Article publié dans les *Cahiers de la Fondation Archives Jean Piaget*
Université de Genève, p 75-96 (1991).

Les environnements intelligents d'apprentissage

Patrick Mendelsohn
TECFA
- Faculté de Psychologie et des Sciences de l'Education -
Université de Genève

La complexité technologique du monde dans lequel nous vivons fait que progressivement les recherches sur l'apprentissage par enseignement ont acquis, dans la communauté des psychologues, une importance aussi grande que celles sur l'apprentissage par induction (Bastien, 1989). Les premières concernent l'étude de la transmission de tous les acquis d'une culture qu'il n'est bien sûr pas nécessaire que chaque enfant reconstruise par lui même - Bruner (1964), Vygotsky (1978) ont largement insisté sur cette donnée du développement cognitif -. Les secondes s'intéressent aux apprentissages plus "solitaires" (au sens de moins social) puisqu'elles étudient les aptitudes que le sujet a, de par ses propres expériences sur le réel, d'extraire et d'élaborer de nouvelles connaissances. Si ce dernier type d'apprentissage a fait l'objet de nombreuses recherches en psychologie cognitive, il n'en est pas de même pour l'apprentissage plus "socialisé" des connaissances acquises en milieu scolaire. Il a fallu la tenacité de certains didacticiens (peut être trop orientés sur les contenus dans leur approche de l'apprentissage par enseignement) et surtout l'apparition et le développement des technologies éducatives pour provoquer un renouveau de la recherche dans ce domaine (Gagné, 1987). Les nouvelles technologies du traitement de l'information permettent, en effet, d'imaginer et de réaliser des scénarios pour assister et améliorer l'efficacité de l'apprentissage par enseignement. De plus, elles sont devenues un moyen privilégié pour élaborer un savoir pertinent sur l'apprentissage et la gestion des connaissances. Trois arguments peuvent être mis en avant:
- premièrement, la mise au point de systèmes informatiques dédiés à l'enseignement nécessite l'explicitation de contenus d'enseignement adaptés à l'apprenant;
- en second lieu, cette mise au point suppose aussi la gestion de l'interaction entre le système et l'apprenant (conception de l'interface);
- enfin, ces technologies permettent de construire des systèmes de recueil de données originaux sur l'apprentissage et ses conditions d'application.

La première partie de cet article m'amènera à préciser quelques points de vocabulaire à partir d'une représentation schématique des concepts les plus significatifs du domaine. Dans une seconde partie, j'aborderai un certain nombre de questions de recherches parmi celles qui sont discutées actuellement par la communauté scientifique. Je terminerai par une brève présentation d'un des projets de recherche de TECFA dont je partage la responsabilité scientifique avec P. Dillenbourg et D. Schneider: Memolab. Cette présentation me permettra surtout de montrer la difficulté et l'intérêt qu'il y a d'opérationaliser, sur un système concret, les questions théoriques que je vais détailler maintenant.

Environnements d'Apprentissage et Informatique

Classification des systèmes existants

Les environnements d'apprentissage implémentés sur ordinateurs peuvent être décrits sur deux axes orthogonaux: l'un caractérise l'environnement informatique, l'autre les apprentissages visés par le système. Le premier axe représente l'ouverture du système informatique relativement aux actions que le sujet peut réaliser sur l'environnement. Cet axe permet de situer un logiciel sur une dimension "ouverture - fermeture". Un logiciel "ouvert" permet à l'élève de manipuler une combinaison très large de commandes et de fonctions (un langage de programmation comme LOGO est un exemple de système "ouvert"). Un logiciel fermé, par contre, n'autorise que des choix restreints dans un domaine bien délimité; il repose sur le concept de dialogue interactif (par exemple, un questionnaire à choix multiples). Le second axe représente le type de connaissances sur lesquelles le logiciel est censé pouvoir apporter une aide du point de vue des apprentissages. On peut ainsi positionner sur cet axe, à une extrémité, les environnements d'apprentissage peu spécifiques dont on attend des effets sur les mécanismes les plus généraux du fonctionnement cognitif. Nous appellerons provisoirement ces aptitudes des métaconnaissances: planification de l'action, heuristiques de résolution de problèmes, etc. Sur l'autre extrémité de cet axe, on trouve les systèmes spécialisés construits pour permettre l'acquisition d'une expertise bien définie comme le calcul ou encore la géographie des pays d'Amérique latine. Ces connaissances ont la propriété d'être dépendantes d'un contexte spécifique.

Le produit cartésien de ces deux dimensions délimite quatre zones distinctes dans lesquelles il est possible de situer les principaux environnements d'apprentissage disponibles actuellement (figure 1):

< insérer ici la figure 1>

Les progiciels. Ces logiciels professionnels (traitements de texte, tableurs, gestionnaires de bases de données, logiciels de dessin) peuvent être considérés comme "ouverts" du point de vue informatique. Ce sont, en fait, des langages de commandes puisqu'il n'y pas de contrainte dans ces environnements du point de vue de ce qu'il est possible de rédiger, de classer ou de dessiner. Ils offrent, de plus, la possibilité de construire des macrocommandes similaires aux procédures d'un langage de programmation classique. Mais ce sont des logiciels spécialisés que la pratique pédagogique destine principalement à une activité et à un contenu précis: rédiger un texte, faire un schéma, résoudre un problème numérique, classer des objets, etc. Leur conception repose sur une analogie fonctionnelle avec ces tâches et l'apprentissage de ces logiciels se fait souvent par "imitation" et transfert analogique de compétence (ce qui n'est d'ailleurs pas sans poser des problèmes).

