HAL
open science

# Une architecture pour la surveillance et le guidage de l'apprenant dans un hypermédia éducatif

Thierry Beltran

## To cite this version:

Thierry Beltran. Une architecture pour la surveillance et le guidage de l'apprenant dans un hypermédia éducatif. Premier colloque Hypermédias et Apprentissages, Sep 1991, Châtenay-Malabry, France. pp.95-110. edutice-00000783

**HAL Id: edutice-00000783**

**https://edutice.hal.science/edutice-00000783**

Submitted on 17 Feb 2005

**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.

# NE ARCHITECTURE POUR LA SURVEILLANCE ET LE GUIDAGE DE L'APPRENANT DANS UN HYPERMÉDIA ÉDUCATIF

**Thierry Beltran**
Laboratoire A.P.I. (Apprentissage, Pédagogie, Intelligence artificielle)
Université Paul Sabatier - IUT A
50, chemin des Maraîchers
31 062 Toulouse

*Dans cet article, nous proposons une architecture pour le développement de systèmes hypermédias éducatifs. La définition* ***d'entités de dialogue*** *au niveau de l'interface va permettre d'uniformiser et de surveiller l'interaction avec l'apprenant ; d'autre part, l'introduction de* ***nouveaux types de liens*** *pour l'hypermédia éducatif va permettre la mise en place du guidage de l'apprenant. L'architecture que nous proposons permet à un hypermédia éducatif d'agir tantôt comme un système support de formation (consultation...) tantôt comme un système d'enseignement (exercices, cours guidé...). En d'autres termes, notre objectif a été le développement d'une architecture basée sur l'hypermédia, autorisant le partage de l'initiative entre l'apprenant et le système d'EAO.*

## 1. Introduction

Les systèmes hypermédias sont intéressants pour l'EAO (Enseignement Assisté par Ordinateur) car ils apportent naturellement un certain nombre de fonctions qui font défaut à la majorité des logiciels éducatifs (grande interactivité, exploitation de bases de données multimédia volumineuses...). De plus, avec ces systèmes, la séparation traditionnelle entre l'outil de l'auteur et le "produit éducatif" peut être effacée. Pour toutes ces raisons, les systèmes hypermédias sont de plus en plus utilisés comme des outils auteur pour l'enseignement (prototypage et réalisation d'un cours). Mais, bien que les systèmes hypermédias soient potentiellement très prometteurs pour l'enseignement, leur utilisation pose de nouveaux problèmes et leur efficacité n'est pas encore établie pour certaines utilisations. Alors que les systèmes d'EAO sont basés sur le guidage de l'apprenant (*contrôle didactique*), dans les hypermédias la prise d'initiative de l'usager est la principale stratégie. Nous pouvons donc penser que les systèmes hypermédias actuels doivent être vus comme des supports de formation et non pas comme des logiciels d'enseignement au sens de l'EAO.

Nous remarquons les deux points suivants : premièrement, l'EAO/l'EIAO a développé des outils satisfaisants du point de vue des auteurs (analyseurs de réponses, modules de diagnostic, modules experts, outils de prototypage...), mais aboutissant

souvent à des logiciels éducatifs trop directifs et ennuyeux du point de vue de l'apprenant ; deuxièmement, d'une façon complètement opposée, les systèmes hypermédias sont agréables à utiliser pour l'apprenant, mais leur liberté de navigation les rend incompatibles avec de nombreuses approches pédagogiques.

A partir de ces remarques, nous proposons *une architecture pour un système hybride*, dans lequel l'exploitation de l'information hypermédia peut être combinée à des formes plus directives, dans le but de guider et contrôler l'apprentissage de certains points. Notre objectif est donc de fournir une architecture qui puisse couvrir une grande variété de logiciels éducatifs (du plus directif d'entre eux à l'environnement de découverte) en permettant, le plus possible, une manipulation "hypermédia" des informations.

## 2. Le cadre de recherche

Nos travaux ont été réalisés dans le cadre du projet *Atelier de Génie Didacticiel Intégré* ou "AGDI" (Beltran 89), (Bessagnet 90a, 90b), (Gouardères 90). Le domaine de recherche est l'amélioration de la production des logiciels éducatifs à l'aide des méthodes et outils du génie logiciel.

Pour améliorer la production de logiciels éducatifs évolués, les informaticiens doivent fournir aux auteurs des environnements de développement qui puissent évoluer en fonction des *besoins pédagogiques* et des *technologies éducatives*. Le projet AGDI est donc concerné par plusieurs thèmes de recherche : la contribution des techniques et outils du génie logiciel à la mise au point d'une méthodologie de conception des logiciels éducatifs, la réalisation de modules d'EIAO réutilisables... Nos travaux portent plus particulièrement sur l'utilisation de l'hypermédia dans différents types de logiciels éducatifs (didacticiels classiques, systèmes Tuteurs Intelligents, systèmes coach...). Nous pensons que le partage du contrôle entre un système hypermédia (contrôle de l'usager) et des modules d'EAO (contrôle du système) pourrait être une réponse aux problèmes précédemment soulignés (principalement : la liberté de l'usager dans un système hypermédia, les limites des logiciels éducatifs, les réalisations ad hoc...).

## 3. Le besoin de contrôle dans les environnements hypermédias éducatifs

Il n'est pas du tout évident que le fait de naviguer dans les nœuds d'un hypermédia puisse garantir un apprentissage quelconque. La flexibilité offerte par une organisation hypertexte des informations ne devrait pas se résumer à des cheminements incontrôlés parmi les nœuds de grandes bases de données. Quelques systèmes d'EAO, inspirés de l'hypertexte révèlent le besoin de contrôle, à des instants particuliers du processus d"apprentissage.

❐ Dans StrathTutor (Mayes 88, Mayes 89), les liens sont calculés à partir d'un ensemble d'attributs prédéfinis par l'auteur.

❐ HitchHicker's Guide (Allinson 88, Allinson 89) est basé sur la mise à disposition d'outils de guidage spécifiques. Les fonctionnalités de l'hypertexte sont ainsi étendues pour aider l'usager à explorer conceptuellement le domaine (parcours guidés, index, questions...).

