programmes. Dans un premier temps, nous considérerons le point de vue le plus restrictif selon lequel enseigner la programmation revient à enseigner une telle notation sous la forme d'un (ou plusieurs) langage(s) de programmation. Ce point de vue sera ensuite étendu en raisonnant sur les récurrences qui dépassent les spécificités des langages de programmation.

1.1. la visée strictement utilitaire

Réduire l'enseignement de la programmation à celui d'un ou de plusieurs langage(s) de programmation signifie que l'on poursuit un but fortement utilitaire. Il s'agit dans ce cas de favoriser l'acquisition de connaissances à propos de détails assez spécifiques. On peut en distinguer quatre sortes, liées respectivement (du plus spécifique au plus général) aux systèmes, aux versions, aux compilateurs et aux langages. Chaque sorte implique des variations possibles dans l'écriture d'un programme, variations qui doivent être connues et comprises par un programmeur. Symétriquement il pourra être utile d'employer certaines variations (pour des raisons de performance le plus souvent) dans l'écriture de différentes versions d'un programme. Le futur programmeur doit intégrer ces diverses sortes de variabilité car il est susceptible (même en tant que programmeur non professionnel) de rencontrer des systèmes différents, des versions différentes, des compilateurs différents, voire des langages différents. Plus précisément,

- Il existe des détails propres à la machine ou au système cible. Ainsi un éditeur de logiciels peut proposer deux versions d'un compilateur pour un langage donné, chacune ne pouvant s'exécuter que sur un certain type de machines, voire

sur le même type de machine, mais sous un certain type de système d'exploitation. Par exemple, le compilateur Turbo Pascal¹ est actuellement associé au système MS-DOS et aux ordinateurs compatibles IBM PC, mais il a existé une version pour les ordinateurs à microprocesseurs 8 bits doté du système d'exploitation CP/M. On pourrait de même imaginer une version spécifique au système OS/2 à l'instar de celle (existante) spécifique à l'interface Windows.

- D'autres détails sont spécifiques à la version (*release*). Les changements de fonctionnalités entre deux versions d'un logiciel sont la raison même qui justifie le lancement d'une nouvelle version (qui ne doit pas se limiter à l'amélioration des performances et/ou à la correction des erreurs de la version antérieure). Par exemple, il a existé 8 versions de Turbo Pascal (versions 1 à 7 en passant par 5.5).
- Certaines spécificités sont propres aux choix de l'éditeur d'un compilateur. Ainsi Microsoft a positionné son QuickPascal comme concurrent du Turbo Pascal de Borland. Il s'agit toujours du même langage, Pascal, avec toutefois certains détails divergents. Dans ces conditions, certaines modifications syntaxiques s'imposent avant toute recompilation.
- Enfin, il existe des différences entre des langages d'une même famille. Il n'est pas question d'espérer une recompilation immédiate d'un programme Pascal si on veut l'adapter en C ou en Ada, même si, dans certains des cas assez simples abordés par les novices en programmation, la structure est suffisamment proche pour que de simples modifications syntaxiques permettent d'obtenir le résultat recherché.

Une vision purement utilitaire impose donc de tracer clairement la frontière entre ce qu'il est raisonnable d'espérer retrouver de façon persistante au milieu de la variété des systèmes, des versions, des compilateurs et des langages (respectivement), et ce qui sera inévitablement hétérogène. L'objectif est utilitaire car il s'agit d'offrir au programmeur une plus grande aisance face à une multitude de situations ; on se situe alors clairement du côté de la pratique.

Or l'existence de ces variabilités dans la pratique de l'informatique est à l'origine de questions qui se situent, quant à elles, sur un plan théorique dans la discipline informatique.

Nous en citerons deux directement mises en cause dans les quatre sortes ci-dessus:

- Les problèmes de *portabilité* entre systèmes ont conduit d'abord à des *normes* (qui restent du côté pratique), mais très vite à des *questions théoriques sur la formulation* de ces normes. Ainsi les problèmes de définition de la syntaxe des langages de programmation ont été un des moteurs des recherches sur les

¹ choisi comme base d'exemple pour sa grande popularité dans l'enseignement de l'informatique en France.

grammaires formelles (p. ex. [Wilson & Clark]). De même, les problèmes de définition de la sémantique des langages ont suscité bien des recherches (sans toutefois aboutir, pour ce qui est des normes, à des succès aussi éclatants qu'en syntaxe).

- Certains proposent même de transgresser les différences entre langages pour résoudre des problèmes de *réutilisabilité*. On a depuis longtemps envisagé et parfois mis en oeuvre des possibilités de réutilisation de code compilé (p. ex. [Wilson & Clark]). A un niveau d'abstraction plus élevé, on envisage notamment de définir des bibliothèques d'objets normalisées qui seraient communes à différents langages à objets.

En partant d'une vision extrêmement utilitaire, se limitant donc à l'acquisition cumulative des connaissances de détails somme toute très spécifiques en vue de la pratique, on débouche assez rapidement sur des questions essentielles touchant à la discipline informatique. Dans un contexte d'enseignement, on trouvera là des exemples qui justifient le choix de considérer l'informatique comme une discipline et non comme un simple outil. Même dans un cadre très restrictif où l'on considère qu'enseigner la programmation revient à enseigner un (ou des) langage(s) de programmation, nous voyons qu'une simple réflexion un tant soit peu systématique sur la pratique amène très vite à des questions théoriques profondes.

1.2. la visée abstraite

Étendre le point de vue utilitaire exposé précédemment revient à en prendre le contre-pied. En effet, dans ce cas, l'enseignement d'un (ou de) langage(s) de programmation n'est plus le but de l'enseignement de la programmation, mais un *moyen*. Par conséquent, selon cette stratégie, la programmation transcende les différents langages.

Si l'objectif est d'enseigner la programmation au delà d'une masse de détails techniques, il va être nécessaire d'opérer une distinction entre la notation choisie pour ce qui relève de ces détails techniques (le langage de programmation retenu) et la notation choisie pour exprimer ce qui est général. Le problème est alors d'identifier ou de construire, un langage de description qui transcende tous les langages de programmation.

Face à ce problème, les réponses sont extrêmement variées. Il n'est pas question ici d'en dresser une présentation exhaustive, mais nous allons tenter d'en suggérer la variété (cf. aussi [Boule] pour un rapport avec les mathématiques).

1.2.1. les langages de programmation universels

Tout d'abord, remarquons qu'historiquement (cf. [Wexelblat] et [Wilson & Clark]) certaines tentatives ont visé à l'élaboration d'un *langage de programmation universel*. Citons brièvement PL/I et, plus récemment, Ada. Aucun de ces deux langages n'a effectivement détrôné leurs concurrents.

PL/I (Programming Language number One) a tenté d'opérer une synthèse entre Fortran et Cobol tout en les dépassant. PL/I offrait en effet plus de possibilités pour la programmation scientifique (p. ex. par une gestion des représentations internes plus fine que celle de Fortran), plus de possibilités pour la programmation de gestion (p. ex. par une gestion plus étendue des chaînes de caractères que celle de Cobol) et des possibilités nouvelles (p. ex. la gestion de pointeurs). Cette synthèse avec dépassement a accouché d'un langage bien trop complexe pour qu'on puisse espérer le voir remplacer tous les principaux langages sur les principales machines.

De même Ada n'a pas eu le succès escompté. Il s'agissait dans ce cas d'opérer la synthèse entre des éléments de base d'une famille de langages qui avaient fait leur preuve, en particulier Pascal, et des conceptions plus modernes ou plus avancées de la programmation (en particulier les types de données abstraits, l'exécution concurrente, le temps réel). Le problème de la taille des compilateurs est apparu au début, mais, l'expérience aidant, il s'est vite estompé. Toutefois la complexité et une certaine lourdeur de la syntaxe ont freiné une pénétration qu'on annonçait illimitée. Enfin l'émergence des objets a favorisé l'élaboration d'autres langages modernes (y compris une nouvelle norme Ada 9X, intégrant les objets au langage Ada initial).

Concevoir la généralité de la programmation à travers la variété des langages de programmation impose donc, semble-t-il, de s'affranchir de ces derniers.

1.2.2. les langages génériques de description

Toutefois l'impossibilité pour un langage de programmation d'être effectivement universel² ne remet pas nécessairement en cause l'existence d'un *langage générique de description*, qui ne serait alors *pas* un langage de programmation.

Une première approche, fort usitée en matière d'enseignement de la programmation, est le choix d'une *notation algorithmique*. L'objectif premier d'une telle notation est de se libérer de certains détails pesants qu'on souhaite reléguer aux langages de programmation. Ces notations apparaissent en réalité comme des langages non exécutables et elles trahissent inmanquablement (surtout dans les contextes éducatifs) leur finalité : la réécriture des algorithmes dans un véritable langage de programmation. En tant que langages génériques, les notations algorithmiques souffrent d'une trop grande proximité avec les langages de programmation : leur finalité est flagrante, trop liée à un langage ou à une famille de langages et l'immense variété de ces notations ne réduit pas vraiment la variété au moins aussi grande des langages de programmation. L'intérêt ergonomique de

² ce qui ne remet pas en cause la possibilité *théorique* pour un langage d'être universel (cf. les machines de Turing).

telles notations ne saurait être nié, mais elles ne constituent pas cet aspect essentiel de la programmation qui transcenderait les divers langages.

Si une notation informatique ne répond pas à cette attente, c'est peut-être qu'il faut se tourner vers un autre type de langage qui a déjà fait ses preuves en matière d'universalité. C'est l'option retenue par certains chercheurs, notamment les fondateurs de la programmation structurée : C.A.R. Hoare et E. W. Dijkstra (cf. [Dahl], [Gries], et [Dijkstra] sur la question de l'enseignement³). Ce langage générique, c'est le langage mathématique, qui a fait ses preuves comme langage de description universel des phénomènes naturels. Concernant la programmation, il convient de signaler que deux axes s'inscrivent tous deux dans cette option : la sémantique dénotationnelle où l'effet des structures syntaxiques d'un langage est décrit par des fonctions mathématiques et la sémantique axiomatique où cet effet est formalisé au moyen d'une description de type logique.

Cette deuxième option a introduit une rigueur plus grande, à la fois en programmation et dans la discipline informatique toute entière. Ainsi C.A.R. Hoare déclarait « Les algorithmes et les programmes sont des objets mathématiques, et doivent être étudiés comme tels. » Ce souci de rigueur ne peut qu'être approuvé. Toutefois ce point de vue laisse en suspens une question de taille : dans le cadre d'une pratique, ne faut-il pas appréhender intuitivement l'objet avant de pouvoir l'étudier comme un objet mathématique ? Certes on considère généralement que les concepts de la physique quantique ne peuvent être appréhendés efficacement sans passer par la formalisation mathématique tant les phénomènes sont contraires à l'intuition. Mais la situation est bien différente en informatique ! Le constant recours à l'intuition est une *donnée constitutive* de l'informatique (cf. l'évolution des interfaces vers toujours plus de convivialité ou... d'intuitivité, l'évolution des langages de programmation et de requêtes vers des niveaux d'abstraction toujours plus loin de détails matériels, etc.). Dans le cas de l'informatique, par opposition à celui de la physique quantique, nous sommes enclin à penser que l'appréhension des concepts ne peut se faire efficacement sans recours à l'intuition (tant les situations d'usage sont contraires à la rigueur des mathématiques, serions-nous tenté d'ajouter).

Il s'avère risqué d'assimiler trop vite la programmation à un phénomène *naturel*. Il nous semble, au contraire, important d'affirmer la présence d'un élément d'ordre *culturel*, ou *social*, irréductible. Le langage de description générique recherché est donc le langage naturel. C'est d'ailleurs la conclusion pratique à

³ Dijkstra va jusqu'à suggérer que, pendant leur premier semestre en programmation, les étudiants se voient refuser l'accès à tout ordinateur afin qu'ils se pénètrent de la puissance expressive de la simple manipulation formelle de symboles, en dehors donc de tout rapport aux aspects opérationnels.

laquelle sont parvenus les comités de normalisation pour ce qui est de la définition de la sémantique des langages de programmation⁴.

Une simple notation ne suffit pas pour décrire la généralité de la programmation, il faut avoir recours à toute la "force" (et toutes les "faiblesses") du langage naturel pour cela. Mais ne peut-on trouver quelques principes généraux qui se dégageraient de la variété des langages de programmation ? Autrement dit, y a-t-il des concepts qui transcendent tous les langages ?

1.2.3. paradigmes et contextes de programmation

L'hypothèse de concepts qui transcenderaient les langages de programmation (par famille ou dans leur ensemble) est sous-jacente à l'étude comparative des langages de programmation, branche de l'informatique davantage approfondie dans le monde anglophone que francophone et, par conséquent sans doute, plus présente dans les cursus de l'un que de l'autre. (D'où des références majoritairement anglophones : [Appleby], [Barron], [Ghezzi], [Watt]. En français : [Wilson & Clark], traduit de l'anglais).

L'étude comparative des langages de programmation s'est appuyée initialement sur des données de type historique et descriptif. Il s'agissait d'organiser les langages en familles et de dégager quelques idées-force propres à chaque famille. La vision moderne de ces "familles" conduit à la notion de *paradigme* (cf. particulièrement [Appleby] pour l'initiation et [Watt] pour un niveau plus avancé). On reconnaît implicitement par ce terme qu'il existe non seulement des raisons (historiques p. ex.) qui président aux choix de conception d'un langage, mais aussi des hypothèses, parfois assez fortes, sur la nature du calcul ou sur le processus de résolution de problèmes.

Comme on peut le constater en situation d'enseignement, certains apprenants s'adaptent plus facilement à un paradigme qu'à d'autres. Ainsi certains vont préférer le paradigme impératif (celui des langages traditionnels comme Pascal ou C) au paradigme fonctionnel/applicatif de Lisp ou ML alors que d'autres vont être séduits par la programmation logique de Prolog⁵. Mais les études psychologiques ([Hoc]) font apparaître des différences qui ne suivent pas vraiment les frontières entre paradigmes. [Green] fait ressortir une différence profonde entre des styles plus ou moins soignés (« neat » vs. « scruffy ») et [Petre], qui s'intéresse aux programmeurs experts, souligne que les spécialistes s'appuient très souvent sur une notation personnelle qui emprunte à plusieurs paradigmes. Malgré leur indéniable importance, les paradigmes de programmation ne parviennent donc pas à circonscrire sur le plan des concepts l'activité de programmation.

⁴ ce qui ne remet pas en cause l'intérêt théorique, voire pratique à long terme, des recherches en sémantique informatique.

⁵ Partant de là, on peut s'interroger sur les réactions de débutants face aux langages multi-paradigmes [Placer] dont l'utilisation demeure pour l'instant confinée aux laboratoires de recherche, donc à un public déjà extrêmement averti.

Notons que l'étude comparative des langages de programmation s'intéresse aussi aux "à-côté" des langages eux-mêmes. Les choix de conception impliquent bien souvent des stratégies spécifiques de réalisation des compilateurs, des interprètes, voire des supports d'exécution. Ces aspects techniques ne concernent pas dans leur détail le débutant et nous n'y insisterons pas davantage. Toutefois un "à-côté" prend une importance grandissante dans les choix de conception : programmer oui, mais dans quel *contexte* ? S'agit-il de la programmation à grande échelle (*programming in the large*) où le programmeur intègre une équipe pour réaliser des projets de plusieurs dizaines de milliers, voire centaines de milliers de lignes ? Au contraire, s'agit-il d'une programmation "jetable", appelée aussi développement rapide d'applications (*rapid application development*), qui a parfois recours à des aides visuelles ?

On est bien loin des espoirs nourris au début de l'informatique d'arriver à un langage universel. Ces différentes situations de programmation commencent à être bien connues et font l'objet de nombreux travaux visant à développer des environnements et des outils spécifiques à telle ou telle situation. Toutefois il s'avère délicat d'intégrer de telles préoccupations à l'enseignement de la programmation⁶ pour les débutants. S'il est vrai que les bases demeurent les mêmes quelle que la soit le contexte de programmation, peut-être manque-t-il alors un moyen de le situer par rapport à ces bases.

En résumé, les concepts mis en oeuvre dans les différents langages de programmation ne suffisent pas à rendre compte de la totalité de l'activité de programmation. On peut être alors tenté par une approche hybride, ce que nous appelons *formalisations incomplètes*.

1.2.4. les formalisations incomplètes

L'existence admise de bases de la programmation précise la réponse à notre question de départ. Il est nécessaire de faire appel au langage naturel pour décrire la programmation et, plus précisément, ses concepts de base. Mais comment noter ces bases si l'on désire les rapprocher de la formulation dans un langage de programmation ? Nous retrouvons là des considérations déjà entrevues à propos des notations algorithmiques, mais ici sous l'angle d'une tentative de formalisation de concepts et non comme une relaxation des contraintes syntaxiques d'un langage de programmation.

1.2.4.1. les pseudo-codes

Traditionnellement on cherche à se rapprocher d'un langage de programmation en présentant un *pseudo-code* (voire notation algorithmique). Le pseudo-code est une notation, souvent fortement inspirée d'un langage de programmation, pour la définition d'algorithmes (dans le cas le plus abstrait) et de programmes (dans le cas le moins abstrait). On considère alors que les bases

⁶ il existe bien sûr des cours spécialisés (en Génie Logiciel, p. ex.), mais qui s'adressent à un public déjà spécialisé.

correspondent à un ensemble de schémas de base, plus ou moins directement en rapport avec un langage de programmation. Des études psychologiques ont constaté des différences dans l'utilisation de leur répertoires de schémas entre les programmeurs débutants et experts. Certaines méthodes d'enseignement s'appuient explicitement sur un ensemble de schémas (cf. [Gram]). Toutefois la connaissance des schémas ne suffit pas : il faut pouvoir les mettre en oeuvre, choisir le bon et l'adapter correctement. Il faut compléter cette connaissance par des conseils qui prennent alors la forme de proverbes, des principes, de recommandations (cf. [Kernighan] ou [Ledgard]), .

1.2.4.2. les spécifications formelles

On peut également tenter de *formaliser* le processus de programmation en s'appuyant sur des spécifications formelles qu'il est envisageable soit d'exécuter directement, soit de transformer de façon systématique en programme exécutable. Mais cela revient à déplacer le problème de départ : en supposant que l'élaboration d'un programme devienne immédiate une fois la spécification formelle écrite, comment écrire la spécification formelle ?

En d'autres termes, les approches s'appuyant sur une formalisation incomplète facilitent mais ne résolvent pas le "programming problem", c'est-à-dire la mise en application de savoirs en vue de la résolution par informatique d'un problème donné.

1.3. systématiser le tâtonnement

Face à tant d'incertitudes, certains adoptent une position bien modeste, en soulignant que la programmation reste un art (cf. le titre de [Knuth]). Ainsi le Professeur Jacques Pitrat suggère de considérer la programmation comme relevant de la "pensée intime". Cette notion de pensée intime due au psychologue Pierre Vermersch [Vermersch] dans ses travaux sur l'introspection capture l'activité de tâtonnement intérieur, de réflexion désorganisée de la pensée. Mais, en admettant qu'une part importante de la programmation relève d'un tâtonnement inspiré, cela n'aide guère le formateur. Quand bien même la réalité de la pratique serait essentiellement celle d'un tâtonnement inspiré, il faudrait trouver les moyens de le systématiser, même artificiellement, à la simple fin de le rendre transmissible à des apprenants.

Nous voudrions proposer un point de vue qui permette de systématiser ce discours sur le tâtonnement afin de faire ressortir un des intérêts de l'enseignement de la programmation sur le plan des habiletés cognitives.