Les micromondes. Les environnements d'apprentissage, connus sous la terminologie de "micromondes", sont comme les progiciels, des systèmes informatiques ouverts; l'élève doit explorer un domaine, avec un minimum d'aide de la part du système, en combinant les primitives d'un langage de commandes. L'exemple le plus connu est, sans aucun doute, le monde de la "tortue Logo" dans lequel l'"explorateur" peut "apprendre", à un robot virtuel muni d'un crayon, à tracer toutes sortes de figures sur un écran. L'objectif assigné à cet apprentissage est ambitieux, l'élève apprend à "apprendre", il se sert de l'environnement pour "réfléchir" ses connaissances et construire de nouveaux objets. Pour les concepteurs de tels "mondes", ce sont bien des aptitudes de hauts niveaux, transférables à de nombreuses situations, qui sont visées par la maîtrise de cet environnement. La conception de ces systèmes est celle d'un langage de programmation et l'apprentissage de tels langages est de type "constructiviste". Il s'agit en effet pour l'apprenant de construire des objets de plus en plus complexes à partir de "schèmes élémentaires" disponibles dans un catalogue et d'une grammaire permettant de créer ces assemblages.

Les didacticiels. Ce sont les classiques logiciels d'enseignement assisté par ordinateur qui mettent en situation, plus ou moins interactive, un élève et un problème à résoudre. La gamme des activités possibles est assez vaste mais chaque séquence est fermée par un choix de réponses restreint à celles qui sont interprétables par le programme. Ces logiciels sont donc généralement considérés comme des environnements favorisant peu l'initiative de l'élève et sont très spécialisés dans leurs objectifs pédagogiques. La conception de ces environnements repose sur le dialogue interactif et l'apprentissage consiste généralement pour le sujet à mémoriser et à entraîner des séquences de procédures associées à certains concepts.

Les tutoriels. A la mise en situation de résolution de problème, caractéristique des didacticiels, certains concepteurs ont ajouté une composante tutorielle plus ou moins "intelligente" dans laquelle sont représentés le modèle de l'élève, le savoir-faire de l'enseignant et l'expertise du domaine à enseigner. Si l'ouverture de ces systèmes est encore limitée par les possibilités de dialogue "homme-machine" en langage naturel, la prétention de ces logiciels sophistiqués est de permettre des apprentissages de haut niveau sur le raisonnement, la logique et la compréhension par un guidage à initiative mixte entre le système expert "maître" et l'élève. La conception de ces systèmes est voisine de celle des aides "on-line" disponibles sur les logiciels professionnels grâce auxquelles l'apprentissage d'un dispositif complexe peut être optimisé par un guidage efficace.

Les environnements intelligents d'apprentissage (ILE)

A partir de cette présentation schématique - qui n'a probablement qu'une valeur "didactique" et non formelle - il est maintenant plus facile de situer ce que nous entendons par "Intelligent Learning Environment" (ILE). La terminologie pour désigner de tels environnements est déjà abondante. Elle comporte des dénominations telles que "intelligent micro-worlds" (Feurzeig, 1986) ou encore "articulated micro-worlds" (Frederiksen & White, 1988).

Le concept d'environnement intelligent d'apprentissage permet de dépasser l'opposition simpliste et manichéenne entre les défenseurs de l'apprentissage par induction (résultant des seules activités exploratoires du sujet) et les partisans des tutoriels inspirés (de près ou de loin) par l'enseignement programmé (caricature de l'apprentissage par enseignement). Ces systèmes doivent être capables de favoriser l'acquisition de concepts et de procédures associés à un domaine de connaissance. Il s'agit ni plus ni moins de construire ce que l'on pourrait appeler "une station de travail thématique" que chaque élève pourrait adapter à son goût et à ses besoins. En résumé:

1. Un ILE est un système qui réalise la synthèse entre, d'une part, les avantages de l'exploration libre et de la construction progressive des objets de connaissance (comme dans les micro-mondes classiques) et, d'autre part, l'intérêt du guidage propre aux systèmes tutoriels. L'idée centrale est de permettre à l'apprenant de transformer rapidement et efficacement ses expériences en connaissances organisées.

2. Un ILE est aussi un système qui privilégie l'idée que la meilleure façon d'apprendre c'est de se retrouver dans une situation (quasi) réelle de conception et de travail. Plutôt que de construire des logiciels orientés sur l'explicitation formelle de connaissances scolaires, nous pensons qu'il serait plus profitable de concevoir des outils et des environnements qui assisteraient l'élève efficacement dans les problèmes auxquels il doit faire face dans sa carrière d'apprenant (voir aussi Brown, 1989). Il apparaît que l'apprentissage incident de l'ensemble des connaissances nécessaires à la résolution d'une tâche pose moins de problème de motivation ou d'attention, si l'intérêt pour la tâche est assuré à un niveau élevé

3. Un ILE c'est enfin un logiciel qui permet d'entraîner un certain nombre de procédures de "calcul" pour acquérir les automatismes nécessaires à la résolution de problèmes récurrents (orthographe, opérations arithmétiques, décisions statistiques, etc). Il doit aussi prendre en compte les champs conceptuels du domaine et les définitions de termes pour donner à l'apprenant la possibilité de communiquer efficacement avec les spécialistes d'un champ de compétence.

Cette définition - obtenue par recoupement à partir des cases du schéma de la figure 1 - a laissé de côté deux concepts qui me paraissent importants et sur lesquels je voudrais apporter maintenant quelques précisions supplémentaires: il s'agit de la notion de guidage, et de celle d'"intelligence".

