

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

## Chapitre 3

# Les apports de l'intelligence artificielle

### 3.1. Introduction

Nous avons vu que l'enseignement programmé est finalement, tout au moins dans la vision américaine, la rencontre entre une théorie psychologique et des machines à enseigner. L'essor des ordinateurs et l'extension des recherches conduisent à l'EAO. La complexité croissante des programmes à construire pour assurer une adaptation optimale à l'apprenant rend peu à peu indispensable le recours à des techniques de programmation avancée et des solutions, au moins partielles, vont être offertes par l'intelligence artificielle. De programmes fermés, ne tenant pas compte des réponses de l'apprenant (censées être correctes dans le modèle skinnérien), on arrive à des programmes « connaissant » suffisamment ce qu'ils enseignent pour générer des exercices et leurs solutions ou répondre à des questions.

Si la cybernétique a convaincu de l'intérêt de contrôler le processus d'enseignement, aucune théorie de l'apprentissage n'est suffisamment explicite pour permettre d'exercer efficacement ce contrôle à l'aide d'une machine. Dans l'attente de nouvelles théories et même pour participer à leur élaboration, d'autres voies semblent intéressantes à explorer. Or, dans un souci de participation plus active de l'apprenant, lui laisser une certaine marge d'initiative paraît important. A un contrôle parfois « tatillon », se substitue un pilotage plus souple, basé sur des principes de guidage relativement généraux, souvent empruntés aux enseignants eux-mêmes, bien qu'aucune théorie scientifique ne permette réellement d'en étayer la validité. Le cheminement qui va être suivi par l'élève n'est plus entièrement prédéterminé, ce qui conduit à séparer le matériel de cours et les stratégies d'enseignement. Cette séparation offre de nouvelles possibilités d'adaptation en permettant notamment de générer problèmes et commentaires de remédiation spécifiquement pour chaque étudiant.

Mais au-delà de l'initiative que l'on souhaite laisser à l'apprenant, l'engager dans une véritable activité apparaît de plus en plus important. Une session EAO se déroule de manière essentiellement cyclique : présentation d'informations puis d'une question, réponse de l'apprenant à la question posée, analyse de cette réponse, branchement. S'agissant de faire réaliser des tâches complexes à des apprenants, une telle boucle de contrôle n'est plus adéquate. Il faut concevoir des programmes donnant aux apprenants les moyens de réaliser des tâches, capables d'observer et d'interpréter leur comportement puis d'intervenir au moment opportun et de manière appropriée sur la base de leur analyse. La réalisation de tels projets nécessite des connaissances, tant au niveau des processus de résolution suivis par les humains qu'au niveau de l'implantation en machine de tels processus, cœur du projet scientifique de l'intelligence artificielle. Les différents travaux sur les processus mentaux, développés à la fin des années cinquante et au cours des années soixante par Newell et Simon, notamment les analyses de protocoles de résolution de problèmes (enregistrement des sujets exprimant à voix haute les mouvements de leur recherche mentale pour résoudre un problème) et les différents programmes de résolution développés sur ordinateur ont ouvert la voie.

L'apport de l'intelligence artificielle apparaît comme celui d'une ingénierie de la connaissance. Quelles connaissances peut-on représenter dans une machine et quelles améliorations peuvent-elles apporter dans l'enseignement avec l'ordinateur ? De nombreuses recherches vont s'intéresser à la résolution de problèmes : comment résoudre un problème ? Comment apprendre à résoudre un problème ? Comment une machine peut-elle résoudre un problème ? Comment une machine pouvant résoudre un problème peut-elle enseigner à un élève ou aider un élève à résoudre...

Pour réaliser des tuteurs artificiels pleinement adaptatifs, il faut implanter des connaissances en machine pour les trois composantes qui interagissent dans la formation : le sujet, l'élève et le professeur, c'est-à-dire *quoi, pour qui et comment* (*What, who, how*, Self, 1974). Dans le présent chapitre, nous allons nous centrer sur les apports de l'intelligence artificielle pour le traitement de ces trois composantes et nous renvoyons le lecteur au chapitre cinq consacré aux tuteurs (artificiels) intelligents pour une vision plus générale. Tout d'abord, nous verrons les stratégies d'enseignement intégrées à des systèmes laissant une certaine marge de manœuvre à l'apprenant. Ensuite, nous traiterons le problème central de la modélisation de l'apprenant. Enfin, nous allons nous intéresser à la représentation du domaine dans le cadre de la résolution de problèmes, en nous appuyant essentiellement sur les mathématiques, champ qui a été le plus exploré. Rappelons qu'une des exigences pour qu'un système puisse être réellement adaptatif est qu'il puisse se modifier lui-même. Il est ainsi légitime d'étudier la façon dont les machines peuvent effectivement apprendre et quelles en sont les incidences pour l'enseignement. Mais, au-delà même de l'apprentissage, la possibilité de découverte automatique ne peut être passée sous silence.

Si, dans un premier temps, l'application des techniques de l'intelligence artificielle s'est fondue dans le courant de la technologie éducative, l'apport de capacités de raisonnement à des machines à enseigner va peu à peu conduire à modifier le statut et les rôles possibles de cette machine. Nous tenterons de le montrer à la fin de ce chapitre.

### 3.2. Expérimenter de nouvelles stratégies d'enseignement

L'objectif de l'EAO est de construire des programmes d'instruction incorporant du matériel de cours bien préparé, découpé en leçons optimisées pour chaque étudiant. Comme nous l'avons déjà mentionné, le déroulement d'une session est cyclique : présentation d'informations et d'une question, réponse de l'apprenant, analyse de la réponse puis continuation ou branchement à une autre partie du cours. Toutes les questions doivent être soigneusement spécifiées par l'auteur du cours, de même que l'ensemble des réponses attendues des apprenants à chacune d'entre elles. Si un branchement doit intervenir, le critère de déclenchement et le nouveau point de continuation dans le programme doivent être donnés de façon explicite. Une telle approche, qui contraint de tout spécifier à l'avance, présente deux inconvénients majeurs. La préparation du cours requiert un temps considérable et le résultat obtenu est très peu flexible, et donc difficilement modifiable. Changer un ensemble de questions ou des éléments du cours conduit à un gros travail de réécriture du programme.

Si les programmes dits génératifs, présentés dans le chapitre précédent, permettent de dépasser les limitations qui viennent d'être évoquées, les premières réalisations se limitent à des domaines essentiellement numériques. Il s'agit alors d'étendre cette approche à d'autres domaines, en donnant des connaissances et certaines capacités de raisonnement aux systèmes.

Dans un premier temps, les programmes qui sont développés s'appuient surtout sur les éléments de connaissance qu'ils incorporent pour laisser une plus grande marge d'initiative à l'apprenant. Il n'y a pas de référence à des théories de l'apprentissage très précises, si ce n'est la mention de l'intérêt de l'*apprentissage par l'action* dans la résolution de problèmes (Koffman, 1972). Comme une planification complète a priori, c'est-à-dire prévoyant explicitement chaque réaction du programme associée à chacune des réactions possibles de l'élève n'est plus ni possible, ni souhaitable, il faut se baser sur des principes généraux pour piloter le dialogue avec l'apprenant. Les chercheurs vont ainsi s'inspirer des dialogues socratiques, définir des modes de guidage favorisant un apprentissage par la découverte et développer un mode d'intervention discret sur le modèle de l'entraîneur sportif (*coach*). Les stratégies d'enseignement mises en place vont aussi s'appuyer sur une évaluation des connaissances et des faiblesses des apprenants, c'est-à-dire sur une forme de modélisation de l'apprenant.

En dehors des articles spécialisés, certains d'entre eux étant réunis dans le recueil de Sleeman et Brown (1982), la plupart des systèmes brièvement décrits ci-dessous sont présentés dans des ouvrages de synthèse : Barr et Feigenbaum (1982), O'Shea et Self (1983) et surtout Wenger (1987), synthèse très complète sur les tuteurs intelligents. Une littérature abondante et suffisamment accessible étant disponible, nous nous contenterons d'illustrer les grands courants de recherche par ces différents systèmes. Notons qu'une grande partie d'entre eux (SCHOLAR, WHY, SOPHIE, WEST) a été développée au sein d'un laboratoire privé, BBN (Bolt, Beranek et Newman Inc.) situé à Cambridge dans le Massachussets.

### **3.2.1. De nouvelles formes de dialogue avec l'apprenant**

Pour engager un dialogue avec l'apprenant sur un domaine, les systèmes doivent disposer d'une certaine compréhension de ce domaine, c'est-à-dire d'un formalisme pour représenter les connaissances de ce domaine et de mécanismes spécifiques pour les manipuler. Mais le dialogue étant finalisé, l'objectif étant de faire acquérir certaines connaissances à l'apprenant, les systèmes doivent suivre des stratégies particulières, de type tutoriel. Le problème est de déterminer jusqu'à quel point ces stratégies de dialogue peuvent être indépendantes du domaine spécifique enseigné et si elles permettent effectivement de s'adapter aux apprenants.

#### *3.2.1.1. SCHOLAR et l'initiative mixte*

L'*initiative mixte* caractérise un mode d'interaction dans lequel l'apprenant et le système peuvent, à tour de rôle, prendre l'initiative et poser des questions. Le programme SCHOLAR, généralement considéré comme le premier tuteur intelligent, ou tout au moins comme le premier programme d'enseignement utilisant les techniques d'intelligence artificielle, permet ce type de dialogue. Ce programme, écrit par Carbonell (1970), est conçu pour enseigner des connaissances factuelles sur la géographie de l'Amérique du Sud. Dans l'exemple de session présenté figure 3.1, on peut remarquer que l'élève interroge le système en langue naturelle (en utilisant toutefois des phrases simples) et que ce dernier est capable de répondre à des questions très générales comme " Parlez-moi du ..." et de revenir ensuite au sujet de conversation précédent.

<p><b>O) Quel est la langue parlée en Argentine ?</b>  E) Probablement l'espagnol  <b>O) Très bien.</b>  <b>Quelle est la surface approximative de l'Argentine ?</b>  <i>...L'élève change de sujet.</i>  E) Parle-moi du Pérou.  <b>O) Le Pérou est un pays.</b>  <b>Il est situé en Amérique du Sud.</b>  <b>La capitale du Pérou est Lima.</b>  <b>Maintenant répondez à la question à laquelle vous n'avez pas répondu avant.</b>  <i>... L'élève y répond.</i>  E) La surface de l'Argentine est de 1 100 000 miles carré.</p>	<p><b>O) Approximativement correct, mais une valeur plus exacte est 1 200 000.</b>  <b>Le pays frontalier à l'est du Pérou est ---</b>  E) Catholicisme.  <i>...L'étudiant fait une erreur qui est corrigée.</i>  <b>O) Faux. Le catholicisme n'est pas un pays mais une religion.</b>  <b>La réponse correcte est : Brésil.</b>  <b>Où est le Pérou ?</b>  <i>... l'élève change de nouveau de sujet.</i>  E) Est-ce qu'il fait chaud à Asuncion ?  <b>O) Comme Asuncion fait partie du Paraguay, oui.</b>  E) Quelle est la différence la plus importante entre le Pérou et le Paraguay ?</p>
---	---

**Figure 3.1.** Session (traduite librement) avec SCHOLAR (Barr et Feigenbaum, op.cit., p. 238).  
Les interventions de l'ordinateur sont en gras et les commentaires en italique.

SCHOLAR utilise un réseau sémantique pour représenter la connaissance. A chaque nœud de ce réseau, qui correspond à un objet ou un concept géographique, est associé un ensemble de propriétés (des listes attribut-valeur) et à chacune de ces propriétés est affecté un indice d'importance, utilisé pour mesurer le degré de pertinence d'une propriété selon le contexte du dialogue. Des mécanismes de parcours dans le réseau sémantique et des règles d'inférence permettent de répondre aux questions posées par l'apprenant. Un agenda conserve la liste des sujets qui sont abordés. A l'aide d'un mécanisme de *raisonnement plausible* sur un ensemble de connaissances incomplet, SCHOLAR est capable de répondre à des questions sur des faits non préalablement stockés en mémoire (Barr et Feigenbaum, op.cit., p. 239). Avec cette dernière possibilité, le programme peut *s'adapter* à des situations inattendues, illustrant la supériorité de cette approche vis-à-vis des approches classiques de l'EAO.

Si les stratégies implantées dans SCHOLAR sont indépendantes du contenu même du réseau sémantique et peuvent ainsi être utilisées dans d'autres cadres, elles semblent plus délicates à appliquer à des domaines ne concernant plus simplement des faits mais aussi des procédures. En outre, la session présentée en figure 3.1 montre bien que la continuité dans le dialogue est faible. SCHOLAR n'établit aucun plan pour traiter d'un point spécifique à l'aide d'une succession de questions.

NLS-SCHOLAR (Grignetti et al., 1975), conçu pour apprendre à des utilisateurs novices à utiliser l'éditeur NLS<sup>1</sup>, reprend les idées de SCHOLAR pour les appliquer à un domaine non plus simplement factuel, mais aussi procédural. Système à initiative mixte, l'étudiant peut poser des questions ou y répondre, réaliser des tâches qui lui sont proposées ou qu'il choisit, NLS-SCHOLAR complète l'apprentissage par explication et par questionnement par une forme d'apprentissage par l'action. L'utilisateur peut non seulement décrire une commande mais aussi l'écrire directement en NLS, c'est-à-dire essayer ce qu'il apprend sur le système même qu'il est en train d'apprendre. De son côté, le système peut aussi prendre la main et montrer comment réaliser des actions d'édition. Grâce au couplage avec NLS, il lui est possible de tenir compte du contexte courant de l'utilisateur pour particulariser sa réponse à la situation en cours. Utilisant un réseau sémantique comprenant les descriptions des actions et de leurs buts, la description des procédures nécessaires pour accomplir ces actions ainsi que leurs effets et conséquences, NLS-SCHOLAR est apte à répondre à des questions du type *que se passerait-il si ?* Autant système de formation que système d'aide, NLS-SCHOLAR préfigure les assistants en ligne.

Afin d'étendre les capacités de SCHOLAR dans la conduite de dialogues tutoriels, Collins a analysé des protocoles de dialogue de ce type avec des enseignants (Wenger, op.cit., p. 37). Il en a tiré un ensemble de règles, permettant de conduire des dialogues de type socratique, qu'il a utilisé ensuite dans un domaine, la météorologie, dont les connaissances ne sont pas uniquement d'ordre factuel. Ce système, nommé WHY (Stevens *et al.*, 1982), traite de l'étude des causes des chutes de pluie.

### 3.2.1.2. WHY et le dialogue socratique

L'objectif d'un dialogue de type *socratique* consiste à conduire l'apprenant, par des questions successives, à formuler des principes généraux à partir d'exemples, à examiner la validité de ses propres hypothèses afin d'y déceler éventuellement des contradictions pour finalement extraire les déductions correctes à partir des faits qu'il connaît<sup>2</sup> (Wenger, op.cit., p. 39). Cela implique, pour le tuteur, une compréhension des hypothèses formulées par l'apprenant et la maîtrise d'un dialogue orienté vers la recherche de contradictions.

Le fonctionnement de WHY est conforme aux principes précédents. Le système demande à l'étudiant de proposer différentes causes pouvant expliquer les chutes de pluie, de rechercher les causes principales et annexes puis de formuler une règle générale. Tant que la règle proposée est fautive ou incomplète, le système fournit un

---

1. NLS est le système développé par Engelbart et son équipe. Nous y reviendrons au chapitre 6 consacré à l'hypertexte.

2. Notons que d'autres auteurs, pour favoriser une participation plus active de l'apprenant, suggèrent d'adopter une sorte de perspective socratique inversée, en le laissant poser des questions et en répondant à celles-ci. L'apprenant reprend alors le contrôle de l'interaction.

contre-exemple et demande à l'apprenant de modifier cette règle en conséquence jusqu'à ce qu'une règle correcte et suffisamment générale soit trouvée. Le contrôle du dialogue est assuré par un ensemble d'heuristiques, traduisant la méthode socratique. En voici un exemple : « Si un étudiant donne comme explication un facteur qui n'est pas suffisant, alors donnez-lui un contre-exemple, c'est-à-dire un exemple dans lequel ce facteur est présent mais donne un résultat différent. Par exemple, si un étudiant dit que la présence d'eau explique le fait qu'on fasse pousser du riz en Chine, demandez-lui si on fait pousser du riz en Irlande » (Collins, 1977)<sup>3</sup>. WHY utilise une représentation des connaissances sous forme de scripts, c'est-à-dire de suites ordonnées d'événements correspondant aux différentes étapes temporelles ou causales des processus agissant sur la chute des pluies.

Toutefois, deux faiblesses importantes du système apparaissent : l'absence d'une stratégie tutorielle globale et l'insuffisance de la représentation des connaissances à base de scripts, à la fois pour expliquer le processus de chute des pluies et pour diagnostiquer et corriger les conceptions erronées des apprenants. Afin d'exercer un contrôle plus global dans la conduite du dialogue, l'analyse de nouveaux protocoles d'enseignants permet de mettre en évidence les stratégies qu'ils utilisent, correspondant à deux buts généraux : perfectionner le *modèle causal* de l'apprenant en commençant par les facteurs les plus importants et en ajoutant graduellement d'autres facteurs plus subtils et affiner ses procédures en appliquant ce modèle à de nouvelles situations. C'est le développement chez l'apprenant d'un tel modèle causal, c'est-à-dire la compétence de répondre à des questions, de fournir des explications et de faire des prédictions sur les relations causales concernées par les chutes de pluie, qui est l'objectif de WHY. Pour les auteurs (Stevens *et al.*, op.cit.), les dialogues tutoriels sont analysés comme un processus d'interaction dans lequel le tuteur cherche à “ déboguer ” les connaissances de l'apprenant en diagnostiquant et en corrigeant ses incompréhensions conceptuelles. Une analyse de ces conceptions erronées les conduit alors à l'idée que représenter des connaissances sur des processus physiques nécessite de *multiplier points de vue*. A la représentation initiale à base de scripts, doit s'ajouter la représentation des relations fonctionnelles entre les divers éléments qui interviennent. Ces points de vue multiples, c'est-à-dire la connaissance du tuteur sur le domaine, vont déterminer la structure de but du dialogue et le processus de correction (diagnostic d'erreur et remédiation).

En analysant le travail effectué autour de WHY, il est intéressant de constater qu'une recherche sur un modèle d'enseignement a finalement abouti à la nécessité d'analyser en profondeur les conceptions des apprenants, conduisant à modifier à la fois les stratégies d'enseignement et la représentation des connaissances du domaine. Au-delà de WHY, les recherches sur les modèles mentaux et les points de vue multiples se sont d'ailleurs poursuivies avec succès.

---

3. L'exemple ici choisi concerne la culture du riz.

Si WHY et les méthodes socratiques invitent l'apprenant à s'engager dans un réel processus d'investigation, l'interaction reste entièrement fondée sur un dialogue entre un apprenant et une machine. Fournir assistance et guidage à un apprenant, celui-ci ayant d'autres tâches à exécuter que répondre à des questions, va amener à explorer d'autres pistes.

### **3.2.2. Guider l'apprenant dans des tâches complexes**

D'après Sleeman et Brown (1982, p. 1), l'intérêt des chercheurs s'est tourné vers la conception d'environnements facilitant un apprentissage par l'action, transformant la connaissance factuelle en connaissance pratique et expérimentale. L'ambition est d'arriver à combiner des expériences de résolution de problèmes et la motivation de l'apprentissage par la découverte avec le guidage efficace d'interactions tutorielles. Les connaissances ainsi acquises devraient être ancrées dans un contexte personnel signifiant.

#### **3.2.2.1. SOPHIE et les environnements d'apprentissage réactifs**

A la suite d'un système à initiative mixte dans le domaine de la météorologie (Brown *et al.*, 1973), Brown et Burton introduisent l'idée d'environnement d'apprentissage réactif (Brown *et al.*, 1982), dans le cadre du projet SOPHIE (*SOPHisticated Instructional Environment*). L'objectif est d'entraîner les apprenants au diagnostic de pannes dans le domaine des circuits électroniques, en travaillant à l'aide d'un laboratoire simulé sur ordinateur. Un défaut ayant été introduit dans un des composants, la tâche de l'élève est d'isoler ce composant défectueux à l'aide d'une série de mesures. L'environnement est réactif dans le sens où il ne fait que répondre aux actions et sollicitations de l'apprenant et ne prend jamais l'initiative. On suppose qu'avant d'utiliser le système, l'étudiant a une connaissance suffisante de l'électronique pour faire le travail qui lui est demandé et qu'il va apprendre en élaborant puis en testant des hypothèses relatives aux causes de dysfonctionnement du circuit électronique. Pour les auteurs, la technologie rend ainsi les expérimentations plus faciles (en fournissant des instruments adaptés) et plus sûres (dans un environnement simulé, il n'y a pas de catastrophe à redouter !) et permet aux étudiants d'apprendre à partir de leurs erreurs.

