

INTÉGRATION D'UN HYPERTEXTE DANS UN ENVIRONNEMENT D'APPRENTISSAGE À INITIATIVE MIXTE

**Daniel Schneider, Boris Borcic, Pierre Dillenbourg,
Mélanie Hilario, Patrick Mendelsohn**

TECFA, Faculté de Psychologie et des Sciences de l'Education
Université de Genève
9, Route de Drize, 1227 Carouge, Suisse.

***Résumé :** Nous discutons l'intégration d'un hypertexte dans un environnement d'apprentissage complexe et nous présentons brièvement l'hypertexte que nous avons développé dans le cadre d'ETOILE (Experimental Toolbox for Interactive Learning Environments). Notre ambition en matière de logiciels éducatifs vise à élaborer des environnements riches, au sein desquels plusieurs conceptions de l'apprentissage seraient non seulement juxtaposées mais intégrées. L'hypertexte possède un rôle multiple dans des environnements d'apprentissage intégrant les deux pôles reconnus du logiciel éducatif : l'approche micro-monde et l'approche tutorielle. Il peut servir de pont entre des styles d'apprentissage différents au sein d'un même environnement.*

INTRODUCTION

La conception de logiciels éducatifs a pris une tournure importante cette dernière décennie à la suite d'avances techniques et des changements intervenus dans les théories d'apprentissage et d'enseignement. Le TECFA depuis trois ans s'est lancé dans l'élaboration d'environnements d'apprentissage avancés. Un tel environnement se caractérise par des éléments tels que :

- la multiplicité de styles d'enseignements (la même matière peut être enseignée de façon diverse, et pour certaines matières, il existe des styles préférables) ;
- la multiplicité de sources d'apprentissage (expériences, "coaching", exploration d'hypertextes, etc.) ;
- l'utilisation d'interfaces riches qui autorisent des manipulations directes d'objets ;
- l'utilisation de techniques d'intelligence artificielle pour l'intégration d'experts et de tuteurs qui aident à résoudre des problèmes ou qui permettent d'intégrer des raisonnements pédagogiques.

On peut distinguer deux approches fondamentales du logiciel éducatif :

- l'approche **micro-monde**, basée sur l'exploration par l'apprenant d'une situation-problème bien définie ;
- l'approche **tutorielle**, basée sur l'implémentation d'agents dotés, soit d'une expertise quant au contenu à enseigner, soit d'une expertise pédagogique.

Ces deux approches reposent sur différentes théories de l'apprentissage, le constructivisme pour l'approche micro-monde, le comportementalisme 'évolué' ou l'approche socioculturelle pour le mode tutoriel. Nous considérons que ces théories ne s'excluent pas mutuellement, mais éclairent chacune le processus complexe d'apprentissage sous un angle particulier. Cette contribution suggère que l'utilisation d'un hypertexte dans un système favorise l'articulation des approches micro-monde et tutorielle. Nous décrivons l'intégration de l'hypertexte à deux niveaux :

- **intégration dans l'organisation pédagogique** d'un environnement d'apprentissage à initiative mixte ;
- **intégration dans l'architecture informatique** d'un environnement d'apprentissage orienté-objet.

Nos réflexions sont illustrées par les deux couches de logiciel que nous avons développées dans notre groupe :

- MEMOLAB, un environnement permettant aux futurs psychologues d'acquérir des compétences de base en matière de construction de plans d'expérience ;
- ETOILE ("An Experimental Toolbox for creating Interactive Learning Environments"), une boîte à outils permettant de créer des systèmes semblables à MEMOLAB pour d'autres contenus d'apprentissage.