2. une approche pragmatique

À plusieurs reprises dans la section précédente, nous avons eu l'occasion de constater une manière de divorce entre pratique et théorie en informatique. Les tentatives de théorisation s'appuient pour l'essentiel sur le langage mathématique et fournissent la part scientifiquement recevable de l'informatique (comme discipline). Les réflexions menées sur les aspects plus conceptuels (et par suite moins, voire pas

du tout, formalisés) ne sont généralement pas organisées en théories et apparaissent plutôt comme un recueil de "recettes", fort utiles mais désespérément étrangères à une systématisation. La question sous-jacente serait donc de savoir si la visée vers la généralité qui porte toute entreprise scientifique (cf. la célèbre formule d'Aristote "il ne peut y avoir de science que du général") implique d'exclure toute considération sur la pratique.

Dans le cas de l'informatique, une telle exclusion paraît bien peu justifiée puisque tant d'aspects dépendent de l'usage (citons, outre ceux que nous avons déjà rencontrés, le problème des instructions d'entrées clavier/sorties écran qu'il est pratiquement impossible de présenter sans faire référence à l'utilisateur). Toutefois, plus il est question d'usage, plus on s'éloigne, semble-t-il, d'un discours scientifique pour rejoindre un discours au mieux technique. Nous voudrions sortir de ce débat en proposant d'introduire une réflexion sur l'usage lui-même au niveau scientifique de l'informatique. L'idée en elle-même n'est pas neuve ; on trouvera divers projets où elle était en germe à des degrés divers (citons, en intelligence artificielle, les systèmes "espions" qui observent leur propre fonctionnement ([Pitrat90a] et [Pitrat90b]) et, en conception de systèmes, les propositions p. ex. de [Landauer] à propos d'une conception "centrée sur l'usager", *User Centered Design*).

Pour cela, nous distinguons deux étapes. Tout d'abord changer de point de vue sur l'objet, sortir de la vision du symbole que proposent les mathématiques et envisager un point de vue plus général, inspiré de celui de la sémiotique ([Eco], [Everact]). Cela nous conduit à recontextualiser l'objet. Il faut ensuite se donner les moyens conceptuels (et non pas formels) d'appréhender le rapport entre l'objet en contexte et son contexte.

2.1. recontextualiser l'objet

La pratique de l'informatique met en jeu l'interaction de deux processus : un processus cognitif chez l'usager et un processus physique dans l'ordinateur, à prendre ici comme système matériel, en l'occurrence un dispositif électronique. Toutefois l'informatique impose de constituer un objet symbolique qui est le lieu de rencontre de ces deux processus. Nous avons choisi d'appeler *forme* (sous-entendu symbolique) un tel objet ([Batard95]). Une forme est une représentation double : à la fois de quelque chose de l'ordre du mental (lien avec le processus cognitif) et de quelque chose de l'ordre du matériel, objectivable dans l'ordinateur (lien avec le processus physique).

La conception courante de l'informatique en tant que discipline vise à extraire les formes de la situation, c'est-à-dire à "oublier" les processus sous-jacents, physique d'abord, mais aussi cognitif. On a alors à faire à une informatique décontextualisée, qui est susceptible d'être formalisée. Ce choix a permis d'atteindre un niveau d'abstraction où la programmation n'était plus irrémédiablement liée à une machine spécifique. Il ne s'agit donc pas de renier cette conception, mais de l'étendre en constatant les limites.

Un exemple de limite nous est donné par les problèmes de représentations internes. Il est inévitable d'aborder ces problèmes dans l'enseignement de la programmation, même (surtout ?) face à des débutants. En effet, certaines erreurs ne peuvent s'expliquer sans y faire référence (p. ex. les dépassements d'indices peuvent impliquer la modification de variables non affectées par ailleurs, les dépassements de capacité peuvent expliquer ce qui serait considéré à juste titre comme une aberration en mathématique — par exemple obtenir un résultat négatif pour la somme de nombres positifs). De plus, certains choix, fortement contraints ou laissés à l'appréciation du programmeur, ne se justifient réellement que par rapport à des questions de représentations internes.

Insistons bien sur le fait que recontextualiser l'objet ne signifie pas qu'il doive rester prisonnier d'un contexte pesant : nous ne saurions nous faire avocat de la "bidouille". Nous voulons souligner que le contexte est nécessaire pour éclaircir certains points. Par conséquent, il nous paraît primordial d'intégrer cet aspect au niveau du discours scientifique, et pas de le cantonner à celui du jargon technique. Par suite, cette problématique de l'usage pourra également être harmonieusement intégrée à un discours pédagogique.

2.2. faire référence à la situation

Notre proposition est de systématiser un groupe de trois concepts qui permettent de faire référence à la situation ([Batard91]). Les notions existent déjà dans le jargon (cf. [Johnson] pour une approche moins systématique) mais, comme elles portent sur des aspects implicites, elles n'ont pas été systématisées. (Notons que le problème des connaissances implicites a déjà été rencontré dans la réalisation de tuteurs informatiques pour l'apprentissage de la programmation, [De Wolf]).

2.2.1. le fonctionnement

Pour expliciter les liens entre une forme et le processus physique sous-jacent, nous proposons de considérer un *concept de fonctionnement*. En d'autres termes, pendant l'analyse d'une forme, il peut être fait référence à ce processus en considérant un fonctionnement.

C'est utile quand le fonctionnement s'oppose, par exemple, aux propriétés mathématiques de la forme prise comme écriture mathématique. Par exemple, en Turbo Pascal, sous certaines conditions, la forme $X+1$ peut être négative alors que la valeur désignée par la variable de type INTEGER X est positive (p. ex. après $X:=\text{MAXINT}$ où MAXINT est le plus grand INTEGER représentable, c'est-à-dire 32767).

Le concept de fonctionnement peut servir soit comme élément explicatif (dans l'exemple, pour expliquer pourquoi la propriété mathématique n'est pas vérifiée), soit comme élément heuristique (l'exemple servirait alors p. ex. à illustrer une étude de la représentation interne des INTEGERS ou à montrer les différences de comportement d'un programme en fonction des valeurs retenues pour les options du compilateur).

2.2.2. L'implémentation

Pour expliciter les liens entre une forme et le processus cognitif sous-jacent, nous proposons de considérer un *concept d'implémentation*. De même que dans le cas du fonctionnement, il devient possible de faire référence à cet autre processus en considérant une implémentation.

Le mot *implémentation* est depuis peu entré dans les dictionnaires non techniques du français dont il a longtemps été exclu malgré des origines latines. En anglais, il signifie *mise en oeuvre, réalisation* notamment d'idées. Le jargon informatique anglo-saxon, puis français, l'a trouvé idéalement adapté pour décrire le rapport entre des idées ou des intentions et une forme. Notre choix de garder ce terme est une manière d'hommage au bon sens des praticiens. Notons qu'à nos yeux, il ne faut pas le confondre avec *implanter* qui signifie mettre en place et qui a une connotation beaucoup plus matérielle.

Là encore, le concept prend tout son intérêt quand le processus informatique s'oppose à ce qui paraîtrait être le simple bon sens. Prenons un exemple, toujours en rapport avec la programmation et Turbo Pascal : pour aboutir à l'exécution du programme chargé dans l'éditeur de texte de Turbo Pascal (la fenêtre courante pour les versions à partir de 6), la suite normale de commandes (héritée d'une informatique non interactive) est Compile, puis Run. Dans un souci de commodité pour le programmeur habitué à l'informatique interactive, les concepteurs de Turbo Pascal ont autorisé l'utilisation de la commande Run sur un texte non compilé. Dans ce cas, Run équivaut à Compile, puis Run. Le code compilé étant stocké en mémoire, il n'est pas utile de le recompiler dans le cas de deux commandes Run successives, à moins bien sûr que le texte source n'ait été modifié. Dans ce cas, le système compile le nouveau texte et exécute le nouveau code compilé. Toutefois il existe des cas où, malgré une absence de modification du texte source, il est indispensable d'effectuer une nouvelle compilation —citons la modification d'options de compilation au sein de l'environnement intégré ou la modification de fichiers de ressources dans la version Windows. Si l'on suit la logique du programmeur, c'est-à-dire celle liée à ses idées et intentions, il est absolument aberrant de demander une nouvelle exécution (par Run) d'un programme, sans doute faux ou incomplet, duquel on vient de changer les options de compilation ou les ressources. Si l'on suit la logique implicite du système, il est absolument logique de *ne pas* recompiler puisque le texte source n'a pas été modifié, ne pas recompiler impliquant d'exécuter le code déjà en mémoire.

C'est sur de telles divergences que se bâtit l'expérience du programmeur qui apprend à ne pas confondre les signes auxquels il est sensible et ceux en fonction desquels le système réagit (cf. aussi [Soloway] pour une vision à deux niveaux, mécanisme et explication). La difficulté est que la recherche, fort louable, d'une intuitivité toujours plus grande en informatique-outil favorise un oubli un peu trop systématique de ces divergences dans l'informatique-science.

2.2.3. le traitement

Pour expliciter les liens entre formes, nous proposons de considérer un *concept de traitement*. Il devient possible de faire référence à une forme sous-jacente, implicite, à partir d'une forme en considérant un traitement.

Ce concept, pressenti depuis le début de l'informatique comme central (cf. le nom *data processing* pour la première informatique de gestion, cf. également la contribution de J.-P. Bertrandias sur « mathématiques et informatique » in [Cornu]), correspond à une construction assez minutieuse. En effet, à partir d'une forme et d'un traitement, il est possible de faire référence à une autre forme qui, selon les cas, sera un résultat (celui obtenu en appliquant le traitement à la forme) ou une donnée (celle qui permet d'obtenir la forme initiale). Mais le traitement, pris en tant que concept, peut être lui même associée à une forme : c'est tout l'intérêt de l'informatique que de permettre la description sous forme symbolique de procédures se situant au niveau de l'action (C'est d'ailleurs en cela qu'elle se distingue des mathématiques vues comme logique formalisée).

Le point important pour notre propos ici est de reconnaître la part d'implicite que met en jeu la lecture des formes par le programmeur. Ainsi, si nous considérons l'extrait de programme qui suit : `if (x==1) then` on peut l'interpréter comme un programme Pascal erroné à partir du deuxième signe = ou comme un programme C erroné à partir de `then`. L'analyse d'une forme s'effectue donc toujours par rapport à des implicites, dont certains relèvent du processus physique sous-jacent, d'autres du processus cognitif sous-jacent et d'autres enfin relèvent que l'organisation des formes retenues (qu'on l'appelle support d'exécution, structure, architecture, ou autre).

Conclusion

Nous avons consacré une première section à envisager plusieurs des stratégies déjà mises en oeuvre en vue de dégager un corpus de base, une quintessence de la programmation. Nous avons constaté que la part entre informatique-outil et informatique-science était difficile à faire. En conséquence, nous avons introduit dans une deuxième section quelques concepts destinés à préciser la place de l'usage⁷ en informatique ([Batard93]).

Les propositions que nous avançons concernent au premier chef la didactique de l'informatique. En effet, les concepts d'implémentation, de traitement et de fonctionnement autour desquels nous avons articulé notre analyse des aspects implicites de l'interaction homme-machine peuvent servir de points de repère pour faire ressortir auprès des apprenants l'existence de ces implicites ainsi que les savoirs et les risques d'erreurs qui y sont associés. Outre un bénéfice purement

⁷ pour être complet, il faudrait préciser que l'usage est l'activité de l'utilisateur réel et qu'il faudrait, pour être strict, l'opposer à l'utilisation qui est la construction conceptuelle de l'informaticien qui raisonne en fonction d'un utilisateur toujours fictif, par construction. Pour plus de détails, nous renvoyons à nos travaux.

technique et utilitaire, les apprenants peuvent retirer, sur le plan des habiletés cognitives, un avantage certain de cette sensibilisation car elle leur permet de s'interroger sur leurs pratiques (et d'ailleurs pas uniquement sur leur pratique de l'informatique).

Malgré certains présupposés diamétralement opposés sur la nature des objets informatiques, nous rejoignons Dijkstra ([Dijkstra]) qui militait pour faciliter l'oubli de l'aspect opérationnel d'un programme et pour la reconnaissance de son texte comme matériau susceptible d'être travaillé formellement. Cet oubli ne peut se concevoir qu'au prix d'une réflexion sur d'autres oublis, imposés par l'usage de l'informatique. Selon nous, il ne suffit plus de jeter l'anathème ou de porter aux nues tel ou tel point de vue, mais il faut intégrer cette dimension pragmatique en considérant que l'informatique nous projette dans un système de signes spécifique, et non pas seulement dans un monde formel où règnent les lois mathématiques. Cette option prend une ampleur grandissante, des chercheurs d'horizons très variés aboutissant à des conclusions voisines (pour le point de vue sémiotique [Andersen90], [Andersen93], pour le point de vue intelligence artificielle [Bachimont], pour le point de vue épistémologique sur informatique et mathématique [Vaudène]).

Quant à l'interrogation sur les pratiques, avantage retiré de l'apprentissage de la programmation, elle correspond à un questionnement sur la systématisation et la formalisation d'activités jugées comme allant de soi et elle rejoint par là un effet déjà constaté de l'informatique que ce soit comme remise en cause de l'organisation du travail dans son application courante, ou, pour reprendre une remarque de J. Pitrat, comme moyen de penser différemment certains problèmes, dans le cadre de l'intelligence artificielle.

Les paléontologues considèrent que l'homme est devenu homme quand il a commencé à fabriquer des outils. La fabrication d'outils a certainement contribué à exercer les habiletés cognitives des premiers hommes : anticipation, planification, mémorisation... Après quelques milliers d'années, la fabrication d'outils logiciels, même très frustrés comme peuvent être ceux dus à des débutants, ne fait pas exception : l'interrogation sur les pratiques est un moyen d'évolution essentiel et il est heureux que l'informatique (outil ou discipline) la suscite à un haut degré.

Eric BATARD

LABORATOIRE D'INFORMATIQUE DE L'UNIVERSITE DE TOURS
&

GRUPE CODISIMA (UNIVERSITE DE CAEN)

E3I - Technopole - boîte n°4

37913 TOURS CEDEX 9

Tél. : 47 36 14 14 e-mail : batard@univ-tours.fr

Bibliographie

- [Andersen90] **Andersen, P. B.** *A Theory of Computer Semiotics*. Cambridge University Press, 1990.
- [Andersen93] **Andersen, P.B.et al.(ed).** *The Computer as Medium*. Cambridge University Press, 1993.
- [Appleby] **Appleby, D.** *Programming Languages: Paradigms and Practice*. McGraw Hill, 1991.
- [Bachimont] **Bachimont, B.** *Le contrôle dans les systèmes à base de connaissances*. Hermès, 1992.
- [Batard91] **Batard, E.** *Rôle et place de l'utilisation dans les conceptions de l'informatique*. Thèse de doctorat en informatique, Université Paris VI, Paris, décembre 1991.
- [Batard93] **Batard, E.** "Ce que l'informatique attend de ses usagers : une réflexion sur certains concepts de base", communication au congrès AFCET'93, Paris, 8-10 juin 1993.
- [Batard95] **Batard, E.** "L'anthropomorphisme en intelligence artificielle : erreur de jeunesse ou problème mal posé", communication aux 3èmes journées francophones sur l'intelligence artificielle distribuée et les systèmes multi-agents, Chambéry, 15-17 mars 1995.
- [Barron] **Barron, D. W.** *An Introduction to the Study of Programming Languages*. Cambridge University Press, 1977.
- [Boule] **Boule, F.** "Mathématiques et informatique", *Bulletin EPI*.
- [Cornu] **Cornu, B.** (éd) *L'ordinateur pour enseigner les mathématiques*. Presses Universitaires de France, 1992.
- [Dahl] **Dahl, O.-J, Dijkstra, E.W. & Hoare, C.A.R.** *Structured Programming*, Academic Press, 1972.
- [De Wolf] **De Wolf M. & Milgrom E.** "Tuteurs informatiques pour l'apprentissage de la programmation", *Technique et Science Informatique*, v. 11 (1992) n° 6, pp. 9-37.
- [Dijkstra] **Dijkstra, E.W.** "On the Cruelty of Really Teaching Computing Science", *Communications of the ACM*, Vol. 32, n°12 (décembre 1989), pp. 1398-404.
- [Eco] **Eco, U.** *Le signe*. Editions Labor, 1988. (également Livre de Poche, 1992.)
- [Everaert] **Everaert-Desmedt, N.** *Le processus interprétatif*. Pierre Mardaga éditeur, 1990.
- [Ghezzi] **Ghezzi, C. & Jazayeri, M.** *Programming Language Concepts*. Wiley, 1982.
- [Gram] **Gram, Anna** *Raisonnement pour programmer*. Dunod, 1986.
- [Green] **Green, T.R.G.** "The Nature of Programming" in [Hoc], pp. 21-44.
- [Gries] **Gries, D.** *The Science of Programming*. Springer Verlag, 1981.
- [Hoc] **Hoc, J.-M. et al.** (ed) *Psychology of Programming*. Academic Press, 1990.
- [Johnson] **Johnson G. J.** "On Metaphor and the Difficulty of Computer Discourse", *Communications of the ACM*, v.37 n°12, pp. 97-102.

- [Kernighan] **Kernighan, B.W. & Plauger, P.J.** *The Elements of Programming Style*. Addison Wesley.
- [Knuth] **Knuth, D.E.** *The Art of Programming*. 3 tomes parus, Addison Wesley, 1973.
- [Landauer] **Landauer, K.** *The Trouble with Computers*. MIT Press, 1995.
- [Ledgard] **Ledgard, H.F.** *Programming Proverbs*. Hayden, 1975. En français *Proverbes de programmation*. Dunod, 1978.
- [Petre] **Petre, M.** "Expert Programming and Programming Languages" in [Hoc], pp. 103-115.
- [Pitrat90a] **Pitrat, J.** "La nouvelle voie de l'intelligence artificielle", *Technologies*, (9/90), pp. 191-200.
- [Pitrat90b] **Pitrat, J.** *Métaconnaissances, futur de l'intelligence artificielle*. Hermès, 1990.
- [Placer] **Placer, J.** "Multiparadigm Research : A New Direction in Language Design", *SIGPLAN Notices*, b. 26 n°3 (3/91), pp. 9-17.
- [Soloway] **Soloway E.** "Learning to Program = Learning to Construct Mechanisms and Explanations", *Communications of the ACM*, v.29 n°9, pp. 850-8.
- [Vaudène] **Vaudène, D.** *Une contribution à l'étude des fondements de l'informatique*. Thèse de doctorat d'état en informatique, Université Paris VI, octobre 1992. Publiée comme le rapport du LITP 92.75.
- [Vermersch] **Vermersch, P.** *L'entretien d'explicitation en formation initiale et continue*. ESF, 1994.
- [Watt] **Watt, D. A.** *Programming Language Concepts and Paradigms*. Prentice Hall, 1990.
- [Wexelblat] **Wexelblat, R. L.** (éd.) *History of Programming Languages*. Academic Press, 1981
- [Wilson & Clark] **Wilson L. & Clark R.** *Comparative Programming Languages*. Addison Wesley, deuxième édition 1993. En français, *Langages de programmation comparés*. Addison Wesley France, 1994.

Evolution des langages de programmation, évolution des méthodes d'enseignement

Jean VICARD

présentée le mercredi 10 avril 1996 à 12h30

1 - Introduction

C'est peu de dire que le métier d'informaticien en général, et de développeur en particulier, a évolué au cours des trente dernières années. Par voie de conséquence, les méthodes et les contenus des enseignements se doivent d'accompagner, voire d'anticiper ces évolutions.

Ainsi l'utilisation sur un micro ordinateur, d'un environnement de type « windows » met immédiatement les étudiants en présence d'un système complexe à haut niveau à d'abstraction.

Pour leur permettre de tirer rapidement partie de ces nouveaux environnements, il me paraît important d'insister dès le début de leur formation sur plusieurs notions.

La première notion est la réutilisation du logiciel. On ne va pas refaire sa "fenêtre mais réutiliser les éléments logiciels qui permettent de la « Fabriquer ».

La seconde notion est la maîtrise de la complexité qui conduit à introduire rapidement des éléments de structuration tels que les sous-programmes.

