

BRUILLARD Éric (1997). *Les machines à enseigner*.  
Éditions Hermès, Paris, 320 p.

## Chapitre 4

# Des micromondes aux environnements d'apprentissage ouverts

### 4.1. Introduction

Vers la fin des années soixante, une nouvelle orientation dans l'usage éducatif des ordinateurs commence à se développer. En effet, d'une part l'ordinateur conquiert une place croissante dans de nombreuses activités humaines et, d'autre part les thèses behavioristes sont fortement remises en question. Combien de temps encore quelqu'un qui ne connaît rien à l'ordinateur pourra-t-il être considéré comme instruit, se demande ainsi Luehrmann (1972) ? En effet, si « l'ordinateur se révèle si efficace qu'il peut être programmé pour simuler le processus d'enseignement, ne devrions-nous pas apprendre aux élèves à maîtriser ce puissant outil intellectuel ? ». Pour citer un slogan maintes fois repris, il ne s'agit plus que l'ordinateur programme l'apprenant mais de donner à l'apprenant la possibilité de programmer la machine. La cybernétique et l'enseignement programmé ont dégagé l'idée de contrôle, mais ce contrôle, au lieu d'être confié à une machine, peut être rendu à l'apprenant.

Sous l'influence, entre autres, des théories piagésiennes, les principes même de l'enseignement programmé sont controversés. Ainsi, Ted Nelson (1987, p. 129) récuse le découpage des savoirs en séquences de présentation, qu'il qualifie d'artificiel. Sa position rejoint celle de Papert (1993, p. 63). Pour ce dernier, quand la connaissance est distribuée parcimonieusement en minuscules parties, on ne peut rien faire d'autre que de la mémoriser en classe et la mettre par écrit dans les tests. On retrouve là certaines conceptions classiques des philosophies de l'éducation (Dewey, Decroly...) fondées plutôt sur l'activité des apprenants et l'analyse de leurs besoins et de leurs intérêts que sur une organisation détaillée préalable des connaissances à apprendre. La connaissance n'apparaît plus seulement comme la somme d'un certain nombre de compétences élémentaires<sup>1</sup>. Vis-à-vis des

---

1. Pour Nelson, précurseur dans le domaine des hypertextes, les interconnexions arbitraires d'un domaine de connaissance, qui ne respectent pas la page imprimée, sont les structures fondamentales que l'apprenant doit traiter et comprendre (Nelson 1987, p. 31).

développements issus de l'enseignement programmé, il s'agit bien d'une rupture, puisque la finalité n'est plus de concevoir une machine à enseigner indéfiniment adaptable mais de créer des environnements contrôlés par les apprenants et dans lesquels ils peuvent exprimer leurs idées et en explorer les conséquences.

Rendre le contrôle à l'apprenant pose bien entendu la question d'un langage ou d'une forme de communication avec la machine suffisamment simple, tout au moins au démarrage, pour ne pas exiger de compétences trop importantes. Il faut donc trouver un bon compromis entre puissance, expressivité et simplicité de mise en œuvre. Les langages issus des recherches en intelligence artificielle, conçus pour faciliter l'*expressivité* afin d'explorer des problèmes complexes, s'avèrent de bons candidats. Il n'est pas question de se cantonner à l'apprentissage *de* la programmation à l'aide de tel ou tel langage spécifique, mais plutôt de faciliter des apprentissages *par* la programmation. Sous l'impulsion de Papert émerge alors la notion de micromonde.

En dehors de cette question du contrôle, exercé par l'apprenant ou par la machine, les multiples potentialités offertes par l'ordinateur conduisent les chercheurs à s'intéresser à la modélisation. Ainsi, Hébenstreit (1971) soutient que toute méthode qui ne vise qu'à perfectionner l'enseignement conçu comme transmission de connaissances est vouée à l'échec<sup>2</sup>. Pour lui, le but majeur de l'enseignement consiste à raccourcir le temps d'apprentissage en enseignant non pas les connaissances, mais les modèles qui rendent ces connaissances cohérentes et par conséquent opérationnelles. Les développements autour des projets WHY et SOPHIE, relatés dans le chapitre précédent, vont aussi montrer l'intérêt du travail autour des modèles. D'une façon analogue aux résolveurs, pour lesquels les contraintes d'enseignement conduisent à privilégier les *boîtes de verre* aux *boîtes noires*, l'aspect qualitatif, permettant notamment de mieux prendre en compte les relations de causalité, s'affirme fondamental. Les modélisations et simulations, à visée éducative, vont aussi tirer parti des capacités graphiques des ordinateurs.

En effet, l'essor progressif de possibilités graphiques des ordinateurs conduit peu à peu à imaginer d'autres modes de communication avec la machine. Dans un premier temps, les travaux avec les résolveurs de problèmes, présentés dans le chapitre précédent, montrent que l'emploi de la langue naturelle n'est pas nécessairement la meilleure solution pour faire faire des actions à la machine. Le choix d'un ensemble limité de primitives, adapté aux problèmes à traiter, apparaît pertinent. Toutefois, dans un cadre éducatif, donner des outils pour réaliser une tâche n'est pas toujours suffisant, il faut pouvoir *dialoguer* avec l'apprenant sur la tâche en cours, pour l'aider à la traiter ou améliorer ses performances. Faut-il pour

---

2. Il précise d'ailleurs que cette critique s'adresse particulièrement à l'enseignement programmé qui se présente comme une technique visant à une utilisation optimale de certains mécanismes d'apprentissage, c'est-à-dire visant en définitive à apprendre plus de choses en moins de temps.

cela favoriser un dialogue avec une machine fondé sur l'imitation du dialogue entre un précepteur et un apprenant ? Les différentes réponses à cette question alimentent de nombreuses recherches.

Si le rôle central des rétroactions a bien été mis en évidence, ces dernières ne sont pas forcément de nature langagière mais peuvent, par exemple, consister dans la modification d'une image affichée à l'écran. De plus, le traitement du langage naturel par un ordinateur s'avère beaucoup plus complexe qu'on avait pu le croire. Au lieu d'essayer de rendre des formes linguistiques en quelque sorte exécutables, par la construction de programmes aptes à *comprendre* le langage naturel, il va s'agir de traiter des formes visuelles dont le fonctionnement réfère à des objets concrets, ou tout au moins connus de l'utilisateur, ce qui lui permet de les piloter. Le développement de telles idées, souvent conçues pour des finalités éducatives, favorise l'émergence des interfaces graphiques, popularisées dans les années quatre-vingt. Les ordinateurs, désormais souvent individuels, vont alors être utilisés comme des outils pour résoudre des problèmes ou effectuer des tâches spécifiques, voire pour jouer. Par contrecoup, la notion de micromonde se généralise peu à peu et perd une partie de ses spécificités éducatives.

Ce chapitre essaye de rendre compte des différents courants qui viennent d'être évoqués. Partant de l'apprentissage par la programmation et de la notion de micromonde, ainsi que des évaluations effectuées, par ailleurs essentiellement autour de LOGO, nous arriverons aux environnements d'apprentissage qualifiés d'ouverts, c'est-à-dire laissant complètement l'initiative à l'apprenant. Toutefois, l'abandon de toute forme de contrôle par la machine ne va pas donner les résultats escomptés. Les chercheurs vont tenter de définir des compromis entre liberté totale de l'apprenant et contrôle strict de son activité. C'est ce que nous verrons au chapitre suivant.

## **4.2. L'apprentissage par la programmation**

### **4.2.1. Le développement de LOGO**

La genèse de LOGO a déjà été maintes fois décrite (Papert, 1980, 1993, p. 171, Ross, 1986 ; Lawler, 1987...). Au début des années soixante, Wallace Feurzeig a l'idée d'utiliser des langages de programmation de type conversationnel pour améliorer l'enseignement des mathématiques. Il imagine que la programmation interactive rend possible la communication de puissantes intuitions mathématiques aux étudiants. Après des essais encourageants avec un langage destiné aux ingénieurs, Feurzeig pense obtenir de bien meilleurs résultats avec un langage spécifiquement conçu à cette fin. Au sein du laboratoire privé BBN (Bolt, Beranek and Newman, Inc.) sous l'impulsion de Bobrow et Solomon, il travaille alors avec Papert pour concevoir un tel langage. Ce langage nommé LOGO, dialecte du LISP,

langage de prédilection pour les recherches en intelligence artificielle, a donc initialement été conçu dans le but d'aider les élèves à développer une compréhension plus profonde d'idées mathématiques jugées importantes.

Partant du constat que les méthodes mathématiques formelles demeurent pour la plupart des élèves mystérieuses et artificielles, Feurzeig et Papert (1968) pensent que l'expérience de la programmation peut contribuer au franchissement de cette *barrière formelle* et favoriser les relations entre procédures formelles et compréhension intuitive des problèmes. Illustrant les préceptes heuristiques décrits par Polya (formuler un plan, diviser les difficultés, etc.), la programmation peut amener les élèves à comprendre de façon précise un certain nombre de concepts-clés. Faire travailler les étudiants dans une sorte de *laboratoire mathématique* peut les aider à comprendre leurs propres processus de pensée en verbalisant leurs expériences. Feurzeig et Papert prennent comme exemple les différents rôles que peut jouer la lettre  $x$  en mathématiques (nombre, variable, fonction, variable quantifiée universellement dans l'expression  $(x + 1)^2 = x^2 + 2x + 1$  et existentiellement quantifiée dans une équation telle que  $2x + 1 = 5$ ) induisant des complexités difficilement traduisibles dans un monde familier. Dans la programmation, les distinctions apparaissent concrètement, on doit alors y faire face et la nature physique de la machine fournit une meilleure référence concrète qu'une quelconque tâche abstraite. Le premier travail de Feurzeig et Papert consiste à réaliser une progression dans le domaine de l'algèbre (tâche poursuivie essentiellement par Feurzeig<sup>3</sup>).

Rapidement, s'intéressant à des apprenants de plus en plus jeunes, au but initial limité aux mathématiques s'ajoutent des objectifs d'apprentissages d'aptitudes de pensée plus générales à l'aide de la programmation. S'appuyant sur les idées de Dewey, Montessori et Piaget, Papert pense que les enfants apprennent en faisant et en réfléchissant à ce qu'ils font (Papert 1970, p. 61). C'est l'enfant lui-même, à travers ses expériences, qui est le moteur de l'apprentissage, constructeur de son propre savoir. Il s'agit de le motiver suffisamment en l'engageant dans des activités de nature expérimentale (anticiper un résultat, être surpris, deviner une loi générale...). En s'inspirant d'un robot conçu par Grey Walter dans le cadre de recherches en cybernétique, Papert adjoint la fameuse tortue au LOGO afin de rendre l'ordinateur plus intéressant pour les enfants. Il décrit cet apport fondamental (Papert, 1993, p. 175) comme un moyen de capturer dans une forme computationnelle quelque chose de physique, analogue au fait de marcher et de dessiner. La programmation peut alors conduire les apprenants à une réflexion plus explicite et mieux articulée sur leurs propres processus cognitifs et en cela affecter favorablement leur développement cognitif. Une telle hypothèse, selon Lawler (1987), s'appuie sur trois importantes caractéristiques de LOGO :

- la capacité de créer de nouvelles procédures d'une façon interactive,

---

3. Nous le verrons dans le prochain chapitre (Feurzeig, 1985).

- la capacité pour l'enfant de simuler le mouvement de la tortue avec les mouvements de son propre corps et d'assimiler ainsi des connaissances sur la géométrie tortue à partir de ses propres connaissances sensori-motrices<sup>4</sup>,
- le fait que ces procédures, parce que l'apprenant les crée lui-même, représentent concrètement ses propres pensées.

L'important travail effectué au sein du laboratoire LOGO du MIT (*Massachusetts Institute of Technology*), dans le courant des années soixante dix, conduit à l'émergence de la notion de micromonde (développée dans la section suivante).

Si ces recherches apparaissent en rupture vis-à-vis de l'enseignement programmé, elles s'intègrent dans la mouvance des premières applications de l'intelligence artificielle à l'éducation. Ainsi, dans un cours destiné à des étudiants en sciences sociales, John Seely Brown (Brown, 1972), abondamment cité dans le chapitre précédent pour son rôle majeur dans le développement de l'EIAO (SOPHIE, WEST, BUGGY...), utilise LOGO et la programmation fonctionnelle récursive comme des outils conceptuels. De même, un livre d'introduction à l'intelligence artificielle écrit par Alan Bundy (Bundy, 1982) montre l'intérêt de LOGO et son lien fort avec au moins une partie de la communauté des chercheurs en intelligence artificielle. En fait, l'un des apports essentiels de LOGO est de montrer qu'il est possible pour de jeunes enfants de piloter un ordinateur et ce résultat va stimuler l'imagination des chercheurs.

LOGO est introduit en France au milieu des années soixante-dix, notamment sous l'impulsion de Perriault (qui ramène la première tortue en France en 1972) et de Bossuet. Plusieurs institutions jouent un rôle majeur : l'Ofrateme, les Irem (Instituts de recherche sur l'enseignement des mathématiques) et l'Université Paris VIII. A la fin des années soixante-dix, leurs efforts sont coordonnés au sein d'une RCP (Recherche coordonnée sur programme)<sup>5</sup> et le premier colloque se tient à Clermont-Ferrand en décembre 1982. Notons que, dans le contexte français, ces chercheurs ont essayé de faire exister une alternative au LSE (Langage symbolique d'enseignement), alors dominant et quelque peu hégémonique.

---

4. « Nous offrons à l'enfant un milieu où la tâche n'est pas d'apprendre des règles formelles, mais d'acquérir suffisamment d'intuitions, de perceptions intérieures, sur la manière dont on se déplace dans l'espace pour permettre la transposition de ce savoir intime en programmes qui conféreront le mouvement de la tortue. » Papert (1980, p. 255). C'est l'idée de *syntonie corporelle* : permettre aux enfants de réintroduire leur propre corps dans les mathématiques.

5. Il faudrait retracer plus précisément l'historique de l'introduction de LOGO en France. Au risque d'oublier quelques personnes, on peut citer Perriault, Boudinot, Robert, Le Touzé, Blondel du côté de l'Ofrateme ; cinq IREM ont été en pointe : Orléans (Rouchier), Dijon (Delannoy), Paris (Bossuet), Paris Nord (Bourbion), Le Mans (Vivet) ; enfin, à l'université Paris VIII, Pérolat, Greussay, Tanguy et Wertz utilisent un dialecte de LISP dans un esprit similaire à LOGO. L'expérience d'Arc et Senans en 1979 constitue une étape importante et le rapport Simon (1981) sur l'éducation et l'informatisation de la société consacre une place à l'apprentissage autonome.

#### 4.2.2. *SMALLTALK et le DYNABOOK*

La voie ouverte par LOGO est suivie par une équipe autour d'Adèle Goldberg et Alan Kay, dans le cadre du groupe de recherche sur l'apprentissage à Xerox Parc, et conduit notamment à la conception du langage SMALLTALK.

Alan Kay (1984) crédite aux travaux de McLuhan sa compréhension de l'ordinateur comme médium et non comme simple outil voire comme support.

« L'ordinateur est protéiforme et peut agir comme une machine ou comme un langage que l'on façonne et exploite ; un ordinateur peut simuler de façon dynamique les détails de tout autre instrument, y compris ceux qui n'ont pas d'existence physique. Ce n'est pas un outil, bien qu'il puisse se comporter comme tel. C'est le premier "métamédia" dont les degrés de liberté nous confèrent des possibilités de représentation et d'expression qui n'ont encore jamais été rencontrées avant et ont rarement été étudiées. Et plus important, c'est plaisant et ainsi intrinsèquement intéressant. »

Kay et Goldberg (Kay et Goldberg, 1977 ; Goldberg, 1979) imaginent un dispositif qu'ils nomment DYNABOOK. Sorte d'ordinateur-carnet, conçu pour « graver la métaphore de McLuhan dans le silicium », il cristallise leur rêve de conception d'un médium dynamique personnel<sup>6</sup> pour la pensée créative. Ils le comparent à une flûte, instrument qui est la propriété de son utilisateur et répond instantanément et de manière cohérente à ses souhaits<sup>7</sup>. Il sert à matérialiser les idées et, grâce à la rétroaction, à baliser les pistes suivies par la pensée. On retrouve les idées à la base de LOGO : les mathématiques deviennent un langage vivant avec lequel les enfants peuvent déclencher des phénomènes intéressants. D'après Kay et Goldberg, le DYNABOOK constitue une sorte de nouveau monde pour les éducateurs, uniquement limité par leur imagination et leur ingéniosité. Il offre ainsi la possibilité : de montrer des relations historiques complexes, impossibles à mettre en évidence dans des livres statiques et linéaires ; de simuler des expériences de laboratoire autrement trop onéreuses ou trop complexes à installer ; d'aider à la production de poésies et de textes en prose en fournissant des facilités d'édition...