ETOILE consiste en plusieurs modules, dont un moteur d'inférence orienté-objet, un formalisme pour exprimer un curriculum, un ensemble de tuteurs, un coach (l'entraîneur qui gère l'intervention des tuteurs), des outils d'interface, des outils de gestion de fenêtres d'application, un hypertexte, etc. Une description complète de ces systèmes n'est pas pertinente ici. Un auteur doit, pour créer un système, d'abord préciser un curriculum en forme de réseau de buts à atteindre. Il doit ensuite créer des experts du domaine capables de résoudre des problèmes associés à un but pédagogique et créer les micro-mondes (l'interface). Notre boîte à outils s'adresse avant tout à des chercheurs ayant des connaissances de programmation en Lisp. Le lecteur trouvera une description plus détaillée d'ETOILE et de MEMOLAB Dillenbourg et al. (1992).

2. INTÉGRATION PÉDAGOGIQUE

Le rôle d'un hypertexte se conçoit différemment dans l'approche micro-monde et dans l'approche tutorielle. Dans l'approche micro-monde, l'hypertexte est exploré librement par l'apprenant, alors qu'en mode tutoriel, c'est le tuteur qui sélectionne les noeuds de l'hypertexte que l'apprenant devra lire. Un environnement à initiative mixte se définit précisément par la symétrie (ou la quasi-symétrie) des fonctionnalités offertes à l'apprenant et aux agents informatisés qui interagissent avec cet apprenant (tuteurs, experts...). Les tuteurs et l'apprenant disposent du même langage de commandes pour manipuler l'hypertexte. Plus précisément, ils disposent d'un langage fonctionnellement équivalent mais syntaxiquement différent : une

même commande sera une expression LISP pour le tuteur et un élément de l'interface (bouton ou menu) pour l'apprenant. En ce qui concerne nos systèmes, ce langage comporte les commandes usuelles : sélection d'un noeud ou d'une note (la seconde évitant un changement de page) dans le texte, sélection d'un noeud dans la hiérarchie des noeuds super-ordonnés ou dans l'historique des noeuds rencontrés, retour au noeud précédent, recherche systématique d'une information...

Par sa conception même, l'hypertexte s'intègre facilement à l'approche micro-monde : son principe de base est l'exploration d'un monde fini (l'hypertexte) au moyen d'un langage de commandes succinctes. Un hypertexte n'est en réalité qu'un exemple particulier de micro-monde. L'exploration repose entièrement sur les initiatives de l'apprenant, mais ne prévoit pas les initiatives du tuteur. La difficulté que nous avons dû surmonter concerne surtout l'intégration de l'hypertexte au fonctionnement des tuteurs. Plus précisément, cette difficulté consiste à déterminer **à quel moment le tuteur décidera d'ouvrir l'hypertexte pour l'élève et quelle partie de cet hypertexte demandera-t-il à l'apprenant de lire**. En outre, si le système comporte plusieurs hypertextes, comme c'est le cas dans MEMOLAB, le tuteur devra choisir l'hypertexte qu'il convient d'ouvrir. Pour répondre à ces questions, il convient de décrire brièvement le fonctionnement de nos tuteurs.

ETOILE comporte cinq tuteurs qui diffèrent par leur style d'enseignement. Ces tuteurs répondent différemment aux questions que nous venons de poser, un tuteur directif aura, par exemple, tendance à proposer à l'apprenant de lire l'hypertexte avant de s'exercer dans le micro-monde. 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. Chaque tuteur est en fait une base de règles qui détermine et supervise l'interaction entre l'apprenant et un expert du domaine. Un expert du domaine est une base de règles capable de résoudre les problèmes soumis à l'apprenant. Seul cet expert dispose de connaissances spécifiques au contenu de l'enseignement. Cette architecture nous a permis de doter les tuteurs de règles pédagogiques indépendantes du contenu de l'enseignement.

De façon analogue, le tuteur n'a pas de connaissances sur l'hypertexte qu'il va manipuler, c'est-à-dire qu'il ignore le contenu des noeuds de l'hypertexte. **Le tuteur manipule des relations entre diverses représentations du contenu à enseigner**, et cela à deux niveaux de granularité :

- **chaque objectif** composant le curriculum du système est associé à un **noeud élevé** de l'hypertexte ;
- **certaines règles** de l'expert sont associées à des noeuds de l'hypertexte qui décrivent de façon plus lisible la micro-compétence encodée dans la règle.