Bien entendu, il n'y a réutilisation possible d'un logiciel que celui-ci répond à des critères de qualité. Une condition nécessaire pour produire un logiciel de qualité est de déceler les erreurs le plus tôt possible dans le cycle de développement. Un langage fortement typé est donc tout à fait souhaitable.

De mon point de vue, le langage Ada héritier du langage Pascal permet dans une large mesure de tenir ce pari et permet ensuite d'aborder le langage C++ armé d'un certain nombre de principes.

2 - Les apports du langage Ada dans l'apprentissage de la programmation.

Le langage Ada reprend la base syntaxique et les principes généraux qui ont fait le succès du langage Pascal.

2.1 Les types scalaires

Le langage offre de grandes possibilités pour décrire les types scalaires. Il est en particulier tout à fait possible de définir un type en exprimant l'ensemble de ses valeurs dans le *domaine du problème* et non pas comme beaucoup de langages l'imposent parfois dans le *domaine de la machine*.

Ainsi lorsque les jours de la semaine font partie du domaine du problème étudié, il est tout à fait souhaitable de définir un type tel:

```
type T_JOUR is (LUNDI, MARDI, MERCREDI, JEUDI, VENDREDI, SAMEDI,
                DIMANCHE);
```

Lorsque le domaine du problème est celui des probabilités, le langage permet de définir un type adapté.

```
type T_PROBA is delta 0..01 range 0..1.0;
```

Nous avons ainsi la certitude de ne pas engendrer des probabilités supérieures à 1, voire négatives. Lorsque nous devons manipuler des entiers dans l'intervalle 0 à 100.000 il est possible de définir le type adapté au problème.

```
type T_ENTIER is range 1..100_000;
```

Le type T_ENTIER est bien entendu préférable au type INTEGER. En effet, pour ce dernier type, l'ensemble des valeurs peut varier d'une machine à l'autre.

2.2 Utilisation des attributs

L'utilisation systématique des attributs conduit à une programmation qui rend possible la réutilisation.

Exemple

```
type T_TAB is array (T_POSITIF range <>) of T_ENTIER;
```

```
function SOMME (Le TABLEAU : in T_TAB) return T_ENTIER is
```

```
    S: T_ENTIER := 0;
```

```
begin
```

```
    for I in T 'RANGE loop
```

```
        S := S + T (I);
```

```
    end loop;
```

```
end SOMME;
```

Utilisations possibles de SOMME

```
T1 : T_TAB (1..100);
```

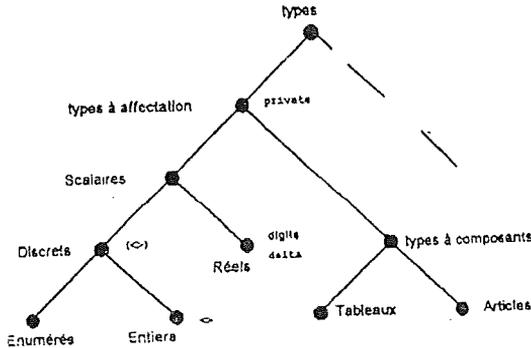
```
T2 : T_TAB (35..74);
```

```
S := SOMME (LE_TABLEAU =>T1) + (LE_TABLEAU => T2 (20..50));
```

On peut constater dans un dernier exemple, que les tableaux T1 et 2 sont de même type bien que n'ayant pas le même intervalle d'indice, qu'une tranche du tableau a toutes les propriétés d'un tableau, que dans une boucle *for* la variable est déclarée automatiquement et utilisable uniquement à l'intérieur de la boucle.

2.3 Hiérarchisation des types

L'utilisation des attributs conduit à préciser les différentes catégories de types. Ces différentes catégories sont représentables par le graphe de la figure 1. Ce graphe est obtenu de la façon suivante. Les sommets représentent une catégorie, les arcs représentent une relation est un. Cette hiérarchisation conduit à introduire rapidement la notion de généricité qui est un manière simple de réutiliser du logiciel.



2.4 Les sous-programmes

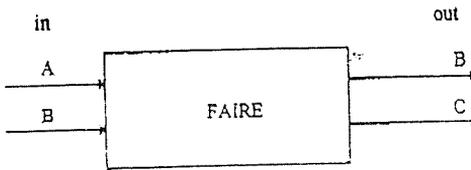
La notion de sous-programme est la première notion qui permet de mettre en oeuvre des règles de décomposition d'un problème " complexe" en problèmes plus simples.

Le langage Ada offre comme le langage Pascal la notion de fonction et de procédure mais il apporte, quelques avantages supplémentaires.

Tout d'abord la distinction possible entre la *spécification* d'un sous-programme et de son *corps* conduit à mettre en évidence la différence entre le QIJOI et le COMMENTT. Cette différence étant souvent source de confusion pour les débutants.

2.4.1 - Les procédures

Elles sont introduites comme des actions spécifiques simplifiant l'écriture de l'algorithme. Ainsi, à partir du schéma fonctionnel de la figure 2 caractérisant l'action FAIRE.



On peut dire que pour FAIRE
 on a *besoin* de A (in)
 on *modifie* B (in out)
 on *produit* C (out).

On passe très facilement à la spécification de la procédure

```

procédure FAIRE (A : in T_1;
                 B : in out T_2;
                 C : out T_3);
  
```

On remarque qu'on n'a à ce niveau nul besoin de parler du passage des paramètres par valeur, par adresse par référence. On se situe là encore au niveau du problème, pas au niveau de la machine. On ne s'intéresse qu'au comportement de la procédure.

L'appel d'une procédure peut se faire par position comme en Pascal ou en C, cependant, l'appel par nom offre des avantages.

```
FAIRE ( A => X, B => Y, C => Z );
```

Il explicite le lien entre le *paramètre effectif* et le *paramètre formel*.

2.4.2. Les fonctions

Outre les possibilités "classiques" habituelles, elles permettent de familiariser l'étudiant avec la *surcharge des opérateurs* qui est un élément de maîtrise de la complexité dans la mesure où celles-ci est employée à bon escient. Les paramètres d'une fonction sont in par défaut, ce qui est la garantie de la non modification des paramètres d'appel de la fonction.

En résumé, l'introduction de la notion de sous-programme permet d'introduire les notions de *décomposition*, *spécification*, *encapsulation*, *d'unité de compilation*. En outre, la possibilité de compiler séparément la spécification d'un corps permet d'imposer des contraintes à l'étudiant chargé de développer le corps et par conséquent d'affirmer la notion de contrat.

2.5 Les paquetages

Ils permettent d'introduire la notion de composant logiciel réutilisable. C'est aussi une manière simple d'introduire la notion de classe.

Exemple

le type abstrait F-FRACTION

package P_FRACTION is

 type T_NUMERATEUR is range -1000 .. 1000;

 subtype T_DENOMINATEUR

 is T_NUMERATEUR range 1 .. T_NUMERATEUR'LAST;

type T_FRACTION is private;

procedure LIRE (IA FRACTION: out T_FRACTION);

procedure ECRIRE (LA FRACTION : in T_FRACTION);

function "+" (GAUCHE, DROITE : in T_FRACTION
 return T_FRACTION;

function "+" (GAUCHE : in T_NOMBRE;
 DROITE : in T_FRACTION return T_FRACTION;

function "+" (GAUCHE in T_FRACTION;
 DROITE : in T_NOMBRE return T_FRACTION;

...
EXC_TROP_GRAND, EXC_TROP_PETIT,
 EXC_TROP_FIN : exception;

private

type T_FRACTION is

record

LE_NUMERATEUR : T_NUMERATEUR;

LE_DENOMINATEUR: T_DENOMINATEUR;

end record;

end P_FRACTION;

2.6 - Les exceptions

Tous les étudiants (les enseignants aussi d'ailleurs) qui commencent à programmer en Ada sont confrontés à l'exception CONSTRAINT_ERROR qui apparaît à chaque fois qu'on tente d'affecter à une variable une valeur hors du domaine du type ou lorsqu'on tente d'atteindre un composant de tableau hors du domaine de l'indice de celui-ci. L'introduction des exceptions notamment dans les paquetages est un moyen simple de préciser ce qui se passera dans les cas limites. Ce problème n'est en général que peu abordé lorsqu'on enseigne la programmation avec le langage Pascal ou le langage C. Il est très important de sensibiliser l'étudiant au fait qu'un produit logiciel comme un produit matériel ne doit (devrait) pas tomber en panne.

2.7- La généricité

C'est à travers la généricité que la réutilisation du logiciel prend toute sa signification. Elle peut être illustrée par des exemples simples puis plus compliqués et elle permet de tirer partie de a hiérarchie des types illustrée sur la figure 1. Le premier exemple "classique" concerne la procédure ECHANGER. L'échange de 2 variables de même type est réalisé au moins dans 2 ou 3 cas spécifiques avant d'aborder la généricité. Il est donc "facile" d'introduire le modèle générique suivant pour l'échange de 2 variables.

```

generic
type T_INCONNU is private;
procedure G_ECHANGER (L_UN : in out T_INCONNU;
                      L_AUTRE : in out T_INCONNU );

procedure G_ECHANGER (L_UN : in out T_INCONNU;
                      L_AUTRE : in out T_INCONNU) is
  AUX : T_INCONN;
begin
  AUX := L_UN; L_UN: L_AUTRE; L_AUTRE := AUX;
end G_ECHANGER;

```

A ce niveau, il est important de remarquer que la généricité n'est pas une extension de texte comme on peut le faire en C l'aide du précompilateur Si la compilation d'une unité générique (sous-programme ou paquetage) est correcte, les instantiations spécifiques le seront également. Là encore, la notion de contrat est sous-jacente.

L'instanciation explicite évite la confusion entre les paramètres génériques et les paramètres d'un sous-programme.

```

procedure ECHANGER is
  new G_ECHANGER (T_INCONNU => T_ENTIER);

procedure ECHANGER is
  new G_ECHANGER (T_INCONNU => T_TAB);

```

Il est ainsi possible de faire fabriquer par l'étudiant un ensemble de composants réutilisables. On peut instancier une procédure ECHANGER sur le modèle G_ECHANGER pour tout type T qui a la propriété suivante: T_INCONNU est un type à affectation. Cependant comme l'indique Bertrand Meyer [6], la généricité n'est pas une alternative à l'héritage. L'utilisation progressive du paquetage TEXT_IO familiarise l'étudiant avec l'instanciation générique et la réutilisation systématique d'un composant logiciel bien conçu.

2.8 - Les Tâches

Les tâches permettent d'introduire le parallélisme fondé sur le concept du RENDEZ-VOUS.

2.9 Aspects Spécifiques du langage Ada 95

Parmi les nouveautés apportées par la norme 95, les types étiquetés permettent de faire du langage Ada un langage objet. Il me semble qu'il est tout à fait possible d'introduire la notion d'héritage de polymorphisme sans introduire beaucoup d'éléments nouveaux dans le langage.

Exemple

```
package EXEMPLE is
  type C_PERE is tagged
    record
      ...
    end record;

  procedure FAIRE (A : in out C_PERE);
- Dérivation
  type C_FILS is new C_PEERE with
    record
      nouveaux champs
    end record
end EXEMPLE;
```

```
Package body EXEMPLE is
  procedure FAIRE (A: in out C_PERE) is
  begin
  ....
end FAIRE;
```

```
procedure FAIRE (A : in out C_FILS) is
begin
...
  FAIRE (C_PERE (A)); -- -- appel de FAIRE avec conversion de type
...
end AIRE;
end EXEMPLE;
```

Dans cet exemple le type (la classe) C_FILS hérite du type (la classe) C_PERE;.

Polymorphisme

Le polymorphisme est rendu possible par le nouvel attribut CLASS qui indique dans la procédure TAITER ci-dessous que celle-ci peut être appelée par n'importe quel objet de la classe C_PERE ou d'une classe dérivée de C_PERE.

```

procedure TRAITER (A : out C_PERE ' CLASS ) is
begin
  FAIRE (A); -- -- liaison dynamique à ce niveau
end FAIRE;

```

La liaison dynamique apparaît à l'appel de la procédure FAIRE. La « bonne » version de FAIRE sera choisie au cours de l'exécution selon que A sera de la classe C_PERE ou de la classe C_FILS. On voit ainsi, que la version Ada95 permet à partir d'une faible extension du langage (tagged, CLASS) d'introduire l'héritage et le polymorphisme.

3 - L'enseignement de C++ fondé sur la connaissance de Ada

L'enseignement du C++ doit être précédé d'une présentation générale du langage C. Cette présentation met en évidence les points communs, mais surtout les différences entre les 2 langages. Elle permet également de montrer le « comment » d'un passage de paramètre dans une procédure, et d'expliquer la nécessité du passage par adresse ou par référence.

3.1 Principes dégagés par l'apprentissage de Ada

La richesse des types conduit à ne numériser que lorsqu'on ne peut pas faire autrement.

Si on numérise, on doit préférer un type "personnel" adapté au problème à un type prédéfini dépendant de la machine.

Ces 2 remarques associées au fait que le langage Ada est très fortement typé élimine un certain nombre d'erreurs qui dans d'autres langages ne seraient détectées qu'à l'exécution.

Dans une procédure ou une fonction, les données sont « élaborées » avant le début des actions, ce qui conduit à préciser la notion de *constructeur*.

La partie privée d'un paquetage permet d'encapsuler des données ou des types. Pour un paquetage, une fonction, comme pour une procédure, la séparation de la spécification et du corps permet de distinguer le comportement de la réalisation.

On définit des types, de procédures, des fonctions pour obtenir un comportement donné.

L'utilisation des exceptions permet de préciser le comportement du logiciel dans les cas Limites, elle apprend à l'étudiant à avoir une *vision contractuelle* du développement du logiciel. L'utilisation de la généricité conduit à un effort de synthèse et habitue l'étudiant « penser à la réutilisation possible » de ce qu'il fait.

3.2 - La notion de classe

Elle est introduite en étudiant la différence de comportement entre les entiers en Ada et les entiers en C. Déclarer un entier en C, c'est réserver un certain espace mémoire dans lequel on pourra mettre un entier mais également d'autres choses. Pour hisser C++ au niveau Ada, il faut construire une classe C_ENTIER. C'est un excellent exercice que de demander aux étudiants de concevoir une classe C_ENTIER ayant le même type de comportement que la « classe entier » du langage Ada. Cet exemple très simple permet d'illustrer un certain nombre d'éléments de la programmation objet. On aboutit en général à une classe qui a l'allure suivante;

```
class ENTIER
[
private:
int DEBUT;
int FIN;
int VALEUR;
public
ENTIER (int = - 32768, int = 32767, int = 0); //constructeur
int FIRST (void) const;
int LAST (void) const;
ENTIER operator + (ENTIER) const;
...
]; //end class ENTIER
```

Ainsi, un objet ENTIER pourra se déclarer de la façon suivante:

```
ENTIER E (10, 20, 22);
```

On peut remarquer également à ce propos la similitude d'écriture entre par exemple

```
cout << E. FIRST ();
et
PUT (E' FIRST; );
```

Constructeur

La notion de constructeur peut être vue comme l'élaboration d'un objet E. Celle ci doit engendrer un objet correct, d'où la nécessité d'introduire une exception.

```
ENTIER :: ENTIER (int DEB, int FIN, int VAL)
: DEBUT (DEB), FIN (FIN)
{
if ( (VAL >= DEBUT) && (VAL <= FIN) )
VALEUR = VAL;
else
```

```

cout << " CONSTRAINT_ERROR: FICHER = "
      << _FILE_ << " LIGNE: " << _LINE_ ;
//end if '
} //end ENTIER

```

Dans cet exemple, on « simule » la levée d'exception par un message d'erreur.

constructeur par copie

Le « type » array en Ada fournit un autre exemple de comportement objet. Lorsqu'on déclare un tableau, on a un certain nombre de garanties. Le type tableau en C n'est, comme pour une variable, qu'une simple réservation d'espace sans, ce qui est plus grave aucune, concernant les débordements. Concevoir une classe TABLEAU rendant des services de même nature que ceux rendus par array en Ada est aussi un exercice très formateur qui permet d'illustrer en particulier la nécessité de définir un *constructeur* par copie et de prévoir un *destructeur*.

```

class TABLEAU
{
private:
int DEBUT;
int FIN;
int LONGUEUR,
int* P;
public
TABLEAU ( int, int );
TABLEAU (const TABLEAU &); //constructeur par copie
int FIRST (void) const // attributs
int LAST (void) const; // de la classe
int LENGTH (void) const; // tableau
int operator [ ](int); // surcharge de l'opérateur[ ]

-TABLEAU // destructeur
]; // end class TABLEAU

```

Si on ne dispose pas d'une version de C++ incluant les exceptions il est possible de «simuler» par un affichage à l'écran de la violation du contrat.

```

int& TABLEAU::operator [ ] ( int i )

if ( ( i >= DEBUT) && ( i <= FIN) )
return P [i - DEBUT];
//end if
cout C<<<< " CONSTRAINT_ERROR << _FILE_ << _LINE_ ;
//end operator [ ]

```

3.3 - La généricité

La généricité existe en langage C, elle est mise en oeuvre par template. Là encore, il faut bien remarquer que la syntaxe proposée n'est pas des plus agréable. Il n'est pour s'en convaincre que de comparer la réalisation d'une fonction générique G_ECHANGER. En C++ cela donne:

```
<template class INCONNUE
void ECHANGER (INCONNUE& L_UN, INCONNUE& L_AUTRE)
{
INCONNUE AUX;
AUX = L_UN; L_UN = L_AUTRE; L_AUTRE = AUX;
} // end ECHANGER
```

Comme le montre l'exemple suivant, il n'y a pas d'instanciation explicite

```
void main (old)
{
int A = 5, B = 7
struct DATE
{
...
} // end struct
DATE D1 = {12,1,1977};
DATE D2 = {20,6,1970}
char S1 [] = " OUI ";
char S2 [] = " NON ";

ECHANGER (A, B); //correct
ECHANGER (D1,D2); // correct
ECHANGER ( S1,S2 ); // erroné
} //end main
```

Là encore le « détour » par le langage Ada n'est pas un handicap bien au contraire. De plus comme en C++, il n'existe pas de classification des types, on n'a pas toujours la certitude d'obtenir une fonction correcte à partir de l'utilisation d'un template.

Une autre illustration intéressante de template est de rendre générique la classe 'TABLEAU, faisant ainsi de cette classe une classe comparable au « type » array de Ada.

```
template <class TAB_GEN>
class TABLEAU
{
private:
int LONGUEUR;
```

```
int DEBUT;
int FIN;
TAB_GEN* P;
public
} //end class TABLEAU
```

On peut ainsi déclarer des tableaux de tout type:

```
TABLEAU<int> T (1,100);
TABLEAU<DATE> TD (3,10);
TABLEAU<char> TC (25, 50)
TABLEAU<int >* TX = new TABLEAU<int>(5,14);
```

3.4 L'héritage

La généralité a familiarisé l'étudiant avec la notion de réutilisation du logiciel. L'héritage doit être abordé comme réutilisation avec personnalisation. Bien entendu, cette notion doit être entièrement définie si on ne dispose pas d'un compilateur Ada 95. La programmation sous «Windows» habitue rapidement les étudiants à l'héritage. La notion délicate est la notion de polymorphisme car elle va (en apparence) à l'encontre du typage fort.

Exemple

```
class PERE
class FILS : public PERE;
PERE* E;
FILS* F;
E = F; //violation apparente de la règle du fort typage,
//    contrôle différé
```

4 - Conclusion

Commencer l'enseignement des bases de la programmation par Ada fait que lorsqu'on aborde C et C++, l'étudiant est déjà familiarisé avec les principaux concepts de la programmation objet. Or ce qui est la source de bien des difficultés en C++, c'est en particulier le fait que les « anomalies » syntaxiques occultent souvent les concepts que l'on veut dégager. En commençant avec Ada, il me semble qu'on évite l'écueil. En outre son utilisation familiarise l'étudiant avec un certain nombre d'outils de *génie logiciel*.

