
De la généralisabilité d'un environnement d'apprentissage.

Pierre Dillenbourg

Melanie Hilario

Patrick Mendelsohn

Daniel Schneider

Boris Borcic

TECFA, Faculté de Psychologie et des Sciences de l'Education

Université de Genève

9, Route de Drize, 1227 Carouge, Suisse.

RESUME : Cette contribution présente ETOILE, un outil logiciel pour la création d'environnements interactifs d'apprentissage. Notre contribution ne se centre pas sur l'outil en lui-même mais sur le processus de généralisation de connaissances pédagogiques. ETOILE a été développé par abstraction d'un environnement d'apprentissage particulier, MEMOLAB. Notre analyse comporte une double comparaison, d'une part, entre des prototypes successifs de ETOILE, à des degrés d'abstraction croissants, et d'autre part, entre des styles d'enseignement variés, mais exprimés dans un langage commun. Nous avons observé une évolution de la conception de l'espace didactique: d'une approche combinatoire, basée sur un seul jeu de règles dont le comportement varie selon un ensemble de paramètres, nous sommes passés à une approche énumérative, où chaque style d'enseignement correspond à un jeu de règles différent. Pour définir un style d'enseignement indépendamment d'un contenu, les règles pédagogiques ne portent pas directement sur le contenu mais supervisent les interactions "adidactiques" entre un expert du domaine et l'apprenant. Ces interactions ont interceptées par les tuteurs et prennent alors un sens didactique.

1. Questions de recherche

Le projet dont nous rapportons ici les développements les plus récents avait pour objectif de répondre à deux questions, présentées ci-dessous.

Comment construire des environnements d'apprentissage à initiative mixte, intégrant les approches tutorielle et micro-monde ? Le système associé à cette première question est un environnement d'apprentissage, dénommé MEMOLAB. Ce système a pour objectif l'acquisition des compétences de base en psychologie expérimentale, à travers la création et la simulation d'expériences sur la mémoire. Il

apporte deux réponses, deux façons de fournir à l'apprenant l'aide minimale pour transformer ses activités en apprentissage. Le premier support est le tutorat: nous avons créé des **agents** (tuteurs) susceptibles de fournir une aide directe à l'élève (explications, guidage, feed-back,...). Le second support est indirect. Il consiste à **structurer l'environnement** d'apprentissage en une séquence de micro-mondes. La structure de l'environnement traduit en termes informatiques [Dillenbourg, 1992; Dillenbourg et Mendelsohn, à paraître] la théorie du développement proposée par Case [1985].

Comment généraliser les deux réponses fournies à la première question? La seconde réponse de la première question, la structuration de l'environnement, est elle-même définie en termes généraux. Nos efforts se sont donc portés sur la généralisation de la première réponse: les agents pédagogiques. Cet article rapporte nos tentatives de définir une expertise pédagogique indépendamment du contenu d'enseignement. Ces efforts ont abouti à la création d'une boîte à outils pour la conception d'environnements tels que MEMOLAB. Cet outil s'appelle ETOILE ('experimental toolbox for intelligent learning environments')[Dillenbourg et al, 1992].

2. La méthode

Si on pose la question "qu'il y a-t-il de commun à tout logiciel pédagogique?", les réponses que l'on obtient décrivent le plus souvent des fonctions d'interaction entre l'utilisateur et le didacticiel (présentation, analyse de réponse, type de branchement). Ces fonctions sont pédagogiquement neutres, elles n'encodent pas de choix pédagogiques. En fait, cette question est trop générale. En outre, elle est posée trop tôt: elle correspondrait à la démarche d'un psychologue du développement qui se contenterait d'observer des adultes. Notre objet d'étude est la genèse d'une architecture pédagogique indépendante du domaine enseigné. Nous avons observé le processus d'abstraction des stratégies d'enseignement et de leur représentation informatique. Pour cela, nous sommes partis d'un système particulier, dénommé MEMOLAB, et nous en avons progressivement abstrait les composantes généralisables. Ce processus d'abstraction n'a évidemment pas été linéaire; il a nécessité plusieurs restructurations majeures du système.