❐ (Storrs 88) propose un modèle pour des logiciels éducatifs basés sur l'hypertexte. Réalisé à l'aide d'HyperCard, le logiciel se compose de deux piles : la première contient les informations à apprendre et peut être consultée de manière classique ; la deuxième contient les concepts traités ainsi que les relations qu'ils entretiennent.

❐ Le système SEVE (Ferret 87) est construit en Prolog avec un ensemble de fenêtres permettant une interaction orientée hypertexte. Ce système permet à l'auteur de combiner des séquences guidées et des séquences de découverte. L'utilisation de l'hypertexte dans SEVE est proche d'un système d'aide.

Ces réalisations constituent un premier pas vers la définition de logiciels éducatifs hypermédias. Nous pensons en effet que ce type de système devrait permettre des combinaisons dynamiques d'informations en regard de modèles didactiques, de profils d'élèves ou de modèles du domaine traité, guidant plus ou moins l'apprenant dans la base de données. Cette idée est de plus en plus exprimée lorsque l'utilisation des systèmes hypermédias est envisagée pour l'enseignement (Jonassen 90), (Mühlhauser 90), (Beltran 91). Dans la suite, nous proposons une architecture, inspirée du modèle PAC (Coutaz 88), pour la réalisation de systèmes hypermédias éducatifs.

## 4. Une architecture pour des systèmes hypermédias éducatifs

Trois systèmes composent l'environnement étudié : un *système hypermédia* dédié à la présentation des connaissances et la recherche d'information, un *système d'EAO / EIAO* qui assure le contrôle pédagogique et le séquencement des nœuds, un *système de surveillance* qui joue le rôle d'interface entre les deux systèmes précédents (Cf. Figure 1).

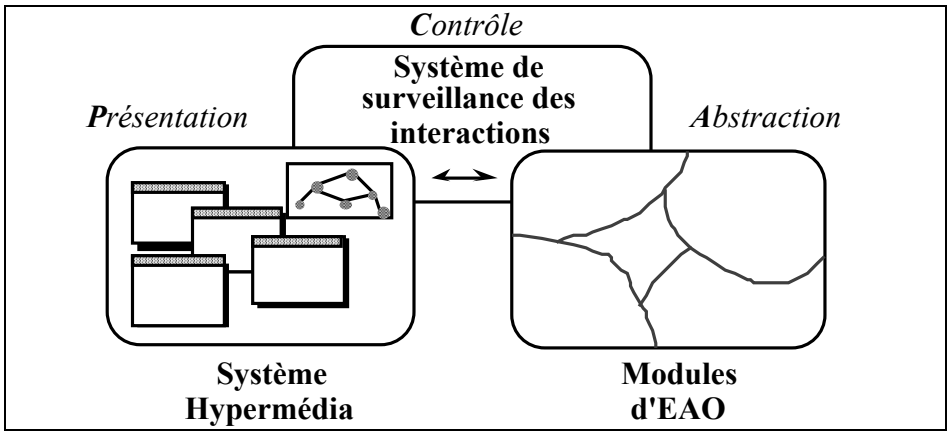



Figure 1 : L'architecture générale pour un système hypermédia éducatif

### *4.1. Les avantages de l'architecture proposée*

Cette architecture est indépendante de la structure interne des modules d'EAO (analyseurs de réponses, représentation des profils d'apprenant...) ; la structure interne du logiciel éducatif peut être simple (didacticiel "classique") ou complexe (tuteur intelligent), de ce fait les fonctionnalités du système global ne sont pas prédéfinies et peuvent être adaptées aux besoins des auteurs ou aux situations d'apprentissage envisagées.

Le second avantage de cette architecture est la réutilisation (par l'auteur) de modules d'EAO ou de parties d'hyperdocument d'une réalisation à l'autre : les modules d'EAO communiquent uniquement avec le système de surveillance et ne reçoivent que des informations abstraites (réponses, comportements, erreurs de manipulation...), ils ne sont pas concernés par l'interaction de bas niveau et la gestion des éléments de l'interface (défilement de texte, sélection de boutons ou menus...). En effet, les liens pédagogiques ne sont pas définis au niveau de l'interface : ils sont gérés et activés par le système de surveillance (*chemins prédéfinis*) ou par les modules d'EAO (*chemins dynamiques*).

De plus, cette séparation entre les modèles didactique et médiatique est intéressante pour l'auteur lors des étapes de conception et prototypage du logiciel éducatif hypermédia : il est possible d'étudier séparément les aspects médiatiques du cours et le comportement pédagogique du système.

### *4.2. Le mécanisme de surveillance*

Pour l'implémentation de notre architecture, nous avons choisi un mécanisme basé sur les événements (Cf. Figure 2). Alors que l"interaction de bas niveau (souris, clavier...) et les liens classiques sont gérés par le système hypermédia, des événements

d'un niveau d'abstraction plus élevé (réponses, erreurs, activation de liens significatifs...) sont envoyés au système de surveillance qui contient une base de règles associée à l'**Unité d'Apprentissage** en cours. Cette base de règles contient les chemins prédéfinis (règles de suivi) et des règles de "Breakpoints" chargés d'activer les modules d'EAO pour diagnostiquer et réagir à certaines interactions en décidant, par exemple une nouvelle orientation du cours.

La figure suivante montre comment le mécanisme par événements permet au système de surveillance de contrôler les actions de l'usager et de ne traiter que l'interaction cohérente d''un point de vue pédagogique.