Jean VICARD

Maître de conférences

IUT de VILLETANEUSE Université PARIS XIII
 Adresse: 74, Parc de Maugarny 95680 MONTLIGNON
 tél. : 34.16.35.68

5 Bibliographie

- 1 - **John Barnes**. Programmer en Ada, Inter Editions, Paris 1995
- 2 - **Jean-Paul Bodeveix, Mamoun Filali, Arnal Sayah** . Programmation en C++, Inter Editions, Paris 1994
- 3 - **Grady Booch**. Ingénierie du logiciel avec Ada, Inter Editions, Paris 1991
- 4 - **M. Gauthier**. Ada: un apprentissage. Dunod Paris 1989
- 5 - **Narain Gehani**. Ada Introduction avancée, Eyrolles, Paris 1988
- 6 - **Bertrand Meyer**. Conception et programmation par objets, Inter Editions, Paris 1990
- 7 - **Jean-Pierre Rosen**. Méthodes de génie logiciel avec Ada 95, Inter Editions, Paris 1995
- 8 - **B. Stroustrup**. Le langage C++, Addison-Wesley, 1992

L'enseignement de la conception logicielle

Patrice RICHARD

présentée le jeudi 11 avril 1996 à 10h30

1. Introduction

De nombreuses années d'expériences d'enseignement et de développement de logiciels de grandes tailles destinées à l'industrie nous ont conduit au fil des ans à changer nos objectifs dans le domaine de la formation .

Nos premiers balbutiements dans l'enseignement de la programmation, il y a plus de vingt ans, se limitaient à faire apprendre à IIOS étudiants la syntaxe d'un langage. Nous étions satisfaits de leur travail quand ils nous remettaient un programme qui tourne avec quelques vagues explications sur ses fonctionnalités et surtout, nous y tenions beaucoup, un bel organigramme.

Cette attitude découlait naturellement de la formation que nous avions reçue.

C'était l'époque des querelles de chapelles, chacune défendant âprement les qualités de son langage favori. Ces qualités se réduisaient souvent aux manipulations plus ou moins "bidouillesque" que le langage permettait de réaliser.

Notons cependant qu'à notre décharge, nous nous contentions des moyens matériels disponibles à l'époque: des cartes ou rubans perforés, quelquefois des bandes magnétiques et des mémoires centrales plus souvent avec 8 Koctets que de 64 Koctets.

Quand nous racontons cela à nos élèves cela les fait discrètement sourire et pourtant, quand nos programmes finissaient par tourner, nous étions heureux. Ce qui ne nous empêchait pas de pester quand il fallait continuer ou améliorer les travaux d'un autre chercheur avec pour seul document une belle suite d'instructions.

Une belle suite d'instructions est encore, malheureusement, le seul résultat produit par de nombreux programmes.

Nous allons dans la suite de cet article aborder successivement les différentes typologies comportementales vis à vis de la programmation, un rapide résultat de la vision industrielle de la conception de logiciels pour terminer par quelques solutions expérimentées destinées à faire changer les mentalités.

2. Les différentes approches humaines

Les différentes approches présentées ci-après sont stéréotypées. Il est bien évident que, dans la réalité, elles ne sont pas aussi tranchées.

2.1 Le professionnel

Il travaille avec l'objectif de concevoir un produit logiciel. Le développement d'un tel produit a toujours pour cadre un projet, que cela soit explicite ou non. Un projet se caractérise par: un client (interne ou externe), un temps prévu(souvent sous-estimé), un planning (découpage des différentes phases du projet dans le temps), une équipe de conception et de réalisation (d'une personne à plusieurs dizaines, voire centaines de personnes) et un coût estimé en Homme/Mois.

Pour mener à bien un projet logiciel d'une certaine envergure, 18 mois par exemple, il est presque inconcevable de nos jours de ne pas utiliser une méthode de conception.

A cet environnement de travail se surajoute la qualité. Nous développerons plus loin la différence existante entre une méthode qualité et un état d'esprit qualité, le Graal recherché étant le zéro défaut.

2.2 Le chercheur

Il a pour principal objectif de publier, nécessité imposée par le déroulement des carrières universitaires.

Pour publier, il doit souvent écrire des programmes pour valider ses idées scientifiques. Cette activité ne se limite pas qu'aux chercheurs en informatique, loin s'en faut.

Il est rare que cette production logicielle soit associée à une vision projet. Si le temps est connu, trois ans pour une thèse, le planning est flou, s'il existe ! Le client est rarement identifié (est-ce le directeur de thèse, un industriel ou le thésard lui-même ?), le coût en temps n'est pas évalué (il est bien connu que les week-ends et les nuits sont faites pour travailler) .

La méthode la plus souvent employée est la méthode, dite naturelle. Elle consiste à considérer comme une évidence l'écriture de logiciels, capacité innée chez tout individu.

Quand il a le temps, il rédige la documentation associée au logiciel, après que celui-ci soit terminé.

Quant à la qualité logicielle, elle se limite à vérifier que le logiciel fait bien ce pourquoi il a été écrit.

Cette vision caricaturale, un peu ironique, recouvre des faits réels.

Ce n'est pas la faute du chercheur. On lui demande de faire un travail de recherche, non de concevoir un produit.

Ceci est partiellement faux, car de grands projets de recherches peuvent engendrer plusieurs thèses parallèles ou successives. Dans ce dernier cas, un logiciel peut avoir plusieurs écrivains. Nous avons un cas similaire quand le chercheur se fait aider par des étudiants à qui il confie une partie de la réalisation.

En fait, il se différencie profondément du professionnel car il ne soucie peu de la maintenance du logiciel qu'il a produit. Ce qui nous fait dire un peu brutalement qu'une bonne partie des logiciels produits dans le cadre universitaire sont destinés à aller à la poubelle. Cette situation représente un gâchis monstrueux du travail fourni par des dizaines de milliers de chercheurs de par le monde.

2.3 L'élève ou l'étudiant

Sa problématique personnelle se différencie nettement des deux autres profils précédents: Il n'a pas d'objectif défini, simplement un passé expérimental .

Deux situations extrêmes se rencontrent quotidiennement, pour formateur:

Le pur débutant devient de plus en plus rare, les jeux étant une excellente initiation à l'approche d'un système informatique, tant à la fois d'un point de vue manipulateur que d'un point de vue structurel ou organisationnel.

L'expérimenté qui, à titre personnel, le plus souvent, ou comme produit du système éducatif, se considère et est considéré comme tel par ses condisciples, un peu moins initiés, comme un spécialiste (terme pudique pour qualifier un "peut-être" vrai technicien du domaine).

Cette dernière situation conduit le formateur à une perte relative de la maîtrise pédagogique et temporelle de la formation qu'il a à assurer. Quand son programme tourne, l'élève est satisfait par ce qu'il a contenté, du moins le croit-il, son professeur: "Vous me demandez un programme qui fonctionne, voilà, c'est fait!"

3. La conception de logiciels et le développement de programmes

3.1 Le cycle de vie du logiciel

Il peut se résumer par le schéma suivant:

3. La conception de logiciels et le développement de programmes

3.1 Le cycle de vie du logiciel

Il peut se résumer par le schéma suivant :

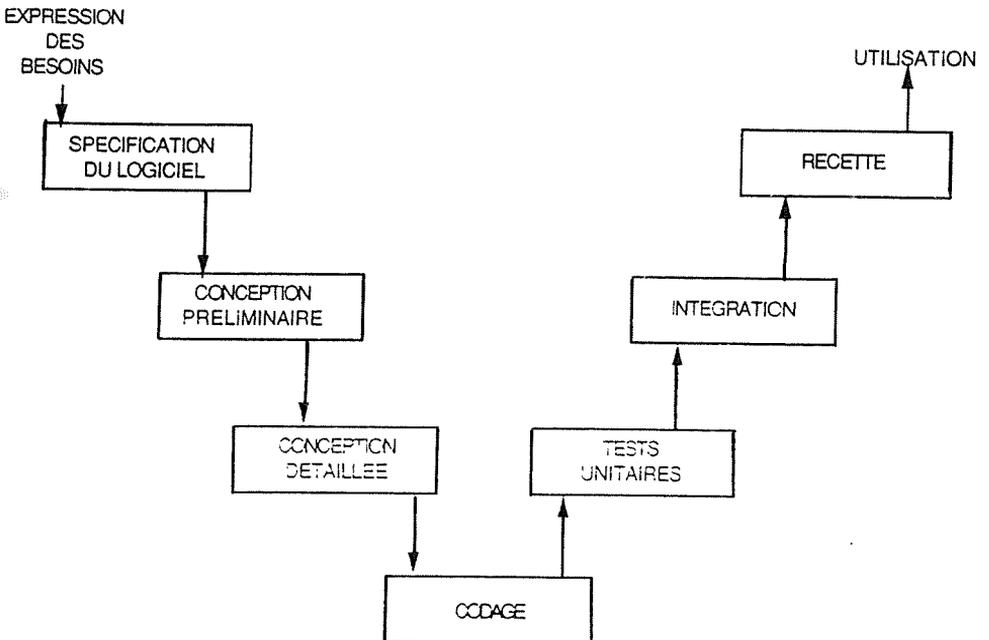


FIGURE 1

La phase de spécification permet de décrire les fonctionnalités prévues du logiciel.

La conception préliminaire décrit les articulations entre les différents modules ainsi que l'objectif de chacun.

La conception détaillée consiste à préciser, pour chaque module, les fonctionnalités internes en utilisant une méthode de programmation.

Ce schéma classique montre bien que le codage, même si c'est une phase importante ne vient qu'après que le futur logiciel soit totalement décrit dans ses fonctionnalités et son architecture.

Une fois que chaque module est écrit, il y a une phase de test spécifique au module, ce sont les tests unitaires.

La phase d'intégration vérifie l'interconnexion des différents modules et le bon fonctionnement global du logiciel.

La phase de recette est généralement effectuée par l'entité client pour accepter le logiciel avant sa mise en exploitation réelle.

C'est à ce moment que commence la vie du logiciel qui entre dans la phase de maintenance. C'est surtout à partir de ce moment que la qualité avec laquelle a été écrit le logiciel facilite ou non les évolutions du produit logiciel ainsi que le débogage éventuel.

3.2 La documentation associée au logiciel

Chacune des phases précédentes engendre la production de documents. Mais il y a aussi d'autres documents qui concernent l'ensemble des phases.

3.2.1 La documentation du produit logiciel

Elle se compose des documents suivants

Les spécifications externes

Ce document regroupe une description globale des fonctionnalités attendues et les contraintes imposées par l'environnement de réalisation tant du point de vue matériel que du point de vue logiciel (langage utilisé, machine cible, etc.).

L'architecture

Elle comporte deux aspects, l'un fonctionnel, l'autre physique.

La partie fonctionnelle décrit l'articulation entre les différents modules en identifiant pour chacun les informations échangées.

La partie qualifiée de physique regroupe un descriptif précis des interfaces entre les modules prévus dans l'architecture fonctionnelle.

Le manuel utilisateur

Il contient la description de toutes les commandes disponibles dans le logiciel et les effets du paramétrage quand il existe.

La documentation des modules

Elle contient, outre le listage du module logiciel, sa propre spécification externe, sa propre architecture, la liste des tests prévus pour valider le module et le cahier de suivi où seront notées toutes les modifications que subira le module au cours de son existence.

3.2.2 Les documents associés au développement du logicielLe plan de développement

Il expose de manière précise la méthode et les moyens mis en oeuvre pour le développement du logiciel.

Le dossier de conduite du projet

Il contient principalement le planning prévu et les différentes versions de celui-ci.

Le dossier qualité

Ce dossier regroupe les méthodes et moyens pour s'assurer de la qualité du logiciel en fonction des différents critères retenus.

3.3 La temporalité du développement et de la rédaction des documents

Si certains s'enchaînent logiquement, spécification et architecture par exemple, il est moins évident de commencer à rédiger le manuel utilisateur du moins une ébauche, dès le début du projet et non en phase finale .

Le dossier de recette s'écrira lui aussi en même temps que les spécifications.

3.4 La qualité logicielle

Il existe des méthodes pour l'appliquer, mais il ne faut pas exagérer l'impact d'une méthode si celui qui l'applique n'a pas lui-même un esprit qualité.

4 L'univers pédagogique**4.1 Pour l'étudiant****4.1.1 L'expérience antérieure**

L'introduction dans l'environnement actuel de l'étudiant, qu'il soit familial ou scolaire du micro-ordinateur lui donne un vernis ou de réelles connaissances en programmation. Il n'est pas rare de rencontrer des élèves connaissant déjà un langage de programmation. L'enseignant se trouve confronté à une situation courante en formation continue: l'hétérogénéité du public.

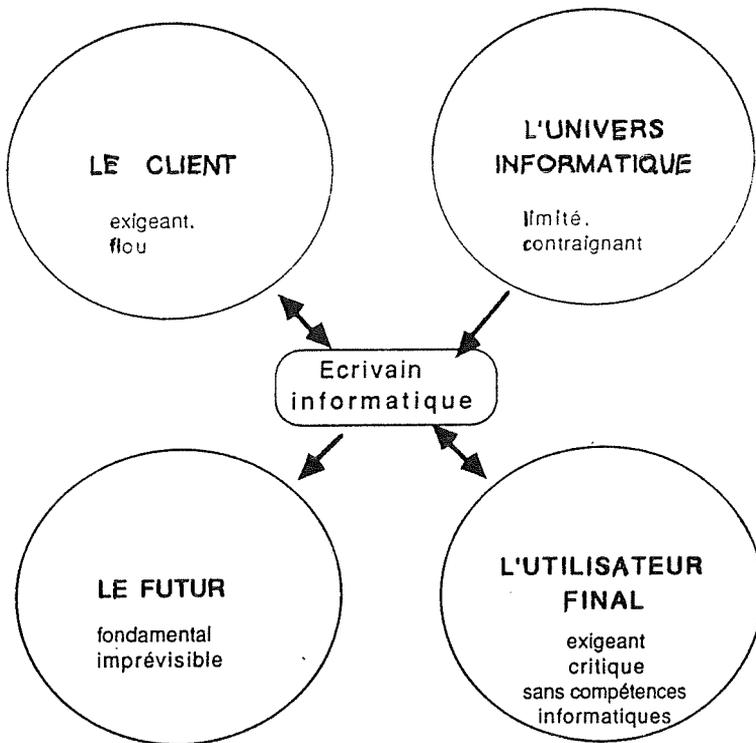
Cependant, il est très rare que les élèves aient eu, au cours de leurs études, une formation quelconque à une méthode.

Quant à la notion de maintenance et de qualité logicielle, ces mots sont presque inconnus.

4.1.2 Les objectifs pédagogiques

Pour remplacer le cours de langage par un cours de conception de logiciel, il faut préciser à l'élève qui n'a que peu d'expérience de la réalité industrielle, les univers au centre du quel se situe l'activité de programmation .

Pour notre part nous utilisons le schéma suivant:



Dans ce schéma apparaît la notion d'écrivain informatique. Cette notion nous semble importante, car l'objectif d'un écrivain est d'être lu et si possible compris.

L'introduction de cette volonté d'être lu et compris par un autre que soit même change les objectifs de la programmation qui ne se limite plus alors au simple alignement d'ordres destinés à une machine dans un langage choisi, mais à la conception de logiciels, même de petite taille.

De ce fait l'enseignant peut exiger des identificateurs significatifs et des commentaires, eux aussi, significatifs. Cette exigence est parfaitement acceptée par le public quand on arrive à les convaincre que, même dans le cadre scolaire, quelqu'un d'autre peut reprendre leur programme. Cette situation est symbolisée dans le schéma précédent par l'univers FUTUR.

La qualité des messages, destinés à l'utilisateur, appelée professionnellement l'interface utilisateur, nécessite des messages clairs et concis. Cette notion est symbolisée par le pavé UTILISATEUR.

Le pavé CLIENT symbolise que le programmeur doit traiter le sujet donné, ni plus, ni moins.

Quant au dernier pavé, l'univers INFORMATIQUE, il rassemble les contraintes qu'impose le langage de programmation, l'univers de travail...

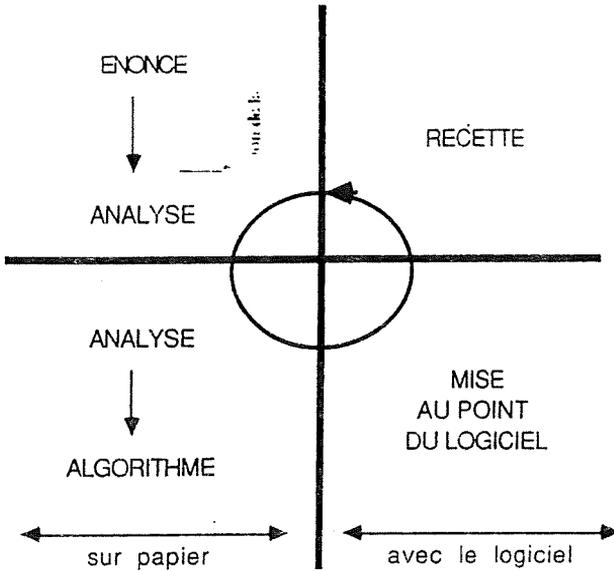
4.1.3 Les moyens de mise en œuvre

Pour un élève, il serait, du moins au début, de vouloir appliquer la totalité des composantes d'un projet logiciel, méthode et documentation.

Cependant, il nous semble formateur de ne pas se limiter qu'au listage. Même un petit programme devrait outre la suite d'instructions comporter une spécification, et la liste des tests. Ces documents sont rédigés avant toute écriture d'une ligne de code.

Plus l'élève progresse dans sa formation plus il se rapproche Des conditions réelles du développement de logiciel.

Cependant, il est fondamental que, dès le début de la formation, un très fort accent soit mis sur la qualité interne des listages. Pour convaincre les élèves que ce que l'on écrit n'est pas toujours facilement compréhensible par un autre, il est possible de recourir à la mise en situation connue sous le nom de 4x4 représentée par le schéma suivant:



Cette figure schématise les quatre phases du déroulement de la conception d'un programme.

Après avoir partagé les élèves par groupes de trois, on donne à chaque équipe un énoncé de problème différent. Chaque groupe dispose environ de deux heures pour effectuer la première phase, l'analyse du problème posé. Un document écrit est remis à la fin de la phase.

L'enseignant donne à une autre équipe le document remis et cette nouvelle équipe devra continuer le travail, c'est-à-dire d'écrire un algorithme à partir de l'analyse fournie.

La suite s' imagine sans peine, une troisième équipe mettra au point l'algorithme fourni par écrit par la deuxième équipe.

Par expérience, il est bien rare que l'on puisse arriver à la dernière phase, celle de la recette.

Cette mise en situation, même si elle prend une partie du temps disponible de l'enseignement, évite au professeur de perdre le sien à répéter inlassablement les mêmes exigences de qualité du listage.

4.1.4 Le contrôle

Pour renforcer les exigences de qualité et convaincre qu'il est préférable d'avoir un programme qui ne tourne pas bien écrit plutôt qu'un programme illisible qui tourne, nous avons mis au point un barème de notation évolutif. Lors des premiers contrôles, La notation comporte deux volets. Le premier concerne l'algorithme (ou le programme) proprement dit, dix points pour sa qualité de fonctionnellement. Les autres dix points sont a priori acquis à l'élève à condition que son travail satisfasse aux critères suivants:

L'entête du listage doit être complet (Nom, Sujet, Date).

Le logiciel doit afficher un titre à l'utilisateur.

Les commentaires doivent être significatifs.

Les identificateurs doivent être parlants, pas d'identificateurs comportant une lettre sauf exception acceptable.

L'interface utilisateur doit être de bonne qualité, en particulier en distinguant les cas du fonctionnellement normal des erreurs produites par l'utilisateur.

Toutes les données introduites par l'utilisateur doivent être vérifiées pour un éventuel rejet.

La présentation du listage doit être agréable à lire en particulier être bien indenté pour mettre en valeur l'imbrication des différentes parties le composant.