La recherche autour de SOPHIE s'est étalée sur plus de cinq ans et a conduit à l'élaboration de trois versions successives (SOPHIE I, SOPHIE II et SOPHIE III). Notons que, dès la première implantation, les auteurs ont intégré une interface en langue naturelle très efficace pour communiquer avec l'élève. Les raisons qui ont poussé les concepteurs à introduire des modifications sont intéressantes<sup>4</sup>.

---

4. Brown et al. 1982 ; Wenger, op.cit. p. 59-65 ; Barr et Feigenbaum, op.cit. p. 247-253



SOPHIE I combine un modèle mathématique du circuit et un module interactif permettant l'analyse et la critique du travail de l'étudiant. En fait, à partir des mesures déjà effectuées par l'apprenant, le système peut évaluer, d'un strict point de vue logique, les hypothèses qu'il formule ou la demande d'une nouvelle mesure. L'élève expérimente librement et choisit les mesures qu'il désire effectuer. Si le système peut lui fournir des retours sur la validité logique de ses propositions, il ne possède pas de connaissance directe sur le diagnostic de pannes et ne prend aucune initiative. Les concepteurs s'aperçoivent qu'il ne suffit pas de laisser l'élève expérimenter pour qu'il apprenne efficacement.

Ils décident de doter leur nouvelle version de capacités d'expertise accrues afin de fournir à l'étudiant une information précise sur les fautes qu'il a commises et d'en faire un environnement plus prescriptif en ajoutant un système d'EAO classique pour préparer l'apprenant au travail avec le laboratoire simulé, en planifiant des séquences d'activités. L'expert intégré dans le système est capable de résoudre les tâches de diagnostic données à l'apprenant, en rendant compte de son comportement. Ainsi, l'apprenant peut alternativement expérimenter par lui-même et observer le comportement de cet expert artificiel. Un jeu a aussi été développé dans lequel un joueur introduit un défaut qu'un autre joueur doit découvrir en effectuant des séries de mesures, le résultat de chacune de ces mesures devant être prévu par le premier. Le même jeu est aussi pratiqué par des équipes de deux personnes<sup>5</sup>.

Ainsi, dans le cadre d'une utilisation réelle en formation professionnelle, les auteurs n'ont pas hésité à combiner plusieurs approches : de l'EAO, de la découverte libre, des démonstrations, des jeux. Le passage de SOPHIE I à SOPHIE II illustre bien l'alternative dans la conception des systèmes : laisser le maximum de liberté d'exploration en minimisant les interventions de l'ordinateur ou limiter les possibilités d'exploration offertes de manière à permettre un guidage plus précis de l'apprentissage. Concilier ces deux exigences va être un des objectifs de SOPHIE III. En outre, les expérimentations mettent en évidence l'incapacité du système à exploiter les erreurs des apprenants. En effet, d'une part l'environnement est simplement réactif et ne peut prendre l'initiative d'explorer les incompréhensions ou suggérer d'autres approches et d'autre part, l'expertise implantée dans le système ne permet pas de rendre compte des raisonnements causaux utilisés par les experts humains.

---

5. L'intérêt du travail en équipe est fortement souligné par les auteurs (Brown *et al.*, op.cit., p. 237). D'une part, les interactions entre pairs sont stimulantes pour l'apprentissage et, d'autre part, des discussions échangées, le chercheur peut récupérer d'importantes informations stratégiques sur le pourquoi du choix ou du rejet de telle ou telle mesure, les sujets étaient beaucoup plus prolixes quand ils doivent convaincre un pair que lorsqu'ils travaillent isolément.

Trois importantes améliorations apparaissent nécessaires :

- construire un programme expert plus puissant et dont le fonctionnement se rapproche du raisonnement des experts humains,
- intégrer des techniques de modélisation de l'apprenant,
- développer des stratégies de guidage discret pour à la fois limiter la quantité des interruptions et décider des suggestions à apporter à l'apprenant.

Jugeant peu sage d'explorer ces trois pistes simultanément, les auteurs décident, dans le cadre du projet SOPHIE, de se concentrer sur la conception du module d'expertise simulant les méthodes utilisées par les experts humains. Pour explorer les problèmes de modélisation de l'élève et de guidage discret (*coaching*), ils choisissent de traiter des domaines considérés comme étant beaucoup plus simples, respectivement les mathématiques élémentaires (BUGGY) et les jeux (WEST).

Le développement de SOPHIE III conduit à des réflexions très semblables à celles des auteurs de WHY. L'expertise du système semble encore insuffisante pour satisfaire les demandes des apprenants vis-à-vis d'explications causales de nature qualitative. L'une des conclusions (Brown *et al.*, op.cit., p. 279) est que l'on ne sait pas véritablement ce que signifie le fait comprendre le fonctionnement d'un dispositif complexe. En particulier, on ne connaît pas les modèles mentaux des experts concernant le fonctionnement d'un système, ni comment ces modèles sont appris du fait qu'ils ne sont certainement pas explicitement enseignés. A partir de là se développe l'idée de *réseau causal*, traduisant l'ensemble des relations de causalité des différentes parties de la structure d'un système physique. Ce réseau est, d'après les auteurs, un modèle mental que l'on peut, métaphoriquement parlant, « faire tourner mentalement » (*runnable in the mind's eye*).

Concernant l'interaction didactique avec l'apprenant, le projet SOPHIE a oscillé entre des modes très libres et des modes très guidés. La recherche d'un compromis entre ces deux extrêmes conduit à élaborer des formes d'interventions que l'on peut qualifier de discrètes. C'est le projet WEST qui va servir de cadre à cette exploration.

### 3.2.2.2. WEST et l'interaction didactique discrète

Les jeux électroniques, apparus au début des années soixante-dix, semblent fournir un bon moyen pour étudier l'impact d'interventions didactiques ponctuelles dans un contexte informel, c'est-à-dire non dédié à l'apprentissage. En effet, ils rencontrent un écho important. Goldstein et Carr (1977) imaginent que certains peuvent avoir un attrait éducatif puissant, car ils sont de plus en plus répandus, stimulent l'activité et conduisent à mobiliser des connaissances mathématiques et scientifiques sérieuses dans un but précis ; ils sont en outre très motivants. Ils présentent toutefois une limite, un joueur laissé à lui-même peut s'avérer incapable d'acquérir les compétences nécessaires. L'intégration de certaines formes de guidage devrait aider le joueur à améliorer ses performances et par là même lui donner l'occasion d'apprendre.

Pour assurer ce guidage, Goldstein et Carr (ibid.) décrivent ce qu'ils nomment un « moniteur informatique discret » (*computer coach*<sup>6</sup>), programme qui, en quelque sorte, « regarde par-dessus l'épaule » du joueur ou de l'apprenant durant son activité et formule des critiques ou des suggestions pour améliorer sa performance. Ils présentent cette métaphore comme un « paradigme athlétique pour la formation intellectuelle ». Deux environnements vont s'inscrire dans cette perspective : WEST (Burton et Brown, 1982) et WUSOR (Goldstein et Carr, op.cit. ; Goldstein, 1982).

WEST est une sorte de jeu de l'oie inspiré de la conquête de l'Ouest (*How the West Was Won*). Le trajet comporte soixante-dix cases au total, certaines d'entre elles correspondant à des villes. Le but du jeu est d'atteindre le premier la dernière ville. A chaque tour, un joueur dispose de trois nombres tirés au hasard avec lesquels il doit construire une expression arithmétique utilisant les opérateurs d'addition, soustraction, multiplication et division, ces opérateurs et les nombres tirés ne pouvant être utilisés plus d'une fois. Le joueur déplace alors son pion d'un nombre de pas égal à la valeur numérique de cette expression. Dans le jeu, la valeur la plus grande n'est pas forcément la plus avantageuse. En effet, atteindre une ville fait passer directement à la ville suivante et atterrir sur une case occupée par un adversaire renvoie ce dernier deux villes en arrière. Les nombres tirés sont d'ailleurs petits pour que ces derniers déplacements soient souvent les plus intéressants.

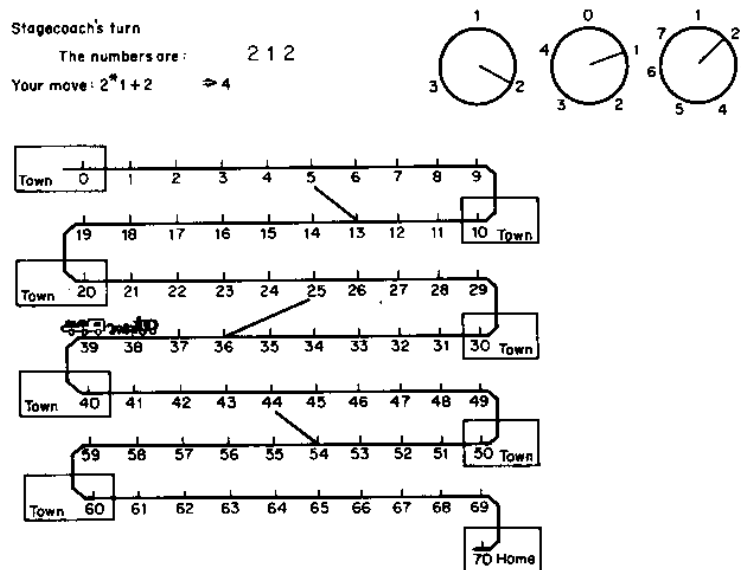
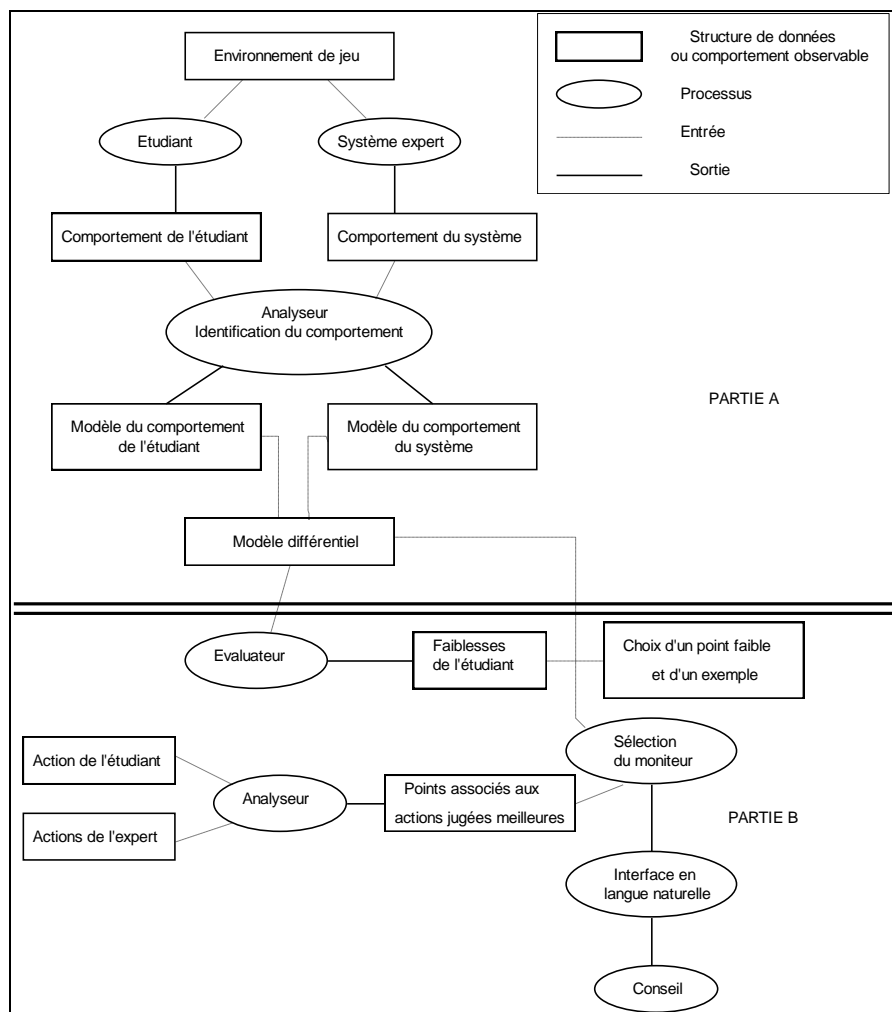


Figure 3.2. Exemple d'écran de WEST (Solomon, 1985).

6. Le terme anglais *coach* désigne à la fois un entraîneur sportif et quelqu'un qui donne des leçons particulières pour aider un élève à préparer un examen.

Pour guider le joueur, le « moniteur informatique » ne dispose que de peu d'informations. Il doit, à partir des choix successifs de déplacement, inférer quelles sont les compétences et concepts mis en œuvre ou manquants. Il se focalise sur certains aspects importants qu'il va pouvoir identifier dans les comportements observables. La figure 3.3 schématise le double processus de modélisation de l'apprenant et de fonctionnement du « moniteur informatique discret ».



**Figure 3.3.** Diagramme des processus de modélisation et d'intervention dans WEST (Burton et Brown, 1982)

La partie A présente la construction du modèle de comportement de l'apprenant. Ce modèle est un résumé des performances successives de l'apprenant au cours de la résolution d'une série de problèmes, qui sont ici des choix de déplacement dans le jeu. Chacun de ces choix est analysé de manière à en extraire les aspects importants et le même travail est effectué sur les choix de l'*expert* dans le même contexte. La comparaison des deux résultats fournit un modèle d'expertise partielle<sup>7</sup> qui permet d'identifier les faiblesses de l'étudiant.

La partie B schématise le fonctionnement du « moniteur informatique discret ». Quand le déplacement choisi par l'élève n'est pas celui de l'expert, l'« évaluateur » utilise le modèle différentiel pour créer une liste de ses points faibles. L'expert génère la liste de tous les déplacements jugés meilleurs que celui choisi par l'élève, chacun de ces déplacements étant traité par l'analyseur. A partir de ces deux listes (les points faibles de l'élève et les points correspondants aux déplacements jugés meilleurs), le « moniteur » détermine un point particulier illustré par un exemple de déplacement. En fonction de principes généraux de guidage, il décide ou non d'interrompre le déroulement de la partie. En cas d'interruption, le point choisi et l'exemple de déplacement correspondant sont fournis à un générateur d'explications.

Le travail de Brown et Burton est fondé sur une vision constructiviste de l'apprentissage. L'élève apprend en construisant de nouvelles connaissances à partir des connaissances dont il dispose. Les erreurs qu'il fait peuvent s'avérer constructives s'il arrive à en déterminer la cause et à opérer les ajustements nécessaires. La tâche principale du « moniteur informatique » est de fournir à l'apprenant les informations nécessaires pour l'aider à rendre ses erreurs constructives. Pour cela, il ne doit pas intervenir trop souvent afin de laisser à l'élève la possibilité de juger lui-même son comportement. Le problème du « moniteur » est donc de déterminer à la fois à quel moment interrompre l'activité de l'apprenant et que faire une fois qu'on l'a interrompue.

Sur la base de différentes expérimentations, Burton et Brown ont dégagé un ensemble de principes généraux susceptibles de guider les décisions d'un « moniteur », qu'ils ont intégré dans leur système. Cette liste de douze règles est présentée en figure 3.4.

Le jeu WUMPUS, sorte de version électronique moderne de *Thésée et le Minotaure* nécessitant des compétences de base en logique et en probabilité, a aussi servi de champ d'expérimentation pour la conception d'un « moniteur informatique » nommé WUSOR et même, tout comme SOPHIE, à trois versions successives. Goldstein et Carr (1977) décrivent son fonctionnement comme l'interaction entre trois programmes spécialisés, l'*Expert*, le *Psychologue* et le *Tuteur*.

---

7. La notion de modèle d'expertise partielle est détaillée à la section 3.3.1.1.

<p>Guider de manière discrète</p> <ol style="list-style-type: none"><li>1. Avant de donner un conseil sur une méthode, s'assurer que le joueur la maîtrise mal.</li><li>2. Pour illustrer une méthode, utiliser uniquement un exemple pour lequel le résultat est largement supérieur à celui du joueur.</li><li>3. Après lui avoir donné un conseil, permettre au joueur de l'appliquer immédiatement en l'autorisant à rejouer.</li><li>4. Si le joueur est en train de perdre, ne l'interrompre que si l'on peut lui éviter de perdre.</li></ol> <p>Maintenir l'intérêt du jeu</p> <ol style="list-style-type: none"><li>5. Ne pas intervenir deux fois de suite, quelle qu'en soit la raison.</li><li>6. Ne pas intervenir avant que le joueur n'ait eu l'occasion de découvrir le jeu par lui-même.</li><li>7. Ne pas donner uniquement des critiques. Féliciter l'élève pour ses choix judicieux.</li><li>8. Après avoir donné un conseil, lui permettre de rejouer mais ne pas l'imposer.</li></ol> <p>Multiplier les occasions d'apprentissage</p> <ol style="list-style-type: none"><li>9. Faire en sorte que l'ordinateur joue toujours de manière optimale.</li><li>10. Si l'élève demande de l'aide, lui fournir plusieurs niveaux de conseil.</li></ol> <p>Tenir compte de l'environnement ludique</p> <ol style="list-style-type: none"><li>11. Si le joueur perd largement, ajuster le niveau de jeu<sup>8</sup>.</li><li>12. Si l'étudiant a pu faire une erreur d'inattention, être magnanime mais afficher un message explicite pour le cas où cela n'aurait pas été de l'inattention.</li></ol>
--

**Figure 3.4.** Les 12 principes de guidage discret de WEST (Burton et Brown, 1982)

« L'Expert avertit le Tuteur quand le choix du joueur n'est pas optimal et précise quelles compétences sont nécessaires pour découvrir de meilleures possibilités. Ces compétences sont des sujets de dialogue possibles pour le Tuteur. Le Psychologue inspecte le comportement de l'apprenant et formule des hypothèses sur les compétences qu'il possède (enregistrées dans le *modèle des connaissances* de l'apprenant) et sur les modes d'enseignement susceptibles de lui permettre d'acquérir de nouvelles compétences (enregistrés dans le *modèle d'apprentissage*). Le Tuteur utilise ces modèles pour individualiser ses interventions avec le joueur. Le *modèle des connaissances* guide la sélection d'un sujet parmi ceux qui sont proposés par l'Expert tandis que le *modèle d'apprentissage* influence le choix d'une stratégie d'explication. Enfin, un Orateur convertit l'explication formelle générée par le Tuteur dans une forme appropriée à l'apprenant. » Doit s'ajouter un Auditeur,

---

8. Le principe 11 est contradictoire avec le principe 9. Si le joueur s'aperçoit que l'ordinateur ne fait pas les meilleurs choix, il risque de se sentir diminué et de perdre de l'intérêt au jeu. Mais s'il perd tout le temps, il peut se décourager. Pour l'éviter, le programme a la capacité de générer des mauvais tirages au joueur piloté par l'ordinateur.

module de compréhension du langage naturel dont le rôle est de traduire les questions du joueur.

L'Expert, le Psychologue et le Tuteur, auxquels il faut adjoindre l'Orateur et l'Auditeur, forment une sorte de communauté de spécialistes communiquant entre eux. Cette description du fonctionnement d'un « moniteur informatique » traduit bien la connexion étroite reliant l'expertise du domaine à enseigner, le modèle des connaissances de l'apprenant et les stratégies d'enseignement.

Les systèmes qui viennent d'être présentés illustrent quatre stratégies d'enseignement différentes, l'initiative mixte, le dialogue socratique, l'environnement réactif et le guidage discret. Perçues essentiellement comme des prototypes de recherche, mis à part SOPHIE, ces réalisations montrent la faisabilité et l'intérêt des approches incluant des techniques d'intelligence artificielle. Intégrer de l'expertise dans des machines permet effectivement de développer de nouvelles formes d'interaction avec les apprenants. En revanche, la difficulté de faire acquérir de nouvelles connaissances à des apprenants à l'aide d'une machine, en est d'autant plus mise en évidence. Peu à peu, se dessine la nécessité de passer de micro-décisions didactiques, comment réagir localement à telle réponse ou telle action de l'élève, à de véritables stratégies d'enseignement prenant en charge globalement le dialogue avec l'apprenant. En l'absence de modèle à leur disposition, les chercheurs se tournent vers les enseignants et analysent leur pratique pour alimenter les programmes qu'ils développent. Ensuite, ces programmes servent pour expérimenter et améliorer les stratégies mises en place.