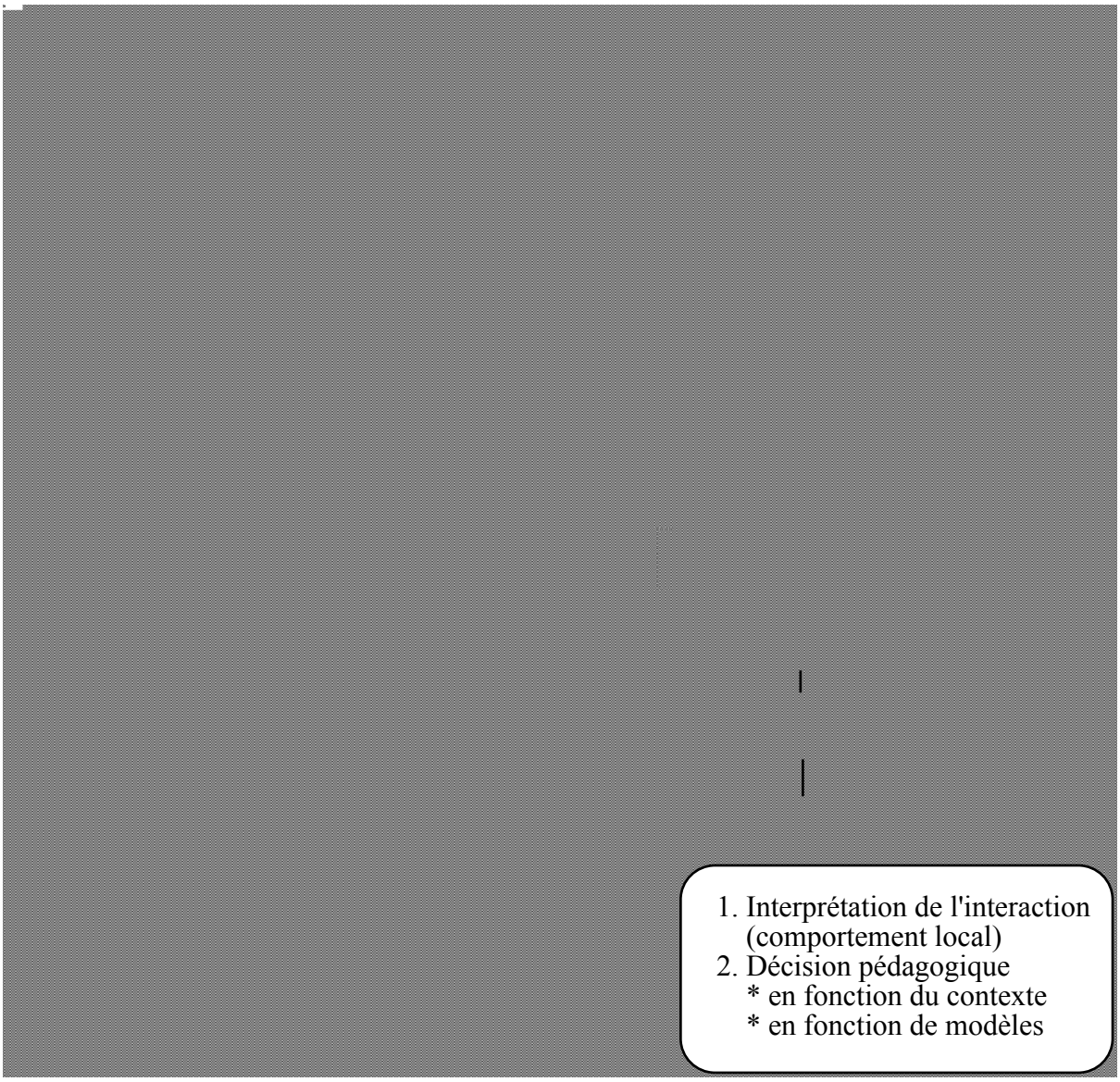



Figure 2 : La surveillance de l'interaction.

## 5. L'hyperdocument "EAO"

La spécialisation de l'interaction hypermédia à des fins d'EAO, nous conduit à introduire la notion d'**hyperdocument EAO**. Pour que l'apprenant et le système puissent prendre des initiatives dans la conduite de la session, nous avons créé de *nouveaux types de liens.* Ces liens ne sont pas directement "fixés" dans l'interface et n'activent pas directement un nœud donné. Certains objets de l'interface (**les entités de dialogue**, Cf. 7.2) sont chargés d'émettre des événements caractérisant l'interaction. A partir de ces événements (indépendants de l'interface), les règles de suivi ou de breakpoint peuvent être déclenchées. Si une règle de suivi est activée par les événements reçus, un nouveau nœud est présenté à l'apprenant (chemin prédéfini). Si une règle de "breakpoint" est activée, les modules d'EAO sont sollicités pour interpréter l'interaction, dans ce cas une nouvelle orientation peut être décidée. Si aucune règle n'est activée, le système se met en attente d'autres événements et le nœud courant n'est pas changé.

### *5.1. Les nouveaux types de liens*

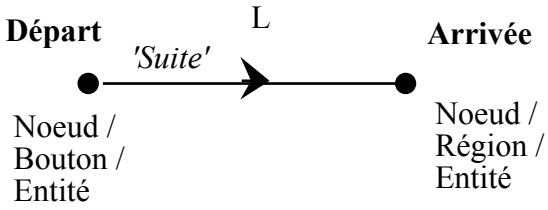
Les liens, comme dans tout hypergraphe, représentent des connections entre deux nœuds (liens de référence) ou groupes de nœuds (liens hiérarchiques). Dans un hypergraphe "classique", les liens représentés sont utilisés uniquement par l'usager, c'est lui qui pilote le système. Pour l'hypergraphe "EAO", nous allons définir d'autres liens, *contrôlés par le système* (liens de suite pédagogique, liens de référence "conditionnels") ou *activables par le système* (liens de réponse, liens de déroutement), de manière à permettre la gestion de l'apprentissage, c'est-à-dire la canalisation de l'apprenant vers une partie du cours déterminée par le tuteur.

Une caractéristique commune des deux premiers types de liens est que leur activation est déclenchée par une *action volontaire* de l'utilisateur, contrairement aux deux autres types qui ne sont activables que par le système. Ces liens ne sont pas "programmés" dans l'interface, ils ne sont donc pas gérés par le système hypermédia. Ce sont des liens virtuels, représentés par des règles dans la base du système de surveillance. Plusieurs bases de règles peuvent être prévues, permettant de ce fait, de modifier le comportement du système vis-à-vis de l'usager, sans modifier, ni l'interface, ni les modules EAO.