Les tests présentés doivent être complets. Le dossier présenté doit contenir une page de garde.

Le dossier présenté doit contenir une spécification, même sommaire au début de la formation.

Si tous ces points sont scrupuleusement respectés, même ceux qui sont subjectifs, comme la qualité des commentaires, l'élève a dix points. Sinon pour chacun des éléments de la liste donnée, il perd deux points.

Cela permet à un élève dont le programme est complètement faux d'avoir la moyenne et, à contrario, à un élève ayant un programme correct, moins de la moyenne.

Plus la formation se déroule plus l'importance donnée au programme proprement dit augmente, mais avec toujours la possibilité de perdre des points si la qualité du travail laisse à désirer.

4.2 Pour le chercheur

4.2.1 L'expérience antérieure

Elle est fonction de son passé universitaire. Cela peut aller du parfait débutant, cas heureusement de plus en plus rare, à un programmeur confirmé.

Mais, il est exceptionnel de trouver des chercheurs ayant une culture projet et qui, en développant le logiciel nécessaire à leurs travaux, ont pour objectif complémentaire de penser que celui-ci pourra servir à d'autres chercheurs.

4.2.2 Les objectifs pédagogiques

Pour gagner en efficacité et éviter que le chercheur se polarise sur la mise au point de ses programmes, il faut impérativement l'aider par une formation, car l'informatique n'est pas obligatoirement son principal but.

La principale action que nous avons mis en pratique dans notre laboratoire était de réunir des jeunes chercheurs et d'estimer avec eux leurs compétences. Les compétences minimales requises sont la connaissance du ou des langages prévus, la connaissance d'une méthode analyse pour concevoir les modalités du futur logiciel, la connaissance d'une méthode de conception logiciel.

4.2.3 Les moyens de mise en oeuvre

Dans le cas d'une insuffisance dans l'un des domaines, nous avons toujours essayé de trouver une formation adéquate en interne si possible pour des questions de coût. Dans le pire des cas, nous lui procurons un ou plusieurs ouvrages de référence et le nom d'une personne expérimentée auprès de laquelle il peut trouver des aides. Cette apparente perte de temps s'est montrée bénéfique par la suite.

Si le chercheur désire se faire aider d'étudiants, il doit posséder aussi une certaine expérience de la conduite du projet. Cette lacune est plus difficile à combler.

Pour minimiser le temps du formateur et d'éviter de ne former qu'une ou deux personnes, nous proposons souvent à d'autres laboratoires concernés par la même problématique de nous envoyer leurs jeunes chercheurs.

4.2.4 Le contrôle

Cette activité est à la charge du responsable du laboratoire ou de son délégué. Il suffit de rencontres régulières pour vérifier que les principes sont bien appliqués sans augurer de la validité des résultats scientifiques attendus. Un bon logiciel peut donner de faux résultats si les idées de départ le sont.

4.3 Pour le professionnel

4.3.1 L'expérience et les exigences professionnelles

Il est rare actuellement d'embaucher quelqu'un qui n'aurait pas de solides compétences dans le développement de logiciel. Cependant si tel était le cas, un stage suffit à combler le manque.

A notre avis, il y a une formation, souvent omise chez les jeunes embauchés, c'est celles touchant à la conduite du projet et surtout à la vision du rôle de chef de projet. Ceci est fondamental pour le projet afin de ne pas altérer la bonne marche de l'équipe en identifiant mal le rôle de chacun.

4.3.2 Le contrôle

C'est le rôle du chef de projet. Lors des réunions d'avancement, il vérifie l'application stricte des règles méthodologiques; c'est la clé de réussite de son projet.

5 Une conclusion basée sur l'expérience

Ayant eu à former les trois types de publics schématisés ci-dessus, bien qu'en adaptant la formation ou le mode de contrôle, nous sommes arrivés à ce qui pour nous est devenu une évidence. Il y a pourtant différence quant à l'exigence de qualité, même chez les apprentis programmeurs. Le mode de contrôle est directement lié à la population à former. La vision de « l'écrivain informatique » prime sur tout apprentissage ou utilisation d'un langage. Cette simple conclusion est devenue une nécessité économique et il serait souhaitable qu'elle soit présente à l'esprit de tout enseignant concerné par le domaine. Nous avons constaté par ailleurs que la rigueur exigée dans nos cours avaient des retombés très positives dans d'autres secteurs de la formation prise dans sa totalité.

Patrice RICHARD

E.I.V.L 6-8, Rue Anne de Bretagne 4100 BLOIS - France

Tél. :33 54572425 (direct) 33 54572420 (secrétariat)

33 54572421 (télécopie)

Thème5 : La formation continue

Formation des enseignants en informatique dans le secondaire: Ingrédients et recettes pour le fonctionnement d'un centre de formation

Etienne Vandeput

présentée le mercredi 10 avril 1996 à 14h45

Sous-titre: Histoire d'une formation construite en plusieurs étapes sur base d'une analyse de la situation de terrain

Résumé: *En Belgique francophone, depuis quatre ans, trois organismes du réseau d'enseignement confessionnelⁱ concernés par la formation des enseignants, le CeFIS, ICAFOC et la FPEⁱⁱ tentent d'unir leurs forces dans une même direction: influencer favorablement l'évolution des programmes de cours, la formation en informatique et la didactique des enseignants du secondaire. De part leurs fonctions et leurs milieux respectifs de travail, les membres de cette union (professeurs d'université et du secondaire, inspecteurs de l'enseignement secondaire, responsables de formations) sont en contact avec un nombre très important d'enseignants. Pour être franc, il doit exister très peu d'enseignants du réseau, concernés de près ou de loin par l'informatique, qui n'aient eu le moindre contact avec l'un des organismes précités. Cette remarque n'est pas faite pour témoigner d'une situation de monopole, mais pour garantir l'expérience non négligeable et de la source d'évaluation solide du travail accompli qui en résultent. Au travers de cette communication nous essaierons de dégager et de transférer un maximum d'enseignements de cette expérience synergétique qui se poursuit, espérant qu'ils puissent trouver un écho à d'autres endroits, même si les ressources matérielles et humaines sont différentes, même si la culture est différente. Précisons donc que le contexte local de cette expérience a peu d'importance. Il est cependant difficile de l'éviter pour pouvoir justifier la chronologie des choses. Ce texte est donc un compromis entre le récit qui le cadence, l'analyse qui en est la substance et la citation qui, parfois mieux qu'une longue explication, en illustre le propos.*

Les résultats présentés dans cette communication sont à mettre en rapport avec la description du paysage informatique belge (Où en est-on?) et les perspectives d'avenir (Où va-t-on?) présentées lors de la Troisième Rencontre

ⁱ un des trois réseaux d'enseignement (voir tableau qui suit)

ⁱⁱ CeFIS: Centre pour la Formation à l'Informatique dans le Secondaire (dépendant du Département Education et technologies des Facultés Universitaires de Namur).

ICAFOC: Institut Catholique pour la Formation Continue.

FPE: Formation Permanentes des Enseignants.

Francophone de Didactique de l'Informatique à Sion en 1992 par Ch. Duchâteau et F. Sassⁱⁱⁱ.

1. Description du paysage

1. Toutes disciplines confondues

La structure de l'enseignement en Belgique n'est pas simple. Le tableau suivant permet un raccourci. Il s'inspire d'un tableau plus complet que l'on retrouvera dans [1]. En Belgique francophone^{iv}, on distingue trois réseaux d'enseignement: celui de la Communauté Française, celui des provinces et communes et le réseau confessionnel. L'organisation de l'enseignement de la Communauté (environ 50% des établissements) est centralisée, il n'en est pas de même dans l'enseignement libre (confessionnel). Dans ce réseau, l'organisation n'est pas du ressort d'un seul pouvoir organisateur, mais d'une multitude de pouvoirs organisateurs locaux. En ce qui concerne les provinces et communes, ce sont ces entités respectives qui organisent l'enseignement. On comprend pourquoi une politique générale pédagogique et de choix d'équipement est toujours difficile à mener. L'uniformité n'existe (et encore!) qu'au niveau des subsides, car la Communauté prend en charge le traitement des enseignants et la distribution d'un subside par élève. Elle existe aussi au niveau de la dénomination de grilles de cours qui peuvent cependant cacher des contenus, des méthodologies, des niveaux de formation très différents.

L'enseignement en Belgique			
Néerlandophones	Francophones et germanophones		
	Les trois réseaux		
	Communauté Française	Communes et Provinces	Confessionnel (catholiques, autres)
			Secondaire: FESeC GRIP

ⁱⁱⁱ C. DUCHATEAU, F. SASS En Belgique, où en est-on? où va-t-on? Actes de la Troisième Rencontre Francophone de Didactique de l'Informatique (Sion, du 6 au 11 juillet 1992) EPI Paris 1993.

^{iv} Jaloux des Suisses, nous avons aussi une Belgique néerlandophone et une Belgique germanophone

On pourrait croire que l'organisation de l'enseignement est tout à fait morcelée dans le réseau confessionnel. La réalité est un peu différente. Un organisme, la FESeC^v, assure des services communs à différents pouvoirs organisateurs (inspection, rédaction de programmes, formations,...) Une des émanations de la FESeC est le GRIP^{vi}, un groupe de travail qui a permis à différents responsables de l'enseignement de l'informatique de se rencontrer, puis de s'associer dans un but commun. Nous en reparlerons.

Retenons essentiellement de ce qui précède que, traditionnellement, les réflexions didactiques et pédagogiques doivent tenir compte de conditions de terrain extrêmement différentes. Ajoutons à cela qu'au cours de la dernière décennie, le recentrage sur l'élève des préoccupations éducatives est manifeste. Il en résulte, dans de nombreux programmes de cours, un déplacement progressif de l'intérêt pour des contenus de matières précis et détaillés vers les méthodes et les attitudes à développer chez les élèves, quelles que soient les conditions de travail auquel il est confronté. Cela ne signifie pas que les contenus disparaissent des programmes, mais leur description se limite souvent à l'énumération des concepts vraiment incontournables dans le domaine concerné. Un minimum de "savoir" essentiel doit être transmis, mais l'accent est mis sur le développement des "savoir-faire" (gestion des ressources extérieures disponibles) et des "savoir-être" (gestion de ses propres ressources). Cette façon de voir les choses n'est pas forcément en contradiction avec les objectifs d'un enseignement qui doit "permettre à tous les jeunes de comprendre et transformer le monde"^{vii} qui les entoure, monde qui cultive la diversité et la multiplicité à tous les niveaux, diversité et multiplicité qu'il faut aujourd'hui apprendre à gérer.

1. En ce qui concerne l'informatique et ses programmes...

Le cours d'informatique est apparu dans l'enseignement secondaire (12 à 18 ans) au milieu des années 70. Il n'a jamais eu, jusqu'à ce jour, le statut de cours obligatoire. Il peut être organisé à n'importe quel niveau, au gré de l'établissement. L'informatique s'intègre dans des activités aussi différentes que des cours de programmation, des cours de traitement de texte, de gestion, de dessin technique, de comptabilité...

Cette situation n'est pas de nature à faciliter l'écriture d'un programme de cours. Pourtant, suite à une enquête fouillée réalisée dans un grand nombre d'écoles, la FESeC (à l'époque FNESeC, car encore "Nationale") confie à un groupe de spécialistes de la didactique de l'informatique (le GRIP déjà cité) l'édition d'un nouveau programme. Ce programme, publié en mai 1992, est présenté la même année à Sion au colloque de l'AFDI. Intitulé "Informatique, un atelier pour tous", il est d'abord pensé comme nouvelle mouture d'un programme qui faisait la part très (trop) belle à la programmation, tenant compte des nombreuses remarques des enseignants quant à la difficulté de l'enseigner à un nombre croissant d'élèves

^v La FESeC est la Fédération de l'Enseignement Secondaire Catholique.

^{vi} Le GRIP est le Groupe de Réflexion sur l'Informatique et la Pédagogie.

^{vii} P. MEIRIEU Enseigner, scénario pour un métier nouveau ESF Paris 1989.

intéressés par l'informatique. Son souci est donc de tenir compte des capacités cognitives d'un maximum d'apprenants (pour tous). Mais il n'est pas question de négliger tout ce qui fait l'essence de l'informatique et qui n'a pas disparu, même des environnements informatisés les plus sophistiqués. Il s'agit de placer les élèves en situation de résoudre des problèmes (atelier). Le contexte de résolution de problèmes permet à la fois de s'intéresser à la manière de résoudre et d'observer, voire de corriger, les attitudes et les comportements. Ce programme est très adaptable, ...trop diront certains.

Compte tenu de l'esprit dans lequel il est rédigé, il concerne le cours d'activité complémentaire "Informatique" mais se veut un programme d'inspiration pour les autres cours où l'informatique des utilisateurs fait une apparition plus ou moins importante. Ainsi, l'enseignant chargé d'un cours de traitement de texte peut y puiser des idées de démarches qu'il risque de ne pas retrouver dans un programme plutôt lié à la dactylographie. De même en ce qui concerne des cours comme, comptabilité, dessin, etc.

Ce programme très ouvert apparaît rapidement comme le reflet d'un idéal que la formation initiale des enseignants ne permet pas d'atteindre. Trop peu pragmatique, il désoriente un certain nombre d'entre eux qui s'attendent, faute d'un recul suffisant, à y retrouver des indications très précises sur le ou les logiciels à employer, les commandes à décrire, le temps à consacrer à telle et telle activité. Cette attente est évidemment en porte-à-faux avec les objectifs qui ont été décrits d'un enseignement moderne.

Les auteurs du programme ont prévu cette réaction. Ils soupçonnent un désarroi des enseignants ^{viii} :

"Le programme sera complété par des fiches pédagogiques dont l'objectif sera d'aider les enseignants à préparer leur cours."

Ces fiches seront élaborées par des équipes d'enseignants en collaboration avec les inspecteurs d'informatique et des personnes ressources compétentes dans la didactique de l'informatique."

Cette promesse est renforcée [1]:

"Et maintenant... Il reste à conduire les efforts sur un triple front: celui d'une réflexion sur les traits essentiels de l'informatique dans une perspective d'apprentissage des éléments qui fonderaient une "culture en informatique"; celui aussi de la création de fiches didactiques proposant séquences pédagogiques, scénarios d'enseignement et occasions d'apprentissage; celui enfin de l'expérimentation dans les classes des propositions élaborées."

^{viii} GRIP Informatique, un atelier pour tous (Perspectives) FNESeC Bruxelles 1992

C'est à peu près à ce moment que naît une synergie dont l'action se prolonge aujourd'hui. Son histoire nous permet de tirer des enseignements tous azimuts dans le domaine de l'assistance pédagogique des professeurs d'informatique.

1. Histoire commentée de la mise en place d'une formation

Comment le GRIP, dont la mission première est la rédaction d'un programme, se transforme-t-il en association CeFIS - ICAFOC - FPE organisatrice d'une formation d'enseignants poursuivant à la fois des objectifs intellectuels et sociaux^{ix} ?

De la difficulté de définir une fiche pédagogique...

Le concept de fiche pédagogique existe dans l'esprit des personnes du GRIP. Leur donner une forme précise n'est pas très facile. L'essentiel du problème réside (une fois de plus) dans la formation des enseignants^x. C'est donc un problème très délicat. D'abord parce qu'une politique globale est impossible à mener, ensuite parce que les niveaux de connaissances des enseignants sont très différents et que l'évaluation personnelle qu'ils font de leurs capacités se réfère généralement à la connaissance superficielle qu'ils ont des produits de l'informatique.

Le problème de la formation des enseignants nous inspire quelques réflexions.

La première sous forme d'un exemple, ou plutôt d'une comparaison. Je connais très bien la Renault Twingo. Aucun de ses détails ne m'échappe et je la conduis très souvent. Suis-je, pour ces raisons, un bon moniteur d'auto-école, un enseignant efficace du code de la route ou encore, un excellent professeur de mécanique automobile?

A ses débuts, le cours d'informatique dans le secondaire accorde une large part à l'improvisation. Il bénéficie alors de l'effet de nouveauté et de la promotion sociale qu'il apporte à ceux qui le suivent. Ainsi, dans le début des années 80, les étudiants supportent bien qu'un cours d'informatique se donne dans un local où un seul ordinateur, muni d'un écran monochrome de basse résolution est disponible à la fréquence d'une fois tous les trimestres.

Un des moteurs de l'enseignement réside en effet dans ce qu'il apporte comme promotion par rapport à l'ensemble des membres de la société. Ce moteur est capable de supporter un carburant de moins bonne qualité, entendons par là un manque de pédagogie. Mais le côté promotion finit toujours par s'estomper et est

^{ix} Oui, sociaux, car plus que jamais et insensiblement, l'informatique, son matériel et ses produits risquent d'être vecteurs d'exclusion et/ou de sélection, voire de dépendance, d'aliénation pour certains et de puissance et mépris pour d'autres. Et cela, alors qu'une hypothétique convivialité la présente sous des dehors attirants. Si les enseignants n'en sont pas conscients, qui d'autre le sera.

^x Mot préféré au mot "maîtres" dans un cadre tel que celui d'une classe-labo d'informatique?

progressivement remplacé par la nécessité. Aujourd'hui, par exemple, savoir lire et écrire ne constitue plus une promotion car la plupart des gens sont capables de le faire. C'est une nécessité. Il s'agit d'une plate-forme inévitable pour pouvoir accéder à tout autre type de promotion. Dans le cas d'un apprentissage par nécessité, la pédagogie doit être bien présente. C'est ainsi que, depuis plusieurs années déjà, l'enseignement de la programmation a vu son statut se transformer de technique en discipline. Dans l'enseignement de l'informatique des utilisateurs il reste peu de place pour l'improvisation. Apparaissant au rang de la nécessité, son auditoire s'est considérablement étendu et il n'est plus disposé à tolérer des errements dans la méthodologie.

Une évolution du même genre est constatée lorsqu'une nouvelle formation dans un domaine de l'informatique est organisée à l'intention des enseignants ou des élèves, que ce soit dans une école secondaire ou supérieure, et même quelquefois à l'université. Lors des premières sessions, elle est fréquentée par des apprenants prédisposés, dont le niveau de connaissance n'est pas nul, qui souhaitent mettre en valeur ces connaissances et ces prédispositions. Au bout d'un temps, les bons éléments du début sont remplacés par des éléments, certes toujours intéressés, mais qui connaissent davantage de difficultés et qu'il faut encourager. Pour ceux-là, «mettre le contact" ne suffit pas, il faut un moteur qui tourne bien et relance constamment la mécanique.

Ce moteur est parfois mal choisi. Disons plutôt que beaucoup d'enseignants confondent, pour une activité donnée, "pôle d'attraction" et "motivation". Un exemple. Les élèves ont envie de réaliser le journal de l'école. C'est sans doute un magnifique pôle d'attraction pour l'apprentissage du traitement de texte. Il faut pourtant trouver un moteur d'apprentissage, sinon les étudiants se lassent très vite du nouveau jouet, l'abandonnent et en demandent un autre. Ainsi par exemple, il est à peu près certain (il y a peu de certitudes en pédagogie), que donner à l'élève les moyens de résoudre seul et rapidement un problème, c'est lui donner l'envie immédiate d'en résoudre un autre. Bien exploité, ce principe garantit, par une complexité croissante des problèmes proposés, le bon fonctionnement d'un groupe en formation et l'acquisition par ses membres de compétences transférables.

Malheureusement, la voie choisie est souvent différente. Face à une situation problème rencontrée par l'étudiant, l'enseignant, de bonne grâce, débloque lui-même la situation. Ayant trop peu réfléchi à un modèle mental qui pourrait véhiculer chez l'étudiant une meilleure connaissance de l'outil, il ne peut que donner un vague commentaire sur sa pratique personnelle. La frustration qu'engendre chez l'étudiant une telle démarche le conduit très vite à une "non-envie" de comprendre et l'entretient dans l'illusion qu'il a réalisé lui-même ce travail. C'est alors la porte ouverte à une logique du *"Hâtons-nous d'obtenir le résultat, c'est la seule chose qui compte!"*

En fait, ou bien, l'explication fournie n'en est pas une. Propos entendus en classe:

"C'est normal que ça ne marche pas, tu n'as pas appuyé sur Enter!"
"Tu dois faire Outils Options Scénario dans le menu."
...et pour les situations vraiment inexplicables:
"C'est une erreur du système..."

