

## Les logiciels d'apprentissage : panoplie ou éventail ?

Erica Vries (de)

► **To cite this version:**

Erica Vries (de). Les logiciels d'apprentissage : panoplie ou éventail?. Revue Française de Pédagogie, INRP/ENS éditions, 2001, <<http://www.upmf-grenoble.fr/sciedu/edevries/publicat.htm>>. <edutice-00000038>

**HAL Id: edutice-00000038**

**<https://edutice.archives-ouvertes.fr/edutice-00000038>**

Submitted on 7 Oct 2003

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## Les logiciels d'apprentissage : panoplie ou éventail ?

*Sorting out environments for computer-supported learning*

**Erica de Vries**

Laboratoire des Sciences de l'Éducation, Université Pierre-Mendès-France de Grenoble

### *Résumé*

L'objectif de cet article est d'élaborer une typologie des logiciels d'apprentissage. Un premier problème qui se pose est celui de la diversité des logiciels et des multiples façons dont on pourrait les classer. La typologie proposée est fondée sur la fonction pédagogique visée par les enseignants ou les concepteurs. En outre, nous proposons une caractérisation sur trois aspects : les tâches proposées aux élèves, le point de vue théorique sous-jacent à la conception et la manière dont sont traitées les connaissances. La typologie comprend huit fonctions pédagogiques qui correspondent globalement aux types de logiciels que l'on peut trouver dans la littérature sur l'apprentissage assisté par ordinateur. Ensuite, nous examinons la nécessité, l'exhaustivité et l'exclusivité des catégories. Enfin, la dernière section présente trois approches de comparaison entre logiciels. Une implication est qu'une évaluation de logiciels éducatifs exige une adaptation des mesures de performance aux fonctions pédagogiques recherchées.

### *Abstract*

The main goal of this article is to elaborate a typology in the domain of computer-supported learning. The first problem that has to be addressed concerns the diversity of computer-supported learning programs and the multiple ways in which one could classify them. The proposed typology is based on the pedagogical function aimed at by the teachers or designers. Moreover, we propose a characterisation on three aspects : the tasks proposed to the learners, the theoretical viewpoint underlying the design and the way in which domain knowledge is dealt with. The typology comprises eight pedagogical functions that globally correspond to the types of programs that one can find in the literature on computer-supported learning. We subsequently examine the necessity, exhaustiveness and the exclusiveness of the categories. Finally, the last section presents three alternative approaches to the comparison between computer-supported learning programs. An emerging implication is that evaluation of educational computer programs demands an adaptation of performance measures to the pedagogical functions aimed at.

## INTRODUCTION

Aujourd'hui, on peut qualifier de panoplie l'ensemble des termes désignant l'utilisation de l'ordinateur dans l'enseignement. Parmi les termes de cette collection en désordre, on peut trouver les technologies de l'information et de la communication dans l'enseignement (TICE), le multimédia éducatif et l'Internet pédagogique. La prolifération des ouvrages, des séminaires et des colloques portant ces intitulés appelle à réfléchir sur un moyen de structurer ce domaine des *nouvelles technologies éducatives*. Cette structuration est d'autant plus urgente qu'elle permettrait aux membres des diverses communautés de concepteurs, de praticiens, d'utilisateurs et de chercheurs de se retrouver autour d'une même problématique. En cela disant, nous admettons qu'il y ait non pas une problématique, mais bien plusieurs problématiques concernant l'apprentissage assisté par ordinateur.

Comme nous allons le montrer, il y a d'abord des critères de classement de logiciels qui se situent au niveau des aspects de surface concernant par exemple la technologie logicielle employée.

Or, notre but est de faire un classement basé sur des critères ayant une pertinence pour les sciences de l'éducation. Ce classement devra montrer en quoi se distinguent les différentes manières d'exploiter l'ordinateur pour l'enseignement et pour l'apprentissage. Ainsi, l'objectif de cet article est à la fois d'élaborer des aspects pertinents pour qualifier l'utilisation de l'ordinateur pour l'apprentissage et de les mettre en œuvre pour proposer une typologie des logiciels éducatifs. La typologie sera ensuite évaluée selon les critères de nécessité, d'exclusivité et d'exhaustivité des catégories. Enfin, nous allons nous arrêter brièvement sur une approche comparative. A ce propos, nous aborderons la question des mérites des différents types d'utilisation de l'ordinateur pour l'apprentissage.

## ENSEIGNEMENT ET INFORMATIQUE

En plus des termes mentionnés plus haut et dans le but de désigner plus précisément la relation enseignement-ordinateur, nous pouvons citer les abréviations EAO (*Enseignement Assisté par Ordinateur*), EIAO pour initialement *Enseignement Intelligemment Assisté par Ordinateur* et plus tard pour *Environnement Interactif d'Apprentissage avec Ordinateur* (Baron, Gras et Nicaud, 1991), ou encore EGO (*Enseignement Géré par Ordinateur*) et EBO (*Enseignement Basé sur Ordinateur*). Plus généralement, l'objectif de l'introduction de l'ordinateur dans l'enseignement est de permettre le développement d'une culture informatique et de favoriser le processus d'apprentissage (Gabriel, 1998). Aujourd'hui, le concept intégrateur d'EIAH (*Environnement Informatique pour l'Apprentissage Humain*) est utilisé soulignant à la fois le côté informatique et le côté humain de l'entreprise (Vivet, 1997). Dans cette section, nous opérons une première organisation en prenant en compte successivement la diversité des logiciels connus et la place qu'occupe l'ordinateur dans le système éducatif.

### Une diversité de logiciels éducatifs

Nous constatons une très grande diversité de logiciels et d'utilisations potentielles de l'ordinateur pour l'apprentissage. Si on voulait en dresser un bilan, on pourrait les classer selon la *matière* (maths, français, histoire, géographie...), selon le *niveau scolaire* des élèves (enseignement primaire, secondaire, supérieure, formation professionnelle et continue...), et selon la *technologie* matérielle ou logicielle employée (Internet, hypermédia, CD-ROM, réseaux...). Ce classement pourrait se faire d'ailleurs à l'aide des fiches techniques livrées avec les logiciels mêmes. Par conséquent, il ne se distinguerait guère d'un catalogue de vente. Chaque logiciel éducatif se situerait de façon naturelle dans une cellule du classement (ex. logiciel d'exploration d'Internet pour la géographie en 6<sup>e</sup>, un CD-ROM pour les maths en CM2, etc.). Son grand défaut se situe dans le manque d'informations sur la *façon* dont la matière est enseignée. C'est précisément la méthode d'enseignement qui nous paraît un aspect important pour qualifier l'utilisation de l'ordinateur pour l'apprentissage.

