

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

Chapitre 5

Des tuteurs intelligents aux environnements interactifs

5.1. Introduction

L'usage des techniques et des langages d'intelligence artificielle au début des années soixante-dix conduit à l'émergence de deux écoles de pensée : celle attachée aux environnements exploratoires, exposée au chapitre précédent et celle intéressée à la conception de tuteurs automatisés dits intelligents, décrite dans le chapitre III. Dans un premier temps, cette dernière voie constitue une réponse à la trop grande combinatoire imposée par l'EAO, c'est-à-dire à la nécessité de prévoir à l'avance chaque test, chaque décision, chaque branchement à une ressource de remédiation ou d'exposition. Pour sortir de cette impasse, doter l'ordinateur de connaissances et de capacités de raisonnement donne des résultats encourageants. Dans un second temps, de nouvelles théories de l'apprentissage sont peu à peu prises en compte dans les réalisations informatiques.

Dans la tradition d'une certaine vision de l'intelligence artificielle, réaliser un tuteur intelligent est une tentative pour concevoir un programme tournant sur un ordinateur capable de se conduire d'une telle façon qu'elle serait jugée comme un *bon enseignement* si elle était faite par une personne humaine. Cet objectif apparaît assez rapidement difficile à atteindre. En effet, comme nous l'avons indiqué en conclusion du chapitre III, les auteurs ne sont pas satisfaits des performances globales de leurs systèmes. S'agissant de concevoir des programmes ayant une triple expertise, celle du domaine à enseigner, celle de l'enseignement et celle de l'analyse des compétences et connaissances correctes et erronées des élèves, les premières réalisations n'arrivent pas à prendre en compte simultanément tous ces aspects. Les chercheurs sont amenés à décrire les architectures souhaitées à partir de modèles généraux. Le réexamen des théories de l'apprentissage conduit en outre à concevoir des architectures fondées sur de nouvelles théories, voire même destinées à les valider.

Les modèles proposés ne s'avèrent pas totalement satisfaisants et, en tous cas, peu adaptés à la réalisation effective de tuteurs intelligents. Héritage des recherches sur les micromondes et les environnements d'apprentissage ouverts, le rôle central joué par l'interface, définissant réellement le cadre de travail de l'apprenant, est de plus en plus reconnu. La nécessité d'apporter un soutien à l'apprenant, du simple conseil au guidage en pas-à-pas, est aussi mise en lumière. On espère que l'ajout d'un tutorat intelligent aux environnements ouverts va permettre de transformer plus facilement les conceptions erronées de l'apprenant en expériences constructives d'apprentissages. Ainsi, opérer une synthèse entre micromondes et tuteurs devient souhaitable, pour concilier, comme le défendent Sleeman et Brown (1982, p. 3), l'expérience de résolution de problèmes et la motivation de l'apprentissage par la découverte avec le guidage des interactions tutorielles.

Pourtant, ces objectifs sont en partie contradictoires car, pour bien guider, il faut contraindre les chemins suivis par l'apprenant et les exercices à des réponses et erreurs vraisemblables qui peuvent être complètement spécifiées à l'avance. Pour dépasser ces limitations, le système doit avoir sa propre expertise de résolution de problèmes, son propre diagnostic ou capacité à modéliser l'étudiant et des compétences propres dans la génération d'explications. Il faut savoir *quand* interrompre l'activité de résolution d'un apprenant, *quoi* lui dire et *comment* lui dire (ou plus généralement *quoi faire* et *comment le faire*). L'intelligence artificielle, dont les réalisations sont prometteuses, semble en mesure de fournir les réponses en offrant des techniques avancées au service de nouveaux paradigmes d'apprentissage.

Des querelles de désignation (Nwana 1990) montrent que ce champ a de nombreuses branches. On parle de *Systèmes Tuteurs Intelligents* (ITS en anglais) ou d'*Enseignement Intelligemment Assisté par Ordinateur* (EIAO, ICAI en anglais) pour des raisons historiques. On trouve aussi des *systèmes tutoriels à base de connaissance*, *systèmes tutoriels adaptatifs* (Streitz 1988), ou des *systèmes de communication de connaissance* (Wenger 1987). Les tentatives de synthèse avec les environnements exploratoires amènent de nouvelles dénominations. On parle d'*Environnement d'Apprentissage Interactif* (ILE en anglais), d'*Environnement d'Apprentissage Adaptatif*, d'*Environnement d'Apprentissage Situé* (SLE, situated learning environments) ou d'*Environnement Interactif d'Apprentissage avec Ordinateur* (EIAO, sigle rebaptisé, Baron 1991). On pourrait citer bien d'autres dénominations défendues par différents auteurs. En fait, ces distinctions de nom ne sont pas simplement des querelles byzantines autour de recherches très voisines. Elles révèlent les points de vue adoptés vis-à-vis de la place et du rôle des ordinateurs dans des processus d'apprentissage et d'enseignement, ainsi que des prises de position implicites ou explicites sur la nature de la connaissance et la façon de l'acquérir.

Ce chapitre, partant des modèles généraux de tuteurs intelligents, retrace l'évolution du milieu des années quatre-vingt jusqu'au début des années quatre-vingt-dix, à la fois continuation des problématiques initiées dans les années soixante-dix issues de l'enseignement programmé, synthèse avec les environnements d'apprentissage ouverts et ouverture à de nouveaux rôles dévolus à l'ordinateur. Les modifications intervenant dans la conception des interfaces homme-machine et leurs

implications sur la réalisation des tuteurs intelligents seront particulièrement importantes.

5.2. Les tuteurs intelligents

Si les premiers modèles généraux décrivant l'architecture des tuteurs intelligents datent des débuts de l'intégration des techniques d'intelligence artificielle à l'EAO, ils vont être popularisés au moment où ce domaine va s'affirmer au milieu des années quatre-vingt. Tentatives pour fonder la réalisation de systèmes répondant aux critères exigés pour être considérés comme intelligents, à l'exception de modèles fondés sur des outils déjà développés, il ne vont pas s'avérer d'une très grande utilité, sauf pour diffuser largement les idées générales de ce champ de recherche. Nous allons rapidement décrire quelques exemples en détaillant les tuteurs conçus par Anderson et ses collègues, fondés sur la base de sa théorie ACT*.

5.2.1. Modèles généraux

On trouve une description générale de l'architecture d'un tuteur intelligent dans de très nombreux articles de synthèse. On peut citer, entre autres Sleeman et Brown (1982), Ross (1987), Yazdani (1987), Wenger (1987), Burns et Capps (1988), Nicaud et Vivet (1988). L'idée générale tourne autour des 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). Hartley et Sleeman (1973) ont les premiers spécifié les divers ingrédients d'un tuteur intelligent : connaissance du domaine, connaissance de la personne qui reçoit l'enseignement (modèle élève), connaissance des stratégies d'enseignement, connaissance de la façon d'appliquer la connaissance des stratégies d'enseignement aux besoins d'une personne.

La conception dominante organise les tuteurs intelligents autour de quatre modules : (1) le domaine, (2) le modèle élève, (3) l'expert pédagogue et (4) l'interface.

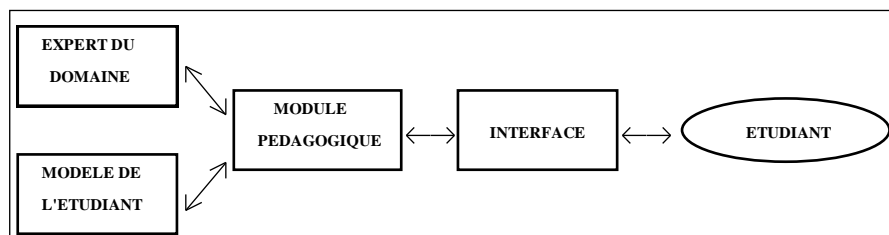


Figure 5.1. *Modèle général d'un tuteur intelligent*

Divers autres modèles ont été proposés, s'écartant de manière plus ou moins importante du modèle général précédent. O'Shea (O'Shea et al., 1984) décrit un

modèle constitué de 5 anneaux : (1) histoire de l'élève, (2) modèle de l'élève, (3) un ensemble de stratégies d'enseignement, (4) un générateur d'exercices qui orchestre l'information des trois modules précédents pour produire les sorties, (5) un administrateur responsable du contrôle général. Dans cette architecture, la connaissance du domaine n'est pas explicitement représentée. Par ailleurs, il n'y a pas de module spécifique lié à l'interface ou l'environnement.

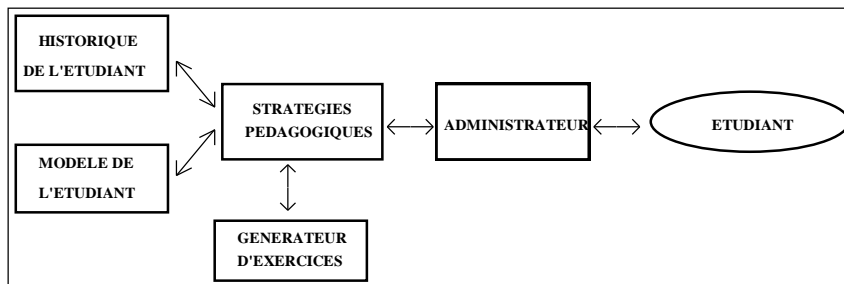


Figure 5.2. *Modèle de O'Shea (1984)*

Comme nous le verrons, Anderson, concevant des tuteurs sur la base de sa théorie ACT*, exclut le modèle de l'élève et le remplace par un catalogue des erreurs possibles et un modèle de l'élève « idéal ». Wenger (1987, pp.13-25) reprend le modèle traditionnel dans une perspective de communication : le domaine est l'objet de la communication, le modèle élève caractérise le destinataire de la communication, le modèle pédagogique gère l'aptitude à la communication, l'interface correspond à la forme de la communication.

Burns et Capps (1988) précisent certaines caractéristiques des différents modules : un module expert sur le domaine, un module de diagnostic de l'élève, un module gérant le curriculum et l'instruction, un environnement d'instruction et une interface homme-machine. Ce module de gestion du curriculum et de l'instruction :

- assure le contrôle de la représentation de la connaissance pédagogique pour choisir et séquencer la matière,
- peut répondre aux questions de l'élève sur les buts de l'enseignement et sur son contenu,
- intègre des stratégies pour déterminer quand les élèves ont besoin d'assistance et fournir l'aide appropriée.

Les divers modules peuvent être de plus répartis en différents sous-modules dont le découpage n'est pas toujours cohérent avec l'organisation standard. Le modèle classique correspond souvent peu aux implantations réelles et les concepteurs d'architectures génériques reconnaissent leur intérêt limité. Ainsi, Self (1988) souligne que la division d'un tuteur intelligent en modules domaine, élève et tuteur est plus une forme d'explication qu'un guide pour l'implantation. D'un autre côté, on admet des définitions plus fonctionnelles, évitant les interprétations abusives de développement. Miller (1988) décrit ainsi un système tuteur intelligent comme la

conjonction de trois éléments : (1) un instrument éducatif, (2) une interface à un programme d'application et (3) un système basé sur la connaissance.

Toutefois, des architectures moins génériques, fondées sur des éléments effectivement réalisés et testés, peuvent constituer une base de travail fort utile pour développer des tuteurs intelligents. Ainsi, AMALIA (figure 5.3 ; Vivet, 1988), conçu comme une extension de CAMELIA (voir chapitre III), repose sur la coopération de deux systèmes à base de connaissances, utilisant des plans et des métarègles, le premier concernant l'expertise du domaine, le second l'expertise pédagogique. Les deux systèmes s'appuient sur le même moteur d'inférence gestionnaire de plans, KEPLER (figure 5.4). AMALIA repose sur trois idées importantes :

- l'usage d'un langage de représentation des connaissances suffisamment riche pour travailler en profondeur les explications fournies à l'élève sur la résolution d'un problème ;
- l'usage de métarègles pour le contrôle de la résolution effectuée par le système et par l'élève, notamment pour le choix des connaissances à utiliser, à favoriser ou à expliciter ;
- la représentation des connaissances pédagogiques sous forme de plans, des métarègles permettant de contrôler le déroulement de la session, c'est-à-dire d'effectuer les choix des plans pédagogiques et leur éventuel changement en cours de route.

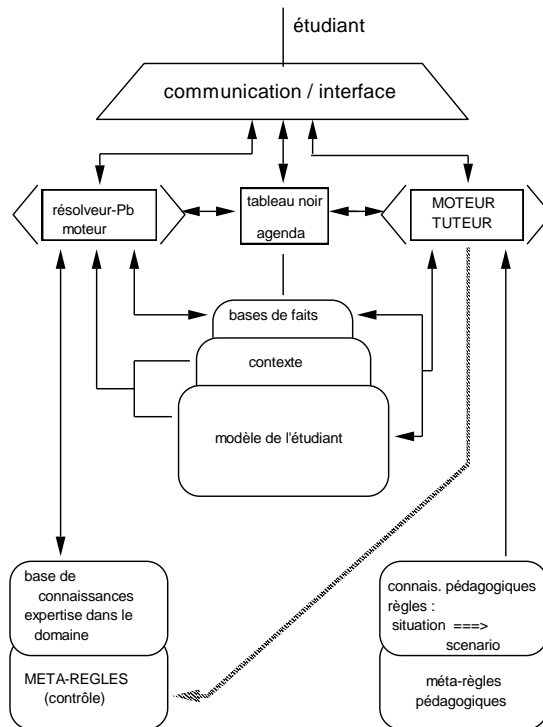
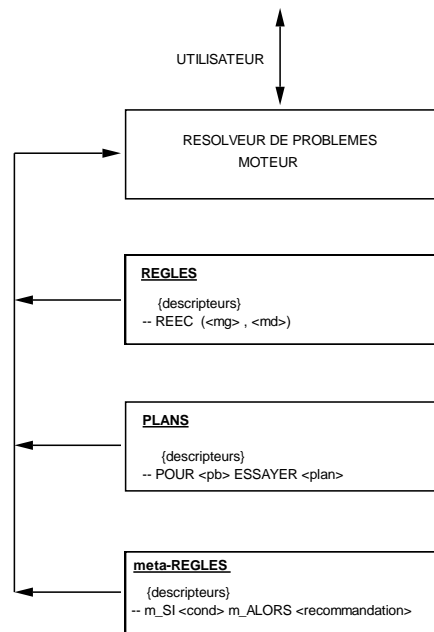


Figure 5.3. *Architecture d'AMALIA*

Comme le souligne Delozanne (1992, p.39), si AMALIA n'a jamais été complètement implanté, l'idée de faire coopérer deux systèmes experts a conduit à l'élaboration d'une architecture multi-agents (Futtersack, 1990) et la mise en œuvre de plans et métarègles pédagogiques interprétés par le moteur KEPLER a été testée à la fois dans QUIZ, tuteur enseignant les stratégies d'enchères au bridge (Labat et Futtersack, 1990) et dans DIADÈME, système d'évaluation interactive (Jacobini, 1993). Par ailleurs, dans le domaine du calcul algébrique, le travail autour des explications a été développé dans ELISE (Delozanne, op. cit.) et dans CAMELEON (Bruillard, 1991).

**Figure 5.4.** *Architecture de KEPLER*

Une analyse rétrospective sur le projet AMALIA (Bruillard et Vivet, 1994) amène à réfléchir sur les compétences nécessaires à l'emploi des coquilles génériques pour la réalisation des applications. Très souvent, ces coquilles sous-tendent un modèle très abstrait que peu de personnes, hormis les concepteurs, arrivent à comprendre et maîtriser. La traduction dans de tels modèles n'est pas toujours aisée. Dans le projet AMALIA, la faiblesse du modèle n'est pas intrinsèque mais liée à la complexité de son instanciation dans des situations d'usage. En effet, l'explicitation de l'expertise pédagogique (ici sous forme de plans) s'avère ardue. L'écriture des plans est délicate du fait de la difficulté à les relier aux situations d'usage associées. La création d'une situation pédagogique, avec explicitation de

l'activité « souhaitée » de l'apprenant, est un préalable à l'élaboration de plans pédagogiques.