Le guidage. C'est un art subtil s'il en est un. Il existe tout une gamme de guides, mais ils ont tous au moins deux objectifs en commun (Wenger, 87):
1. Le pointage d'opportunités: ce rôle consiste à faire prendre conscience à l'utilisateur des possibilités non encore explorées du système au moment opportun pour l'intégration de ces nouvelles connaissances.
2. La transformation des échecs en expériences: le guide doit pouvoir détecter les erreurs de parcours par rapport à un objectif donné et proposer de nouveaux points de départ sans pour autant systématiquement empêcher le sujet de se tromper.

Pour Burton and Brown (1979), ces objectifs se traduisent par un certain nombre de règles de bonne conduite de la part du guide:
1. Interventions effectives pour éviter les effets "plancher" et "plafond" susceptibles de détruire l'intérêt pour la tâche à réaliser.
2. Les interventions doivent avoir lieu au bon moment et être judicieuses du point de vue de la tâche. Cela suppose un "monitoring" efficace de l'activité du sujet.
3. Le guide doit rester silencieux s'il "juge" son intervention non pertinente.

A propos du terme "intelligent" dans ILE. Il existe au moins trois manières de concevoir ici le terme intelligent. Ces acceptions n'étant pas exclusives, je penche plutôt pour une définition cumulative:
1. Ce terme se réfère tout d'abord à une technique informatique: "l'intelligence artificielle" dont le but est de reproduire des comportements que l'on a l'habitude de qualifier d'intelligents: raisonnement, démonstration, apprentissage, planification, enseignement. Dans le contexte où nous l'utilisons ici, cela suppose qu'un ILE repose sur un système à base de connaissances. Il peut prendre des décisions cohérentes relativement au domaine qu'il est censé enseigner et aux stratégies d'apprentissage qu'il utilise.
2. Ce terme peut aussi désigner les connaissances représentées dans le système informatique sous la forme de règles formelles de décision. En particulier celles qui concernent l'expertise pédagogique et le guidage dans l'espace des solutions que doit découvrir l'élève. Dans cette conception, un environnement d'apprentissage conduit inévitablement à "imiter" un tuteur

humain et à restituer la composante déductive et rationnelle de la pratique pédagogique. La performance du système dépend alors, en grande partie, de l'efficacité des formalismes informatiques de l'IA (systèmes de production, réseaux sémantiques, planification automatique...) et de la possibilité de représenter nos connaissances sur les processus d'enseignement dans de tels langages. Le modèle de référence pour cette composante de l'environnement reste le maître et sa compétence à guider efficacement l'élève.
3. On peut enfin entendre, sous cette terminologie, la référence à la qualité intrinsèque de l'environnement en tant que système capable de révéler et de stimuler des aptitudes spécifiques chez l'apprenant (réflexion, contrôle, planification). Dans cette perspective, les environnements d'apprentissage permettent au chercheur de disposer d'un véritable laboratoire pour étudier les transformations que l'on pourrait "artificiellement" apporter à la pratique pédagogique. Il est possible d'imaginer, par exemple, de donner à l'enseignant et à l'apprenant, la possibilité de contrôler des paramètres non habituels de l'apprentissage: charge cognitive en mémoire de travail, représentation structurée de l'histoire d'une session, aide à la planification, etc. Cette troisième acception du terme "intelligent" est un défi plus subtil pour les sciences de l'apprentissage puisqu'elle suppose que nous soyons capables de traduire ce que nous savons du fonctionnement et du développement cognitif du sujet dans l'architecture et l'interface du système. Nous rejoignons là une idée défendue aussi par Bastien (1989), qui pense que les tutoriels intelligents sont une méthode de recherche originale pour la psychologie cognitive. Cette démarche contraint à expliciter formellement les mécanismes cognitifs supposés être en jeu dans l'apprentissage par enseignement et l'efficacité du tutoriel devient alors un moyen de validation pour les mécanismes supposés.

Questions de Recherche

Quels modèles du fonctionnement cognitif choisir pour construire un environnement d'apprentissage?

Si, comme nous l'avons affirmé de façon péremptoire, un ILE est un laboratoire pour les sciences de l'apprentissage, il nous reste à choisir les modèles théoriques les plus aptes à une implémentation. Un exemple historique et réussi du pôle "tutoriel" nous a été donné par Anderson (Anderson and Reiser, 1985) et ses désormais célèbres "Geometry Tutor et Lisp Tutor" qui sont une implémentation de son modèle ACT* (Anderson, 1982). Un autre exemple, représentant les micromondes, nous a été proposé par Papert avec l'environnement LOGO. Ce dernier est un prototype remarquable des conceptions constructivistes du développement (Papert, 1980). En remontant le cours de l'histoire, on peut encore citer l'enseignement programmé (caricature du didacticiel), comme une fidèle réplique des conceptions behavioristes de l'apprentissage de Skinner. A ces archétypes qui représentent trois des axes de notre schéma, il faut ajouter tous les autres projets (parmi lesquels je situe aussi le nôtre) qui n'ont que la solution ingrate, mais combien nécessaire, de tenter des implémentations partielles de fragments de modèles. Je ne ferai pas ici une revue détaillée de la question, mais le lecteur intéressé peut se reporter à la synthèse de Wenger (1987) pour ce qui concerne les tutoriels et à l'ouvrage de Solomon (1986) pour les environnements classiques. Je me contenterai ici de mentionner les pistes qui me paraissent les plus prometteuses:

Le concept de "schéma". Son rôle dans les activités mentales supérieures (compréhension, résolution de problème, inférences, etc) a fait l'objet d'une synthèse claire et bien documentée de Richard (1990). L'intérêt de ce concept est qu'il permet une représentation facilement manipulable sur un écran graphique et que sa validité psychologique est aisément vérifiable. Le système TAPS (pour Training in Arithmetic Problem Solving) de Derry (1990) présente une utilisation originale de cette notion de schéma. La conceptualisation d'un problème (analogue du concept de représentation chez Richard) est pour cet auteur un sous-produit des stratégies de résolution de problème. Le simple fait d'activer un schéma produit simultanément une représentation et un moyen pour atteindre le but. C'est justement en raison de cette interaction que l'ordinateur s'avère être un bon outil de conceptualisation dans l'enseignement. L'ordinateur permet d'apporter la "colle" pour mettre ensemble des blocs de connaissances isolés. Les métaconnaissances sont ici définies comme l'ensemble des moyens dont le sujet dispose pour élaborer de nouveaux schémas à partir de schémas "primitifs".