#### Les liens de suite pédagogique

Les liens de suite pédagogique sont activés par l'apprenant. Le nœud à présenter est prévu par l'auteur, pour un parcours donné.

*Représentation graphique :*




L'interaction avec un objet "Départ" (nœud, bouton, entité de dialogue) provoque l'émission de l'événement "Suite (Départ)". La règle pour la surveillance de ce type de lien a la forme suivante :

**Si Suite (Départ) Alors Aller (Arrivée)**

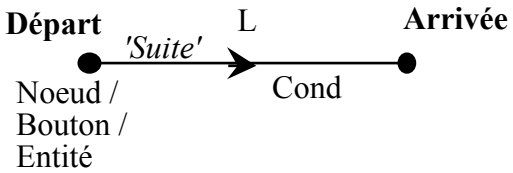
Différents effets peuvent être obtenus à partir d'une même action de l'usager.

## Les liens de référence "conditionnels"

Les liens de référence "conditionnels" sont des liens qui doivent vérifier une condition avant de pouvoir être déclenchés. Ces liens sont utilisés pour prendre en compte différentes "vues" de la base en cours, en fonction du contexte considéré. Nous avons défini deux types de liens conditionnels pour l'hypergraphe "EAO".

- Le premier type caractérise des liens qui sont proposés à l'apprenant uniquement dans certains cas (tant qu'une condition n'est pas vérifiée, le lien n'est pas affiché).

*Représentation graphique :*



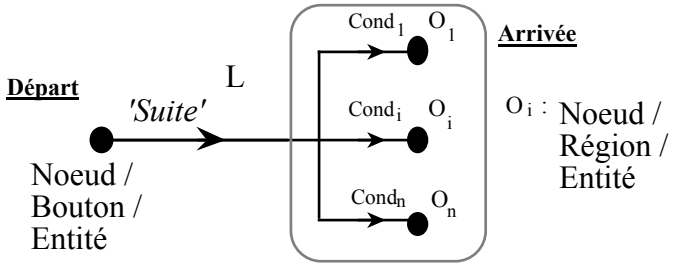

L'activation d'un lien de ce type provoque l'envoi du message "Suite(Départ)", où Départ est la référence de l'objet émetteur. Les règles de surveillance ont la structure suivante :

**Si Cond Alors Afficher (Départ)**

**Si Suite (Départ) Alors Aller (Arrivée)**

- Le second type de lien "conditionnel" caractérise des liens dépendants du contexte. Un lien de ce type est utilisé pour relier le nœud à plusieurs autres nœuds, l'aiguillage s'effectuent en fonction du contexte d'utilisation.

*Représentation graphique :*




Le lien est représenté par autant de règles que de conditions qui le caractérisent. Les conditions peuvent porter sur le nombre de fois qu'un lien a été activé, sur les temps écoulés, etc. Les règles de surveillance ont donc la structure suivante :

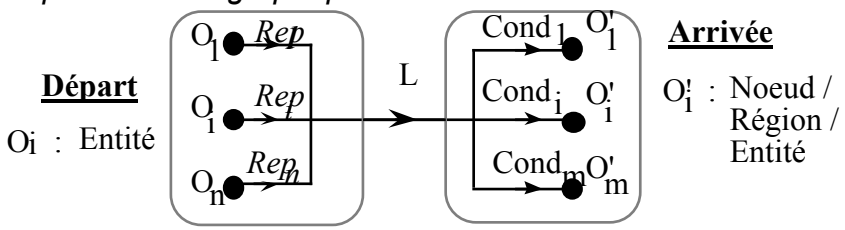
**Si Suite (Départ) Et $Cond_1$ Alors Aller ($O_1$)**

**Si Suite (Départ) Et $Cond_n$ Alors Aller ($O_n$)**

**Les liens de réponse**

Les nœuds de départ contiennent un ou plusieurs objets d'interaction avec lesquels l'apprenant doit communiquer (Cf. 7.2). Ces objets peuvent être des questions, des zones sensibles, des éléments interactifs... L'interaction avec chacun de ces éléments peut provoquer l'envoi d'un événement particulier *Rep (Obj)*, comprenant la réponse donnée. Lorsque l'ensemble des événements correspondant à un lien est reçu par le système de surveillance, le lien est activé. Notons qu'un lien de réponse n'est pas physique, dans le sens où il n'est pas présenté explicitement à l'utilisateur.

*Représentation graphique :*




Un lien de réponse peut avoir un ou plusieurs nœuds de départ. Les liens à plusieurs branches représentent des liens qui sont déclenchés uniquement lorsque plusieurs réponses doivent être fournies. Les règles de suivi auront la structure suivante :

**Si Rep (Départ) Alors Appel_Tuteur (Arrivée)**

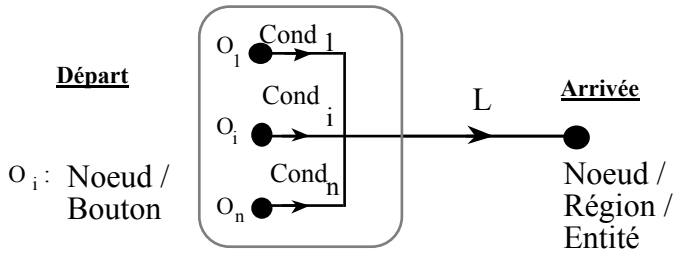
pour les réponses que le système doit analyser dès qu'elles ont été données par l'apprenant,

**Si Rep ($O_1$) Et Rep ($O_2$) Et ... Alors Appel_Tuteur (Arrivée)**

lorsque plusieurs réponses doivent être cumulées avant l'analyse.

**Les liens de déroutement**

Les liens de déroutement sont uniquement utilisés par le système pour guider l'apprentissage et ne peuvent pas être activés par l'utilisateur. Ils sont déclenchés lorsqu'une ou plusieurs conditions sont vérifiées. Les conditions peuvent porter sur les limitations de temps, d'essais ou d'autres variables du système.