Du point de vue de la recherche, les revues de question se focalisent souvent sur une technologie en particulier, indépendamment de la matière ou du niveau d'enseignement. A titre d'exemple, on peut trouver des articles traitant de l'efficacité des hypermédias pour l'apprentissage (cf. Dillon & Gabbard, 1998; Tergan, 1997) ou de l'enseignement à distance (cf. Dessus, Lemaire, & Baillé, 1997). Parallèlement, des inventaires et des classifications de logiciels éducatifs mentionnent différents *types de logiciels* (Alessi & Trollip, 1991; Bruillard, 1997; Collis, 1996; Déro, 1996; Sandberg & Barnard, 1993), comme les hypermédias, les simulations, les micro-mondes, les tuteurs intelligents et les tutoriels. Certaines de ces classifications sont établies en référence à l'évolution historique (voir aussi Saettler, 1990), des courants de recherche (voir aussi Baron, 1990) ou encore des progrès technologiques. Une distinction importante par rapport à l'évolution historique est celle qui oppose le concept behavioriste du concept cognitiviste dans les technologies éducatives (Saettler, 1990). Une différenciation dans les courants de recherche sépare ceux qui cherchent à créer des outils et à exploiter de nouvelles possibilités techniques de ceux qui étudient comment des objectifs pédagogiques peuvent être atteints (Baron, 1990). Malgré l'intérêt de ces classifications et de cette

dernière distinction, la question se pose comment réagir face à l'éclatement des possibilités techniques et au foisonnement des objectifs pédagogiques annoncés auxquels on assiste aujourd'hui.

### **La place de l'ordinateur dans l'enseignement**

Dans cet amalgame technologie - type de logiciel - utilisation - méthode, il s'agit de trouver l'aspect ou les aspects qui permettraient d'organiser, de structurer, de classer les logiciels éducatifs. Tout d'abord, en suivant Baron et Bruillard (1996), nous pouvons identifier trois acceptions de l'informatique dans l'enseignement. La première est l'ordinateur comme *médium* : de nombreux logiciels sont conçus pour être utilisés dans un but d'acquisition de connaissances par les élèves. Ils constituent une méthode ou technique conçue pour instrumenter les activités d'enseignement. Le rôle de l'ordinateur est ainsi spécifique à des situations d'enseignement. Deuxièmement, l'ordinateur et l'informatique peuvent être vus comme *matière*. Aujourd'hui, la gestion et le traitement d'informations à l'aide de l'ordinateur sont devenus importants dans presque tous les domaines. L'enseignement de ces processus de gestion et de traitement avec des logiciels (ou même des progiciels) est devenu un enjeu important et de plus fortement lié à l'enseignement des contenus dans ces mêmes domaines (Baron & Bruillard, 1996, 2001). Troisièmement, on peut voir l'ordinateur dans l'enseignement en tant qu'*outil de production*. Des logiciels tels que des traitements de textes sont utilisés avec un objectif lié au résultat, i.e. un texte. Dans cette vision, l'ordinateur n'est qu'un ustensile plus ou moins complexe permettant de produire (Baron & Bruillard, 1996).

Les trois acceptions incorporent en définitive différentes interprétations de l'ordinateur comme instrument dans une situation pédagogique. Or, la notion même d'instrumenter l'enseignant n'est pas évidente. Sous-jacente à cette notion est l'idée qu'il y aurait à la fois un certain *besoin d'intermédiaire* entre l'enseignant et l'apprenant et un *processus de conception* visant à satisfaire ce besoin à travers le développement d'artefacts. Il semblerait précisément que les logiciels éducatifs sont conçus pour répondre à des besoins aussi différents que de vouloir montrer, faire découvrir, faire s'entraîner, etc. Nous proposons dans cet article un approfondissement de cette notion de besoin à satisfaire, c'est-à-dire du moteur à l'origine de la création des différents logiciels éducatifs. D'ailleurs, nous verrons plus loin que ces besoins peuvent s'apparenter à des méthodes pédagogiques. En ce faisant, nous nous restreindrons à l'étude de l'ordinateur selon la première acception, c'est-à-dire comme médium dans la transmission de savoirs, et dans une moindre mesure à la troisième acception, l'ordinateur en tant qu'outil de production. Bien que nous ne nions pas l'importance de la deuxième acception, l'ordinateur en tant que matière, elle trouve moins sa place dans cet article. L'existence de différents instruments réalisés avec un même objet, l'ordinateur, est due à un aspect singulier : la possibilité de créer des artefacts fondamentalement différents pour satisfaire au moins autant de besoins.

### **L'ORDINATEUR : UN ENSEMBLE D'ARTEFACTS**

La caractéristique essentielle de l'ordinateur est qu'il est programmable pour construire des artefacts fondamentalement différents. C'est ce que nous pouvons appeler une machine à construire des machines ou un méta médium (Kay, cité par Bruillard, 1997). Cette caractéristique est rendue possible par la séparation entre la partie matérielle et la partie logicielle de l'ordinateur. Alors qu'un outil est généralement conçu pour remplir une fonction précise (un marteau pour taper), la partie logicielle d'un ordinateur peut être conçue pour remplir des fonctions multiples. Elle est le fruit d'un processus de conception qui, partant d'un besoin global, passant par des fonctions distinctes, aboutit à des spécifications concrètes. Dans le domaine de la conception, les fonctions qui découlent du besoin global sont appelées des *exigences de prestation* et l'on emploie le terme *matérialisations* pour les spécifications concrètes (de Vries & de Jong, 1999). Dans le cas de la conception d'environnements informatiques pour l'apprentissage humain, le besoin global qu'un enseignant ou un concepteur doit satisfaire est celui de faire apprendre. La conception de ces environnements nécessite une interprétation de ce besoin en diverses exigences de prestation : des moyens pour présenter de l'information, pour donner des exercices, etc. Les exigences de prestation sont ensuite traduites en

matérialisations : fournir textes et images (pour présenter de l'information), concevoir des invites pour recueillir des réponses (pour les exercices), etc. Ainsi, de multiples environnements sont possibles selon les différentes fonctions considérées lors de la conception. La diversité des logiciels éducatifs provient du fait qu'il existe des spécifications concrètes très variées et des fonctions très différentes pour satisfaire le même besoin global de faire apprendre.

