



**HAL**  
open science

## Un système d'hypertextualisation pour générer des systèmes d'aide en ligne intelligente pour des applications logicielles

Jean-Claude Hochon, Fabrice Evrard

► **To cite this version:**

Jean-Claude Hochon, Fabrice Evrard. Un système d'hypertextualisation pour générer des systèmes d'aide en ligne intelligente pour des applications logicielles. Deuxième colloque Hypermédias et Apprentissages, Mar 1993, Lille, France. pp.183-195. edutice-00000539

**HAL Id: edutice-00000539**

**<https://edutice.hal.science/edutice-00000539>**

Submitted on 16 Jul 2004

**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.

## UN SYSTÈME D'HYPertextUALISATION POUR GÉNÉRER DES SYSTÈMES D'AIDE EN LIGNE INTELLIGENTE POUR DES APPLICATIONS LOGICIELLES

**Jean-Claude Hochon** <sup>(1)</sup> <sup>(2)</sup>, **Fabrice Evrard** <sup>(2)</sup>

(1) XILOG (Conception et Services pour l'Ingénierie du LOGiciel)  
76, rue de la Colombette - 31000 TOULOUSE - FRANCE

Tel : (+33) 61.62.84.72 - Fax : (+33) 61.63.67.26

&

(2) ENSEEIHT-IRIT - Groupe GRAAL - Dept INFORMATIQUE  
2, rue Camichel - 31071 TOULOUSE Cedex - FRANCE

Tel : (+33) 61.58.83.68 - Fax : (+33) 61.62.09.76 -

E-mail : {hochon, evrard}@enseeiht.fr

***Résumé** : Nous présentons dans ce papier un système hypertextuel permettant de générer des aides en ligne intelligentes. Celui-ci utilise l'hypertextualisation de textes formellement continus pour structurer une base de données documentaires afin d'obtenir des systèmes d'aide en ligne pour des applications d'un environnement informatique. L'interface utilisateur du système que nous décrivons ici fournit des fonctionnalités permettant une activité annotationnelle et une consultation intelligente des documents résultants.*

### 1. LA PROBLÉMATIQUE DE L'ÉTUDE

Depuis l'apparition de l'ordinateur, des applications logicielles de tous genres ont été développées dans le but de faciliter les tâches auxquelles celles-ci se rapportent. Mais compte tenu de la complexité de plus en plus croissante de ces applications, leur utilisation est devenue problématique, créant ainsi de nouveaux besoins d'assistance à leur appropriation. La vulgarisation des systèmes d'aide en ligne n'a commencé que vers 1975 (Kearsley, 88) ; mais jusqu'à présent, bien qu'étant une composante essentielle des applications logicielles, cette fonctionnalité n'y est pas automatiquement intégrée par manque d'outils génériques et de méthodes. De ce constat est née notre motivation de concevoir un environnement permettant de développer aisément et à moindre coût des systèmes d'aide en ligne pour des applications informatiques.

Un système d'aide est tout dispositif susceptible d'apporter de l'assistance à un opérateur dans un domaine donné ; l'objectif d'un tel système est de permettre à l'opérateur de poser des questions puis de répondre à celles-ci. Dans le domaine de l'informatique, une étude de l'existant nous a permis de classer les différents

systèmes d'aide en ligne. Les outils d'assistance à l'utilisateur existent sous différentes formes : documentation papier, aides en ligne, didacticiels, systèmes hypertextes.

### 1.1. La documentation papier

Elle est souvent constituée d'une grande masse de manuels : manuels d'utilisation, manuels de référence, manuels de programmation, tutorials...

Un **manuel utilisateur** contient une description complète de l'environnement du système de commandes ; il décrit de façon didactique certains concepts de base du système de commandes.

Un **manuel de référence** contient une liste complète de toutes les commandes du système ; il s'agit en général d'une référence alphabétique de toutes les commandes ; chaque entrée comporte la syntaxe, les arguments, les options, la description fonctionnelle, la valeur renvoyée, la liste des commandes associées, des exemples, le contexte d'utilisation... Cette liste est généralement organisée en chapitres et sous-chapitres afin d'accéder plus aisément à l'entrée dont le lecteur a besoin.

Un **manuel du programmeur** décrit tous les concepts du système de commandes ; il contient une description formelle exhaustive de celui-ci ; cette description est parfois suivie d'explications et d'exemples. Ce manuel est parfois fusionné avec le manuel de référence.