Ou bien il n'y a pas d'explication du tout et l'enseignant prenant possession du clavier de l'élève, résout le problème en trois ou quatre raccourcis clavier sous les yeux (la première fois) ébahis de l'étudiant qui s'habitue bien vite à ce que l'enseignant résolve tous ses problèmes.

La difficulté vient donc dans une première phase, de faire prendre conscience aux enseignants des dangers que présentent ces façons de faire. Extrait d'un éditorial de la revue de l'EPI^{xi} :

"Même s'il semble aller de soi, le système éducatif tel que nous le connaissons est relativement récent, c'est un acquis social considérable. Nous devons tout faire pour le préserver et assurer son évolution qui doit tenir compte du déploiement des technologies modernes.

Les enseignants doivent être préparés dès maintenant à être des initiateurs à l'emploi réfléchi et maîtrisé de ces technologies; ils doivent être capables, pour leurs élèves, d'en exploiter le meilleur et d'en éviter le pire."

On aperçoit malgré tout une oasis dans ce désert de réflexion sur les pratiques: la lente prise de conscience chez les enseignants qui ont une longue expérience de l'enseignement de l'informatique de la nécessité d'une méthodologie adaptée à l'apprentissage de l'informatique des utilisateurs. Leur vécu non négligeable y contribue.

Ces réflexions nous permettent de comprendre que l'essentiel du travail à accomplir consiste à faire évoluer cette situation, ce qui demande beaucoup de patience. Il faut faire admettre l'idée que la gestion de grands projets passe d'abord par l'acquisition et l'assimilation correctes par l'élève d'un certain nombre de concepts et la capacité d'utiliser ces concepts, lorsqu'ils sont acquis, pour résoudre des problèmes simples mais d'une façon parfaite et complète.

Pour y parvenir, le groupe responsable de l'édition du programme va procéder en plusieurs étapes, utilisant tous les moyens disponibles pour arriver à ses fins.

Les balbutiements

Il y a une inertie à vaincre. La mécanique est grippée (sans mauvais jeu de mot). Pourtant l'année qui suit la rédaction du programme (1992-1993) est riche en enseignements. La toute première étape est d'imaginer la rédaction de fiches pédagogiques ayant un double rôle: d'abord, très concrètement, fournir à

^{xi} Revue de l'EPI Editorial n°76 Décembre 1994 Paris 1994

l'enseignant une application, une activité intéressante à faire en classe, ensuite, éveiller son imagination, lui donner des idées et surtout lui faire percevoir dans quel esprit il est souhaitable qu'il parle aux élèves de l'informatique, ce que le programme ne fait que de manière très discursive.

Les participants à la constitution du nouveau programme continuent donc de se réunir en formulant, lors de leurs rencontres, le voeu pieux d'apporter, lors de la prochaine réunion, une fiche pédagogique de leur cru sur un sujet particulier. La fiche pédagogique doit avoir le format A4, reprendre un certain nombre d'indications telles: le thème, l'objectif poursuivi, le public cible, la durée de l'activité etc.

Il y a deux obstacles majeurs à ce mode de fonctionnement. Le premier est le manque de permanence dans le travail. Entre deux réunions, les activités des personnes d'un groupe de travail sont nombreuses et la remémoration d'un travail promis se fait généralement la veille de la rencontre, en consultant l'agenda pour la journée du lendemain.

Les productions sont peu nombreuses, et cela, en dépit du fait que les participants ont en tête pas mal d'idées intéressantes. La réalisation, cela demande du temps, ou à tout le moins, l'occasion de se consacrer exclusivement à la chose pendant un certain temps.

Le second obstacle c'est que pour faire percevoir à l'enseignant dans quel esprit il est bon de faire découvrir l'informatique, la superficie d'une feuille A4 est insuffisante. Il faut garder à l'esprit le problème de la formation des enseignants, pour la plupart, des autodidactes ayant un faible bagage informatique et n'ayant que trop peu souvent réfléchi à leurs pratiques personnelles et encore moins à leur pratiques d'enseignant en informatique. Fournir une activité toute prête à l'enseignant n'a de sens que dans la mesure où celui-ci peut rattacher la description de l'activité à des notions, des concepts (acquis ou à acquérir), au taux de difficulté d'assimilation qu'éprouvent généralement les élèves lors de leur découverte. Par exemple, les professeurs de mathématique ont une expérience de la difficulté que peut poser la notion de limite d'une fonction, d'abord parce qu'ils ont été eux-mêmes confrontés au problème de son assimilation dans un cadre organisé (celui du cours de mathématique qu'ils ont reçu), ensuite parce qu'ils ont probablement vécu très souvent cette difficulté chez des dizaines d'élèves qui découvrent ce concept sur base de pré-requis fort semblables. Si donc un exercice sur les limites est fourni à un professeur de mathématique, celui-ci aura tôt fait de cerner avec exactitude, ce que cet exercice permet d'évaluer, ou ce qu'il met en jeu comme connaissances préalables. Cette capacité de juger de l'opportunité de l'exercice, il la doit à l'existence implicite, dans l'ensemble des professeurs de mathématique, d'un consensus sur les concepts et sur leur importance relative. Dans le domaine de l'informatique des utilisateurs, tout cela est loin d'aller de soi. Le consensus fait défaut, la formation de base aussi, l'expérience didactique est faible et s'appuie sur

des échantillons d'élèves qui détiennent, à des niveaux fort différents, des éléments de connaissance souvent mal organisés ou mal assimilés.

Bref, le travail réalisé au cours de cette année est décevant et prouve, si nécessaire, qu'une permanence est indispensable. Le hasard fait parfois bien les choses car l'année suivante voit apparaître des crédits nouveaux et (provisoirement) durables pour la formation des enseignants sur base d'une décision gouvernementale. Ce que le groupe souhaite mettre en oeuvre ne demande qu'à être rédigé correctement sous forme d'un projet circonstancié. Ce qui est fait par Charles Duchâteau, directeur du CeFIS et François Sass, responsable informatique à la FESeC. Le projet, bien ficelé, est accepté et subsidié sans réticences. La première étape se doit d'être une étape de sensibilisation, de prise de conscience. Cette étape il faut soigneusement la préparer.

Naissance des fiches pédagogiques (documents de base)

L'acceptation du projet résout d'un seul coup le problème de la permanence. Les crédits sont suffisants pour assurer au cours de cette année (1993-1994) le détachement à mi-temps de deux personnes du groupe ayant assuré un certain nombre de formations d'enseignants, l'une d'elles exerçant aussi une mission de conseiller pédagogique en informatique. Elles prendront en charge tous les problèmes liés à la rédaction des fiches pédagogiques. Le groupe évolue donc; il se compose maintenant de quatre personnes ayant participé activement aux travaux du GRIP: les dépositaires du projet et les deux personnes détachées qui sont les chevilles ouvrières du projet.

Une constatation préalable, le type de travail envisagé ne peut en aucun cas avoir la prétention d'être exhaustif, tant en ce qui concerne le nombre de domaines touchés par l'informatique qu'en ce qui concerne un de ces domaines en particulier. Dans une première phase, libérés de cette contrainte d'exhaustivité, les membres du groupe rédigent, à l'intention des enseignants et selon leur disponibilité, des documents sur des thèmes et des sujets qui les intéressent particulièrement et qui leur semblent correspondre à un type de difficulté rencontré dans l'enseignement de l'informatique sous quelque forme que ce soit. Parmi les sujets traités, on trouve par exemple:

- la ligne de commande du système d'exploitation MS-DOS et la sécheresse de sa syntaxe comme prétexte à une réflexion sur le traitement formel,
- un essai de classification des difficultés liées à l'apprentissage de la programmation et la recherche d'exercices progressifs associés,
- une méthode de construction pas à pas des modules répétitifs en programmation,
- les déplacements automatiques du curseur comme prétexte à une autre réflexion sur le traitement formel et comme point de départ à l'écriture de macro-commandes,
- une méthode de formulation des requêtes à un gestionnaire de fichiers,
- une comparaison imagée pour mieux percevoir les différents modes d'adressage des cellules d'une feuille de calcul...

(NDLA: ces différents sujets peuvent faire l'objet d'une description plus approfondie lors d'un exposé oral)

Ces documents se veulent à la fois des réservoirs d'activités possibles, mais aussi, des documents qui permettent à l'enseignant de mettre à jour certaines de ses connaissances, voire lui fassent découvrir l'importance de certains concepts auxquels il n'avait pas l'habitude de s'intéresser. Les membres du groupe se réunissent régulièrement pour se faire part mutuellement de leurs remarques à propos des documents produits, qui, en conséquence, prennent de plus en plus de volume. Si bien que le nom de "fiche" finit par désigner un fascicule dont la taille atteint quelquefois plusieurs dizaines de pages.

Les contacts d'auteurs, les remarques mutuelles qui s'y font donnent lieu à des débats intéressants faisant progresser considérablement le travail. Les réunions s'éternisent parfois, mais elles sont l'occasion de resserrer, voire de conforter les points de vue.

Le groupe est soumis à une contrainte importante! Le projet est subsidié dans le cadre de la formation des enseignants. Et même si le travail en cours est lié, à terme, à des formations potentielles, il faut faire entrer les enseignants dans le jeu à un moment donné. Une solution élégante et efficace est alors trouvée. La plupart des membres du groupe étant déjà formateurs à différents niveaux depuis un certain nombre d'années, contactent des enseignants investis d'une certaine expérience, parmi les personnes qu'ils ont connues au cours des formations déjà organisées. Une bonne vingtaine de personnes répondent à l'appel de collaboration. Il est décidé de travailler en petits groupes sur différents thèmes, selon les affinités:

- le(s) système(s) d'exploitation
- l'algorithmique et la programmation
- les traitements de texte
- les gestionnaires de fichiers
- les tableurs

Les participants disposent des documents rédigés par les membres du groupe pour faire leur choix et se faire une idée du type de travail qui les attend. Un calendrier de rencontres (quatre au départ, suivies de quatre autres pour certains groupes) est fixé. Parmi les réactions des participants lors du premier contact, certains s'étonnent de l'absence de travaux concernant les logiciels de dessin. Il est décidé de créer un groupe qui doit, dès le départ, réfléchir à la constitution d'un document de base sur ce sujet.

Dans le travail qui suit, le rôle des participants est d'apporter leur expérience, leurs idées, leurs remarques sur les documents produits et surtout de garantir que le travail effectué est en accord avec les situations de terrain. Leur présence est particulièrement appréciée lors de la constitution de banques d'exercices car, en interaction avec leurs cours, ils peuvent apporter une appréciation rapide de leur bien-fondé.

Plusieurs enseignements sont à tirer de ce type d'expérience. Ces enseignements paraissent transférables à d'autres situations du même genre. Ils s'adressent principalement à la personne responsable d'un tel groupe de travail.

Concernant la gestion du travail de groupe

Il s'agit bien d'un travail de groupe, même (et surtout), s'il ne réunit que trois ou quatre personnes. Il faut donc qu'une personne (au moins) se préoccupe de sa dynamique. C'est à cette personne qu'il revient de définir une discipline de groupe et d'en rappeler régulièrement les règles:

Respecter les délais fixés

Le faible nombre de participants aide à ce que chacun d'entre eux se sente investi d'une certaine responsabilité. Comme le travail est coordonné, un retard de la part d'un des membres a une incidence sur la progression du groupe. Le responsable ne doit pas hésiter à lancer des rappels à l'ordre si nécessaire.

Respecter les consignes d'encodage

Lorsque des textes doivent être partagés ou regroupés, il vaut mieux exiger un encodage sans mise en forme. Autant dire que cette règle n'est pas facile à faire accepter. Les participants disposent de traitements de texte sophistiqués. Ils ne peuvent éviter les effets de mise en page, même si on le leur demande explicitement. Sans doute, considèrent-ils que la transmission d'un texte sans mise en forme traduit une incompétence. Pourtant, pour des documents nécessitant une uniformisation de la présentation, c'est le moyen le plus efficace de travailler.

Il est toutefois possible de trouver un accord sur quelques conventions simples qui évitent une relecture des textes. Lorsque les participants utilisent le même traitement de texte (ce qui n'est pas souvent le cas) un choix de modèles de document et de styles de mise en page est possible. Ces choix doivent rester simples et peu nombreux. Dans le cas de traitements de texte différents, on peut se limiter à des choix convertibles (par exemple, le choix du gras pour les mots très importants et de l'italique pour les mots moins importants à dégager du reste du texte). En tout état de cause, il faut interdire aux rapporteurs de faire étalage des connaissances qu'ils ont de leur outil, ce qu'ils ne manqueront pas de faire si on ne les en empêche.

Se limiter aux productions annoncées par le groupe

Cette règle est d'autant plus efficace que des échéances sont placées et que le travail est partagé. Au moment du partage du travail, des idées surgissent parfois chez les uns et les autres. Il faut voir dans quelle mesure ces idées nouvelles sont intégrables au travail déjà réalisé et éviter une transformation permanente, voire une dérive des objectifs du travail.

Cette personne est aussi responsable de la production. C'est à elle d'animer, de raccourcir les débats déviants et d'exploiter les idées intéressantes, même si elles mettent à mal un planning.

Concernant le contrôle de la rédaction des documents

Lorsque le travail doit déboucher sur la production d'un document, la personne qui prend en charge cette tâche doit canaliser, contrôler, voire corriger cette production. C'est dire qu'elle doit garder en tête les objectifs du travail, son contrôle général ainsi que celui du style rédactionnel et de la mise en forme du document final. Ces tâches sont assez lourdes, même si la collaboration des membres du groupe est fort utile pour la lecture et la correction des documents.

Elle doit impérativement participer aux réunions de travail et, si elle est également chargée de l'animation, il est nécessaire d'étaler les rencontres pour lui permettre de réorganiser les données du travail précédent. De la sorte, les participants peuvent percevoir le bon avancement du travail. Bien s'assurer aussi que toutes les productions personnelles sont examinées. Cela ne signifie pas nécessairement qu'elles doivent être reprises dans le document final, mais ce serait une erreur de n'avoir pris en compte, à aucun moment, tout ou partie du travail d'une ou plusieurs personnes du groupe. Si certaines productions doivent être éliminées, il vaut mieux que ce soit par le biais d'une discussion que de manière unilatérale.

Il ne faut pas perdre de vue non plus, que les documents sont destinés à être diffusés et accueillis par leurs utilisateurs potentiels, assez souvent sans discernement. Le responsable de la rédaction doit avoir à coeur de faire contrôler la correction du contenu (sur le fond) par ses pairs ou des personnes extérieures compétentes. Dans le cas contraire, les effets peuvent être désastreux.

Concernant l'animation du groupe

Etre l'âme du groupe au vrai sens du terme, alimenter la réflexion, préparer et proposer des pistes, cela paraît très important dans la mesure où, comme dans tout groupe constitué, aucun membre ne se sent investi de cette responsabilité. Même s'il accepte de donner le meilleur de lui-même, le participant attendra souvent qu'on le lui demande. Il ne faut donc pas faire trop rapidement l'hypothèse que chacun est bien conscient du travail à faire et qu'il va le faire sans qu'on lui fournisse des consignes précises. Ainsi, on constate que sur base d'un exercice proposé et après en avoir analysé ensemble l'opportunité et le champ d'action, il est possible de demander aux participants d'imaginer des exercices semblables sur le fond et légèrement différents sur la forme. Ce type de travail est extrêmement intéressant car il permet de multiplier les situations analogues, une chose qui n'est que très rarement réalisée dans les situations d'apprentissage proposées en informatique. On peut alors imaginer des séquences qui se fixent davantage sur l'acquisition de capacités instrumentales. Le nombre d'exercices autorise en sus une meilleure gestion des différences de niveaux puisque certains exercices seront d'une plus grande complexité tout en intégrant les mêmes concepts, parfois en plus grand

nombre. A titre d'exemple, après avoir montré qu'un exercice de programmation concernant la validation d'une donnée fournie par l'utilisateur est un bon support pour tester la construction d'une répétition, on peut demander aux personnes du groupe d'imaginer d'autres scénarios de validation de données^{xiii} : "Ecrire une procédure qui a pour but de forcer l'utilisateur à introduire une réponse au clavier sous une des formes suivantes: O, o, N ou n.", "Ecrire une procédure qui a pour but de forcer l'utilisateur à introduire au clavier une chaîne de huit caractères.",...

La production par imitation donne de très bon résultats et fait appel à l'expérience personnelle et à l'imagination. En revanche, il est souvent difficile d'obtenir d'autres stratégies d'exercices, mettant en jeu les mêmes concepts ou des concepts nouveaux. En d'autres mots, lorsqu'une veine est trouvée, la tendance est fréquemment de l'exploiter à fond, sans penser à prospecter d'autres endroits. Comparant avec la série d'exercices proposée auparavant, il faudra souvent suggérer une idée nouvelle (autre que la validation) pour qu'un nouveau filon puisse être exploité. Ainsi, si on propose un exercice de saisie de données successives, la saisie étant stoppée par une valeur sentinelle, de nouveaux exercices vont fuser à nouveau. "Ecrire un programme qui permette à l'utilisateur d'introduire des nombres au clavier (arrêt lorsqu'il introduit la valeur 0). Le programme fait afficher le nombre de valeurs introduites, à l'exception de la valeur d'arrêt." "Même énoncé que le précédent, mais le programme fait afficher en plus, la somme de toutes ces valeurs." "Même énoncé que le précédent, mais le programme fait afficher en plus, la moyenne arithmétique et géométrique de toutes ces valeurs." "Même énoncé qu'au premier exercice, mais le programme fait afficher en plus, la plus petite et la plus grande des valeurs introduites."

De ce côté, le rôle de l'animateur est important. Il doit posséder un recul suffisant et l'habitude d'une réflexion sur ses pratiques qui l'autorisent à proposer des voies nouvelles et faire la sélection des choses qui lui seront éventuellement proposées. Parfois, accidentellement, des exercices proposés font partie d'une autre catégorie. Il s'agit alors, non de les rejeter, mais de montrer qu'il s'agit bien d'une autre catégorie à retenir et à exploiter plus tard.

Concernant l'efficacité du travail

Le nombre de participants par groupe doit être très limité (trois ou quatre personnes) car leur production, qui peut être importante, doit être gérée correctement.

Les travaux demandés doivent faire l'objet de consignes précises, au risque sinon de tomber dans la dispersion. La production de ces travaux doit faire l'objet d'une correction de la part du responsable du groupe, voire des autres membres.

Cette correction doit tenir compte des exigences de départ. Les activités proposées devant concerner des notions et des concepts précis, il arrive qu'elles

^{xiii} Revue de l'EPI Editorial n°76 Décembre 1994 Paris 1994.

témoignent, de la part de celui qui les propose, d'une mauvaise connaissance des concepts en question. C'est d'autant plus vrai, si des pistes de résolution ou des solutions sont fournies. N'oublions pas que la formation initiale fait défaut, même chez les enseignants chevronnés. Dans ces cas-là, des solutions boiteuses sont généralement fournies. La réponse est trouvée, objectif atteint, ce qui fait la preuve d'un esprit de débrouille issu probablement d'une expérience d'autodidacte. Mais est-ce prudent de fournir de tels pistes à d'autres? Un exemple. Comme travail, on demande aux enseignants de rechercher des exercices proposant des requêtes en gestion de fichiers, dans un cadre strict excluant les tris et indexations^{xiii}. L'un d'eux propose un exercice qui demande l'affichage du nom du pays le plus pauvre du monde en terme de PNB. La solution qu'il fournit est un tri par ordre décroissant des PNB et un affichage de la liste des dix premiers enregistrements. Ce type de déviation est courant, car "trouver une solution", quels que soient les moyens, reste une préoccupation essentielle pour tous ceux qui ont découvert l'informatique de manière un peu anarchique. Les vieux démons sont donc toujours présents.