Cette première approche repose sur la question classique 'comment appliquer une stratégie pédagogique à différents contenus?'. Il existe cependant une autre façon de détacher une méthode de son contenu: il s'agit de définir **plusieurs stratégies didactiques applicables à un même contenu**. L'avantage de cette seconde approche est de forcer le concepteur à définir un langage de haut niveau pour la représentation des stratégies didactiques. Vu le niveau de généralité visé, nous parlerons de styles d'enseignement (enseignement étant pris au sens large), plutôt que de stratégie didactique. Le travail dont nous rapportons les résultats combine donc ces deux approches, allant du particulier au général, et visant à la multiplicité des styles d'enseignement.

3. Evolution de la conception de l'espace didactique

Si on observe l'évolution de MEMOLAB, on s'aperçoit que cette évolution repose sur un changement dans la conception de l'espace didactique. Par espace didactique, nous entendons l'ensemble des activités d'enseignement ou d'apprentissage initiées par un agent pédagogique, toute décision et intervention toute prise par celui-ci. Ces deux conceptions, combinatoire et énumérative, sont décrites ci-dessous. Dans les deux cas, les connaissances pédagogiques du ou des tuteurs sont représentées par un ensemble de règles de production.

3.1. L'approche combinatoire

Dans la première version de MEMOLAB, le style d'enseignement était décrit par dix **paramètres** tels que le degré de contrôle de l'élève, le type de stratégie d'apprentissage (inductive, déductive ou analogique), le degré d'inertie du tuteur (s'il s'adapte rapidement aux comportements de l'apprenant ou préfère conserver une certaine stabilité dans son comportement), etc. L'ensemble des combinaisons de toutes les valeurs possibles de tous ces paramètres était supposé couvrir l'espace didactique. Cette supposition fut mise à l'épreuve lors de l'implémentation du tuteur. ETOILE comportait un seul tuteur, doté d'une base de règles. Les conditions de ces règles faisaient référence aux paramètres cités. La valeur de chaque paramètre était déterminée par un autre agent, nommé le coach.

La mise au point des règles du tuteur nous a progressivement contraint à définir des règles très nombreuses et complexes (plus de dix prémisses). Leur nombre et complexité était lié à la nécessité de considérer, de façon presque individuelle, les différentes combinaisons de valeurs de paramètres afin d'éviter les combinaisons qui n'avaient pas ou peu de sens. Par exemple, on associe rarement une approche inductive avec un faible degré d'initiative pour l'élève. Une des principales conséquences de cette démarche était qu'il nous était très difficile de pouvoir abstraire des règles d'une telle complexité. Il nous est apparu que cet espace n'était pas continu, qu'il comportait des trous, des combinaisons de valeurs de paramètres peu intéressantes. Ces 'trous' étant nombreux, il devenait plus fastidieux de définir l'espace et tous ses 'trous' que de définir directement les zones habitables.

3.2. L'approche énumérative

L'approche énumérative définit un certain nombre d'îlots au sein de l'espace didactique. Chaque îlot ou style d'enseignement correspond, par rapport à l'approche combinatoire, à un ensemble de valeurs de paramètres qui jouit d'une forte consistance interne. Chaque style d'enseignement est implémenté comme un jeu de règles, hermétiquement séparé des autres jeux. A l'intérieur de chaque jeu, certains paramètres sont implicitement fixés. Les règles ne doivent donc traiter que les

paramètres qui peuvent varier, ce qui diminue le nombre de combinaisons à prendre en considération, et simplifie dès lors l'ensemble des règles.

Nous avons défini cinq styles, chaque style étant attribué à un tuteur. Les cinq tuteurs sont respectivement dénommés Skinner, Bloom, Vygotsky, Piaget et Papert. Chaque style pédagogique s'inspire de ces différentes approches théoriques, mais n'en emprunte le nom qu'à titre honorifique. Pour décrire ces styles didactiques, il convient d'expliquer comment nous représentons un style d'enseignement indépendamment des contenus de l'enseignement.

4. Comment se référer au domaine dans des règles didactiques?

Pour exprimer des styles d'enseignement indépendamment du contenu de l'enseignement, la solution la plus fréquente consiste à remplacer les éléments de contenu par la catégorie (variable) à laquelle appartiennent ces éléments au sein d'une classification quelconque des unités de connaissance. Ceux-ci sont décrits en termes de chapitres, unités, chunks, ou objets. La difficulté réside à mettre au point une terminologie qui puisse convenir à des disciplines dont le contenu est parfois structuré de façon très diverse. Notre langage se limite à deux concepts pédagogiques, le concept d'objectif et de problème (voir section 5). En ce qui concerne le contenu même, nous avons opté pour une approche différente du processus d'abstraction. Nos règles pédagogiques ne manipulent pas directement le contenu mais déterminent le mode d'interaction entre l'apprenant et un expert du domaine (une base de règles). Comme ce fut souvent le cas en EIAO, les connaissances sont distribuées entre plusieurs agents: un/des tuteur(s) et un/des expert(s). L'expert résout des problèmes en interaction étroite avec l'apprenant. Le tuteur supervise ces interactions afin d'intervenir lorsqu'il considère que c'est nécessaire.