Au début des années quatre-vingt-dix, les chercheurs prennent de la distance vis-à-vis des divers modèles généraux d'architecture de tuteurs intelligents. En effet, ces modèles, qu'Elsom-Cook qualifie du type *flèches et cadres*, censés être indépendants des domaines d'enseignement concernés, apparaissent inadéquats. Ils fournissent une description structurelle statique (précisant les différents composants et leurs connexions) qui rend mal compte de l'aspect hautement interactif recherché. Leur champ d'applicabilité semble délicat à délimiter puisqu'aucun tuteur construit sur cette base n'est réellement opérationnel. Les travaux se focalisent soit sur un des aspects (domaine, élève, tuteur), soit sur la globalité en restreignant les contraintes, mais seules les architectures *papier* répondent aux exigences formulées.

On retrouve là des problèmes assez analogues à ceux auxquels se sont heurtés l'enseignement programmé, puis l'EAO. L'établissement d'une forme générique, que l'on tente d'instancier au mieux, est plus un problème d'ingénierie que de recherche. Les divers langages auteurs proposés en EAO n'offrent pas plus de possibilités qu'un langage de programmation traditionnel. Ils ne fournissent aucune base théorique de développement, mais facilitent la définition des affichages écrans et des interactions, c'est-à-dire étendent le nombre de concepteurs potentiels en simplifiant certains aspects, de nature essentiellement technique, du développement des produits. La conception d'architectures génériques de tuteurs intelligents semble procéder de la même démarche. Le recours à des modèles généraux associés à des outils de développement dédiés (dans des démarches de type génie logiciel ou génie didacticiel) permet de réaliser, à moindres frais, non pas des systèmes tuteurs intelligents, mais des systèmes d'EAO avancés. Concevoir un *tuteur intelligent*, avec toutes ses composantes, nécessite des efforts importants et reste une tâche loin d'être triviale. Elle l'est d'autant moins que de nombreux obstacles subsistent et qu'aucun modèle n'arrive réellement à s'imposer¹.

Toutefois, pour conduire des expérimentations auprès d'apprenants et valider certaines architectures spécifiques, il est nécessaire de disposer de produits suffisamment avancés fondés sur des hypothèses et des théories explicites et traitant de domaines d'enseignement particuliers. C'est la direction de travail suivie par Anderson et ses collègues, intéressante de par l'ancrage de leurs tuteurs dans des théories psychologiques, la volonté de confrontation avec des usages réels et la recherche d'interfaces adaptées.

5.2.2. La théorie ACT* et les tuteurs associés

¹ Cette absence de consensus sur des modèles est aussi un signe d'une évolution du champ des applications de l'informatique à l'éducation et une reconnaissance de la complexité et de la multiplicité des approches possibles et de leur dépendance aux domaines traités et aux contextes d'usage.

Anderson est avant tout un psychologue et l'objectif qu'il poursuit dans l'implémentation de systèmes tutoriels est d'explorer ses théories sur l'apprentissage et la compréhension. On retrouve ici la démarche suivie par les premiers tenants de l'enseignement programmé, c'est-à-dire des réalisations réellement fondées sur des théories psychologiques de l'apprentissage. Trois domaines ont conduit à des réalisations importantes : l'apprentissage du langage LISP avec LISP-TUTOR (Reiser et al., 1985), la géométrie élémentaire avec GEOMETRY TUTOR (Anderson et al., 1985) et les manipulations algébriques avec TEACHER'S APPRENTICE (Lewis et al., 1987 ; Anderson, 1992).

La théorie ACT ou ACT* (*Active Control of Thought*) traite de l'acquisition d'habiletés (*skills*) à l'aide d'un ensemble de mécanismes. La résolution de problèmes par les humains est modélisée par un ensemble de règles de production et la théorie décrit comment ces règles sont apprises et exécutées dans le processus de résolution des problèmes. Anderson (1987, p.457) fait le lien avec les théories behavioristes. Il attribue le rejet de la théorie S-R (*stimulus-réponse*) au fait qu'elle impliquait que la connaissance n'était rien d'autre que l'accumulation de liens S-R et que le nombre de tels liens, nécessaires pour produire un comportement intéressant, était astronomique. Pour lui, les systèmes de production fournissent un cadre plus puissant du point de vue computationnel que la théorie S-R. Selon la théorie ACT, la connaissance (ou tout au moins les habiletés de résolution de problème) n'est que l'accumulation de règles de production, position qui renouvelle, tout en étant finalement pas si éloignée, les approches behavioristes².

Anderson (1984) décrit les mécanismes d'apprentissage et leur application à l'acquisition d'aptitudes à la démonstration en géométrie. Le point essentiel est la *compilation des connaissances*, c'est-à-dire le processus qui permet de passer d'une représentation déclarative à une représentation procédurale, processus graduel intervenant à travers la pratique. Les apprenants codent d'abord l'information de manière déclarative (ACT utilise un réseau sémantique). Des procédures d'interprétation générales permettent d'utiliser cette information suivant certains critères dans une situation spécifique. Lorsque cette connaissance déclarative a été utilisée de multiples fois d'une manière particulière, des processus d'apprentissage automatiques commencent à créer de nouvelles procédures permettant d'appliquer directement la connaissance sans recourir à la phase d'interprétation (les apprenants associent certaines caractéristiques des problèmes comme prédictifs du succès des opérateurs). Ce qu'Anderson appelle la *compilation procédurale* (passage de la connaissance déclarative à un ensemble d'opérateurs exécutant des tâches spécifiques) se compose de deux processus principaux : la *composition* ou définition d'une nouvelle règle combinant les effets de plusieurs règles et la *procéduralisation* construction des versions spécialisées des règles de production en éliminant la recherche d'informations dans la mémoire à long terme. Il distingue en outre quatre manières pour effectuer de meilleurs choix dans un arbre de recherche : l'*analogie*, la *généralisation*, la *discrimination* (restreindre le champ d'application d'opérateurs

² Si la barrière de complexité à laquelle s'est heurté l'enseignement programmé est liée aux limitations des théories scientifiques sur l'apprentissage, des formes de tradition demeurent.

généraux) et la *composition* (remplacer une suite d'opérateurs par un opérateur unique). D'après Anderson, cette théorie se rapproche de celles développées sur l'expertise aux échecs, les experts ont encodé beaucoup de règles s'appliquant à des cas spécifiques.

Sur la base de la théorie ACT, l'objectif d'Anderson (1987) est de développer des tuteurs conduisant à des applications effectives, ce qui suppose :

1. une interprétation fine de l'état cognitif courant de l'apprenant,
2. une définition précise de l'état cognitif désiré,
3. une théorie de l'apprentissage décrivant de quelle façon l'expérience transforme l'état cognitif de l'apprenant.

L'enseignement s'appuie sur la méthodologie dite du traçage de modèle (*model tracing*) consistant à simuler dynamiquement, à l'aide d'un système de production, la résolution effectuée par l'apprenant et à utiliser cette simulation pour interpréter son comportement. Ceci implique de modéliser convenablement l'apprenant à deux niveaux, ses comportements d'entrée-sortie et ses productions, ces dernières devant correspondre aux pas de résolution effectivement faits. Il faut surveiller et contrôler (*monitor*³) l'apprenant à chaque production en supposant qu'il y a une correspondance étroite (quasiment une bijection) entre les unités du modèle interne du système, les comportements observables de l'élève et ses étapes de résolution. Ainsi, le modèle cognitif doit rendre compte de la façon dont un apprenant peut (devrait) résoudre des problèmes et est utilisé pour interpréter le comportement des apprenants et les guider à travers le curriculum. Il représente l'idéal de ce que l'étudiant doit accomplir. Ce modèle doit produire toutes les solutions acceptables pour un problème, mais aussi être complété par quelques erreurs caractéristiques relevées par observation des apprenants.

Le style tutoriel est très directif puisque l'étudiant reçoit une rétroaction dès qu'une erreur est détectée ou plus généralement dès qu'il entre une information s'éloignant de la solution correcte connue du tuteur. Cette approche simplifie le processus de diagnostic puisque le tuteur connaît à tout moment l'état de la solution et le but courant. En outre, l'existence même d'un tel modèle réduit considérablement la complexité du diagnostic en identifiant les comportements observables et les états mentaux de l'apprenant.

La clé de la méthodologie du traçage de modèle est la possibilité d'interagir avec l'apprenant en pas à pas au cours de la résolution et d'interpréter son comportement en termes de règles cognitives. Pour Anderson (1992, p.3), le succès provient de trois facteurs. Le premier et le plus important est la création d'un modèle cognitif

³ Le *monitoring* en anglais est distinct du *tutoring*. Ce dernier correspond à l'ensemble des décisions prises et des actions exercées par les intervenants formateurs au sein d'un environnement d'apprentissage dans le but d'agir sur les conditions de travail de l'apprenant pour optimiser son apprentissage. Le *monitoring* consiste en une procédure d'observation, de recueil, de traitement et de mise en forme d'information en provenance de l'environnement d'apprentissage, sur les caractéristiques et les comportements des différents éléments constitutifs, utiles aux différents niveaux de régulation du système évaluatif.

adéquat et opérationnel et la communication de ce modèle à l'apprenant (même par simple communication verbale). Le deuxième réside dans la minimisation du temps d'apprentissage en protégeant l'apprenant du potentiel coût dévastateur des erreurs. Enfin, le troisième est le contrôle et la vérification (*monitor*) de l'acquisition des connaissances (règles individuelles) et l'encouragement ou la promotion des apprenants ayant maîtrisé ces règles. Il semble que les tuteurs conçus par Anderson et son équipe accélèrent considérablement le temps d'apprentissage d'un contenu d'enseignement en optimisant le processus d'apprentissage à travers l'individualisation (Anderson, *ibid.*, p.4, parle d'un facteur 3).

La méthode préconisée par Anderson repose également sur la conception d'une interface adaptée. L'objectif est de rendre les activités donnant les informations intermédiaires au moins aussi faciles et naturelles que la verbalisation. En effet, certains comportements ne sont pas interprétables dans les termes du modèle élève (dans 20% des cas selon Anderson, 1987 p.439). Il est alors nécessaire d'interrompre l'étudiant pour continuer le traçage de modèle. Dans 80% des cas, on communique à l'apprenant son erreur et on l'encourage à reprendre un chemin correct. Lewis (Lewis et al. 1987) décrit ainsi les contraintes que doit respecter une interface d'un programme d'EIAO :

- Etre aussi facile que possible à utiliser, c'est-à-dire minimiser le nombre d'actions nécessaires pour communiquer avec le tuteur.
- Comporter une structure ou représentation aussi congruente que possible avec la structure sous-jacente des problèmes à résoudre.
- Etre hautement interactive et fournir le plus d'informations possibles sur les étapes intermédiaires de résolution.
- Pouvoir pointer les erreurs de bas niveau quand elles arrivent (contrôle ou '*monitoring*' continu)
- Pouvoir faire varier la charge de la mémoire de travail en donnant accès aux informations reliées au problème.

Les tuteurs construits par Anderson et son équipe ont été expérimentés dans des conditions réelles. S'il semble que LISP-TUTOR (Corbett et Anderson, 1991) et GEOMETRY TUTOR (Schofield et Evans-Rhodes, 1989 ; Anderson et al., 1990) ont donné des résultats encourageants, l'intégration dans les classes du tuteur algébrique s'est heurtée en revanche à un obstacle imprévu (Anderson, 1992). Il s'agit d'un conflit entre les techniques de manipulation construites dans le tuteur et les techniques que les enseignants voulaient utiliser. A partir de cette constatation, Anderson et son équipe se sont centrés plus sur les conceptions qu'ont les enseignants des diverses compétences à acquérir que sur les leurs propres, en essayant d'introduire des outils de résolution de problèmes puissants (pour intégrer cette contrainte). Les enseignants servent d'experts et les ingénieurs de la connaissance tentent de coder cette expertise. On peut remarquer là une transition entre une approche scientifique hors contexte scolaire et une approche d'intégration éducative filtrée par les enseignants.

Les objections formulées par Guin (1990 ; 1991) sur GEOMETRY TUTOR, quant à ses possibilités réelles d'usage dans des situations scolaires, donnent un éclairage contrasté. Elle relève, entre autres, des insuffisances dans les expertises

mathématiques et pédagogiques du logiciel (imprécision dans les règles et dans les énoncés, figures très fréquemment particulières, formalisme excessif, choix des exercices et des types d'aide très discutables, etc.), peut-être dues au fait que les fichiers d'exercices ont été constitués par un seul enseignant. Mises à part les exigences sans doute différentes en France et aux Etats-Unis quant à l'apprentissage de la démonstration en géométrie, il semble, en fait, que le processus de réalisation d'un produit réellement utilisable en classe n'a pas été mené à terme, contrairement aux affirmations des auteurs.

A la suite de GEOMETRY TUTOR, la conception de ANGLE (Koedinger et Anderson, 1990) s'est fondée sur une analyse plus profonde de l'expertise de démonstration en géométrie, confirmant la nécessité des analyses épistémologiques et didactiques préalables à la conception d'un tuteur. Dans ce domaine difficile qu'est l'apprentissage de la démonstration en géométrie, il faut souligner la richesse et la qualité des travaux des chercheurs français, tant dans la conception de solveurs (MENTONIEZH, Py, 1990, 1996 ; GEOMUS; Bazin, 1993 ; UBL, Pintado, 1994) et d'environnements ouverts (CHYPRE; Bernat, 1996) que dans l'analyse des situations scolaires (notamment les travaux autour de DEFI, Almouloud et Giorgiutti, 1993).

5.3. Interface et module environnement

Nous venons de voir que, dans la perspective suivie par Anderson, l'interface joue un rôle déterminant, à la fois pour des raisons pratiques, afin d'aboutir à un usage effectif, et théoriques, puisque sa conception est étroitement associée à la méthodologie du traçage de modèle. L'interface avait été plutôt négligée dans les diverses architectures proposées pour les tuteurs intelligents. Différentes raisons permettent de l'expliquer :

- Les interfaces graphiques sont assez récentes, les premiers développements de l'EIAO n'ont pu les prendre en compte.
- La conception d'une interface est un travail long et difficile. Donadi (Donadi et al., 1989) rapporte que la refonte de l'interface de Wordstar entreprise pour produire Wordstar 2000 a nécessité un effort quasiment équivalent à l'écriture d'un programme entièrement nouveau. Le travail requis pour passer d'une maquette à un produit ne se justifie donc que pour une expérimentation ou une diffusion.
- Une conception erronée considère l'interface comme transparente dans le sens où elle offre une plus grande simplicité d'accès sans changer les caractéristiques essentielles d'un programme.
- L'orientation même des recherches prenant l'enseignant humain comme modèle bute sur l'ensemble des phénomènes que la machine ne peut prendre en compte et néglige de considérer les spécificités de ce nouveau médium.

La tendance s'est peu à peu inversée, le travail sur l'interface jouant un rôle prépondérant. Les réalisations concrètes prennent en compte l'utilisateur de manière importante, non comme un être abstrait idéal, mais comme un des éléments centraux du système. On essaye ainsi d'adapter la technologie aux contraintes de cet

utilisateur et non à partir de l'imitation d'autres modes de communication. Dans le cadre de l'EIAO, l'interface est centrale dans le sens où elle est l'unique mode d'accès à l'environnement et le module environnement “*définit les types de problèmes que l'élève a à résoudre et les outils dont il dispose pour le faire*” (Burton, 1988).

Avant d'aborder certaines caractéristiques des interfaces actuelles, il est bon de se remémorer quelques étapes dans leur développement. Walker (1990) décrit succinctement les différentes générations de machines, les modes opératoires associés et leurs limitations. Il remarque que cinq générations d'interaction utilisateur avec les ordinateurs nous ramènent en quelque sorte au point de départ. En effet, les utilisateurs de la première génération avaient un accès dédié à l'ordinateur et un contrôle direct. Les utilisateurs disposent maintenant d'un panneau de contrôle virtuel sur leur propre machine.

| Génération | Mode opératoire | Barrière |
|------------|--------------------------------|---------------------|
| Première | Boutons, prises | Panneau frontal |
| Deuxième | Cartes perforées | Compteur |
| Troisième | Télétype, temps partagé | Terminal |
| Quatrième | Systèmes de menus | Hiérarchie de menus |
| Cinquième | Contrôles graphiques, fenêtres | Ecran |

Figure 5.5 Cinq générations de modes d'interaction avec l'ordinateur (Walker, 1990)

Les quelques repères de la figure 5.5 fournissent des indications sur l'influence de l'état de la technique sur l'informatique éducative. Ainsi, les grands projets des années soixante-dix ont été liés à la troisième génération et dominés par les contraintes du temps partagé. Alors que l'existence d'un réseau devait quasiment être cachée à l'utilisateur pour lui donner l'impression d'être tout seul, se développent maintenant les possibilités de travail en groupe.