De toute évidence, le point de vue théorique sur l'apprentissage sous-jacent à la conception de batteries d'exercices n'est pas le même que celui sous-jacent à la conception de tuteurs intelligents. Un logiciel éducatif peut donc être vu comme une implémentation d'un point de vue théorique particulier (voir aussi Duffy & Jonassen, 1991). Nous observons que ce point de vue est souvent implicite et sa fonction dérivée rarement fondée sur des résultats de recherche. Inversement, un point de vue théorique explicite ne commande pas de façon déterministe la conception de fonctions pédagogiques et de spécifications particulières. Enfin, bien qu'il existe des études ponctuelles sur l'efficacité de tel ou tel logiciel en tant qu'implémentation d'un point de vue théorique, les connaissances scientifiques sur ce sujet sont ni développées, ni structurées, ou en tout cas ne donnent pas de prescriptions claires pour l'utilisation de l'ordinateur pour l'apprentissage. Notre démarche est donc d'élaborer la notion de fonctions à remplir par l'ordinateur dans des situations visant un apprentissage. Notons que nous nous basons dans un premier temps sur la fonction visée lors de la conception d'un artefact. Une approche complémentaire serait de regarder de plus près l'usage effectif, puisqu'un même logiciel peut mener à différentes *utilisations* et/ou être intégré dans différents *usages*. Par exemple, un hypermédia peut être utilisé pour faire explorer une vaste base de données, pour donner un accès à une information enrichie sur un sujet donné, pour faire personnaliser une base de données ou même pour en faire construire un par les élèves eux-mêmes (Duffy & Knuth, 1990). Ainsi, les enseignants peuvent organiser différents types d'activités d'élèves. Il nous paraît néanmoins essentiel pour une organisation des recherches sur l'apprentissage à l'aide de l'ordinateur de faire une typologie des fonctions pédagogiques que peuvent remplir les logiciels éducatifs. La typologie présentée ci-dessous prend en compte les fonctions pédagogiques visées lors de la conception. Un des éléments de ce choix est que nous pouvons distinguer différents points de vue théoriques sur l'apprentissage à l'origine de la conception qu'il convient d'explicitier. Parallèlement, il faut garder à l'esprit que l'aboutissement de ce travail nécessite une étude des usages effectifs qui en partie se référeront aux utilisations prévues dans la typologie. Cette dernière permettrait alors d'établir des relations entre utilisations prévues et usages effectifs et ainsi de dégager de nouvelles problématiques de recherche.

## UNE TYPOLOGIE DES LOGICIELS EDUCATIFS

La section précédente montre qu'un logiciel éducatif peut être caractérisé par la fonction visée lors de sa conception. Cette fonction découle d'un point de vue théorique et mène aux spécifications du logiciel. Elle traduit la volonté de l'enseignant et/ou du concepteur de créer l'environnement idéal pour les élèves.

### Les huit fonctions pédagogiques

La typologie que nous proposons comprend huit fonctions pédagogiques. Chacune des fonctions présentées correspond globalement à un des types de logiciels mentionnés précédemment. Elles se distinguent principalement dans le rôle joué par l'ordinateur indépendamment de la technologie employée. Nous proposons de caractériser les fonctions sur trois aspects.

D'abord, nous examinons les *tâches* proposées aux élèves, comme par exemple lire, faire des exercices, tâches qui sont très fortement liées au rôle de l'ordinateur. Gardons en mémoire qu'il s'agit ici des tâches prévues ou prescrites dans une situation d'apprentissage. Elles jouent un rôle plus ou moins hypothétique dans les processus d'apprentissage.

Ensuite, chaque fonction est l'expression d'un *point de vue théorique* sur l'enseignement et l'apprentissage. Les principaux points de vue rencontrés sont le béhaviorisme, le cognitivisme, le

constructivisme et la cognition située. Pour le besoin de la typologie, ils seront présentés dans leur forme extrême. D'une part, leurs divergences paraîtront exagérées. D'autre part, ce sont précisément ces formes non nuancées qui sont souvent émises en relation avec les prétentions sur l'efficacité de tel ou tel logiciel éducatif.

Enfin, le point de vue théorique influence la façon dont sont traitées les *connaissances* (voir aussi Linard, 1996). Chaque type de logiciel incarne une façon particulière de considérer la connaissance du domaine à enseigner. En fait, ces trois aspects, la tâche proposée, le point de vue théorique et le statut des connaissances, ont évolué en partie suite au développement des technologies éducatives. C'est une des raisons pourquoi nous pensons qu'ils sont adaptés pour caractériser les différents types de logiciels éducatifs.

### *Présenter de l'information*

Une fonction évidente de l'ordinateur dans l'enseignement est celle de présenter de l'information sur l'écran. Les termes pour désigner ce type de logiciel sont : *support de cours* ou *tutoriel*. Les logiciels remplissant cette fonction présentent des pages écran avec, comme dans un manuel scolaire, du texte, des explications, éventuellement alternés avec des questions auxquelles l'élève doit répondre avant de pouvoir continuer. La tâche proposée aux élèves est donc de *lire* et d'étudier ce qui est présenté sur l'écran. Son activité concrète se limite à des actions ayant pour effet de tourner des pages (changer l'affichage sur l'écran) et de répondre à des questions.

Le point de vue théorique incarné est *cognitivist*. Il s'agit d'un apprentissage comme une acquisition de nouvelles connaissances. Dans cette vision (voir par exemple Mayer, 1987), le mode de présentation de l'information influence les processus d'apprentissage (attention, encodage, récupération) et ces processus à leur tour influencent le résultat de l'apprentissage (prestation, performance). Ce point de vue est objectiviste dans le sens où l'on considère que le monde peut être structuré et organisé en entités, propriétés et relations, indépendamment de l'expérience individuelle. Apprentissage égale alors acquisition de cette structure objective et un logiciel en tant que support de cours doit la présenter au mieux. Gagné (1985) et Merrill (1983) ont formulé des théories sur la façon de présenter des connaissances et leurs recommandations ont été utilisées pour concevoir des séquences d'enseignement par ordinateur. TICCIT développé par l'équipe de Merrill en est un exemple.

La position prise par rapport aux connaissances est qu'il faut choisir les informations pertinentes et en élaborer une *présentation ordonnée*. Cette position a été baptisée « encodage implicite » des connaissances (Wenger, 1987) ; le logiciel est le résultat à la fois d'une structuration des connaissances par un pédagogue expert dans la matière et de ses décisions sur la meilleure façon de les présenter.