Dans un premier temps, le langage SMALLTALK est implanté sur des petits ordinateurs. Son rôle est de fournir un outil de programmation et de résolution de problèmes, une mémoire interactive pour la conservation et la manipulation de données, un éditeur de textes ainsi qu'un médium d'expression à travers le dessin, la

---

6. Le DYNABOOK concrétise aussi des idées développées par Bush (voir chapitre 6) dans le sens d'une machine à usage personnel.

7. C'est le terme d'*ordinateur crayon* qui décrit le mieux comment, à l'avenir, les enfants devraient se servir de l'ordinateur. On se sert d'un crayon tant pour griffonner que pour écrire, tant pour gribouiller que pour dessiner, tant pour les billets passés en douce que pour les devoirs imposés. (Papert 1980, p.261).

peinture, les images animées, la composition et la génération musicale. Durant le déroulement du projet, plusieurs étapes, au niveau matériel et logiciel, se succèdent. Chacune d'elles correspond à ce que Kay et Goldberg nomment un DYNABOOK transitoire, permettant de réaliser diverses études expérimentales avec des enfants et de créer des situations de laboratoire.

Goldberg (1979) souligne l'écart fantastique entre les processus généraux de conception/réalisation et la programmation conventionnelle (dans le sens de la connaissance et l'application de suites linéaires et conditionnelles, la liaison et la nomination de la gestion de données d'entrée-sortie). SMALLTALK autorise une flexibilité importante et permet au DYNABOOK d'offrir de grandes classes d'application : des programmes d'enseignement de type EAO incluant simulations et jeux, l'interaction avec un tuteur ou un « moniteur » pour une tâche dirigée (EIAO), un contrôle du médium lui-même par la programmation.

Le type d'accès à l'ordinateur préconisé implique à la fois la disponibilité d'applications et la mise à disposition de la machine, à l'endroit et au moment où on le désire. Dans le cadre de la programmation proprement dite, il est possible d'explorer différents paradigmes :

- *programmation par l'exemple* : réalisation d'animations à l'ordinateur,
- *programmation par la spécification de contraintes* : THINGLAB (Borning, 1977), laboratoire de création de simulations, permet de traiter des interactions complexes traduites par des contraintes entre différentes parties d'une structure. Son intérêt réside dans l'élaboration d'hypothèses sur des relations possibles ou probables entre les éléments, par la description d'un modèle conçu à partir de ces relations. C'est notamment intéressant pour imaginer des conjectures en géométrie<sup>8</sup> ou pour explorer certains concepts en physique,
- *programmation par la description de règles* (découverte de lois cachées dans un jeu),
- *programmation par la description d'un algorithme* : un kit de simulation est ainsi fourni.

Pour les concepteurs, chaque DYNABOOK transitoire fournit l'occasion d'étudier l'impact de différentes formes de représentation pour manipuler des informations picturales ou textuelles ou les décrire pour effectuer une recherche électronique.

Le logiciel TRIP (Gould et Finzer, 1981) permet ainsi de réaliser des animations correspondant à des problèmes simples sur les notions de temps, de vitesse et de distance. Il est destiné à des élèves capables de résoudre des équations algébriques mais éprouvant des difficultés à traduire des énoncés en équations. Pour traiter un problème, l'élève dispose de l'énoncé, d'un embryon de diagramme et d'un ensemble d'icônes. En déplaçant ces icônes sur le diagramme à l'aide d'une souris, l'élève a pour tâche de construire une représentation imagée du problème. On lui demande en

---

8. Nous y reviendrons à la fin de ce chapitre dans la partie consacrée à la géométrie.

outre d'estimer une solution et, sur la base des éléments ainsi fournis, une animation est déclenchée. Cette dernière étape peut être répétée afin de permettre à l'élève de converger vers une solution. Les différentes estimations et leurs conséquences sont enregistrées dans un tableau à partir duquel l'apprenant induit une équation algébrique décrivant le problème à résoudre. La figure 4.1 correspond à l'écran final de l'élève à l'issue de son travail avec le logiciel TRIP.

Le train A quitte L.A. à une heure et roule à 75 miles par heure jusqu'à S.F. situé à 400 miles de L.A. A deux heures, le train B quitte S.F. et roule vers L.A. à 50 miles par heure. A la même heure, un contrôleur remarque qu'un aiguillage n'a pas fonctionné et que les deux trains roulent sur la même voie. Combien de temps a-t-il pour éviter une collision ?

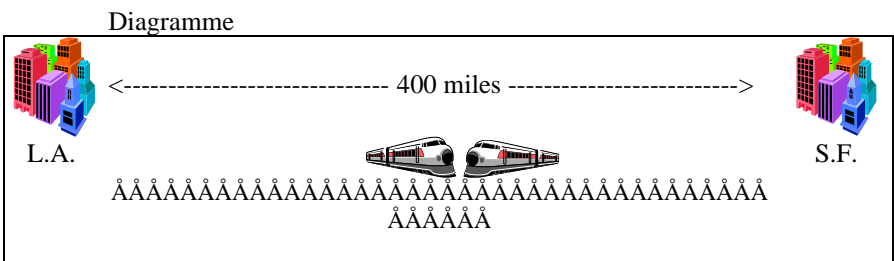


Table de trajet	Départ	Vitesse	Distance	Temps
A	1.00	75 mph	270.0 miles	3 heures 36 minutes
B	2.00	50 mph	130.0 miles	2 heures 36 minutes

Table d'essais	Train B Temps	Train A Temps	Train A Distance	Train B Distance	A + B Distance	
Essai 2	2 h 45 mn	3 h 45 mn	281.2 miles	137.5 miles	418.7 miles	
Essai 3	2 h 35 mn	3 h 35 mn	268.7 miles	129.2 miles	397.7 miles	
Essai 4	2 h 36 mn	3 h 36 mn	270.0 miles	130.0 miles	400.0 miles	
Algèbre	t	t + 1	75 (t + 1)	50 t	75(t+1)+ 50 t	75 (t+1) + 50 t = 400

**Figure 4.1.** Ecran final de TRIP (adapté de Gould et Finzer, 1981)

TRIP ne dispose d'aucune connaissance du domaine et n'intègre aucun modèle de l'élève. L'équation finale doit être spécifiée à l'avance. En revanche, TRIP offre des facilités d'aide et de vérification. Son objectif, selon les auteurs, est développer des aspects intuitifs en résolution de problèmes. Une petite expérimentation a indiqué que TRIP aidait les apprenants à représenter des problèmes de manière imagée sans



toutefois améliorer de façon notable leurs performances. Une difficulté est par ailleurs apparue : les élèves, parfois, ne savaient pas où regarder durant le déroulement d'une séquence animée. Les auteurs ont suggéré différentes améliorations pour faire de TRIP un instrument d'expérimentation pour la recherche en éducation, mais ce système ne semble pas avoir été repris.

Même si les travaux autour de SMALLTALK n'ont pas eu la diffusion extraordinaire du LOGO dans les communautés éducatives<sup>9</sup>, ils occupent une place importante du fait de leur empreinte sur le développement de l'informatique, plus particulièrement sur les interfaces et les langages à base d'objets. Ils ont certainement eu une influence considérable sur la conception des environnements informatiques popularisés dans les années quatre-vingt.

#### **4.2.3. Langages et environnements de programmation**

Pour mieux situer la place de LOGO vis-à-vis d'autres usages éducatifs de l'ordinateur, l'analyse de Cynthia Solomon est éclairante (Solomon 1985, également citée dans Linard 1990, p. 164-169). Elle distingue deux grandes classes : l'ordinateur comme un *livre interactif* contrôlant l'utilisateur et l'ordinateur comme un *moyen d'expression* sous contrôle de l'utilisateur. Elle décrit quatre exemples correspondant à des paradigmes généraux, dans le domaine des mathématiques élémentaires :

1. Suppes, de l'université de Stanford, développe des exercices (*drill and practice*) pour un apprentissage par cœur dans une vision comportementaliste. Certains de ses travaux ont été présentés au chapitre 2.

2. Davis, université de l'Illinois, est intéressé par l'interaction socratique et l'apprentissage par la découverte. Contrairement à Suppes, il considère que les représentations graphiques et visuelles sont très importantes et font partie intégrante des logiciels qu'il destine aux enfants. Le projet Madison, développé dans le cadre de PLATO, a conduit à la réalisation de nombreux environnements intéressants.

3. Dwyer, de l'université de Pittsburgh, s'appuie sur une métaphore fondée sur sa propre expérience de pilote (du vol accompagné au vol en solitaire). Dans le cadre du projet Solo, il développe un modèle basé sur des considérations culturelles dans lequel les personnes, les idées et les technologies avancées interagissent pour former une « communauté d'apprentissage » (Dwyer, 1976).

4. Papert adopte une approche constructiviste, apprentissage de type piagétien dans ce qu'il appelle le *pays des mathématiques*.

---

9. Malgré les efforts de différents chercheurs, notamment d'Isabelle Borne et de Pierre Cointe en France.

Solomon souligne que les quatre ont des conceptions différentes concernant les compétences de base en mathématiques et s'appuient sur des théories de l'apprentissage différentes. Les approches suivies par Suppes et Davis entrent dans le cadre de l'usage de l'ordinateur comme livre interactif et ne nous intéressent pas directement dans ce chapitre. En revanche, selon Dwyer et Papert, la meilleure expérience d'apprentissage est d'apprendre à dominer l'ordinateur. Solomon (op.cit., p. 100) critique néanmoins l'approche de Dwyer sur l'inconsistance entre la vision de l'environnement qu'il souhaite et les outils qu'il choisit pour le construire, qu'elle qualifie de faciles à apprendre mais difficiles à utiliser. En effet, Dwyer détourne des outils créés pour d'autres fins que des finalités éducatives. En particulier, il adopte le langage BASIC parce qu'il est facilement disponible. Que l'on soit ou non d'accord avec Solomon, la question du choix des outils ou d'un langage dans le cadre d'un apprentissage par la programmation apparaît essentielle.

Les langages informatiques, en eux-mêmes, n'ont pas de vertus éducatives particulières, mais ils respectent certaines contraintes ou intègrent certaines caractéristiques permettant de développer des activités qui, elles, sont intéressantes du point de vue éducatif. C'est bien dans des façons spécifiques d'exploiter les possibilités offertes par ces langages que l'on peut espérer développer des capacités souhaitées chez les apprenants. Sans qu'il soit possible de fournir des arguments définitifs, il est cependant possible de proposer certaines caractéristiques qui apparaissent importantes pour faciliter un apprentissage par la programmation.

Tout d'abord, ces langages doivent être faciles à apprendre et faciles à utiliser, ce qui induit une certaine proximité avec une classe de problèmes, mais peut constituer par ailleurs une limitation s'il est nécessaire d'apprendre une nouvelle syntaxe pour aborder une nouvelle classe de problèmes. Ensuite, l'*interactivité* et l'*extensibilité* (c'est-à-dire la possibilité d'enrichir le langage de base, de travailler sur des couches successives<sup>10</sup>) semblent être importantes. L'apprenant doit pouvoir contrôler la conséquence de ses actions, suivre une démarche naturelle et contrôlable en utilisant une syntaxe proche de la langue naturelle. Une liste de problèmes et d'activités intéressants doit pouvoir être proposée et la structure du langage considéré doit correspondre aux objets manipulés. Enfin, au-delà de certaines facilités de mise en œuvre, le langage doit offrir une puissance suffisante, en intégrant des concepts efficaces de programmation.

En LOGO, chaque commande a un effet immédiat et visible, la syntaxe utilisée est relativement proche de la langue naturelle et les messages d'erreurs sont explicites. Si on s'accorde avec les caractéristiques précédemment évoquées, LOGO semble constituer un bien meilleur choix que BASIC. En outre, sur un plan plus technique, si on se focalise sur l'apprentissage des mathématiques, LOGO offre d'autres atouts qui

---

10. Cette possibilité est classiquement utilisée en intelligence artificielle. Pour résoudre des problèmes complexes on se crée les outils et les langages permettant de formuler ces problèmes d'une manière plus agréable.

peuvent être jugés intéressants comme le fait que ce langage soit non typé, qu'il utilise des opérateurs infixes, qu'il permette la distinction entre variable, mot et procédure avec des procédés syntaxiques, qu'il intègre le traitement de listes et la récursivité.

Dans l'apprentissage par la programmation, les langages utilisés sont un moyen, le but étant de construire quelque chose. Il s'agit, pour l'apprenant, de traduire ses intuitions sous la forme d'un programme. Il convient de trouver une forme de communication entre l'élève et la machine qui ne perturbe pas trop les apprentissages visés. On peut penser à une interface transparente (dans le sens immédiat, sans obstacle) ou suffisamment cohérente avec les objets manipulés. L'expression de Papert, « outil pour penser avec » correspond bien à cette dernière idée. L'apprentissage par la programmation ne peut cependant se passer d'un certain apprentissage de la programmation. Une connaissance minimale est en effet indispensable pour disposer des capacités d'expression suffisantes : savoir utiliser les outils, comprendre le but, interpréter les erreurs et savoir les corriger. Un compromis est à trouver entre l'apprentissage de l'outil et les services qu'il peut rendre. Mais, dans une pédagogie de projet, ce qui semble, a priori, un inconvénient peut s'avérer un avantage. Acquérir des connaissances par la programmation peut s'accompagner ou engendrer de nouveaux savoirs et savoir-faire en programmation. L'ordinateur est un dispositif technique ouvert qui incite les élèves à pousser leurs connaissances à la limite pour améliorer leurs projets à travers une variété illimitée d'*effets*. Comme le souligne Papert (1993, p. 69) le fait d'apprendre plus sur les techniques informatiques devient partie prenante du projet, ce qui arrive difficilement avec le papier et le crayon.

En dehors du LOGO lui-même, d'autres environnements peuvent être de bons supports à l'apprentissage par la programmation. C'est le cas des extensions de langages de programmation traditionnels, comme les extensions de LOGO et plus particulièrement celles à base d'objets (Borne, 1983 ; Drescher, 1987), SMALLTALK (Goldberg 1979 ; Borne et Lemoyne, 1987), PROLOG (Ennals, 1984 ; Nichol, 1988)... Des environnements de programmation, censés être plus abordables pour des utilisateurs débutants, permettent l'introduction de nouvelles métaphores, comme les métaphores spatiales pour BOXER (Di Sessa, 1985), ou d'autres paradigmes de programmation comme la programmation par l'exemple pour TINKER (Lieberman, 1987).

Aucun d'entre eux n'a cependant rencontré un succès (en terme de diffusion) comparable à celui de LOGO. En effet, si dans les années soixante-dix, la programmation apparaissait comme quelque chose de nouveau et d'intéressant, voire d'indispensable (la nécessité de connaître l'informatique se traduisait par la maîtrise de la programmation), l'essor des outils généraux au milieu des années quatre-vingt a consacré pour beaucoup le déclin, puis le rejet de la programmation. Le prestige de la pensée informatique, qui apparaissait féconde, s'est peu à peu étioilé. Mais,

derrière l'idée, un peu trop vague, d'apprentissage par la programmation, se cachent des notions sans doute plus profondes, notamment celle de micromonde, qui ont accompagné l'évolution de l'informatique.

### 4.3. Les micromondes

Dans cette section nous allons préciser la notion de micromonde en en retraçant brièvement la genèse telle qu'elle est décrite par Lawler (1987), et en explicitant les concepts principaux qui y sont attachés. Nous présenterons ensuite divers exemples. Remarquons cependant qu'il n'y a pas de définition patentée du concept de micromonde. Plusieurs auteurs (Lawler, 1987 ; Thompson, 1987 ; Laborde et Laborde, 1991 ; Hoyles et Noss, 1993...) ont apporté leur contribution, reprochant souvent une utilisation trop lâche de cette notion dans la littérature, mais sans réellement aboutir à une acception définitive. Il faut certainement se résoudre à approcher cette notion de micromonde, sans espérer l'enfermer dans une définition parfaitement circonscrite. Elle se transforme au gré de l'évolution générale de l'informatique, ce qui atteste de son aspect dynamique et fécond. Pour simplifier, nous réserverons l'expression micromonde transitionnel pour désigner les micromondes conçus dans l'esprit des premiers travaux autour de LOGO que nous allons tout d'abord présenter.