4.1 Interactions élève-expert

Dans ETOILE, l'essentiel des activités d'apprentissage réside dans la résolution de problèmes par l'élève, en interaction avec un expert. Le terme 'interaction' couvre un continuum de modes de participation allant de la démonstration (l'expert fait tout, l'élève observe) jusqu'à l'exercice (l'élève fait tout, l'expert le suit) en passant par des formes de collaboration plus équilibrées. Le choix d'un mode d'interaction est déterminé par le tuteur en fonction de son style pédagogique, de la phase d'apprentissage et du comportement de l'élève.

La principale caractéristique de notre approche réside dans le fait que la collaboration entre l'élève et l'expert concerne la manipulation de l'interface, plutôt que des connaissances symboliques internes. Ce choix résulte de nos travaux antérieurs en matière de collaboration entre l'élève et la machine [Dillenbourg et Self, 1992] et par les conceptions récentes en matière de cognition distribuée [Resnick, 1991] et de collecticiels (Shrage, 1990). Imaginons deux bases de règles distinctes qui partagent une même et unique base de faits. Tout fait produit par une

règle est ajouté à la base de faits et devient susceptible d'instancier les conditions d'une règle de chacune des deux bases. Le processus de collaboration possède ainsi un haut niveau de granularité et d'opportunisme. Si l'on transpose ceci à une collaboration homme-machine, comment s'assurer que l'élève et l'expert partagent la même base de faits, c'est-à-dire la même représentation du problème? Notre solution est la suivante (voir figure 1): les règles de l'expert reposent sur la représentation externe du problème, telle qu'elle apparaît à l'écran aux yeux de l'apprenant, plutôt que sur une base de faits interne. Si une règle de l'expert transforme la représentation du problème, cette transformation est également perceptible pour l'élève. Réciproquement, le raisonnement de l'élève le conduira à effectuer des modifications du problème, lesquelles seront automatiquement prises en compte lorsque l'expert tentera d'activer de nouvelles règles.

Cette forme de collaboration nous permet une méthode de diagnostic très simple, proche du 'model tracing' d'Anderson [Anderson; Farrell and Sauer, 1984]. Cette méthode consiste à détecter la règle de l'expert qui aurait conduit à la même transformation de l'interface que celle réalisée par l'apprenant. En réalité, l'accent est davantage placé sur les mécanismes d'appropriation que de diagnostic. C'est Newman [1989] qui suggère que la collaboration peut remplacer le diagnostic. Il s'inspire des modèles socio-culturels de l'apprentissage selon lesquels l'apprenti intériorise le cadre conceptuel de l'expert en observant comment celui-ci associe la contribution de l'élève à sa propre stratégie de résolution de problème. Cet élève observe la stratégie de l'expert, non de façon abstraite; il découvre cette stratégie instanciée et mise en oeuvre dans un contexte particulier.

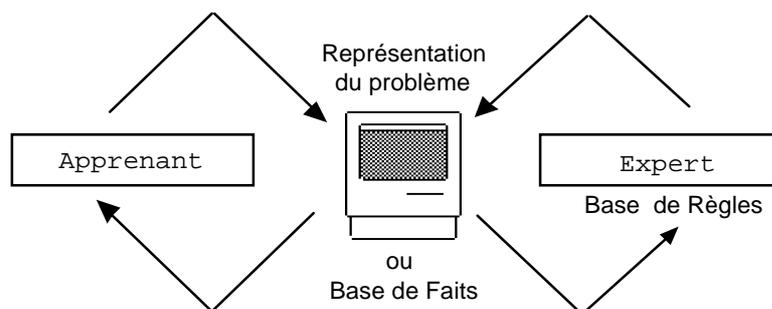


Figure 1: Résolution conjointe de problèmes par l'expert et l'apprenant

4.2 Les tuteurs supervisent les interactions expert-apprenant.

Les interactions entre l'expert et l'apprenant sont "adidactiques". L'expert informatisé a pour seul but de résoudre des problèmes conjointement avec son partenaire humain. L'expert est égocentrique. S'il pose des questions ou corrige certaines erreurs, c'est uniquement parce qu'il cherche à réaliser la tâche qui lui est soumise. Les interactions "adidactiques" entre l'expert et l'apprenant sont

interceptées par le tuteur qui leur attribue un sens en fonction de ses intentions didactiques. Nous illustrons ceci par deux exemples.