Nous allons essayer dans la suite de faire ressortir les grandes idées associées à la conception des interfaces et d'en dégager les implications concernant les systèmes à vocation éducative. Les notions de métaphore et de manipulation directe et l'intérêt des représentations graphiques structurées dans les environnements d'apprentissage invitent à réfléchir sur les modes de communication personne-machine. Toutefois, de nombreuses situations nécessitent le recours à des formes plus ou moins simplifiées de dialogue, à l'aide de simples langages de commande ou d'un sous-ensemble de la langue naturelle.

5.3.1. Interfaces graphiques et représentations visuelles

Les développements autour des micromondes, décrits au chapitre précédent, ont montré l'intérêt, pour communiquer avec une machine, de manipuler des objets concrets représentés à l'écran. L'apport des écrans graphiques, souris, icônes, menus

déroulants..., a considérablement amélioré les interfaces, dont la prise en main est grandement facilitée. Pourtant, la simplification d'accès à l'ordinateur est loin de résoudre tous les problèmes de communication, d'autant plus s'il s'agit d'apprendre à l'aide d'une machine.

5.3.1.1. Métaphores et manipulation directe

La conception des interfaces semble actuellement dominée par l'idée de métaphore qui, selon Erickson (1990), désigne un réseau invisible de termes et d'associations qui sous-tend la façon dont on parle et pense un concept. Les métaphores fonctionnent comme des modèles naturels ; elles permettent de tirer parti de notre connaissance des objets concrets et familiers et de nos expériences pour structurer des concepts plus abstraits. Deux classes de métaphores (Coutaz 1990, p.59) interviennent principalement dans la conception de l'interface avec un ordinateur. La première s'inspire du monde réel et suit la métaphore du bureau (dossiers, fichiers, outils) lancée par Smalltalk, rendue célèbre par le Star de Xerox (Smith, 1982), reprise par les machines Lisa et popularisée avec le Macintosh de la société Apple. L'interface s'efforce d'être la reproduction miniature du monde du bureau (extension de la notion de micromonde) et la manipulation de symboles s'effectue par l'intermédiaire d'objets représentés à l'écran. L'autre classe d'interface ne représente plus explicitement l'univers manipulé mais se présente comme un médium linguistique ; on communique par l'intermédiaire de menus, d'un langage de commande voire de la langue naturelle. Si, dans ce dernier cas, l'utilisateur manipule des structures linguistiques, avec un monde miniaturisé électroniquement, il peut être engagé dans l'action de manière directe. Comme le souligne Nanard (1990), « l'impression d'engagement direct correspond à ce que l'utilisateur ressent lorsqu'il peut agir directement et librement sur les représentations des objets de son propre monde et percevoir de façon immédiate leur réaction ». Cette possibilité de manipulation directe peut faciliter la libre exploration d'un domaine (Hutchins et al. 1985). Elle est caractérisée, au plan informatique (Shneiderman 1982, Hutchins, Hollan et Norman, 1985), par :

- une représentation permanente des objets intéressants,
- des actions physiques ou des pressions sur des boutons étiquetés à la place d'une syntaxe compliquée,
- des opérations rapides (réaction du système aux actions de l'utilisateur), incrémentales (chaque modification reste de faible amplitude) et réversibles (possibilité de revenir à l'état antérieur ce qui donne une certaine confiance) dont l'impact sur les objets est immédiatement visible.

La manipulation directe peut s'appuyer sur différentes particularités : un aspect *mimétique*, dans le cas où la forme physique des actions qui sont exécutées imite les opérations désirées dans le domaine considéré ; une correspondance *sémantique* s'il existe un couplage étroit entre l'interface et le domaine, correspondant à l'idée de « construire la sémantique du domaine dans le comportement de l'interface », en rendant les modèles conceptuels du domaine et de l'interface aussi similaires que possible ; une caractéristique *temporelle* avec rétroaction immédiate à propos des actions, du modèle conceptuel et de ce qui est perçu comme résultat. Les systèmes

peuvent être ainsi considérés comme étant directement manipulables dans l'un, deux ou tous les sens précédents.

Si les interfaces de manipulation directe ne sont pas exemptes de limitations, leur développement accompagne l'intérêt manifesté par de nombreux chercheurs sur les représentations visuelles et spatiales.

5.3.1.2. Intérêt des représentations visuelles et spatiales

Au cours du chapitre III, nous avons eu l'occasion de souligner l'importance de ces types de représentation, notamment pour la résolution de problèmes. Les différences entre les diverses représentations ne viennent pas tant des notations utilisées pour les écrire mais des opérations que l'on peut faire sur elles. Une représentation n'est en effet utile que si on a les moyens de l'utiliser, c'est-à-dire si on est capable de s'en servir pour faire des déductions. Il peut être difficile de reconnaître ce qui est pertinent dans une situation donnée si celle-ci n'est pas présentée sous une forme qui s'apparente à nos règles de production (compétence liée à des représentations et notations types).

Larkin et Simon (1987) plaident pour l'usage de représentations spatiales. Elles s'avèrent souvent supérieures à des descriptions verbales pour résoudre des problèmes, parce qu'elles préservent explicitement les informations de nature géométrique et topologique concernant les composantes du problème. Résoudre un problème exposé dans une forme appauvrie consiste parfois à le traduire en utilisant une meilleure représentation. De même, les représentations sous forme de diagramme (informations indexées par un endroit dans un plan) offrent des avantages importants vis-à-vis des représentations purement langagières (informations indexées par une position dans une liste). Elles permettent de regrouper les informations utilisées de manière conjointe, évitant ainsi une recherche des éléments nécessaires pour faire une inférence ou la reconnaissance d'expressions symboliques. En outre, elles permettent d'effectuer un grand nombre d'inférences perceptives, celles-ci étant très faciles pour les humains⁴.

Sur ce dernier point, on remarque une complémentarité entre les techniques d'intelligence artificielle symbolique et connexionniste dans l'association de fonctions perceptives et de fonctions symboliques de traitement. Néanmoins, bien que des inférences perceptives soient aisées, rien n'assure qu'elles soient toujours utiles dans la résolution d'un problème. Si les représentations graphiques permettent sans doute de réduire la recherche et la charge de la mémoire de travail en organisant les informations de manière spatiale, pour certaines tâches, telle la compréhension de programmes informatiques, les représentations textuelles peuvent s'avérer meilleures. Comme le remarquent Cox et Brna (1995), l'aspect facilitateur des représentations graphiques repose en partie sur leur caractère particulier ou spécifique dans le sens où, lorsque plusieurs modèles sont possibles, une

⁴ « Contrairement au discours verbal, la représentation visuelle n'est pas limitée par ce dont une personne est capable de se rappeler du début à la fin d'un discours. La juxtaposition de plusieurs concepts et de leurs relations est analogue à un système ; le tout est plus que la somme des composantes prises séparément. » (Meilleur et Mitchell, 1989).

représentation graphique ne peut normalement n'en représenter qu'un seul, à moins de la compléter de diverses manières (ombrage, annotation, animation...). Il est possible que cet aspect spécifique soit partagé par les représentations internes utilisées par les humains dans leurs raisonnements. Même si les représentations graphiques fournissent des rétroactions plus vivantes et plus claires que des explications purement verbales, leur expressivité est limitée, notamment pour exprimer des choses abstraites.

Les techniques de visualisation (avec ou sans possibilités d'animation) facilitent l'émergence de modèles mentaux. En partie métaphores, analogies, symboles, l'important n'est pas de savoir si elles sont vraies, dans une perspective pragmatique, mais si elles sont utiles. De nombreux systèmes tirent bénéfice de représentations graphiques contrôlées dynamiquement par ordinateur. On peut citer ainsi l'utilisation de points de vue (Moyse, 1989), de diagrammes (Hall, 1989), d'animation (Nathan et al. 1989), de sortes de graphes de preuve (BRIDGE, Bonar et Cunningham, 1988 ; ANGLE, Koedinger et Anderson, 1990) ou de formes de tableau de bord (GUIDON-WATCH, Richer et Clancey 1987). Pour Eisenstadt (1989), l'usage de diagrammes fournit une alternative aux techniques difficiles issues de l'intelligence artificielle, en particulier dans l'apprentissage de la programmation. Notons toutefois que construire soi-même ses représentations est certainement plus formateur qu'en utiliser de déjà fabriquées. Cox et Brna (1995) remarquent qu'à leur connaissance, aucun système n'offre à l'apprenant une aide dans la sélection, la construction et l'utilisation d'un ensemble de représentations graphiques et non graphiques en cours de raisonnement. Par ailleurs, le choix d'une bonne représentation est une tâche non triviale et les apprenants peuvent rencontrer des difficultés à l'effectuer de manière optimale. Une aide intelligente peut s'avérer utile voire indispensable.

Les possibilités étendues de manipulation de représentations graphiques peuvent conduire à un réexamen des modes de pensée, voire même à une mise en cause des modes de codage ou de représentation traditionnels imposés par l'histoire et dont la justification ne tient plus qu'à la tradition. Une difficulté reconnue dans de nombreux domaines nécessitant de résoudre des problèmes est que l'écriture d'une solution ne reflète pas le raisonnement requis pour la construire (Polya, chap III ; Reiser et al., 1989) ou que les notations usuelles cachent les éléments perceptifs qui pourraient faciliter sa découverte⁵. L'idée d'une refonte des notations habituelles n'apparaît pas si insensée. Kaput (1986) a ainsi réalisé une étude dans ce sens concernant le domaine mathématique.

Partant de questions de communication avec une machine, les réflexions sur la manipulation de représentations contrôlées par ordinateur conduisent à s'interroger sur les modifications offertes par l'informatique dans l'environnement de travail des apprenants, voire d'utilisateurs de niveaux d'expertise très différents. Le travail sur DYNABOARD (Kaltenbach et Frasson, 1989) va ainsi dans ce sens: « *Comme des analogies concrètes appropriées sont difficiles à trouver pour les formes abstraites de raisonnement mathématique, il faut traiter directement ce symbolisme abstrait...*

⁵ C'est notamment le cas de nombreuses notations mathématiques. Ainsi, l'écriture linéaire d'une expression algébrique cache souvent sa structure arborescente.

Des facilités informatiques doivent être développées pour permettre de transformer graduellement des énoncés mathématiques en une terminologie plus familière et réciproquement ». La prise en compte de l'interface entre l'ordinateur et l'apprenant déborde largement le cadre d'une simple communication. Dotant l'utilisateur de capacités accrues d'expression, tout au moins dans certains domaines, fournissant des modes de représentation et d'explication multiples, l'environnement qui structure, assiste et contraint l'utilisateur prend une place prépondérante. L'ordinateur peut jouer simultanément plusieurs rôles qu'il faudrait judicieusement articuler.

5.3.2. Langages et interfaces graphiques

Si divers modes de communication entre l'apprenant et l'ordinateur s'avèrent possibles, se pose le problème du choix parmi ces différents modes et de leur éventuelle complémentarité. Les interfaces graphiques présentent des limitations importantes qu'il importe d'analyser, la génération d'explications et la conduite de dialogues demeurent encore essentielles dans les systèmes conçus sur l'apprentissage, la langue naturelle disposant de nombreux atouts vis-à-vis d'autres modes de communication. Enfin, l'interface proprement dite doit être adaptée à l'apprenant et évoluer.

5.3.2.1. Limites des interfaces graphiques

La manipulation directe, pour séduisante qu'elle apparaisse, est loin de répondre à toutes les nécessités de la communication entre une personne et une machine. Tout ne peut s'exprimer par l'intermédiaire d'opérations physiques sur des objets. C'est notamment le cas des processus. En effet, si on les représente par des objets, ils deviennent alors très abstraits et si on utilise des animations, ces dernières intègrent une dimension temporelle pas toujours facile à traiter. En outre, opérer directement sur des objets représentés à l'écran suppose qu'ils sont effectivement visibles, ce qui n'est pas toujours possible.

De même, si les métaphores facilitent la prise en main des environnements logiciels, trouver les meilleures métaphores demeure loin d'être simple. Ce qui risque de poser problème est la différence entre ce que perçoit et comprend l'utilisateur et la chose réelle. Différents critères interviennent pour juger de l'intérêt et de la pertinence d'une métaphore : son degré de structure, l'applicabilité de cette structure, la possibilité de la représenter, son caractère approprié à l'audience visée, son extensibilité. Plusieurs auteurs, tels Kay (1990) ou Meyrowitz (1989 à vérifier) sont très critiques vis-à-vis de la métaphore du bureau qui, comme le souligne Nelson (1990) amène à apprendre par *approximation* plutôt que par *compréhension*. Certaines métaphores utiles dans des contextes limités tendent à devenir des boulets car une fois instituées, chaque fonction doit en devenir une partie. Une trop grande soumission à une métaphore risque d'empêcher l'émergence d'idées réellement nouvelles. Le problème est donner corps à des idées qui correspondent à une

structure conceptuelle intentionnelle et un *sentiment* intentionnel pour l'utilisateur (Nelson 1987 p. 68). Pour l'utilisateur, il s'agit avant tout de traduire ses intentions dans le langage d'entrée de l'interface puis d'interpréter la réponse du système et évaluer s'il se rapproche de ses buts et l'une des choses les plus ardues est de trouver une architecture qui puisse servir de représentation mentale pour les intentions de l'utilisateur (Rohr et Tauber 1985). La séparation entre le contenu et l'interface ne peut jouer que sur des métaphores limitées. Le rapport entre connaissance et représentation de cette connaissance est primordial, le but est la manipulation conjointe. Comme le souligne Norman, il y a prédominance de la tâche sur l'outil. Notons enfin que la phénoménologie amène une méfiance vis-à-vis des modèles (Winograd et Florès 1986) et la nécessité de l'analyse en situation.

5.3.2.2. Dialogue et explications

Ainsi, s'il semble que l'usage de représentations graphiques soit essentiel, il est loin d'épuiser les besoins de la communication entre une personne et une machine dans un but d'apprentissage. Cet intérêt envers les représentations graphiques est à certains égards assez neuf dans le champ de l'informatique éducative et s'explique en partie par les possibilités accrues des matériels des années quatre-vingt. Dans la décennie précédente les chercheurs se focalisaient sur le traitement du langage naturel, souvent d'ailleurs en relation avec l'analyse de réponse chère à l'EAO traditionnel.

Comme le signale John Self (1985), le langage naturel a un avantage énorme pour les débutants : ils savent à peu près l'utiliser. Il est cependant trop souvent verbeux, vague et ambigu. Quand un système utilise une langue quasi naturelle (plus exactement un sous-ensemble souvent réduit), un usager peut facilement surestimer les capacités de compréhension de ce système et « aider les utilisateurs à estimer les limites des dialogues avec l'ordinateur est un des problèmes les plus ardues dans la conception des systèmes interactifs ». L'utilisation d'autres techniques nécessite de surmonter le même écueil, c'est-à-dire de faire comprendre aux utilisateurs ce dont est capable le système et ce qu'ils peuvent faire avec. Un échec amène soit une sous-utilisation soit, au pire, un rejet complet du système.

L'aspect dialogue ne doit pas être totalement écarté, bien au contraire, il conserve un rôle essentiel dans les processus d'apprentissage et d'enseignement. Brennan (1990, p.393) prétend même que les interfaces à manipulation directe ont du succès par les caractéristiques qu'elles partagent avec la conversation réelle.