#### 4.3.1. La notion de micromonde transitionnel

L'origine du mot *micromonde* remonte à Winograd (1972) et son programme nommé SHRDLU, qui opère à l'intérieur d'un domaine miniature (un micromonde) et simule le guidage d'un bras robot manipulant des cubes sur une table<sup>11</sup>. Dans le contexte de l'usage éducatif des ordinateurs, ce mot a été introduit par Minsky et Papert dans un rapport interne du MIT (Minsky et Papert, 1972). Il se rapproche du terme microcosme (Lawler, 1987) dans le sens d'une image réfléchissante du macrocosme. Il a été popularisé par le livre de Papert (1980) pour désigner des domaines conçus avec une intention génétique<sup>12</sup>.

L'idée essentielle sous-jacente à la notion de micromonde est de créer un environnement, respectant différentes contraintes de construction, avec lequel d'autres personnes pourront exercer leur créativité. Lawler caractérise les micromondes comme des *mondes virtuels pour l'action créative*, dont les objets ont

---

11. La méthodologie utilisée en intelligence artificielle consistant à travailler sur des réductions de la réalité, abstraites de leur domaine supposé d'application a été critiquée (Dreyfus, Winograd...). En effet, cette démarche conduit à des simplifications souvent trop importantes. Les micromondes en éducation correspondent en fait à une démarche sensiblement différente.

12. Le chapitre 5 en particulier s'intitule "Micromondes : des incubateurs de savoir"

des propriétés communes à la fois avec les objets formels de la science et les objets plus concrets de l'expérience sensible. Papert (1987) les décrit comme des objets qui d'une certaine façon sont semblables à ceux avec lesquels on travaille dans le monde réel et, d'une autre façon, sont semblables à des objets abstraits. Ce sont des *objets transitionnels*<sup>13</sup> qui aident à manipuler les objets abstraits, des passerelles entre les apprentissages intuitifs et formels. Lawler les associe à sa théorie de la connaissance basée sur l'idée de réseau de micro-vues<sup>14</sup>. Dans le cas des mathématiques, Thompson (1987) précise qu'un micromonde fournit une *sémantique dynamique à un système formel*, constituant le cœur d'un système axiomatique intuitif dans lequel les élèves peuvent définir de nouveaux objets et opérations et étudier interactivement leurs propriétés. Un micromonde donne corps à la structure d'un concept et la tâche de l'apprenant consiste à intégrer mentalement cette structure pour se l'approprier<sup>15</sup>. La métaphore de Papert du « pays des mathématiques » illustre l'idée d'un monde à la fois familier et étrange que l'on va pouvoir peu à peu explorer et domestiquer et auquel on peut attribuer un sens par référence à sa propre expérience.

A l'aide d'un micromonde, on tente d'établir un lien sémantique fort entre le formel (ou l'abstrait) et le réel (simulé ou de référence) en garantissant la conservation du sens grâce aux connaissances de l'apprenant sur le fonctionnement des objets réels. La figure 4.2 présente schématiquement les différents éléments constitutifs d'un micromonde transitionnel.

Le monde réel ou de référence et le monde formel sont reliés par un processus de modélisation ou d'exemplification. Les objets visibles à l'interface se comportent d'une manière cohérente avec le modèle formel sous-jacent et leur aspect externe, c'est-à-dire à l'interface, rappelle les objets du monde réel. En fait, les micromondes se fondent sur l'idée de l'existence de modèles mentaux, donnant du sens aux activités, et sur la possibilité de faire évoluer ces modèles par expérience dans les environnements proposés. Un modèle mental est vu ici comme une représentation dont les objets symboliques se comportent d'une manière similaire aux objets dans les situations où ils sont représentés. D'après Greeno (1990), de tels modèles

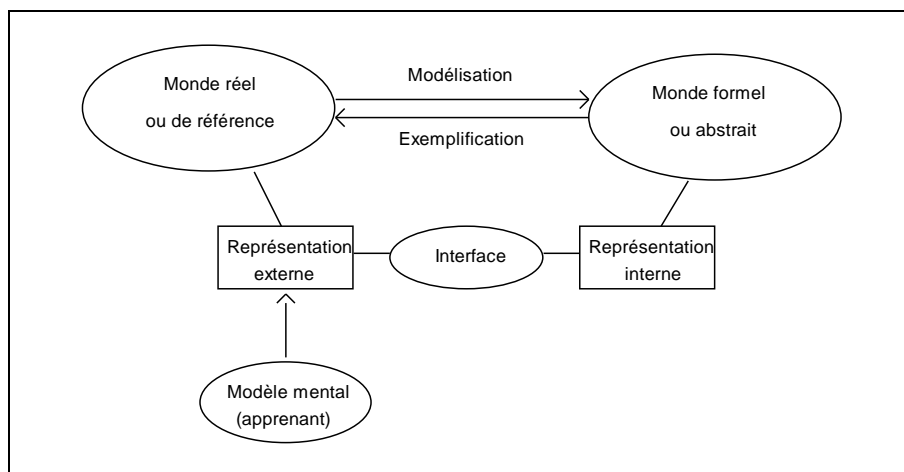
---

13. Voir l'analyse de Monique Linard, op. cit., p. 151-160. Nous ne considérons pas l'interprétation psychanalytique de la notion d'objet transitionnel que l'on peut consulter dans le même ouvrage.

14. Lawler (1987) distingue les minimondes, environnements informatiques conçus à partir de la description d'un domaine par un expert, des micromondes qui réfèrent à des exploitations partielles de ces minimondes, c'est-à-dire à des ensembles d'activités qu'il est possible d'y développer, correspondant à des micro-vues pour l'apprenant.

15. D'après Papert (1987), si on peut déjà manipuler des blocs grâce à la technologie du bois, jusqu'à présent, on ne peut manipuler des symboles que dans la tête ou au travers des moyens abstraits offerts par le papier et le crayon. On ne dispose encore d'aucune façon satisfaisante de rendre externe la manipulation des symboles.

mentaux permettent de développer un raisonnement génératif à propos de nouvelles situations ayant une structure significative.



**Figure 4.2.** Schéma général d'un micromonde transitionnel

D'un point de vue opératoire, les micromondes désignent des mondes artificiels dans lesquels on agit sur des objets, dont le comportement respecte certaines contraintes de fidélité et de cohérence. Pour Groen (1984), un micromonde est une structure (ensemble d'états et de transformations décomposables entre ces états) munie de certaines propriétés, les deux plus importantes étant :

1. la possibilité d'annuler une transformation pour retourner à l'état précédent,
2. l'existence d'applications (au sens mathématique du terme) sur d'autres structures qui sont des représentations d'actions concrètes dans le monde réel.

Comme on peut le constater, l'approche micromonde est fortement éloignée des préoccupations courantes de l'EAO (Enseignement Assisté par Ordinateur). Alors que ce dernier s'intéresse plutôt aux faits et aux aptitudes, la finalité des micromondes mathématiques est dans la construction d'une signification et de relations entre des objets mathématiques. Les apprenants doivent construire des objets et des relations entre ces objets, puis considérer ces relations comme de nouveaux objets pour opérer dessus. Les micromondes sont conçus pour faciliter ce processus et adoptent souvent une approche en couches successives (Thompson, 1987). Cette capacité d'évolution conjointe du micromonde et de son utilisateur, c'est-à-dire le fait que le micromonde se développe au fur et à mesure que la connaissance de l'élève évolue, est souvent jugée essentielle (Balacheff, 1994).

Il y a ici une nette rupture avec l'EAO sur le plan des théories psychologiques sous-jacentes. Papert s'inspire des travaux de Piaget et s'inscrit dans les recherches sur l'intelligence artificielle. Rappelons que Piaget décrit les constructions mathématiques comme des “*abstractions réfléchissantes*”, processus permettant le passage des opérations concrètes aux opérations hypothético-déductives. La conception des micromondes en est fortement inspirée. Ainsi, Hoyles et Noss (1993) définissent un cadre général pour la construction de micromondes comme suit :

- offrir un ensemble d'outils logiciels soigneusement conçus, à la fois fonctionnels et amusants
- engager les élèves et les inciter à distinguer la façon dont les outils ont été construits et les relations entre les rétroactions de la machine et leurs propres entrées afin qu'ils puissent s'appropriier les caractéristiques mathématiques locales incorporées dans les micromondes
- structurer les activités de telle sorte qu'elles favorisent la généralisation et la synthèse d'idées dans un cadre mathématique cohérent.

Déterminer un ensemble structuré d'activités apparaît essentiel pour que les micromondes puissent jouer un rôle éducatif. Une telle nécessité amène toutefois une tension entre une structuration imposée et l'exploration libre souhaitable pour favoriser la prise de risque de la part des apprenants et le développement de leur autonomie. Les concepteurs et utilisateurs de micromondes vont devoir tenir compte de cette tension, laisser un temps d'exploration suffisant dans le cadre d'activités, pas entièrement planifiées mais soigneusement régulées.

#### **4.3.2. Caractéristiques et exemples**

L'exemple de référence de la notion de micromonde transitionnel est certainement celui de la géométrie tortue, associé au langage LOGO (Papert, 1980). Dans le cadre mathématique, il permet une première approche d'une géométrie locale (notion de déplacements et d'angles de rotation). LOGO, en tant que langage de programmation, a permis la réalisation de nombreux micromondes, une bonne partie d'entre eux se rapportant aux mathématiques élémentaires. Dans ces dernières applications, l'enjeu est l'approche sémantique d'un système formel à partir de composantes exemples isomorphes (numération, structures additives, problèmes arithmétiques, algèbre élémentaire...).

##### *4.3.2.1. Quelques exemples de micromondes*

Le travail de Pearla Nesher (1989), sur l'apprentissage de la notion de nombre, rentre tout à fait dans ce cadre<sup>16</sup>. Rappelons que les nombres sont des objets

---

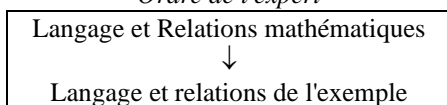
16. Elle dit préférer éviter le terme micromonde, non rigoureusement défini et utilisé de manière trop lâche et qualifie l'environnement qu'elle a développée de système d'apprentissage (*learning system*).

abstrait, alors que les mots *nombre* sont des qualifieurs quantitatifs d'objets d'une classe. L'objectif est d'acquérir une certaine compréhension de cette notion abstraite de nombre. Le système d'apprentissage conçu par Nesher inclut deux composantes :

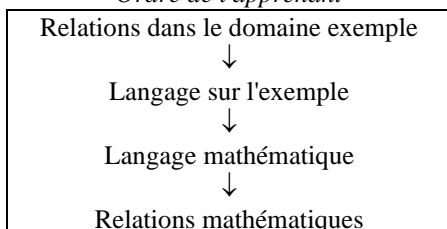
1. la composante connaissance, c'est-à-dire la connaissance de l'expert (adaptée aux apprenants),
2. la composante d'exemplification, correspondant à un domaine illustratif, isomorphe à la composante connaissance et dont le rôle est de servir d'exemple.

Pour que la composante d'exemplification remplisse son rôle, le domaine choisi doit être familier à l'apprenant. Ce dernier doit être acclimaté aux objets exemples, être capable d'utiliser un langage familier pour décrire et communiquer les relations entre ces objets et saisir intuitivement les choses vraies. Mais la seule familiarité est insuffisante. Le choix de la composante exemple doit assurer une correspondance à la composante connaissance à enseigner, qui elle n'est pas encore connue de l'enfant et constitue l'enjeu de l'apprentissage. Si les exigences de familiarité et de nouveauté peuvent apparaître contradictoires, les aspects familiers servent d'ancres pour développer une compréhension d'un nouveau système de relations et un nouveau langage.

*Ordre de l'expert*



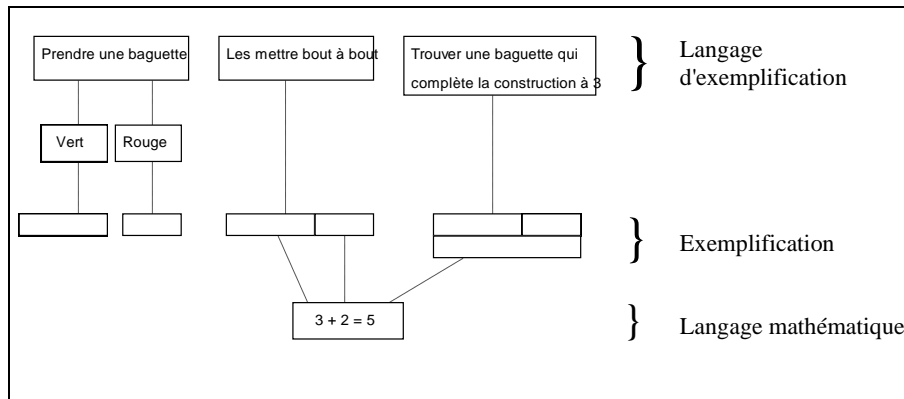
*Ordre de l'apprenant*



Dans le passage entre la composante connaissance et la composante d'exemplification, Nesher (op.cit. p. 199) distingue l'ordre de l'expert de celui de l'apprenant. L'expert s'appuie sur les relations et le langage mathématique et en cherche des équivalents dans la composante exemplification. Le cheminement de l'élève est tout à fait différent. Il part de la composante exemple. De nouvelles configurations et relations basées sur l'exemple lui sont décrites dans une langue familière ordinaire. Ensuite, le langage mathématique est utilisé pour

décrire les nouvelles configurations dans l'espoir que, en combinant l'usage de ce langage formel et l'environnement qu'il comprend, il puisse se construire une image mentale des objets et des relations mathématiques. Nesher précise que ce dernier stade n'est jamais bien assuré et ne peut être facilement testé. Seul l'usage correct et aisé du langage mathématique peut être observé. Les composantes d'exemplification jouent ainsi le rôle d'interprétation sémantique du modèle formel (ce sont en quelque sorte des instances d'un système formel).





**Figure 4.3.** Une exemplification de " $3 + 2 = 5$ "

Concernant la numération et l'addition, plusieurs systèmes, tels ARITHMEKIT (Brown, 1985), DIENES block (O'Shea et al., 1988) et Blocks Microworld (Fraivillig et al., 1993), s'appuient sur une transposition informatique de la manipulation de blocs de Diènes. L'objectif est de faciliter le transfert entre la manipulation des blocs et l'arithmétique papier/crayon, essentiellement la numération de position et les algorithmes d'addition et de soustraction sur les entiers. La même équipe autour de O'Shea a conçu SHRINK-A-CUBE pour aider les enfants à développer et comprendre les principes de l'échange et de la position. Notons que de nombreux environnements assez similaires ont été créés (Pea, 1987). Dans COINLAND et NUMBERLAND, (Hamburger et Lodgher, 1989) il s'agit d'amener l'élève le plus près possible de l'ultime transition papier/crayon. On a ici des micromondes successifs autorisant peu d'opérateurs, la légalité de leur application correspondant quasiment aux possibilités physiques, ce qui évite les erreurs et impasses.

En fait, une grande partie des mathématiques élémentaires a été traitée : les fractions et la proportionnalité (Hoyles et al., 1989), les manipulations algébriques (McArthur et al., 1988; Feurzeig, 1986), les problèmes algébriques posés en langue naturelle (voir Greeno, 1987), les isométries (Thompson, 1987), les nombres négatifs (Schwarz et al., 1992)....

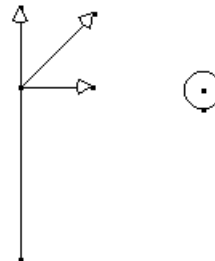
OPERA (Schwarz 1992) est un modèle causal pour l'acquisition des structures algébriques. Il est constitué de trois systèmes de représentation : de *type machine*, les opérateurs sont des machines opérant sur des trains (nombre de wagons et sens de parcours), de *type table*, montrant un opérateur comme une fonction associant à une série d'entrées une série de sorties, de *type pile*, montrant la structure des expressions algébriques. Ces représentations expriment différents aspects des opérateurs, les tables montrent l'idée d'application, les machines expriment les

opérateurs comme des processus et les piles donnent corps aux structures algébriques. Les activités sont définies par un ensemble d'outils et d'actions et une formulation contenant un but. Pour Schwarz, OPERA crée un langage approprié et un domaine de référence qui induit des interactions génératives dans des situations nouvelles.