### *Dispenser des exercices*

La deuxième fonction est celle de dispenser des exercices. Ce type de logiciel est connu sous le nom d'*exercices répétés* (*drill* en anglais). Le rôle joué par l'ordinateur est celui de stockage et de distribution de multiples exercices. La tâche proposée à l'élève est de *faire des exercices*. Selon Alessi et Trollip (1991), les exercices répétés ne sont appropriés qu'après avoir suivi un enseignement classique. L'objectif est que l'élève s'entraîne pour obtenir aisance et vitesse dans une matière.

Le point de vue théorique sous-jacent est *béavioriste*. Cette théorie postule une relation directe entre les conditions d'enseignement et le résultat de l'apprentissage (sans processus intermédiaires, Mayer, 1987). Les items (exercices) sont considérés comme des stimuli et les actions de l'élève comme des réponses. La récompense (renforcement) réside dans le fait d'avoir donné la bonne réponse, de pouvoir passer à l'item suivant, ou, dans certains cas, d'obtenir un événement auditif ou visuel de la part de l'ordinateur. Les premières « machines à enseigner » (des machines mécaniques au début) ont été développées par Crowder (1960), Pressey (1926) et Skinner (1958) cités par Burton,

Moore, & Magliaro (1996). Burton et ses collègues donnent une excellente revue du béhaviorisme et l'enseignement programmé (voir aussi Bruillard, 1997 et Depover, 1987).

Le point de vue béhavioriste pose que les connaissances doivent être définies en termes de performances et de comportements observables. Ainsi, le comportement désiré est connu d'avance ; ce qui est à acquérir par l'élève, ce sont les associations entre items et bonnes réponses. On peut dire que les connaissances sont exemplifiées en *associations à former* par l'élève. Aujourd'hui, il existe de multiples logiciels offrant des exercices répétés.

### *Véritablement enseigner*

La troisième fonction est de faire d'un ordinateur un enseignant, c'est-à-dire d'exploiter l'ordinateur pour exécuter une tâche qui est habituellement exécutée par un humain. L'adverbe *véritablement* souligne qu'il s'agit ici de la fonction de l'enseignant humain lui-même (non instrumentée par l'ordinateur). Le rôle de l'ordinateur peut alors être celui d'un entraîneur, d'un guide, d'un tuteur, d'un expert, d'un enseignant ou même d'un autre élève. Ce type de logiciels issu de l'intelligence artificielle est connu sous le nom de *tuteurs intelligents*. Pour enseigner tel un enseignant humain, un tuteur informatique doit posséder les connaissances du domaine à enseigner, s'adapter aux connaissances et erreurs de l'élève, adopter une stratégie pédagogique et pouvoir communiquer avec l'élève. Ces capacités correspondent aux quatre modules d'un tuteur intelligent : le modèle du domaine, le modèle de l'élève, le module pédagogique et l'interface. Les tâches proposées à l'élève sont diverses ; la plupart permettent à l'élève de *dialoguer* avec un tuteur, mais il existe également des logiciels qui proposent à l'élève de jouer guidé par un tuteur. Notons que la fonction *enseigner comme un humain* en elle-même sera traduite, pour un tuteur intelligent précis, en une fonction spécifique qui dépendra de la stratégie pédagogique impliquant un comportement humain (guidage, questionnement, etc.).

Le point de vue sous-jacent est *cognitivist* puisque l'approche de l'intelligence artificielle est basée sur l'idée selon laquelle on peut représenter les connaissances d'un expert de façon à ce que l'ordinateur puisse les posséder et les manipuler. Au moyen de la stratégie pédagogique, le logiciel fera ensuite acquérir ces connaissances à l'élève. A l'aide du modèle de l'élève, il pourra vérifier la part des connaissances expertes acquises. Il s'agit donc d'un appariement des connaissances de l'expert, de l'ordinateur, et de l'élève.

Les connaissances sont idéalement incorporées dans le logiciel indépendamment de la méthode d'enseignement choisie. Elles sont intégrées en tant que *représentation* des connaissances du domaine à enseigner. A l'inverse d'un support de cours, cette façon de traiter les connaissances a été baptisée « encodage explicite » des connaissances (Wenger, 1987) ; le logiciel possède à la fois une représentation des connaissances et des procédures pour exploiter ces connaissances en fonction des décisions du module pédagogique. De ce fait, un tuteur intelligent doit être considéré comme une méta-fonction pédagogique, puisque, comme en témoigne les exemples cités ci-après, des stratégies pédagogiques variées peuvent être implémentées.

Les exemples classiques de tuteurs intelligents sont SCHOLAR (Carbonell, 1970) proposant un dialogue en initiative mixte sur la géographie de l'Amérique du Sud, WHY (Stevens & Collins, 1977) proposant un dialogue socratique sur la météorologie et WEST (Brown & Burton, 1975) fournissant un guidage discret lors d'un jeu nécessitant des opérations mathématiques (tous décrits par Wenger, 1987 et Bruillard, 1997).

### *Captiver l'attention et la motivation de l'élève*

Une quatrième fonction de l'ordinateur dans l'enseignement peut être de motiver les élèves en exploitant les défis associés aux jeux. C'est l'objectif des logiciels que l'on appelle communément des *jeux éducatifs*. Le rôle de l'ordinateur est celui d'une console de jeu et la tâche proposée à l'élève est de *jouer*. La volonté de motiver l'élève peut conduire à des réalisations concrètes très différentes (voir aussi Rieber, 1996). Un jeu peut prendre la forme d'une série de questions - réponses, de recherche d'objets ou de résolution de problèmes.

Le point de vue théorique emprunte des notions des théories de motivation pour l'enseignement (la théorie de Malone, cité dans Alessi et Trollip (1991) par exemple) mais beaucoup de jeux éducatifs sous forme de séries de questions - réponses s'apparentent à des exercices répétés et répondent à un point de vue plutôt *behavioriste*. Le challenge posé à l'élève et la récompense pour une bonne performance sont des principes importants dans les jeux. Par rapport aux exercices répétés, les jeux éducatifs proposent davantage un habillage de questions suscitant la curiosité et l'imagination.

#### *Fournir un espace d'exploration*

La cinquième fonction préconise d'exploiter l'ordinateur pour rendre disponible du texte, du son, des images, et des vidéos. Cette fonction est possible grâce aux liens informatiques matérialisés comme des boutons ou des mots soulignés sur l'écran. Ce type de logiciel est connu sous le nom d'*hypermédia* ou de *multimédia*. Le rôle de l'ordinateur est de fournir un espace d'exploration correspondant aux concepts et relations d'une matière et la tâche proposée à l'élève est d'*explorer* cet espace. La différence avec un support de cours est que l'information est organisée en concepts et relations, mais que l'ordre de présentation n'est pas défini à l'avance. L'ordre dans lequel les informations vont apparaître sur l'écran sera décidé par l'élève.