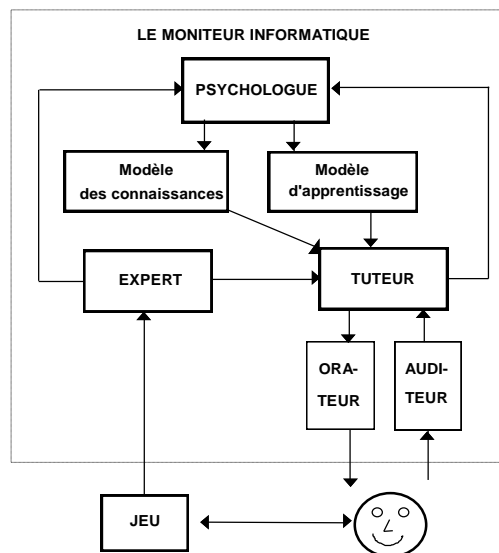


Figure 3.5. Schéma général d'un moniteur d'après Goldstein et Carr (1977)

Les chercheurs, se réclamant d'une approche constructiviste, issue des travaux de Piaget, commencent à accorder une importance primordiale à l'activité de l'élève et tentent d'installer des modes d'interaction et de guidage susceptibles de rendre cette activité productive, c'est-à-dire favorisant l'acquisition de nouvelles connaissances. Arriver à construire une représentation suffisamment fidèle des connaissances de l'élève devient alors primordial, le modèle de l'apprenant semblant être au cœur de l'interaction didactique. Les travaux autour de WHY et SOPHIE conduisent même à des conclusions similaires allant encore plus loin. Dans des domaines nécessitant des raisonnements complexes, le but de l'enseignement ne peut se traduire par l'acquisition de quelques comportements préalablement spécifiés, mais par le fait d'installer un certain *modèle* dans la tête de l'apprenant. Pour cela, il faut sans doute trouver des moyens pour tenter de savoir ce qui est dans la tête de l'apprenant et donc le modéliser.

### 3.3. Modéliser l'élève

*« Les esprits sont des systèmes complexes et embrouillés qui évoluent suivant des processus de développement élaborés. En décrire un, même à un seul instant de son histoire, doit être très difficile. De prime abord, on peut supposer que c'est encore plus ardu de décrire l'histoire complète de son développement. Ne devrions-nous pas nous contenter d'essayer de décrire uniquement ses " performances finales " ? Nous pensons exactement le contraire. Uniquement une bonne théorie des principes du développement de l'esprit peut conduire à une théorie maniable sur la façon dont il fonctionne en définitive. »*

(Minsky cité par Matz, 1982, p. 48).

Comme nous venons de le voir, la mise en place d'un modèle de connaissances de l'apprenant devient indispensable pour réaliser des tuteurs artificiels suffisamment adaptatifs. Une telle préoccupation n'intervenait pas dans les productions classiques de l'EAO, basées sur le cycle présentation-question-analyse de la réponse-branchement, se satisfaisant de tactiques de parcours prédéterminées associées aux réponses correctes ou erronées de l'élève. Pourtant des précurseurs avaient abordé cette question. Les systèmes adaptatifs de Pask intégraient l'historique des réponses de l'élève, non plus seulement la réponse courante, pour contrôler le parcours. Des modèles stochastiques avaient été développés, mais si ces modèles permettaient d'obtenir de bonnes prédictions sur les performances des apprenants relatives à certaines tâches, ils ne fournissaient pas de modèle de l'état de leurs connaissances.

Le besoin de disposer d'une représentation des connaissances d'un apprenant s'est vite imposé pour la réalisation de tuteurs capables d'adapter leurs stratégies d'enseignement (Hartley et Sleeman, 1973 ; Self, 1974). Cette nécessité a conduit les chercheurs à tenter d'induire un modèle de l'apprenant construit dynamiquement en



s'appuyant sur son comportement observable. Pour ce faire, différents problèmes sont à résoudre : tout d'abord, celui du choix d'une représentation pour enregistrer les données sur l'apprenant, constituant son modèle de connaissances ; ensuite, la mise en place de mécanismes pour initialiser puis mettre à jour ce modèle tout au long de l'interaction avec le système et, enfin, l'implantation d'un processus d'interprétation de ce modèle afin d'orienter les procédures de décision liées aux stratégies d'enseignement. Ces trois problèmes sont très fortement reliés et la solution adoptée pour l'un conditionne de manière très importante la façon de traiter les deux autres.

Nous allons passer en revue les différentes solutions élaborées pour modéliser l'apprenant en détaillant, dans un premier temps, les types de modèles utilisés puis les techniques de diagnostic associées.

### 3.3.1. Les différents types de modèles

Un modèle de l'apprenant est une structure de données, au sens informatique, qui caractérise, pour le système d'enseignement, l'état d'un sous-ensemble des connaissances de cet apprenant<sup>9</sup>. Il va se définir par l'écart entre les propres connaissances (supposées) de l'apprenant et les connaissances cibles, enjeu de l'apprentissage, telles qu'elles sont représentées dans le système. La façon de concevoir cet écart conduit à distinguer deux grandes classes de modèles :

- les *modèles d'expertise partielle* ou de superposition (*overlay*, Carr et Goldstein, 1977) dans lesquels la connaissance de l'élève n'est qu'un sous-ensemble de la connaissance cible,
- les *modèles différentiels*<sup>10</sup> qui incorporent des « connaissances fausses », correspondant à des perturbations des connaissances expertes ou des préconceptions erronées.

#### 3.3.1.1. Les modèles d'expertise partielle

Les modèles d'expertise partielle consistent en une structure de données comprenant l'ensemble des connaissances et compétences élémentaires à acquérir et précisant pour chacune d'elles le degré de maîtrise de l'apprenant. L'idée sous-jacente à ce type de modèle est que l'élève présente des lacunes ou des

9. Il ne s'agit ici que de connaissances. Les modèles dont on discute n'intègrent pas d'autres caractéristiques de l'apprenant comme son style cognitif ou des variables de type affectif.

10. La terminologie n'est pas toujours très claire. On parle parfois de modèle différentiel pour WEST, puisqu'il est constitué par l'analyse des différences entre les actions choisies par l'élève et celles de l'expert. C'est pourtant un modèle de type expertise partielle, n'incluant pas de déviation par rapport à la connaissance correcte. Nous employons ici le terme différentiel (en reprenant la terminologie choisie par Nicaud et Vivet, 1988) pour désigner un modèle incluant des règles ou connaissances erronées.

connaissances encore mal assurées, en quelque sorte des faiblesses, qu'il s'agit d'identifier pour lui permettre de progresser. L'objectif du système tuteur est alors de compléter les connaissances de l'élève pour qu'il acquière l'ensemble des connaissances précisées dans le modèle.

Des modèles d'expertise partielle ont été développés pour réaliser les “ moniteurs informatiques discrets ” WEST et WUSOR, que nous avons présentés dans la section précédente. La figure 3.6, extraite de WEST <sup>11</sup>, présente un extrait du modèle élève, c'est-à-dire une liste d'expressions algébriques et le jugement lors de leur utilisation ou non-utilisation par l'apprenant dans les situations de jeu. Par exemple, la troisième colonne montre qu'une expression de la forme  $(A + B) - C$  a été utilisée pour effectuer trois déplacements. Un a été jugé comme étant le meilleur, un autre correct et le dernier pauvre. Une fois, une expression de cette forme aurait été utilisée par l'expert et ne l'a pas été par l'apprenant.

Le modèle complet de l'apprenant, dans WEST, est divisé en trois parties : les compétences mathématiques (forme des expressions, usage des parenthèses), les compétences liées directement au jeu (usage des raccourcis, des villes ou l'arrivée sur la case de l'adversaire) et des compétences stratégiques générales (par exemple, observer les coups de son adversaire pour les utiliser à son tour). Une fois que le modèle est suffisamment rempli, c'est-à-dire après un certain nombre de coups, son analyse permet de mettre en évidence certaines caractéristiques du joueur.

	$(A+B) \times C$	$(A+B)-C$	$A/(B \times C)$	$(A \times B)-C$	$(A \times B)+C$	$(A \times B)/C$	$A \times (B-C)$
Déplacements	1	3					
Meilleur		1					
Correct		1					1
Pauvre		1		1			
Manqué	2	1	1	1	1	1	

**Figure 3.6.** Un extrait du modèle d'expertise partielle de WEST (Burton et Brown, 1982)

Pour mettre à jour le modèle d'expertise partielle, le système doit détecter les faiblesses de l'apprenant en analysant les actions qu'il effectue. Les compétences en jeu ne sont pas toujours directement observables et la mise à jour du modèle nécessite un mécanisme spécifique pour les relier au choix de l'apprenant. En fait, on compare ce choix avec le ou les choix censés être effectués par l'expert dans la même situation afin d'évaluer sa pertinence. Une difficulté est qu'un mauvais choix peut impliquer plusieurs compétences élémentaires et qu'il n'est pas facile de déterminer quelle(s) compétence(s) l'apprenant ne maîtrise pas. Le problème est

11. L'architecture de WEST est schématisée en figure 3.3.

analogue, en cas de choix correct, lorsque plusieurs méthodes permettent d'aboutir au résultat<sup>12</sup>.

On peut souligner plusieurs faiblesses des modèles d'expertise partielle. Tout d'abord, ils sont basés sur une hypothèse réductrice forte consistant à supposer qu'il suffit de situer les connaissances de l'apprenant vis-à-vis de celles de l'expert en termes de maîtrise plus ou moins grande. Même si plusieurs niveaux peuvent être distingués<sup>13</sup>, la connaissance est distribuée en connaissances ou compétences élémentaires censées être largement indépendantes les unes des autres. En outre, ces modèles ne donnent aucune indication permettant d'expliquer pourquoi un apprenant n'a pas effectué le meilleur choix possible. L'une des conclusions de Goldstein (1982), dans le cadre de son travail sur WUSOR, est de constater que l'approche expertise partielle amène une simplification trop grande. En particulier, modéliser la connaissance de l'apprenant comme un sous-ensemble de la connaissance de l'expert ne permet pas de rendre compte de la façon dont la connaissance évolue par des processus comme l'analogie ou la généralisation... Elle développe l'idée de *graphe génétique* pour représenter la connaissance procédurale du point de vue de son évolution. Un tel graphe est composé de nœuds correspondant aux règles traduisant les procédures et de liens précisant les relations d'évolution entre ces règles. Pour Goldstein, le « moniteur », au lieu de s'appuyer sur l'ensemble de compétences non ordonnées du modèle d'expertise partielle, peut travailler sur un graphe de compétences liées par des relations d'évolution. Si la difficulté est bien de déterminer cette genèse des connaissances, la critique de Goldstein souligne la nécessité de passer d'un paradigme basé sur l'expert (qui est celui du modèle d'expertise partielle) à un paradigme basé sur l'apprenant.

Malgré leurs limitations, les modèles d'expertise partielle ont des qualités très appréciables. D'une part, ils se déduisent relativement aisément de l'analyse d'un domaine, de sa décomposition en connaissances et compétences élémentaires. En ce sens, ils prolongent les modèles de performance issus de l'enseignement programmé. D'autre part, ce sont des modèles robustes, qui fonctionnent *en première approximation* dans des domaines où les stratégies d'enseignement peuvent être fondées essentiellement sur un découpage en connaissances élémentaires, sans nécessiter d'analyse plus profonde des conceptions des apprenants. Ils vont ainsi servir de base à la réalisation de nombreux systèmes.

---

12. La solution standard consiste à respectivement créditer ou « blâmer » toutes les compétences en cas de succès ou d'échec (ce qui revient souvent à incrémenter ou décrémenter un compteur associé à chacune de ces compétences). On espère qu'il y aura suffisamment de situations discriminantes pour que les forces et faiblesses apparaissent.

13. Le passage de WUSOR I à WUSOR II a consisté notamment à diviser l'ensemble de règles à acquérir en cinq niveaux de difficulté croissante.

### 3.3.1.2. *Les modèles différentiels*

Les travaux précédents modélisent la connaissance de l'élève comme un sous-ensemble de la connaissance correcte. Ils ne permettent pas de prendre en compte des méthodes ou résultats incorrects que l'élève peut avoir acquis. Or, des études vont montrer que de nombreuses erreurs ne sont pas dues à un comportement erratique des apprenants, mais à l'application correcte de procédures fausses. Pour élaborer un modèle des connaissances des apprenants, il faut prendre en compte ces erreurs de type systématique, que les chercheurs vont désigner par le terme « bogue »<sup>14</sup> (*bug*).

L'exemple le plus caractéristique est l'étude des erreurs faites par les élèves dans la soustraction, avec BUGGY (Brown et Burton, 1978) et l'ensemble des travaux qui en sont issus. On trouvera dans Wenger (op.cit., chapitre 8)<sup>15</sup> l'histoire des étapes de la « réparation des bogues ». Nous allons nous contenter d'en retracer les grandes lignes.

Sur la base de l'analyse d'opérations élémentaires produites par des élèves, Brown et Burton vont caractériser le processus de résolution des élèves par un réseau procédural. L'ensemble des actions à effectuer pour faire une soustraction (en colonnes) est décomposé en une hiérarchie d'étapes correspondant à des compétences élémentaires, ces dernières étant éventuellement erronées. Pour chaque élève, le réseau procédural ainsi constitué est censé être une réplique fidèle de l'état des connaissances de cet élève, incluant ses connaissances fausses. En supposant l'élève consistant, c'est-à-dire, appliquant sans perturbation les connaissances qu'on lui attribue, le réseau est exécutable dans le sens où, pour un problème donné, il fournit la réponse susceptible d'être donnée par cet élève. Le réseau modélise l'ensemble du comportement de l'élève, supposé être stable. S'il est prédictif, il ne se contente pas d'estimer la probabilité d'une réponse correcte ou incorrecte, il fournit la réponse même de l'élève, la validité de cette réponse ne dépendant que des caractéristiques du problème posé à l'élève. Notons que les travaux de Young et O'Shea (1981) consacrés à la soustraction rejoignent les précédents en se focalisant toutefois sur les erreurs les plus fondamentales.

Il faut remarquer que la technique de soustraction enseignée aux Etats-Unis n'est pas celle utilisée en France. Les deux méthodes sont illustrées par un exemple (figure 3.7). La première, que l'on qualifie de méthode naturelle en France et qui sert parfois d'introduction à la technique de la soustraction, s'appuie directement sur les

---

14. Le terme de bogue désigne en informatique une erreur dans un programme. Il s'agit en général plutôt d'une erreur de codage que d'une erreur d'analyse importante. Dans le cadre de la modélisation des apprenants, un bogue va désigner des erreurs procédurales de nature locale plutôt que des incompréhensions profondes.

15. Nous renvoyons le lecteur intéressé à cet ouvrage ou à la traduction française de Cuppens (1988).

groupements de la numération de position. Elle ne modifie que le premier membre de la soustraction par l'intermédiaire d'un système d'emprunts (1 dizaine donne 10 unités, etc.). Cette méthode s'appuie directement sur les caractéristiques de l'écriture des nombres en numération décimale.

$$\begin{array}{r}
 1 \ 13 \\
 \cancel{2} \ \cancel{4} \ 15 \\
 - \ 8 \ 9 \\
 \hline
 1 \ 5 \ 6
 \end{array}
 \qquad
 \begin{array}{r}
 2 \ 4 \ 5 \\
 - \ 1 \ 8 \ 9 \\
 \hline
 1 \ 5 \ 6
 \end{array}$$

**Figure 3.7.** Deux méthodes de soustraction

La seconde méthode s'appuie sur l'idée d'invariance d'une distance et compense en ajoutant une dizaine en bas quand on ajoute dix unités au nombre du haut. Elle est sans doute plus difficile à justifier pour de jeunes apprenants mais certainement plus économique au niveau des écritures. En particulier, la méthode naturelle est plus délicate quand il faut emprunter sur un 0. Les erreurs faites par les élèves dépendent directement de la méthode de soustraction pratiquée.

Le repérage des bogues dans la soustraction est complété par d'autres travaux (Burton, 1982). Tout d'abord, un jeu permettant de s'entraîner à la détection des bogues est conçu. Comme il apparaît difficile de décrire verbalement les procédures erronées, diagnostiquer les erreurs consiste à faire la même erreur sur un certain nombre de cas. La réponse est jugée correcte si le joueur trouve les résultats de l'élève (que ces résultats soient en fait corrects ou non). Deux systèmes de diagnostic automatique sont ensuite construits : DEBUGGY qui travaille en différé à partir de problèmes résolus par un élève et IDEBUGGY qui est une version interactive. La mise au point de ces programmes, qui ont apparemment donné des résultats satisfaisants, conduit à des traitements délicats des phénomènes de bruit, écarts entre un comportement parfaitement consistant et un comportement réel qui ajoutent des erreurs non prévues par la théorie.

Le diagnostic s'effectue à partir de la décomposition de la technique générale et de l'ensemble des bogues préalablement repérés en un certain nombre d'éléments insécables, des sortes de sous-compétences primitives. Est considérée comme sous-compétence primitive, toute partie isolable de la compétence qu'il est possible de méconnaître. A chacune d'elles, on associe des problèmes dans lesquels elle intervient, ce qui permet de repérer les éventuels bogues correspondants. On en déduit un nombre fini de problèmes pour discriminer les bogues. D'après Burton (op.cit.), une série de 12 problèmes peut permettre d'opérer une sélection parmi environ 1 200 bogues composés (et 20 problèmes pour s'assurer d'une consistance minimale des bogues repérés). Toutefois, des obstacles apparaissent, tels un certain manque de stabilité dans le comportement des apprenants et des phénomènes de

migration des bogues, c'est-à-dire des déplacements dans les manifestations des erreurs.

La théorie REPAIR de Brown et Van Lehn (1980) va tenter de rendre compte de l'apparition des bogues constatés chez les élèves. Ils la décrivent pour l'essentiel comme des *réparations* locales effectuées face à des *impasses* dans le déroulement d'une technique. Par exemple, dans la soustraction ci-dessous, un élève ne sait pas quoi faire quand il doit enlever une unité à un 0. Face à cette impasse, il imagine une solution. Considérer que 0 est une sorte de 10 et donc de mettre 9 sans décrémenter le chiffre de la colonne de gauche. Il peut aussi décider de laisser le 0 tel quel.

$$\begin{array}{r} 9 \\ 20,5 \\ - 138 \\ \hline 167 \end{array}$$

Van Lehn essaye de mettre au point sa théorie en tentant de justifier par des arguments cognitifs les choix de représentation. En effet, la difficulté est d'arriver à engendrer ces différentes impasses réparations de telle sorte qu'elles couvrent l'ensemble des bogues constatés en évitant de générer des bogues non observés et peu plausibles sur le plan psychologique. Comme REPAIR fonctionne par perturbation de la connaissance correcte, il importe de déterminer des processus de contrôle de cette perturbation pour garantir un minimum de plausibilité.

Par la suite, Van Lehn développe une théorie intitulée STEP dans le but de spécifier un modèle d'enseignement évitant l'émergence des bogues. Il s'intéresse à l'apprentissage inductif à partir d'exemples et cherche à décrire ce qu'il nomme des *conditions de félicité* pour garantir un apprentissage sans erreur (partir d'un ensemble minimal d'exemples et ajouter les difficultés une à une). Toutefois, le modèle proposé, fondé sur un apprentissage par induction pure, apparaît peu réaliste. Il va être complété dans sierra, qui combine STEP et REPAIR, tentative de modélisation de l'apprentissage à partir d'une suite de leçons. Ce modèle génératif, reconstituant un processus d'apparition des bogues, fournit une meilleure couverture que les théories précédentes. Nous aurons l'occasion de revenir un peu plus loin sur les travaux de Van Lehn.

Pour illustrer les modèles différentiels, nous allons nous appuyer sur le système de modélisation algébrique de Sleeman, qui concerne le domaine de la résolution des équations du premier degré. Ce système, initialement développé à Leeds, s'est d'abord appelé LMS (*Leeds Modelling System*, Sleeman 1982, 1983) puis PIXIE (Sleeman, 1987) quand Sleeman est allé à Stanford.

La figure 3.8 donne des exemples de règles correctes et de règles erronées (dont le nom commence par M, pour *mal-rule* en anglais) pour résoudre des équations du

premier degré. A chaque règle sont associés un nom, un niveau, la description des conditions dans lesquelles elle est susceptible de s'appliquer et enfin son action sur l'expression ou la partie de l'expression algébrique correspondante. A partir des actions effectuées par un apprenant, LMS infère un modèle basé sur ces différentes règles. Le système tente d'aboutir à la solution proposée par l'apprenant en activant certaines règles correctes et d'autres erronées. Lorsque le système obtient une solution semblable à celle de l'élève, il peut émettre une hypothèse quant à l'erreur effectuée. Connaissant exhaustivement les règles correctes ou fausses pouvant être employées, le problème de diagnostic consiste à trouver un chemin dans un espace de recherche.