La façon dont un tuteur exploite un lien curriculum-hypertexte ou un lien expertise-hypertexte dépend de son style d'enseignement. Nous analysons deux exemples. Dans un premier exemple, le coach sélectionne un tuteur et lui assigne l'objectif pédagogique X. Le tuteur dispose de l'information selon laquelle la théorie liée à l'objectif X se trouve dans le noeud Y de l'hypertexte Z. Une des caractéristiques d'un style d'enseignement réside dans l'exploitation de ce lien. Les deux règles

pédagogiques ci-dessous expriment respectivement une approche déductive et inductive de l'apprentissage :

Si l'apprenant n'a encore réalisé aucune activité pour l'objectif X,
et que l'objectif X est liée au noeud Y de l'hypertexte Z
Alors ouvrir l'hypertexte Z au noeud Y

Si l'apprenant a réalisé avec succès 5 problèmes propres à l'objectif X,
et que l'objectif X est liée au noeud Y de l'hypertexte Z
Alors ouvrir l'hypertexte Z au noeud Y

Considérons un second exemple. L'apprenant et l'expert résolvent conjointement un problème, pas à pas. Le tuteur supervise leurs interactions. L'apprenant fait un pas que l'expert considère comme erroné. Par exemple, dans MEMOLAB, il crée deux groupes expérimentaux dont les effectifs diffèrent largement. L'expert dispose de règles de réparation qui lui permettent de corriger un état intermédiaire erroné. L'expert met une telle règle en oeuvre, non afin d'enseigner quelque chose à l'élève, mais dans le but "égocentrique" de résoudre le problème. Par contre, le tuteur qui décèle cette interaction, possède des intentions didactiques en fonction desquelles il interprétera l'interaction. Il utilisera pour cela le lien entre la règle de réparation et un noeud de l'hypertexte. Cette exploitation se fera de façon diverse selon le style didactique du tuteur. Voici deux règles pédagogiques qui illustrent des styles différents, la première conduisant à une intervention systématique, la seconde limitant les interventions aux apprenants qui ont connu précédemment d'autres difficultés :

Si l'expert met en oeuvre la règle de réparation R,
et que la règle R est liée au noeud Y de l'hypertexte Z,
Alors ouvrir l'hypertexte Z au noeud Y.

Si l'expert met en oeuvre la règle de réparation R,
et qu'auparavant, l'apprenant a échoué à plusieurs problèmes,
et que la règle R est liée au noeud Y de l'hypertexte Z,
Alors ouvrir l'hypertexte Z au noeud Y.

Dans l'approche que nous avons décrite, le tuteur décide éventuellement d'ouvrir un hypertexte à un endroit particulier. A ce moment, il cède le contrôle à l'apprenant qui explore librement ce texte. En réalité, nous avons intégré l'hypertexte aux fonctionnalités du tuteur mais nous n'avons pas tenté de réaliser la réciproque : le tuteur ignore ce que l'apprenant fait dans l'hypertexte. Notre implémentation nous permet d'envisager un suivi par le tuteur de l'itinéraire de l'apprenant dans l'hypertexte. Il est envisageable de créer des règles pédagogiques qui analysent cet itinéraire, par exemple pour guider l'apprenant. Nous n'avons pas implémenté ce type d'expertise pédagogique, mais d'autres auteurs se sont penchés sur des aides intelligentes à l'exploration d'un hypertexte. (Cf. Woodhead 1991 : 35-56 pour un survol d'hypermédias "intelligents".)