Le point de vue théorique adopté se situe entre un point de vue *cognitivist* et *constructivist* selon les cas. Certains auteurs soulignent des aspects plutôt cognitivist lorsque l'on suppose que le logiciel présente les concepts et relations d'une matière sous forme de réseau sémantique à acquérir par l'élève (van Oostendorp & Hofman, 1998). D'autres soulignent des aspects constructivist lorsque l'on accentue le côté actif de l'élève qui doit à chaque instant décider comment procéder et qui, chemin faisant, construit ses propres connaissances (Spiro, Feltovich, Jacobson, & Coulson, 1991).

Dans un espace d'exploration, les connaissances sont exposées dans une *présentation en accès libre*. Il y a de multiples exemples de systèmes, à la fois issus de la recherche, tels que *Citizen Kane* (Spiro & Jehng, 1990) et *PERSEUS* (Neuman, Marchionini, & Morrell, 1995) et dans le commerce (encyclopédies, CD-ROM culturels).

#### *Fournir un environnement pour la découverte de lois naturelles*

Une sixième fonction de l'ordinateur prévoit de laisser l'élève découvrir des lois de par lui-même. Ce type de logiciel, appelé *simulation*, imite une partie de la réalité. Les simulations incorporent des lois ou des règles de fonctionnement, par exemple relatives à la chute d'un objet, à la croissance d'une population ou aux procédés de fabrication dans une usine. L'élève apprend en agissant sur une simulation d'une façon similaire à la façon dont il agirait dans une situation réelle. Il peut changer la valeur des variables et en observer les effets sur d'autres variables. Les tâches à exécuter par les élèves sont de *manipuler*, d'*observer*, et d'interpréter les résultats.

Le point de vue théorique est *constructivist* ; les élèves, par motivation intrinsèque, recherchent activement l'information. La compréhension est plus que l'absorption simple ; les élèves structurent et organisent l'information. La connaissance existe qu'en étant construite par une personne et n'a une signification que par cette construction. Ainsi, les simulations prônent un apprentissage par la découverte et par l'action (*learning by doing*). Le point de vue est également celui de la cognition située ; promouvoir des activités authentiques dans des situations ressemblant à la réalité (Brown, Collins, & Duguid, 1989). Selon cette théorie, les activités authentiques créent davantage d'opportunités pour construire de connaissances exploitables dans des situations futures (transfert).

Dans les simulations, les connaissances existent en tant que *modélisation* ; les lois du domaine gouvernent les réactions du logiciel. Des exemples de simulations sont *QUEST* (White & Frederiksen, 1989) et *SHERLOCK* (Lesgold, Lajoie, Bunzo, & Eggan, 1992) cités dans une revue de questions sur les simulations par de Jong & van Joolingen (1998).



*Fournir un environnement pour la découverte de domaines abstraits*

La septième fonction ressemble à la sixième à la différence que l'élève découvre des domaines abstraits. Ce type de logiciel est appelé *micro-monde*. Ces environnements peuvent fournir une rétroaction qui n'a pas *a priori* d'équivalent dans la réalité. Les objets manipulés dans un micro-monde ont des propriétés communes à la fois avec des objets formels du domaine et avec les objets concrets du monde réel (voir aussi Bruillard, 1997). La tâche proposée à l'élève est de *construire* et de manipuler ces objets.

Le point de vue théorique sous-jacent est *constructiviste* : l'élève apprend en construisant et en interprétant les réactions procurées. Les connaissances sont comme *matérialisées* par l'informatique. Parmi les exemples, nous citons LOGO pour la programmation (Papert, 1980), CABRI pour la géométrie (Laborde, 1995), et *Tarski's World* pour la logique du premier ordre (Barwise & Etchemendy, 1990).

*Fournir un espace d'échange entre élèves*

Enfin, la huitième fonction mentionnée ici se fixe comme objectif de permettre aux élèves d'apprendre en communiquant. Ce type de logiciel d'*apprentissage collaboratif* (CSCL - *Computer-supported collaborative learning*) est conçu pour être utilisé par plusieurs élèves côte à côte ou à distance, en synchrone ou en asynchrone. L'ordinateur peut avoir le rôle de mémoire collective, de point de fixation du dialogue et de l'action, de représentation des éléments de la discussion ou encore de médium de communication. La tâche proposée aux élèves est de *discuter*, d'argumenter et/ou d'écrire des textes et de résoudre des problèmes ensemble.

Le point de vue est celui de la *cognition située*. Un des objectifs est de créer un contexte permettant aux élèves d'apprendre à communiquer au sujet d'un domaine, comme une activité authentique équivalant à l'échange entre personnes exerçant le même métier (Lave & Wenger, 1991). Ainsi, les connaissances ne sont pas présentes initialement dans le logiciel mais seront *construites par l'élève* lui-même en interaction avec d'autres élèves.

Les exemples suivants accentuent les rôles mentionnés ci-dessus : CSILE (Scardamalia & Bereiter, 1994) en tant que mémoire collective, *Envisioning Machine* (Roschelle, 1992) en tant que point de fixation du dialogue et de l'action, BELVEDERE (Suthers & Weiner, 1995) en tant que représentation des éléments de la discussion et C-CHENE (Baker & Lund, 1997) en tant que médium de communication.