Si l'élève réalise une opération qui ne correspond à aucune règle de l'expert, ce dernier va se plaindre. Etant égocentrique, l'expert demandera à l'élève s'il est d'accord que sa dernière action soit 'défaite', c'est-à-dire que l'environnement soit restauré dans son état préalable. L'apprenant peut refuser. Le tuteur observe ces interactions sans réagir. Néanmoins, si l'apprenant manifeste plusieurs refus au cours d'un même problème, le tuteur interviendra pour le contraindre à écouter l'expert. Le nombre de refus de l'élève précédant l'intervention autoritaire du tuteur est une des caractéristiques du style d'enseignement de ce tuteur. Par exemple, le tuteur 'Bloom' interviendra plus tôt que le tuteur 'Piaget'.

D'autre part, l'expert dispose de 'règles de réparation'. Elles permettent à l'expert de corriger un état intermédiaire du problème dans le strict objectif d'atteindre son but. Ces règles correspondent en quelque sorte aux 'malrules' utilisées dans les méthodes classiques de diagnostic cognitif. Elles sont néanmoins égocentriques et "adidactiques", comme les experts. Le format d'une 'malrule' est 'si tel état.. faites quelque chose de faux', alors qu'une règle de réparation est du type 'si tel état incorrect... corrigez le comme ceci'. Lorsque l'expert applique une règle de réparation, il le signale à l'apprenant. Lorsque le tuteur intercepte ce type de message, il peut par exemple ouvrir un hypertexte afin de présenter des explications complémentaires. (Lorsque l'auteur définit une règle de réparation, il lui associe un noeud de l'hypertexte; le tuteur gère ensuite ce lien). La décision d'un tuteur en présence d'un cas de réparation n'est pas systématiquement d'ouvrir l'hypertexte. Cette décision dépend de son style d'enseignement: le tuteur 'Papert' par exemple ne fournira pas ce type d'aide.

5. Implémentation d'ETOILE

La création d'un environnement d'apprentissage avec ETOILE ne consiste pas à définir un scénario plus ou moins ouvert comme dans la plupart des systèmes-auteur. ETOILE fournit à l'auteur des agents de base, les tuteurs et le coach, entre lesquels nous avons distribué les responsabilités pédagogiques. Le coach est hiérarchiquement supérieur aux tuteurs et gère le curriculum: il sélectionne les buts, décide des changements d'interface, sélectionne l'expert si plusieurs experts sont disponibles pour l'objectif courant et enfin, et surtout, sélectionne le tuteur. Le tuteur gère l'interaction locale avec l'apprenant, il détermine les activités d'apprentissage, sélectionne le problème à résoudre et le mode de collaboration entre l'expert et l'apprenant.

La tâche de l'auteur est de compléter cette micro-société pédagogique en créant d'autres agents (les experts) ainsi que les interfaces (le micro-monde). ETOILE spécifie les contraintes concernant la définition des experts, afin qu'ils interagissent avec l'élève et avec les tuteurs selon les modalités propres à ETOILE. L'auteur doit en outre définir les problèmes que l'apprenant devra résoudre et l'interface

disponible pour cela. Enfin, l'auteur pourra construire un ou plusieurs hypertextes. Ces composantes sont organisées autour du curriculum. Le curriculum constitue le squelette de l'environnement d'apprentissage. Les différentes ressources fournies par ETOILE permettront ensuite de construire les activités d'apprentissage autour de cette ossature de base. Le curriculum sera mis en oeuvre par les différents agents du système.