Règle	Niv.	Ensemble de conditions	Action
FIN2	1	$X = M/N$	(M/N) ou évaluation
SOLVE	2	$M * X = N$	$X=N/M$ ou $\infty$
ADDSUB	3	## $M \pm N$ ##	## (évaluation) ##
MULT	4	## $M * N$ ##	## (évaluation) ##
XADDSUB	5	## $M * X \pm N * X$ ##	## $(M \pm N) * X$ ##
NTORHS	6	## $\pm M = ##$	## = ## - $\pm M$
REARRANGE	7	## $\pm M \pm N * X$ ##	## $\pm N * X \pm M$ ##
XTOLHS	8	## = $\pm M * X$ ##	## - $\pm M * X = ##$
BRA1	9	## (N) ##	## N ##
BRA2	10	## $M * (N * X \pm P)$ ##	## $M * N * X \pm M * P$ ##
MSOLVE	2	$M * X = N$	$X=M/N$ ou $\infty$
MNTORHS	6	## $\pm M = ##$	## = ## $\pm M$
M1BRA2	10	## $M * (N * X \pm P)$ ##	## $(M * N * X \pm P)$ ##
M2BRA2	10	## $M * (N * X \pm P)$ ##	## $(M * N * X \pm M \pm P)$ ##

Avec M, N et P entiers. ## signifie une suite quelconque

Figure 3.8. Règles et fausses règles dans le système LMS de Sleeman (1983)

	$2X + 5 = 9$		$2X + 5 = 9$
NTORHS	$2X = 9 - 5$	MNTORHS	$2X = 9 + 5$
ADDSUB	$2X = 4$	ADDSUB	$2X = 14$
SOLVE	$X = 4/2$	SOLVE	$X = 14/2$
FIN2	2	FIN2	7

Figure 3.9. Résolution correcte et résolution erronée de  $2X + 5 = 9$

Par exemple, la figure 3.9 montre une résolution correcte et une résolution fautive de l'équation  $2X + 5 = 9$ . L'élève développe sa solution, étape par étape, en entrant des expressions successives interprétées par LMS ligne à ligne, qui n'intervient qu'en cas d'erreur syntaxique sur une expression. Notons que l'élève ne choisit pas des

opérateurs permettant de transformer l'expression courante, mais entre des expressions complètes. Les règles correctes et les règles fausses sont internes au système. Lorsque l'exercice est achevé, un *modeleur* entre en action afin de déterminer les règles pouvant fournir les mêmes expressions successives.

Alors qu'un modèle d'expertise partielle invite à des stratégies d'enseignement centrées sur le fait de combler les lacunes de l'apprenant, les modèles différentiels vont induire des stratégies basées sur la remédiation. Pourtant, si une description des bogues est certainement importante pour caractériser les apprenants, il n'est pas évident qu'elle soit suffisante pour déterminer les types d'intervention à effectuer. Etiqueter un comportement ne donne pas *ipso facto* les informations nécessaires pour choisir une thérapeutique adaptée. Cette difficulté va amener des réflexions intéressantes sur le niveau d'intervention associé à une erreur. En fait, les deux approches de modélisation que nous venons de voir, expertise partielle et modèle différentiel, s'appuient essentiellement sur la vision du système enseignant et étendent les travaux de l'enseignement programmé : rechercher faiblesses, lacunes, erreurs et déterminer des actions en retour. La différence avec l'enseignement programmé et l'EAO vient du fait qu'il n'y a pas toujours de branchement explicite et que tout n'a pas besoin d'être entièrement prévu à l'avance, les détections de lacunes ou d'erreurs comme les décisions d'actions peuvent être effectuées dynamiquement. Les théories de l'enseignement et de l'apprentissage font cependant encore défaut et les chercheurs vont peu à peu s'attacher à les définir. Néanmoins, les modèles de l'élève des types précédents vont être intégrés dans différents systèmes.

### 3.3.2.3. La classification de Van Lehn

La classification des différents modèles de l'élève proposée par Van Lehn (1988), bien qu'anticipant légèrement sur des systèmes que nous présenterons un peu plus loin, permet de mettre en perspective les différents choix des concepteurs et d'y associer différentes méthodes de diagnostic. Van Lehn prend en compte trois paramètres.

1. Les *entrées*, c'est-à-dire les éléments observables utilisables par le module diagnostic. Le classement s'opère suivant la quantité et la qualité des éléments disponibles. Il peut s'agir uniquement d'une réponse ou d'un résultat, comme dans le cas de BUGGY ou LMS, d'un ensemble d'étapes intermédiaires ou de réponses successives comme dans WEST ou WHY (accumulation d'un certain nombre de réponses<sup>16</sup>) ou de l'ensemble des pas élémentaires du raisonnement ou de la technique mise en œuvre. Dans ce dernier cas, certains chercheurs ont pu, bien imprudemment, affirmer que les réponses ou entrées de l'élève dans le système pouvaient étroitement correspondre aux états mentaux de l'apprenant. Van Lehn parle ainsi de la possibilité de s'approcher des états mentaux à l'aide d'informations complémentaires obtenues de manière indirecte (ensemble de questions, analyse de

---

16. LMS pourrait aussi être rangé dans cette catégorie, puisqu'il peut disposer de plusieurs pas successifs de résolution.



protocoles verbaux). Disons que le diagnostic s'appuie sur un maximum d'éléments possibles.

2. Les *connaissances utilisées par le module expert*. Elles sont représentées essentiellement sous une forme déclarative, comme le réseau sémantique de SCHOLAR, ou sous une forme procédurale, c'est-à-dire préstructurée en fonction du but à atteindre. Dans le cas d'une représentation procédurale, il est possible de préciser une hiérarchie de sous-buts à résoudre, associés aux différentes actions, ce qui amène à distinguer représentation procédurale plate (sans sous-buts) ou hiérarchique.

3. Les *différences élève-expert*, c'est-à-dire le type de modèle choisi : expertise partielle ou modèle différentiel. Dans ce dernier cas, on peut disposer soit d'une énumération complète des erreurs dans un domaine donné, c'est le cas de BUGGY, soit d'une possibilité de les reconstituer dynamiquement à partir d'un assemblage de bogues partiels comme dans LMS.

La figure 3.10 présente le tableau à trois dimensions proposé par VanLehn pour classer les différents modèles de l'apprenant. En italique sont indiquées les techniques de diagnostic correspondant à ces différents cas.

Entrées	Mode de Représentation des connaissances		
	Procédurale plate	Procédurale hiérarchique	Déclarative
Etats mentaux		**Kimball CALCULUS ** LISP TUTOR ** GEOMETRY TUTOR ( <i>Traçage de modèle</i> )	GUIDON
Etapes intermédiaires	WEST WUSOR ( <i>Traçage de compétences élémentaires</i> )	** MACSYMA Advisor ** SPADE ( <i>Reconnaissance de plan</i> )	* SCHOLAR * WHY ( <i>Réseau sémantique</i> )
Résultat final	** LMS, PIXIE ** ACM ( <i>Recherche de chemin</i> )	* BUGGY * DEBUGGY * IDEBUGGY ( <i>Arbre de décision / Génération / test</i> )	* MENO * PROUST ( <i>Génération / test</i> )

\*\* Librairie de bogues partiels

\* Librairie de bogues

Expertise partielle

**Figure 3.10.** Classification de Van Lehn (1988)

REMARQUE : LISP TUTOR et GEOMETRY TUTOR seront présentés au chapitre 5. GUIDON, MACSYMA Advisor et ACM seront présentés dans ce chapitre. SPADE, MENO et PROUST sont des systèmes qui concernent l'apprentissage de langages de programmation. Tous ces systèmes sont décrits dans Wenger (op.cit.).

Si les chercheurs s'accordent sur la nécessité d'introduire un modèle de l'apprenant, il semble que les différents exemples répertoriés dans la classification de Van Lehn ne leur donnent pas entièrement satisfaction. Une première limitation vient de leur faible pouvoir explicatif. Les modèles fonctionnent au niveau des comportements observables et tentent de caractériser ces comportements. Ainsi, comme le remarquent O'Shea et Self (1983, p. 152), les modélisations de l'élève tendent à opérer à un mauvais niveau : elles fournissent des informations sur les tentatives d'un élève pour résoudre un problème spécifique, mais pas directement sur sa compréhension des aptitudes générales mises en jeu. Il manque la représentation mentale que se fait l'apprenant du problème. Les modèles précédents supposent en fait que la conceptualisation de l'élève reste proche de celle de l'expert. Au vu des conclusions des projets WHY et SOPHIE sur la nécessité de prendre en compte les modèles mentaux des apprenants, Van Lehn propose ainsi d'étendre sa classification en intégrant des modèles de ce type. Cet objectif va être celui de nombreuses recherches dans le domaine des tuteurs intelligents.

### 3.3.2. Les techniques de modélisation

La classification de Van Lehn que nous venons de présenter nous amène à préciser les techniques à développer pour gérer les modèles d'élève, c'est-à-dire pour modéliser un apprenant. Toutefois, cette modélisation est loin de se limiter à un problème technique ; elle s'appuie sur l'articulation de théories de différentes natures, sous-jacentes à la réalisation de ces modèles, des théories descriptives, génératives, explicatives et prescriptives.

Les théories basées sur la recherche des bogues (comme BUGGY) sont de type *descriptif*, un bogue n'étant qu'une notation particulière d'une procédure incorrecte. Elles permettent essentiellement de décrire le comportement d'un apprenant dans la réalisation de certaines tâches. Elles tentent de reconstruire son cheminement, c'est-à-dire de décrire *comment* fait l'apprenant. Les *théories génératives* (comme REPAIR) essaient de reconstituer la génération même des erreurs dans le processus d'apprentissage. Elles fournissent une sorte de descriptif de l'historique de l'apparition des erreurs. Les *théories explicatives* s'intéressent aux causes des erreurs en prenant en compte l'organisation des connaissances acquises par l'apprenant. Elles sont fondées sur des théories psychologiques et peuvent faire appel à bien d'autres facteurs que les seuls facteurs cognitifs. Elles cherchent à décrire *pourquoi* l'apprenant fait telle ou telle erreur. Les *théories prescriptives* (comme STEP) s'intéressent d'un côté à caractériser l'enseignement pour éviter ou minimiser

l'apparition de bogues, et de l'autre à préciser les modes d'intervention à effectuer en cas d'erreur détectée.

En reprenant une métaphore médicale classique, dans laquelle les erreurs sont comparées aux maladies, les théories descriptives s'attachent à décrire les symptômes, les théories génératives l'apparition et l'évolution des maladies, les théories explicatives recherchent les causes des maladies, enfin les théories prescriptives concernent l'aspect thérapeutique, qu'il soit préventif ou qu'il intervienne après la détection de la maladie. Le diagnostic, pour le médecin, consiste à rechercher une maladie à partir des signes observables, les symptômes. Pour un système enseignant, il s'agit de trouver une sorte d'état cognitif caché, la connaissance de l'apprenant dans le domaine d'apprentissage, à partir de ses comportements observables.

La distinction opérée par Landa (1972) entre diagnose et diagnostic, même si elle n'apparaît pas très clairement dans les travaux des chercheurs, permet de clarifier certaines difficultés de la modélisation. Pour diagnostiquer, il faut connaître les maladies et les symptômes qui leur sont associés. Le traitement ne vise pas uniquement à faire disparaître les symptômes, mais à guérir la maladie et à en faire disparaître les causes. Or, sans une bonne connaissance des maladies, on risque de ne traiter que les symptômes. C'est une critique importante qui va souvent être adressée aux travaux dans le domaine de la modélisation des apprenants.

### 3.3.2.1. La diagnose

La diagnose est la connaissance d'une maladie, qui s'acquiert par l'observation des signes diagnostiques, des symptômes. L'analogue, dans le domaine de l'apprentissage, concerne l'ensemble des connaissances erronées acquises par les apprenants. Cet ensemble peut être trouvé à partir d'une étude disponible dans le domaine, par l'analyse du comportement des élèves (avec divers protocoles) ou grâce à un *modèle prédictif* s'appuyant sur une théorie de l'apprentissage dans le domaine concerné.

L'analyse préalable des erreurs est un processus coûteux (Hennessy *et al.*, 1989) qui n'est jamais totalement fini, une nouvelle erreur pouvant toujours intervenir. Si les premiers systèmes (BUGGY, LMS) ont été conçus à partir de l'analyse d'un grand nombre de protocoles, les chercheurs vont essayer de se dispenser de cette nécessité. Des théories génératives ou reconstructives vont tenter de mettre en évidence des méthodes généralisables d'apparition des erreurs. Un certain nombre de *modèles par perturbation* sont ainsi proposés. Concernant la soustraction, Young et O'Shea (1981) traduisent les erreurs par des manipulations syntaxiques de règles correctes : modification, ajout ou omission d'une condition ou d'une conclusion d'une des règles ; confusion entre opérateurs primitifs du domaine ( $\times$  et  $+$ ) ; généralisation du contexte d'application, analogies incorrectes, construites à partir de plans stéréotypés plus simples ou de problèmes résolus antérieurement.

Pour Matz (1982), dans le domaine de l'algèbre, les erreurs les plus communes proviennent de l'extrapolation de techniques dans des contextes où les règles apprises ne s'appliquent plus. Elle suppose que l'apprenant cherche à réduire la distance entre ces règles et les nouveaux problèmes qui lui sont soumis, en se ramenant à un problème déjà vu ou en modifiant une règle connue pour qu'elle s'adapte à la nouvelle situation.

Ainsi, de grandes catégories d'erreur correspondent :

- à un usage abusif des propriétés de linéarité :

$$(a + b)^2 = a^2 + b^2 \text{ ou } \frac{1}{3} = \frac{1}{x} + \frac{1}{7} \text{ donne } 3 = x + 7$$

- ou à des généralisations basées sur des ressemblances de surface :

$$(x - 3)(x - 4) = 0 \text{ se résout par } (x - 3) = 0 \text{ ou } (x - 4) = 0$$

d'où  $x = 3$  ou  $x = 4$

$$\text{De même, } (x - 5)(x - 7) = 3 \text{ donnera } (x - 5) = 3 \text{ ou } (x - 7) = 3,$$

d'où  $x = 8$  ou  $x = 10$

La difficulté de ce genre d'approche pour expliquer les erreurs produites est la sélection des techniques d'extrapolation censées être utilisées par les apprenants. Pour étayer ses choix, Matz avance trois raisons : les techniques établissent un lien entre les anciens et les nouveaux problèmes, elles sont familières aux étudiants et se sont avérées pertinentes dans des situations passées.

Dans un domaine tout à fait différent, celui de l'enseignement de l'anglais comme langue étrangère, Yazdani (1989) a effectué un travail similaire. Elle montre que des correspondances entre la grammaire anglaise et celle de la langue maternelle des apprenants permet une meilleure prise en compte des erreurs grammaticales que les larges taxinomies des erreurs courantes utilisées dans les systèmes précédents.

La recherche des sources des erreurs et de leur mode de génération en liaison avec les connaissances des sujets est encore très embryonnaire. On constate cependant que les erreurs proviennent essentiellement de déviations sur des propriétés de surface et de préconceptions erronées. Il apparaît que les erreurs sont les résultats des tentatives raisonnables bien qu'infructueuses d'adapter des connaissances préalablement acquises à de nouvelles situations. Toutefois, les types d'adaptation susceptibles d'être choisis semblent dépendre fortement des caractéristiques des divers domaines de connaissances. L'analyse de ces caractéristiques est un champ d'étude privilégié pour l'épistémologie et la didactique. Des théories explicatives des erreurs devraient faciliter les explications et les suggestions de remédiation et permettre de faciliter la génération des bogues.

### 3.3.2.2. *Le diagnostic*

Disposant d'analyses précises sur les « maladies et leurs symptômes », l'objet du diagnostic va consister à déceler certains « symptômes » dans le comportement de l'apprenant pour éventuellement déterminer sa « maladie ». Ce comportement n'étant accessible qu'au travers de certains observables qui reposent sur les possibilités offertes par l'interface, il faut l'inférer à partir des observables disponibles. Cette inférence va s'appuyer sur des techniques d'intelligence artificielle.

La classification de Van Lehn (figure 3.10) indique les techniques de diagnostic utilisées dans différents systèmes. Dans le cas des modèles d'expertise partielle (WEST, WUSOR), il s'agit de rechercher les compétences élémentaires intervenant dans les actions de l'apprenant. Nous avons déjà donné quelques précisions sur cette technique. Dans les autres cas, il s'agit de trouver les bogues dans les connaissances de l'apprenant. Pour ce faire, le système de diagnostic va essayer, en utilisant des règles correctes et des règles erronées, de présenter un comportement identique à celui de l'élève, c'est-à-dire conduisant aux mêmes éléments observables. La qualité et la quantité de ces observables vont nécessiter le recours à des techniques plus au moins complexes.

Dans le cas de la résolution d'un problème ou de l'application d'une technique complexe, l'apprenant va fournir différentes étapes. Trouver l'ensemble des règles utilisées entre deux étapes successives va être la tâche du système de diagnostic. La difficulté croît suivant l'écart entre les étapes. Ainsi, la méthode du traçage de modèle (LISP TUTOR ou GEOMETRY TUTOR) suppose que les étapes successives traduisent les différents états mentaux significatifs de l'apprenant dans sa résolution du problème et donc correspondent à l'application d'une seule règle. Dans d'autres cas (LMS par exemple), l'écart entre deux étapes nécessite l'application de plusieurs règles. Il s'agit de combiner différentes règles correctes ou erronées, c'est-à-dire de reconstituer un chemin pouvant *expliquer* la transition entre les deux étapes<sup>17</sup>. La combinatoire risque rapidement de devenir importante, enjoignant de définir des critères pour la limiter. Dans son travail de conception d'un système conseiller sur le logiciel de calcul formel MACSYMA, Genesereth (1982) met en œuvre des techniques de reconnaissance de plans. En effet, dans des domaines complexes où on ne peut fixer une procédure à l'avance, plusieurs chemins peuvent conduire à la solution d'un problème ce qui oblige à analyser les étapes d'une solution pour inférer le plan de résolution suivi par un étudiant.

Notons que ces différentes techniques aboutissent à ce que l'on peut appeler un diagnostic comportemental. L'analyse porte avant tout sur la détermination de *comment* fait l'apprenant et non sur *pourquoi* il le fait. Elle reste au niveau des symptômes et des maladies, des erreurs de surface, sans être à même d'en élucider les causes, c'est-à-dire les erreurs profondes. La critique fondamentale aux types de

---

17. Le problème consiste à trouver un chemin dans un espace de recherche, technique que nous verrons un peu plus loin dans ce chapitre.

modélisation que nous venons de voir est leur faible pouvoir explicatif. Ces modèles invitent à des interventions du type compléter et corriger alors qu'un examen plus complet des causes des erreurs pourrait conduire à d'autres modes d'intervention. Si les modèles différentiels permettent la prise en compte des erreurs de nature persistante, on constate que certains bogues résistent à la détection et à la remédiation pendant des années. Pire, ils peuvent refaire surface spontanément après une remédiation apparemment réussie (Van Lehn, 1990). La question est de savoir s'il est possible de remédier à un niveau local ou si l'existence de bogues exige une réorganisation en profondeur des connaissances déjà acquises et le développement de processus de contrôle efficaces sur ces connaissances. Certains auteurs soulignent la nécessité d'aider l'élève à établir des liens entre les procédures et les concepts, pour favoriser les liens entre le sens et la syntaxe, de même que l'intérêt de la manipulation conjointe de points de vue multiples sur la connaissance (une des conclusions de WHY). Peu à peu, les chercheurs sont amenés à s'intéresser à la métacognition, c'est-à-dire à la connaissance et au contrôle des processus de pensée. Les machines ne doivent plus seulement avoir une connaissance de leur domaine d'enseignement et être capables de résoudre les problèmes posés à l'élève, mais disposer de connaissances sur leurs propres connaissances pour justifier et expliquer leurs solutions.

### 3.4. Résoudre et communiquer

Pour enseigner, selon l'expression de Siklössy, une machine doit « connaître » ce qu'elle « enseigne ». Ce principe est à la base du courant qui a introduit les techniques d'intelligence artificielle dans l'enseignement assisté par ordinateur. D'abord utilisée par les programmes adaptatifs, puis les programmes génératifs, la simple capacité de résoudre les problèmes posés à l'apprenant s'est peu à peu montrée insuffisante. Non seulement, la machine doit pouvoir résoudre, mais cette résolution doit être présentée à l'apprenant. Dans la représentation en machine de l'expertise d'un domaine se posent notamment des problèmes d'adéquation entre le raisonnement effectué par la machine et celui compréhensible par un expert ou un apprenant considéré comme un novice.

Dans un premier temps, les chercheurs distinguent l'expert *démonstratif* (articulate<sup>18</sup> expert), capable d'expliquer ses décisions en termes correspondants (à un certain niveau d'abstraction) à l'homme et l'expert dit *opaque*, la boîte de verre par opposition à la boîte noire (*glass-box* and *black-box* experts, Goldstein et Papert, 1977). Il s'agit souvent de rendre les routines d'inférences moins puissantes (ou de contraindre les représentations utilisées par la machine) afin qu'elles s'adaptent au raisonnement humain et rendent le fil du raisonnement plus compréhensible à un

---

18. Le terme anglais *articulate* n'est pas simple à traduire. Il s'applique à quelqu'un qui s'exprime bien et facilement. Les adjectifs éloquent ou loquace ne semblant pas très appropriés, j'ai préféré le terme démonstratif.

apprenant. Intégrer en outre, de manière explicite, les concepts stratégiques et organisationnels va s'avérer nécessaire.