Les micromondes ne sont pas limités aux mathématiques et de très nombreux domaines ont été abordés. On peut ainsi citer les environnements destinés à faciliter l'apprentissage de la lecture (Lawler, 1987 ; Cohen, 1987), l'expression écrite (Sharples, 1985), l'analyse et la génération de textes (Robert, 1986), bien que ces micromondes soient légèrement différents des micromondes transitionnels en mathématiques que nous venons de voir.

L'apprentissage de la physique est un domaine privilégié pour le travail avec les micromondes. Leur usage a ses racines dans les travaux sur les préconceptions et ce que l'on appelle la physique naïve (Viennot, 1979). A l'aide d'une petite expérience avec un simple jeu de cibles et le pilotage d'une tortue dynamique, diSessa (1982) montre que la plupart des personnes ont une vision de la dynamique correspondant aux idées d'Aristote, non à celles de Newton. Les notions d'accélération et de vitesse sont difficilement perçues. Par l'intermédiaire de LOGO, Alain Texier (1985) montre comment aborder de nombreux concepts physiques et modéliser des situations. Son objectif est de permettre à l'apprenant d'établir un lien entre sa perception intuitive d'un phénomène et la formulation rationnelle qu'il peut élaborer.

La tortue démarre du point situé en bas et à gauche. On peut lui donner des impulsions dans une direction que l'on choisit (aucune ne permettant d'atteindre directement la cible). Une première impulsion verticale permet de faire avancer la tortue verticalement à vitesse constante. Pour atteindre la cible, de nombreuses personnes pensent qu'il suffit de donner une nouvelle impulsion horizontale au moment où la tortue est à l'horizontale de la cible. Mais, la loi de composition des forces donne à la tortue un mouvement en diagonale.



L'expérience de di Sessa (1982)

Les recherches autour de WHY et SOPHIE (présentées au chapitre précédent) ont montré les limites des simulations quantitatives, dues pour l'essentiel à leur incapacité à fournir un compte rendu complet de la causalité qui sous-tend les inférences. Cela conduit au développement du raisonnement qualitatif sur le comportement des systèmes physiques (Bobrow, 1984) et à l'idée que la description du comportement d'un système doit dériver de la structure même de ce système, c'est-à-dire de l'analyse du comportement de ses différentes composantes et de leurs

interrelations. A la suite de ses travaux sur QUEST<sup>17</sup> (White et Fredericksen 1987), dédié à l'étude des circuits électriques, Barbara White (1993) prône une approche de l'enseignement scientifique fondée sur une progression à travers des modèles causaux de complexité croissante, initialement présentés aux apprenants à un niveau d'abstraction intermédiaire. D'après elle, les approches traditionnelles se centrent trop sur des représentations abstraites négligeant des formes de représentation visuelles et linguistiques plus accessibles. Il s'agit pour les étudiants de se créer des modèles conceptuels par interaction avec des micromondes donnant corps à ces modèles. A l'aide de tels modèles, les étudiants devraient pouvoir comprendre et expliquer le comportement de systèmes physiques et ainsi être amenés à se représenter mentalement les mécanismes de causalité implicites dans les abstractions telles que  $F=mg$ . En effet, les modèles doivent permettre de donner du sens aux abstractions de niveau supérieur et aux formalismes mathématiques associés ainsi qu'aux situations réelles. D'après White, il s'agit d'aider l'apprenant à construire un modèle causal par expérimentation avec une vue physique.

Dans le cadre de la formation professionnelle technique, Fischer développe des *micromondes de complexité croissante* (Fischer et al. 1978 ; Fischer, 1988). Les compétences cognitives sont enseignées grâce à un entraînement dans des environnements de plus en plus complexes simulés sur ordinateur. La pratique porte d'emblée sur tous les aspects d'un domaine de compétence, mais en la confrontant à des environnements de complexité croissante. Fischer s'appuie sur l'exemple du ski, pour lequel l'invention des skis courts et des fixations de sécurité a permis de mettre les débutants beaucoup plus rapidement sur la piste<sup>18</sup>. Notons que cet exemple est assez différent des précédents puisqu'en formation professionnelle, l'objectif est d'arriver à atteindre un certain niveau de performance.

#### 4.3.2.2. Rôle des représentations et des erreurs

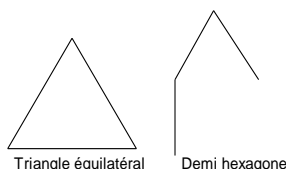
Une des caractéristiques centrales des micromondes est liée à la notion d'erreur. En effet, l'utilisateur est amené à confronter ses idées intuitives avec des représentations informatisées. Si ces idées sont fausses, les objets du micromonde ne vont pas avoir le comportement attendu. L'effet de rétroaction est théoriquement immédiat de par la familiarité de l'apprenant avec les objets du micromonde. L'exemple prototypique est celui du dessin du triangle équilatéral en LOGO. En effet, la tortue parcourt les côtés de la figure et sa rotation correspond à l'angle externe et non à l'angle interne (dont on a tendance à se rappeler la mesure de  $60^\circ$ ). En utilisant la valeur de l'angle interne, on tombe sur une figure tout à fait inattendue

---

17. Les premiers travaux, autour de STEAMER et QUEST, sont abondamment décrits dans Wenger (op.cit., p.79-100).

18. Coutaz (1990, p.79) parle de *roulettes de sécurité* correspondant aux roulettes que l'on fixe de part et d'autre de la roue arrière des vélos afin que les enfants évoluent sans risque de chute, à propos de l'organisation des fonctions d'un système en niveaux de complexité croissante.

(un demi hexagone) ce qui conduit l'apprenant à réexaminer sa démarche pour localiser la source de l'erreur et y remédier.



Dans le cas de la géométrie tortue, l'anthropomorphisme est essentiel. L'apprenant peut jouer le rôle la tortue et essayer d'effectuer lui-même la figure voulue. En prenant conscience de ses mouvements, il s'aperçoit alors de l'origine de son erreur. L'erreur est vue comme l'expression d'une conception erronée, comme un obstacle sérieux qu'il faut identifier puis surmonter, non comme une simple déviation. Cette vision de l'erreur comme obstacle épistémologique apparaît plus en accord avec les positions des didacticiens (Brousseau, Martinand) que les premiers travaux sur les modèles élèves décrits au chapitre précédent. Notons que la connaissance du comportement des objets du micromonde permet à l'apprenant de percevoir ses erreurs sans qu'il soit nécessaire de les lui indiquer. Dans le cas, où les objets sont étrangers, le repérage de l'erreur devient plus problématique. Ainsi, l'erreur linguistique est plus difficile à repérer sur une production (les élèves ayant tendance à être très permissifs et à accepter les productions linguistiques les plus barbares). Cependant, même si l'élève arrive à percevoir une erreur, il n'arrive pas toujours à en trouver la cause. Cette difficulté importante va peu à peu conduire les chercheurs à introduire des formes d'aide et de guidage.

Un deuxième point très important concerne la notion de représentation. Comme l'ont mis en évidence de nombreux travaux en intelligence artificielle, trouver une bonne représentation d'un problème est souvent un grand pas vers sa solution. En tant qu'outil d'expression, un micromonde fournit les outils pour se construire ses propres représentations pour traiter un problème. Di Sessa (1987) décrit ainsi l'expérience d'un étudiant travaillant la géométrie sur un cube et réfléchissant à partir d'un développement plan du cube. Bien que ce ne soit pas le formalisme le plus utilisé en mathématiques, l'étudiant a peu à peu appris à se servir de cette représentation jusqu'à ce qu'il maîtrise un formalisme suffisant pour traiter des problèmes de géométrie sur le cube. Il s'est construit sa propre façon d'y réfléchir. Un tel comportement rejoint celui d'un expert dans la résolution de problèmes. Il passe beaucoup de temps à se représenter ou à modéliser le problème qui lui est soumis et recherche la solution à partir de cette représentation. Di Sessa souligne « qu'une telle construction de petits formalismes sur une période de temps suffisamment longue peut être l'une des expériences les plus profondes des mathématiques ». Il s'agit d'avoir une véritable expérience par l'intermédiaire d'un monde artificiel. Le livre remarquable d'Abelson et di Sessa (*Turtle Geometry*, 1980), regorge d'exemples intéressants comme l'exploration de géométries non standards (sur des surfaces particulières non planes) ou des simulations de poursuites entre un prédateur et une proie, montrant bien que la notion de micromonde ne se limite pas à des environnements à visée éducative conçus pour de jeunes apprenants. Ainsi, un des objectifs centraux dans les micromondes est soit de

fournir des représentations suffisamment *naturelles* ou *évidentes* (dans le sens où elles sont connues ou facilement maîtrisées par l'apprenant) pour qu'elles soient opérationnelles, soit de permettre de se les construire et de les explorer pour en acquérir une maîtrise suffisante.

Comme nous l'avons vu, le choix des activités est important. Elles sont de nature exploratoire (souvent pour s'assurer de la compréhension minimale du comportement des objets du micromonde), puis amènent à résoudre des problèmes (pour faire évoluer les modèles mentaux), à la production ou la construction d'objets particuliers (comme des figures spécifiques en géométrie tortue) jusqu'à la modélisation. S'il est toujours possible d'inventer de nouvelles activités (grâce à la flexibilité du langage informatique sous-jacent), elles prennent sens dans un contexte particulier lié à la finalité visée. Si en EAO, le programme en machine et le curriculum peuvent être étroitement imbriqués, ce n'est pas le cas avec les micromondes. L'enseignant doit adapter le curriculum aux buts de l'enseignement et aux niveaux des élèves. Pour Papert (1987), créer un curriculum (dans le sens d'un ensemble cohérent de ressources d'aide à l'apprentissage tout au long d'une période scolaire) consiste à créer un réseau de micromondes, chacun d'entre eux se centrant sur différents domaines de connaissance. Un tel réseau n'est pas très aisé à concevoir et les micromondes vont en partie butter sur cet écueil, dû au lien difficile à établir entre les activités proposées et les connaissances requises dans les programmes scolaires prescrits.

#### **4.3.3. Evaluation de LOGO et des premiers micromondes**

Dès la fin des années soixante-dix, sous l'impulsion de Papert, LOGO rencontre un succès important et se généralise avec l'essor des micro-ordinateurs. Fondé sur une pédagogie constructiviste plus en accord avec les idées des enseignants les plus innovants que les théories béhavioristes sous-jacentes à l'enseignement programmé et à l'EAO, LOGO apparaît comme une voie prometteuse pour utiliser les ordinateurs dans l'éducation. Le problème est alors de fournir une validation expérimentale aux intuitions de ses promoteurs. Si les premières expérimentations menées par les chercheurs s'apparentent à des recherches-actions, l'extension des utilisateurs conduit à s'intéresser de manière plus précise aux bénéfices escomptés par la pratique du LOGO.

Une question centrale est celle du choix des méthodes d'évaluation. En effet, les chercheurs ne savent pas vraiment mesurer tout ce qui constitue la valeur éducative d'une innovation. Or, il ne s'agit plus ici d'une mesure de performances qui peut être accomplie grâce à des études différentielles (avec pré-test, post-test, groupe expérimental et groupe de contrôle), mais plutôt d'une recherche de transferts de capacités, incitant la poursuite d'études de nature génétique centrées autour de cas. Certaines expérimentations sur LOGO ont utilisé des mesures traditionnelles, alors

qu'il apparaît difficilement possible d'isoler une seule variable dans un tel environnement. Papert s'en est souvent défendu, en affirmant qu'il ne faut *pas demander ce que LOGO peut faire, mais ce que l'on peut faire avec LOGO* (Papert, 1985). D'une part, contrairement à l'EAO où l'enseignant est souvent exclu, ce dernier joue un rôle primordial avec LOGO et, d'autre part, Papert insiste sur l'aspect social et interactionnel du travail avec les ordinateurs, notamment les relations entre pairs. Le prototype des premières recherches autour de LOGO va consister à prendre un groupe d'étudiants, voire un seul (comme Lawler avec sa fille, Lawler 1981), travailler assez longtemps avec eux pour tester l'innovation et espérer des résultats qualitativement différents.

Les expérimentations autour de l'usage de LOGO n'ont jamais donné de très bons résultats. Ainsi, dans le domaine mathématique, les élèves ayant travaillé avec LOGO n'ont pas de meilleures performances que les autres. Un point positif est néanmoins souvent souligné par les enseignants (Howe et al. 1980 ; Ross et Howe 1981) : ces élèves arrivent à argumenter de manière sensée sur des points mathématiques et à expliquer clairement diverses difficultés. Un panorama sur dix ans d'utilisation (Ross et Howe 81, Howe et al. 84) permet d'identifier certains avantages de LOGO comme environnement favorisant : des projets d'élèves importants, la discussion et l'expérimentation réflexive, des interprétations et illuminations sur le sens (Hoyles, 1985). Au niveau des contenus (Feurzeig et al. 1981), LOGO favorise la rigueur de pensée, la compréhension de concepts généraux (variables, procédures, fonctions, récursivité) et la capacité à les utiliser, la maîtrise de la planification et de la décomposition en sous-problèmes et la recherche des erreurs. Ces premiers résultats mettent aussi en évidence le rôle central des enseignants (Vivet, 1980).

Par la suite, la majorité des études empiriques se sont attachées à déterminer dans quelle mesure programmer en LOGO augmente la capacité des enfants à résoudre des problèmes, c'est-à-dire la possibilité du transfert de capacités développées lors du travail sur l'ordinateur vers des domaines extérieurs à l'informatique. Ce problème du transfert est un peu la pierre d'achoppement de la pédagogie. Si on peut définir cette notion de transfert comme "simplement appliquer des informations d'une catégorie connue à une nouvelle instance" (McCoy Carver, 1986), les chercheurs ont une idée plausible mais vague des similarités entre les tâches source et but, et les moyens de faciliter cette application ne sont pas encore très clairs. Le bilan des recherches est décevant (Crahay, 1987). Pour Mendelsohn (1988), «...il ne semble pas raisonnable d'attendre des effets de transfert de l'apprentissage de la programmation sur des compétences et des aptitudes de haut niveau ». L'apprentissage d'un langage de programmation se traduit moins par un gain en connaissances générales de haut niveau (stratégies générales de résolution de problèmes, heuristiques ...) que par la possibilité d'utiliser des *micro-expertises* puissantes et organisées permettant de traiter, avec une plus grande efficacité, des informations pertinentes dans des situations de résolution de problèmes ayant une

certaine analogie avec la situation d'apprentissage. D'après Crahay (1987), un tel résultat n'a rien d'étonnant, puisque « les psychologues qui se sont penchés sur les mécanismes de résolution de problèmes ne cessent de répéter qu'il n'est pas possible d'identifier un ensemble d'opérations mentales ou de capacités spécifiques qui soient efficaces dans tous les cas.... l'hypothèse des structures d'ensemble ne correspond pas aux données empiriques recueillies ces dernières années ; de même on abandonne l'hypothèse de procédures de traitement de l'information opérant en toutes circonstances.»

Marcel Crahay insiste sur les déviations et récupérations du constructivisme sous-jacent à LOGO par les doctrines pédagogiques traditionnelles :

- *récupération rationaliste*, l'idée que « toute faculté, une fois développée à son potentiel maximal, se manifeste en toutes circonstances »,
- *récupération empiriste*, où on réduit l'évaluation à l'appréciation de la quantité de connaissances maîtrisées et à une mesure de transfert général,
- *récupération libertaire*, où, sous prétexte de non-directivité, le rôle de l'enseignant est minimisé voire carrément supprimé.

Des recherches vont ainsi essayer de relier les interventions des enseignants aux apprentissages souhaités et mettre en évidence la nécessité d'une pratique de LOGO médiatisée par l'enseignant, donc d'une forme d'apprentissage explicite. Concernant les transferts d'activité de débogage, McCoy Carver (1986) montre que les enfants peuvent apprendre des aptitudes intellectuelles de haut niveau si les habiletés qui les composent sont spécifiées et enseignées directement. Une fois apprises, ces aptitudes peuvent être transférées lorsqu'elles sont reconnues comme se rapportant à la nouvelle tâche. De Corte et ses collègues (de Corte et al., 1991) obtiennent des résultats similaires, montrant que si les enfants acquièrent une connaissance suffisante des primitives et concepts du LOGO ainsi qu'une maîtrise des compétences intellectuelles dans le contexte de la programmation, un transfert à d'autres activités se produit sans qu'il soit nécessaire de prévoir un entraînement explicite au transfert. Pea et Kurland (1984 ; Pea et al. 1985) précisent que le développement des aptitudes liées à la résolution de problèmes dépend du fait que l'on ait effectivement montré aux enfants comment le langage pouvait être utilisé. L'évaluation porte aussi sur le mode d'enseignement. Le travail de Swan (1989) indique qu'un enseignement explicite et la pratique médiatisée de la programmation LOGO, et seulement de telles formes d'instruction et de pratique, induisent le développement de quatre stratégies particulières de résolution de problèmes : formation de sous-but, chaînage avant, essais et erreurs systématiques, analogie. Ces résultats témoignent de la supériorité de l'instruction explicite sur la découverte guidée et des environnements informatiques sur les manipulations concrètes pour l'enseignement et l'apprentissage de telles stratégies.