3. LE SYSTÈME HYPERTEXTE D'ETOILE

L'hypertexte de notre boîte à outils "ETOILE" devait être convivial pour l'utilisateur, mais flexible et ouvert pour le programmeur. Pour remplir ces critères nous nous sommes inspirés du "Document Examiner" du système d'exploitation Genera de Symbolics (cf. Walker 1987, Nielsen 90 : 38-39). Ce système a été un des premiers hypertextes utilisés réellement et fréquemment. En 1985 déjà, la majorité des utilisateurs des stations Symbolics ont préféré en règle générale le "document examiner" (constituant plus de 10 000 noeuds et de 23 000 liens) à la documentation écrite. Les principes de design étaient (a) de garder l'interface simple et (b) de maintenir une double structure, linéaire (comme une suite de livres ayant de tables de matières bien structurées) et en forme de réseau liant les noeuds de façon thématique à travers les sections. Un autre principe était de permettre l'accès à certaines informations (noeuds) du "Document Examiner" en dehors du programme de lecture proprement dit. Par exemple, on peut afficher une définition technique (la description d'une fonction lisp par exemple) à l'endroit où on en a besoin (l'éditeur de programmes) sans aller dans la fenêtre hypertexte. Notre système implémente un sous-ensemble très réduit des fonctionnalités du "Document Examiner", mais qui suffit momentanément à nos besoins.

Bush (1945) avait formulé dans les années trente le grand principe des hypertextes : la sélection par association. Ce n'est que depuis très peu de temps, que l'on trouve sur le marché certains outils (comme Guide ou Storyspace) qui marient l'idée du texte "associatif" à la gestion de longs textes. Ces logiciels n'étaient pas suffisamment "ouverts" pour nos besoins, ce qui nous a amené à programmer notre propre logiciel.

Le fait que nos hypertextes disposent d'une taxonomie de noeuds semblables aux textes linéaires (parties, chapitres, sections...), autorise une lecture linéaire du texte pour les apprenants non familiers avec des hypertextes. Cette linéarisation à option de l'hypertexte permet en outre d'imprimer ceux-ci, ce qui est un avantage non-négligeable. Selon ses besoins, l'utilisateur d'un didacticiel pourra utiliser la version écrite, plus facile à gérer, ou la version interactive, qui elle est plus puissante en raison des liens d'hypertexte et de son intégration avec le reste du système. Pour faciliter la navigation, l'utilisateur peut afficher un diagramme arborescent qui montre la hiérarchie des noeuds de section d'un "livre" ou encore du chapitre courant. En cliquant sur un noeud du graphe, il peut se déplacer facilement. Un graphe similaire pour la vision en réseau est en développement. Dans deux panneaux ("panes") sont affichés les noeuds du parcours effectué ("history") et le chemin pour se retrouver au départ ("context"). Ces aides à la navigation constituent le minimum nécessaire pour se retrouver dans un hypertexte de taille modérée. Le réseau est très puissant pour faciliter des activités d'exploration, par contre la structure hiérarchique semble être plus indiquée pour la présentation systématique d'un ensemble de connaissances. Comme le montre la discussion par Woodhead (91 : 117-133) de la rhétorique des "gotos", hiérarchies, réseaux, boucles, etc., il est clair qu'un bon système doit à la fois assurer l'intégration de noeuds ("chunking") et permettre leur exploration par tous les chemins raisonnables. Autrement dit, dans notre système, nous avons opté pour une organisation "classique" de l'information en formes de "livres" pour éviter que l'apprenant ne se sente perdu et pour favoriser

une exploration systématique des chapitres, mais nous avons greffé dessus une structure de réseau pour l'accès à toutes les informations associées. Il est également possible de relier entre eux plusieurs hypertextes. Ceci nous permet par exemple de faire lire simultanément à un apprenant un texte sur un paradigme de la mémoire et un autre sur la méthodologie de l'expérience en question.