5.1 Définition d'un curriculum

Un curriculum est un graphe de buts connectés par des relations de pré-requis. Chaque but est défini par quatre composantes: les problèmes, le contexte, l'expertise et la théorie (voir figure 2). Les deux premières décrivent un objectif en termes de performance, les deux autres en termes de compétence. Commençons par la **description de l'objectif en terme de performance.**

- Un but se définit comme une **classe de problèmes** que l'élève devra pouvoir résoudre à la fin d'une session. Cette classe doit être définie en extension: l'utilisateur crée un ensemble fini d'exemples pour la classe 'problème' et complète les différents attributs associés à cette classe. Certains attributs sont généraux, tels celui où est mémorisé le texte de présentation du problème ou celui indiquant l'indice de difficulté du problème. L'auteur pourra ajouter des attributs spécifiques au domaine enseigné en créant une ou plusieurs sous-classes de la classe problème.
- Comme dans toute définition d'objectif qui se respecte, l'auteur devra préciser **dans quel contexte l'apprenant sera amené à résoudre les problèmes posés.** Dans le cas présent, le contexte désigne l'ensemble des facilités offertes par l'interface au moment où l'élève est confronté à une tâche. Le contexte désigne donc l'interface: fenêtres, menus, boutons, facilités,... Cette interface doit être programmée au moyen du Common Lisp Interface Manager (CLIM), condition de son intégration avec les autres composantes du système. Dans la plupart des environnements d'apprentissage, l'interface demeure identique tout au long de la progression dans le curriculum. ETOILE permet à l'auteur de spécifier une interface propre à chaque but. Celui-ci pourra par exemple distinguer deux objectifs concernant le même ensemble d'équations si l'interface du second but offre à l'élève des opérateurs algébriques qui n'étaient pas disponibles auparavant. Cette fonctionnalité nous a permis de mettre en oeuvre l'architecture cognitive de MEMOLAB.

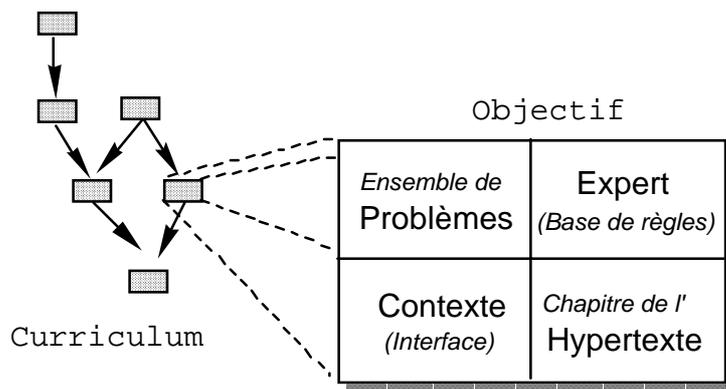


Figure 2: Définition des objectifs dans ETOILE

La **description de l'objectif en termes de compétence** est double. Le même ensemble de connaissances est représenté de façon exécutable, par une base de règles, et de façon inspectable, par un chapitre de l'hypertexte. Certes, la base de règle est inspectable et pourrait être accessible à l'apprenant. En imaginant que l'on résolve les difficultés liées à la lisibilité des règles, il n'en resterait pas moins un conflit entre la structuration d'une expertise en vue de son exécution et en vue de sa présentation. Aussi, avons nous préféré utiliser deux représentations redondantes, mais qui sont chacune plus performantes dans leur fonction.

- Un **expert du domaine** est une base de règles capable de résoudre n'importe quel problème associé au même objectif. L'écriture de ces règles est soumise à des contraintes syntaxiques strictes, mais également à des contraintes liées à notre conception de la collaboration entre l'expert et l'apprenant. L'auteur peut ainsi définir plusieurs experts pour un même but. Si leur expertise est considérée comme équivalente, cela implique que ces experts défendent des points de vue alternatifs sur la tâche. Par contre, si un expert-X est considéré comme quantitativement ou qualitativement inférieur à un expert-Y, l'expert-X devrait être associé à un objectif pré-requis à celui où se situe l'expert-Y.
- Les connaissances de l'expert du domaine doivent aussi être représentées sous une forme lisible par l'apprenant. Chaque compétence sera décrite à différents niveaux de granularité. Pour ce faire, chaque objectif est associé à un **chapitre de l'hypertexte**. En outre, les règles (en particulier les règles de réparation) peuvent être directement associées à un noeud de ce même hypertexte.

5.2. Le moteur d'inférence.

Chaque agent dans ETOILE dispose de sa base de règle. Ces bases sont exécutées par un moteur d'inférence que nous avons développé pour répondre aux exigences générées par notre conception particulière de ETOILE. Ce moteur d'inférence, écrit en Lisp orienté objet (CLOS), est lui-même orienté objet: la base

de faits est constituée de l'ensemble des objets en vie dans le système. Les conditions des règles se réfèrent aux valeurs des attributs des objets de différentes classes. Les conclusions des règles modifient ces mêmes attributs-valeurs.