*Représentation graphique :*

Les règles de surveillance ont la structure suivante :

**Si $Cond_1$ Et $Cond_2$ Et ... Alors Aller (Arrivée)**

pour l'activation directe du lien (règle de suivi),

**Si $Cond_1$ Et $Cond_2$ Et ... Alors Appel_Tuteur (Arrivée)**

pour la représentation des breakpoints.

En résumé, l'hyperdocument EAO se caractérise par l'utilisation de nouveaux types de liens permettant le contrôle de l'hypertexte par les modules internes de l'application éducative. L'introduction de ces types de liens permet donc *le partage de l'initiative entre le système et l'apprenant*. La figure suivante récapitule les types de liens contrôlés ou activés par le système de surveillance.

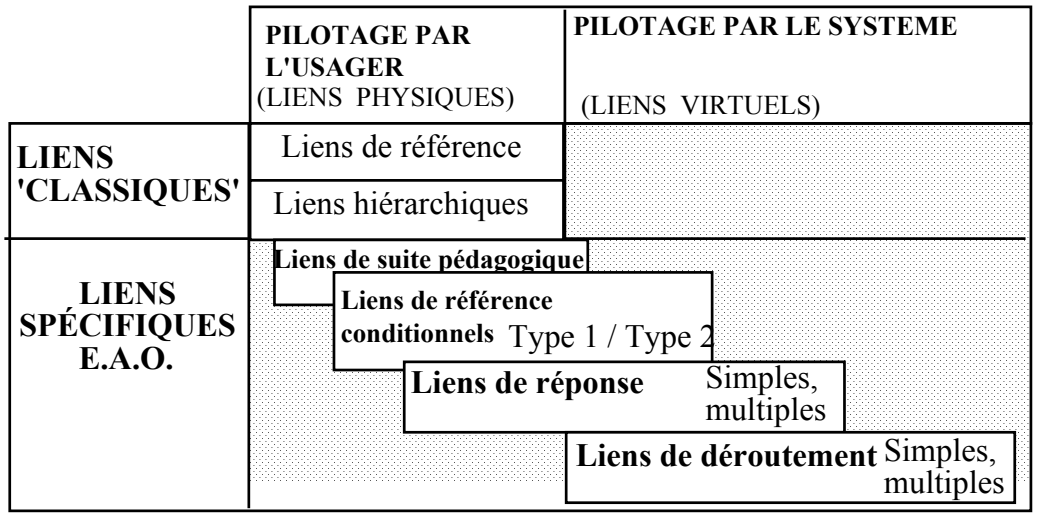



Figure 3 : Les liens de l'hyperdocument EAO

Plusieurs bases de règles peuvent être prévues, permettant de ce fait, de modifier le comportement du système vis-à-vis de l'usager, sans modifier ni l'interface ni les modules EAO.

## 6. Exemple

De manière à illustrer cette présentation théorique, nous donnons un exemple d'hyperdocument éducatif dans lequel les liens introduits sont utilisés.

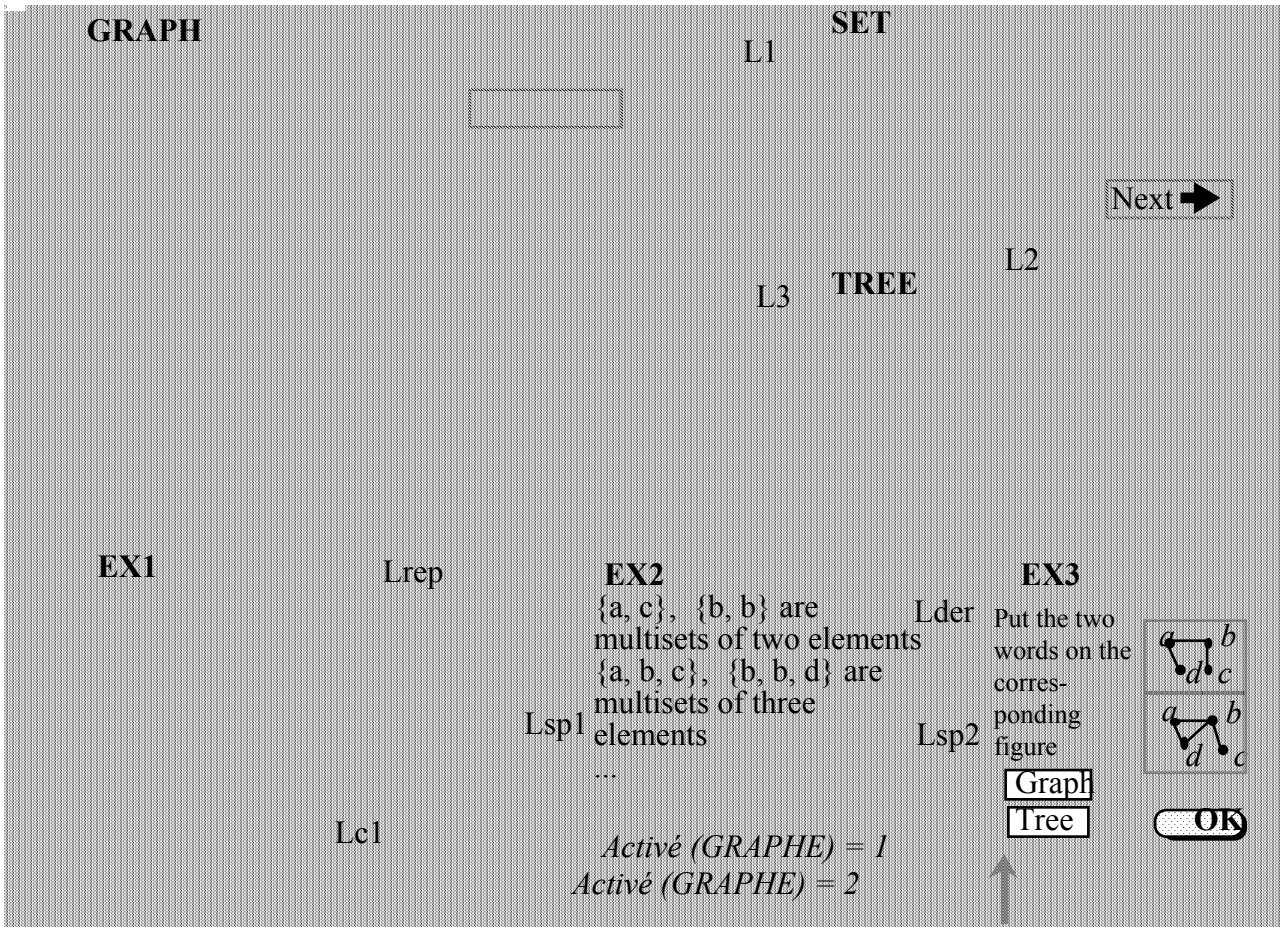



Figure 4 : Exemple de liens dans l'hyperdocument EAO.