Le concept de "chunking". Un autre concept intéressant, celui de "chunking", est souvent présent dans les logiciels professionnels. Une version minimale de ce mécanisme est offerte par la possibilité de construire des macrocommandes à partir de séquences d'instructions élémentaires. Cette option est maintenant un standard du marché sur presque tous les logiciels. Elle est souvent fournie avec une procédure d'apprentissage, qui permet de saisir la séquence pendant sa première exécution. Ce mécanisme d'unitarisation des connaissances n'a cependant pas fait l'objet d'une implémentation explicite dans les environnements d'apprentissage (sauf sous la forme du concept de procédure en programmation). J'ai moi même proposé une version développementale de ce processus dans le projet "Schèmes Informatiques Programmables", en m'inspirant des travaux de Case (Case, 1985; Mendelsohn, 1988). Une version plus élaborée de ce même principe est mise en oeuvre dans notre groupe (Dillenbourg, Hilario, Mendelsohn et Schneider, 1990) sous l'appelation "language shift"; elle sera présentée brièvement dans la troisième partie de cet article.

Importance de la méthode d'instruction dans l'évaluation des effets d'un environnement d'apprentissage.

Comme nous l'avons déjà mentionné, on a pensé pendant longtemps que les micromondes et plus généralement les environnements d'apprentissage du type "simulation" avaient la propriété de favoriser per se des apprentissages transférables à de nouveaux domaines (Papert, 1980). En fait, il s'avère que le contexte pédagogique dans lequel est utilisé le logiciel - en général, la méthode d'enseignement - est plus important pour obtenir des transferts significatifs que la qualité intrinsèque de l'environnement. Plusieurs auteurs (Kurland & Pea, 1984; Lehrer R., Guckenberg T. & Sancilio L, 1988; Littlefield J., Delclos V.R., Lever S., Clayton K., Brandsford J. & Franks J., 1989) ont ainsi montré que, si l'on s'intéresse aux transferts d'apprentissage d'aptitudes cognitives de haut niveau, les méthodes non structurées (prônées par les partisans de l'exploration libre) avaient des résultats moins bons que les méthodes structurées (mediated learning, situated learning, integrated learning) . Ce résultat est dû au fait que ces environnements ne contiennent aucune structure de contrôle permettant de mettre en relation les connaissances antérieures du sujet et celles qu'il est en train de construire. Cette mise en relation doit donc être - implicitement ou explicitement - prise en charge et organisée par la méthode d'enseignement. Dans un ILE, c'est à la composante "tuteur" et à l'interface de prendre en charge ce contrôle.

Le problème du transfert des connaissances: Que doit-on attendre des métaconnaissances ?

Les recherches en psychologie cognitive nous montrent que cette question est un domaine où l'intuition peut facilement être prise en défaut (Kotowsky, Hayes & Simon; 1985). Chaque fois que ce problème est posé, en particulier dans le cas des apprentissages scolaires, il convient de garder à l'esprit que les acquisitions sont organisées hiérarchiquement, chaque connaissance s'appuyant sur des éléments situés au niveau immédiatement inférieur dans la hiérarchie

(Mendelsohn, 1988). Une première conséquence de cette hiérarchie est que la notion de transfert n'a pas le même sens pour chacun des niveaux d'organisation considérés:
1. Au niveau le plus élémentaire, on trouve l'ensemble des compétences qui permettent à l'enfant de maîtriser le codage des informations (les systèmes de représentation et de traitement). Ces codes sont spécifiques aux différents domaines du traitement de l'information et ont toujours un caractère conventionnel: le langage et sa syntaxe, les opérateurs du système numérique, de la logique des classes et des relations et des opérations spatiales. On ne peut pas parler à ce niveau de transfert d'apprentissage d'une tâche à l'autre, sauf à considérer que ces compétences sont utilisables dans des contextes extrêmement variés.
2. A partir de tels instruments de codage - que l'on peut considérer comme d'authentiques outils cognitifs au sens de Derry (1990) - le sujet peut construire des réseaux associatifs d'informations (les schémas) qui de proche en proche deviennent ce que nous appelons des connaissances. Celles-ci sont en grande partie liées à notre expérience propre et ne sont transposables d'un domaine à l'autre que par analogie entre deux situations (l'analogie pouvant être considérée comme la mise en place d'une nouvelle association entre deux domaines jusqu'alors indépendants).
3. A un niveau supérieur, on trouve les stratégies qui sont des connaissances stockées sous forme de règles d'action. Leur rôle est d'apporter une meilleure gestion des ressources cognitives en diminuant considérablement les recherches par essais et erreurs. Elles peuvent s'appliquer à toutes les situations qui possèdent les mêmes caractéristiques du point de vue du fonctionnement du sujet.