La documentation papier est la plus répandue mais très peu utilisée car l'information recherchée est difficilement trouvable et peut renvoyer à des notions inconnues (tout comme dans un dictionnaire) ; elle ne s'adapte pas à l'utilisateur et ne prend pas en compte le contexte de la demande.

### 1.2. Les aides en ligne

Les systèmes d'aide en ligne opérationnels structurent l'information en hiérarchie (exemple : l'aide en ligne des environnements UNIX et VMS). Avec l'évolution des Interfaces Homme-Machine, l'interface utilisateur de ces systèmes a été améliorée par l'utilisation du langage naturel - c'est le cas du système INTERIX (Guez, 87) - ou par celle des interfaces évolutives à manipulation directe - c'est le cas des systèmes **AnswerBook** de l'environnement Solaris de *Sun Microsystems Computer Corporation*, **Bookreader** de l'environnement DecWindow de *DEC* et **NeXTHelp** de l'environnement NeXTstep de *NeXT Corporation*.

Permettant d'une part d'éviter l'alternance entre l'écran et le papier et d'autre part d'accéder à la documentation séquentiellement, par mots-clés ou par thèmes, les aides en ligne sont de plus en plus intégrées aux logiciels ; malheureusement, elles ont souvent une organisation livresque et s'adaptent rarement à l'utilisateur malgré des recherches récentes dans le domaine des Sciences Cognitives permettant d'individualiser ou de personnaliser des relations homme-machine par modélisation cognitive de l'utilisateur (Katzeff, 90) ou par traitement des traces de son activité (Kerr, 90).

### 1.3. Les didacticiels

Le didacticiel d'un système permet à un utilisateur de se former sur celui-ci. Devant les problèmes posés par l'autoformation à l'aide de manuels traditionnels, plusieurs solutions ont été abordées :

- l'enseignement programmé par l'utilisation de livres adaptés à l'autoformation est basé sur le concept d'unité élémentaire, séquence pédagogique comprenant une scène (présentation d'une information à l'utilisateur), une sollicitation (invitant l'utilisateur à répondre à une question) et une validation (feed back informatif concernant la réponse de l'utilisateur) ;
- profitant de la vulgarisation de l'ordinateur, les didacticiels (encore appelés systèmes d'Enseignement Assisté par Ordinateur) ont remplacé l'enseignement programmé qui a pour défaut majeur d'être purement livresque ;
- l'utilisation des techniques de l'Intelligence Artificielle a ouvert une nouvelle voie ; les didacticiels sont devenus des tutoriels intelligents d'où le nom de systèmes d'Enseignement Intelligemment Assisté par Ordinateur (Nicaud et al., 88).

Il existe différents types de didacticiels (SEAO classiques, tuteurs intelligents, environnements d'apprentissage...), spécifiquement construits pour résoudre chacun un problème particulier, ce qui ne facilite guère leur réutilisabilité.

### 1.4. Les systèmes hypertextes

Ces systèmes permettent d'accéder à des informations et de consulter la documentation en offrant des moyens d'accès évolués (accès direct et navigation au sein de structures non linéaires), d'où une amélioration considérable de l'adéquation à la demande. Les informations sont stockées dans des noeuds (unités d'information) interconnectés (Daniel-Vatonne, 90). De plus, les systèmes hypermédia intègrent plusieurs modes d'information au niveau des noeuds : texte, graphique, image, son, animation. Mais la grande liberté de circulation dans l'information autorisée par ces systèmes conduit souvent l'utilisateur à se perdre et donc sortir du contexte de sa demande.

### 1.5. Critiques

Les inconvénients majeurs de ces outils d'assistance résident au niveau (Hochon et al., 92b) :

- des interfaces utilisateurs et de leur ergonomie : l'adaptabilité du système (aides en ligne, didacticiels) à l'utilisateur est rarement prise en compte (Frasson, 91) ;
- de leur structuration : ces systèmes sont généralement structurés en termes de logique de fonctionnement plutôt qu'en termes de logique d'utilisation (Richard, 83) ; souvent, l'utilisateur ne peut pas formuler ses buts. Cet inconvénient découle du fait que l'aide en ligne est conçue comme étant une version informatisée de la documentation papier qui elle-même est conçue selon la logique fonctionnelle de l'application décrite ;

- de leur diversité : il faut noter que pour un même environnement (i.e. un même système d'exploitation), ces différents systèmes peuvent varier considérablement, obligeant chaque fois l'utilisateur à passer par une phase d'appropriation du système d'aide avant celle de l'application. L'évolution des environnements (MacOS, MS-Windows, NeXTStep) tend à favoriser une unification de l'interface utilisateur de ces systèmes à condition de disposer d'outils génériques de développement de systèmes d'aide ;
- de leur faible productivité : ceci est dû à l'absence de générateurs de systèmes d'aide. Cependant, on distingue des logiciels permettant de réaliser des aides en ligne pour diverses applications ; mais le fait qu'ils ne soient pas conçus pour cette tâche, leur utilisation pour le développement de systèmes d'aide demande des efforts non négligeables, ce qui rend celle-ci très coûteuse ; dans l'environnement Macintosh, HyperCard en est un exemple.