Outre la nécessité d'assurer un apprentissage explicite, les expérimentations vont soulever certaines difficultés de conception des micromondes. Ainsi, l'approche de

Dienès, intégrée dans divers micromondes concernant les mathématiques élémentaires, ne fonctionne qu'avec des enseignants charismatiques. La reconnaissance des éléments structurels par induction ne marche pas, les invariants doivent être présentés de manière explicite. Afin de surmonter cette difficulté, les théories de l'apprentissage situé suggèrent un modèle de généralisation graduelle : des structures de connaissance largement applicables devraient être peu à peu induites à partir de l'exposition à de nombreuses instances du monde réel. Un tel processus est très difficile à réaliser, principalement du fait que les liens entre les différentes instances du monde réel doivent être rendus explicites. Rendre la plus stable possible la forme externe des outils informatiques peut favoriser les reconnaissances de similarité entre différents domaines et faciliter leur appropriation. Toutefois, la conception reste délicate. D'après Neshet, définir rigoureusement les opérations et les conditions de validité de la composante connaissance dans la composante exemple demeure l'objectif essentiel. Il faut spécifier correctement la tâche qui incombe à l'élève et les objets de connaissance à modéliser. Thompson (1987) constate que l'approche initialement prévue avec son micromonde MOTIONS fut partiellement inopérante, en partie parce que les élèves étaient trop absorbés par les détails de construction et perdaient de vue les notions mathématiques qu'ils étaient censés apprendre.

En dehors des mathématiques élémentaires ou des apprentissages premiers, la conception des micromondes soulève de nombreuses difficultés. L'exemple de la réécriture est éclairant sur ce point. LOGOGRAM (Robert 1985, 1986) est un environnement de réécriture plutôt destiné à l'apprentissage de la langue et la maîtrise de la grammaire. Il est inspiré des travaux de Rowe (1976) et est assez proche d'autres réalisations en LOGO (Sharples, 1985) et en PROLOG (Kahn, 1984). Les expérimentations révèlent des obstacles importants. Tout d'abord, du côté des élèves, la non-conformité du résultat obtenu avec celui désiré ou attendu n'est pas toujours décelée. D'une part, les critères d'acceptabilité sont plutôt lâches et, d'autre part, en raison des choix aléatoires, les transformations opérées sur les textes lors des réécritures ne sont pas toujours visibles. Ensuite, la vision de la grammaire incorporée dans le micromonde apparaît assez éloignée de la grammaire traditionnellement enseignée dans les classes primaires et les enseignants disposent, en général, de peu de connaissances linguistiques ce qui rend problématique l'intégration d'un tel environnement.

Dans une méta-analyse des recherches consacrées à LOGO, Valcke (1991) critique le manque de scientificité d'un grand nombre d'entre elles, ce qui rend difficile leur comparaison et met en doute les effets significatifs souvent constatés. Il note que l'exploration libre n'est plus le mode d'intervention privilégié avec LOGO et que les recherches récentes s'orientent vers des hypothèses en relation avec les mathématiques, la cognition et la métacognition. Dans le cadre d'une thèse, Lhermenier-Marinho (1993) fait un tour d'horizon des recherches concernant l'influence de LOGO sur les capacités cognitives et métacognitives des enfants



(ibid., p. 56, 57, 65). Ses analyses l'amènent à la conclusion que l'activité avec LOGO n'entraîne pas une amélioration des processus intellectuels envisagés globalement, mais plutôt de capacités spécifiques (mathématiques, orientation spatiale, application de connaissances géométriques) et/ou donne lieu à un meilleur investissement scolaire de la part des enfants qui y sont soumis (ibid., p. 345).

Dans une revue assez complète des recherches autour de LOGO, Clements et Meredith (1993) rappellent que ses concepteurs y voyaient un cadre conceptuel pour l'apprentissage des mathématiques. Or, de nombreuses recherches ont supposé qu'une exposition directe des concepts mathématiques dans le contexte de la programmation LOGO pouvait permettre d'accroître les performances, ce qui n'a jamais pu être véritablement confirmé. Concernant la géométrie, il semble que des expériences en LOGO peuvent aider les enfants à prendre conscience de leurs intuitions mathématiques et faciliter la transition du visuel vers une pensée plus descriptive et analytique. Des recherches ont cependant mis en évidence la persistance d'incompréhensions sur les mesures d'angle et des difficultés de coordination entre la rotation de la tortue et l'angle construit, surtout en l'absence d'un guidage approprié par les enseignants. La conclusion de Clements et Meredith sur la géométrie est que les effets les plus positifs impliquent des suites bien planifiées d'activités LOGO et la médiation par les enseignants du travail des élèves, avec d'autres activités, pour les aider à forger des liens entre LOGO et d'autres expériences. En algèbre, un besoin d'aide est patent pour les processus d'abstraction et de généralisation. Clements et Meredith (1993) notent l'aspect facilitateur de l'interaction sociale. Pour eux, néanmoins, le facteur critique est la vision claire des buts de LOGO que doit avoir l'enseignant. Si une intégration complète dans le cursus mathématique leur apparaît nécessaire, avec l'idée chère à Papert d'apprendre aux enfants à être des mathématiciens plutôt que leur enseigner les mathématiques, la recherche montre que les enseignants trouvent très difficiles de créer un environnement d'apprentissage qui encourage la créativité dans les structures scolaires actuelles.

En résumé, si l'évaluation de LOGO a fait couler beaucoup d'encre, cela témoigne indubitablement de l'intérêt qu'il a suscité. L'un des résultats essentiels est sans doute de montrer la nécessité d'y associer un mode de soutien, qu'il soit intégré dans l'environnement lui-même ou assuré par un enseignant, assistance déterminante pour garantir une certaine efficacité du dispositif. Comme le note Balacheff (1994), « un micromonde offre à l'élève un environnement riche, mais il ne peut, par lui seul, garantir un apprentissage donné. Pour réaliser un tel objectif, le micromonde doit être inclus dans un dispositif didactique dans lequel il donnera les moyens de constitution d'un milieu ad hoc. Ce sont les caractéristiques de la réalisation de ce milieu qui garantiront (relativement) l'apprentissage escompté ». Mais, au-delà du LOGO lui-même, les idées sous-jacentes à sa conception ont influencé les travaux sur les environnements informatiques destinés à l'apprentissage, notamment en

montrant l'intérêt de manipuler des objets ayant un comportement visualisable sur un écran et de fournir des outils de modélisation aux apprenants.

#### **4.4. Extension de la notion de micromonde**

Avec le développement rapide de l'informatique, les idées développées initialement dans une finalité éducative sont appliquées dans d'autres contextes. Comme le rappellent O'Shea et Self (op.cit., p. 203), le cas de SMALLTALK est sur ce point exemplaire. Au début des années soixante-dix, Kay et Goldberg (1977) se centrent sur les enfants considérés comme étant leur communauté d'utilisateurs. Par la suite, SMALLTALK devient un média de communication pour les enfants de tous âges. Enfin, au début des années quatre-vingt, les concepteurs estiment qu'il devient nécessaire de ne pas abuser les lecteurs avec l'idée que SMALLTALK-80 est un langage pour les enfants alors qu'il est plutôt adapté au développement des logiciels pour les professionnels. L'intérêt éducatif de SMALLTALK se limite alors à la production de 'kits', c'est-à-dire des ensembles de composantes spécialisées ainsi que des outils pour les manipuler, permettant de développer des activités éducatives ouvertes.

Dans ce contexte, la notion de micromonde est en quelque sorte revisitée. La possibilité de manipulation de représentations graphiques devient un point essentiel, intégré à de nombreux systèmes conçus pour l'éducation. Les interfaces à manipulation directe se développent et le fait de fournir un langage de commande à l'apprenant apparaît parfois discutable. Dans cette section, nous allons passer en revue un certain nombre de réalisations qui se rattachent à la notion de micromonde.

##### **4.4.1. Les multimondes et le pilotage de représentations multiples**

La notion de multimonde se caractérise par la possibilité de travailler en parallèle ou de manière conjointe avec différentes représentations d'un même objet ou d'un même phénomène. Ainsi, en mathématiques, il s'agit d'exploiter le fait que certains concepts interviennent dans divers domaines ou divers cadres physique, géométrique, numérique, graphique... Des modes d'action spécifiques sont attachés à chacun des mondes permettant d'en modifier un et de visualiser l'effet sur l'autre (ou les autres). L'observation et la manipulation favorisent ainsi la compréhension du fonctionnement des concepts dans divers domaines liés entre eux, idée qui rejoint le travail de Régine Douady (1986) autour des jeux de cadre.

L'exemple le plus classique en mathématiques est celui du lien entre un monde géométrique et un monde algébrique (représentation graphique des fonctions, systèmes d'équations linéaires...). Le travail de Dubourg (1995) est consacré au développement d'un environnement de type micromonde sur les liens entre les

équations de droite et leur représentation cartésienne. Certains concepts ont des correspondances (des interprétations) aisément identifiables. Ainsi, dans l'expression  $y=ax+b$ ,  $b$  est l'ordonnée à l'origine et  $a$  la pente de la droite. Quand on considère un système linéaire à deux inconnues, l'interprétation géométrique comme intersection de droites est clair. Les correspondances sont moins évidentes avec la manipulation des équations pour l'élimination d'une inconnue (il s'agit en fait d'un changement de repère, s'arranger pour prendre l'une des droites comme nouvel axe ce qui élimine l'une des inconnues).

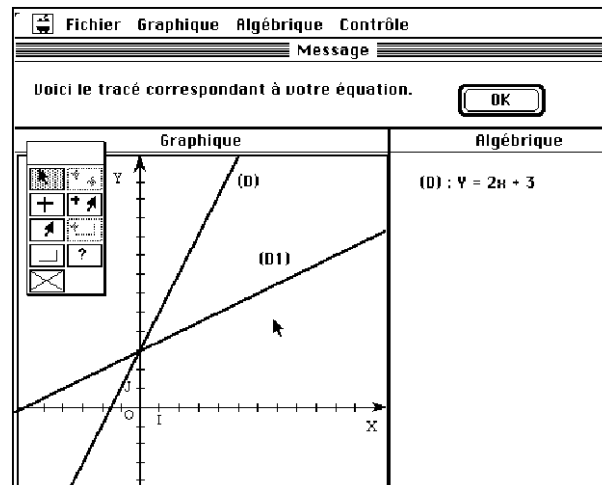
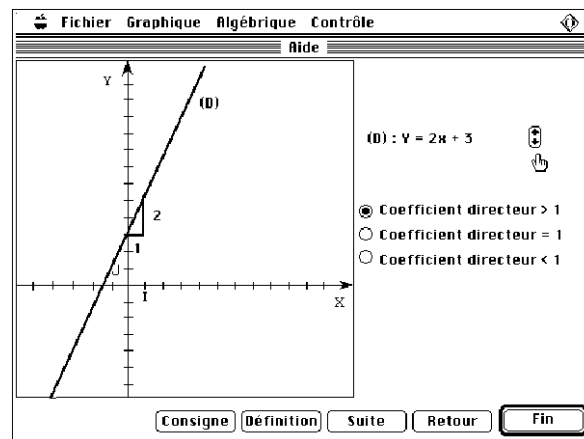


Figure 4.4. Copies d'écran du logiciel REPERES (Dubourg, 1995)

Concernant le lien entre la représentation symbolique des fonctions et leur représentation graphique, Yerushalmy (1991) a étudié la perception qu'ont les

étudiants du concept de fonction. Elle conclut, entre autres, que la relation entre les manipulations algébriques et la représentation visuelle ne s'opère pas spontanément. Si l'usage de représentations multiples peut aider les élèves à comprendre des concepts importants, cela ne résout pas les problèmes de manipulation algébrique. Elle suggère de nouvelles approches de l'algèbre basées sur des travaux autour de représentations liées entre elles (Yerushalmy, 1993). Comme le note Artigue (1995), la visualisation simultanée de plusieurs registres de représentation ainsi que la visualisation dynamique de leurs correspondances donne sans doute des résultats encourageants mais encore loin de répondre complètement aux attentes.

On peut noter d'ailleurs que d'autres représentations, moins habituelles, offrent d'autres perspectives sur les fonctions (tables, graphes, symbolisme algébrique). Arcavi & Nachmias (1993) utilisent une représentation nommée PAR (*Parallel Axes Representation*) avec deux axes parallèles.

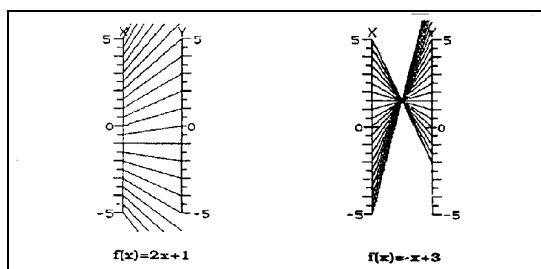
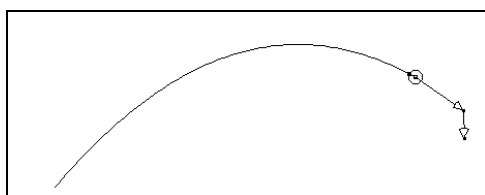


Figure 4.5. Exemple de représentation

L'axe de gauche est utilisé pour représenter le domaine de définition de la fonction et l'axe de droite son image. L'association entre un nombre et son image est représentée par un segment les joignant. La figure ci-contre montre deux exemples de la représentation nommée PAR. On remarque que certaines caractéristiques des fonctions affines sont aisément interprétables : le point fixe, l'aspect croissant ou décroissant, l'aspect dilatant ou contractant, etc.



En physique, *the Envisioning Machine*, programme conçu par Jeremy Roschelle (Greeno, 1990) est consacré aux concepts de vitesse et d'accélération. Il utilise deux fenêtres, appelée le Monde Observable et le Monde Newtonien dans lesquels un objet (une balle) se déplace suivant les principes de Newton. Dans le monde Newtonien, on associe deux flèches : la première a pour origine le centre du cercle (elle représente la vitesse), la seconde a pour origine l'extrémité de la première (elle représente l'accélération).

L'apprenant peut modifier directement ces deux flèches et observer les trajectoires correspondantes. Ainsi les concepts de vitesse et d'accélération sont représentés par des flèches (des vecteurs) ayant des effets sur les mouvements et les étudiants manipulent des symboles qui se comportent physiquement. Une activité intéressante consiste à reconstituer dans le monde Newtonien une trajectoire dans le monde Observable.

ARK ou *Alternate Reality Kit* (Smith, 1986), conçu en Smalltalk-80, consiste en une collection d'objets *physiques* qui peuvent être manipulés via une main simulée. Grâce à cette main, pilotée grâce à une souris, l'utilisateur peut changer la position ou la vitesse d'un objet (en le déplaçant ou en le lançant), envoyer des messages aux objets (par pression sur des boutons) et introduire de nouveaux objets et de nouveaux boutons dans l'environnement (par sélection dans des menus). Avec les fonctionnalités précédentes, il est possible de créer des simulations interactives ou simplement de travailler dans un laboratoire simulé. L'idée est de faciliter une compréhension intuitive des règles qui pilotent la simulation en rendant ces règles aussi accessibles que des objets physiques. L'apprenant inspecte la structure du modèle lui-même plutôt que simplement observer ses performances. On peut ainsi explorer différents mondes, en choisissant par exemple des formules d'attraction particulières entre deux masses (sans forcément la choisir inversement proportionnelle au carré de leur distance). ARK a eu une influence importante dans le développement des interfaces à manipulation directe (Hennessy et al., 1989).