Dans l'exécution d'une tâche, on distingue le langage lié à la tâche elle-même et le langage portant sur le traitement de la tâche. Pour ce dernier, le langage naturel, étant son propre métalangage, semble être le meilleur candidat, même s'il a de nombreux défauts. A la suite des travaux autour de SCHOLAR, WHY ou GUIDON, exposés au chapitre III, beaucoup de chercheurs se sont intéressés à l'aspect dialogue entre le système et l'apprenant. Leur travail se fonde le plus souvent sur des dialogues tutoriels réels pour la constitution de corpus et de grilles d'analyse (Elsom-Cook, 1990 ; Cauzinille et al., 1992). Différents systèmes incluent une modélisation

du dialogue. On peut citer rapidement KANT (Baker, 1989), NAIADE (Joab, 1990), STUDIA (Chevallier, 1992), SAVANT3 (Dessalles, 1992), DISCO (Lenne, 1995). A la suite des dialogues de type socratique, le problème de l'argumentation demeure important, bien que peu traité, mis à part dans des systèmes s'attachant à la mise en évidence d'hypothèses implicites ou la détection de contradiction.

Un des points essentiels demeure celui de l'explication. Un effort important a été entrepris du côté des systèmes experts. D'après Kassel (1989), la deuxième génération de systèmes experts vise à améliorer leurs capacités d'explication, principalement en rendant explicite la stratégie de contrôle et les règles heuristiques. Il s'agit de traiter la métaconnaissance et de concevoir des systèmes aptes à contrôler leur résolution et leur trace. En particulier, un résolveur contraint doit pouvoir n'utiliser que des connaissances censées être connues par l'apprenant, et être capable d'adapter le détail de la trace au niveau supposé de l'apprenant (tuteur algèbre, Ferret et Jimenez, 1987 ; APLUSIX, Nicaud 1988, NAIADE, Joab 1988 ; etc.). La trace elle-même est un objet mathématique intéressant qui peut être présenté à l'apprenant sous diverses formes, historique de la résolution (pistes abandonnées comprises), reflet de la pensée de l'apprenant (ALGEBRALAND, Collins et Brown, 1988). Toutefois, il apparaît difficile de produire directement une explication satisfaisante pour l'apprenant. Des explications complémentaires s'avèrent très souvent nécessaires ce qui conduit à l'établissement d'un dialogue. L'explication apparaît alors comme une négociation du sens entre apprenant et système. D'après Delozanne (1994), il est nécessaire d'insérer les explications données par le système dans des scénarios d'interaction qui leur assurent une pertinence vis-à-vis des objectifs du système et de l'utilisateur.

Quand on compare les possibilités des machines avec ce que font les enseignants humains, il convient de prendre en compte les spécificités des différents modes de communication. Le cours dit traditionnel est en fait complété par des modes de communication non verbaux (intonation, gestes, etc.). Dans ce cadre, l'opposition entre faire et faire-faire, traitement spatial de l'information et manipulation directe d'un côté, langage naturel de l'autre, se résout dans une intégration de différentes modalités d'interaction. Une tendance générale dans la conception des interfaces, est de retrouver l'analogique (exemple, le geste, le déplacement...). Le développement du multimédia, de la réalité virtuelle et des interfaces multimodales laissent présager des possibilités réellement nouvelles, notamment de par l'ajout du sensoriel et du perceptif permettant de dépasser le cognitif. Ce n'est pas sans poser des problèmes redoutables quant à l'apprentissage, en particulier pour ce qui concerne la pertinence des différentes modalités au cours de l'apprentissage.

5.3.2.3. Interfaces et apprentissage

L'interface est un milieu qui filtre les possibilités d'expression de l'apprenant (en fait toute action que permet ou contraint l'interface constitue une aide pour l'utilisateur) et qui peut être réactif⁶. La question que l'on peut se poser consiste à savoir s'il y a

⁶ Une interface doit répondre à deux contraintes contradictoires, la transparence, elle doit s'intégrer totalement dans le monde humain et la résistance, pour ne pas perdre son rôle structurant pour l'utilisateur. « Entre l'expert et le novice, il y a un déplacement de la résistance.

des spécificités des interfaces pour apprendre. Si une telle interrogation n'attend pas de réponse définitive, certaines caractéristiques méritent toutefois d'être mentionnées. Ainsi, Nicol (1990) décrit quatre manières de concevoir des interfaces pour aider les apprenants :

- *Territoire non familier* : fournir la réplique et donner des aperçus pouvant servir d'organiseurs avancés pour les apprenants découvrant un nouveau territoire (le problème majeur dans la première utilisation d'un système complexe est souvent le manque de guidage)
- *Processus de modélisation* : modélisation soigneuse et facilitation du processus de résolution de problèmes
- *Changement d'état* : utiliser le mouvement et des animations précises pour les représenter
- *Interfaces comme entraîneurs (coaches)* : utiliser l'interaction de manière à capitaliser l'intelligence de l'utilisateur.

Pour tenir compte de la multiplicité des apprenants et du fait que l'apprentissage est un processus dynamique, l'interface doit être adaptable. Trigg (1987) distingue quatre dimensions. Un système adaptable peut être :

- *flexible*, s'il fournit des modèles d'entités et de comportements qui peuvent être interprétés et utilisés différemment,
- *paramétrable*, s'il existe un intervalle de possibilités de comportements parmi lesquels l'utilisateur peut choisir
- *intégrable*, s'il peut être connecté ou admettre dans son environnement d'autres applications
- *personnalisable*, s'il autorise l'utilisateur à le changer en créant ses propres raccourcis d'utilisation, en ajoutant des fonctionnalités, en particularisant son comportement vis-à-vis du système.

De l'adaptabilité on passe à l'adaptivité, c'est-à-dire à la capacité pour un système de découvrir automatiquement, par observation à partir de l'interface de communication, les choix de l'utilisateur. La problématique des tuteurs intelligents est bien celle de concevoir des systèmes adaptables et de leur assurer l'adaptivité optimale. Il n'est pas sûr que cette dernière soit indéfiniment optimisable. Les recherches actuelles s'orientent vers la notion d'*agent actif*, ce qui consiste à mettre de l'intelligence dans les différents outils.

On peut remarquer que les notions de modèle élève et de modèle utilisateur sont distinctes. Le premier modèle est a priori peu lié au médium utilisé et fournit une représentation des connaissances et des croyances de l'apprenant quant au domaine d'apprentissage considéré. Le second intègre des caractéristiques propres et des préférences de l'utilisateur relativement à l'usage du système lui-même. Des variables communes peuvent être introduites dans les deux cas, concernant par exemple le style cognitif. Un modèle utilisateur peut s'avérer indispensable pour

L'expert a besoin de résistance pour contrôler le comportement de la machine, tandis que le novice peut considérer la résistance comme un obstacle, puisqu'il n'est pas en mesure d'en évaluer la portée. ». Caelen (1996) qualifie l'interface d'adaptateur d'impédance entre l'homme et la machine.

assurer une bonne maîtrise du système informatique sans pour autant modifier les tâches à effectuer avec le système. Une individualisation minimale est ainsi assurée mais elle concerne assez peu les connaissances en jeu.

En conclusion, il est intéressant de remarquer le rôle essentiel dévolu à l'interface. Cependant, le développement d'outils *intelligents* et de représentations exécutables, avec lesquelles on peut opérer par manipulation directe, s'intègre dans les micromondes autant que dans les tuteurs. Cette base commune peut constituer une ébauche à une synthèse entre ces deux approches pédagogiques. C'est ce que nous allons étudier dans la suite de ce chapitre. Auparavant, un certain nombre d'oppositions entre les deux approches se cristallisent autour de la pertinence voire de la possibilité d'une modélisation de l'apprenant répondant d'une manière suffisante à certains critères de fidélité ou d'opérationnalité.

5.4. Quelle modélisation de l'élève ?

Si l'insuffisance des modèles comportementaux a été soulignée au cours du chapitre III, les débats autour de la modélisation de l'apprenant ont certainement marqué la fin des années quatre-vingt. Des positions souvent idéologiques ont été prises, certains auteurs soutenant l'impossibilité de constituer dans une machine une réplique de l'état de connaissances d'un apprenant. Les chercheurs ont été unanimes dans la reconnaissance de l'énorme difficulté d'une telle modélisation. Pour surmonter les difficultés repérées, outre un travail à long terme avec des méthodologies plus formelles, diverses pistes ou solutions ont été proposées. Il s'agit d'améliorer les techniques ou les théories de référence et d'élargir le champ des observables. Tout d'abord l'utilisation de nouvelles techniques, notamment celles issues des travaux sur l'apprentissage automatique, ont été essayées. Ensuite, compte tenu du petit nombre d'observables disponibles, les chercheurs ont essayé de simplifier la tâche de diagnostic en intégrant d'autres informations fournies par l'apprenant ou en limitant l'étendue du modèle à constituer. Toutefois, l'identification de leurs utilisations possibles vont permettre de mieux préciser les caractéristiques des modèles de l'apprenant à implanter. Un travail de clarification sur les différents usages des modèles élèves a ainsi été exposé par Self (1988).

5.4.1. La classification de Self

Pour Self, un modèle de l'élève permet de répondre à des questions de quatre types :

- que peut faire l'apprenant, ce qui réfère à ses connaissances procédurales ?
- qu'est-ce qu'il connaît, c'est-à-dire quelle est sa connaissance conceptuelle ?
- quel type d'apprenant est-il, quelles sont ses caractéristiques individuelles ?
- qu'a-t-il fait, c'est-à-dire quel est l'historique du travail, cet historique étant constitué par une transcription de la session basée sur l'interprétation des événements jugés significatifs ?

Self (1988) a identifié vingt utilisations différentes d'un modèle élève dans des systèmes EIAO existants. Il en a dégagé six fonctions principales, qui dépendent des caractéristiques du modèle élève et de l'existence d'un modèle du processus d'apprentissage : la correction de l'erreur (modèle différentiel), la détermination du point suivant à étudier pour compléter les connaissances de l'élève (modèle d'expertise partielle), le contrôle de la stratégie globale d'interaction, la construction d'un diagnostic plus fiable ou plus précis (lorsque plusieurs interprétations sont a priori possibles), la prédiction de la performance ou de l'apprentissage (si on dispose d'un modèle du processus d'apprentissage) et enfin l'évaluation de l'élève (ses réponses et ce qu'il est censé savoir) ou des potentialités d'un système.

| <i>Fonction / Modèle élève</i> | <i>Objet Type d'action</i> | <i>Choix de l'action</i> |
|--|--|--|
| <i>Corrective</i> Différentiel (incorporant des bogues) | Aider à éliminer les connaissances erronées <i>Choisir une rétroaction (décision locale)</i> | <ul style="list-style-type: none"> • présentation de l'erreur commise • remédiation directe • remédiation indirecte • contre-exemple • présentation à l'élève du chemin qu'il est supposé avoir suivi • examen rétrospectif (source de l'erreur) • message : recommencer • repli tactique (génération d'un problème dans lequel cette erreur n'intervient pas) |
| <i>Elaborative</i> Expertise partielle | Compléter les connaissances <i>Choisir un sujet à aborder (décision locale)</i> | <ul style="list-style-type: none"> • fondé sur une progression • par comparaison expert-apprenant • par analyse interne de la connaissance de l'apprenant • laissé à l'étudiant (dans une liste filtrée) |
| <i>Stratégique</i> Quelconque | Contrôler l'interaction <i>Changer</i> | <ul style="list-style-type: none"> • le plan de conduite de session • le style d'interaction |
| <i>Diagnostique</i> Quelconque mais incertain | Vérifier et préciser le modèle élève <i>Choisir une action pour lever une ambiguïté</i> | <ul style="list-style-type: none"> • sur le modèle élève (générer un problème discriminant ou interroger l'apprenant) • de l'élève lui-même sur la connaissance à apprendre |
| <i>Prédictive</i> (*) | Réguler la modélisation et le choix des actions <i>Simuler l'apprenant pour calculer</i> | <ul style="list-style-type: none"> • sa performance future (afin de limiter l'espace de recherche) • l'effet des actions didactiques (pour choisir la meilleure et élaborer le nouveau modèle) |
| <i>Evaluative</i> (*) | <i>Evaluer a posteriori</i> ----- <i>Mesurer l'efficacité potentielle</i> | <ul style="list-style-type: none"> • l'apprenant (résumé de ses performances et connaissances supposées) • d'un système (comparer différentes stratégies didactiques avec un apprenant simulé) |

Figure 5.6 Utilisations des modèles élève d'après Self (1988)

(*) Modèle quelconque incluant une modélisation du processus d'apprentissage

Concernant les deux dernières utilisations, en plus d'analyser, interpréter ou expliquer les actions passées, disposer d'un modèle du processus d'apprentissage avec un apprenant simulé fournit des critères d'analyse intéressants. Le calcul de l'état *supposé* suivant du modèle élève donne des arguments pour choisir une action didactique. A partir d'une mesure d'évaluation sur les modèles élèves, il est possible d'exécuter la procédure d'apprentissage suivant un ensemble d'actions potentielles et de choisir celle qui a l'effet le plus bénéfique. Self présente ce catalogue comme un commencement, demandant d'identifier plus soigneusement les contextes dans lesquels ces différentes utilisations sont pertinentes et d'améliorer les techniques pour mettre en œuvre ces techniques plus efficacement.

Ainsi, un modèle élève constitue à la fois une source d'informations sur l'apprenant et une représentation de ses connaissances. Le choix d'une action peut se faire en réponse aux comportements de l'apprenant d'une manière opportuniste (fonction correctrice ou élaborative), planifiée (fonction stratégique) ou par l'évaluation de son effet supposé sur le modèle élève (fonction prédictive).

Si la classification proposée par Self a abondamment été citée, elle va susciter deux approches complémentaires. La première intègre la modélisation de l'apprenant dans la construction de tuteurs intelligents et restreint sa portée. Il s'agit de déterminer les croyances de l'apprenant pour guider les interventions du tuteur. La seconde approche, plus formelle, considère le problème de la modélisation indépendamment de son usage dans un tuteur automatisé.

5.4.2. Améliorer le diagnostic

5.4.2.1. Par l'apport de nouvelles techniques

Nous avons déjà mentionné au chapitre III l'intérêt des techniques de reconnaissance de plans. La tâche prototypique consiste à essayer de reconnaître la structure d'une séquence observée pour discerner le but poursuivi, sachant qu'il peut y avoir plusieurs buts, voire même aucun. De nombreux systèmes ont recours à ce type de technique. L'approche suivie par Walsh (1988) dans PLATO est intéressante, car elle représente explicitement des heuristiques dans l'expertise du domaine, heuristiques qui guident la reconnaissance des plans de l'élève. On peut remarquer que si une interaction de type traçage de modèle (Anderson, 1989) ne conduit pas à exprimer un diagnostic cognitif sur les erreurs, une extension de cette interaction implique l'utilisation de techniques de reconnaissance de plans.

En complément des techniques de reconnaissance de plans, il apparaît important de prendre en compte le contexte dans lequel une erreur intervient ainsi que de définir des priorités *plausibles* (Noël et Py, 1993). Dans son travail autour des calculs sur les fractions, Visetti (1986) s'attache à reconstituer un plan mis en œuvre par un élève dans la conduite d'un calcul. Il s'appuie sur la notion de contexte, vu comme un couple <tâche, situation>, et des relations d'ordre (généralisation, particularisation) entre ces contextes. Les erreurs sont ainsi des connaissances déviées intervenant dans des contextes formels. Pour éviter un trop grande

multiplicité de déclarations de procédures erronées, il est amené à représenter des corrélations liant les comportements observés dans différents contextes.

Un autre problème est celui de la constitution d'une bibliothèque de bogues. Kuzmycz et Webb (1992), dans un travail consacré à la soustraction, décrivent un système qui ne requiert pas l'existence d'une telle bibliothèque et qui prend en compte les erreurs multiples, les problèmes de migration et de bruit⁷. En supposant que les erreurs procédurales sont structurellement proches des procédures correctes, Gustein (1992), dans un système sur les fractions, propose d'identifier les facteurs de variation. Pour Ohlsson (1992), une solution consiste à représenter le domaine comme un ensemble de contraintes. Les violations de ces contraintes de la part de l'apprenant indiquent les connaissances incorrectes et incomplètes et peuvent guider les réponses d'un système tuteur intelligent. Une telle approche est susceptible d'éliminer la nécessité de créer des modèles exécutable aussi bien de l'élève que de l'expert et de réduire les computations requises pour la modélisation de l'élève à de simples appariements.