Tableau 1. *Les huit fonctions pédagogiques et leurs caractéristiques*

<b>Fonction pédagogique</b>	<b>Type de logiciel</b>	<b>Théorie</b>	<b>Tâche</b>	<b>Connaissances</b>
Présenter de l'information	tutoriel	cognitiviste	lire	présentation ordonnée
Dispenser des exercices	exercices répétés	behavioriste	faire des exercices	association
Véritablement enseigner	tuteur intelligent	cognitiviste	dialoguer	représentation
Capter l'attention et la motivation de l'élève	jeu éducatif	principalement behavioriste	jouer	
Fournir un espace d'exploration	hypermédia	cognitiviste constructiviste	explorer	présentation en accès libre
Fournir un environnement pour la découverte de lois naturelles	simulation	constructiviste cognition située	manipuler, observer	modélisation
Fournir un environnement pour la découverte de domaines abstraits	micro-monde	constructiviste	construire	matérialisation
Fournir un espace d'échange entre élèves	apprentissage collaboratif	cognition située	discuter	construction de l'élève

### **Nécessité, exhaustivité et exclusivité**

Le Tableau 1 présente le résumé des huit fonctions de la typologie et de leurs caractéristiques. Pour observer ses qualités et défauts, nous examinons la typologie à l'aide des critères de nécessité, d'exhaustivité et d'exclusivité. D'abord, les huit fonctions sont *nécessaires* puisqu'elles regroupent les logiciels décrits dans la littérature depuis le début de l'enseignement assisté par ordinateur. Néanmoins, la quatrième fonction, captiver l'attention et la motivation (jeux éducatifs), pourrait être supprimée. Cette fonction n'est pas exactement semblable aux autres parce qu'elle doit elle-même être précisée davantage en termes de fonctions plus spécifiques. Ensuite, étant donné la rapidité du développement technologique, nous ne pouvons garantir l'*exhaustivité* de la typologie. De nouveaux types de logiciels peuvent se rajouter et il s'agirait alors pour chaque type de déterminer la fonction pédagogique visée par ces concepteurs. Enfin, les huit fonctions ne sont pas *exclusives*. Les logiciels développés aujourd'hui incorporent souvent plusieurs fonctions. Certains des logiciels mentionnés plus haut sont précisément des logiciels hybrides : CSILE, un logiciel d'apprentissage collaboratif, invite les élèves à créer des liens hypermédia, SHERLOCK, une simulation, incorpore un tuteur intelligent, WEST, un tuteur intelligent, se présente sous forme de jeu et les micro-mondes tels que LOGO et CABRI, sont souvent utilisés en collaboration par des binômes. C'est la fonction accentuée par les enseignants-concepteurs qui est prise en compte dans le classement. Dans le paragraphe suivant, nous en verrons plus précisément l'intérêt pour la comparaison et l'évaluation des différentes approches.

### **QUELLE APPROCHE POUR EVALUER OU COMPARER ?**

Jusqu'à-là, nous avons évité de comparer les huit fonctions ou de les évaluer ; nous les avons énumérées comme autant d'approches complémentaires. Il paraît difficile de juger comment on apprend ou enseigne le mieux ; cela dépend de ce que l'on entend par bien apprendre. Est-ce connaître par cœur des définitions, de pouvoir traduire des mots d'anglais, de savoir participer à une discussion sur un concept difficile ? Plus les logiciels divergent au point de vue théorique, plus il est difficile de trouver un test commun pour les comparer. Pour élucider cette question de la comparaison entre logiciels et de leur évaluation, nous présentons brièvement trois voies alternatives.

#### **Exploiter au mieux les capacités de l'ordinateur**

La première voie se réfère aux *capacités singulières* de l'ordinateur. Les deux premières fonctions pédagogiques – présenter de l'information et dispenser des exercices – ne les exploitent pas spécifiquement à part la capacité de stockage et d'adaptation de séquences. En outre, pourquoi utiliser l'ordinateur pour une fonction qui est mieux remplie par un autre support ? ; par exemple un manuel scolaire pour faire lire des informations ? A l'inverse, les autres fonctions sollicitent plus spécifiquement les capacités de l'ordinateur en créant des environnements qui ne sont pas réalisables autrement. Les capacités de calcul servent par exemple à ralentir, accélérer ou arrêter un processus complexe dans un but pédagogique. Une manière d'évaluer un logiciel éducatif procède donc en estimant la plus-value offerte par l'ordinateur par rapport à d'autres supports. Du côté de la recherche, cette voie d'évaluation s'est avérée être une mission impossible. Les situations de comparaison sont tellement divergentes qu'elles ne se prêtent pas à des conclusions viables. Du côté de la pratique, il peut néanmoins être intéressant de la mener au niveau conceptuel dans le but d'évaluer l'intérêt d'un investissement dans l'informatique par rapport à des moyens classiques.

#### **Adapter fonction et matière**

La deuxième voie focalise sur l'*adéquation* entre le type de logiciel (sa fonction) et la matière enseignée. Cette approche pragmatique, au lieu d'adhérer à une seule théorie d'apprentissage,

préconise d'adopter un point de vue en fonction des caractéristiques des connaissances en jeu. Par exemple, Woolf (interviewée par Sandberg et Barnard, 1993) aborde l'apprentissage des concepts de façon constructiviste, des savoir-faire avec la cognition située, et des classifications de manière béhavioriste. Cette approche a d'ailleurs été qualifiée d'*électrique* par Sandberg et Barnard. Elle mène à rechercher un type de logiciel en fonction de la matière : des tutoriels pour exposer des définitions de la géographie, des exercices répétées pour s'entraîner en mathématiques et des environnements collaboratifs pour discuter des causes de l'extinction des dinosaures par exemple. L'adéquation d'un type de logiciel pour enseigner une matière spécifique peut dans une certaine mesure être pressentie. Néanmoins, l'efficacité réelle mérite d'être vérifiée (voir par exemple Spiro, Feltovich, Jacobson, & Coulson, 1991, pour l'utilisation d'hypertexte pour des domaines complexes).

### **Evaluer les logiciels d'après leurs affordances**

L'usage effectif d'un logiciel sera largement déterminé par ses affordances. Le terme *affordance* réfère aux propriétés réelles et perçues qui déterminent comment un objet peut potentiellement être utilisé (Norman, 1988). Un objet bien conçu invite à son utilisation appropriée : une porte invite à l'ouvrir, une chaise à s'asseoir. Une affordance permet à l'utilisateur de se servir d'un objet sans explication. Ainsi, la troisième voie d'évaluation est de jauger les logiciels sur les affordances effectivement présentes. Bien que l'ordinateur ait des affordances potentielles de calcul et de visualisation, de manipulation de représentations externes et de collaboration (voir Pea, 1993), l'affordance dans un contexte donné dépendra des caractéristiques du logiciel et de la situation d'utilisation.