Il faut donc bien se garder, en la matière, d'adopter la position simplificatrice qui consiste à considérer le transfert d'apprentissage comme un processus unitaire. Toute connaissance doit être replacée dans son contexte d'acquisition et analysée en référence à son rôle dans l'économie de l'organisation cognitive. Ainsi on pourrait penser que les connaissances situées en haut de la hiérarchie - du fait qu'elles sont indépendantes du domaine d'application - sont plus facilement transposables. C'est probablement la raison pour laquelle les psychologues se sont intéressés à elles en premier et qu'il est relativement facile de les simuler sur ordinateur (dans les systèmes experts, par exemple). Cependant, elles ne permettent pas de faire l'économie de connaissances plus élémentaires comme les schémas (connaissances organisées en réseaux associatifs thématiques). Celles-ci sont propres à l'histoire de chaque sujet et par voie de conséquence ont un degré de "portabilité" limité aux analogies possibles entre les situations rencontrées. Ces connaissances sont celles qui posent le plus de problèmes aux chercheurs en intelligence artificielle, car il est en effet très difficile de formaliser l'histoire personnelle de nos acquisitions (les recherches sur les réseaux connexionnistes sont une tentative dans cette direction). C'est pourtant cette histoire qui explique en grande partie notre capacité à évoquer une solution quand nous sommes confrontés à un problème.

Brown (1989) fait remarquer, au sujet de ce problème du tranfert, que nous sommes tous prisonniers d'une intuition probablement erronée. En effet, plusieurs travaux expérimentaux, dont celui de di Sessa (1983), prouvent que les connaissances abstraites et générales sont, non seulement plus difficiles à acquérir, mais encore difficilement transférables. Il devient par contre de plus en plus clair que les connaissances spécifiques à un domaine sont non seulement plus facilement acquises, mais qu'elles possèdent un certain degré de transférabilité grâce à la compatibilité entre règles et aux possibilités offertes par le raisonnement par analogie (Kotowsky, Hayes et Simon, 1985; Lave, 1988).

Une des conséquences de cette analyse nous amène à considérer que le transfert de connaissance est plus un mécanisme "spontané", lié à la manière dont les connaissances sont organisées les unes par rapport aux autres, qu'un processus sous contrôle explicite de l'apprenant. Il nous semble donc important qu'un ILE favorise au maximum l'articulation entre les niveaux de connaissance par une mise en relation (a) d'une variété d'activités sur ce domaine, (b) d'une variété de représentations de ce même domaine et (c) de l'association entre un domaine particulier et les autres domaines de connaissances.

Apprentissage et hiérarchie des buts.

Un enseignement est efficace si et seulement si le but poursuivi par l'enseignant est à la fois explicite pour l'élève et à sa portée. Il existe malheureusement trop souvent un décalage entre les objectifs fixés par l'enseignant (concevoir, par exemple un programme modulaire) et ceux que se donne implicitement l'apprenant (par exemple, faire que le programme tourne). Lehrer et al. (1988) proposent ainsi un modèle de l'apprentissage de la programmation dans lequel les buts sont organisés suivant un modèle hiérarchique. Pour vérifier qu'un but est atteint, le sujet procède par appariement entre une norme et ce qu'il perçoit. Si la distance entre ces deux entités est trop importante, deux possibilités s'offrent à lui: soit il réorganise les buts de niveau inférieur pour qu'ils s'adaptent à sa performance; soit il s'appuie sur une norme plus élevée dans la hiérarchie pour justifier la distorsion observée. Ces auteurs procèdent, suivant ce schéma, à une analyse fine et très intéressante des différents objectifs possibles pour la mise au point de programmes LOGO:
- localisation de la séquence où se trouve l'erreur,
- analyse des différences entre résultats observés et résultats attendus,
- correction de l'erreur,
- contrôle des effets de la mise au point sur le programme,
- constat que les erreurs sont des opportunités pour apprendre.

Sans entrer plus avant dans les détails de cette recherche, je voudrais retenir de cette idée d'analyse hiérarchique des buts quelques considérations utiles pour la mise au point d'un ILE et plus généralement pour l'enseignement d'une expertise:
1. Le "tuteur" doit être un modérateur de la nouveauté. C'est lui qui décide du bien fondé d'attirer l'attention de l'élève sur une norme qu'il n'a pas encore prise en compte, au moment où cette dernière est accessible.
2. Il est évident que les normes du sommet de la hiérarchie mettent plus longtemps à se mettre en place que celles de la base. Elles sont par contre beaucoup plus stables et résitent mieux au changements (les idées fortes se développent lentement).
3. Le point sur lequel porte l'attention de l'élève détermine la norme qui sert de référence à un moment particulier de l'instruction. Le contrôle est alors exercé depuis ce niveau sur les normes inférieures.
4. La construction de buts élevés dans la hiérarchie des normes ainsi définies favorise la mise en relation de procédures entre différents domaines d'activités. Elle représente pour ces auteurs la possibilité d'établir des ponts qui seront autant d'opportunités pour réaliser des transferts de connaissances.

Le problème du modèle de l'élève.

Ce thème a été un élément majeur du débat dans la communauté des "tuteurs intelligents" pendant toute la décennie passée. Quelle représentation le système doit-il avoir du fonctionnement de l'élève? Jusqu'à quel point doit-il pouvoir rendre compte des actions que l'élève réalise? A quelles décisions cette représentation doit-elle être utile? C'est finalement Self (1988) qui a en quelque sorte mis un terme à ce débat en proposant d'abandonner l'idée de réaliser un modèle parfait de l'élève. Son argumentation porte essentiellement sur deux points:

1. Il est illusoire de penser atteindre dans ce domaine un résultat satisfaisant étant donné nos connaissances actuelles sur les processus cognitifs du sujet apprenant.
2. N'est-il pas préférable de donner au sujet le maximum de moyens pour exprimer lui-même ce qu'il est en train de faire par une interface "intelligente"? On fait ainsi d'une pierre deux coups: d'une part on obtient des renseignements précieux sur la stratégie utilisée par l'élève (sans être obligé de faire des paris théoriques risqués); d'autre part on contribue à aider le sujet à "réfléchir" ses propres actions, ce qui procure un avantage indéniable du point de vue des acquisitions.