La génération d'exercices est un moyen de tester une hypothèse de modélisation de l'apprenant ou lever une ambiguïté entre plusieurs modèles de performance possibles. Si dans IDEBUGGY des exercices sont associés à chaque bogue, LMS/PIXIE utilise des versions génératives inverses des règles de résolution. Il s'agit de trouver des problèmes critiques (Evertz et Elsom-Cook, 1990) pour discriminer les erreurs suspectées. Des méthodes d'analyse automatique de protocoles permettent également d'affiner le diagnostic (Ngyuen-Xuan et al., 1993).

Des techniques avancées en intelligence artificielle sont aussi utilisées, comme l'apprentissage symbolique automatique (Kodratoff 1986 ; Gilmore et Self, 1988 ; Costa et al., 1987 ; Bento et Costa, 1988), les logiques floues (comme dans TAPS, Derry et al., 1989) ou les logiques de la croyance pour la création des modèles de l'élève. Les erreurs de l'apprenant sont considérées comme l'utilisation de conditions erronées conduisant à sélectionner un opérateur inadéquat. Les circonstances de choix forment les exemples positifs et négatifs à partir desquels se construit l'apprentissage. Ainsi, le module diagnostic construit par Talbi et Joab (1992) repère les connaissances procédurales, correctes ou erronées, que l'apprenant a utilisé lors de sa résolution. Il engendre des généralisations des productions de l'apprenant tout en contrôlant leur vraisemblance et analyse les contextes d'application pour délimiter précisément la partie "condition" des règles erronées de l'apprenant.

Dans le cadre du développement de TAPS, tutoriel sur les problèmes algébriques posés en langue naturelle, Derry et ses collègues (Derry et al., 1989 ; Derry et Hawkes, 1993) distinguent le diagnostic dit local basé sur la performance courante de l'élève, surtout lié à l'utilisation des schémas, du diagnostic global qui doit faciliter les décisions globales du tuteur. Leur problème est : (1) de rendre des jugements valides en dépit des 'bruits' obtenus, (2) de maintenir et mettre à jour des modèles individuels détaillés et accessibles pour le tuteur et (3) d'utiliser ces modèles pour guider les interventions aussi bien au niveau global (curriculum) qu'au

⁷ Le fonctionnement de ce système est proche de celui de ACM décrit au chapitre III.

niveau local (niveau opportuniste). La solution qu'ils préconisent consiste à implanter des modèles flous correspondant au fait de penser en des termes linguistiques imprécis de type humain. Au niveau purement diagnostic (Derry et al. 1989), ils s'appuient sur un catalogue représentant 98% des erreurs faites, et une méthode de déviation utilisant une représentation intermédiaire (*graphic reification*).

Ce dernier type de méthode, cherchant à simplifier le diagnostic en l'associant plus étroitement aux interventions tutorielles et utilisant des environnements adaptés permettant un premier filtrage, devient tout à fait classique.

5.4.2.2. En le simplifiant

Plusieurs méthodes de simplification du diagnostic peuvent être utilisées. La plus simple consiste à se concentrer sur les erreurs caractéristiques ou les plus courantes (ou des priorités que l'on peut qualifier de plausibles). Ainsi, dans leur étude sur la soustraction, Young et O'Shea (1981) soulignent l'existence de 15 erreurs fondamentales. Pour eux, les multiples bogues répertoriés dans BUGGY ne sont pas assez profonds et sont souvent trop proches, ce qui les rend inutilisables pour la remédiation. Les mêmes réserves sont faites par Nicolson (1988). Son système, nommé SUMMIT ne tient compte que des erreurs les plus fréquentes (8 erreurs d'addition couvrent 92,3% des cas, 8 erreurs en multiplication couvrent 91,4% des cas).

Le débogueur automatique CHIRON (Sack, 1988), conçu comme une réétude de PROUST (Johnson 1985) dédié à l'apprentissage du langage PASCAL, s'appuie sur des considérations similaires. 70% des erreurs sont dues à des détails (syntaxe, substitution d'opérateurs, etc.) ou à de mauvaises coordinations d'éléments de programme. Les systèmes de diagnostic fondés sur des règles erronées (*mal-rules*) sont inadéquats notamment du fait que tous les bogues sont supposés de même niveau et sont souvent trop spécifiques, que l'introduction d'un nouveau bogue implique la création d'une nouvelle règle et que les différentes parties de programme sont supposées indépendantes. CHIRON travaille avec des plans définis dans un langage de *frames* et des généralisations de "fragments" avec des hiérarchies prédéfinies (instance / classe, partie / tout). Il peut disposer aussi de plusieurs représentations (tranches) du même programme.

Vu la difficulté de mise en place de modèles élèves performants, Self (1988) propose quelques principes à appliquer pour les construire d'une manière effective : éviter de deviner, demander à l'élève de vous dire ce que vous avez besoin de savoir ; ne pas diagnostiquer ce que l'on ne peut traiter ; tenir compte des croyances de l'élève et ne pas les cataloguer comme bogues ; ne pas feindre l'omniscience, adopter un rôle de collaborateur "faillible".

Deux techniques principales sont à même de simplifier considérablement la recherche du diagnostic : faire déclarer à l'apprenant son but ou ses intentions, utiliser des représentations intermédiaires. La déclaration des buts, intégrée notamment dans EPIC (Twidale, 1989), n'est pas dénuée d'intérêt sur le plan

didactique. BRIDGE (Bonar et Cunningham, 1988) cumule ces deux approches. Une idée maintes fois reprises consiste à donner des conseils stratégiques à partir de la mise en évidence des propriétés structurelles de l'espace de recherche. O'Shea et ses collègues (O'Shea al., 1988 ; Hennessy et al., 1989), en vue de créer ce qu'ils appellent un diagnostic par consultation (*diagnostic consultant*), fournissent aux enfants des moyens d'expression afin de connaître les techniques qu'ils utilisent. Dans SHOPPING ON MARS, le langage GADL (Graphical Arithmetic Description Language) a été créé pour permettre à l'élève de communiquer à la machine ce qu'il a fait pour obtenir un résultat (mode de calcul) ou à l'inverse permettre à la machine de présenter un nouvel algorithme. Une interface de manipulation directe est utilisée et tous les pas de la solution de l'élève sont explicitement présents à l'écran.

5.4.2.3. Mais tenant compte de nouveaux aspects

Même si des techniques spécifiques ou des modes d'interaction permettent de simplifier le diagnostic, sa nature est souvent un peu différente. En effet, vu l'insuffisance de modèles fondés sur des sous-ensembles ou des déviations des connaissances expertes, les chercheurs ont tenté de prendre en compte des conceptualisations du domaine fondamentalement différentes de celles de l'expert. Ces *modèles du fonctionnement cognitif* (Paliès, 1988) cherchent à prévoir le comportement d'un élève en situation d'apprentissage. Il ne s'agit plus uniquement de décrire l'ensemble des connaissances mises en jeu pour résoudre un problème, mais aussi de considérer la représentation mentale que se fait l'apprenant du problème. On cherche à intégrer les connaissances profondes de l'élève dans le modèle, dans un système global de croyances : croyances du sujet dans le domaine considéré, mais aussi croyances du sujet sur sa propre connaissance. En effet, une croyance erronée d'un élève sur ce qu'il comprend constitue un obstacle très important à l'apprentissage.

D'après du Boulay et Sloman (1988), les élèves devraient être modélisés comme des êtres affectifs, avec des motivations à la fois intellectuelles et non-intellectuelles et des capacités à jouer un rôle dans l'apprentissage. Les chercheurs s'intéressent de plus en plus aux aspects de motivation et commencent à proposer des réalisations. Fondé sur les tactiques de motivation définies par Malone et Lepper (1987 ; voir aussi Lepper et Chabay, 1988), un tuteur compétent sur les aspects motivations est construit par del Soldato et du Boulay (1995). La difficulté est de détecter l'état de motivation d'un apprenant et de construire un planificateur s'y appuyant. Matsubara et Nagamachi (1996) proposent eux aussi un système de motivation.

Notons que peu à peu, on passe d'un modèle élève, plutôt de type statique, à un modèle apprenant de type dynamique qui doit représenter l'évolution dans la phase d'apprentissage. Maintenir des modèles apprenant de type dynamique complexifie le diagnostic. Considérer un tel modèle comme une description psychologiquement crédible d'un partenaire collaborant (Gilmore et Self, 1988) conduit à favoriser la collaboration entre un système éducatif informatique et un utilisateur, processus susceptible de faciliter le développement d'un tel système en réduisant la nécessité d'une complète précision de la représentation du domaine et du modèle élève (Cummings et Self, 1989).

Si l'idée de construire le modèle élève en utilisant un dialogue de coopération avec la machine apparaît intéressante, les problèmes de maintenance de la vérité et de non monotonie demeurent redoutables. S'agissant de prédire la performance, les théories explicatives n'étant pas encore suffisamment opératoires, les aspects probabilistes, très utilisés dans les théories de l'apprentissage dans l'enseignement programmé, refont peu à peu surface. En effet, ils permettent d'obtenir un meilleur effet prédictif (voir Reye, 1996 et les logiques de croyance). Corbett et Knapp (1996) décrivent ainsi un modèle probabiliste du traçage de connaissance.

Toutefois, l'un des points importants est celui de la formalisation théorique du processus humain d'acquisition des connaissances. Expérimenter sur la base de modèles (fonctions prédictives et évaluatives du modèle dans la classification proposée par Self) est une voie prometteuse.

5.4.3. Des modèles formels et computationnels

Les théories cognitives apparaissent utiles pour contraindre les architectures (voie suivie par Anderson) ou pour fournir un cadre théorique pour le travail empirique à la fois formatif et sommatif. Mais un travail de nature théorique, indépendant d'architectures peut aussi être mené. Self (1992) recommande de suivre une approche formelle non seulement pour fournir un langage précis et universel pour décrire les modèles de l'apprenant mais aussi pour établir une théorie dans laquelle des théorèmes peuvent être prouvés. La mathétique⁸ computationnelle de Self se propose ainsi d'étudier l'apprentissage (de développer des formalisations computationnelles des processus d'apprentissage) et de lui donner de solides fondations théoriques tout en gardant un intérêt vers les applications, c'est-à-dire pour la conception de systèmes informatiques facilitant l'apprentissage. Dans une telle approche, un modèle de l'apprenant est vu comme l'ensemble des croyances d'un tuteur intelligent sur les croyances d'un apprenant. La prise en compte des connaissances incomplètes et/ou inconsistantes, de même que la révision des croyances s'intègre dans ce travail théorique.

Si des méthodes formelles peuvent fournir des résultats importants, les méthodes expérimentales peuvent elles aussi donner de précieux renseignements. Cependant, les expérimentations avec des apprenants humains ne sont ni faciles à organiser ni faciles à contrôler et à interpréter. L'apprentissage automatique peut être utilisé pour dériver une démarche d'instruction effective à partir d'une théorie descriptive de l'apprentissage. Ainsi, Ohlsson (1992) propose de tester des méthodes d'enseignement sur des modèles exécutables de simulation de l'apprentissage. Disposer d'élèves artificiels, c'est-à-dire plus concrètement de programmes d'apprentissage automatique, permet de les entraîner avec différentes formes d'instruction. Il n'est plus nécessaire de détecter la connaissance et les mauvaises

⁸ Le terme de mathétique a été employé par Gilbert, comme une méthode particulière de conception de programmes en enseignement programmé ainsi que par Papert (1980).

conceptions de l'élève en ligne. Le programme peut définir la connaissance disponible pour l'apprenant. Ceci permet de multiples expérimentations.

Van Lehn et ses collaborateurs ont beaucoup travaillé selon cette approche. La première réalisation fut SIERRA, programme d'apprentissage automatique conçu pour modéliser l'apprentissage de l'arithmétique par de jeunes enfants ainsi que la résolution d'équations algébriques élémentaires et d'autres aptitudes mathématiques calculatoires simples (apprentissage à partir d'exemples pris dans un livre). De même, CASACADE a été conçu pour modéliser le processus d'auto-explication repéré par Chi et Bassok (1989). D'après Van Lehn et al. (1994), il est maintenant possible de réaliser des systèmes d'apprentissage automatiques dont le comportement est consistant avec les données issues d'apprenants humains. De telles applications peuvent aider trois groupes distincts d'utilisateurs :

1. *les enseignants* qui peuvent s'entraîner sur des élèves simulés, ayant diverses caractéristiques contrôlables (stratégies d'apprentissage ou connaissances initiales)
2. *les élèves* qui peuvent apprendre en coopération avec un apprenant simulé (qui peut être à la fois expert et co-apprenant)
3. *les développeurs de ressources éducatives*, qui peuvent effectuer des tests dans les phases initiales de la conception.

Pour mener des expérimentations avec des *élèves artificiels*, il est nécessaire de formaliser à la fois les parties pertinentes de la connaissance de l'apprenant humain avant de recevoir l'instruction devant être donnée et l'instruction elle-même. Le modèle produit alors une représentation formelle de la connaissance de l'apprenant humain à l'issue de cette instruction et une représentation formelle du comportement de l'apprenant durant la phase d'apprentissage. Des simulations de période plus longue sont faites par exécution de nombreux cycles du processus précédent. Dans cette formalisation, il faut tenir compte de la granularité plus ou moins fine de la représentation de l'apprenant et de la façon de modéliser le processus d'apprentissage.

L'étude menée par Ohlsson, Ernst et Rees (1992) constitue une application intéressante des apprenants simulés. Ils ont comparé l'apprentissage de deux techniques opératoires de la soustraction en croisant avec la connaissance du sens de la soustraction. Ils ont simulé quatre apprenants, deux *compréhendant* la signification de la soustraction et deux ne voyant dans la soustraction qu'une manipulation de symboles. A chaque paire, ils ont enseigné à un élève la méthode dite naturelle (par emprunt et échange) et à l'autre la méthode utilisée en France par incrémentation du chiffre de la colonne adjacente du nombre à soustraire. Ils ont trouvé que cette dernière méthode était toujours la plus facile à apprendre, et particulièrement aisée quand l'élève simulé *compréhendait* la sémantique de la numération décimale⁹.

⁹ Sans entrer dans des considérations méthodologiques très profondes, une des raisons qui peut expliquer ce résultat est que la méthode par compensation n'introduit pas de cas particulier comme l'emprunt sur un 0 dans la méthode naturelle. L'ajout d'une unité marche dans toutes les situations ce qui conduit à une économie de pensée sans doute déterminante.

S'il apparaît difficile de juger de la pertinence et de fiabilité de résultats obtenus par des méthodes utilisant des apprenants artificiels, des systèmes de ce type ont un intérêt certain en formation. Ainsi, le système STEPS (Ur et Vanlehn, 1995) apprend par interaction avec un tuteur humain. Il résout des problèmes de physique, montrant son travail sur l'écran d'une station de travail, sous le contrôle d'un tuteur humain. Travailler avec un tel étudiant simulé est sans doute intéressant pour l'entraînement des enseignants humains.

Quels sont les critères d'applicabilité de l'apprentissage automatique vers l'apprentissage humain (prise en compte de l'affectif, de la motivation, des phénomènes d'oubli, ...)? Peut-on concevoir un apprentissage humain *sans signification*, ici sur l'apprentissage de la soustraction? Sans pouvoir répondre à ces questions et en concédant que les modèles utilisés sont fortement réducteurs, les enseignants peuvent certainement tirer bénéfice d'un entraînement avec des apprenants artificiels. Comme avec tout programme de simulation, ils peuvent en effet tenter des choses qu'ils n'auraient sans doute pas pu essayer autrement et observer sans risque les résultats obtenus.