Différents éléments constituent l'hyperdocument éducatif : nœuds, boutons, régions, entités de dialogue et liens. La figure 4 représente six nœuds d'un hyperdocument éducatif : "GRAPH", "TREE", "SET", "EX1", "EX2", "EX3".

Deux de ces nœuds ("SET" et "TREE") sont considérés comme des "nœuds communs" ils ne sont pas surveillés et n'émettent aucun événement (l'auteur a décidé qu'ils constituaient des compléments au cours). Les liens qu'ils contiennent sont donc des

liens classiques, figés dans l'interface (en gris clair sur la figure). Ces liens sont "statiques", contrairement aux liens "EAO" qui sont décrits par des règles (en gris foncé sur la figure). Le parcours représenté par les liens L1, L2, L3 est donc considéré comme un parcours hypertexte classique, sans contrôle de la part du système.

Les autres nœuds appartiennent au cours ("nœuds pédagogiques") ; de la même façon que les objets interactifs qu'ils contiennent (entités de dialogue et boutons), ils émettent des événements à destination du système de surveillance.

La base de surveillance pour les nœuds représentés contient les six règles suivantes :

Si Rep (E1) et Rep (E2) alors Appel_tuteur (EX1, R4) */ lien de réponse*
Si Suite (B4) et Activé (GRAPH) = 1 alors Aller (EX2) */ lien conditionnel*
Si Suite (B4) et Activé (GRAPH) = 2 alors Aller (EX3) */ lien conditionnel*
Si Suite (B5) alors Aller (EX2) */ lien de suite*
Si Suite (B6) alors Aller (EX3) */ lien de suite*
Si Temps (EX2) > 3 alors Aller (EX3) */ lien de déroutement*

Seule la première règle est une règle de "breakpoint", elle pourra donc provoquer un appel aux modules d'EAO pour une analyse des réponses. Si aucune décision n'est prise, la session continuera à se dérouler telle qu'elle était prévue (plan prédicatif), par contre si les modules d'EAO décident de changer de base de surveillance ou de réorienter l'usager vers une autre partie de l'Hypertexte, le cours de la session est modifié.

***Remarque*** : *Plusieurs bases peuvent être définies par le concepteur et peuvent ainsi représenter différents parcours ou comportements du système vis-à-vis de l'apprenant. De plus, l'application peut elle-même décider de changer la base de surveillance à un moment donné de la session.*

## 7. Une réalisation

Pour tester les différents aspects de l'architecture proposée, nous avons développé un prototype avec HyperCard 2.0 connecté à un système expert et des modules d'EAO utilisés au laboratoire API. Dans cette application, chaque carte est un nœud, chaque pile est une Unité d'Apprentissage, chaque chemin prévu (plan prédicatif) constitue un groupe de règles pour le système de surveillance. Chaque lien est représenté par une ou plusieurs règles dans la base de travail du système expert associée à l'Unité d'Apprentissage.

### *7.1. L'implémentation des liens "EAO"*

- Les **liens classiques** sont de simples boutons d'HyperCard. Ils font directement référence aux nœuds pointés (script : *"Go Card x"*).

- Les **liens de suite pédagogique** et les **liens de référence conditionnels** sont aussi des boutons d'HyperCard, mais ils ne font pas directement référence à d'autres nœuds : chaque bouton envoie un événement au système de surveillance à l'aide du script suivant :

```
On mouseUp
    get the short name of the target      {prend le nom du lien}
    SendLink it                           {ajoute l'événement à la base de travail}
    Resolution                            {lance un cycle de résolution du système expert}
end mouseUp
```

- Les **liens de réponses** sont représentés, au niveau de l'interface par des entités interactives (Cf. 7.2), développées à cet effet selon le modèle PAC (Coutaz 88). Chaque entité d'interaction appartient à un type prédéfini (zones sensibles, qcm...) et est composée de plusieurs éléments d'interface (boutons, zones de textes, boites à cocher, images, sons...). Chaque entité gère elle-même l'interaction avec l'apprenant (aides, contrôles syntaxiques, changements d'aspects, scrolling...) et envoie un événement réponse indépendant des objets d'interface. Nous avons développé un éditeur d'entités basé sur une approche orientée objet (Cf. 7.2), cet outil peut être utilisé par l'auteur et par l'apprenant.

- Les **liens de déroutement** sont uniquement représentés au niveau du système de surveillance par des règles (ils ne sont pas visibles par l'apprenant).

### *7.2. "Les liens de réponse" au niveau de l'interface usager*

Les liens de réponse sont représentés par des objets composés appelés "**entités de dialogue**". Chaque entité est créée à partir d'une classe caractérisant l'interaction attendue (questions à choix multiples avec réponse unique ou non : QCM_RU, QCM_RM ; sollicitations à zones graphiques : ZI_RU, ZI_RM). Lorsqu'il conçoit un lien de réponse, l'auteur doit choisir le type de dialogue envisagé (classe de l'entité) et utiliser l'éditeur d'entités adéquat pour paramétrer les composants de la nouvelle instance (aspect, valeurs...).

La figure suivante illustre l'approche orientée objet adoptée pour décrire les entités de dialogue. Seules les dernières classes de la hiérarchie sont instanciables (qcm à cases à cocher ou à liste textuelle : “qcm_ru_cc, qcm_rm_cc, qcm_ru_l, qcm_rm_l, zones interactives...), les autres classes servant à "factoriser" les comportements communs.