Cette approche permet d'intégrer les diverses composantes d'un même système. Toute composante étant implémentée sous la forme d'objets, il est par exemple possible d'ouvrir un chapitre de l'hypertexte dans la conclusion d'une règle, ou inversement d'activer une base de règle depuis un noeud de l'hypertexte (pour exécuter un exemple associé au texte présenté, par exemple). Plus important encore, cette forte intégration permet un lien étroit entre les actions de l'expert et les événements qui se produisent à l'écran.

Par contre, cette universalité pourrait se révéler très coûteuse au moment de l'exécution (un nombre trop important de faits pouvant instancier les règles). Pour éviter ce problème, l'utilisateur partitionnera l'ensemble des faits et celui des règles. Partitionner les objets est chose aisée dans un environnement orienté-objet: il suffit de spécialiser les règles sur une sous-classe quelconque. Quant aux règles, elles peuvent être partitionnées en bases de règles et, à l'intérieur d'une base, en paquets de règles, qui sont hermétiquement dissociés des autres paquets ou bases et réduisent donc considérablement le nombre de règles à considérer à chaque cycle.

6. Conclusion

ETOILE ne constitue pas un outil fini, prêt à être distribué à un large public. En particulier, il ne dispose pas d'un interface-auteur. La création d'un système au moyen d'ETOILE exige que l'auteur développe les composantes spécifiques en Common Lisp et qu'il puisse écrire des jeux de règles. ETOILE est un outil de recherche. L'économie de création d'un interface pour l'auteur nous a permis d'investir nos efforts dans la représentation de compétences pédagogiques. Notre intérêt s'est porté sur l'observation des difficultés rencontrées pour abstraire des règles et sur les conceptions sous-jacentes de l'espace didactique. Ces efforts ont conduit à une conception originale des interactions entre les agents, que l'on peut résumer comme suit. L'interaction entre l'expert et l'apprenant est centrée sur l'interface et est "adidactique", certaines interactions adidactiques sont interceptées par le tuteur qui leur attribue un sens selon ses propres intentions didactiques. L'influence qu'a le tuteur sur les interactions expert-apprenant est déterminée par le style d'enseignement du tuteur. La conception de la collaboration entre l'expert et l'élève établit un lien entre les travaux en matière d'outils auteur et le courant 'cognition située'. Le choix d'un style d'enseignement est effectué par le coach sur base de l'efficacité de ces styles au cours du travail préalable de l'apprenant.

Références

ANDERSON, J.R.; FARELL, R.G.; and SAUERS, R. [1984] Learning to program in LISP. Cognitive Science, vol. 8, no 2, pp. 87-129.

CASE R. [1985] *Intellectual Development: from Birth to Adulthood*. New York: Academic Press.

DILLENBOURG, P. [1992b] The Language Shift: a mechanism for triggering metacognitive activities. in P. Winne and M.Jones. *Adaptive Learning Environments.: foundations and frontiers*. Springer-Verlag. Hamburg

DILLENBOURG, P.; et MENDELSON P. [à paraître]. The genetic structure of the integration space. In E. Costa [Ed], *New Directions for Intelligent Tutoring*. Hamburg: Springer-Verlag.

DILLENBOURG, P. & SELF, J.A. (1992) A computational approach to socially distributed cognition. *European Journal of Psychology of Education*. Vol. 3, N 4.

DILLENBOURG, P., HILARIO, M., MENDELSON, P., SCHNEIDER D. and BORCIC, B. [1992] The Memolab Project. Intermediate Report. TECFA Document. TECFA, University of Geneva.

MENDELSON P. [1991] Les modèles cognitifs dans les environnements d'apprentissage: implémentation ou contemplation? *Actes des 13 èmes Journées Francophones sur l'Informatiques*. Université de Genève, Suisse. pp 53-60.

NEWMAN, D. [1989] Is the Student necessary ? Apprenticeship as a model for ITS. *Proceedings of the 4th International Conference on AI and Education*. May 1989, Amsterdam, Netherland. pp 177-184.

RESNICK L., LEVINE J., and TEASLEY S. (1991). *Perspectives on Socially Shared Cognition*. Hyattsville, MD: American Psychological Association.

SCHRAGE M. (1990). *Shared Minds*. New York: Random House