J'ajouterai à cette argumentation une remarque d'ordre plus général. L'objectif premier d'un environnement d'apprentissage n'est pas de "jouer" à l'enseignant mais de proposer un ensemble d'outils et de procédures qui favorisent la résolution d'un problème et donc par la même l'acquisition de schémas. Il n'est donc pas nécessaire d'avoir - pour accomplir cette mission - une vision parfaite et détaillée de ce que fait l'élève sur le système; il est par contre important de lui donner des outils puissants et à sa mesure. Je rejoins sur ce point la position de Derry (1990) selon qui les environnements d'apprentissage doivent être conçus comme des "outils cognitifs" qui expédient des parties de "calculs" ne pouvant être pris en charge par l'apprenant seul. Le système joue ainsi le rôle de "levier" pour alléger la charge mentale induite par la somme des traitements à effectuer lors d'une résolution de problème. La mise en place de ces outils cognitifs passe par une "intériorisation" d'outils réels dont l'efficacité est jugée suffisamment bonne par l'élève. Un exemple spectaculaire de ce type d'intériorisation nous est proposé par Hatano and Osawa (1983) qui ont étudié les calculateurs prodiges japonais. Ces champions de boulier sont capables de réaliser mentalement des opérations numériques complexes par un codage spatial des nombres isomorphe à celui des abbaques qu'ils utilisent en compétition.
Nous pensons que ce changement de paradigme permet de contourner la difficulté de construire un modèle de l'élève fiable, tout en prenant en compte les acquis les plus significatifs des études sur le fonctionnement cognitif.

La meilleure façon d'apprendre c'est encore d'enseigner: Révision d'un vieux principe pédagogique.

Cette affirmation, que tout enseignant a pu aisément constater par lui-même, est un des plus anciens principes pédagogiques connus. Une révision de ce principe nous est proposée par quelques uns des projets du Epistemology & Learning Group du Media Lab (Harel and Papert, 1990; Jackson, 1990). Ces auteurs ont donné à ce principe une nouvelle jeunesse et un habillage plus moderne en utilisant la terminologie "Instructional Software Design for Learning". La "mécanique" de ce principe est fort bien analysée par ces auteurs: la construction de didacticiels demande au concepteur de définir une hiérarchie de buts explicites (voir plus haut, Lehrer et al. 1988), de représenter les connaissances à enseigner, de planifier les leçons, d'évaluer les résultats. L'ensemble de ces tâches peuvent être résumées par la seule idée de "réflexion" qui est, comme nous l'avons vu, le moteur de tout apprentissage par enseignement. Le problème non résolu qui se pose à propos de la mise en oeuvre de ce principe est celui de l'emboîtement des multiples compétences nécessaires à cette activité de conception de logiciels d'enseignement: connaissances de l'ordinateur, connaissances en programmation, en communication (interface), et connaissances sur le contenu à enseigner. Harel and Papert (1990) résolvent le problème de manière élégante par un second principe: The "Integrated Learning Principle" qui s'énonce ainsi "learning more can be easier than learning less". Je ne suis pas loin de penser qu'ils ont partiellement raison mais (a) cela reste à démontrer et (b) ce n'est sûrement applicable que lorsque certaines conditions bien spécifiques sont remplies (en particulier si le domaine se prête à une simplification pédagogique pertinente). Cette position révèle en tous cas un problème important pour la conception des ILE, celui de la complexité apparente et des buts assignés au système.

Implémentation d'un environnement d'apprentissage: Memolab

Memolab: un système hybride.

Ces questions nous ont permis de définir l'architecture d'un système fondée sur l'intégration de plusieurs composantes. Les principales caractéristiques de ce système sont des tentatives pour opérationnaliser les questions de recherche que nous venons d'exposer. Memolab est un système prototype qui permettra l'apprentissage de la méthode expérimentale en Sciences humaines (figure 2). Le contenu particulier sur lequel les sujets peuvent expérimenter porte sur l'étude des paradigmes de la mémoire humaine. C'est un système hybride, il est composé des éléments suivants:

< insérer ici la figure 2>

1. Un langage de commandes permettant de construire des objets de complexité croissante sur un domaine bien délimité de connaissances. Ces commandes sont organisées suivant une hiérarchie qui respecte un ordre "psycho-logique" de construction des acquisitions de ce domaine (Beauvois, Roulin et Tiberghien, 1990). Le micro-monde simulé dans le système permettra de réaliser des expériences sur la mémoire en combinant différentes sources de variations sur le traitement expérimental, les sujets, et les paramètres de la réponse.
2. Une composante de guidage sous la forme d'agents spécialisés pour assurer le contrôle de ce qui n'est pas encore accessible à la maîtrise de l'apprenant. Plusieurs agents sont prévus pour Mémolab - ils feront l'objet d'une implémentation progressive - chacun prenant à sa charge une des composantes de l'activité.
3. Un ensemble de problèmes types à résoudre, comme dans les didacticiels classiques, pour assurer la formation de "schémas" (chaîne associative de concepts et de procédures). Ces schémas permettront au sujet d'agir suivant des plans réputés efficaces lorsque certains paramètres déclencheurs seront présents dans un problème donné. Il s'agira plus précisément de traiter le problème de la décision statistique dans la méthode expérimentale: le concept de variance (intra et intergroupe); le concept d'hypothèse, ainsi que les choix d'outils statistiques associés aux caractéristiques des variables dépendantes.
4. Un ensemble de logiciels d'aide au travail: hypertexte sur le domaine de connaissances (theory of human memory), "notebook" pour décharger la mémoire de travail, calculatrices spécialisées, éditeur de schémas (paradigm map) etc.