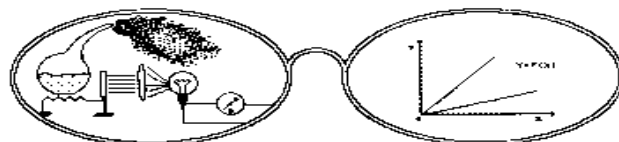
En physique, de nombreuses réalisations s'appuient sur des laboratoires simulés pour offrir des environnements d'exploration et de résolution de problèmes intéressants. De tels micromondes permettent non seulement de simuler les phénomènes d'un domaine mais ils peuvent en plus générer des explications sur le comportement étudié. Ils suscitent des activités d'exploration et de résolution de problèmes. De nombreuses réalisations pourraient être citées : laboratoire de physique (Kamsteeg et Bierman 1988), optique géométrique (Reimann, 1988), chocs élastiques, le système DiBi (Stumpf et al. 1988), le laboratoire électronique ELAB (Bocker et al. 1989), etc.

Il est non seulement possible de travailler sur des objets simulés rappelant les objets réels, mais aussi de piloter des objets physiques par l'intermédiaire d'un ordinateur.

#### **4.4.2. De la tortue simulée sur écrans aux micro-robots physiques**

Nonnon (1986) développe le concept de lunette cognitive. Il déplore la distance temporelle entre une action expérimentale et sa représentation graphique dans l'enseignement physique traditionnel. Pour réduire cette distance, Nonnon (ibid., p. 38) propose de présenter en simultanéité l'action expérimentale et sa

représentation graphique, ceci afin de permettre à l'élève une assimilation directe de cette représentation.



*Figure 4.6. La lunette cognitive (Nonnon, 1986)*

C'est l'idée de "lunette cognitive", intégrée dans une stratégie globale d'enseignement correspondant à un modèle d'acquisition par induction expérimentale. L'élève, via le clavier de l'ordinateur, peut provoquer des interactions en jouant ou en manipulant les variables et en voir directement les effets. Pour Nonnon (ibid., p. 11), les élèves doivent programmer leur environnement concret en utilisant le langage des concepts véhiculés par les sciences expérimentales. Il souhaite que l'étudiant devienne un programmeur-expérimentateur : qu'il puisse sélectionner une variable, l'isoler en neutralisant les autres, prédire une interaction entre cette variable et une variable ou vecteur de variables dépendantes, faire exécuter l'expérience, vérifier sa prédiction et enfin établir une loi.

Les idées de Nonnon s'inscrivent dans le courant de la robotique pédagogique, souvent connu par les activités de pilotage par programme LOGO d'objets construits en LEGO ou en Fischer Technik. D'après Lawler (1987), l'association avec LEGO est en germe dans l'idée de permettre à des enfants de construire des tortues de sol. Il devient possible de construire n'importe quel dispositif et de le contrôler par programme. LOGO peut ainsi être utile pour enseigner les concepts du contrôle de processus (Ross, 1986 ; Resnick et Ocko, 1991). Construire et piloter des micro-robots débouchent sur des environnements intéressants aussi bien la formation d'adultes que celle des enfants, favorisant des formes d'apprentissage de type coopératif (Leroux, 1995).

En effet, si LOGO est initialement destiné à la formation des enfants, il peut être un outil intéressant pour faire accéder des adultes à une culture minimale en informatique, notamment pour renouveler certaines idées en matière de dessin industriel ou de conception de machines-outils (Vivet, 1982 ; 1984). Le laboratoire informatique de l'université du Maine va jouer un rôle majeur dans le développement de telles idées. Plusieurs expérimentations vont montrer que faire construire des micro-robots s'avère une activité intéressante et profitable (Leroux, 1995).

Les travaux plus récents de Papert sont consacrés à développer une approche qu'il qualifie de cybernétique, contribuant à une réévaluation du concret (Papert

1993, p. 189). Passer de l'intelligence artificielle à la cybernétique c'est, pour lui, passer de prototypes plutôt basés sur la logique à des prototypes basés sur la biologie (ibid., p. 182). Un travail sur le vivant peut engager plus fortement les enfants, les relations affectives étant plus grandes. L'aspect concret invite à penser en termes de relations plutôt qu'en termes de propriétés. On s'imagine à l'intérieur du système. On peut ainsi approcher des concepts difficiles comme l'auto-organisation ou les structures émergentes si on a soi-même établi la simulation (qui peut être très simple). On peut citer ici les travaux autour des multi-tortues (Resnick, 1991a) et la simulation de colonies de fourmis (Resnick, 1991b). Papert cherche à conjuguer des façons non formalisées de connaître les mathématiques (approche de type ethno-mathématique) et pense que la construction d'un modèle peut permettre une compréhension qualitative d'un système complexe en en construisant un plus simple qui partage les mêmes principes (ibid., p. 21). Il renoue avec les principes de la cybernétique.

D'une certaine manière, il s'agit de construire des modèles et éventuellement de les modifier et d'en tirer parti pour comprendre des fonctionnements complexes. Cette approche est celle des simulations dites actives.

#### **4.4.3. Les simulations actives**

L'idée de simulation active est une synthèse entre les activités dans les micromondes et celles couramment développées dans les simulations. Des exemples concernant l'apprentissage de l'informatique ont été développés au Centre Mondial de l'Informatique. Trois environnements ont ainsi été créés : le *microprocesseur* (Jean-Luc Méheust et Gérard Weidenfeld), l'*interpréteur LOGO* (Nicole Rocland et Eric Bruillard) et la *simulation d'arbre* (Isabelle Péreira). Ces environnements exploratoires ont été conçus sur des modèles voisins et sont destinés à l'apprentissage des connaissances informatiques de base. Ils cherchent à tirer parti de métaphores familières et à proposer un modèle simple de principes généraux de fonctionnement dans un environnement de résolution de problèmes. Ce modèle peut être progressivement complexifié en vue d'intégrer les contraintes techniques et les approfondissements que celles-ci déterminent.

Dans le cas de l'interpréteur LOGO (Bruillard et Rocland, 1986 ; Bruillard et Péreira, 1987), l'objet est dans un premier temps de comprendre la manière dont les programmes LOGO sont interprétés par la machine et, dans un deuxième temps, d'explorer les possibilités de ce type d'interprète.

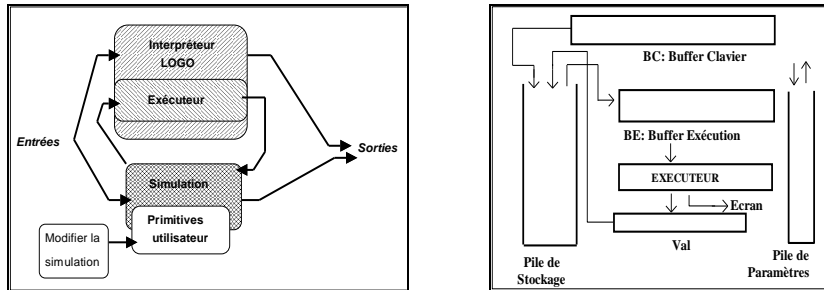
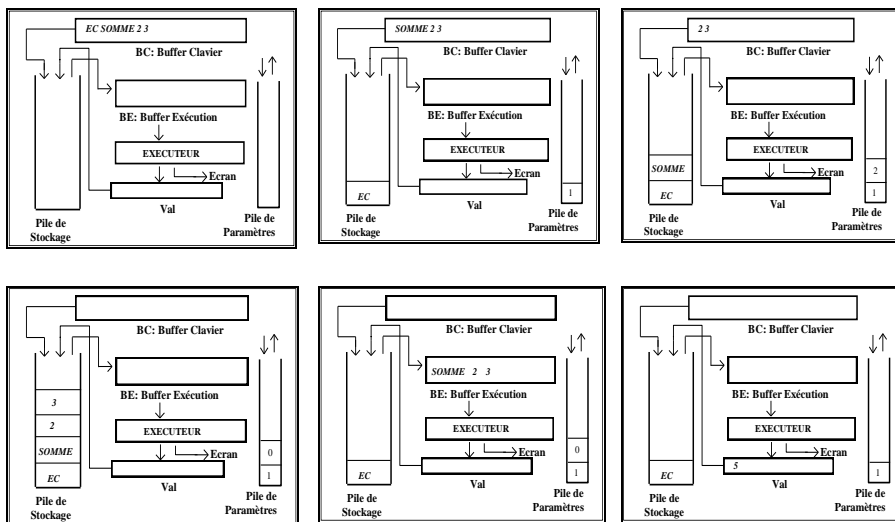


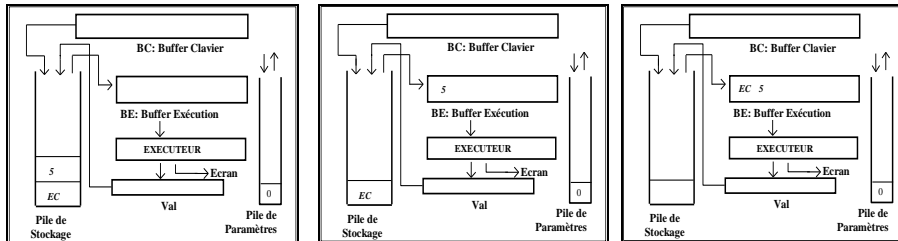
Figure 4.6. L'interpréteur LOGO

La simulation est construite au-dessus de LOGO et intègre un interpréteur LOGO réel pour son propre fonctionnement. Il est possible de comparer la simulation et le LOGO sous-jacent en exécutant des programmes soit directement soit par l'intermédiaire de la simulation. La fidélité de la simulation est vérifiée par ce type de comparaisons. On peut s'en écarter en acceptant dans la simulation des programmes autrement refusés et explorer des modes de fonctionnement particuliers.

La figure 4.7 donne les principales étapes de l'interprétation de la phrase **EC SOMME 2 3**







**Figure 4.7.** *Interprétation de la phrase EC SOMME 2 3 avec le simulateur LOGO*

Le buffer (ou tampon) clavier contient la phrase à interpréter et le buffer exécution contient les expressions à envoyer à l'exécution. La pile de stockage est un intermédiaire entre les buffer clavier et exécution. La pile de paramètres permet de gérer l'empilement et le dépilement dans la pile de stockage. VAL récupère les objets éventuellement retournés par l'exécuteur.

Dans ce fonctionnement, on construit en quelque sorte un résolveur de type *boîte de verre* (c'est-à-dire capable d'expliquer ou de donner à voir son fonctionnement) comme sur-couche d'un résolveur de type *boîte noire*. La simulation offre une matérialisation, un cadre d'interprétation visuel et concret d'un processus abstrait. Elle permet le lien avec un monde formel réel mais non accessible. On cherche à diminuer la complexité en se centrant sur des aspects particuliers sans tenir compte de phénomènes pouvant être imbriqués. Les détails réels du fonctionnement de l'interpréteur sont inutiles, une représentation, non pas fidèle sur le plan du fonctionnement, mais conforme au plan des entrées-sorties est recherchée.

Divers modes d'intervention sont prévus, correspondant à des formes de programmation à différents niveaux :

- Découverte des constituants de l'environnement, c'est-à-dire de la situation telle qu'elle se présente et des primitives permettant de la modifier. Dans le cas de l'interpréteur, diverses boîtes représentent des piles et des buffers, un réseau de communication entre ces boîtes symbolise les transferts possibles entre ces éléments, les primitives permettent d'effectuer ces transferts.

- Résolution de problèmes en mode pas-à-pas par activation d'une séquence de commandes. Par exemple, une phrase LOGO est proposée par le système ou donnée par l'utilisateur, et ce dernier doit faire fonctionner le dispositif pour l'*exécuter*. Il peut contrôler visuellement l'effet des primitives invoquées. Un mode CACHE est également proposé dans lequel l'utilisateur ne connaît pas les éléments de la phrase LOGO qu'il manipule (ils sont symbolisés par des carrés) et doit effectuer le travail en se basant uniquement sur les effets des primitives sur la simulation.

– Ecriture d'un algorithme général de résolution. C'est une étape de généralisation et de programmation, l'utilisateur doit réaliser une procédure permettant l'exécution de n'importe quelle phrase LOGO.

A tout moment, des aides variées sont accessibles : un mode *démonstration*, un mode permettant le retour en arrière afin de restaurer une situation antérieure, un mode *cinéma* permettant de revoir le déroulement des opérations à partir d'un instant donné. L'environnement autorise aussi un travail spécifique autour des erreurs. Il est possible d'introduire des phrases syntaxiquement fausses et d'en observer l'effet sur la simulation, afin de voir à quel moment du cycle d'interprétation l'erreur est détectée et le contexte auquel est associé le message d'erreur fourni. Cette activité effectuée en mode *CACHE prouve* ainsi que la machine n'effectue pas d'analyse sémantique de la phrase proposée ni recherche des intentions de l'utilisateur (croyance naïve de beaucoup de débutants), mais qu'il s'agit uniquement d'un dysfonctionnement à un moment donné dans un processus totalement automatisé.

Dans la simulation d'interpréteur LOGO, il ne s'agit pas uniquement de réécrire LOGO en LOGO, comme c'est classique dans ce type de langage (e.g. LISP en LISP, Abelson et Sussman, 1985, ou PROLOG en PROLOG, Shapiro et Sterling 1986), mais de concrétiser un processus abstrait qui est celui du fonctionnement du véritable interprète du langage LOGO. Cette réification s'appuie sur des objets de transition simples (des boîtes et des fils). Le modèle sous-tendant l'interpréteur peut lui-même être changé pour intégrer certains fonctionnements particuliers de LOGO (comme le traitement des opérateurs infixes ou la primitive SI) ou explorer diverses tolérances (suppression des caractères spéciaux). On a une double forme de complexification : niveau d'intervention sur un modèle déterminé (exploration, pas-à-pas, algorithme), changement du modèle pour intégrer d'autres contraintes techniques (c'est notamment le cas de la détection de la récursivité terminale avec des opérateurs commutatifs, qui n'est pas intégré dans les interprètes existants). Ce type d'environnement exploratoire se rapproche des micromondes à complexité croissante (Fisher, 1988). L'une des limitations de ces environnements provient de leur relatif manque de généralité.

#### **4.4.4. Les micromondes en géométrie**

Les micromondes de construction géométrique occupent une place particulière puisqu'ils prolongent des instruments de dessin déjà classiquement utilisés. Ils ont suivi, voire, par certains aspects, précédé l'évolution de l'informatique, notamment dans le domaine des interfaces graphiques. Ils permettent de passer du monde des tracés au monde des objets géométriques. Montrons quelques étapes de leur évolution.

##### **4.4.4.1. SKETCHPAD**

SKETCHPAD, conçu par Ivan Sutherland au début des années soixante (Sutherland, 1963), est un programme qui permet de *converser* avec un ordinateur par l'intermédiaire de tracés de lignes. La communication n'est toutefois pas celle du papier crayon, c'est-à-dire la trace de carbone laissée sur une feuille de papier. Au lieu de simplement simuler le papier, le programme de Sutherland offre la possibilité de spécifier des conditions mathématiques sur des parties déjà tracées qui seront automatiquement satisfaites pour donner au dessin la forme exacte désirée. En effet, le programme fonctionne par contraintes, c'est-à-dire par la déclaration de restrictions placées sur des parties d'un objet complexe ; il stocke les informations donnant au dessin son apparence particulière mais aussi une description structurée des sous-figures intervenant dans le dessin. Les contraintes sont respectées lors d'un déplacement. Autant les figures que les contraintes (comme le parallélisme, l'orthogonalité ou l'égalité de longueur) peuvent être visualisées à l'écran, aussi l'utilisateur peut non seulement savoir qu'elles existent mais aussi les modifier ou les effacer. Afficher une structure de contraintes est essentiel car cela montre comment les éléments dépendent les uns des autres. L'utilisateur dispose d'opérations atomiques (création de points, lignes, cercles et autres contraintes) et d'une opération de copie. Un processus dit de fusion récursive permet de fusionner des objets de même type et de propager la contrainte résultante à tous les autres objets qui en dépendent. D'après Sutherland, ce type de dessin basé sur des contraintes explicites apporte une compréhension importante dans de nombreux cas, notamment pour la construction de lieux géométriques.

Pour de nombreux chercheurs, il semble que SKETCHPAD fut une révélation<sup>19</sup> (Rheingold 1993, p. 87). Ce fut en effet le premier programme graphique de communication homme machine : à l'aide d'un photostyle et d'un écran à tube cathodique on pouvait piloter un ordinateur en dessinant à l'écran. Cette première interface interactive homme/machine fut également celle qui pour la première fois offrait une manipulation directe. L'utilisateur dialoguait avec SKETCHPAD en pointant sur une image à l'écran. Le système répondait en mettant immédiatement à jour le dessin, ce qui rendait la relation entre l'action de l'utilisateur et le graphique affiché à l'écran explicite (Brennan, 1990).