Un contrôle doit être exercé au niveau didactique. Il arrive qu'une activité proposée ne permette pas du tout la poursuite des objectifs annoncés. Un texte lacunaire, comportant des verbes à l'infinitif, est proposé à l'élève en guise d'exercice sur l'insertion, la suppression des caractères et le déplacement rapide du point d'insertion. Il est fort possible que la difficulté de l'élève dans le domaine de la conjugaison des verbes occulte complètement ses difficultés de manipulation et sa méconnaissance des primitives de l'outil logiciel. Dans de telles situations, il faut pouvoir juger de l'opportunité de certaines activités.

Un document de base, fixant quelques éléments fondamentaux et fournissant des exemples est nécessaire comme référence pour les membres du groupe. Il ne faut pas espérer cette production de la part du groupe lui-même. Les idées fondamentales qui sous-tendent tout le travail doivent être produites par des personnes possédant un recul suffisant par rapport aux outils disponibles, des personnes analysant sans cesse leur pratique personnelle, la pratique des élèves et qui enrichissent leur expérience en y consacrant un temps suffisant. Ce n'est pratiquement pas le cas des enseignants pour lesquels l'informatique ne représente qu'une partie de leur activité professionnelle.

Pour en revenir à la chronologie des faits, l'année de "travaux dirigés avec les enseignants" permet aux documents de base proposés en début d'année de s'étoffer d'un nombre très important d'exercices. Leur relecture et les réflexions menées lors des rencontres en améliorent le contenu théorique. Si bien qu'au mois de juin, un certain nombre de documents sont disponibles, traitant des sujets les plus courants. Deux phases se dessinent alors:

- il faut présenter ce travail,

^{xiii} E. VANDEPUT, D. LORGE, A. MARCHANT, G. VINCENT Gestionnaires de fichiers à propos du questionnement... (F.P. 5.2) CeFIS - ICAFOC - FPE Namur 1994

- il faut le faire évaluer.

En référence, une nouvelle fois à [1] à propos des efforts à réaliser:

"celui enfin de l'expérimentation dans les classes des propositions élaborées."

La première phase voit le jour au mois d'août sous forme d'une école d'été de cinq jours. Au cours de cette école d'été, une quarantaine de participants découvrent les travaux réalisés, au travers d'ateliers dans lesquels ils se retrouvent souvent en situation d'élèves. Une démarche qui est très révélatrice pour ces participants, car les véritables situations-problèmes qui leur sont proposées ont comme conséquence de mettre à jour certaines lacunes de leur enseignement et, parallèlement, mettent en évidence l'intérêt des "fiches pédagogiques".

Un exemple pour illustrer; on peut le généraliser à d'autres logiciels, à d'autres enseignants. Au cours de cette école d'été, un groupe manifeste son intérêt pour le dessin assisté par ordinateur. Nous organisons un atelier sur ce thème. Cet atelier s'inspire du travail réalisé avec quelques enseignants en cours d'année et de la fiche pédagogique n° 8.1 sur le dessin^{xiv}. L'expérience nous a déjà montré par le passé que des professeurs utilisant très régulièrement ce type d'outil ont peu réfléchi aux principes généraux liant les logiciels de cette catégorie. Ce sont des fanatiques du logiciel X version m.n qui peuvent obtenir, pour la plupart des problèmes qu'on leur pose, un semblant de solution. Lorsqu'ils sont interrogés sur leur démarche, ils vous parlent en "logiciel", éprouvant d'énormes difficultés à décortiquer leur démarche mentale souvent peu rigoureuse conduisant à un résultat approximatif. Nous décidons de leur tenir un discours sur le codage des dessins. Nos questions: *Qu'est-ce qu'un dessin d'un point vue formel? Quels sont les systèmes de codage que vous connaissez? Quelles sont les primitives de base d'un logiciel de dessin?* les laissent pantois. Leurs commentaires sont plutôt tournés vers le temps qui n'est pas à perdre à de telles futilités et sur la nécessité, en fin d'année, de pouvoir dresser des plans complexes, de pièces usinées, de bâtiments, etc. Nous décidons de les placer en situation concrète de réaliser un dessin apparemment simple (voir figure) et nous leur fournissons deux logiciels simples de dessin (l'un bitmap, l'autre vectoriel). Nous ne pouvons que constater les dégâts: incapacité de gérer un logiciel simple de dessin car différent de l'outil habituel, difficulté de distinguer les modes de fonctionnement d'un logiciel de dessin bitmap et d'un logiciel de dessin vectoriel et, plus grave parfois, difficulté de mettre au point une stratégie de résolution. Cette dernière difficulté est à mettre en relation avec la difficulté de cerner les actions primitives des logiciels des deux catégories.

^{xiv} F. SASS, E. VANDEPUT, J.M. EVERAERT, C. PUISSANT Dessin assisté par ordinateur. Les actions de base (F.P. n°8.1) CeFIS - ICAFOC - FPE Namur 1994

La seconde phase, celle d'une évaluation significative du travail sur le terrain constitue l'essentiel du projet de l'année suivante.

La sensibilisation (une première prise de contact)

Dire que l'intention des membres du groupe est de faire évaluer le travail accompli n'est sans doute pas trahir la vérité. Inconsciemment, chacun souhaite aussi que ces documents provoquent chez les enseignants un effet de remise en ordre des connaissances, voire leur fassent découvrir des concepts que leur pratique personnelle ne leur permet pas encore d'isoler.

Pour évaluer l'intérêt du travail en ce qui concerne la préparation des cours, l'aide d'une quarantaine de personnes est souhaitable. Et cela, pour donner un certain poids à l'évaluation. Un publipostage un peu trop bien organisé nous en amène environ deux cents que nous ne pouvons abandonner. Notre réaction est double. D'une part, la crainte de ne pouvoir gérer correctement un groupe aussi important, d'autre part, la confirmation de ce que nous pensons: beaucoup d'enseignants se sont engagés dans la conquête de l'informatique, avec beaucoup d'impétuosité, ils ont survécu à cette expérience douloureuse et las de cette prodigalité qui les a caractérisés au cours de ces débuts, ils sont demandeurs d'idées



nouvelles, mais aussi de partages (et ils ont beaucoup de choses à partager).

Nous prenons alors la décision de les réunir lors d'une demi-journée au cours de laquelle ils reçoivent les documents (environ six cents pages et quelques disquettes) et un bref aperçu de la façon dont le travail est mené. Vu leur nombre, nous leur demandons de choisir deux sujets parmi la dizaine proposée: un sujet qui les intéresse particulièrement dans le cadre des cours qu'ils donnent et un sujet de réserve. Ces choix nous permettent de les répartir en groupes d'une vingtaine de personnes auxquelles nous demandons deux choses: la première, c'est de participer à deux demi-journées de sensibilisation au sujet choisi; la seconde, c'est d'accepter de nous revoir en fin de l'année scolaire pour nous faire part, oralement et par écrit, de leurs remarques à propos des documents reçus, de leur opportunité, des améliorations possibles, des réactions des élèves, de leur manière d'envisager une future collaboration etc.

A ce niveau, quelques commentaires sont à faire. Le nombre excessif de participants nous amène à réclamer une participation financière des enseignants pour les documents alors que nous avons promis la gratuité. Nous ne recevons aucune doléance à ce propos. L'attente semble à ce point importante que l'aspect

financier, pourtant souvent déterminant dans l'enseignement, passe à l'arrière-plan. Nous supposons que dans l'anonymat relatif de la réunion, beaucoup de personnes nous abandonneront par manque d'intérêt pour ce que nous leur proposons. C'est seulement le cas de quelques personnes qui nous avouent avoir été mal informées de la teneur du projet mais nous fournissent généralement les coordonnées d'un collègue plus concerné. Finalement, le nombre de personnes acceptant de collaborer reste élevé, ce qui nous cause une charge administrative dont nous serions volontiers passés (courrier, notes de frais de déplacement, envois de reçus, d'attestations,...).

L'expérience des deux séances de sensibilisation dans chacun des groupes est plus révélatrice encore, soit du désarroi des enseignants à gérer des situations locales difficiles, soit de leur manque de formation, soit du peu de réflexion sur leur pratique personnelle et celles de leurs élèves. Il faut dire que la manière dont ces séances sont menées ont de quoi les perturber un peu.

L'enseignant qui choisit comme thème le traitement de texte s'étonne lorsqu'on lui demande quelle classification il fait des traitements de base de ce type de logiciel, lorsqu'on lui demande de préciser ce qui est considéré comme mot par son logiciel favori, de comparer ce concept dans différents logiciels, d'en déduire les incidences au niveau du déplacement rapide du point d'insertion, de construire des textes (au sens large du terme) qui mettent à mal ou qui éprouvent la solidité d'une définition trouvée pour ce concept de mot etc. Celui qui choisit comme thème le dessin assisté par ordinateur est tout autant surpris (nous en avons déjà parlé) de ce qu'on lui demande quelle est sa conception du dessin, en vue d'imaginer comment un dessin peut devenir une information pour un ordinateur, de ce qu'on le place en situation de réaliser des dessins qui demandent d'établir une stratégie préalable en lien à la bonne connaissance qu'il a de son logiciel. Quant à l'enseignant qui choisit la gestion d'une base de données, l'amener à réfléchir sur les difficultés des élèves à formuler une requête a aussi de quoi l'étonner, ce qui d'une certaine manière est un comble. A ce propos, il n'est pas rare d'entendre des discours du genre: *"La formulation des requêtes ne pose aucun problème pour les élèves."* Ce qui cache souvent une certaine simplicité des requêtes proposées, voire une méconnaissance de ce qu'elles sont réellement. Ou bien encore: *"Les requêtes, certains élèves ont tout suite compris comment il faut les formuler, les autres ne comprendront jamais."* qui montrent à quel point l'acte d'enseigner est un peu dévalorisé.

En conséquence, ces deux séances ont comme effet, chez bon nombre de participants de mettre en exergue la nécessité d'une réflexion plus profonde sur les concepts véhiculés au travers des outils employés, laissant apparaître des idées d'exercices intéressants mais aussi, parfois, mettant à jour des besoins de formation.

Chacun s'en retourne cependant chez lui, avec pour mission d'envisager ses cours un peu différemment, à la lumière des informations nouvelles qu'il a reçues. Rendez-vous est également pris, en fin d'année, pour faire le point, les professeurs

recevant un guide d'évaluation sous forme d'un questionnaire à nous renvoyer préalablement pour nous permettre de préparer cette réunion.

Le nombre d'enseignants ayant mené à terme l'évaluation est finalement moins important que prévu: environ 75 personnes, soit à peu près la moitié de ceux qui se sont engagés au départ. Les raisons souvent invoquées par ceux qui n'ont pas pu nous aider jusqu'au bout sont principalement la difficulté d'intégrer, en cours d'année, une façon de travailler très différente de la manière habituelle. Il arrive que la matière concernée soit déjà entamée lors de la réception des documents, ce qui rend difficile la mise en route d'un nouveau processus. Il faut se souvenir de ce que la présentation du travail a été, pour beaucoup d'enseignants, l'occasion de nombreuses découvertes. La peur de ne pas bien dominer ces nouveautés joue probablement chez certains. Le statut accordé au cours y est peut-être aussi pour quelque chose. Certains enseignants savent en effet que le cours ne leur sera plus attribué l'année suivante, ce qui constitue un frein à un investissement considérable.

Pour ce qui est des avis que nous recevons, par écrit et de vive voix, ils prouvent que la démarche entamée répond à un véritable besoin. Interrogés sur la manière dont ils envisagent l'avenir, trois tendances se dégagent chez les enseignants^{xv}.

Une demande de poursuite du travail accompli

Plus de documents pédagogiques pour une plus grande aisance dans la préparation des cours.

“Les fiches pédagogiques, telles qu’elles ont été présentées c’est-à-dire avec de nombreux exercices attrayants et quelques explications d’ordre pédagogique sont pour moi une source directement utile”

“Ce qui m’intéresse le plus, ce sont les fiches pédagogiques afin de faciliter le travail de préparation des leçons et l’acquisition des concepts par les élèves. Il serait utile pour nous d’être tenus au courant des modifications apportées aux fiches pédagogiques...également de la création de nouvelles fiches.”

L'expression d'un besoin de rencontres, de partages d'expériences

“Je crois qu’il faut se sentir guidé et soutenu dans cet enseignement car personne ne sait exactement ce qui est bon ou ce qui est le mieux à enseigner. Il serait intéressant de favoriser les contacts entre enseignants.”

“Il me semble qu’il faut continuer de réunir les professeurs de manière à les guider dans l’utilisation des fiches. Ces rencontres nous permettent également de partager nos difficultés avec d’autres.”

^{xv} E. VANDEPUT Fiches pédagogiques en informatique Synthèse de l'évaluation Namur 1995

"Il me serait très agréable de pouvoir participer à l'élaboration de certaines fiches ou de présenter certains exercices qui en feraient partie. L'expérience des professeurs devrait pouvoir se partager beaucoup plus souvent."

Une demande de formation

"Ce type d'initiative doit absolument être poursuivi. Sous cette forme (journées de formation et expérimentation des fiches) ou sous d'autres formes: séminaires, mise en commun d'expériences, développement d'activités, investigation de nouveaux domaines..."

Nous n'avons pu attendre les résultats de l'évaluation pour mettre au point un projet de formation pour l'année suivante. Fort heureusement, dans ses objectifs, ce projet rencontre les trois souhaits exprimés lors de l'évaluation, même si ces trois souhaits sont rarement exprimés ensemble par la même personne.

Une première génération de formés

La formation que nous proposons ne peut ignorer tout le travail accompli. Elle doit donc s'appuyer sur les travaux réalisés et, en tous cas, permettre de les valoriser. Elle se doit même d'encourager le type de réflexion qui a été mené jusque là. Il faut enfin que les participants y découvrent des éléments nouveaux de connaissance.

Un autre point de vue nous guide: dans les établissements d'enseignement, les problèmes d'utilisation ne cessent de se multiplier: virus, modifications fréquentes des configurations dans les laboratoires, choix du matériel, formation permanente des utilisateurs etc.

C'est deux éléments de réflexion nous orientent vers une formation lourde (deux cents heures environ) que nous intitulons "Formation didactique à destination des enseignants d'informatique et des personnes ressources". Elle est articulée autour de huit thèmes:

- codage et traitement formel,
- histoire de l'informatique,
- architectures matérielles et logicielles d'un système,
- gestion d'un pool de machines,
- éléments de structuration d'un travail réalisé avec un progiciel,
- algorithmique à travers les progiciels,
- concept d'objet et répercussions sur le monde de l'utilisation,
- nouveaux outils de communication et de navigation dans les BD.

On peut s'interroger sur le choix de ces différents thèmes. Nous pensons qu'ils rencontrent pas mal de préoccupations, et des plus variées. Quels sont les fondements de l'informatique que ses utilisateurs ne peuvent ignorer? Comment solutionner des problèmes concrets et empoisonnants liés au matériel? Par quel biais continuer à promouvoir les bienfaits pressentis de l'activité de programmation? Comment préparer l'avenir? Autant de questions que nous

devenons de la part des gens que nous côtoyons régulièrement. Ce serait trahir la vérité que d'omettre de signaler la chose suivante. Les sujets traités correspondent aussi à des passions didactiques entretenues par chacun des formateurs potentiels. Le succès de l'acte d'enseigner est souvent lié au rapport qu'entretient l'enseignant avec la discipline qu'il veut transmettre. Ce principe, quelque part a dû nous guider dans ces choix.

La plupart des modules comprennent trois facettes. Le premier côté, théorique, doit permettre d'organiser les concepts fondamentaux. Le but de cette première phase est l'acquisition ou la mise à jour des connaissances lors d'une présentation qui doit être suggestive pour la suite. Un deuxième côté, plus pratique, existe parfois pour contrôler l'assimilation de ces connaissances. Enfin, un côté didactique autorise le partage des points de vue, des expériences et l'élaboration de séquences intéressantes à diffuser, sorte d'appendice à la rédaction des fiches pédagogiques.

A l'heure de rédiger cet article, la formation est sur le point de démarrer. Des enseignements et commentaires pourront être exprimés à l'occasion du Colloque.

Quoi qu'il en soit, nous ne pouvons nous permettre d'organiser cette formation à la légère car nous sommes allés trop loin lors des premiers contacts que nous avons eu avec les enseignants. Beaucoup d'entre eux éprouvent maintenant à nouveau l'envie de se former, comme aux premiers temps.

Comme partout, l'informatique a d'abord été l'apanage des professeurs de mathématique et de sciences. Sans doute, le côté formel des traitements n'avait-il pas de quoi les effrayer. Ces personnes se sont formées ont beaucoup investi et à l'apparition de l'informatique pour tous, ils ont eu deux attitudes différentes: soit une attitude conservatrice (la programmation envers et contre tout), soit une attitude de renoncement. Pour eux, il valait mieux tout abandonner que de pactiser avec le diable des logiciels. Dans les deux cas, l'informatique des utilisateurs semblait peu les intéresser. D'autres personnes qui n'avaient pas leur expérience se sont chargées de sa diffusion, sur base d'une pratique personnelle pas toujours très solide. Mais passé le temps de l'émerveillement, des difficultés didactiques ont surgi que ni les uns ni les autres ne soupçonnaient.

Aujourd'hui, beaucoup d'inconditionnels de la programmation et de l'algorithmique ont compris que l'informatique des utilisateurs est génératrice d'autant (si pas de plus) de problèmes à résoudre que la programmation. Ce nouveau défi semble leur donner une nouvelle jeunesse. Ces personnes se retrouvent sur les lieux de formation.

Autre élément à citer, le nombre d'enseignants qui ont souhaité s'inscrire à cette formation est très nettement supérieur au nombre que nous pouvions décemment accepter. Raison de plus pour prendre en compte les aspirations du public et les intégrer dans la démarche que nous voulons leur faire suivre.

Les fiches pédagogiques ne constituent pas le contenu de la formation. Elles servent toutefois de base et de modèles de réflexion au cours de celle-ci. Il est possible de se les procurer au prix coûtant augmenté des frais d'envoi, en s'adressant à Etienne Vandeput, CeFIS FUNDP, rue de Bruxelles 61, 5000 NAMUR

Etienne VANDEPUT
CeFIS Facultés Universitaires Notre-Dame
de la Paix - Namur -
Rue de Bruxelles 61, 5000 NAMUR

Thème6 : L'Enseignement Professionnel

Intégration du pilotage de micro-robots pédagogiques à un environnement de programmation

Pascal LEROUX

Résumé : *Dans cet article, nous présentons les difficultés liées à l'utilisation d'un langage et d'un environnement de programmation par des publics dits de bas niveau de qualification dans des activités de pilotage de micro-robots pédagogiques. Ces problèmes nous ont conduit à réaliser l'environnement d'apprentissage ROBOTTEACH. Ce système assiste, d'une part, les apprenants dans la création de programmes de pilotage en générant les programmes à partir d'une description du micro-robot manipulé : cette description est construite par les apprenants. D'autre part, ROBOTTEACH est constitué d'un environnement de programmation et de pilotage spécifique. Cet environnement permet aux apprenants de créer, par manipulation directe, des programmes complexes à partir des programmes issus de l'environnement de description et à partir de structures algorithmiques en langage naturel.*

Par ailleurs, nous montrons qu'il est intéressant d'intégrer un contrôle sur des objets réels (tels que les micro-robots) à des environnements de programmation dans un contexte d'initiation à la programmation. Grâce à la «réalité» des micro-robots on donne du sens aux programmes : l'exécution d'une action sur le micro-robot renvoie une image aux apprenants du programme qu'ils ont créé. Cet «effet miroir» est très important dans les phases de débogage de programmes : l'erreur de programme se trouve réifié sous la forme d'un dysfonctionnement du micro-robot.