Memolab: un système intelligent.

Un ILE doit favoriser, de toutes les manières possibles, une représentation fidèle et "intelligente" (au troisième sens du terme) des actions que le sujet réalise sur l'environnement de travail. Le système doit donc explorer toutes les possibilités de "réfléchir" les activités du sujet. Nous entendons par là toute forme de représentation graphique ou symbolique qui permettent à l'apprenant de manipuler l'interprétation que le tuteur se fait des connaissances et des métaconnaissances de l'élève. Cette possibilité de "réfléchir" les actions du sujet sur le système offre un moyen de transformer une expérience en un objet manipulable et du même coup répond à une des objections majeures formulées à l'encontre des micromondes: la difficulté de transformer une expérience en un apprentissage transférable. Comment avons-nous choisi de rendre cette idée opérationnelle ? Trois pistes sont actuellement à l'étude: représentation des macro-actions, selection de schémas, et contrôle de la hiérarchie des structures de but.

1. Le sujet doit lui même décrire, par l'intermédiaire d'un langage de commandes, les opérations successives qu'il effectue dans la construction de son plan d'expérience. La représentation élaborée par le système à partir de cette description fournie par l'élève sera "interprétative", c'est à dire qu'elle ne se contentera pas de proposer un simple historique séquentiel des commandes utilisées par le sujet mais elle ressemblera plutôt à un organigramme des séquences thématiques d'actions. Les connaissances fines et détaillées des stratégies possibles et des erreurs types seront ici capitales pour réaliser cette 'interprétation' - comme dans l'exemple proposé par Bastien (1989).

2. Un autre ensemble d'aide concerne les relations entre le concept de "réflexion" et celui de transfert (c'est à dire la possibilité d'utiliser pour un nouveau contexte des connaissances acquises dans un autre domaine). L'idée consiste à assister l'élève dans la mise au point de schémas "paradigmatiques" à partir des invariants de structure qu'il est amené à manipuler dans la construction d'un plan d'expérience (propriétés de la variable dépendante (Vd) et des sources de variations (Sv)). Il aura ainsi à sa disposition une sorte de "banque de données" personnelles de schémas qui seront composables à volonté pour résoudre de nouveaux problèmes.

3. Enfin, il est prévu de mettre au point un mécanisme de "réunitarisation" analogue à celui proposé par Case (1985) dans son modèle développemental et que nous avons mentionné à propos du projet "Schèmes Informatiques Programmables" (Mendelsohn, 1986). Cette procédure, baptisée "language shift" se met en place en deux étapes: (a) quand l'apprenant interagit avec l'environnement, il utilise un "langage de commande" (MAKE-TASK, RECORD-EVENT) qui lui permet de construire des expériences virtuelles; en même temps, il peut observer ce qu'il a produit comme effet à travers un langage de description (Vd, Sv, Plan apparié, paradigme); (b) un "language shift" se produit lorsque le sujet peut utiliser le langage de description fourni par le système comme un authentique langage de commande (analogie avec le mécanisme de l'abstraction réfléchissante). Ce second stade correspond à la maîtrise d'une nouvelle conceptualisation du domaine enseigné (pour plus de détails sur Memolab voir Mendelsohn, à paraître).

En conclusion, nous soulignerons que l'approche que nous préconisons permet deux stratégies de recherche complémentaires: d'une part nous enrichissons les recherches sur l'enseignement assisté par ordinateur d'une méthodologie de développement et d'une assise théorique qui lui faisaient souvent défaut, d'autre part, nous proposons à la psychologie cognitive développementale un laboratoire pour éprouver ses modèles en testant leur faisabilité. Autre avantage de cette approche, un environnement intelligent d'apprentissage comme MEMOLAB prend en compte la richesse et la complexité des processus cognitifs impliqués dans l'enseignement d'une discipline. Par exemple, l'importance des mécanismes de découverte, le rôle de l'entraînement, le raisonnement par analogie et les effets du guidage de l'action sont intégrés à un environnement comme MEMOLAB, comme ils le sont dans le monde réel. Trop longtemps, la recherche sur en EAO a isolé des processus d'acquisition dont la fonction complémentaire était sous-estimée. Cette ultime remarque nous amène à constater qu'une des définitions les plus classiques de l'intelligence s'appliquerait alors pleinement à notre environnement: celle qui met en avant la richesse des capacités d'adaptation d'un système à son environnement.

Références

Anderson, J. R. (1982). *The Architecture of Cognition.* Cambridge, MA: Harvard University Press.

Anderson, J. R. & Reiser, B. J. (1985). The LISP tutor. *Byte, 10,* (4), 159-175.

Bastien, C. (1989). Plaidoyer pour un apprentissage différencié. *Psychologie Française*, *34*, 4, 271-276.

Beauvois, J. L., Roulin, J. L. & Tiberghien, G. (1990). *Manuel d'études pratiques de Psychologie.* Paris: PUF.

Brown, J.S. (1989). Toward a new Epistemology for Learning. In C. Frasson & J. Gauthier (Eds). *Intelligent tutoring systems at the Crossroad of AI and Education.* Norwood, NJ: Ablax.

Bruner, J. S. (1964). The course of cognitive growth. *American Psychologist,* 19, 1-15.

Burton, R. R. and Brown, J. S. (1979). An investigation of computer coaching for informal learning activities. *International Journal of Man-Machine Studies*, *11*, 5-24.