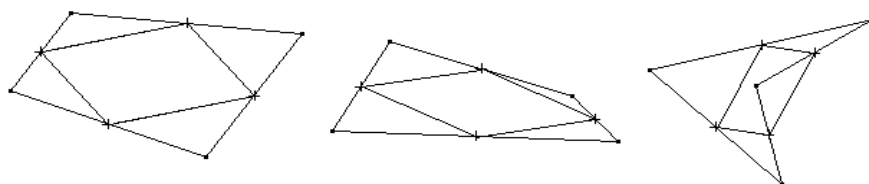
#### 4.4.4.2. THINGLAB

Le système THINGLAB, conçu par Borning (1977) dans le cadre du projet *DynaBook*, peut être considéré comme une extension de SKETCHPAD incluant certains apports de SMALLTALK. Il est aussi basé sur les notions de contraintes et de fusion récursive et est étendu pour inclure des situations plus complexes que les objets géométriques. Il permet de construire des simulations à partir de contraintes,

---

19. Alan Kay fut fortement influencé par ce programme et Nelson qualifia plus tard (dans *The Home Computer Revolution*, 1977) Sketchpad de « programme informatique le plus important qu'on n'ait jamais écrit, ... montrant à quel point le travail humain pourrait être facilité si les ordinateurs étaient réellement conçus pour aider les hommes ».

une contrainte restreignant le comportement d'un ensemble d'objets, c'est-à-dire leurs réponses à des messages. Dans une description du fonctionnement de THINGLAB, Goldberg (1979) fournit un exemple de construction de dessins géométriques, illustrant le théorème suivant : *les quatre segments reliant les milieux des côtés adjacents d'un quadrilatère forment un parallélogramme*. Différents cas de figures sont obtenus en déplaçant sur l'écran un sommet du quadrilatère. Elle précise que le système continue à satisfaire les contraintes spécifiées lors de la construction du dessin fournissant une rétroaction visuelle sur les relations ainsi maintenues. Un tel contrôle peut aider à convaincre les apprenants qu'un théorème est vrai ou les aider à trouver un contre-exemple.



**Figure 4.8.** Exemples de constructions avec Thinglab

Des idées similaires seront reprises dans le cadre du projet CABRI, qui va ajouter la manipulation directe permettant une déformation continue, en temps réel, des constructions réalisées.

#### 4.4.4.3. Les extensions de la géométrie tortue

L'extension de la géométrie tortue du LOGO vers la géométrie euclidienne a déclenché sans conteste un intérêt international. En effet, la géométrie de la tortue est une géométrie locale qui ne permet pas la manipulation des objets traditionnels de la géométrie affine ou euclidienne enseignée dans les collèges et les lycées. Les premières réalisations ont avant tout consisté en la création d'un monde d'objets géométriques (concernant la géométrie plane ou la géométrie dans l'espace) bâti au-dessus de LOGO. Les concepteurs ont d'ailleurs souvent fait preuve d'une imagination limitée dans le choix du nom de leur système, Euclide et ses éléments semblant être une référence obligée.

Citons rapidement et de manière non exhaustive :

- EUCLIDE (Allard et Pascal, 1987) qui a été bien diffusé en France,
- EUCLID cité par Thompson (1987), dû à Greenleaf en 1984, qui donne la possibilité de composer des fonctions et inclut un langage de programmation,
- EUCLIDE *espagnol* (Fabrega, 1988 ; Montès, 1988),
- PLS (Plane Geometry System) créé en Bulgarie (Sendov et Dicheva, 1988),

- LEGO dû à Fuller, Prusinkiewicz et Rambally (1985), programme plutôt dédié aux enseignants pour qu'ils puissent bâtir leurs démonstrations,
- Système auteur pour la géométrie (Floris et Bevacqua, 1989),
- GeoBlock, créé au Japon (Hidaka, 1992).

En fait, si les réalisations conservent des traces du LOGO sous-jacent, leur objectif n'est plus de même nature. Il s'agit de favoriser la construction et éventuellement la manipulation de figures, en substituant aux instruments traditionnels de la géométrie des commandes d'un langage de programmation. Une expérimentation avec EUCLIDE (Artigue, 1994) montre d'ailleurs un parasitage récurrent du travail mathématique par des difficultés de nature informatique ou, tout au moins, des difficultés inhérentes à une réalisation informatique particulière, imposant des contraintes syntaxiques pas toujours bien compréhensibles pour les élèves et difficilement maîtrisées. En outre, la nécessité de déclarer explicitement tous les objets, même si on les perçoit à l'écran, perturbe souvent les élèves. La création de deux points A et B ne donne pas d'existence au segment AB ou à la droite AB, de même, les intersections d'objets doivent être nommés. Enfin, la réexécution des tracés est possible mais lente. Finalement, ces diverses réalisations contraignent l'apprenant à maîtriser, à un certain niveau, un langage de programmation, sans tirer de bénéfices tangibles de cette obligation. On déclare différents objets (soit D la droite passant par un point et orthogonale à la droite D', soit C le cercle de centre O et de rayon 5, etc.) mais l'objectif est de réaliser un certain tracé, pas d'effectuer des traitements sur la suite des propositions déclarées. En conséquence, les améliorations vont consister à faciliter la création et la manipulation des figures géométriques, sans obliger l'apprenant à recourir à une syntaxe complexe. Fondé sur la manipulation directe, le logiciel Cabri-géomètre, réalisé à Grenoble, semble être la forme actuelle la plus achevée des micromondes de construction géométrique.

#### 4.4.4.4. CABRI et la manipulation directe

CABRI, Cahier de Brouillon Informatique, a d'abord été conçu comme un outil d'aide à la recherche en théorie des graphes. Le projet a été constitué en 1982 (Laborde, 1995). « Dans le domaine de la théorie des graphes ou des ensembles ordonnés comme dans celles qui en font l'usage, il arrive très fréquemment que l'on ait recours à la réalisation de *petits dessins*, formés essentiellement de sommets et d'arêtes; cela dans un but de les manipuler, de leur appliquer concrètement certaines transformations, ne serait-ce que pour vérifier telle ou telle propriété, conforter ou infirmer une idée, une conjecture...L'idée est donc naturelle de vouloir élargir les possibilités d'exploration par la mise à disposition des chercheurs du domaine (et d'étudiants) d'un outil graphique où le traditionnel cahier de brouillon serait remplacé par l'écran interactif d'un terminal associé à toute la puissance de calcul et de mémorisation de l'ordinateur ». Plusieurs maquettes et versions de *Cabri-graphes* ont été réalisées puis l'idée est venue de transposer cette réalisation dans le domaine

de la géométrie, celles des figures géométriques de l'enseignement secondaire (Laborde, 1985).

Cabri-géomètre offre une alternative aux méthodes classiques (papier, crayon, règle, compas, ...) à la fois pour la construction de figures géométriques, pour l'exploitation des propriétés de ces figures et pour en dégager de nouvelles connaissances (Baulac et Laborde, 1989). La caractéristique essentielle est le redessin en temps réel de la figure au fur et à mesure que l'on agit sur ses éléments, avec une continuité apparente dans la transformation. Pour ce faire, CABRI construit un graphe de dépendance, graphe orienté sans circuit. La déformation de la figure par déplacement consiste à modifier les informations d'une source du graphe et à propager les effets de cette modification. La redéfinition d'un objet a posteriori consiste à modifier les arcs du graphe.

Au niveau de l'enseignement de la géométrie, CABRI met en œuvre une géométrie que l'on peut qualifier de dynamique qui permet l'explicitation de propriétés en action, la distinction entre ce qui relève de propriétés perceptives et de propriétés géométriques, une révision du statut de la démonstration en géométrie pour les élèves, par la formulation de conjectures invitant à dépasser le cadre d'une simple vérification pragmatique. Une propriété apparaît clairement comme un invariant dans un ensemble de transformations, quelque chose qui reste vérifié pour toutes les configurations que l'on peut obtenir par déformation d'une figure construite<sup>20</sup>. L'environnement est extensible et modulable. En contraignant les outils disponibles, l'exploration d'autres géométries est possible. Par ailleurs, la géométrie des figures est un point de départ de la construction d'autres micromondes (Laborde et Laborde, 1991), notamment dans les domaines de l'optique géométrique et de la mécanique. Notons que CABRI II (Laborde, 1995) offre des fonctionnalités accrues, ouvrant de nouvelles perspectives.

#### *4.4.4.5. Langage de programmation et manipulation directe*

L'évolution des micromondes de géométrie illustre certaines étapes du lien entre le développement de l'informatique et les idées pédagogiques et didactiques fondant les réalisations éducatives. Les environnements de géométrie euclidienne, bâtis comme surcouche de LOGO, correspondent à une extension mais aussi à une forme de scolarisation. Il s'agit pour l'essentiel de pouvoir manipuler, par programme, les entités de la géométrie élémentaire, rendre les outils plus proches des objets à traiter dans l'enseignement, substituer aux tâches manuelles classiques de construction de figures des tâches de pilotage via des formes langagières. Les environnements de manipulation induisent un pilotage à l'écran avec des modes de désignation

---

20. Les démonstrations utilisant le mouvement s'introduisent ici dans un cadre naturel. La notion de geste pourrait même avoir un statut dans la géométrie.

traduisant une volonté de simplifier la tâche de l'utilisateur en lui fournissant le moyen de manipuler directement les objets qu'il connaît. Dans la phase de construction, le processus même de construction n'apparaissant pas fondamental (sauf peut-être pour vérifier ce que l'élève a pu comprendre de l'énoncé), le critère de choix des outils se ramène à un rapport coût / efficacité. Le bénéfice supplémentaire des environnements est dans l'exploration rendue possible, sans travail additionnel, d'un ensemble de configurations, de tracés, facilitant la validation de la construction effectuée et la formulation de conjectures, fondées sur des informations perceptives.

L'abandon progressif d'un langage de commande au profit d'une interface de manipulation directe amène plusieurs commentaires. Du point de vue des performances, tant au niveau de la construction que de l'exploration, la balance penche nettement du côté de la manipulation directe. Toutefois, l'opposition entre interfaces de manipulation directe et langages de commande ne peut se limiter à cela et les deux approches sont complémentaires : la première est plus efficace au niveau de la construction des objets et de leur manipulation physique, mais peut manquer de représentation alternative sur laquelle travailler (difficulté de restitution d'états antérieurs ou de création de nouveaux objets). La manipulation directe et la trace du raisonnement (des actions effectuées) sont certainement à articuler pour favoriser la verbalisation des connaissances. Des comparaisons ont été effectuées entre *Geometric Supposer*, basé sur des menus et *PLS* basé sur un langage. Les menus facilitent l'apprentissage initial, le langage est plus flexible et plus puissant, en fait, offre plus de facilités d'introduction de nouvelles fonctions. L'existence d'un langage revient à immerger le micromonde dans un environnement de programmation, mais cela conduit à un ensemble souvent un peu fourre-tout dont les fonctionnalités sont rarement exploitées.

En fait, ce qui oppose manipulation directe et langage de commande est d'un côté la facilité de construction et d'exploration et de l'autre, l'objectivation et le contrôle sur le processus de construction. Si on manipule facilement ce que l'on voit, le lien entre la figure et sa représentation langagière (sous forme de conjonctions de propositions) reste difficile à assurer. Il y a bien deux mondes, celui des figures (et des outils de traçage et de transformations) et celui des propositions, et ces mondes communiquent encore très imparfaitement (ce qui est attesté par l'impossibilité de reconnaître deux figures identiques différant par leur mode de construction ainsi que certaines difficultés d'articulation entre les aspects continus et discrets...). Le travail de Trilling (1996) et Desmoulins (1996), sur la vérification de la correction d'une figure vis-à-vis d'une spécification, montre le chemin qui reste à parcourir.

En fait, les aspects les plus complexes des micromondes, développés dans un premier temps pour des élèves jeunes, dans le cadre d'environnements simples, restent encore difficiles à mettre en œuvre. Il faut pouvoir *embarquer* suffisamment de connaissances dans les environnements, faciliter leur manipulation et contrôler / aider / vérifier cette manipulation.

#### **4.5. Des environnements ouverts à la découverte guidée**

Le développement de l'informatique, notamment celui des interfaces, conduit peu à peu à accorder une place croissante aux aspects graphiques. Concernant les résolveurs de problèmes, la boîte noire cède la place à la boîte de verre et des primitives sont fournies aux apprenants pour conduire des démonstrations ; la simulation va suivre des chemins analogues. A un modèle déjà implanté ne laissant à l'utilisateur d'autres choix que de modifier la valeur de certains paramètres prévus à l'avance vont se substituer des environnements ouverts, détaillant en pas à pas leur fonctionnement et fournissant des modes d'accès ou de modification au modèle sous-jacent à la simulation.

En outre, l'essor de la micro-informatique élargit considérablement le champ des utilisateurs potentiels, ces utilisateurs n'ayant ni le temps ni les compétences nécessaires à l'acquisition de savoir-faire trop complexes. Les acquis des recherches sur l'apprentissage à l'aide d'ordinateurs, d'enfants jeunes, ouvrent des pistes intéressantes pour la réalisation d'environnements de production diffusés à une très large échelle. L'idée de métaphore, c'est-à-dire l'établissement d'une certaine proximité ou ressemblance entre des modèles mentaux et des représentations informatiques, s'impose dans la conception des interfaces homme-machine. L'objectif est de donner des possibilités réelles d'expression et d'exploration ne nécessitant pas de fortes contraintes concernant le dispositif lui-même. Les jeux, les progiciels généraux (traitements de textes, tableurs...) vont largement s'en inspirer si bien que l'on peut se demander maintenant si la notion de micromonde ne se réduit pas à une forme d'interface particulière, popularisée dans les environnements orientés objets.

De multiples outils informatisés sont maintenant disponibles pour produire, modéliser, explorer, etc. Raccourci d'une certaine démarche informatique, au lieu de se créer des représentations particulières pour aborder et résoudre des problèmes délicats, les représentations pertinentes sont rigidifiées et diffusées, évitant un processus complexe de reconstruction. L'apprenant est doté de formidables moyens d'expression et d'exploration. Comment peut-il en tirer bénéfice ?

Nous allons essayer de donner quelques repères sur ces environnements dits ouverts, tenter d'éclaircir leurs rapports avec la notion de micromonde et en dégager certaines limites dans leur usage éducatif.

##### **4.5.1. Les grands types de modèles**

Dans les environnements informatiques qui se popularisent, il s'agit de piloter des dispositifs technologiques par la manipulation de représentations symboliques



qui réfèrent à des objets ou des actions dans le monde réel. A la suite des micromondes, l'idée de fournir aux apprenants des outils d'apprentissage par exploration se développe. La notion de modèle mental apparaît alors au cœur du problème et deux questions centrales émergent (ESRC, 1988) :

(1) De quelles façons l'interaction avec des outils intégrant des représentations d'un domaine peut faciliter l'apprentissage de ce domaine ?

(2) Le fait de représenter et d'explorer les conséquences de leur propre modèle mental constitue-t-il une aide pour les apprenants ?

Du point de vue des différents outils permettant un apprentissage par la découverte, on distingue généralement deux grandes classes : les outils d'expression, c'est-à-dire de modélisation et les outils d'exploration. Dans les deux cas, des activités de résolution de problèmes peuvent être proposées (par exemple, construire un objet particulier, obtenir un état particulier dans une simulation, détecter la cause d'un dysfonctionnement...). Trois types de modèles peuvent être construits (figure 9), les modèles quantitatifs, les modèles semi-quantitatifs et les modèles qualitatifs.

	Expression	Exploration
Quantitatif	Systèmes de modélisation Tableurs	Simulations scientifiques
Semi-quantitatif	STELLA <sup>21*</sup> Alternate Reality Kits	Certaines simulations
Qualitatif	Systèmes experts Générateurs d'histoire Jeux d'aventure	Systèmes d'aide à la décision Simulations non quantitatives

**Figure 4.9** : Différentes classes de modèles (ESRC, 1988)

Le quantitatif travaille sur des grandeurs mesurables. On peut créer et/ou explorer des relations entre ces grandeurs. L'usage de simulations quantitatives pour faire des inférences est à double tranchant. En effet, beaucoup de connaissances sont encodées, basées sur des implicites et des décisions empiriques, les limites de validité ne sont pas données explicitement et il n'est pas simple de les rendre explicites et vérifiables. Le qualitatif travaille sur des catégories et des grandeurs non mesurables (souvent un choix parmi un nombre fini de possibles). Dans le semi-quantitatif, le raisonnement porte sur des grandeurs dont on cherche essentiellement le signe et pas la valeur, c'est-à-dire on s'intéresse à la direction et non à la grandeur des effets d'une partie d'un système sur une autre. On cherche à savoir si, par

21. STELLA (Withfield, 88) est un logiciel (*Structural Thinking, Experiential Learning Laboratory with Animation*) interactif pour la production de modèles mathématiques. Il semble bien adapté à l'enseignement de la modélisation.