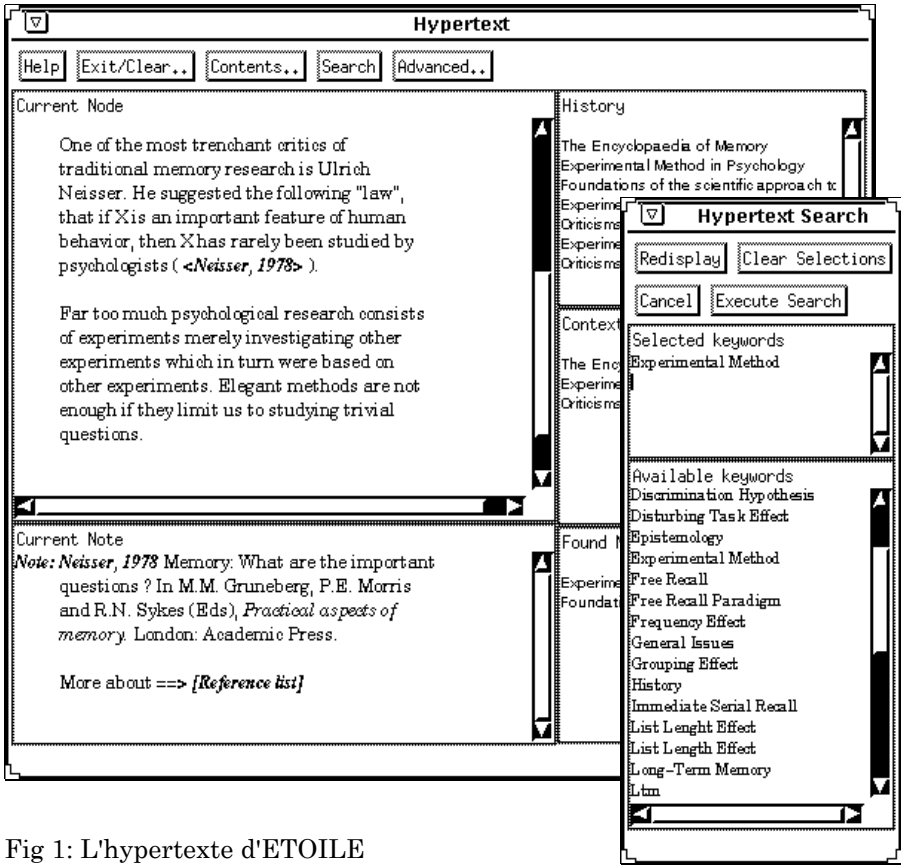


Fig 1: L'hypertexte d'ETOILE

Nous présentons dans la figure 1 une version "Motif" du lecteur hypertexte d'ETOILE. L'utilisateur peut faire varier la taille de ces fenêtres selon ses besoins. Un hypertexte dans ETOILE est rédigé avec un éditeur de textes quelconque et au moyen d'un langage de spécifications de mise en forme. Il s'agit d'un petit sous-ensemble de LATEX comprenant entre autres certaines des commandes de section et d'environnements d'affichage (paragraphes, listes, code, etc.). A cela s'ajoute un petit jeu de commandes qui concerne l'hypertexte, comprenant le saut, le noeud standard et la note qui est affichée dans une petite fenêtre en bas de l'écran. Notez que toutes les commandes de section (chapitre, section, sous-section, etc.) font aussi office de noeud hypertexte. Enfin, le système comprend des commandes qui permettent de piloter les autres parties du système (voir la section suivante).

4. INTÉGRATION INFORMATIQUE

Les systèmes MEMOLAB et ETOILE sont implémentés en Common Lisp orienté-objet (CLOS). Les hypertextes et le moteur d'inférence partagent ce mode de représentation. Les hypertextes, les noeuds de l'hypertexte, les agents, les règles, les objectifs et les problèmes sont exprimés dans ce même formalisme, ce qui facilite l'intégration. Les règles manipulent n'importe quel objet, qu'il s'agisse d'un objet du domaine, d'un noeud de l'hypertexte, voire d'une autre règle. A cet effet, nous avons développé un moteur d'inférence particulier dont la base de faits est constituée des objets du système. L'auteur définit les classes d'objets qui doivent être répertoriés en vue d'instancier des règles.