Au-delà des possibilités d'explication, les recherches autour des conceptions erronées des élèves montrent l'intérêt de faire effectuer les tâches par la machine d'une façon similaire à celle des humains, experts et débutants, pour être à même de proposer un diagnostic suffisamment précis du travail des apprenants. Si certains chercheurs parlent alors de validité psychologique de leurs modèles, il s'agit simplement de garantir une certaine adéquation comportementale. En tous cas, ces diverses exigences conduisent à des réflexions en profondeur sur la façon de représenter la connaissance en machine et donc débouchent sur des problèmes d'épistémologie et de sémiologie.

Dans cette section, bien que des approches générales soient peut-être valides, nous allons nous centrer sur les mathématiques. En effet, vu la complexité de la tâche et les spécificités inhérentes à chaque domaine, il est difficile de s'abstraire des contraintes disciplinaires. De plus, mis à part l'informatique elle-même, les mathématiques ont été le domaine de loin le plus exploré et ont constitué le champ de référence de la résolution de problèmes. Passer en revue les travaux sur les mathématiques nous permet ainsi de comprendre les évolutions des résolveurs de problèmes utilisés dans un but d'enseignement.

La résolution d'un problème est basée essentiellement sur deux principes : trouver une *bonne* représentation et choisir les *bonnes* transformations ou méthodes à appliquer. La difficulté est qu'il n'y a aucune mesure universelle permettant de décider si une représentation ou une transformation est bonne. Cela dépend du problème considéré, mais aussi de l'agent (homme ou machine) devant le résoudre, de ses compétences et préférences. Les écarts entre le mode de résolution de la machine, celui de l'expert ou celui du débutant, les capacités à expliciter les choix effectués vont s'avérer déterminants. La question posée pour la machine est de savoir quel(s) mode(s) de résolution elle doit adopter et quelles connaissances elle doit disposer sur ce ou ces modes de résolution. On arrive à la modélisation du raisonnement mathématique.

Nous allons retracer les étapes principales de cette évolution. Partant des précurseurs ayant essayé de décrire les méthodes utiles pour résoudre des problèmes, passant par les premiers travaux en intelligence artificielle intégrant ces méthodes et par les premiers systèmes d'enseignement intégrant de tels résolveurs, nous arriverons aux systèmes « experts » en mathématiques.

#### **3.4.1. Résoudre automatiquement des problèmes**

Comment faire pour qu'une machine puisse résoudre des problèmes ? De grands mathématiciens se sont penchés sur leur pratique et ont essayé de préciser les mécanismes mentaux intervenant dans les processus de découverte. Leurs réflexions ont profondément influencé les travaux de recherche en intelligence artificielle. Nous allons d'abord rendre compte de quelques uns de leurs points de vue, puis nous verrons comment les chercheurs en intelligence artificielle s'en sont inspirés pour développer des systèmes de démonstration automatique. En fait, deux approches différentes ont été suivies, l'une de nature algorithmique, l'autre de nature plutôt heuristique. Pour intégrer des solveurs de problèmes dans des systèmes à vocation éducative, la deuxième approche s'est montrée plus intéressante, de par sa plus grande proximité avec les modes de résolution *humains*.

#### 3.4.1.1. Hadamard et la psychologie de l'invention mathématique

Prenant comme point de départ des idées exprimées par Poincaré lors d'une conférence à la Société de Psychologie en 1908, Hadamard (1945) a fait une étude sur la psychologie de l'invention dans le domaine mathématique, sur la base d'un travail d'introspection<sup>19</sup> et d'une enquête auprès d'une centaine de mathématiciens. Dans son ouvrage, il met en exergue l'apport de l'inconscient dans le processus d'invention. Rappelant que *cogito* signifie agiter ensemble et *intelligo* choisir parmi, il souligne l'existence de deux processus fondamentaux : l'élaboration de nombreuses combinaisons et le choix des combinaisons utiles.

Selon l'expression de Paul Valéry (ibid., p. 37), « inventer, c'est choisir », il faut être deux pour inventer. L'un forme les combinaisons, l'autre choisit, reconnaît ce qu'il désire et ce qui lui importe dans l'ensemble des produits du premier. Hadamard note l'accord entre le mathématicien et le poète sur le fait que l'invention consiste en un choix. Comment un tel choix peut-il s'opérer ? D'après lui, les règles qui doivent le guider « sont extrêmement fines et délicates, il est à peu près impossible de les énoncer dans un langage précis ; elles se sentent plutôt qu'elles ne se formulent ; comment dans ces conditions imaginer un crible capable de les appliquer mécaniquement ? » En accord avec Poincaré, ce choix lui semble lié à la sensibilité esthétique, gouverné de façon impérative par le sens de la beauté scientifique<sup>20</sup>.

Hadamard décompose le processus de l'invention en plusieurs phases : d'abord la préparation, puis l'incubation, période de repos pour se débarrasser des fausses voies

---

19. On peut remarquer qu' Hadamard prend nettement ses distances vis-à-vis du behaviorisme (Watson). Il considère que ces théories ne peuvent en rien concerner son étude (Hadamard, op.cit., p. 14 et p. 81).

20. Cette idée est commentée par Papert (1980), dans son épilogue sur « l'inconscient mathématique ». Il en fait une des bases de sa vision de l'apprentissage des mathématiques avec les micromondes.



et des hypothèses gênantes (hypothèse de l'oubli<sup>21</sup>) et enfin l'illumination. Elles sont suivies de deux étapes techniques, la vérification puis la finition.

Dans le chapitre 6 (La découverte en tant que synthèse, l'aide des signes), il essaye de caractériser son propre mode de raisonnement. Tout raisonnement doit lui apparaître comme une chose unique, « *je n'ai pas la sensation de l'avoir compris tant que je n'ai pas réussi à le saisir en une seule idée globale* » (ibid., p. 67). Il recherche alors l'aide offerte à la pensée par des représentations concrètes<sup>22</sup>. Cet aspect est confirmé par son enquête auprès des mathématiciens. La majorité déclare ne pas penser sous forme de symboles mais utiliser une approche essentiellement visuelle. Tous évitent l'usage mental de mots mais aussi de signes précis, algébriques ou autres. Ils utilisent des images vagues. Une exception notable est Polya (ibid., p. 83) qui utilise un seul mot, qu'il qualifie de mot *décisif*, un seul mot dans tout un cours de pensée de manière à se rappeler l'idée centrale.

Ainsi, pour Hadamard, il s'agit de penser en images, des images vagues, non isomorphes aux structures mathématiques étudiées, mais offrant différentes combinaisons possibles jusqu'à arriver à un stade où le jeu d'associations est suffisamment établi et peut être reproduit à volonté. D'après lui, nous ne pensons en termes de symboles que lorsque l'accumulation répétitive d'exemples concrets devient suffisamment lassante pour qu'on désire remplacer la multitude de cas particuliers par une abstraction unique. Il avoue d'ailleurs une certaine méfiance vis-à-vis des mots. « Je pense...que plus une question est difficile et complexe, plus nous nous méfions des mots, plus nous estimons que nous devons contrôler cet allié dangereux et sa précision parfois perfide. » (ibid, p. 93)

Il est intéressant de constater que les mécanismes de la découverte décrits par Hadamard, même si leur validité est loin d'être établie scientifiquement, ont inspiré des informaticiens comme Alan Kay ou Douglas Lenat et des techniques comme celle des algorithmes génétiques<sup>23</sup>, notamment le travail qu'il attribue à l'inconscient dans le processus de découverte. Si les descriptions d'Hadamard demeurent relativement générales, Polya va s'attacher à préciser les méthodes permettant de résoudre les problèmes, en insistant sur le rôle central joué par les heuristiques.

---

21. Piaget pense qu'un des rôles de l'inconscient peut être de supprimer les fausses pistes (Beth et Piaget, 1961).

22. Il illustre son propos avec la démonstration du théorème prouvant que la suite des nombres premiers est illimitée. Le commentaire d'Hadamard est repris par Pitrat (1993, p. 125) pour montrer l'intérêt des diagrammes et des figures comme mode de représentation des connaissances.

23. Pour Goldberg (1989, p. 13), les algorithmes génétiques adoptent le mélange d'orientation et de chance décrit par Hadamard d'une telle manière que cela construit efficacement de nouvelles solutions à partir des meilleures solutions partielles des précédents essais.

#### 3.4.1.2. *Polya, les heuristiques et le raisonnement plausible*

D'après Polya (1965, p. 93), l'heuristique ou heurétique, *ars inveniendi*<sup>24</sup> avait pour objet l'étude des règles et des méthodes de la découverte et de l'invention. On en trouve des éléments chez les commentateurs d'Euclide, chez Pappus, et des essais conséquents chez Descartes, Leibnitz et Bolzano. « L'heuristique moderne s'efforce de comprendre la méthode qui conduit à la solution des problèmes, en particulier les *opérations mentales qui s'avèrent typiquement utiles* à l'application de cette méthode (...) Une étude sérieuse de l'heuristique (...) doit s'attacher à l'expérience objective. Une expérience, qui résulte à la fois de la solution des problèmes et de l'observation des méthodes d'autrui<sup>25</sup>, constitue la base sur laquelle on construit l'heuristique. » Le livre de Polya est une exploration de cette notion d'heuristique. La démarche qu'il préconise pour résoudre un problème se compose de quatre étapes principales : comprendre le problème, concevoir un plan, mettre le plan à exécution, examiner la solution obtenue.

Notons que son ouvrage a une structure très particulière. Il est divisé en trois parties, la troisième étant un dictionnaire comprenant soixante-sept articles classés par ordre alphabétique. Cette organisation permettant différents modes de circulation confère à son livre une sorte de structure hypertextuelle. On peut y voir l'affirmation du caractère profondément non linéaire de son propos ou plus simplement son impossibilité d'une exposition entièrement linéaire.

Dans un autre ouvrage, concernant le problème de la justification heuristique, Polya (1954) prend l'exemple d'une démonstration telle qu'elle serait rédigée dans un ouvrage de mathématiques.

« Une conférence ou un livre de mathématiques doivent, tout d'abord, être corrects et non ambigus. Mais nous savons, par une dure expérience, qu'un exposé parfaitement correct et non ambigu peut se trouver loin d'être satisfaisant, paraître peu enthousiasmant, ennuyeux ou même décevant, même si le sujet présenté est intéressant en lui-même. Le blâme le plus grave qu'on puisse adresser à un exposé, acceptable par ailleurs, est l'intervention d'un *deus ex machina*. » (ibid., p. 258).

Or, d'après Polya, un lecteur ou un auditeur intelligent désire deux choses voir que l'étape actuelle du raisonnement est *correcte* et qu'elle est *adéquate*.

Une étape d'un raisonnement mathématique est adéquate si elle est liée de façon essentielle au but, si elle nous en rapproche. Toutefois, il ne suffit pas “ qu'une étape

---

24. On distingue *ars inveniendi* (heuristique, a priori) et *ars disserendi* (démonstration, a posteriori).

25. Comme le remarque Balacheff (1977), le travail de Polya reste essentiellement fondé sur l'introspection. Des méthodes expérimentales, appuyées sur des théories plus précises, ne seront utilisées qu'ultérieurement.

*soit* adéquate, il faut encore qu'elle *apparaisse* au lecteur sous ce jour ” (ibid., p. 259). Cette distinction essentielle sera prise en compte peu à peu dans les démonstrateurs de théorèmes. Ainsi, les nécessités de communication ont des effets sur la présentation de la solution et donc sur la forme même de cette solution. Polya se pose d'ailleurs le problème de l'existence de règles de raisonnement plausible<sup>26</sup>. Il remarque (ibid, p. 232) que reconnaître l'existence d'une règle revient à déléguer une partie de son autorité à une machine et constate que les experts sont peu enclins à le faire. Enfin, il note le rôle particulier joué par le calcul, à la fois heuristique et démonstratif. La question qui va se poser est de savoir si tout se réduit au calcul, sans mésestimer l'importance de l'intuition spatiale soulignée par Hadamard.

D'une manière assez voisine de Polya, Beth (Beth et Piaget 1961, p. 24) distingue trois phases consécutives dans la démonstration : la phase de recherche, la phase d'aménagement et la phase de vérification. C'est uniquement cette dernière phase qui est reproduite dans une publication mathématique. La phase d'aménagement est dépendante de la phase de recherche, son aboutissement conditionne le passage à la phase de vérification. La première phase n'est guère reproduite sous forme compréhensible, elle n'est pas communicable. Beth conclut en disant qu'une simple analyse de raisonnements mathématiques ne fournira pas les données permettant de reconstruire la pensée mathématique productive. Les chercheurs seront ainsi amenés à trouver d'autres méthodes pour dégager les règles du raisonnement. L'observation de mathématiciens, au cours de leur phase de recherche, va fournir des données essentielles. Les suggestions de Polya vont rencontrer un écho favorable du côté des chercheurs en intelligence artificielle qui s'en inspireront pour bâtir leurs systèmes.

#### 3.4.1.3. *L'intelligence artificielle et l'approche heuristique*

Les réflexions d'Hadamard et de Polya conduisent à souligner deux aspects essentiels pour résoudre des problèmes, trouver une bonne forme de représentation offrant de multiples combinaisons et une manière de choisir entre ces différentes combinaisons. Sous l'impulsion de Newel, Shaw et Simon, les chercheurs en intelligence artificielle les traduisent notamment par les notions d'espace problème et d'heuristique.

Un problème de recherche dans un espace d'état est défini par un triplet (I, O, F) où I est un ensemble d'un ou plusieurs états initiaux, O un ensemble d'opérateurs et F un ensemble d'états finaux. Un espace de recherche ou un espace problème est

---

26. « J'ai dit qu'il était désirable d'enseigner à deviner, mais je n'ai pas dit qu'il était facile de le faire. Il n'existe pas de méthode sans danger pour deviner, aussi ne peut-il exister de méthode sans danger pour enseigner l'art de deviner. J'ai peut-être émis quelques opinions peu sensées dans les pages précédentes, mais j'ai évité la plus insensée de toutes, qui serait de prétendre que je possède une méthode infallible pour enseigner à deviner. Néanmoins, il n'est pas impossible d'enseigner à deviner » (ibid., p. 272).

représenté par un ensemble d'états initiaux, intermédiaires et finaux. Les opérateurs permettent de passer d'un état à un autre. La recherche consiste à appliquer un opérateur sur l'état courant jusqu'à l'obtention d'un état terminal. Les stratégies de recherche peuvent être aveugles, c'est-à-dire conduisant à l'application de tous les opérateurs. La recherche heuristique, en revanche, utilise des stratégies d'évaluation et de contrôle pour décider quel(s) opérateur(s) appliquer et où il(s) doit(vent) être appliqué(s). Dans une telle formalisation, il faut noter que les difficultés principales concernent le choix, d'une part, de la représentation (détermination des états et des opérateurs valides) et, d'autre part, de la stratégie de contrôle, permettant de sélectionner les meilleurs opérateurs à appliquer. Les systèmes de diagnostic automatique, notamment LMS, fournissent des exemples de formalisation par un espace problème (voir plus haut).

LOGIC THEORIST (Newell, Shaw et Simon, 1956) est un des premiers programmes utilisant des heuristiques dans la résolution d'un problème. Son domaine est celui de la démonstration de théorème en calcul propositionnel, sur la base de l'axiomatique de Whitehead et Russel. Ce programme a pu démontrer 38 des 52 premiers théorèmes des *Principia Mathematica*. Son successeur est GPS<sup>27</sup> (*General Problem Solver* ou résolveur de problèmes généraux) développé par Newell, Shaw et Simon à partir de 1957 (puis repris par Ernst et Newell en 1967). Ce programme est basé sur un calcul de différences entre un état intermédiaire et un état final, différences classées en types auxquels sont associés des opérateurs pouvant assurer leur réduction.

A la suite de GPS, de nombreux programmes montrant la fécondité d'une approche heuristique ont été réalisés. Signalons simplement deux programmes d'intégration symbolique : SAINT (Symbolic Automatic INTEgrator) écrit en 1961 par James Slagle, puis SIN (Symbolic INtegrator) écrit par Joël Moses en 1967 (voir Moses, 1971).

Avant d'aborder les questions d'utilisation de tels programmes dans des systèmes d'enseignement, deux exemples méritent un détour car ils introduisent des comparaisons intéressantes entre les modes de raisonnement des machines et ceux des humains, experts ou débutants.

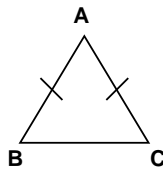
Le premier, STUDENT, programme écrit par Daniel Bobrow en 1964, résout des problèmes algébriques simples formulés en langue naturelle (l'anglais). Si la résolution elle-même ne pose pas de difficulté particulière, le processus de traduction de l'énoncé en équations est intéressant. Il s'appuie sur un certain nombre de schémas de base qui sont comparés avec le texte proposé. Ainsi, sans véritable compréhension de l'énoncé, en se basant uniquement sur ses aspects de surface, le

---

27. Concernant les aspects généraux de l'intelligence artificielle, le lecteur pourra consulter les deux livres suivants : Laurière (1987, p. 238-254 pour une analyse plus détaillée sur GPS) et Barr et Feigenbaum (1981).

programme arrive à résoudre un très grand nombre de problèmes typiques, bien qu'il soit aisément mis en défaut par des formulations un peu complexes. Une version légèrement modifiée a été considérée comme un bon modèle pour la résolution des problèmes par les humains (Paige et Simon, 1966)<sup>28</sup>. Ainsi, une approche manifestement insuffisante conduit à un modèle acceptable des performances des apprenants.

Le second est le démonstrateur de théorèmes en géométrie de Gelernter (1963), un des premiers démonstrateurs automatiques de théorèmes. Grumbach (1994, p. 10) rappelle la démonstration qu'un triangle ayant deux côtés égaux a aussi deux angles égaux proposée par ce système :



En considérant les triangles BAC et CAB, on peut conclure que ces deux triangles sont égaux ( $BA = CA$ ,  $AC = AB$  et l'angle BAC est égal à l'angle CAB). D'après les cas d'égalité des triangles, on en déduit l'égalité des triangles et l'égalité des angles ABC et ACB.

Cette démonstration, à première vue un peu déroutante, montre l'écart entre le tracé perceptif, la représentation et l'objet mathématique (les deux triangles BAC et CBA sont à la fois différents et identiques). Plus généralement, le programme utilise la symétrie syntaxique<sup>29</sup> pour établir des relations par simple permutation des lettres correspondant aux points des figures considérées, ce qui simplifie grandement les démonstrations (Gilmore, 1970). Un autre aspect intéressant est le fait que ce programme s'aide de diagrammes pour guider sa démonstration, à la fois pour rejeter des sous-buts et obtenir certains faits. Chaque fois qu'un sous-but est généré, il est vérifié pour sa consistance en fonction du diagramme. S'il est faux dans le diagramme, il est abandonné. On a estimé que l'utilisation d'un diagramme avec rejet des sous-buts représentant des raisonnements circulaires éliminait 99,5% des sous-buts générés (Barr et Feigenbaum, 1981, p. 105).

Ainsi, même pour les machines, les problèmes traitant de relations spatiales nécessitent l'emploi de représentations *ad hoc*. De nombreux autres programmes de démonstration vont tirer parti de représentations particulières permettant de mieux mettre en évidence certaines propriétés et éviter de longues et fastidieuses

28. Voir Pitrat (1993, p. 42) pour une discussion plus approfondie de STUDENT.

29. Pitrat (1993, p. 68) décrit un exemple d'utilisation de la symétrie syntaxique dans une démonstration un peu plus complexe que la précédente.

recherches<sup>30</sup>. Les machines étant à même de faire certaines démonstrations et de résoudre des problèmes, il devient possible d'utiliser ces capacités dans des systèmes dédiés à l'enseignement.

### **3.4.2. Les premiers programmes d'enseignement de la logique**

Au-delà des problèmes numériques simples, bases des programmes génératifs, le développement de démonstrateurs de théorèmes permet d'accroître le nombre des domaines pour lesquels un tel type de programme s'avère faisable. A la suite de l'arithmétique, de nouveaux champs sont explorés ; le premier d'entre eux, de par sa proximité évidente avec les premiers démonstrateurs, va être la logique.