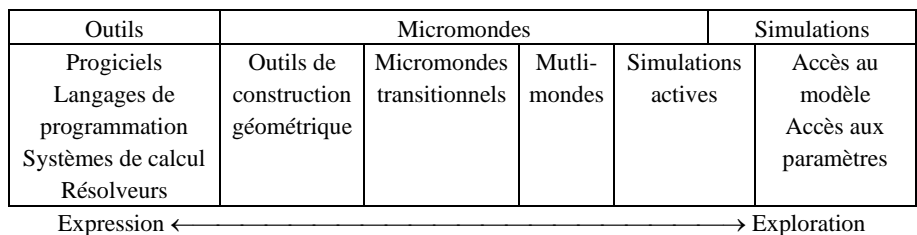
exemple, quand une grandeur donnée augmente, une autre augmente ou diminue (ou reste constante).

La modélisation faisant partie intégrante des programmes scolaires en Grande-Bretagne, une recherche coopérative a été lancée. Elle a d'abord conduit à la réalisation d'un système nommé IQON pour créer des modèles semi-quantitatifs. Les résultats obtenus (Miller et al. 1993) indiquent notamment que des enfants entre 11 et 14 ans pensent plus facilement en termes d'objets et d'événements qu'en termes de variables et qu'ils peuvent s'attaquer à des tâches de modélisation d'une complexité raisonnable. Ils peuvent comprendre des modèles qu'on leur donne à explorer plus complexes que ceux qu'ils sont eux-mêmes capables de construire.

**4.5.2. Caractéristiques générales des environnements ouverts**

Vu leur large spectre, il n'est pas si facile de situer les environnements d'apprentissage les uns par rapport aux autres. En effet, de nombreuses variables interviennent : l'âge des apprenants, le domaine d'enseignement concerné et les types de connaissances en jeu, l'interface, l'équilibre entre les outils fournis pour construire et la complexité du monde à explorer, le degré de contrôle et d'assistance offert par la machine, le type de formation (scolaire ou professionnelle), etc. Nous en retiendrons deux : la focalisation vers des activités d'expression ou d'exploration et l'objectif tourné plutôt vers la réalisation d'un produit ou le processus même de construction.

Partant des activités essentiellement axées vers l'expression à celles plus axées vers l'exploration, nous pouvons nous risquer à une classification relativement sommaire allant des outils aux simulations.



**Figure 4.10.** Classification des environnements d'apprentissage

*Outils.* Les outils permettant de modéliser des situations sont très nombreux : d'une part, les programmes classiques disponibles sur un ordinateur depuis le milieu des années quatre-vingt, tels traitements de textes, tableurs, bases de données,

grapheurs, gestionnaires de plans... et, d'autre part, les programmes plus spécialisés comme les logiciels de calcul formel ou les systèmes experts.

*Micromondes.* Nous avons vu la large palette d'acceptions de ce terme. On peut situer les micromondes à la frontière entre les outils (de type progiciel très spécialisé) et les simulations, à cheval entre la modélisation et l'exploration. On peut ranger dans cette classe, les outils de construction géométrique, les micromondes transitionnels, les multimondes et les simulations actives.

*Simulations.* Elles consistent en un modèle qui est la représentation informatique d'un système, d'un processus ou d'une machine et d'une interface usager. Des aspects de ce modèle sont présentés à l'apprenant qui peut tester des hypothèses en modifiant certains paramètres et en observant sur l'écran les effets produits voire changer différentes caractéristiques du modèle lui-même. En ce sens, les simulations actives sont à la fois des simulations et des micromondes du fait que l'utilisateur a la possibilité de définir des modes de fonctionnement particuliers, de modifier la granularité des actions qui lui sont proposées et d'en créer de nouvelles.

Si, dans tous les cas, des activités de résolution de problèmes sont possibles, la relation entre la tâche et la finalité recherchée, c'est-à-dire le rôle de l'environnement dans l'apprentissage visé, est essentielle. Les activités d'expression conduisent souvent à la *production d'un objet*, un dessin, un texte, une figure, un modèle, etc. L'intérêt peut être lié au produit obtenu, au processus même de production et parfois aux deux.

En se *centrant sur le produit obtenu*, on distingue :

- *la production d'un objet final*, le produit est le résultat de l'activité. L'environnement offre un cadre pour la facilitation de cette production. Il est souvent conçu comme une extension d'outils traditionnels.

- *la production d'un objet transformable.* L'environnement offre, dans un premier temps, des outils facilitateurs pour cette production. Dans un second temps, de nouveaux outils permettent d'explorer la situation obtenue, mettant en évidence des formes de propriétés émergentes. De la même manière, l'objet produit peut être une sorte de modèle. Il s'agit de faciliter la production du modèle, puis d'offrir des outils permettant son exploration et éventuellement sa validation.

Si on *se centre sur le processus même de production*, cette production procure une certaine motivation pour réaliser l'activité proposée mais n'est pas son but profond. On cherchera à faciliter la prise de conscience d'erreurs éventuelles (jugement direct sur la production, comme en LOGO tortue) afin de déclencher un retour sur le mode de production. La *réification*<sup>22</sup> de ce processus de production est

---

22. « On peut penser que le rappel concret de ce qu'a fait l'élève peut l'aider dans ce processus (construction consciente de son savoir). Ce qui manque jusqu'à présent, c'est la rétroaction sur le savoir que l'élève vient de construire. Dans une certaine mesure,

centrale (transformation du processus en un objet sur lequel on peut opérer), en offrant à l'apprenant une réflexion sur sa propre pensée (plus exactement une trace contrainte de sa pensée).

Cette distinction entre la focalisation sur la production elle-même ou sur le processus de production conditionne les choix effectués. Dans le premier cas, le critère essentiel est celui de la performance, s'agissant d'optimiser la qualité ou le temps passé à la réalisation souhaitée. Dans le second cas, les outils fournis ne sont pas forcément les plus adéquats, leurs contraintes de fonctionnement pouvant être jugées utiles dans les situations proposées. Néanmoins, une telle séparation n'est pas toujours aussi tranchée. En effet, l'objet d'apprentissage est finalement la tâche ou le modèle, les liens importants entre les deux interdisant des séparations trop nettes. Ainsi, mener des activités d'apprentissage *par* la programmation implique qu'on ait les éléments suffisants d'apprentissage *de* la programmation. De même, pour motiver effectivement l'apprenant, on ne peut mésestimer l'intérêt intrinsèque du résultat à produire. Le tout est de savoir si le but recherché est que l'apprenant sache réaliser une tâche particulière ou réorganise ses connaissances pour faire face à de nouvelles tâches. S'agit-il de faciliter la réalisation de la tâche ou la compréhension de la tâche elle-même ?

Concernant des activités d'exploration, on retrouve un clivage analogue. On peut se centrer sur la compréhension d'un modèle particulier, avec des activités d'entraînement avec ce modèle, ou sur les concepts généraux associés au(x) modèle(s) traités, avec exploration des liens entre ces concepts et les modèles manipulés. Notons enfin que les activités de résolution de problèmes font le lien entre expression et exploration. Il peut être nécessaire de se créer son langage ou sa représentation, de construire un modèle qui permettra d'effectuer la résolution.

Un environnement d'apprentissage revient à l'interaction de trois modèles : le modèle représenté à l'écran, le modèle sous-jacent à cette représentation et le modèle mental de l'utilisateur ou de l'apprenant.

*Le modèle représenté à l'écran* est caractérisé par :

– *son degré de complexité* qui, dans un souci de performance ou d'appropriation initiale, doit être minimisé pour garantir le succès de l'utilisateur. L'augmentation (ou la diminution) de cette complexité peut s'effectuer par l'adaptation progressive des outils fournis (avec modification éventuelle des buts successifs).

– *son niveau d'abstraction*, c'est-à-dire sa proximité avec le modèle réel simulé ou de référence (ou le modèle mental initial de l'utilisateur). Un faible niveau d'abstraction est recherché pour simplifier la prise en main, la familiarité avec les

---

*l'ordinateur pourrait les déduire de la structure de la représentation de la même manière que nous demandons aux élèves de procéder à ce type d'inférences. » (Lesgold, 1987)*

objets de l'écran facilitant leur manipulation. Une question importante est de savoir si cette familiarité peut être suffisante ou suffisamment extensible pour se passer d'un niveau d'abstraction plus élevé.

– *son type*, c'est-à-dire son mode de fonctionnement. Les transformations que l'on peut effectuer sont contraintes par les comportements possibles de ces objets (modèle causal, déduction à partir d'un modèle formel, ou modèle par contraintes).

Le *modèle mental* correspond à la vision de l'apprenant et la finalité de l'activité est souvent liée à une recherche d'évolution de ce modèle. Suivant les cas, des distances plus ou moins grandes peuvent séparer le modèle mental et le monde représenté. Dans le cas des outils, on cherche à la minimiser, dans une optique de réalisation nécessitant un apprentissage rapide. Dans le cas des micromondes transitionnels, les écarts sont souvent plus importants, le modèle mental permettant une sorte de contrôle sémantique pour décider de la justesse de ce qui est produit. La difficulté est de préserver un décalage entre les conceptions de l'utilisateur et le comportement du système pour laisser la place à des phénomènes d'apprentissage (Laborde, 1995).

Le *modèle sous-jacent* peut être l'objet d'apprentissage (dans les micromondes transitionnels). S'agissant d'outils pour réaliser des tâches quelconques, ce modèle n'est pas censé concerner l'usager. L'accès au monde formel est conditionné par le type de modèle utilisé pour le monde représenté. D'après White (1993), dans le domaine de la physique, seul un modèle causal est adéquat.

Notons que la plupart du temps, les environnements ne se limitent pas à un seul modèle mais proposent une succession de modèles intermédiaires de complexité croissante de par leur représentation à l'écran, les outils mis à disposition des apprenants ou les activités proposées. Une telle suite de modèles favorise un apprentissage par paliers : construction de nouvelles structures à partir d'opérations élémentaires familières, puis découverte de relations nouvelles. Acquérir de nouvelles connaissances, c'est définir de nouvelles actions, aspect génétique qui se traduit dans l'environnement proposé, soit par des modifications faites par l'apprenant, soit par le système. Les simulations, au sens classique, sont plus statiques au regard du développement des connaissances (Balacheff, 1994). Dans les micromondes transitionnels de mathématiques que nous avons vus, l'enjeu est le système formel, on cherche à favoriser un processus d'induction à partir des exemples, structurés par ce système formel, exemples familiers des apprenants, répondant à des règles équivalentes (numération, structures additives, algèbre élémentaire...).

Pour conclure, on peut considérer qu'un environnement d'apprentissage correspond à une sorte de *représentation intermédiaire exécutable* qui entretient des rapports avec d'autres mondes. Cette représentation suscite des activités

d'exploration et d'expression avec diverses limitations imposées ainsi que des facilités pour réfléchir sur les activités conduites grâce à une certaine *réification de cette activité* (consistant par exemple en une restructuration des différentes étapes suivies par l'apprenant). Finalement, la notion de micromonde se rapproche des travaux sur les représentations intermédiaires<sup>23</sup> désormais classiques dans la littérature sur l'EIAO, à la différence qu'avec ces dernières, un cadre plus contraint limite les possibilités créatives des élèves pour favoriser un contrôle plus fin de la machine sur leur activité.

#### 4.5.3. *Limites et difficultés*

La littérature sur les micromondes et sur les environnements d'apprentissage est abondante. Si le travail avec des tels environnements semble séduisant au plan éducatif, les évaluations effectuées montrent que les choses sont plus complexes qu'il n'y paraît de prime abord. Les potentialités sont certes énormes mais les contraintes d'usage amènent à tempérer les enthousiasmes des promoteurs. L'un des écueils traditionnellement repéré est le manque de soutien pouvant être fourni à l'apprenant suivant ses propres démarches. Ce dernier peut rencontrer de sérieuses difficultés pour atteindre son but sans que le système ne puisse lui prodiguer l'assistance requise<sup>24</sup>. En effet, par opposition aux tuteurs, la machine ne contrôle pas l'adéquation entre les outils utilisés par l'apprenant et l'objectif qu'il poursuit. Si l'ouverture est grande, elle s'opère au détriment de l'assistance et du contrôle de la tâche effectuée.

Les recherches convergent sur l'idée de la nécessité d'une assistance durant l'activité. Si cette aide peut être assurée par l'enseignant, en supposant qu'il soit effectivement apte à le faire, la question est de savoir si elle peut être, au moins en partie, assumée par le système lui-même. Introduire des modes de guidage automatisés amène à s'interroger sur leur nature : assistance dans la réalisation des tâches, conseil sur l'usage (ou le non-usage) des outils proposés ou sur la stratégie à employer, explications sur le comportement du système...Par exemple, Thompson (1987), dans son micromonde sur les isométries planes, ajoute une commande nommée *Pourquoi?*. Son rôle est d'expliquer les actions effectuées lorsque leur résultat est inattendu et que les apprenants n'arrivent pas à comprendre un phénomène. La conception préalable d'un ensemble structuré d'activités constitue déjà une forme de guidage ou d'orientation qui semble s'imposer pour parcourir un certain curriculum. Même si, s'agissant d'une production, la réaction de la machine

---

23. Schwarz et Resnick (1993) parlent de modèle intermédiaire, système formel dont les objets se comportent comme les objets mathématiques qu'ils exemplifient et dont le fonctionnement est auto-évident. La connaissance est embarquée dans la sémantique des actions permises à l'utilisateur.

24. Notons que ce problème est analogue à celui de la définition d'outils de navigation dans les hypertextes (voir chapitre 6).

peut porter sur le produit réalisé, non sur la personne qui le réalise, introduire du guidage change la nature de l'environnement. Comme le remarque Solomon (op. cit., p. 42) à propos de WEST, le rôle de la machine n'est plus le même et on peut se demander qui pilote réellement de l'apprenant ou de la machine.

Le problème central est en fait d'arriver à combiner la résolution de problèmes et la motivation de l'apprentissage par la découverte avec un guidage effectif. Les systèmes doivent ainsi d'une part fournir des outils aux utilisateurs mais d'autre part les aider dans leurs tâches. Cette assistance va de la simple rétroaction, visuelle et/ou langagière jusqu'à un suivi en pas-à-pas sur un mode préceptoral en passant par des formes d'aides interactives de type passif (c'est-à-dire fournies à l'initiative de l'apprenant) ou plus actives (décidées par le système). Toutefois, trouver la meilleure forme de guidage reste une tâche délicate. Les théories de *scaffolding* (échafaudage) cherchent à définir des stratégies pédagogiques explicites pour fournir exactement la bonne dose d'assistance afin d'éviter que l'édifice de compréhension ne s'effondre et pas trop pour éviter que le mécanisme d'assistance vienne obscurcir et étouffer la compréhension. On peut se contenter de mettre de l'intelligence dans les rétroactions immédiates fournies, prenant en compte les intentions de l'utilisateur, bien qu'il soit difficile de les deviner si elles ne sont pas déclarées explicitement. Les chercheurs vont chercher à combiner interaction tutorielle et apprentissage par la découverte, en limitant la combinatoire possible pour être à même d'effectuer un certain suivi de l'activité de l'apprenant. C'est ce que nous allons voir au chapitre suivant.

Pour terminer, nous avons attiré l'attention du lecteur, en introduction, sur les interactions entre science, technologie et éducation et sur les difficultés d'harmonisation des différents points de vue associés plus spécifiquement à chacun de ces axes. Le cas des micromondes semble s'appuyer sur un bon compromis, mais pose néanmoins des problèmes délicats :

- du côté de la science, comment légitimer le bien fondé des constructions effectuées et quelle(s) méthodologie(s) d'évaluation employer ?
- du côté des techniques, comment les améliorer pour les rendre réellement accessibles, garantissant à la fois simplicité et puissance d'expression ?
- du côté de l'éducation, comment rendre compatibles les environnements et processus souhaités avec le fonctionnement scolaire usuel ?