Le thème de la modélisation de l'apprenant nous ramène en fait aux processus d'enseignement. Or, comme le signalent Derry et Hawkes (1993), les enseignants humains ne conduisent pas de diagnostics très détaillés avant d'intervenir pour aider les apprenants. Ils interviennent sur la base d'indicateurs leur montrant que les apprenants rencontrent des difficultés durant leur processus de résolution de problème. La question de fond est de savoir si un système peut être adaptable aux besoins de l'apprenant sans modélisation de l'apprenant (McCalla et Greer, 1992). Pour de nombreux chercheurs, la modélisation de l'apprenant est la clé de l'enseignement individualisé. Toutefois, d'autres courants, comme les tenants de l'apprentissage situé, ne croient pas aux représentations mentales (internes) et donc ne croient pas à la modélisation. En fait, si disposer de renseignements sur l'apprenant est utile, faut-il sur cette base exercer un contrôle sur le processus d'apprentissage ou inventer des modes d'interaction plus souples? Cette interrogation nourrit les tentatives de synthèse entre les tuteurs artificiels et les environnements d'apprentissage ouverts inspirés de LOGO.

5.5. Vers une synthèse tuteurs / micromondes ?

« Le concept d'environnement d'apprentissage intelligent généralise les divers types de systèmes développés par les chercheurs en IA et Education. Des systèmes tuteurs intelligents aux micromondes de type LOGO, ces systèmes varient suivant un continuum. Leur position est définie par le degré d'initiative laissée à l'apprenant, la représentation de la connaissance de l'élève, la connaissance du domaine à enseigner, etc. La combinaison des mots 'intelligent' (venant des tuteurs intelligents) et 'environnement' (venant des micromondes) montre les évolutions de la recherche vers la contraction de ce continuum en un point d'union. »
(Dillenbourg et al., 1990)

Dès la fin des années soixante-dix, les chercheurs, conscients des vastes possibilités offertes par les ordinateurs, ont essayé de concilier deux approches, les interactions tutorielles et l'apprentissage par la découverte (Sleeman et Brown, 1982). Mais, si les difficultés et limites de chacune des approches ont bien été analysées, rien n'assure qu'une synthèse soit réellement possible. En effet, dans un enseignement par l'ordinateur, ce dernier est la source du savoir, alors qu'il assume un rôle très différent, de type instrument ou partenaire, dans le cadre d'un apprentissage par la découverte.

Nous avons vu que les recherches autour des environnements d'apprentissage ouverts ont contribué à la réalisation d'environnements de travail, ces derniers pouvant être intégrés dans des tuteurs. Est-il possible de concevoir un environnement de travail et d'apprentissage prescriptif sans en pervertir la nature ? Greeno (1990) distingue ainsi deux rôles principaux de l'ordinateur qu'il qualifie de *didactique et exploratoire*, correspondant à deux conceptions de l'apprentissage et de la cognition. La vision didactique reflète une théorie qui considère la cognition comme un système de procédures et de structures d'information et l'apprentissage comme l'acquisition de structures et procédures cognitives. La vision exploratoire correspond mieux à une théorie considérant la cognition comme une activité fondamentalement située dans des contextes sociaux et physiques et l'apprentissage comme renforçant des capacités dans des activités situées. La préférence de Greeno se tourne vers des environnements d'apprentissage productifs. Pour lui, l'ordinateur fournit des sujets de conversation et des activités engageant apprenants et enseignants, mais la source primordiale de l'apprentissage est l'activité elle-même, non le contenu du programme.

Ainsi, au-delà du rôle même dévolu à l'ordinateur, les oppositions entre les aspects de guidage et de découverte reposent sur des conceptions de l'apprentissage antagonistes. Comme le soulignent Lamontagne et Bourdeau (1992), le problème de l'apprentissage par découverte guidée est celui de l'articulation entre l'induction et la déduction. Alors que les environnements d'apprentissage se centrent sur la découverte et l'apprentissage, les tuteurs se focalisent sur le domaine et l'enseignement. En fait, les différents rôles apparaissent difficilement compatibles et correspondent à des théories largement inconciliables.

Avant de voir les formes de synthèse possibles associées à de nouveaux rôles dévolus à la machine et de présenter certains concepts généraux issus des recherches sur les tuteurs et les environnements d'apprentissage, nous allons brièvement faire le point sur les possibilités de réalisation de tuteurs intelligents et décrire un exemple intégrant successivement des aspects découverte et guidés.

5.5.1. Quelle synthèse ?

Nous ne passerons pas en revue un ensemble de systèmes proposant une synthèse entre tuteurs intelligents et environnements d'apprentissage ouverts. Nous nous contentons d'un exemple qui nous paraît caractéristique d'une approche

intégrant micromonde tuteurs dans des phases successives d'un processus d'apprentissage. Le travail de Feuerzeig sur l'algèbre est exemplaire sur ce point.

5.5.1.1. Un exemple de synthèse : apprentissage de l'algèbre

L'approche de Feuerzeig (1986) s'appuie sur trois composantes :

1. Un travail sur des projets de programmation en LOGO, contexte à la fois riche au niveau algébrique et offrant un contenu signifiant et attirant pour les élèves,
2. Des micromondes avec des représentations iconiques concrètes des objets et opérations formelles,
3. Un système tuteur expert pour aider les élèves à conduire des opérations algébriques complexes incluant un mode démonstration et un mode exercice.

5.5.1.1.1. Phase 1 : les programmes LOGO

L'objectif de cette première phase est d'aider l'élève à construire du sens. Première approche vers le traitement d'expressions algébriques (construction de signifiants à partir de signifiés et travail technique sur les écritures), elle comporte plusieurs types de projets : de l'anglais à l'algèbre, un générateur de phrases aléatoires comprenant trois boîtes *Qui*, *Faitquoi* et *A_qui*, en quelque sorte des machines sources pour des constructions plus complexes ; des dialogues ; des énoncés de problèmes arithmétiques en langue naturelle ; des problèmes algébriques posés comme des *colles* à résoudre.

Pour ce dernier type de projet, les élèves ne sachant pas encore résoudre des équations du genre $5 \times \text{BOX} + 4 = 49$, l'énoncé est trouvé en remplaçant un nombre par une inconnue dans une égalité. Par exemple $3 \times 7 + 4 = 25$ génère le problème $3 \times \text{BOX} + 4 = 25$. Les messages peuvent être juste BON ou FAUX puis inclure une vérification du style : 6 n'est pas bon car $3 \times 6 + 4 = 22$ et 22 n'est pas 25. Ensuite, il s'agit de suggérer aux élèves l'idée qu'il existe une procédure formelle de résolution en prenant des formes simples (se résolvant à l'aide d'une opération réciproque) : $\text{BOX} + 3 = 10$, $4 \times \text{BOX} = 20$...

Les tests effectués montrent l'intérêt de la phase de mélange des phrases algébriques et anglaises et le fait que les élèves arrivent effectivement à acquérir le niveau de compétence requis en programmation LOGO. Feuerzeig cite l'exemple d'un élève ayant découvert par lui-même l'algorithme et le présentant lors d'une réponse fautive en commençant par « *Une façon facile de trouver la réponse est ...* », comme dans certains textes arithmétiques anciens, dans lesquels l'auteur dévoile ses *trucs*.

5.5.1.1.2. Phase 2 : les sacs de billes

Un sac représente le nombre à trouver et les billes représentent chacune une unité. La figure 7 montre les étapes successives d'un problème. On choisit d'abord un nombre, on le multiplie par 2, puis on ajoute 3, on soustrait 2, enfin on multiplie le résultat obtenu par 2 et on obtient 50. En reprenant les transformations réciproques dans l'ordre inverse, on obtient aisément la solution de ce petit problème.

Les fonctions sont vues comme des boîtes noires. L'idée de faire un chemin à l'envers (faire les actions opposées dans l'ordre inverse de l'ordre original) est quelque chose qui fonctionne aussi pour un chemin avec la tortue. Les expérimentations montrent qu'avec un tel environnement, les apprenants sont capables de construire des histoires complexes. Il s'agit pour eux de construire des machines et de les connecter entre elles. Cette représentation n'est cependant pas très adaptée pour le traitement des quantités négatives¹⁰.

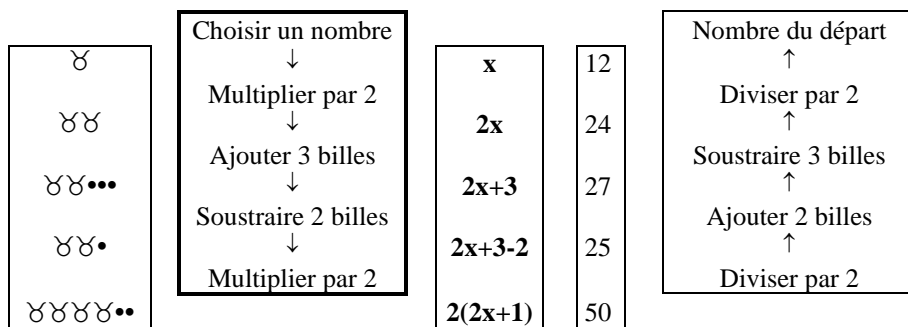


Figure 5.7 Résolution d'une équation avec les sacs de billes

5.5.1.1.3. Phase 3 : le tuteur algébrique

Dans cette dernière phase, l'apprenant travaille avec un banc d'essai algébrique (*Algebra Workbench*). Ce dernier incorpore un expert du domaine, c'est-à-dire un programme capable de résoudre des problèmes algébriques exprimés dans les notations formelles standards. Deux modes sont possibles, le mode élève et le mode tuteur. Dans le mode élève, le contrôle est entièrement laissé à l'apprenant. Il peut choisir lui-même un problème ou chercher à résoudre un exercice proposé par le système. Il travaille en pas-à-pas, appelant éventuellement l'expert algébrique pour effectuer les calculs. Il peut résoudre entièrement un problème sans faire appel à l'expert algébrique ou *l'invoquer* pour vérifier ou faire et expliquer une étape. Dans le mode tuteur ou démonstration, il peut demander à l'expert de résoudre complètement un problème.

¹⁰ Un travail équivalent peut être effectué sur un système de balance avec Poids non marqué / Poids marqués et des opérateurs permettant d'ajouter ou d'enlever un poids marqué ou non marqué. On garde l'état de la balance (équilibré, penche à gauche, penche à droite). Pour les poids négatifs, on peut imaginer un système de ballons ou de poulies pour tirer en sens contraire. D'autres représentations sont aussi imaginables.

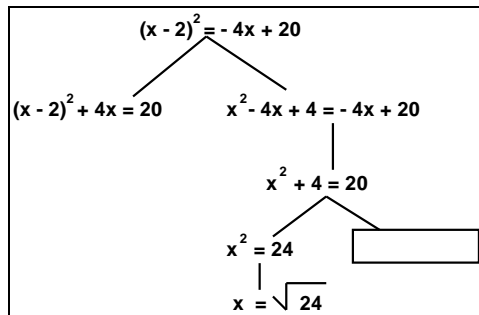


Figure 5.8. Un arbre de résolution

Cet exemple d'écran du Banc d'essai algébrique (Feurzeig, 1986) permet d'avoir une idée du plan de travail de l'apprenant.

L'affichage de l'arbre de résolution permet d'explorer plusieurs pistes et fournit une bonne représentation (réification) de la trace de la résolution, en incluant les étapes qui n'ont pas été poursuivies.

En résumant la démarche adoptée par Feurzeig, les micromondes s'attachent aux problèmes de sens alors que le travail avec l'expert algébrique s'intéresse aux manipulations algébriques, de nature plutôt syntaxiques. Les tuteurs experts et les micromondes sont complémentaires, aucun ne pouvant supporter à lui seul la tâche d'enseignement.

Dans le chapitre consacré aux micromondes, nous avons vu que beaucoup de chercheurs préconisaient une suite de micromondes pour accompagner l'apprenant au cours du processus d'apprentissage. Dans le cadre de la première version du système MEMOLAB (dédié à l'acquisition de compétences de base en psychologie expérimentale), Dillenbourg et Mendelsohn (1992) développent une analyse similaire. D'un côté, l'approche behavioriste se concentre sur l'analyse du contenu et dans sa décomposition en éléments atomiques. De l'autre, l'approche constructiviste engage l'apprenant à explorer des problèmes complexes, ce qui ne fonctionne pas avec des environnements non structurés. Sur la base des théories développées par Robin Case, ils proposent une analyse du contenu dans une granularité intermédiaire avec une suite de micromondes intégrés hiérarchiquement. Le premier permet au novice de résoudre des problèmes très simples. Avec les suivants, il est amené à résoudre des problèmes de plus en plus complexes mais avec des opérateurs de plus en plus puissants. L'utilisation de ces opérateurs implique l'intériorisation des concepts principaux qui constituent l'expertise du domaine. Chaque micromonde est défini par l'association d'un langage de description et d'un langage de commande. Cette association spécifie la relation entre des niveaux successifs : l'ancien langage de description devient le nouveau langage de commande.

Les deux exemples précédents montrent que la conception de systèmes évolutifs invite à définir des paliers successifs et que les différentes phases peuvent correspondre à des environnements de nature très différente. La question se pose néanmoins de savoir s'il est possible de concevoir un enseignement pris entièrement en charge par un système tuteur intelligent.

5.5.1.2. Problèmes dans la conception des tuteurs intelligents

Force est de constater que la construction de tuteurs intelligents sur la base de l'architecture en composantes ou modules se heurte à divers types d'obstacles. Goodyear (1991) mentionne deux doctrines opposées. La première considère que si le champ est encore trop jeune pour construire des tuteurs intelligents complets, on peut apprendre énormément, tout au moins pour le moment, en étudiant des morceaux de tels tuteurs intelligents. De nombreux chercheurs ont ainsi travaillé en profondeur sur certains aspects du problème. Pour Woolf (1988), quatre types de connaissances, ingrédients fondamentaux de tuteurs intelligents, ont été analysées avec succès : le domaine, les théories cognitives, le tutorat et la communication.

L'autre doctrine souligne que l'intégration et l'interdépendance sont centrales, les connaissances partielles nous renseignent peu sur la manière de tout recoller ensemble, la nature de chaque élément étant fortement déterminée par sa relation aux autres. Trois grandes classes de problèmes restent globalement ouverts : les problèmes spécifiques à chacun des différents modules, l'articulation entre ces modules et l'intégration dans les organisations actuelles de la formation.

Concernant chacun des modules, si, comme le souligne Woolf, de nombreuses avancées ont été opérées, il reste encore un immense travail à accomplir. Sur l'expertise du domaine, il n'est pas sûr qu'une représentation permettant à un ordinateur d'accomplir une tâche soit adaptée à enseigner à un être humain à accomplir la même tâche. L'existence même de résolveurs, au lieu d'apporter une solution, déplace les problèmes à résoudre, puisque le résolveur lui-même devient objet d'enseignement. Ces résolveurs sont au demeurant largement incomplets. Nous avons déjà abordé les difficultés de la modélisation de l'élève et il faut bien convenir du fait que l'expertise pédagogique reste très embryonnaire. La travail autour d'AMALIA a notamment montré la complexité inhérente à l'écriture de plans pédagogiques (Vivet et al., 1988 ; Vivet et Bruillard, 1994). Il apparaît que les enseignants humains ne sont pas des modèles très riches sur le préceptorat.

Même si localement, on arrive tant bien que mal à des choses significatives sur chacun des modules, leur coopération est loin d'être assurée : on a l'impression d'être en face d'un jeu de MECCANO dont les pièces ne s'emboîtent pas correctement. Des réussites locales ne sont ni garanties ni forcément des amorces de solutions plus globales. Il n'est pas simple de préciser les contraintes sur chacun des modules pour qu'ils puissent communiquer avec les autres. Il est tout aussi difficile de savoir ce qui est dépendant et ce qui est indépendant alors que les liens entre les modules apparaissent multiples. Cette fausse indépendance des différents modules ou experts, a des répercussions sur la pertinence des environnements proposés : des formes d'apprentissage ne seront elles pas favorisées de par leur plus grande facilité de mise en oeuvre technique ?