Dans un système d'enseignement, un démonstrateur de théorèmes est utile à la fois comme vérificateur de preuves et comme dispositif d'aide lorsque l'étudiant est bloqué. Gable et Page (1980) distinguent trois modes de fonctionnement d'un vérificateur de preuves :

- vérification en pas-à-pas, avec possibilité d'insertion des lignes nécessaires pour rendre la preuve rigoureuse, mais sans autre type de rétroaction,
- vérification des preuves complètes, non des étapes intermédiaires, avec évaluation et pointage des erreurs. L'absence de rétroaction immédiate peut contraindre l'étudiant à réécrire sa démonstration plusieurs fois,
- examen de chaque pas de démonstration, dès qu'il est entré par l'étudiant, avec rétroaction immédiate.

Deux grands types de démonstrateur sont a priori candidats pour s'intégrer dans des systèmes d'enseignement. Basé sur le principe de résolution de Robinson (1965), le premier, bien que très efficace, utilise des méthodes de preuve particulières générant de très nombreuses étapes intermédiaires difficilement compréhensibles, très différentes de celles des étudiants et se révèle quasiment inutilisable. En revanche, le second utilise des heuristiques, comme LOGIC THEORIST ou GPS et s'adapte mieux à l'enseignement.

Illustrant le troisième mode de fonctionnement d'un vérificateur de preuves, l'un des premiers exemples d'enseignement de la logique à l'aide d'un démonstrateur est un programme conçu à Toronto (Vaskévitch, 1972). L'interaction est du type initiative mixte. L'étudiant interroge l'ordinateur en formulant un théorème. L'ordinateur peut vérifier la démonstration proposée par l'étudiant ou l'aider en pas-à-pas à en établir une. L'étudiant conserve la maîtrise du dialogue, mais le langage

---

30. Laurière (1987, p. 34) note que la figure est un abrégé de l'information, pouvant traduire efficacement une chaîne de déductions. Elle permet d'éliminer des sous-buts faux et sert d'aide à la découverte. En particulier, elle peut guider le choix d'objets nouveaux.

de *conversation* est particulier. En effet, la seule forme de question possible est la proposition d'un énoncé comme théorème.

### 3.4.2.1. Le programme de Goldberg

Dans le cadre d'un travail développé à Stanford (à la suite d'un programme écrit par Suppes) et constituant sa thèse, Adèle Goldberg (1974) a développé un tuteur sur la logique mathématique élémentaire. Il a pour objet d'aider des étudiants dans l'apprentissage de la résolution de problèmes de nature purement formelle. Ils peuvent suivre un curriculum spécifique dans le contexte d'une théorie axiomatique particulière ou proposer un système d'axiomes avec leur propre langage pour construire des preuves. Ce système a effectivement été utilisé par de nombreux étudiants.

Goldberg demande à un démonstrateur de prendre en considération la fois :

- les étapes déjà établies par l'étudiant,
- la manière dont ces étapes s'intègrent à des preuves complètes,
- les stratégies de résolution qu'il connaît se rapprochant le plus des idées de l'étudiant,
- les types de procédures qui rentrent dans la découverte de telles solutions.

Son programme s'appuie sur un démonstrateur de théorèmes et fonctionne en quatre étapes :

- on présente un théorème à un étudiant et on lui demande de le prouver,
- l'étudiant tape au clavier des pas de démonstration. Le démonstrateur de théorèmes tente de prouver chacun de ces pas à partir des pas précédents. S'il n'y arrive pas il le signale à l'étudiant,
- sur demande de l'étudiant, le démonstrateur fournit des conseils,
- quand l'étudiant termine une démonstration, le prouveur est utilisé pour suggérer d'autres preuves.

Les figures 3.11 et 3.12 fournissent un exemple d'interaction avec le système développé par Goldberg.

Commutativité	(Co)	$A + B = B + A$
Associativité	(As)	$(A + B) + C = A + (B + C)$
Élément neutre	(El)	$A + 0 = A$
Inverse	(In)	$A + (-A) = 0$
Opération inverse	(Oi)	$A + (-B) = A - B$

**Figure 3.11.** Axiomes de la théorie des groupes

Grâce à la flexibilité offerte par le démonstrateur, l'étudiant n'est pas contraint de suivre des chemins prévus à l'avance (avec des preuves stockées en mémoire) et n'a pas besoin de donner toutes les étapes en détail ; pourvu que les sauts ne soient pas

trop gros, les pas manquants peuvent être reconstitués. Si pour la vérification des pas proposés par l'étudiant, disposer d'un résolveur est certainement avantageux, cela devient indispensable pour fournir des conseils adaptés à la progression de l'élève et pour trouver toutes les possibilités de finir la preuve à partir des étapes déjà réalisées. Dans le fonctionnement de ce système, deux points importants méritent d'être soulignés : la reconnaissance en pas à pas par reconstruction des chemins valides et la possibilité de jouer sur la granularité de la preuve.

Voici le théorème 8 à prouver : Prouver $((A + B) = C) \Rightarrow (A = (C - B))$		
th7	$A + B = A + C \Rightarrow B = C$	
A::B, B::A, C::C-B	(1) $((B + A) = (B + (C - B)) \Rightarrow (A = (C - B))$	Demande de substitution dans la formule précédente
ICo1	(2) $((A + B) = (B + (C - B)) \Rightarrow (A = (C - B))$	Application de la commutativité.
Aide	Rappelez-vous que soustraire un terme est la même chose qu'ajouter son opposé.	
Aide	Utilisez l'axiome (Oi) pour changer l'occurrence 1 de C - B en C + (-B).	La deuxième demande d'aide la rend plus explicite
Oi	$A + (-B) = A - B$	
A::C, B::B	(3) $(C + (-B)) = (C - B)$	
Co1	(4) $(C - B) = (C + (-B))$	
2.4 Re	(5) $((A + B) = (B + (C + (-B)))) \Rightarrow (A = (C - B))$	Remplacement de (4) dans (2).
Aide	Vous voulez transformer B + (C + (-B)) en C. Utilisez Co, As et Th2.	L'aide est ici très précise, elle fournit le cheminement jusqu'à la solution.
Co2	(6) $((A + B) = ((C + (-B)) + B)) \Rightarrow (A = (C - B))$	
As1	(7) $((A + B) = (C + ((-B) + B)) \Rightarrow (A = (C - B))$	
Th2	$(-A) + A = 0$	
A::B	(8) $((-B) + B) = 0$	
7.8 Re	(9) $((A + B) = (C + 0)) \Rightarrow (A = (C - B))$	Remplacement de (8) dans (7).
El	(10) $((A + B) = C) \Rightarrow (A = (C - B))$ Correct.	

**Figure 3.12.** Exemple de session avec le tuteur de Goldberg  
(d'après Gable et Page, 1980)

REMARQUE : dans l'interaction, on demande un axiome ou un théorème qui est alors affiché, il faut alors préciser les substitutions à faire sur les variables de l'expression.



## 3.4.2.2. EXCHECK

EXCHECK est un programme d'enseignement conçu à l'université de Stanford par P. Suppes et son équipe (Blaine et Smith, 1977). C'est un cours complet de niveau universitaire sur la logique, la théorie des ensembles et la théorie de la preuve<sup>31</sup>. Il a longtemps été suivi par plusieurs centaines d'étudiants chaque année. Suppes (1988) affirme que ce cours est probablement le programme intégrant un démonstrateur de théorèmes et utilisé journallement par un nombre important de personnes qui tourne depuis le plus longtemps.

L'intérêt d'EXCHECK est lié à son démonstrateur. Ce dernier cherche à suivre la démonstration d'un étudiant en termes de procédures reflétant fidèlement l'approche humaine, appelée souvent naturelle, de la démonstration de théorèmes. EXCHECK peut suivre un raisonnement informel, incluant l'utilisation de lemmes bien connus et le saut de pas. L'étudiant dispose d'un petit nombre d'opérateurs d'une granularité suffisante (plus complexes qu'un simple pas élémentaire de démonstration) ce qui lui permet de déclarer ses intentions au système dans un langage simple et naturel.

La figure 3.13 présente deux démonstrations d'un théorème élémentaire. On peut remarquer qu'il n'y a pas de différence structurelle importante entre la preuve informelle détaillée et celle réalisée avec EXCHECK.

<p>EXC : Montrer : si <math>A \subset B</math> alors  <math>\neg (B \subseteq A)</math>  <i>Elève : hyp (1)</i>            EXC : <math>A \subset B</math>  <i>Elève : raa (reductio ad absurdum)</i>            EXC : Supposons (2) <math>B \subseteq A</math>  <i>Elève : Définition, sous-ensemble propre pour 1</i>            EXC : (3) <math>A \subseteq B</math> et <math>A \neq B</math>  <i>Elève : 2, 3 établissent</i>            EXC : 2, 3 établissent (4) <math>A = B</math>  <i>Elève : 3, 4 contradiction</i>            EXC : 3, 4 contradiction (5)  <math>\neg (B \subseteq A)</math>  <i>Elève : Cqfd</i>            EXC : Correct</p>	<p>Théorème : si <math>A \subset B</math> alors  <math>\neg (B \subseteq A)</math>            On suppose <math>A \subset B</math> et on va montrer  <math>\neg (B \subseteq A)</math> par un raisonnement par            l'absurde. On suppose donc <math>B \subseteq A</math>.            Comme <math>A \subset B</math>, par définition,  <math>A \subseteq B</math> et <math>A \neq B</math>.            Or <math>A \subseteq B</math> et <math>B \subseteq A</math> ce qui prouve            que <math>A = B</math>. C'est une contradiction,            donc la supposition <math>B \subseteq A</math> est fausse,            sa négation est vraie, donc <math>\neg(B \subseteq A)</math>.</p>
---	--

**Figure 3.13.** Exemple de session avec EXCHECK, d'après Barr et Feigenbaum (1982, p. 284)

31. Ce cours est basé sur un livre de Suppes publié en 1957 (voir Suppes, 1988).

D'après Wenger (op.cit., p. 105), EXCHECK n'a pas une connaissance suffisamment profonde du domaine pour l'enseigner. Néanmoins, la flexibilité de ce démonstrateur opaque permet à l'étudiant de poser des problèmes, de les résoudre avec le système et d'obtenir si nécessaire des retours intelligents bien que locaux. EXCHECK n'intègre pas de modèle de l'élève.

Toutefois, l'utilisation intensive de ce système par les étudiants témoigne incontestablement de son intérêt. La leçon que l'on peut en tirer est que le succès d'un système pour faire des mathématiques interactivement dépend d'une manière décisive du fait de disposer d'un petit nombre de procédures d'inférence naturelle puissantes, avec des critères clairs d'utilisation, suffisantes pour prendre en compte l'ensemble des déductions. Jouer sur la granularité permet certainement une adaptation aux différents apprenants. Si le langage de dialogue n'est pas la langue naturelle, un langage de haut niveau, adapté au domaine, paraît à peu près convenir dans le contexte.

EXCHECK et le programme d'Adèle Goldberg peuvent être considérés comme des environnements réactifs<sup>32</sup>, proches de SOPHIE I dans leur conception. D'une certaine manière, ces deux programmes fournissent des exemples de partenaires de résolution, dans le sens où la machine aide l'étudiant à réaliser une tâche qui lui est prescrite ou qu'il se choisit. L'extension de ces premiers programmes d'enseignement, basés sur un démonstrateur, conduit à explorer plusieurs directions : l'implantation de nouvelles stratégies d'enseignement, l'application à d'autres domaines des mathématiques, l'amélioration des capacités des solveurs. Concernant ce dernier aspect, il s'agit de passer, dans un premier temps, de solveurs opaques à des solveurs « démonstratifs », puis de demander aux solveurs de justifier les choix effectués en cours de résolution.

### **3.4.3. Vers des systèmes experts en mathématiques**

La voie heuristique, c'est-à-dire la modélisation de la résolution de problèmes par un espace d'états et un ensemble d'opérateurs va être suivie dans de multiples domaines des mathématiques. Outre la question du choix des représentations, trouver une bonne manière de sélectionner les opérateurs susceptibles de conduire vers la solution prend une importance considérable. Pour ce faire, l'observation des mathématiciens, durant la phase de recherche d'une démonstration, fournit une aide précieuse. Mais, une fois trouvée une manière de choisir les bons opérateurs à appliquer, il va falloir l'expliquer aux apprenants. Il s'agit de rendre les machines non seulement expertes, mais capables de communiquer leur expertise.

---

32. Goldberg (1974) parle de *responsive-sensitive teaching system*.

### 3.4.3.1. PRESS et la résolution d'équations

PRESS (PRolog Equation Solving System) est un bon exemple de système montrant comment la mise en évidence de sous-buts pertinents permet de guider le choix des heuristiques. Son domaine est celui de la résolution des équations et des manipulations algébriques sur des expressions transcendantes. Il est basé sur des idées exposées originellement par Bundy (Bundy 1975, Bundy et Welham 1981).

Dans son rapport de recherches sur l'analyse des preuves mathématiques (dont le sous-titre est « *lire entre les lignes* »), Bundy (1975) relève qu'un système de haut niveau de descriptions, stratégies et buts, peut être utilisé pour guider la recherche dans un espace de recherche potentiellement explosif. En effet, quand on modifie une expression, de multiples transformations sont possibles. Choisir les transformations au hasard ne conduit pas au résultat. Le mathématicien n'a pas un comportement erratique mais sélectionne avec précision les transformations qu'il effectue. Pour surmonter les phénomènes d'explosion combinatoire (exploration de toutes les transformations possibles), il faut avoir recours à des fonctions d'évaluation. Or, la prise en compte des seules caractéristiques d'une expression, comme sa complexité (tenant compte des opérateurs et des occurrences de l'inconnue), ne suffit pas. En revanche, la détermination de quelques sous-buts fournit les moyens de contrôle nécessaires.

La méthode de résolution de base dans PRESS s'appuie sur trois buts généraux : l'isolation, la collection et l'attraction. L'*isolation* s'applique à une équation contenant une seule occurrence de l'inconnue et consiste à appliquer récursivement la fonction réciproque de la fonction dominant l'inconnue. La *collection* (ou *rassemblement*) consiste à réduire le nombre d'occurrences de l'inconnue et l'*attraction* consiste à rapprocher deux occurrences de l'inconnue. La figure 3.14 donne un exemple de résolution d'une équation par PRESS, utilisant successivement l'attraction, la collection puis plusieurs fois l'isolation. Le rapprochement est lié à une notion de distance assez simple, bien que cachée dans l'écriture linéaire traditionnelle. La distance entre deux occurrences d'un symbole est la taille (ou le nombre de noeuds) du plus petit sous-arbre les reliant. La représentation en arbre la fait apparaître de manière très claire. Dans la partie droite de la figure 3.13, la distance entre les deux occurrences de  $x$  est 6 dans le premier arbre (correspondant à  $\text{Log}(x + 1) + \text{Log}(x - 1)$ ) et 4 dans le second ( $\text{Log}[(x + 1)(x - 1)]$ ), ou 5 et 3 en comptant en noeuds.

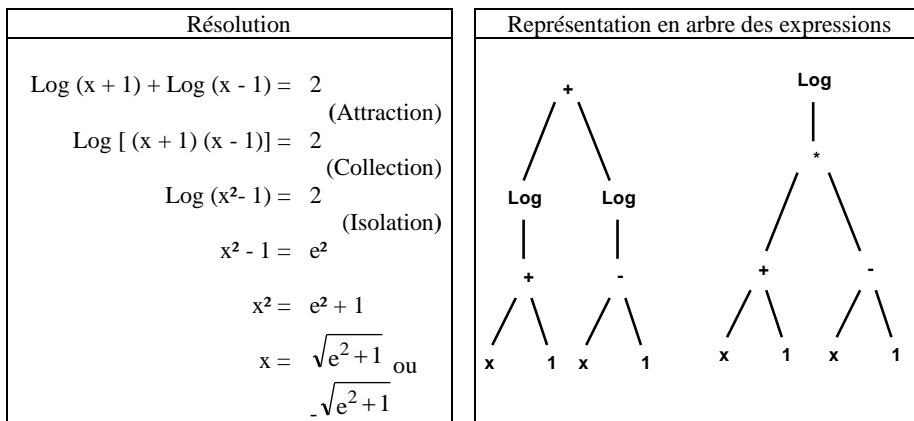


Figure 3.14. Exemple de résolution effectuée par PRESS

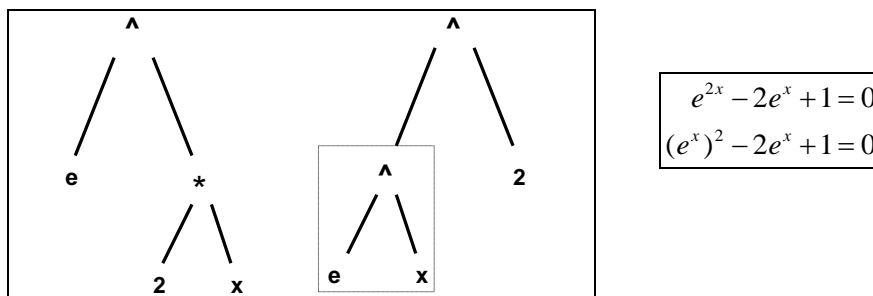
Les mêmes techniques s'appliquent dans l'équation suivante :

$$\begin{aligned}
 (2^{x^2})^{x^3} &= 2 && \text{Attraction} \\
 2^{x^2 \times x^3} &= 2 && \text{Collection} \\
 2^{x^5} &= 2 && \text{Isolation} \\
 x^5 &= 1
 \end{aligned}$$

REMARQUE : la technique d'isolation est en général explicitée dans l'enseignement (pour résoudre une équation, il faut isoler l'inconnue dans un membre). Les autres sont moins systématiquement montrées. Néanmoins, il n'est généralement pas précisé que l'isolation s'effectue en appliquant récursivement la fonction réciproque de la fonction dominant l'inconnue, principe unificateur pourtant très utile. Ainsi, dans une équation du premier degré le passage de  $x + 5 = 3$  à  $x = 3 - 5$  est souvent justifié par le fait qu'on ne change pas une égalité en ajoutant ou soustrayant le même nombre aux deux membres (c'est ce qui est dit dans les classes de collège). C'est aussi l'application de la réciproque de la fonction  $x \rightarrow x + 5$ . Cette explication par l'application d'une fonction réciproque rend mieux compte des difficultés associées aux fonctions non inversibles. De la même manière pour les inéquations, les règles de changement de sens cachent l'aspect fondamental lié à la monotonie de la fonction réciproque.

Dans la méthode suivie par PRESS, des buts équivalents sont poursuivis pour l'élimination des symboles fonctionnels considérés comme rares. Ceci est fait pour éviter de multiplier les règles de réécriture. Cela rend compte, en particulier, de la rationalisation, c'est-à-dire l'élimination des radicaux. Concernant le changement

d'inconnue, le procédé ne peut pas toujours s'effectuer directement, il est souvent précédé de l'*homogénéisation*. Cette méthode est décrite par Bundy et Silver (1981) puis par Silver (1986). Elle consiste à réaliser un prétraitement des équations pour déterminer un changement de variable à effectuer. Le programme regarde tous les sous-termes contenant une seule occurrence de l'inconnue et cherche à les exprimer à l'aide d'une même expression. Par exemple, dans l'équation ci-dessous, le changement de variable  $x = e^x$  apparaît grâce à la transformation de  $e^{2x}$  en  $(e^x)^2$ .



Même si PRESS n'a pas été utilisé comme résolveur d'un système d'enseignement de la résolution d'équations, mais intégré dans un résolveur nommé MECHO concernant la mécanique, il constitue une sorte de preuve effective (puisque le programme est capable de résoudre des équations complexes) de l'intérêt des idées sous-jacentes à sa conception. En particulier, les méthodes qu'il met en œuvre sont sans doute intéressantes à enseigner. En ce sens, PRESS est un des nombreux exemples mettant en évidence des heuristiques utilisables en dehors du système dans lequel elles ont été testées.

### 3.4.3.2. Les recherches en France

Au cours des années soixante-dix, de très nombreux domaines des mathématiques ont été explorés. En France notamment, différents chercheurs vont suivre la voie de recherche ouverte par Jacques Pitrat qui, dès 1966, avait conçu un programme qui ne se limitait pas à démontrer des théorèmes mais générait de nombreux théorèmes jugés *intéressants*. On peut ainsi citer le calcul de limites (Laurent, 1972), la vérification d'égalités par récurrence (Vivet, 1973), la résolution d'équations trigonométriques (Grandbastien, 1974), la théorie des ensembles (Pastre 1976), la topologie (Merialdo, 1979), les exercices d'arithmétique (Bourgoin, 1979), la théorie des groupes (Gillet 1979)... Si les systèmes précédents n'avaient aucune vocation particulière d'enseignement, les chercheurs vont s'intéresser, d'une part aux modes possibles de récupération des connaissances auprès des experts et, d'autre part, aux applications possibles à l'enseignement des résolveurs qu'ils conçoivent.