Si une affordance se réfère à la capacité d'inviter un comportement de la part de l'utilisateur, le comportement souhaité pour un logiciel éducatif est celui qui est considéré comme indispensable pour l'apprentissage (étant donné un point de vue théorique). Notre typologie a précisément permis de montrer qu'il existe de multiples comportements souhaités (voir Tableau 1). En conséquence, l'évaluation concerne à la fois le comportement suscité et l'apprentissage. Suthers (1999) par exemple propose différentes représentations graphiques d'un sujet pour susciter un débat entre élèves (approche de type cognition située, huitième fonction). La discussion entre élèves devrait les conduire à une meilleure compréhension du sujet (voir aussi nos propres recherches dans de Vries, Lund et Baker, à paraître). Lors d'une évaluation de l'environnement proposé, on pourrait alors observer plusieurs cas de figures. Dans le cas idéal, les élèves discutent *et* présentent une meilleure compréhension, mais que faudrait-il conclure si les élèves ont de vives discussions, mais ne présentent pas de meilleure compréhension des concepts discutés ? On pourrait aussi observer les élèves se rallier rapidement à une seule interprétation et ne pas présenter une meilleure compréhension. Dans le dernier cas, on questionnera la conception de l'environnement comme implémentation du point de vue théorique. La notion d'affordance conduit ainsi à rechercher la cause d'un mauvais résultat dans la conception plutôt que dans l'utilisation du logiciel. En effet, il n'est pas rare de devoir constater qu'un logiciel n'est pas utilisé de la façon prévue par les concepteurs. L'ultime test pour un logiciel éducatif serait alors d'établir dans quelle mesure il remplit sa fonction propre. Est-ce qu'un hypermédia – conçu comme espace d'exploration – invite vraiment à explorer, ou une simulation – conçu comme environnement de découverte – à manipuler, observer et interpréter ? Avant d'évaluer si un logiciel permet d'apprendre, il faudrait donc d'abord évaluer s'il remplit la fonction annoncée par ses concepteurs.

## **CONCLUSION**

Le domaine des nouvelles technologies éducatives est vaste, mais la diversité des logiciels d'apprentissage ne nous dispense pas de l'obligation de structurer, d'organiser et de présenter ce domaine d'une façon intelligible pour les sciences de l'éducation. Nous avons proposé le point de vue théorique, la tâche posée à l'élève et le point de vue sur les connaissances comme aspects caractérisant les différentes fonctions remplies par l'ordinateur. Il existe d'autres façons de décrire le domaine, par exemple autour de trois scénarios : celui de la transmission — un récipient vide se remplissant de

connaissances — celui de l'atelier — la connaissance construite par l'élève — et celui de la négociation — la connaissance construite par plusieurs (Sandberg, 1999). Ou encore selon ce que l'ordinateur devrait pouvoir faire — résoudre, expliquer, conseiller, représenter, simuler et organiser (Bruillard, 1997). Notre typologie a le mérite de classer selon des fonctions qui s'apparentent à des méthodes pédagogiques et d'explicitier les théories d'apprentissage sous-jacentes. Présentés ainsi, les logiciels éducatifs peuvent être vus, non plus comme une panoplie, mais comme un éventail de fonctions pédagogiques à notre disposition. Nous avons vu qu'il n'existe pas une méthode unique d'enseigner à l'aide de l'ordinateur, il s'agit donc de chercher des mariages entre méthode d'enseignement et type de logiciel. La typologie met également en cause la façon d'évaluer. Le paradigme pré-test - post-test d'apprentissage pourrait bien être adapté à une catégorie de logiciels mais pas à d'autres. L'explicitation des fonctions et des affordances visées conduira à définir de nouvelles mesures de succès pour évaluer l'efficacité des logiciels éducatifs. L'énonciation des points de vue théoriques sous-jacents permettra d'étudier et de préciser des relations entre ces points de vue, leurs implémentations logicielles, et les affordances, comportements et apprentissages réalisés.

### Remerciements

L'auteur remercie ses collègues du Laboratoire des Sciences de l'Éducation de l'Université Pierre-Mendès-France de Grenoble et en particulier Jacques Baillé, Philippe Dessus et Benoit Lemaire. L'article a largement bénéficié de nombreuses discussions avec eux ainsi que de leurs précieux commentaires sur des versions précédentes.

### BIBLIOGRAPHIE

- ALESSI, S. M. & TROLLIP, S. R. (1991). – **Computer-based instruction: Methods and development** (2nd edition). Englewood Cliffs: Prentice Hall.
- BAKER, M. J. & LUND, K. (1997). – Promoting reflective interactions in a CSCL environment. **Journal of Computer-Assisted Learning**, **13**, 175-193.
- BARON, G.-L. (1990). – L'informatique en éducation : le cas de la France. Note de synthèse. **Revue Française de Pédagogie**, **92**, 57-77.
- BARON, G.-L. & BRUILLARD, E. (1996). – **L'informatique et ses usagers dans l'éducation**. Paris : Presses Universitaires de France.
- BARON, G.-L. & BRUILLARD, E. (2001). – Une didactique de l'informatique ? **Revue Française de Pédagogie**, **135**, 163-172.
- BARON, M., GRAS, R., & NICAUD, J.-F. (1991). – Introduction. Dans M. Baron & J.-F. Nicaud (Eds.), **Deuxième Journées EIAO de Cachan** (pp. 7-8). Cachan : Editions ENS.
- BARWISE, J. & ETCHEMENDY, J. (1990). – **The language of first order logic**. Stanford: Center for the Study of Language and Information.
- BRUILLARD, E. (1997). – **Les machines à enseigner**. Paris: Hermès.
- BROWN, J. S., COLLINS, A., & DUGUID, P. (1989). – Situated cognition and the culture of learning. **Educational Researcher**, **18**, 32-42.
- BURTON, J. K., MOORE, D. M., & MAGLIARO, S. G. (1996). – Behaviorism and instructional technology. In D. H. Jonassen (Ed.), **Handbook of research for educational communications and technology** (pp. 46-73). New York: Macmillan.
- COLLIS, B. A. (1996). – Computers and education. In T. Plomp & D. P. Ely (Eds.), **International encyclopedia of educational technology** (2<sup>nd</sup> ed.) (pp. 402-408). Oxford, UK: Elsevier Science.
- DE JONG, T., & VAN JOOLINGEN, W. R. (1998). – Scientific discovery learning with computer simulations of conceptual domains. **Review of Educational Research**, **68**, 179-202.
- DEPOVER, C. (1987). – **L'ordinateur média d'enseignement. Un cadre conceptuel**. Bruxelles : De Boeck.