Enfin, l'intégration de tuteurs artificiels dans les organisations de formation existantes semble loin d'être aisée. Il n'apparaît pas concevable, à l'heure actuelle, de confier la gestion totale d'un apprentissage quelconque à des machines. La complémentarité avec d'autres modes d'enseignement est donc à trouver. Vivet (1991) conseille ainsi de prendre en compte l'enseignant humain et de lui confier un rôle complémentaire à celui des tuteurs artificiels. Tout comme l'intérêt de

l'intelligence artificielle s'est déplacée des applications de substitution (remplacer l'homme par la machine) aux applications d'assistance à l'opérateur humain, le problème est bien d'intégrer ces divers outils dans la démarche globale, non de faire assurer à une machine l'ensemble du processus d'enseignement. L'évolution conduit d'ailleurs plus vers des systèmes analogues aux SIAD (systèmes interactifs d'aide à la décision ; Levine et Pomerol, 1989), c'est-à-dire des systèmes interactifs intervenant dans les processus d'enseignement et d'apprentissage. Si les systèmes tuteurs intelligents ont certainement un rôle à jouer dans des contextes limités, dans des phases spécifiques de l'apprentissage, des changements de paradigmes amènent à s'intéresser à d'autres formes de travail avec l'ordinateur.

5.5.2. *Quel(s) rôle(s) pour la machine ?*

Dans le chapitre II, nous avons brièvement décrit une classification des applications des ordinateurs à des fins éducatives, au début des années soixante-dix. Au début des années quatre-vingt, les grands courants coexistent. D'un côté, l'EAO a supplanté l'enseignement programmé, les grands projets ont été lancés. Les techniques d'intelligence artificielle commencent à être connues et LOGO est à son apogée. Les micro-ordinateurs ont fait leur apparition depuis la fin des années soixante-dix. Une classification proposée par Taylor (1980) s'appuie sur les grands types de rôle dévolus à l'ordinateur. Il distingue trois classes principales nommées *Tool*, *Tutor* et *Tutee*¹¹, ce qui correspond à l'ordinateur :

- *outil*, qui s'intègre à des tâches ou des situations ayant un intérêt éducatif, l'exemple le plus répandu est celui du traitement de textes, mais on peut citer les outils de calcul, les programmes d'analyse statistique...
- *enseignant*, qui contrôle les séquences d'apprentissage, c'est le cas des tuteurs intelligents succédant à l'enseignement programmé et à l'EAO.
- *apprenant*, qui est sous contrôle de l'élève, l'apprenant enseigne en quelque sorte à la machine. C'est le cas de l'approche LOGO.

5.5.2.1. *La machine n'est plus uniquement vue comme un précepteur*

Dans cette classification, la distinction entre ce qui réfère à l'ordinateur outil et à l'ordinateur apprenant n'est pas toujours très claire. La barrière peut être celle qui existe entre les créateurs et les consommateurs. L'outil peut tout aussi bien servir dans le cadre d'un travail de routine que participer à une expérience créatrice. Nous en avons déjà parlé au chapitre IV en ce qui concerne les environnements d'apprentissage ouverts. Néanmoins, si cette classification recouvrait bien les applications du début des années quatre-vingt, l'évolution de la notion de tuteur intelligent va conduire à la définition de nouveaux paradigmes. En effet, le rôle de l'ordinateur vu comme un enseignant autorisé a été remis en question (Chan 1993). Self (Gilmore et Self 1988) et d'autres chercheurs (Cabrol et al., 1987) ont suggéré de lui confier plutôt un rôle de collaborateur (ou de co-apprenant). Chan et Baskin ont proposé que l'ordinateur puisse simuler deux agents co-existant, un enseignant et

¹¹ Un quatrième rôle peut être ajouté, c'est celui du *Jeu*. Taylor l'avait signalé sans toutefois l'inclure, jugeant qu'il pouvait s'intégrer dans les trois rôles déjà définis.

un compagnon d'apprentissage (dans la métaphore de l'apprentissage avec le prince), conduisant à la fois à des formes de coopération et de compétition. L'ordinateur modélise un apprenant artificiel et les étudiants humains profitent de la coopération, de la compétition ou du fait d'enseigner à un apprenant artificiel. En sus, l'ordinateur peut modéliser des agents multiples ayant des rôles différents, formant une petite société ou un contexte social¹². C'est ce contexte social qu'il semble important de retrouver alors qu'il était oublié dans les formes d'interaction plus classiques.

5.5.2.2. *De l'individuel au collectif*

Les idées de construction sociale ont certainement eu un impact sur l'informatique et l'intelligence artificielle distribuée. L'utilisation, même très limitée, des systèmes tuteurs intelligents dans des classes a permis de se rendre compte du poids essentiel de la dynamique sociale dans l'école et dans des communautés plus larges (Sack et al., 1992).

L'aspect coopératif prend une place grandissante. Les interactions entre les élèves, l'enseignant et le système sont prises en compte. On conçoit une sorte d'espace d'apprentissage incluant des agents humains et des agents artificiels. Les architectures multi-agents se multiplient. Si les apprenants coopèrent entre eux, les programmes, conçus comme des ensembles d'agents, s'intègrent dans des architectures de plus en plus distribuées (SIGCUE, 1992). L'usage d'agents intelligents (Kearsley, 1993) introduit des problématiques nouvelles, renouvelant notamment certaines conceptions des interfaces. C'est ainsi que Kay (1990) défend l'idée de diriger indirectement des agents plutôt que manipuler directement des objets. Boy (1996) décrit des agents comme dénotation, définition, analogie, question suggérant une piste, résolution de problèmes (décomposition en sous-problèmes), gestion de la vidéo, évaluation, liens hypermédiés à un autre document actif. Tout ceci conduit à un relâchement du contrôle, traduit par le rôle de partenaire ou de collaborateur confié à la machine.

5.5.2.3. *Des modes de contrôle plus souples*

A la suite de l'enseignement programmé, les analyses se fondent sur les rapports entre l'enseignement, l'apprenant et la matière à acquérir. Une certaine forme de mécanisation est recherchée, l'objet étant de déceler des mécanismes d'apprentissage et de chercher une stratégie d'optimisation de ces mécanismes sur une matière déterminée. Avec le développement de la micro-informatique, on passe de la salle machine, de la ressource rare et *sérieuse* à l'outil individuel. Dans ce cadre, la nécessité du contrôle pour l'optimisation de l'apprentissage se fait moins impérieuse. On en déduit deux notions clés : respect du temps (considérer plus le temps d'apprentissage que le temps de l'enseignement) et la notion d'outil personnel. On résout soi-même directement des tâches avec la machine. L'outil bureautique peut être utilisé aussi pour l'apprentissage. On peut plus facilement apprendre en faisant, en résolvant des problèmes, sans être comptable de ressources qui peuvent rester plus longtemps disponibles, à l'aide d'outils de différents types.

¹² Le sens vient des situations d'usage par un processus d'abstraction. La signification est négociée et apprendre nécessite la coopération et la critique des pairs.

Une question essentielle est de savoir qui détient le contrôle du processus d'apprentissage. D'un contrôle rigide à un contrôle implicite et à des formes de planification dynamique, l'aspect proprement préceptoral tend à diminuer. Laisser largement le contrôle à l'apprenant apparaît intéressant. La découverte guidée, prônée depuis longtemps (notamment dans le travail autour de SOPHIE) est vue comme un moyen d'opérer une synthèse entre tuteur intelligent et micromonde (Elsom-Cook, 1991). En effet, les expérimentations ont montré la nécessité d'une assistance auprès de l'apprenant. Ajouter un guidage à un processus de découverte permet de faciliter le processus. Les modes d'assistance proches de l'interaction didactique discrète (entraîneur) fournissent des solutions intéressantes. L'apprenant « patauge », le système l'aide en rendant ses démarches explicites : assistance discrète (coaching) pour l'aide à la découverte ou à la compréhension de meilleures stratégies métacognitives basées sur leur expérience. Des environnements d'apprentissage supportant certaines formes d'apprentissage par la découverte guidée (Hennessy et al., 1989) sont ainsi proposés.

L'apprentissage peut aussi être négocié, cette négociation entre l'apprenant et le système portant sur les croyances ou sur les objectifs d'apprentissage. Baker (1994) propose un modèle de négociation basé sur l'analyse des dialogues entre un enseignant et un élève et entre des élèves.

Dans la métaphore du compagnonnage (*apprenticeship*, Collins et Brown 1986 ; Newman, 1989), l'apprenant réalise la tâche avec la guidance, l'assistance et l'évaluation de la machine. Il s'agit d'une forme d'apprentissage par l'exemple, l'apprenant peut comprendre l'approche suivie par un expert en observant des exemples de résolution experte et en voyant comment les actions sont interprétées dans le cadre de la compréhension de l'expert. Les instructeurs humains font partie du processus d'entraînement en interprétant les rétroactions et en suggérant de nouvelles pratiques. Le pouvoir est dans les mains de réels instructeurs et des élèves. Ce mode d'apprentissage correspond à une façon classique de développer de l'expertise. Elle est fondée sur l'étude des limites du travail à partir de l'expertise. Si les travaux en intelligence artificielle se sont beaucoup focalisés sur la problématique du transfert entre l'expert et le novice, l'histoire de GUIDON montre le caractère peu opératoire de d'une démarche non intégrée et les limites du contrôle avec un résolveur insuffisant. Toutefois, la vision pragmatique du compagnonnage met en lumière les difficultés principales de l'introduction de ce type de démarche : compréhension de l'environnement, de ce que fait et que l'on peut faire avec la machine, compétence des instructeurs et la définition de leur rôle. L'apprentissage qualifié de situé est proche du compagnonnage de par sa focalisation sur le contexte dans lequel l'apprenant va utiliser les connaissances.

Le rôle grandissant des outils et les connaissances qu'ils embarquent introduit de l'intelligence dans les environnements de travail proposés aux apprenants. Dans un apprentissage basé largement sur la résolution de problèmes, l'ordinateur fournit divers outils de traitement et d'aide sur l'ensemble du processus ou sur certaines étapes. L'ensemble du continuum entre les tuteurs et les environnements d'apprentissage est balayé, et les divers paradigmes d'interaction élève/machine tels

le dialogue socratique, le compagnon d'apprentissage, l'apprentissage coopératif, le compagnonnage ouvrent un large spectre d'usage.

Le projet initial de l'enseignement programmé et de l'EAO visait à mettre l'enseignant dans la boîte, ce qui revient à se passer de l'enseignant. Les micromondes et environnements d'apprentissage confient un rôle différent au "maître", mais un rôle essentiel (animateur, gestionnaire de ressources, ...). On est ainsi passé de systèmes adaptatifs conçus dans une perspective tutorielle à une perspective d'apprentissage coopératif incluant des composantes tutorielles. Les systèmes EIAO peuvent ainsi être vus comme des systèmes coopératifs particuliers, cette coopération étant souvent multiple : entre un ou plusieurs élèves et le système, entre les élèves et l'enseignant, entre l'enseignant et le système... Cette vision amène une prise en compte de l'enseignant lui-même (Vivet 1988, 1990), souvent exclu dans les perspectives tutorielles strictes, la machine jouant le rôle de précepteur. L'ordinateur assume de multiples rôles : partenaire, co-apprenant, assistant, aide, guide, collaborateur, outil, environnement d'apprentissage, expert du domaine, etc. Il ne s'agit plus de modéliser l'apprenant mais plutôt de modéliser l'interaction.

5.5.3. Les systèmes d'aide et les systèmes conseillers

Une des idées développées dans le cadre de l'apprentissage situé est d'éviter une trop grande séparation entre formation et utilisation dans un contexte de travail. Dans des domaines déjà informatisés, s'appuyant sur ou étant directement liés à l'informatique (enseignement de la programmation, utilisation d'outils généraux : traitement de textes, publication assistée par ordinateur, logiciels de gestion, etc.), l'aspect apprentissage proprement dit se confond dans l'interaction même avec les autres aspects inclus dans l'environnement de travail (documentation, aide, explication). Ainsi, le problème des tuteurs se situe dans le cadre plus général de la communication homme-machine.

D'un autre côté, il y a souvent des difficultés dans l'usage des diverses fonctions d'un logiciel, les concepts associés étant mal maîtrisés, nécessitant des interventions tutorielles. Les différences entre ces dernières et les systèmes d'aide proviennent alors surtout du but poursuivi et de l'organisation globale.

5.5.3.1. Apprentissage et systèmes d'aide

La parenté entre les systèmes d'aide et les systèmes tutoriels est patente. On peut aller d'une simple rétroaction, à une aide interactive intelligente, des conseillers actifs jusqu'aux tuteurs.

On différencie les systèmes d'aide dits passifs (interface par touches ou langage naturel), qui attendent les sollicitations de l'utilisateur et les systèmes d'aide actifs qui observent l'activité de l'utilisateur et sont susceptibles d'intervenir sans requête de ce dernier. Dans un mode passif, l'aide est avant tout réactive, elle ne fait qu'offrir des réponses aux demandes de l'utilisateur. Dans un mode pro-actif, le système observe l'utilisateur et intervient pour fournir des suggestions afin de l'aider à

réaliser son but. Enfin, un mode tutoriel est souvent approprié pour faire découvrir un système à un néophyte.

Un système d'aide pro-actif correspond à un système d'intervention discrète tel WEST ou WUSOR (voir chapitre III). La difficulté est de définir des principes cohérents afin de choisir le moment et le contenu d'une intervention au cours d'une session avec l'utilisateur. Il faut l'aider à résoudre un problème courant et peut-être, au-delà, lui enseigner les connaissances et méthodes sous-jacentes qui peuvent être jugées importantes. Dans une idée de synthèse entre micromondes et tuteurs, s'agissant d'associer une certaine forme de guidage dans les environnements ouverts pour offrir un apprentissage par la découverte guidée, il faut pouvoir combiner l'exploration et un système d'aide actif (Hennessy, 1989).

L'aide peut porter sur le lien entre la tâche à accomplir et l'environnement informatisé, ainsi que sur le domaine de la tâche (par exemple, notions de statistiques ou de comptabilité). Contrairement aux tutoriels, les systèmes d'aide ne peuvent pas être structurés à l'avance, mais doivent *comprendre* les contextes spécifiques dans lesquels l'utilisateur demande ou a besoin d'aide. Du côté des usagers, ceux-ci doivent apprendre à interagir avec ce nouveau médium et se construire un modèle conceptuel du programme avec lequel ils opèrent. Ainsi, un tableur nécessite une acclimation pour réaliser les tâches originelles d'une façon nouvelle. Cela change la façon d'interpréter ces tâches, il faut intérioriser la technologie, avec de la formation ou de l'assistance pour établir un contexte d'usage de l'application, apprendre comment mener à terme ses buts avec l'application et acquérir les compétences nécessaires à la performance.

L'extension d'usage des outils et progiciels généraux amène un problème additionnel. Pour effectuer différentes tâches intégrées dans une session d'apprentissage, l'apprenant peut avoir recours à ces outils alors qu'ils ne sont pas strictement conçus à cette fin. Une première solution consiste à reconstruire des progiciels incluant les fonctionnalités requises pour bâtir un environnement d'apprentissage. C'est le cas d'APPAT (Brouaye et al. 1987), système multi-tableur avec lequel des applications sur la comptabilité et l'étude des suites numériques ont été développées. Si une telle approche donne un contrôle maximal, elle présente deux défauts majeurs, une grande lourdeur de développement et un manque de généralité. Il n'est pas raisonnable de réécrire les logiciels applicatifs nécessaires pour les insérer de manière plus complète dans un tuteur ou un environnement d'apprentissage¹³.

5.5.3.2. Les systèmes conseillers

¹³ Le problème est identique à celui de la création de logiciels d'apprentissage des mathématiques intégrant du calcul formel. Vu l'offre actuellement disponible et le manque de boîtes à outils disponibles, il n'apparaît plus possible de créer des logiciels sans tenir compte des outils existants et diffusés. C'est la démarche suivie par Vivet (1984) dans CAMELIA conçu comme une sur-couche du système de calcul formel REDUCE. Le problème est celui de la création de telle sur-couche et des types de communication avec le (ou les) logiciel(s) applicatif(s) sous-jacent(s).

La création de systèmes conseillers semble par contre une voie de recherche prometteuse. Boulet et son équipe (Boulet et al., 89) ont ainsi développé un logiciel générique pour créer et implanter des systèmes *conseillers*¹⁴ en général et ont déterminé des méthodes pour l'application d'un tel système à des domaines spécifiques et aux requêtes des utilisateurs.