Ainsi le travail autour de SEME, développé par Monique Baron (1982), a pour objet de proposer des formes d'organisation et d'expression des connaissances utilisées dans un calcul formel de manière à permettre au mathématicien lui-même de les fournir en données d'un programme interpréteur général. Le domaine traité

par SEME est celui des sommes de combinaisons, domaine d'application qui est jugé intéressant. En effet, il est peu formalisé et les problèmes à résoudre ne contiennent pas dans leur énoncé la forme du résultat cherché. Un pilotage du calcul s'avère nécessaire. Non seulement, on ne dispose pas d'un répertoire des méthodes possibles face à tel ou tel exercice, mais les méthodes ne fournissent le plus souvent que des indications générales et floues quant à la succession des actions élémentaires conduisant à une solution pour un énoncé donné.

L'activité de résolution de l'expert mathématicien est analysée comme une succession de tâches dont le choix dépend des résultats partiels recherchés. Pour cela, le système utilise :

- des connaissances théoriques. Ce sont celles du cours de mathématiques : définition des combinaisons, valeurs particulières, relations générales, les formules classiques comme celles du binôme  $\sum_{k=0}^n x^k \times y^k \times C_n^k = (x+y)^n$ , de la colonne du triangle de Pascal  $\sum_{k=i}^n C_k^i = C_{n+1}^{i+1}$  ou de la convolution.
- des savoir-faire comme l'effectuation de calculs, le changement de variable, l'application d'une formule ou d'un théorème,
- des méthodes permettant de se ramener à une formule classique, d'établir une relation de récurrence, de décomposer l'expression courante,
- des connaissances expertes qui concernent le choix des méthodes à appliquer et le contrôle global de la résolution.

Ces connaissances, méthodes et savoir-faire sont représentées par des règles de réécriture et des règles de production. La mise au point du système se fait de manière incrémentale, par ajouts successifs de règles pour de nouveaux types d'exercice, en essayant de garantir la lisibilité et la modularité des connaissances exprimées. La trace du raisonnement fourni par l'interpréteur est une aide précieuse. Toutefois, Baron reconnaît (op.cit., p. 144) qu'il n'est pas facile d'exprimer de bonnes heuristiques, du fait de leur caractère non sûr et peu explicite habituellement. SEME permet de les expérimenter et de les modifier aisément, « mais leur qualité suppose une connaissance approfondie et une grande pratique du domaine ».

CAMELIA (Vivet, 1984) est un essai de combinaison et de synthèse des approches algorithmique et heuristique. Construit comme une sur-couche d'un système de calcul symbolique REDUCE (Hearn 1971), le programme tire parti à la fois des performances du résolveur opaque sous-jacent et des capacités de résolution et d'explication d'un système déclaratif.

Trois niveaux de connaissance sont distingués :

- les connaissances de base, qui sont des connaissances *sûres* exprimées sous forme de règles de réécriture,

- les plans, qui correspondent aux théorèmes et aux savoir-faire associés à des problèmes types ; ils comprennent une partie condition et une partie action, composée d'actions dites immédiates faisant éventuellement appel au système de calcul symbolique sous-jacent,
- les métarègles qui permettent une sélection rapide des plans a priori applicables parmi l'ensemble des plans possibles et un classement des plans retenus. L'usage de telles méta-règles est essentiel, à la fois pour le contrôle de la résolution et l'explication des choix suivis par le système.

Une des caractéristiques importantes de CAMELIA est son langage d'expression des connaissances. La partie condition des plans peut contenir des conditions problèmes, c'est-à-dire des conditions qui ont la forme d'une proposition P pour laquelle le système n'effectue pas directement une évaluation mais dispose de connaissances permettant de prouver que P est vraie ou fausse. C'est le cas de problèmes de calcul de primitives où certains changements de variable dépendent de la parité de l'expression considérée. Le système permet d'écrire un plan du type :

si  $f(x)$  est impaire en  $x$ , alors effectuer le changement de variable  $y = \cos x$ .

CAMELIA a été testé sur plusieurs domaines du calcul algébrique : limites, développements limités, primitives, preuve de parité. Il a pu résoudre un grand nombre de problèmes. Le moteur d'inférence utilisé dans CAMELIA sera repris dans d'autres projets présentés au chapitre 5.

Une limite de cette approche, comme celle de SEME, provient du fait qu'on ne donne au système qu'un certain ensemble de règles et qu'on expérimente ses capacités de résolution que sur un catalogue d'exemples. Le domaine de validité est difficile à préciser, on ne peut savoir à l'avance si un nouvel exercice sera ou non résolu par le système. Néanmoins, les différents systèmes précédents ont permis de mieux comprendre les problèmes d'intégration d'expertise dans des systèmes à base de connaissance en vue d'une utilisation à visée éducative.

Dans un registre différent, une équipe de l'université d'Orsay (Bonnet *et al.*, 1981) s'est intéressée à la conception d'un système d'enseignement dans le domaine du calcul des dérivées. L'accent porte plus sur les capacités de dialogue avec l'étudiant offertes par une interface en langue naturelle que sur la réalisation d'un résolveur. Bonnet et ses collègues cherchent à séparer clairement la partie experte et les stratégies d'enseignement. L'architecture qu'ils préconisent inclut trois composantes :

- *un expert* utilisant des règles de production, ayant le niveau de compétence d'un élève de lycée. Cet expert, bien que ses connaissances du domaine soient limitées, est capable de générer et de résoudre des problèmes, d'évaluer des méthodes différentes, de fournir explications, commentaires et critiques sur les méthodes proposées par l'étudiant. Des règles incorrectes sont associées aux règles correctes pour trouver le chemin probablement suivi par un élève,

- *un pédagogue* contient les stratégies d'enseignement et le profil de l'élève. Il doit prendre les décisions sur le type de problème à poser, guider le dialogue avec l'élève durant la résolution du problème et utiliser les réponses de l'élève pour faire des hypothèses sur ses connaissances,
- *une interface en langage naturel*, utilisant une grammaire sémantique.

Un exemple de session fictive

C: Dérivez  $\frac{(x^2 - 1)(x + 5)}{x - 1}$

E: *J'utilise la règle quotient*

C: Vous pouvez trouver mieux

E: *Je ne vois pas*

C: Vous pouvez utiliser la simplification

E:  $(x + 1)(x + 5)$

C: Oui, c'est bien. Et maintenant ?

E: *Je trouve  $2x + 6$*

C: Vous avez trouvé la solution, mais quelle méthode avez-vous utilisée ?

E: *J'utilise la règle du produit.*

C: OK. Nous allons maintenant traiter un autre exercice...

L'architecture proposée est proche de celles en vigueur dans la conception des *tuteurs intelligents*. Il s'agit d'intégrer les performances d'un résolveur de problèmes (qui travaille avec des méthodes théoriquement utilisables par les élèves) à un dialogue de type précepteurat. Ce travail sera continué et aboutira à TANGO, moteur d'inférence conçu par Cordier et Rousset, appliqué au calcul de primitives par Fallier et Pottier. TANGO n'utilise aucune couche algébrique et certaines fonctions comme la reconnaissance de formes particulières telles que  $f^m \times f^n$  se font par l'intermédiaire d'un appel à l'utilisateur (Nicaud, 1987, p. 39).

#### 3.4.3.3. Modélisation du raisonnement mathématique

Dans le cadre de l'approche heuristique, les premiers travaux consistent à recueillir ou à mettre à jour les stratégies, heuristiques, plans, méthodes utilisés par les mathématiciens, à les nommer, à les caractériser. Il s'agit de construire des résolveurs dans divers domaines des mathématiques et, en dehors de préoccupations de conception de systèmes, de faire émerger un corpus de connaissances éventuellement enseignable.

La machine, à qui est dévolue la résolution, joue le rôle de dépositaire des connaissances de l'expert. Le recueil de ces connaissances s'appuie sur l'observation de mathématiciens dans des tâches de résolution (Bundy, 1975 ; Pastre, 1978 ; Pastre 1985). Cette observation, qui est parfois une auto-observation, conduit à la



formulation de connaissances soit inconnues soit éparpillées. Elle permet notamment de mieux comprendre le rôle de la connaissance implicite dans la résolution de problèmes. Pastre (1984, p. 55) tire diverses constatations de l'observation des mathématiciens : l'importance des connaissances accumulées par le mathématicien, et de tout un savoir-faire mathématique, l'importance des représentations, la nécessité de travailler avec un langage non formel dans lequel on peut exprimer des idées, des concepts, des méthodes. Même si le type d'analyse pratiqué ne prétend aucunement à une quelconque validité psychologique (Bundy, 1975), l'éclairage fourni sur la façon de procéder des experts peut efficacement guider la réalisation de résolveurs intégrés à des logiciels d'enseignement.

Parmi les points soulevés par Pastre, on peut noter que l'importance de représentations a déjà été soulignée, notamment par Hadamard et Polya. D'après Simon (1969, p. 135), résoudre un problème, c'est le représenter de façon à rendre sa solution transparente et les informaticiens essayent de tirer parti de représentations non classiques, analogues aux dessins dont s'aide le mathématicien. Par contre, une des difficultés est le choix du langage permettant de communiquer l'expertise à la machine. Pour des résolveurs à visée éducative, l'autre versant du problème est la communication de l'expertise représentée en machine au novice. Il convient donc d'explicitier comment un mathématicien et un novice s'y prennent pour effectuer un calcul, concevoir une démonstration ou plus généralement résoudre un problème. En particulier, quelles sont les connaissances utilisées, comment choisir un fil conducteur (un plan), quand abandonner une voie qui semble vouée à l'échec, etc. ? On tente ainsi de faire des résolveurs qui sont proches du mathématicien tout en tenant compte des difficultés du novice. Le choix du mode de raisonnement est alors déterminant.

En effet, les explications et les traces fournies par la machine donnent une première idée de la solution obtenue et de la recherche de cette solution. Mais, se contenter de présenter la trace d'une solution obtenue apparaît rapidement insuffisant. Comme le remarque Dominique Pastre (1985), si le fait d'imiter les méthodes humaines est une bonne heuristique, il ne suffit cependant pas de savoir que telle méthode va être employée<sup>33</sup> mais *pourquoi* elle va être employée et pourquoi les experts ne l'emploieront pas dans des situations pourtant voisines. La trace accessible ne rend compte que du cheminement du système et n'explique pas le pourquoi des choix effectués. En outre, dans les systèmes experts, les explications éventuellement fournies s'adressent avant tout à l'expert et ne sont pas toujours adaptées à un utilisateur néophyte dans le domaine ou n'ayant pas encore les connaissances suffisantes. On est ainsi conduit à un modèle d'enseignement magistral où l'obtention d'un résultat justifie a posteriori les choix effectués pour y parvenir.

---

33. L'étude préalable du problème permet de prévoir quelle(s) méthode(s) conduit à la solution. Il est raisonnable de penser que les experts vont être amenés à l'utiliser.

L'histoire de GUIDON (Clancey, 1983), système d'enseignement conçu à partir du système expert MYCIN, consacré au diagnostic des maladies infectieuses, illustre bien certaines insuffisances de cette approche. Ce système a une importance historique essentielle. Il a en effet permis de montrer que des systèmes experts, avant tout conçus pour leurs capacités d'expertise, ne sont pas a priori de bons systèmes d'enseignement. Ceci n'est pas lié à des lacunes dans les stratégies d'enseignement mais aux types mêmes de connaissances qui y sont implantées, suffisantes pour les experts mais peu satisfaisantes pour les débutants qui ont besoin de modèles (de type causal) expliquant les diverses règles d'expertise acquises par expérience (Baron, 1984). Afin d'améliorer les capacités d'explication des systèmes, la deuxième génération de systèmes experts va tenter de rendre explicite la stratégie de contrôle et les règles heuristiques et de prendre en compte les relations de causalité.

Deux constats se dégagent : les limites des premiers résolveurs et les difficultés de faire les liens avec les débutants. En conclusion de sa thèse, Pastre (1984, p. 180) soutient qu'il reste encore un énorme travail à faire pour représenter l'*activité mathématique*. Il faut poursuivre l'analyse dans des domaines particuliers en attendant que puissent se dégager des heuristiques générales. La collaboration de mathématiciens est encore nécessaire. L'idéal serait d'avoir un système capable d'*apprendre*, de même que le mathématicien s'est formé petit à petit. Cette direction de recherche va être abondamment explorée.

### 3.5. Apprentissage automatique et découverte

L'apprentissage, pour une machine, consiste à modifier son comportement en fonction de son histoire. Une telle capacité d'apprentissage est indispensable pour qu'une machine puisse être adaptative, nécessité qui, comme nous l'avons vu, a été exprimée dès les années cinquante dans le cadre de la cybernétique. De nombreux travaux d'intelligence artificielle se sont intéressés à ce problème. Citons notamment le programme CHEKER de Samuel (1959) qui améliore ses capacités au jeu de dames. Dans le domaine des mathématiques, l'intégration formelle a suscité un nombre important de recherches sur l'apprentissage automatique (Mitchell *et al.*, 1983).

Concernant les systèmes d'enseignement, de tels travaux permettent d'un côté de modifier les programmes, en les dotant de capacités accrues d'adaptation (guider les résolveurs, améliorer les performances des systèmes de diagnostic et les stratégies d'enseignement des tuteurs) et, d'un autre côté, de simuler certains processus d'apprentissage des humains. Dans cette section, nous allons présenter les premiers programmes d'enseignement qui ont été capables de modifier leur stratégie d'enseignement. Ensuite, nous décrirons brièvement un système de diagnostic, ACM,

et un programme d'apprentissage à partir d'exemples et nous terminerons ce tour d'horizon en examinant les potentialités offertes par les systèmes de découverte.

### 3.5.1. Quelques systèmes d'enseignement « apprenant »

Concevoir un système d'enseignement capable d'améliorer sa stratégie pédagogique au fur et à mesure de son utilisation est un problème qui a été attaqué dès les années soixante. Le programme de géométrie de Smallwood (1962 ; décrit dans O'Shea, 1982b) est le premier tuteur susceptible d'une telle auto-adaptation. Programme classique de type crowdérien (présentation d'un bloc d'informations, question à choix multiples, branchement), son originalité tient à sa capacité de modifier au cours du temps ses décisions de branchement. En fonction des expériences d'enseignement précédentes, le programme cherche à déterminer le « meilleur » bloc à présenter à un étudiant en cours de session. Pour cela, il utilise une mesure des performances d'enseignement basée sur le maintien du nombre attendu d'erreurs de l'apprenant en dessous d'un certain seuil et sur la minimisation du temps prévu pour terminer le cours. Le choix du bloc s'appuie sur un modèle bayésien, mais, pour évaluer des chemins peu, voire jamais, utilisés, il s'effectue aussi de manière aléatoire. Pour un étudiant doté d'un certain niveau de maîtrise, ce choix correspond à celui sélectionné pour d'autres étudiants « passés » ayant parcouru des chemins *similaires* au travers du réseau de branchement. Une des difficultés est de trouver une mesure de similarité des chemins suivis par les étudiants. Le programme de Smallwood se contente d'utiliser le pourcentage de bonnes réponses déjà fournies par l'apprenant durant son parcours.

En l'absence de théorie permettant de guider les décisions de branchement, la méthode définie par Smallwood bute rapidement sur un phénomène d'explosion combinatoire<sup>34</sup>. D'autres programmes vont illustrer l'intérêt pour l'enseignement des capacités d'apprentissage des machines.

#### 3.5.1.1. Le programme d'intégration symbolique de Kimball

INTEGRATION, réalisé en LISP par Kimball en 1972, est une tentative de réaliser un programme « connaissant ce qu'il enseigne », selon le vœu de Siklóssy. Laissant de côté les algorithmes qui ne permettent de traiter les étapes intermédiaires, la recherche d'une primitive s'effectue en appliquant une série de transformations à une expression jusqu'à l'obtention d'une expression de forme connue qui conduit à une réponse immédiate (une primitive connue). Le programme dispose de différentes approches, correspondant à des opérateurs à appliquer à l'état courant.

---

34. En considérant un arbre de décision comprenant cinq branches à chaque nœud et deux réponses possibles de l'étudiant à chaque fois, pour une profondeur 10, il faut envisager près de 10 milliards de trajectoires !

Le programme de Kimball est basé sur l'hypothèse forte que les classifications des problèmes pour les apprenants et le système sont les mêmes. Il utilise une approche probabiliste pour construire un modèle élève afin de comparer ses performances avec un autre élève ou avec le tuteur, choisir des problèmes de remédiation et décrire les effets de l'apprentissage. L'étudiant peut poser un problème ou le tuteur en sélectionner un, adapté à l'élève en fonction de son modèle. Au niveau du résolveur, 130 problèmes ont été archivés avec une solution type associée. Si l'apprenant pose son propre problème ou dévie de la solution stockée, le tuteur répond à une demande d'aide sur la base d'un calcul probabiliste sur les opérateurs applicables. Ce calcul ne garantit pas le caractère approprié du conseil fourni, l'opérateur tenté ne conduisant pas toujours à la solution du problème.

Dès qu'un problème a été résolu, le tuteur compare la solution de l'élève avec la sienne. Si la solution de l'élève est jugée meilleure, le tuteur incorpore la solution de l'élève dans son répertoire. Le seul critère retenu est celui de la longueur d'une solution (nombre de pas). D'après Kimball (1982, p. 304) plus du quart des solutions originales ont été raccourcies et le système a acquis des approches qui n'avaient jamais été utilisées par l'auteur. Plusieurs apprenants ont ainsi contribué à l'amélioration du tuteur. Kimball conclut son article en précisant que cette amélioration par les étudiants montre que parfois les systèmes sont moins performants que ne le pensent leurs concepteurs !

Même si l'expertise de résolution du système de Kimball peut être jugée faible rétrospectivement, l'approche suivie est indubitablement intéressante. Elle montre en effet ce que peut faire un résolveur incomplet, le système d'enseignement jouant partiellement le rôle de partenaire.

### 3.5.1.2. QUADRATIC TUTOR

QUADRATIC TUTOR (O'Shea, 1982a) est un programme d'enseignement adaptatif qui peut modifier sa stratégie d'enseignement. Son domaine est celui de la recherche des racines de certaines équations du second degré de la forme  $X^2 + c = b \times X$  dont les coefficients sont choisis de manière ad hoc, pour que l'apprenant puisse deviner les solutions en utilisant quelques règles simples. En notant  $x_1$  et  $x_2$  les deux racines, dans les exercices proposés, une des deux racines est *évidente* et l'autre se déduit avec une des règles à découvrir :

- $x_1 + x_2 = b$
- $x_1 x_2 = c$  ce qui implique que  $x_1$  et  $x_2$  sont des diviseurs de  $c$
- règle particulière si  $b = c + 1$ ,  $x_1 = 1$  et  $x_2 = c$

On demande à l'apprenant de deviner la solution en appliquant une méthode de découverte. La figure 3.15 présente un exemple de session. La stratégie d'enseignement adoptée consiste à donner à l'élève des exemples soigneusement

choisis pour lui offrir la possibilité de découvrir une des règles. S'il apparaît que l'élève maîtrise une règle, on lui fournit des exemples pour lesquels cette règle n'est pas suffisante ainsi que des exemples pour lesquels elle suffit. Le but n'est pas de le conduire à rejeter cette règle mais à en inventer d'autres.

Le tuteur doit sélectionner les exemples appropriés dans sa base et converser avec l'étudiant, en lui fournissant commentaires et aides et en l'interrompant s'il prend trop de temps. Le tuteur dispose pour cela d'une matrice de difficultés des tâches, d'un modèle élève de type expertise partielle (sur les règles à apprendre) et de stratégies d'enseignement. Quatre buts lui sont assignés : accroître le nombre d'étudiants terminant la session, accroître le score moyen à un post-test, décroître le temps passé par les étudiants et décroître le temps passé par l'ordinateur dans le processus (c'est-à-dire dans le temps de calcul du modèle élève et dans le choix des exercices).

Des paramètres vont varier pour tester différents types d'enseignement. La stratégie d'enseignement est exprimée à l'aide d'un ensemble de règles de production et une composante permet d'effectuer des modifications dans cet ensemble de règles. Pour satisfaire chacun des quatre buts, des actions sont déduites ainsi que les conséquences possibles de ces actions sur les autres paramètres. Les résultats sont analysés par un programme statistique.

Des expérimentations, sur des effectifs assez réduits, donnèrent des résultats jugés encourageants bien que non réellement significatifs. Toutefois, O'Shea suggère qu'une telle approche pourrait être intégrée à des systèmes d'enseignement largement utilisés tels TICCIT et PLATO. Sur la base d'objectifs d'enseignement clairs et d'une théorie de l'instruction explicite, c'est-à-dire exprimée par un ensemble de règles de production et un ensemble d'assertions décrivant l'utilité possible des changements d'actions, il devrait être possible de construire des programmes d'enseignement capables de s'auto-améliorer.

Un tel exemple montre cependant la quantité importante d'expériences à mener pour espérer obtenir des résultats. Pouvoir diminuer le nombre d'expériences nécessaires ou les simuler sur ordinateur peut s'avérer très précieux, d'autres recherches vont s'y consacrer.