Un hypertexte peut être "piloté" par d'autres modules du système. Ainsi, il est possible de faire apparaître un ou plusieurs "lecteurs" hypertexte, d'y charger un livre, d'afficher un noeud précis dans un "lecteur", etc. Le moteur d'inférence d'ETOILE autorise les appels procéduraux. Ceci nous permet par exemple d'inclure dans les conclusions d'une règle pédagogique la procédure qui ouvre un hypertexte donné à un noeud spécifié. Il est aussi possible d'afficher un bouton de "saut" hypertexte ailleurs dans l'interface, qui permet d'ouvrir l'hypertexte et le noeud en question sur l'action d'un click. Dans MEMOLAB, nous utilisons ce mécanisme lors de l'affichage des résultats de la simulation d'une expérience. Le module de simulation, qui fonctionne selon les principes du "case based reasoning", affiche les boutons des noeuds d'hypertexte discutant des expériences similaires, que l'apprenant peut consulter.

Réciproquement, un bouton de l'hypertexte peut être associé à une fonction, exécutée quand l'apprenant 'presse' sur ce bouton. Cette technique permet d'intégrer dans un hypertexte des exemples exécutables. MEMOLAB par exemple comprend un hypertexte dénommé 'Encyclopaedia of Memory'. L'apprenant y explore la littérature relative à l'expérimentation sur la mémoire humaine. Ces expériences sont décrites en langue naturelle, mais l'apprenant peut en demander la simulation dans le laboratoire artificiel où il crée ses propres expériences. Il peut de cette façon mettre en relation son propre travail et celui réalisé par des scientifiques reconnus. Ces boutons d'exécution ne font rien d'autre que d'exécuter une procédure Lisp, en lui transmettant éventuellement des arguments (comme l'identificateur de la fenêtre d'affichage). Lisp joue ainsi dans nos hypertextes un rôle analogue à Hypertalk dans Hypercard, ce qui marie simplicité (des commandes de base de l'hypertexte) et puissance (l'accès à tous les outils externes). Comme dans Hypercard, le développeur peut vérifier son application de façon incrémentale : une nécessité pour un travail de prototypage comme le nôtre.

CONCLUSION

Par sa conception même, un hypertexte s'intègre naturellement dans une approche micro-monde. En l'intégrant en outre dans la partie tutorielle de notre système, l'hypertexte constitue un pont entre ces deux approches, il favorise l'intégration d'agents (tuteurs et experts) au sein d'un environnement exploratoire. Cette intégration est facilitée par la définition d'un mode de représentation unique (objets) pour l'ensemble des composantes du système.

RÉFÉRENCES

- BUSH, V. [1945] "As we may think". *Atlantic Monthly* 176,1 (July 1945), p. 101-108.
- DILLENBOURG, P. , HILARIO, M., MENDELSON, P., SCHNEIDER D. and BORCIC, B. [1992] *The Memolab Project. Intermediate Report*. TECFA Document. TECFA, University of Geneva.
- ENGELBART, D.C., and ENGLISH, W.K., "A research center for augmenting human intellect", *Proceedings of the 1968 Fall Joint Conference, AFIS Press*, 395-410
- Genera User's Guide* [1990], Symbolics Inc., Burlington (MA), p.199-217.
- LAMPOR, Leslie [1986], *LATEX, User's Guide and Reference Manual*, Reading : Addison-Wesley.
- MARCHIONINI, Gary and SHNEIDERMAN, Ben [1990] "Finding Facts vs. Browsing Knowledge in Hypertext Systems", in BOYD-BARRET,O and SCANLON,E., *Computers and Learning*, Reading : Addison-Wesley.
- NIELSEN, Jakob [1990], *Hypertext and Hypermedia*, Boston : Academic Press.
- SEYER, P. [1991] *Hypertext : Concepts and Applications*. Blue Ridge Summit : Windcrest Books.
- WALKER, J.H. [1987] "Document Examiner : Delivery interface for hypertext documents", Proc. *ACM Hypertext '87 Conf.* (Chapel Hill, NC, 13-15 Nov.), 307-323.
- WOODHEAD, Nigel [1990] *Hypertext and Hypermedia, Theory and Applications*, Wilmslow : Sigma Press.