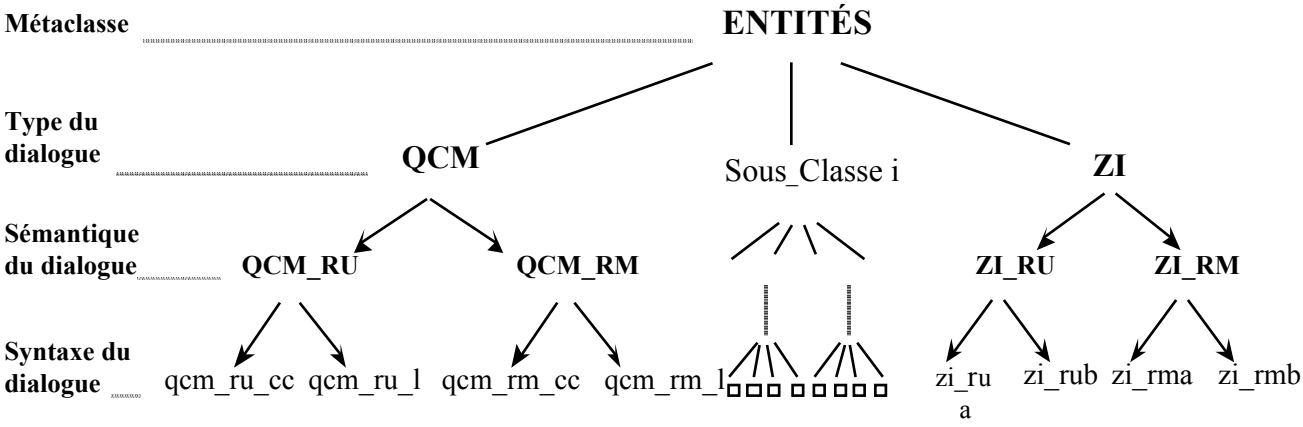
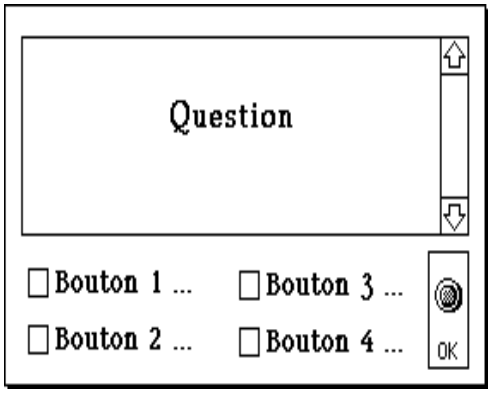



Figure 5 : Représentation des entités de dialogue pour un Hyperdocument EAO

Chaque entité gère l'interaction avec l'usager, en fonction de sa classe (défilement de texte, sélection des boites à cocher, déplacement d'objets, sélection de zones, entrées de texte...) et déclenche un événement réponse lorsque c'est nécessaire. Chaque événement est indépendant de la classe de l'entité qui l'a généré et ne fait pas référence aux composants de l'interface : il peut déclencher une règle de surveillance (lien de réponse) et provoquer une activation des modules d'EAO.



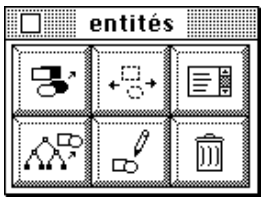

Cette classe d'entité est composée d'un texte de question de boites à cocher pour les réponses et d'un bouton 'Ok' pour la validation. Lorsque ce dernier bouton est activé, l'entité envoie au système de surveillance un événement réponse contenant le nom des cases sélectionnées. L'éditeur général d'entités permet à l'auteur d'instancier, de déplacer ou d'effacer une entité. Un autre éditeur, adapté à la classe considérée permet de modifier les valeurs des attributs : un auteur peut ajouter ou enlever des cases réponses, modifier le contenu et l'aspect des textes, déplacer les composants, tout en conservant le comportement de l'entité.

Figure 6 : Exemple d'entité

La réutilisation de composants peut être effective au niveau de l'interface (copier-coller une entité), au niveau du système de surveillance (copie d'une règle) ou au niveau des modules d'EAO (réutilisation de "briques" logicielles pour l'ajout ou la modification des fonctionnalités du système global). L'architecture à trois systèmes améliore le développement du logiciel éducatif tout au long des différentes phases de son cycle de vie (conception, prototypage, simulation, réalisation, validation et maintenance) à la fois au niveau de l'interface usager et au niveau didactique.

Différents outils permettent à l'auteur concepteur de manipuler les entités de dialogue :

**Palette générale**
Editeur d'entités




Dupliquer,
déplacer,
lister les entités,
instancier une classe,
éditer les composants,
effacer une entité.

**Palette exemple**
Editeur pour une classe particulière

Editer (déplacer, changer l'aspect ...), effacer, nouveau bouton.
Editer (déplacer, changer l'aspect ...), effacer, nouvelle zone.
Editer (déplacer, changer l'aspect ...), du texte de la question, du texte d'aide, ...

Remarque : Certaines fonctions de cet outil auteur peuvent être utilisées par l'apprenant (par
exemple, la modification de l'aspect d'une entité).

Figure 7 : L'édition des entités de dialogue

Actuellement, une **formalisation de l'hyperdocument EAO** sous forme de *base de donnée déductive* est étudiée pour faciliter la description et l'échange des informations entre les différents outils de l'auteur. Pour cela, nous nous basons sur les modèles proposés par (Lange 90) et (Halasz 90) à partir desquels nous proposons la prise en compte des nouveaux types de liens et des entités de dialogue.