<p>Considérez l'équation <math>X^2 + 18 = 9 \times X</math>.  Devinez les solutions  <i>E: Que se passe-t-il si je ne peux pas</i>  SVP, tapez un nombre ou tapez STOP si vous désirez arrêter  <i>E: 1, 2</i>  Non  <i>E: 3</i>  C'est exact, essayez de trouver l'autre réponse  <i>E: 4, 5</i>  Non  <i>E: 6</i>  C'est bien, vous avez trouvé les deux réponses  Considérez l'équation <math>X^2 + 18 = 19 \times X</math>.  Devinez les solutions  <i>E: 3</i>  Non, réessayez  <i>E: 6, 12, 2, 3</i>  Non, réessayez</p>	<p><i>E: 4, 5, 6, 7, 8, 9, 10</i>  Vous n'essayez pas de découvrir une règle pour vous aider  Nous allons regarder un autre exemple.  Essayez de trouver une règle  Les solutions sont en fait 1 et 18  Considérez l'équation <math>X^2 + 48 = 19 \times X</math>.  Devinez les solutions  <i>E: 1, 18, 12, 48, 19, 64, 9, 2</i>  Non  <i>E: 3</i>  C'est exact, essayez de trouver l'autre réponse  <i>E: 6</i>  Non, réessayez  <i>E: 3</i>  Oui mais vous l'aviez déjà devinée correctement  <i>E: 15, 18, 9, 1</i>  Non, cela fait beaucoup d'essais  Essayons un autre exemple  Les bonnes réponses sont en fait 3 et 16.</p>
--	--

**Figure 3.15.** Exemple de session avec QUADRATIC TUTOR (O'Shea, 1982a)

### 3.5.1.3. Apprentissage à partir d'exemples

Modéliser le processus d'apprentissage demeure un objectif de recherche important pour construire des systèmes d'enseignement. Van Lehn (1990), dans ses différentes théories génératives des erreurs, a ainsi essayé de simuler certains processus pour expliquer la genèse d'erreurs constatées auprès d'apprenants. Il s'est par ailleurs intéressé aux conditions, qu'il désigne sous le terme de *conditions de félicité*, pour garantir un apprentissage sans erreur.

D'après Chi et Bassok (1989), les exemples jouent un rôle fondamental dans l'apprentissage. Les bons apprenants apprennent à partir d'exemples en générant des auto-explications. Cela leur permet d'ajouter de la connaissance tacite sur les actions de la solution-exemple et d'induire une compréhension profonde des principes sous-jacents. Guidées par la précision avec laquelle ils contrôlent leurs états de compréhension, de telles explications les poussent à faire des recherches spécifiques pour trouver une réponse alors que les étudiants plus faibles ont des questions peu

dirigées. Chi et Bassok concluent sur la possibilité que des étudiants puissent apprendre, avec compréhension, à partir d'un ou de peu d'exemples<sup>35</sup>.

En l'absence d'explication, si les élèves n'apprennent que des suites d'actions, comme des règles syntaxiques et algorithmiques, ils risquent de ne pas transférer ce qu'ils ont appris à partir des exemples. Ce problème est bien illustré par ALEX, le programme conçu par Neves (1978, doctorat non publié). Ce programme est capable d'apprendre à résoudre des problèmes algébriques simples en déduisant des règles à partir d'étapes consécutives. Par exemple, à partir des pas suivants :

1.  $3X + 4 = 0$
2.  $3X = -4$
3.  $X = -4/3$

Le programme de Neves peut détecter et calculer les différences entre les lignes 1 et 2 et déduire la règle : *s'il y a un nombre dans le membre gauche de l'équation, alors le soustraire des deux côtés*. Une telle règle n'explique pas pourquoi ou dans quelles conditions on doit effectuer une telle soustraction. Cette règle est syntaxique dans le sens où elle s'applique sur la structure d'une équation donnée vérifiant exactement la condition de la règle. Ainsi, dans le cas où la structure courante doit être modifiée pour satisfaire cette condition, les apprenants vont échouer car ils n'ont pas compris comment les conditions de la règle peuvent être dérivées à partir d'autres variations de cette équation.

Si les premiers modèles d'apprentissage automatique servent plutôt à illustrer certaines déficiences dans l'apprentissage humain, ils vont contribuer à la conception de systèmes de diagnostic performant.

#### 3.5.1.4. ACM / DPF et le diagnostic automatique

L'approche diagnostique proposée par Langley et Ohlsson, utilisée dans ACM (*Automatic Cognitive Modeling*, Langley et Ohlsson, 1984) et DPF (*Diagnostic Path Finder*, Ohlsson et Langley, 1988), se rapproche de celle de LMS/PIXIE mais élimine le besoin d'un catalogue de bogues. Elle s'appuie sur des procédures classiques de recherche d'un chemin dans un espace de problème, en leur associant des techniques d'apprentissage automatique.

L'analyse est proche de celle de Newell et Simon. On dispose d'un espace de problème, un état initial, un état final (un critère de terminaison) et un ensemble d'opérateurs : le diagnostic<sup>36</sup> revient à trouver un chemin dans cet espace. La

35. Si cette affirmation apparaît quelque peu discutable, Van Lehn et son équipe travaillent autour d'une telle idée avec des apprenants simulés, notamment pour modéliser le processus d'auto-explication. Nous présenterons quelques résultats dans le chapitre 5.

36. Rappelons qu'il s'agit d'un diagnostic comportemental, c'est-à-dire une reconstruction du cheminement qui a pu être suivi par un apprenant.

méthodologie générale de recherche comporte trois étapes : trouver l'espace de problème (ce qui est souvent loin d'être trivial), chercher un cheminement aboutissant à une solution, c'est-à-dire trouver comment la solution a été engendrée et induire des règles générales de résolution. C'est cette dernière phase qui utilise les techniques de l'apprentissage automatique : généralisation, induction et groupement de pas élémentaires en pas plus importants. Elle suppose d'avoir un certain nombre d'exemples à disposition.

La recherche effectuée dans l'espace de problème peut soit être complète, c'est-à-dire exhaustive, ou partielle, c'est-à-dire sélective. La première paraît plus séduisante, elle est malheureusement impraticable à cause des risques d'explosion combinatoire. Il faut donc se contenter d'une recherche limitée et définir des critères d'évaluation. Ces derniers sont basés sur une idée générale de plausibilité psychologique. Dans DPF, sont distingués des critères absolus qui permettent de rejeter définitivement certaines branches et des critères relatifs influant sur les choix des branches à explorer en priorité. Dans la taxonomie introduite par Ohlsson et Langley, les critères absolus concernent :

- la fermeture causale, c'est-à-dire l'élimination de ce qui ne conserve pas de lien causal avec le problème, le rejet d'hypothèses trop farfelues ;
- l'utilité pour le but poursuivi et donc le rejet des pas superflus ;
- l'absence de duplication, on ne cherche pas à nouveau un résultat déjà obtenu.

Quant aux critères relatifs, ils sont basés sur l'idée de minimiser le chargement de la mémoire et le nombre d'erreurs, de satisfaire le plus de sous-buts, de privilégier les chemins les plus courts et de maximiser le rapport entre le nombre de sous-buts et le nombre d'opérateurs utilisés dans le chemin.

Bien que le test de DPF dans le domaine de la soustraction (Ohlsson et Langley, 1988) ait donné de bons résultats, la généralisation à des domaines plus complexes ne semble pas immédiate. En particulier, la détermination de l'ensemble des opérateurs peut être une forme déguisée d'un embryon de catalogue d'erreurs, et la description des états conditionne ce que l'on peut découvrir. Fournir au programme un ensemble initial d'opérateurs suppose une théorie implicite des types de bogues susceptibles de se manifester. Sans une telle théorie informelle, l'explosion combinatoire de l'espace de recherche ne peut être évitée, laissant le système de diagnostic face à une tâche sans espoir (Evertz et Elsom-Cook, 1990).

Si les techniques d'apprentissage automatique sont utiles pour améliorer les stratégies d'enseignement ou les techniques de diagnostic, elles s'avèrent essentielles pour la construction de solveurs de problèmes mathématiques. Il s'agit d'acquérir de nouveaux théorèmes (utilisables dans les démonstrations ultérieures), de nouvelles conjectures, de nouveaux concepts, de nouveaux exemples et de nouvelles heuristiques ou même de constituer, structurer et faire évoluer des bases de connaissances mathématiques. Mais au-delà de la démonstration de théorèmes dont



la conclusion est donnée, les chercheurs vont s'intéresser aux phénomènes de découverte automatique. L'exemple le plus fameux est AM, programme conçu par Douglas Lenat.

### 3.5.2. AM et la découverte mathématique

Pour Lenat (1982, p. 185), le but de AM est de formuler des définitions et des conjectures intéressantes. A partir de concepts initiaux qui lui sont fournis, AM cherche à créer de nouveaux concepts, à trouver des exemples de ces concepts, à les vérifier et à faire des conjectures. AM débute son travail avec plus d'une centaine de concepts mathématiques. Ceux-ci sont les concepts de base de la théorie des ensembles, auxquels sont ajoutés les listes et les sacs (structures non ordonnées dont les éléments peuvent être répétés), ainsi que les opérations élémentaires de ce domaine. Après l'identification d'un sac contenant  $n$  éléments avec l'entier  $n$ , les concepts créés vont porter sur l'arithmétique élémentaire (par exemple, la notion de nombre premier correspondant aux nombres ayant exactement deux diviseurs).

La théorie est guidée par 243 heuristiques prédéfinies, dont certaines (cf. figure 3.16) donnent des indications sur la qualification ou la quantification d'intérêt sur une notion (Cuppens, 1987). L'ensemble de ces heuristiques constitue une formalisation du concept d'*intéressant* défini par un modèle numérique simple dans lequel la valeur d'un concept est héritée des concepts à partir desquels il est défini.

Ainsi, par exemple, la notion de nombre pair est intéressante car sa frontière est l'ensemble des naturels ; cette frontière est la même que celle des nombres impairs. Les notions de diviseur et de multiplication sont étroitement reliées.

1. Une notion est intéressante s'il y a des conjectures intéressantes sur cette notion.
2. Une notion est intéressante si sa frontière (c'est-à-dire l'ensemble des éléments qui vérifie juste la notion ou la manque de peu) coïncide accidentellement avec une autre notion intéressante.
3. Une notion est intéressante si sa frontière coïncide avec la frontière d'une autre notion très intéressante et très différente.
4. Une notion est intéressante si elle est accidentellement la frontière d'une autre notion très intéressante.
5. Une notion est intéressante si une opération normalement inefficace peut lui être appliquée de manière efficace.
6. Une notion est intéressante si chaque exemple de cette notion vérifie un prédicat rare.
7. Une notion est intéressante si elle est étroitement reliée à une autre notion intéressante.
8. Une notion est sans intérêt si après plusieurs essais peu d'exemples de cette notion ont été trouvés.

9. Si une notion  $x$  possède une propriété intéressante que ne possède pas une de ses spécialisations  $y$ , alors  $x$  et  $y$  sont un peu plus intéressants.
10. Une conjecture sur une notion est intéressante si cette notion est très intéressante.
11. Une conjecture d'existence non constructive est intéressante.
12. Une conjecture d'existence constructive est intéressante si elle est souvent utilisée.

**Figure 3.16.** *Les heuristiques utilisées dans AM pour définir les objets intéressants*

AM montre qu'il est possible de représenter la formation de concepts mathématiques nouveaux et de faire des conjectures en utilisant un processus d'inférence. Ce processus très explosif peut être guidé par des inférences à un niveau méta, qui peut être à son tour guidé par des heuristiques de recherche. En partant d'une base constituée de concepts très généraux, AM a été capable de découvrir beaucoup de concepts arithmétiques intéressants et de faire différentes conjectures<sup>37</sup>. Notons que le système n'utilise pas de démonstrateur pour tenter de confirmer les conjectures ou pour déterminer l'intérêt d'un concept, il reste au niveau des conjectures sans essayer de les démontrer. AM a été généralisé dans EURISKO (Lenat, 1983) dont les performances sont remarquables dans des domaines aussi divers que les jeux, la programmation, les circuits électriques, les heuristiques.

La méthodologie utilisée dans AM <sup>38</sup> rappelle le processus de l'invention mathématique décrit par Hadamard, en particulier l'exploration de nombreuses pistes et la sélection parmi elles de pistes jugées intéressantes. Les expériences menées par Lenat avec AM montrent toutefois qu'au bout d'un certain temps des phénomènes de dégradation apparaissent, le système ne semblant plus en mesure de proposer de nouveaux résultats intéressants.

A la suite d'AM, d'autres systèmes de découverte ont été implantés, notamment sur la théorie des graphes, la théorie des nombres de Conway ou les lois de Képler.

---

37. La notion la plus originale découverte par AM est celle d'entier maximale divisible. Un entier  $n$  est dit maximale divisible s'il a strictement plus de diviseurs que les entiers naturels plus petits.  $\forall p \in \mathbb{N}, p < n \Rightarrow d(p) < d(n)$  où  $d$  est la fonction qui à un entier associe son nombre de diviseurs.

Les premiers sont 1, 2, 4, 6, 12, 24, 36, 48, 60 dont les nombres de diviseurs sont respectivement 1, 2, 3, 4, 6, 8, 9, 10 et 12. Ceci est proche d'un résultat initialement découvert par Ramanujan en 1915 (Cuppens, 1987).

38. Une critique de la méthodologie employée a été formulée par Ritchie et Hanna (1984) concernant les choix initiaux permettant le développement, la réponse à cette critique étant développée dans Lenat et Brown (1984).

En un certain sens, le processus suivi par ces différents systèmes schématise la découverte scientifique (Laublet 1988) :

- découvrir par l'observation d'exemples une propriété particulière,
- conjecturer cette propriété,
- renforcer cette conjecture avec des expériences ou des exemples supplémentaires,
- prouver cette propriété.

Les systèmes de découverte n'ont donné lieu à aucune application directe à l'enseignement. Pourtant, arriver à définir le caractère intéressant d'une notion est certainement essentiel et l'apprentissage humain n'est peut être pas si éloigné de certains processus de découverte. La conception de programmes capables d'aider l'apprenant à mieux diriger ses propres processus de recherche devrait pouvoir s'inspirer des techniques développées dans les systèmes de découverte. Nul doute que des chercheurs vont s'y atteler.

### **3.6. Bilan et perspectives**

Face à l'exigence, pour les systèmes d'enseignement, de connaître ce qu'ils enseignent, l'intelligence artificielle fournit des techniques permettant d'améliorer les systèmes d'EAO, dans ce qui est appelé *IACAI* ou *AI in CAI* (intelligence artificielle dans l'enseignement assisté par ordinateur). L'objectif visé est avant tout celui d'une meilleure adaptation à l'apprenant. Après le rejet des thèses behavioristes, les premières réalisations montrent la nécessité de disposer de nouvelles théories de l'apprentissage et de l'enseignement, souvent inspirées des travaux menés par Piaget. Il s'agit d'engager les apprenants plus activement et de leur fournir d'autres choses à apprendre. Cela requiert de nouvelles méthodologies pour acquérir de multiples types de connaissances, notamment sur la façon de résoudre des problèmes, sur les erreurs des élèves et sur leur origine, sur les stratégies d'enseignement... Ainsi, on tente, avec un certain succès, de recueillir les connaissances des experts, des recherches extensives sur les erreurs produites par les élèves sont entreprises et aux théories prescriptives de l'enseignement se substituent des recherches sur la façon dont les enseignants pratiquent leur métier. Si Brown et Burton (1982) intègrent de manière compilée dans leur système des principes d'intervention discrète, ils suggèrent que ces principes soient fournis de manière déclarative afin d'offrir aux enseignants un environnement dans lequel ils pourraient modifier ou étendre les règles et en constater l'effet sur les élèves.

Peu à peu, les recherches s'orientent vers l'idée de concevoir des tuteurs intelligents, c'est-à-dire des systèmes ayant une triple expertise, celle du domaine à enseigner, celle de l'enseignement et celle des compétences et connaissances, correctes et erronées, des élèves. Toutefois, les premiers systèmes n'arrivent pas à prendre en compte simultanément tous ces aspects. Sleeman et Brown (op. cit. p. 3)

soulignent ainsi que les auteurs ne sont pas satisfaits des performances globales de leurs systèmes. Quatre points particulièrement importants et fortement reliés entre eux sont mis en évidence :

1. Le matériel de cours en réponse à une interrogation ou à une erreur de l'étudiant est souvent à un mauvais niveau de détail, le système supposant trop ou pas assez de connaissance de l'étudiant.

2. Le système suppose une certaine conceptualisation du domaine, forçant la performance de l'apprenant dans ce cadre conceptuel. Aucun système ne peut encore découvrir et travailler avec la propre conceptualisation de l'apprenant et diagnostiquer ses erreurs sur cette base.

3. Les stratégies d'enseignement reposent sur des intuitions non basées sur des principes explicites de contrôle d'un comportement. Il faudrait de meilleures théories de l'apprentissage ou du mauvais apprentissage, c'est-à-dire des théories psychologiques de la représentation des connaissances et de la révision des croyances.

4. L'interaction usager est encore trop restrictive, limitant l'expressivité de l'apprenant et par contre coup limitant les capacités de diagnostic du système.

En fait, les résultats ne sont pas si décevants et les recherches montrent souvent que la complexité des problèmes à traiter est beaucoup plus importante qu'on pouvait le croire au départ. Les leçons apprises grâce aux premiers exemples développés orientent les recherches ultérieures.

Le travail sur les modèles de l'apprenant conduit à considérer que modéliser la compréhension d'un apprenant sur un sujet est conceptuellement étroitement relié au fait de trouver une représentation du sujet lui-même ou du langage utilisé pour en parler, d'où la nécessité de développer des modèles psychologiques non seulement descriptifs mais aussi computationnels.

Le problème central de l'interaction avec l'apprenant est bien identifié. Si plusieurs systèmes font le choix d'une interface en langue naturelle, le travail avec les solveurs de problèmes montre l'intérêt de choisir quelques primitives puissantes, adaptées à la tâche à réaliser. Le développement des interfaces graphiques va proposer d'autres pistes.

Le travail sur la conception de solveurs intégrés à des systèmes d'enseignement révèle différents problèmes intéressants. Tout d'abord, il se fonde, du point de vue éducatif, sur l'idée d'un possible transfert de compétence entre un expert et un débutant par l'intermédiaire d'une représentation en machine. Or, ce transfert apparaît rapidement plus délicat qu'il n'y paraît. Le travail autour de GUIDON montre les limites d'une telle approche, notamment la nécessité d'ajouter des explications de nature causale, c'est-à-dire de justifier les règles livrées par les experts. Ensuite, la nécessaire prise en compte du niveau de connaissances des apprenants conduit à concilier des contraintes souvent antagonistes, l'effectivité dans

la résolution et l'assistance à un apprenant pour faire la résolution. Contraindre un résolveur automatique à n'utiliser que des méthodes connues des débutants ne lui assure pas la certitude d'arriver à une solution. De plus, pour être compréhensibles par les apprenants, on exige des systèmes une justification toujours plus complète des divers choix effectués. De résolveurs fournissant uniquement la trace d'une solution obtenue, c'est-à-dire *la bonne solution* de l'expert, on passe à des systèmes expliquant leurs choix (par la trace de métarègles décrites d'une manière explicite) puis à des systèmes devant rendre compte de l'ensemble des choix possibles en cours de résolution et non plus seulement a posteriori, une fois une solution obtenue.

L'apport de capacités de raisonnement à des machines à enseigner conduit peu à peu à modifier leur statut et les rôles possibles. Simple support de théories générales de l'enseignement ou de l'apprentissage, la machine n'est qu'un médium ou un outil commode. Elle devient ensuite un support pour de nouvelles idées, de point d'arrivée des théories, elle en devient le point de départ. En effet, l'ordinateur s'affirme, en quelque sorte, comme le *lieu* de l'activité de l'apprenant. Alors que les recherches se consacrent à la conception de précepteurs artificiels, censés enseigner à un apprenant unique, les systèmes réellement utilisés en formation, tels EXCHECK ou SOPHIE sont des environnements réactifs. Ils interviennent plutôt comme des partenaires de résolution, offrant une assistance sur le travail en cours, que comme des enseignants prescrivant en pas à pas le travail devant être effectué par l'apprenant. La perspective de concevoir un tuteur intelligent, offrant un enseignement réellement individualisé, nourrit les recherches. Pourtant, dès la fin des années soixante-dix, Sleeman et Brown (1982, p. 9), sur la base de leurs différents travaux, défendent l'intérêt du travail coopératif. Durant les années quatre-vingt, les recherches en EIAO vont prendre en compte de manière importante cet aspect coopératif dans l'apprentissage, comme nous le verrons au chapitre 5.