- DERO, M. (1996). – L'enseignement avec ordinateur. In A. Lieury (Ed.), **Manuel de psychologie de l'éducation et de la formation** (pp. 341-369). Paris : Dunod.
- DESSUS, P., LEMAIRE, B., BAILLE, J. (1997). – Études expérimentales sur l'enseignement à distance. **Sciences et Techniques Éducatives**, **4**, 137-164.
- DE VRIES, E., & DE JONG, T. (1999). – The design and evaluation of hypertext structures for supporting design problem solving. **Instructional Science**, **27**, 285-302.
- DE VRIES, E., LUND, K. & BAKER, M. (à paraître). – Computer-mediated epistemic dialogue: Explanation and argumentation as vehicles for understanding scientific notions. **Journal of the Learning Sciences**.
- DILLON, A. & GABBARD, R. (1998). – Hypermedia as an educational technology: A review of the quantitative research literature on learner comprehension, control and style. **Review of Educational Research**, **68**, 322-349.
- DUFFY, T. M., & JONASSEN, D. H. (1991). – Constructivism: New implications for instructional technology?, **Educational Technology**, May, pp. 7-12.
- DUFFY, T. M. & KNUTH, R. A. (1990). – Hypermedia and instruction: Where is the match? In D. H. Jonassen & H. Mandl (Eds.), **Designing hypermedia for learning** (pp. 199-226). Berlin Heidelberg: Springer.
- GABRIEL, P. (1998). – Enseignement et informatique pour tous : où en sommes-nous ? **Revue Française de Pédagogie**, **124**, 99-108.
- GAGNE, R. M. (1985). – **The conditions of learning and theory of instruction** (4<sup>th</sup> ed.). New York: Holt.
- LABORDE, J. M. (1995). – Des connaissances abstraites aux réalités artificielles, le concept de micro-monde Cabri. In D. Guin, J.-F. Nicaud, & D. Py (Eds.), **Environnements interactifs d'apprentissage avec ordinateur, tome 2** (pp. 29-41). Paris : Eyrolles.
- LAVE, J. & WENGER, E. (1991). – **Situated learning: Legitimate peripheral participation**. Cambridge: Cambridge University Press.
- LESGOLD, A., LAJOIE, S., BUNZO, M., & EGGAN, G. (1992). – A coached practice environment for an electronics troubleshooting job. In J. Larkin & R. Chabay (Eds.), **Computer assisted instruction and intelligent tutoring systems: establishing communications and collaboration** (pp. 201-238). Hillsdale, NJ: Lawrence Erlbaum.
- LINARD, M. (1996). – **Des machines et des hommes** (nouv. éd.). Paris : L'Harmattan.
- MAYER, R. E. (1987). – **Educational psychology**. Santa Barbara: Harper Collins.
- MERRILL, M. D. (1983). – Component display theory. In C. M. Reigeluth (Ed.), **Instructional design theories and models** (pp. 279-333). Hillsdale, NJ: Lawrence Erlbaum.
- NEUMAN, D., MARCHIONINI, G., & MORRELL, K. (1995). – Evaluating Perseus 1.0: methods and final results. **Journal of Educational Multimedia and Hypermedia**, **4**, 365-382.
- NORMAN, D. A. (1988). – **The psychology of everyday things**. New York : Basic Books.
- PAPERT, S. (1981). – **Jaillissement de l'esprit : ordinateurs et apprentissage**. Paris : Flammarion.
- PEA, R. D. (1993). – Practices of distributed intelligence and designs for education. In G. Salomon (Ed.), **Distributed cognitions: Psychological and educational considerations** (pp. 47-87). Cambridge, MA: Cambridge university press.
- RIEBER, L. P. (1996). – Seriously considering play: Designing interactive learning environments based on the blending of microworlds, simulations and games. **Educational Technology Research & Development**, **44**(2), 43-58.
- ROSHELLE, J. (1992). – Learning by collaborating: Convergent conceptual change. **Journal of the Learning Sciences**, **2**, 235-276.
- SAETTLER, P. E. (1990). – **The evolution of American educational technology**. Englewood, CO: Libraries Unlimited.
- SANDBERG, J. A. C. (1999). – Trends and issues in AI and Education: Towards a common research framework. Conférence invitée. 9<sup>e</sup> Conférence Internationale **Artificial Intelligence in Education**, Le Mans, France.

- SANDBERG, J. A. C. & BARNARD, Y. F. (1993). – Education and technology: What do we know? And where is AI? **AI Communications**, 6, 47-58.
- SCARDAMALIA, M. & BEREITER, C. (1994). – Computer support for knowledge-building communities. **The Journal of the Learning Sciences**, 3, 265-283.
- SPIRO, R. J., FELTOVICH, P. J., JACOBSON, M. J., & COULSON, R. L. (1991, May). – Cognitive flexibility, constructivism and hypertext: Random access instruction for advanced knowledge acquisition in ill-structured domains. **Educational Technology**, pp. 24-33.
- SPIRO, R. J., & JEHNG, J.-C. (1990). – Cognitive flexibility and hypertext: Theory and technology for the nonlinear and multidimensional traversal of complex subject matter. In D. Nix & R. J. Spiro (Eds.), **Cognition, education and multimedia: Exploring ideas in high technology** (pp. 163-205). Hillsdale, NJ: Lawrence Erlbaum.
- SUTHERS, D. D. (1999). – Representational bias as a guidance for learning interactions: A research agenda. In S. P. Lajoie & M. Vivet (Eds.), **Artificial Intelligence in Education. Open learning environments: New computational technologies to support learning, exploration and collaboration** (pp. 121-128). Amsterdam: IOS Press.
- SUTHERS, D. & WEINER, A. (1995). – Groupware for developing critical discussion skills. In J. L. Schnase & E. L. Cunnius (Eds.), **Proceedings of Computer Supported Cooperative Learning**. Bloomington, Indiana.
- TERGAN, S.-O. (1997). – Multiple views, contexts, and symbol systems in learning with hypertext/hypermedia: a critical review of research. **Educational Technology**, 27(4), 5-18.
- VAN OOSTENDORP, H. & HOFMAN, R. (1998). – L'effet cognitif de la carte de contenus d'un hypertexte. In J.-F. Rouet et B. de la Passardière (Eds.), **Actes du 4<sup>e</sup> colloque Hypermédias et Apprentissages** (pp. 173-186). Paris : INRP, EPI.
- VIVET, M. (1997). – Préface. Dans E. Bruillard, **Les machines à enseigner** (pp. 11-13). Paris: Hermès.
- WENGER, E. (1987). – **Artificial Intelligence and Tutoring Systems**. Los Altos, CA: Morgan Kaufmann.
- WHITE, B. Y. & FREDERIKSEN, J. R. (1989). – Causal models as intelligent learning environments for science and engineering education. **Applied Artificial Intelligence**, 3(2-3), 83-106.