## 8. Conclusion et perspectives

Le concept d'**hyperdocument EAO** s'est peu à peu imposé comme une solution permettant d'allier les avantages respectifs des systèmes Hypermédias et des systèmes d'EAO (ou EIAO). L'idée d'une architecture unique, autorisant la production d'applications à divers degrés de contrôle est ainsi devenue un objectif prioritaire pour notre étude. Le modèle développé nous a permis de présenter les différents aspects qui définissent le concept d'hyperdocument EAO :

- une *architecture* à trois composantes (système de présentation hypermédia, système de surveillance, modules d'EAO),
- des constituants essentiels (les *entités de dialogue* pour une interaction bénéficiant du modèle PAC, les *liens-EAO* pour le contrôle de l'interaction),
- une *formalisation* en bases de données déductives.

En ce qui concerne l'architecture à trois composantes, nous avons mis en œuvre les mécanismes de gestion des événements et les principes de séparation entre l'interface et les modules pédagogiques : les différents types de liens-EAO sont des liens **dynamiques**, représentés par des règles de production. Ainsi, à partir d'une même action au niveau de l'interface, les différents acteurs (auteur, moniteur, apprenant) peuvent paramétrer le comportement du système. En fonction des types de liens utilisés, le système global sera **plus ou moins directif**.

En ce qui concerne **les entités de dialogue**, plusieurs résultats peuvent être mis en évidence :

- Au niveau de l'apprenant la communication devient "standardisée" car les différentes classes d'entités ont un **comportement uniforme** d'une application à l'autre. D'autre part, la gestion du dialogue local par les entités permet un dialogue à plusieurs fils d'activité (plusieurs sujets d'intérêt) et une réaction adaptée aux erreurs de manipulation.

- Pour l'auteur, la représentation orientée objet des entités de dialogue facilite la **conception** et le **prototypage** d'un hyperdocument éducatif (le choix d'une classe d'entité de dialogue par l'auteur détermine l'interaction). De plus, la **réutilisation** des entités d'un hyperdocument éducatif facilite le travail de l'auteur. Enfin, la **répartition des tâches** entre le médiatiseur et le pédagogue (c'est-à-dire, entre la sémantique et l'aspect du dialogue) est un avantage important pour la mise au point d'une application éducative.

Le prototype développé démontre enfin que différents types de logiciels éducatifs (didacticiels à guidage fort jusqu'aux tuteurs intelligents décidant dynamiquement des orientations du cours), basés sur l'hypermédia, peuvent être développés à partir de l'architecture proposée.

En résumé, deux aspects de notre contribution méritent d'être soulignés :

- une architecture unique permet la réutilisation des différentes parties réalisées,
- un système hybride constitue à la fois un outil d'EAO efficace (utilisation de composants d'EAO) et un outil de formation attrayant (interaction hypermédia).

Bien qu'un système complet n'ait pas encore été développé, nous pensons que la solution basée sur l'utilisation de standards, combinée avec nos efforts pour développer des logiciels éducatifs hypermédias, constitue une étape importante vers de nouveaux environnements de formation mis au point et maintenus par des techniques et outils du génie logiciel.

## Références

(Allison 88) L.J. Allison, N.V. Hammon, Development and evaluation of a CAL system for non-formal domains : the Hich-hiker's guide to cognition, *Computer and education*, n° 12, pp. 215-220.

(Allison 89) L.J. Allison, N.V. Hammon, A learning support environment : the Hich-hiker's guide, In Mc Aleese (Ed.), *Hypertext : from theory to practice.* Norwood, New Jersey : Ablex.

(Beltran 89) T. Beltran, M.N. Bessagnet, M.F. Canut, G. Gouardères : Une approche génie logiciel pour produire des tuteurs intelligents, *Actes des 9èmes journées internationales Les systèmes experts et leurs applications*, Avignon, pp.469-483.

(Beltran 91) T. Beltran : Hypermedia and Learning : an architecture for hybrid systems, *Actes des journées internationales "CALISCE* 91, 9-11 sept. 1991, Lausanne, pp. 167-176.

(Bessagnet 90a) M.N. Bessagnet, T. Nodenot, G. Gouardères : A new approach : Coursware engineerig, *WCCE 90, 5th Word Conference on Computer Education*, Sydney, Australia, pp. 339-344.

(Bessagnet 90b) M.N. Bessagnet, T. Nodenot, G. Gouardères : L'AGDI : un environnement de développement pour des applications EAO, *Actes des journées : Le Génie Logiciel et ses applications,* Toulouse, déc. 1990, pp. 441-461.

(Coutaz 88) J. Coutaz, *Interface Homme-Ordinateur : Conception et Réalisation*, thèse de l'université J. Fourier, Grenoble, 1988.

(Ferret 87) E. Ferret, G. Weidenfeld : Exporatory environments and Intelligent Tutoring Systems, *Deuch journal for education*.

(Gouardères 90) G. Gouardères : Le Génie Didacticiel à la recherche d'un standard : l'AGDI, Revue *TSI* vol.9 n°5, pp. 449-455.

(Halasz 90) Halasz, Swartz : The Dexter hypertext reference model, *NIST'90, Hypertext Standardization Workshop*, 16-18 Juill. 1990.

(Jonassen 90) D.H. Jonassen, H. Mandi : Designing hypermedia for Learning, Computer and Systems Sciences, vol. 67, *NATO ASI series, Proceedings of the NATO Advanced Research Workshop on Designing hypertext / hypermedia for Learning.*

(Lange 90) D. Lange : A formal model for hypertext, *NIST'90, Hypertext Standardization Workshop*, 16-18 Juill. 1990.

(Mayes 88) J.T. Mayes, M.R. Kibby : StrathTutor : the development and evaluation of a learning-by-browsing system on the macintosh, *Computers in Education,* 12, pp. 221-229.

(Mayes 89) J.T. Mayes, M.R. Kibby, H. Watson : Towards intelligent hypertext, In R. Mc Aleese (Ed.), *Hypertext : from theory to practice.* Norwood, new Jersey : Ablex.

(Mûhlhauser 90) M. Mühlhauser : Issues of integrated authoring/learning environments, *WCCE' 90, 5th Word Conference on Computer Education*, Sydney, Australia, pp. 419-424.

(Storrs 88) G. Storrs : An intelligent hypertext-based training system, In P. Duffin (Ed.) *KBS in government : Proceedings of the second european conference,* Blenheim, London.