Case, R. (1985). *Intellectual Development: Birth to Adulthood.* NY: Academic Press.

Derry, S.J. (1990). Flexible Cognitive Tools for Problem Solving Instruction. Paper presented at *AERA Symposium, Computers as Cognitive Tools.* Boston MA.

Dillenbourg, P., Hilario, M., Mendelsohn, P. & Schneider, D. (1990). The Memolab project. TECFA document 90-4. Université de Genève.

Dillenbourg, P., Hilario, M., Mendelsohn, P. & Schneider, D. (1990). The Geneva Manifesto of Learning Environments. TECFA document 90-2. Université de Genève.

diSessa, A. (1983). Phenomenology and the evolution of intuition. In G. Forman & P. Pufall (Eds). *Mental models.* Hillsdale, NJ: Erlbaum.

Feurzeig, W. (1986). Algebra slaves and agents in a LOGO-based mathematics curriculum. *Instructional Science,* 14, 229-254.

Frederiksen, J. R. & White, B. Y. (1988). Intelligent learning systems for science education. Proceedinds of *ITS-88*, Montreal, 250-257.

Gagné, R. M. (1987). Instructional Technology: Foundations. Hillsdale: Lawrence Erlbaum.

Hatano, G. & Osawa, K. (1983). Digit Memory of grand Expert in Abacus-derived mental calculation. *Cognition,* 15, 95-110.

Harel, I. & Papert, S. (1990). Software Design as a Learning Environment. In I. Harel (Ed.), *Constructionist Learning.* Boston: MIT Media Laboratory.

Jackson, I., (1990). Children's Software Design as Context for Studying Collaboration. In I. Harel (Ed.), *Constructionist Learning.* Boston: MIT Media Laboratory.

Kotowsky, K., Hayes, J.R. & Simon, H.A. (1985). Why are some problems hard? Evidence from Tower of Hanoi. *Cognitive Psychology,* 17, 248-294.

Kurland, D. M. & Pea, R. D. (1984). On the cognitive effects of learning computer programming. *New Ideas in Psychology,* 2 (2), 137-168.

Lave, J. (1988). *Cognition in Practice.* Boston, MA: MIT Press.

Lehrer, R., Guckenberg, T. & Sancilio, L., (1988). Influences of LOGO on children's Intellectual Development. In R.E. Mayer (Ed.): *Teaching and Learning Computer Programming, Multiple Research Perspectives.* New York: Academic Press.

Littlefield, J., Delclos, V. R., Lever, S., Clayton, K., Brandsford, J. & Franks, J. (1989). Method of teaching, transfer of general skills, and attitudes toward school and computers. In R.E. Mayer (Ed): *Teaching and learning computer programming, multiple research perspectives.* New York: Academic Press.

Mendelsohn, P. (1986). Computer programming, knowledge organization and cognitive development. Proceedings of the *Second European Conference on Developmental Psychology.* Rome.

Mendelsohn, P. (1988). Psychologie cognitive et processus d'acquisition des connaissances. *European Journal of Psychology of Education,* Numéro spécial Hors série, 22-28.

Mendelsohn, P. (à paraître). Les modèles cognitifs dans les environnements d'apprentissage: implémentation ou contemplation? Actes des 13èmes journées francophones sur l'informatique, Genève, 9-10 janvier 1991.

Papert, S. (1980). *Mindstorm: Children, Computers and Powerful Ideas.* New York: Basic Books.

Richard, J. F. (1990). *Les activités mentales: comprendre, raisonner, trouver des solutions.* Paris: Armand Colin.

Self, J. (1988). Bypassing the intractable problem of student modelling. Proceedings of the *ITS-88 conference*. Montreal, 18-24.

Solomon, C. (1986). *Computer Environments for Children: A reflection on theories of Learning and Education.* Boston, MA: MIT Press.

Vygotsky, L. S. (1978). Mind in Society. in M. Cole, V. John-Steiner and E. Souberman (Eds), *The Development of Higher Psychological Processes.* Cambridge: Harvard University Press.

Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems. Computational and Cognitive Approaches to the Communication of Knowledge.* Los Altos: Morgan Kaufman.

Résumé

Le concept d'environnement intelligent d'apprentissage permet de dépasser l'opposition simpliste et manichéenne entre les défenseurs de l'apprentissage par induction (résultant des seules activités exploratoires du sujet) et les partisans des tutoriels inspirés (de près ou de loin)

par l'enseignement programmé (caricature de l'apprentissage par enseignement). Ces systèmes doivent être capables de favoriser l'acquisition de concepts et de procédures associés à un domaine de connaissance. Dans cette perspective, les environnements d'apprentissage permettent au chercheur de disposer d'un véritable laboratoire pour étudier les transformations que l'on pourrait "artificiellement" apporter à la pratique pédagogique. Il est possible d'imaginer, par exemple, de donner à l'enseignant et à l'apprenant, la possibilité de contrôler des paramètres non triviaux de l'apprentissage: charge cognitive en mémoire de travail, représentation structurée de l'histoire d'une session, aide à la planification...
La première partie de cette conférence amènera à préciser quelques points de vocabulaire à partir d'une représentation schématique des concepts les plus significatifs du domaine. Dans une seconde partie, seront abordées un certain nombre de questions de recherches parmi celles qui sont discutées actuellement par la communauté scientifique. Cet exposé se termine par une brève présentation d'un des projets de recherche de TECFA dont la responsabilité scientifique est partagée avec P. Dillenbourg et D. Schneider: Memolab.

Communication présentée au 11 ème cours avancé de la Fondation ARCHIVES JEAN PIAGET, Genève, 10-14 septembre 1990. A paraître dans les actes du colloque.