Le travail d'un système conseiller consiste à observer le comportement d'autres systèmes et à raisonner sur les actions observées sitôt qu'elles sont lancées. Son rôle est de produire des conseils à partir de l'espionnage de l'interaction entre un utilisateur et un environnement d'apprentissage, conseils portant sur le contenu du domaine d'étude ou sur des méthodes permettant de résoudre une classe de problèmes. L'objet n'est pas de faire un guidage en pas-à-pas mais plutôt de suggérer des méthodes à utiliser pour faire progresser la démarche de l'apprenant.

Paquette et son équipe ont développé plusieurs systèmes conseillers. Un des premiers systèmes nommé COPERNIC (Paquette, 1991) est dédié à l'induction de lois physiques. Créé au moyen d'extensions au système PRISME/LOUTI, il inclut des heuristiques adaptées des programmes BACON modélisant l'activité de chercheurs qui induisent des lois à partir de données expérimentales. La trop grande intégration du module conseiller au système LOUTI a conduit à l'élaboration d'un système générique nommé EpiTalk (Paquette et al., 1994 ; Giroux et al., 1996). Il s'agit de pouvoir greffer un système conseiller à toute application dotée d'outils d'aide à la tâche, sans en perturber le fonctionnement. Le système conseiller est ici qualifié d'épiphyte dans le sens où il est indépendant du système hôte dont il observe le fonctionnement, son comportement étant celui des plantes épiphytes qui vivent par-dessus des plantes existantes sans perturber leur fonctionnement normal. Un tel conseiller peut intervenir aussi bien sur le plan du contenu que de la méthode dans des scénarios d'usage individuel ou collectif. Les premières réalisations techniques et les premières applications semblent montrer qu'EpiTalk permet de développer rapidement des systèmes conseillers opérationnels. L'une des limites est l'obligation de découper l'activité en une hiérarchie de tâches, ce qui suppose une activité suffisamment organisée et planifiée.

De tels systèmes conseillers généralisent les premiers travaux sur l'interaction didactique discrète (les « entraîneurs ») tels WEST et WUSOR. Ce mode d'interaction apparaît approprié dans de nombreux contextes. Ainsi, Van Lehn (1996) décrit des formes de résolution de problème tutorées correspondant à des sortes de cours particulier autour de la résolution d'ensembles d'exercices. Les tâches de résolution sont élémentaires (minimales) et peuvent être assistées par des techniques du type de celles utilisées par Anderson (traçage de modèle). Mais, par derrière, de nombreuses règles sont susceptibles de fournir des explications et les justifications que les étudiants doivent connaître. On dispose ainsi d'une forme d'aide conceptuelle éventuellement complétée par une aide sur le processus même suivi par les étudiants afin de les inciter, le cas échéant, à modifier leurs techniques d'étude.

¹⁴ Le terme anglais est *advisor*. Les québécois utilisent le terme d'*aviseur*.

Bien d'autres environnements accordent une place centrale à l'aide et au conseil. SCHNAPS (Blondel, 1996), environnement d'apprentissage de la résolution de problèmes de chimie quantitative, se consacre aux problèmes de diagnostic et d'aide. Fondé sur une analyse de situations d'interaction entre un élève et un enseignant, l'aide est décomposée en assistants indépendants spécialisés ayant chacun leur propre logique de comportement. Sur la base de leur travail avec Sherlock II, système d'apprentissage par l'action sur le diagnostic de pannes de circuits électroniques, Katz et al. (1996) pensent, qu'après une période initiale de suivi relativement directif, des dialogues de conseil doivent guider l'étudiant afin qu'il réponde à ses propres questions plutôt que de lui dire ce qu'il faut faire.

Les exemples précédents tendent à montrer une certaine convergence vers des environnements ouverts privilégiant l'activité et la résolution de problème, incluant un guidage souple, pas trop intrusif mais effectif, assuré par un conseiller ou un ensemble d'assistants.

5.5.4. Points de débats

5.5.4.1. Les théories de l'apprentissage et de l'enseignement

Même si les positions respectives des différents chercheurs sont très diverses, une tendance forte, soutenue par la majorité d'entre eux, indépendamment de leur conception de la connaissance et de l'apprentissage, est de privilégier l'activité de l'apprenant. Le contexte d'apprentissage et les interactions sociales sont également reconnues fondamentales.

Si on fait le point, au début des années quatre-vingt-dix, sur les théories de l'enseignement et de l'apprentissage, on peut considérer qu'aucune théorie effective sur la façon dont les élèves apprennent ne s'est imposée. Néanmoins, les conceptions des chercheurs ont fortement évolué depuis les années soixante. La plupart d'entre eux se réclament dorénavant du constructivisme. Mais, comme le montre William Winn (1992), cette invocation recouvre des positions très éloignées. Si tout le monde s'accorde sur le fait que les apprenants construisent eux-mêmes leurs connaissances, les avis divergent quant aux implications sur l'enseignement de ce principe de base, pouvant remettre en question l'idée même d'une *technologie éducative*. Pour certains, la construction des connaissances requiert un peu plus que la seule adjonction de systèmes d'aide ou d'intervention discrète à des stratégies d'enseignement traditionnelles. Pour d'autres, la conception constructiviste implique que chacun d'entre nous connaît le monde d'une manière différente et qu'il n'est pas possible de penser concevoir un enseignement ayant des résultats prévisibles.

Le courant dit de la cognition située (JAIE 1993, vol.4, n°1) amène un changement de paradigme, en rupture nette avec les approches symboliques de la cognition. Selon ses tenants (Lave, Wenger, Clancey...), l'expérience est

primordiale, la réalité ne peut être objectivement décrite ou modélisée, les représentations prennent sens grâce à la perception¹⁵. Cette position induit un rejet des modèles au profit d'expériences d'apprentissage *signifiantes*. Comme l'affirme Clancey (1992), la connaissance est constamment créée et prend sens au travers des interactions sociales, ce n'est pas un processus qui se déroule exclusivement dans la tête des individus. Le problème est donc de concevoir des outils informatiques qui améliorent les processus quotidiens par lesquels les personnes construisent leurs propres connaissances des tâches et les méthodes qui structurent leurs comportements. On passe des tuteurs intelligents aux outils pour apprendre.

Plusieurs chercheurs (Jones et al., 1992 ; Dillenbourg, 94) opposent ainsi deux conceptions dominantes de la connaissance : celle de l'ingénieur manipulant la connaissance comme un objet et celle de l'anthropologue qui analyse les manifestations des processus cognitifs sous-jacents. Les différents modèles d'acquisition sont des modèles de transmission, de négociation, d'apprenti(ssage), d'exploration, de construction guidée. Aux deux extrêmes sont la transmission, avec la métaphore du récipient à remplir, et l'engagement signifiant de l'apprenant, vu comme un artisan.

Une question importante demeure : est-il possible d'assurer un apprentissage par instruction ? Après l'abandon de l'aspect normatif des recherches en enseignement programmé, de nombreuses investigations ont concerné les connaissances des enseignants experts. En codant leur expertise, on devrait pouvoir conseiller les enseignants novices sur bon nombre de situations traditionnellement rencontrées dans les classes.

A partir d'une étude sur le logiciel PC-Paint, Goodyear (1991, p.17) dresse deux premières constats sur les processus d'enseignement. Tout d'abord, il attire l'attention sur le besoin de conceptualisations de haut niveau. Beaucoup d'efforts sont en effet nécessaires pour négocier et clarifier la nature et le but de l'activité pendant qu'elle se déroule. Ensuite, il suggère de prendre sérieusement en compte la nature interactive de l'enseignement. On ignore encore les ressources de résolution de problèmes que cette interaction rend disponible et on conserve une vision trop opportuniste et située du problème.

Comme le souligne Jones (Jones et al., 1992), il n'y a aucune manière optimale d'enseigner à un ensemble d'élèves. En EIAO, aucune théorie ne permet de décider de l'enseignement approprié suivant le contexte et la situation éducative, s'appuyant sur un contrôle partagé entre l'élève et le système, intégrant l'idée de l'enseignant comme facilitateur¹⁶. S'il est sans doute important de développer une telle théorie de l'apprentissage par instruction, deux questions fondamentales restent

¹⁵ La représentation nécessite de manipuler quelque chose qui peut prendre forme et être ensuite perçu. Elle émerge comme une construction coordonnée dans un processus de généralisation d'actions.

¹⁶ Ce rôle de facilitateur est un rôle complexe de planification, de surveillance (*monitor*), de remédiation, d'intervention, de correction et de direction des activités des apprenants isolés ou des groupes.

posées : peut-on contrôler l'apprentissage ? Doit-on construire une théorie computationnelle de l'enseignement indépendante de la technologie ?

5.5.4.2. Concepts principaux

Ce qui précède montre un champ foisonnant, dont il est difficile de saisir les contours. Au-delà d'architectures spécifiques ou de choix généraux sur le type d'environnement à construire ou le type de rôle confié à l'ordinateur, certains concepts se retrouvent. Ces derniers illustrent les résultats et interrogations du domaine. Deux points méritent d'être soulignés : la notion d'interactivité et ce qui a trait à la connaissance et à ses représentations (réification, représentations intermédiaires, points de vue).

L'*interactivité* est un concept sans arrêt redécouvert au gré des innovations technologiques (au sens large). Elle se transforme à l'aide d'ajouts d'adjectifs, comme *interactivité signifiante* allant de la simple sollicitation jusqu'à l'idée d'engagement. La réalité virtuelle la prolonge avec l'idée d'*immersion*. Elle est associée souvent à la motivation, encore difficile à prendre en compte explicitement dans un système, voire aux jeux (idée d'entraîneur dans WEST et WUMPUS). Tout comme la nécessité de centration sur l'apprenant, elle est sans arrêt réaffirmée par les chercheurs (chacun essayant d'être plus *interactif* que le précédent !). On peut cependant remarquer que, dans le contexte de la recherche sur les environnements d'apprentissage, l'interactivité sous-tend avant tout l'idée d'une activité réelle avec une machine, qui ne peut se restreindre à un ensemble de choix parcimonieusement concédés à l'apprenant. Le fonds désormais commun de constructivisme sur lequel s'accordent les chercheurs amène à confier une place centrale à cette activité dont la machine peut être à la fois le lieu, l'assistant, le guide ou le partenaire. L'interactivité prenant un sens infiniment plus limité dans nombre de discours autour du multimédia, peut-être faudrait-il parler d'environnements hautement interactifs à propos des environnements d'apprentissage vus comme des compromis entre tuteurs intelligents et micromondes.

L'évolution de la vision de la connaissance et de ses représentations a sans doute été centrale dans le développement de l'EIAO. La cybernétique a introduit l'idée de *réification*. Celle-ci soutient la pensée réflexive, centrale dans le processus d'auto-évaluation et d'auto-amélioration. Elle souligne l'intérêt des processus métacognitifs favorisés, par exemple, par l'enregistrement de l'interaction entre le système et l'apprenant, disponible pour le système et pour l'apprenant. Les *représentations intermédiaires* facilitent l'usage de divers types de raisonnement. C'est le cas des représentations congruentes au raisonnement (*reasoning congruent*) qui encouragent l'apprenant à résoudre, le conduisent à travailler sur l'information contenue dans la solution et rendent l'information cachée visible. Les notions de *point de vue*, de *contexte* et de *granularité*¹⁷ (Greer, 1992) facilitent l'apprentissage de

¹⁷Pour Greer la notion de contexte peut être décrite comme un niveau de détail suivant lequel un concept peut être regardé. Ce niveau réfère à un niveau d'approximation/abstraction et au niveau d'agrégation auquel le concept ou l'objet en discussion peut être perçu. En décrivant les concepts de cette manière (suivant ces niveaux), un tuteur intelligent peut représenter les concepts comme des objets dans une structure hiérarchique appelée la

connaissances complexes. La connaissance étant parfois négociée pour favoriser la coopération entre l'apprenant et le tuteur (Moyse et Elsom Cook, 1992). les points de vue multiples sur un domaine et la représentation unique incluant des connaissances incertaines prennent une grande importance. L'intérêt de disposer d'une multiplicité de points de vue est d'encourager les étudiants à construire leur propre vision du domaine et de rendre ainsi externe leurs modèles de compréhension.

Enfin, la diversité des formes d'apprentissage amène une multiplicité des styles d'interaction, des systèmes d'aide (tuteur, conseil, aide), des formes d'exploration libre ou guidée, des apprentissages coopératifs, des formes de contrôle rigide ou implicite, vers la planification dynamique...

5.6. Bilan et perspectives

Clancey et Soloway (1990), dans un tour d'horizon sur intelligence artificielle et éducation, relèvent les principales tendances du domaine dans les années quatre-vingt-dix. Ils notent un large spectre d'alternatives aux dialogues de type tutoriel : des modèles précis et détaillés des étudiants pour la résolution de problème contraignant les possibilités de faire des erreurs (Anderson), des modèles auto-évidents (White et Frederiksen) évitant de recourir à des modèles de l'élève ou à des stratégies tutorielles difficiles à mettre au point, une liberté totale laissée à l'apprenant. Les efforts de recherche sont motivés par des raisons psychologiques et la recherche est fortement tournée vers les expérimentations. Les auteurs remarquent qu'il n'y a peut-être aucun autre domaine de l'intelligence artificielle dans lequel l'évaluation soit aussi importante et aussi difficile.

Malgré la grande diversité des approches suivies, des interrogations profondes demeurent sur les mécanismes d'apprentissage. Si l'enseignement programmé et le paradigme béhavioriste ont buté sur leur incapacité à raisonner dans le domaine de connaissance, sur l'enseignement et sur l'élève, les tuteurs intelligents butent encore sur leur incapacité à raisonner sur le processus de construction des connaissances par l'apprenant. Toutefois, l'enseignement demeure un champ d'expérimentation intéressant pour l'informatique et l'ordinateur un outil pour les chercheurs en éducation.

Nous avons évoqué, en introduction de ce livre, des interviews de chercheurs en intelligence artificielle et éducation (Sandberg et Barnard, 1993). Une des questions était centrée sur l'avenir du champ. Si presque tous sont favorables au développement des simulations, certains semblent abandonner l'idée même de tuteur intelligent, du fait de la difficulté de leur conception (Elshout) ou parce qu'ils

hiérarchie de granularité. Les transitions de niveau en niveau, correspondant à des changements dans l'instruction, la compréhension ou le point de vue, peuvent être représentés par des changements de focalisation dans cette hiérarchie.

considèrent qu'elle est basée sur des principes erronés (Schank¹⁸). D'autres, tel Anderson, défendent un point de vue opposé. Pour eux, les fondements sont corrects il est important de développer des tuteurs intelligents. La difficulté est réelle, mais le développement de bons outils devrait rendre leur conception possible et intéressante. Van Lehn, quant à lui, pense que le développement des tuteurs intelligents est souvent trop coûteux. Une tendance commune émerge néanmoins, l'abandon d'une vue individuelle de la cognition au profit de son insertion dans un contexte plus large.

Il apparaît que le thème des tuteurs intelligents, au sens strict, ne soit plus un sujet majeur dans le champ de l'Education et de la Technologie. Les tuteurs intelligents deviennent un outil de recherche pour l'étude des techniques d'intelligence artificielle ou des concepts théoriques sur l'apprentissage et l'enseignement. La tendance va vers des environnements d'apprentissage plus ouverts dans lesquels toutes sortes de technologies, intelligentes ou non, jouent un rôle.

Toutefois, le vieux rêve de création d'une machine à enseigner, qui a nourri les recherches de l'enseignement programmé, de l'EAO puis de l'EIAO, ne s'est pas éteint et l'apport de nouvelles techniques permet de le faire renaître. Plusieurs auteurs pensent qu'il est possible à des systèmes d'enseignement automatisés d'émuler les propriétés souhaitées des tuteurs humains dans un enseignement individuel de type préceptoral, de faire de l'instruction automatique (Regian et Shute, 1992). Néanmoins, si les premiers chercheurs ont porté leur attention sur la machine, il semble de plus en plus que l'intelligence soit à trouver dans l'interaction entre l'utilisateur et la machine. L'extension des capacités humaines est avant tout préconisée, dans la tradition des recherches sur les hypertextes qui font l'objet du chapitre suivant.

¹⁸Pour lui, le concept de système tutoriel est erroné. Nous n'avons pas à être le tuteur de quelqu'un.