1. Introduction

Le Laboratoire d'Informatique de l'Université du Maine (LIUM) s'intéresse depuis plusieurs années à la création d'environnements d'apprentissage en technologie fondés sur les approches micromonde au sens de Papert [Papert 81]. Ces environnements utilisent des micro-robots¹ modulaires de type Fischertechnik® (Cf. figure 1), montables et pilotables à partir d'un micro-ordinateur [Vivet 86]. Une méthode pédagogique appelée ATRIUM² a été élaborée au sein du LIUM [Leroux 95]. Elle consiste à mettre en interaction un maître et des groupes de deux à trois apprenants avec des ressources matérielles (des micro-robots, des notices) et des ressources logicielles pour piloter les micro-robots.

¹ Nous définissons sous le terme générique «micro-robot» des modèles réduits de bras manipulateurs et autres systèmes automatisés.

² Alphabétisation Technologique en Robotique et Informatique de l'Université du Maine.

Dans le cadre de ces recherches, nous avons réalisé un EIAO³ (ROBOTEACH), support de la méthode ATRIUM [Leroux 95]. Il est délibérément orienté pour aborder ce que nous appelons le problème de l'alphabétisation technologique (apprentissage de la culture technologique de base telle que les concepts de mouvement, de capteurs/moteurs, d'axe, de chaîne cinématique, de logiciel de pilotage, de cycle). Cette alphabétisation s'adresse, d'une part, à la formation initiale d'élèves qui seront amenés à travailler dans des contextes de productique ou d'automatismes et, d'autre part, à la requalification d'ouvriers travaillant dans des unités de production qui subissent la mutation des machines traditionnelles vers des machines automatisées.

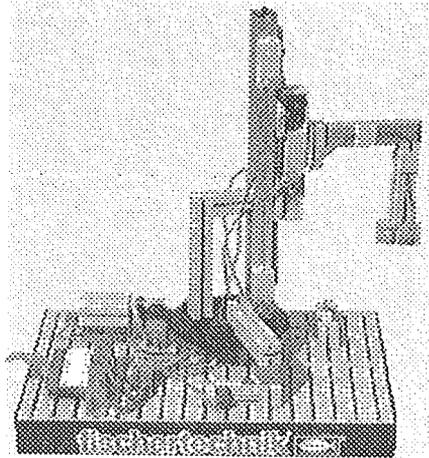


Figure 1 : Exemple d'un micro-robot

ROBOTEACH a été conçu à partir d'analyses effectuées sur l'usage d'un premier environnement apprentissage dans lequel LOGO était employé pour le pilotage des micro-robots. Les expérimentations ont montré qu'il existait des problèmes liés à la programmation des mouvements des machines en particulier pour des publics dits de bas niveau de qualification. Dans cet article, nous exposons d'abord les problèmes rencontrés au niveau de la programmation (Cf. section 2) puis nous présentons le système ROBOTEACH en explicitant l'aspect assistance aux apprenants dans la création de programmes de pilotage (Cf. section 3).

Même si l'objectif de ROBOTEACH n'est pas de former des programmeurs, certaines notions de programmation, telles que les structure de répétition, sont nécessairement abordées pour la réalisation d'actions complexes par les micro-robots (par exemple déplacement de pièces métalliques en automatique). Par conséquent, nous pensons que le système ROBOTEACH n'est pas exclusivement réservé à l'alphabétisation technologique mais que son utilisation peut être étendue à l'initiation à la programmation. Nous expliquons notre point de vue sur ce sujet

³ Environnement Interactif d'Apprentissage avec l'Ordinateur.
Sè Rencontre Francophone de Didactique de l'Informatique-Monastir-Tunisie.10-12/04/96

dans la section 4 en montrant notamment l'intérêt du contrôle d'objets réels dans un contexte d'initiation à la programmation.

2. Difficultés liées à la programmation pour des publics dits de bas niveau de qualification

Le premier environnement d'apprentissage en micro-robotique pédagogique développé au LIUM a été conçu sur la base d'un double micromonde : un micromonde matériel constitué de micro-robots et le micromonde logiciel LOGO pour programmer et piloter les machines [Parmentier et al. 91]. Cet environnement a surtout fait l'objet d'expérimentations dans le cadre de projets de recherche menés en partenariat avec des entreprises et les milieux de la formation professionnelle (projets QUADRATURE [Parmentier et al. 91, Vivet et al. 91], PLUME [Parmentier et al. 93]). Ces expériences ont montré l'intérêt de tels environnements d'apprentissage dans des contextes de remise à niveau ou de requalification pour des publics dits de bas niveau de qualification. Elles ont aussi montré les limites de l'utilisation de LOGO avec ce type de public. Nous avons classé les difficultés rencontrées en deux catégories : celles liées à l'apprentissage de notions de programmation et celles survenant lors de l'utilisation de l'environnement de programmation.

2.1. Problèmes liés à l'apprentissage des notions de programmation

Nous avons constaté au cours des expériences QUADRATURE et PLUME que la programmation des mouvements des micro-robots était plus perturbante que motrice. Le non respect de la syntaxe des commandes a été une des principales causes d'erreurs ; ce n'est cependant pas la plus compliquée à corriger. Par contre, la création des procédures élémentaires⁴ (Cf. figure 2) a été plus difficile à assimiler par les stagiaires.

POUR ALLER_A_DROITE	
SI ETAT "E1 = 1 [STOP]	si le micro-robot est déjà à droite alors on interrompt la procédure
MARCHEP "M1	on met en marche le moteur M1 dans le sens de rotation positif
SCRUTE "E1	on scrute un changement d'état de l'interrupteur E1 (le changement d'état de l'interrupteur sera le signal d'arrivée du micro-robot à droite)
ARRET "M1	on arrête le moteur M1
FIN	

Figure 2 : Exemple d'un programme de pilotage écrit en LOGO

⁴ Un programme élémentaire assure l'exécution au niveau du micro-robot d'un mouvement élémentaire (mouvement permettant d'aller d'un interrupteur à un autre interrupteur, les deux interrupteurs étant associés à un même axe) ou d'une action élémentaire (par exemple l'activation d'un aimant, l'ouverture d'une pince).

La réalisation de telles procédures nécessite l'association correcte de primitives de base (par exemple MARCHEP, SCRUTE) avec des instructions de condition ce qui implique de la part des apprenants une maîtrise des primitives de base. Or dans un environnement de programmation, les stagiaires ont déjà du mal à écrire correctement des lignes de commande. En leur demandant en plus de maîtriser les primitives de base, nous les « surchargeons cognitivement » de notions et de détails à prendre en compte pour piloter leurs machines.

En fait, les difficultés rencontrées sont inhérentes à l'apprentissage de tout langage de programmation : non respect de la syntaxe (mauvaise écriture de noms de commandes, oubli d'espaces) mais très certainement aussi⁵ à des problèmes d'algorithmique (confusion dans l'emploi des instructions conditionnelles et de répétition) et d'analyse (mauvais enchaînement des commandes pour réaliser un mouvement). Or l'objectif de ces expérimentations n'était pas de former des programmeurs mais de fournir un environnement dans lequel les stagiaires puissent piloter leurs micro-robots tout en comprenant leur fonctionnement. Par le biais de cette observation, un nouvel objectif de recherche se faisait jour : comment apporter une aide logicielle à la programmation des mouvements ?

2.2. Problèmes d'utilisation de l'environnement de programmation

En dehors des difficultés de programmation attendues pour des néophytes en informatique, le principal blocage fut constaté au niveau de l'édition et l'exécution de programmes. Pour faciliter la création de procédures, les usagers de LOGO disposent d'un éditeur pleine page. Le passage de la partie édition à la zone de commandes se fait normalement aisément. Certains stagiaires n'ont jamais pu assimiler cette différence. Ils confondaient sans cesse les deux zones : ils exécutaient des commandes dans l'éditeur et vice et versa. En fait cette transition cognitive entre deux mondes conceptuellement différents leur était inaccessible : ils étaient perdus.

3.ROBOTEACH: un assistant logiciel à la création de programmes de pilotage

Le but à atteindre dans le développement de ROBOTEACH était double : il s'agissait d'assister les apprenants dans la création des programmes de pilotage et de leur fournir un environnement de programmation et de pilotage spécifique, minimisant les problèmes syntaxiques lors de l'écriture des programmes, et facilitant les tests des programmes. Ce double objectif a été atteint en intégrant dans ROBOTEACH deux environnements :

- un environnement de description permettant aux apprenants de décrire un micro-robot au système, ce dernier assurant la génération automatique des programmes de pilotage ;
- un environnement de programmation et de pilotage par manipulation directe.

⁵ Ce sont des hypothèses que nous émettons en l'absence d'une étude fine de ces phénomènes.

Les apprenants connaissent du point de vue mécanique le micro-robot qu'ils ont construit et les actions qu'ils veulent lui faire faire. Par contre, ils ne savent pas créer les programmes élémentaires qui leur permettent de piloter la machine. Le système⁶ a des connaissances sur la génération de ces programmes et sur le diagnostic de pannes mécaniques et électriques des micro-robots. Par conséquent, les apprenants et le système coopèrent afin d'aboutir au pilotage correct du micro-robot [Leroux 95].

La coopération se traduit par la séparation des tâches à effectuer. Le groupe d'apprenants a pour tâche, d'une part, de décrire le micro-robot au système à travers l'environnement de description et, d'autre part, d'élaborer les programmes complexes⁶ dans l'environnement de programmation/pilotage à partir des programmes générés par l'environnement de description. Quant au système, il génère les primitives de base et les programmes élémentaires à partir de la description donnée par les apprenants et assiste le groupe lors de la description et de la création des programmes complexes.

3.1. Environnement de description

L'environnement de description permet la description de n'importe quel micro-robot et la génération des programmes de pilotage associés. La description se fait à l'aide d'objets graphiques (Cf. figure 3). Le système peut apporter une assistance à la description, variable en fonction du public formé et du type d'activité; il fait appel au maître quand il ne sait plus répondre aux sollicitations des apprenants. L'assistance consiste à donner des conseils pour déterminer les pannes mécaniques ou électriques du micro-robot ou aider les apprenants à retrouver les erreurs dans leur description (par exemple un mauvais nom attribué à un moteur).

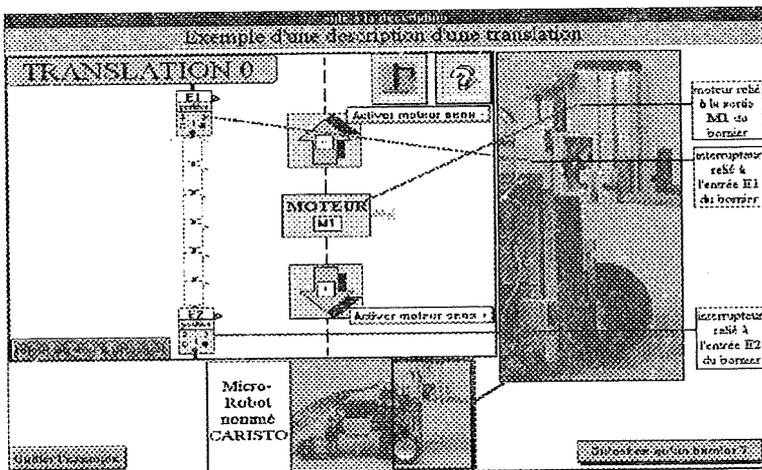


Figure 3 : Lien entre une description graphique et le micro-robot

⁶ Un programme complexe est composé d'une combinaison de programmes élémentaires et éventuellement de structures algorithmiques d'itération et/ou de répétition.

À tout moment il est possible de tester la description en exécutant des actions au niveau du micro-robot : prenons l'exemple du test d'un programme élémentaire. Un programme élémentaire est constitué d'une série d'actions qui assurent le contrôle du micro-robot. Dans une exécution de programme avec compte rendu (Cf. figure 4), chaque action est exécutée et réifiée sous la forme d'une ligne d'information précisant ce que fait le système. En leur proposant ces comptes rendus d'exécution, nous offrons aux apprenants la possibilité d'accéder au contenu des programmes élémentaires sous une forme explicative et de comprendre le fonctionnement de ces programmes. Un programme élémentaire n'est plus présenté comme une succession de lignes de codes mais comme une succession d'événements. Nous espérons ainsi rendre les programmes plus transparents et compréhensibles pour des néophytes en programmation.

Objectif du programme : aller à l'interrupteur E3 en activant le moteur M3 dans le sens de rotation négatif	
1. on lit la valeur de l'interrupteur E3 pour savoir si cet interrupteur est actionné. —> la valeur lue est 0. L'interrupteur étant câblé en logique positive, on en déduit qu'il n'est pas actionné.	
2. on met en marche le moteur M3 dans le sens de rotation négatif.	
3. on surveille un changement d'état de l'interrupteur E3.	
4. l'interrupteur est actionné alors on arrête le moteur M3.	
Fin du programme : l'interrupteur E3 est atteint.	OK

Figure 4 : Compte rendu d'une exécution d'un programme élémentaire

3.2. Environnement de programmation et de pilotage

Comme nous l'avons signalé dans la section 2, il était nécessaire de créer un environnement de programmation dans lequel les contraintes syntaxiques des langages de programmation sont supprimées tout en conservant une richesse dans la structure des programmes. Il est clair qu'un tel environnement n'élimine pas les difficultés conceptuelles liées à la programmation [Anjaneyulu et al 92] mais au contraire, permet de focaliser l'attention sur les notions fondamentales. Par ailleurs, nos objectifs pédagogiques en programmation sont limités. Il s'agit simplement de valider de petits programmes de débutants ce qui ne nécessite pas une puissance de traitement importante.

Dans l'environnement de programmation/pilotage, les apprenants créent leurs propres programmes en manipulant les programmes générés par la description (Cf. figure 5). Pour cela, ils disposent des principales fonctionnalités des environnements de programmation, c'est-à-dire l'édition, l'impression, la suppression et l'exécution de programmes. Toutes les manipulations se font par des clics souris (nous parlons de manipulation directe). Aucune ligne de programme n'est à taper ; donc les erreurs syntaxiques sont impossibles. L'écran de ce module est toujours constitué d'une barre des outils en haut et d'un texte sur l'objectif en

bas. Entre les deux, les boutons et zone d'affichage varient selon les activités engagées par l'utilisateur (création, modification ou exécution de programmes).

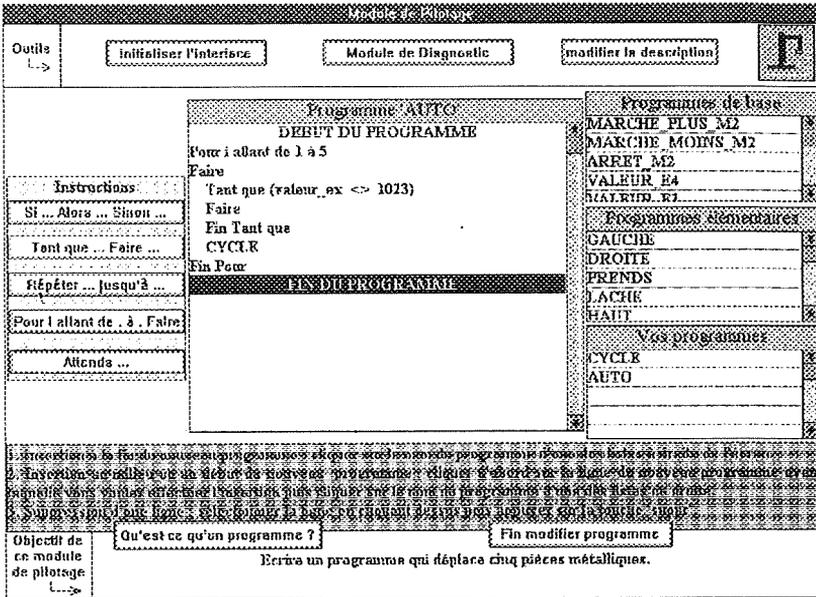


Figure 5 : Le module interface de l'environnement de programmation et de pilotage

Dans la phase d'édition d'un programme, le centre de l'écran est décomposé en trois parties (Cf. figure 5). À gauche on trouve la liste des instructions qu'il est possible d'intégrer dans une procédure, au milieu la zone d'édition du programme et à droite trois listes de programmes : les primitives de base et les programmes élémentaires issus de la description ainsi que les procédures créées par l'utilisateur.

Les conditions associées aux instructions «*si alors sinon*», «*tant que*» et «*répéter jusqu'à*» sont composées comme suit :

(condition_élémentaire) ou (condition_élémentaire)

(condition_élémentaire) et (condition_élémentaire)

(condition_élémentaire)

la «condition_élémentaire» correspondant à la condition suivante :

(programme_de_base_de_lecture_d_une_entrée opérateur valeur).

Les opérateurs possibles sont =, <>, <=, >=, <, > et les valeurs 0 ou 1 quand il s'agit d'une entrée binaire, ou, entre 0 et 1023 pour une entrée analogique. Voici un exemple de condition élémentaire : (ETAT_E1 = 1).

Parmi les outils disponibles dans l'environnement, deux sont permanents : l'initialisation de l'interface et le module de diagnostic. Ce dernier permet au groupe de tester élément par élément son micro-robot sans entrer dans des considérations de programmation. L'interface physique de contrôle du micro-robot peut quelquefois conserver en mémoire des informations. Son initialisation permet d'effacer les informations présentes. Deux autres outils peuvent être mis à la disposition du groupe (c'est au maître d'en décider au moment du paramétrage⁷ de ce module) :

- une modification de la description. L'exécution des programmes de base et des programmes élémentaires contribue à une nouvelle validation de la description. Dans le cas où une erreur de description a été décelée, ROBOTTEACH donne la possibilité aux apprenants de modifier leur description. Au retour de l'environnement de description, les programmes de base et les programmes élémentaires sont automatiquement mis à jour.

- une demande d'assistance. Cette assistance est en cours de développement. Nous l'envisageons sous deux formes : une assistance au débogage de programmes et une assistance à la création des programmes complexes. Selon le même principe que l'assistance à la description, l'agent d'assistance au débogage de programmes apporte des conseils aux apprenants afin qu'ils trouvent et corrigent leurs erreurs. L'assistance à la création de programmes complexes sera effectuée sous la forme d'un partenariat entre les apprenants et le système [Bruneau et al. 91].

4. Utilisation de ROBOTTEACH pour l'initiation à la programmation

Le système est actuellement dédié à la micro-robotique pédagogique mais cette utilisation n'est pas exclusive. Après quelques modifications (Cf. section 4.2), nous pensons qu'il est adapté pour l'initiation à la programmation.

4.1. Intérêt d'employer des micro-robots pour la programmation

L'objet «micro-robot» est le centre d'intérêt des environnements d'apprentissage que nous avons développés en micro-robotique pédagogique. Sa «réalité» fait qu'il ne laisse pas insensible les apprenants mais qu'il a plutôt tendance à les attirer. Malgré les difficultés qu'ils avaient dans les phases de programmation, les publics d'adultes des formations QUADRATURE et PLUME ne se sont jamais découragés : ils avaient toujours pour objectif de faire bouger leurs micro-robots et ils devaient y arriver. À travers les activités de programmation et de pilotage, ils ont pour la plupart acquis quelques notions de programmation ; ils l'ont fait de plus avec plaisir. Or nous pensons qu'ils n'auraient pas acquis ces mêmes notions avec autant de plaisir si nous n'avions pas établi cette liaison entre le monde réel et la programmation.

L'apprentissage de la programmation se fait traditionnellement sur des objets abstraits tels que des fonctions mathématiques ou au mieux sur des objets virtuels tels que la tortue écran LOGO. Apporter une nouvelle dimension aux objets

⁷ Les aspects création des sessions pédagogiques et paramétrage des activités ne sont pas abordés dans cette article. Nous faisons référence ici à [Leroux 95].