Partant de ces critiques, nous avons défini un modèle d'environnement pour une nouvelle génération de systèmes d'aide intelligente. Celui-ci, basé sur l'hyper-textualisation de textes formellement continus, offre un cadre permettant de développer puis d'utiliser aisément des systèmes d'aide en ligne, d'éviter les écueils de la documentation papier et de s'adapter à son utilisateur en tenant compte du contexte de sa demande et bien évidemment de son évolution.

Nous nous proposons de présenter brièvement les fonctionnalités et l'architecture logicielle de ce modèle, puis de développer le système d'information qui a été conçu autour de la notion de documents.

## **2. LES FONCTIONNALITÉS DE L'ENVIRONNEMENT**

Les documents électroniques constituent le matériau d'applications d'une vaste famille de systèmes (traitements de textes, PAO, OCR...) ; ils fournissent par ailleurs des données à des systèmes tels que ceux de l'EAO, l'EIAO, les aides en lignes... Nous nous intéressons ici aux fonctionnalités de systèmes didactiques tels que les aides en ligne.

Nous les avons définies à partir de la taxinomie de la documentation en ligne conçue par Shirk (Shirk, 88) et dont les différents niveaux sont : les messages système, les messages d'erreur, les facilités d'aide, les guides de référence, les tutoriels et les didacticiels. Nous avons classé les fonctionnalités qui découlent de cette taxinomie en quatre catégories distinctes.

### **2.1. Les fonctionnalités de consultation**

Comparables aux fonctionnalités des systèmes hypertextes, elles permettent d'accéder à l'information par différents modes (direct, contextuel, conditionné, par le contenu) et de naviguer astucieusement dans la base du système. Les différents types de questions permettant d'accéder à l'information sont les demandes de définition (i.e. définition d'un concept), les demandes de caractéristiques d'entrées (i.e. les arguments d'une fonction), les demandes de l'effet de commandes (i.e. résultat d'une fonction), les vérifications de définition ou de caractéristiques d'entrées, les demandes de relations entre entrées (i.e. synonymie, hiérarchie...). Nous distinguons trois modes de navigation : la navigation séquentielle (elle est

définie soit par l'organisation séquentielle du document ou par un chemin prédéfini), la navigation aléatoire (au vu des informations affichées, l'utilisateur décide de choisir sa destination) et la navigation visuelle (l'affichage graphique du réseau des informations permet de choisir sa destination).

## **2.2. Les fonctionnalités de guidage**

Elles permettent de délimiter l'espace de navigation de l'utilisateur en fonction de son profil (ou de sa demande) et de le guider dans sa recherche. La délimitation de l'espace de navigation est caractérisée par le profil de l'utilisateur (contraintes fixées par l'utilisateur, modélisation cognitive, traces d'activités) et/ou par la portée de la demande (contexte de la demande, concepts mis en œuvre). Le guidage est caractérisé par les stratégies d'apprentissage (Hochon et al., 92b) que le concepteur de la base d'informations a développées (réseaux ordonnés d'apprentissage des concepts du domaine). Nous présentons ici un modèle de l'utilisateur du type overlay (Meirieu, 89) basé sur des informations simples concernant l'utilisateur :

- situation générale (nom, prénoms...),
- domaine cognitif (liste des concepts acquis, niveau...),
- domaine affectif (choix du mode pédagogique, liste des objectifs de la session...),
- historique (nombre de sessions, concepts étudiés...).

Le modèle cognitif que nous utilisons est caractérisé par le fait que les informations proposées à l'utilisateur vérifient son profil qui est calculé. Les informations relatives aux utilisateurs du système sont constamment mises à jour au cours d'une session et sauvegardées en fin de session.

Le calcul du profil de l'utilisateur se fait par analyse des concepts acquis et des objectifs de l'utilisateur (concepts à acquérir) ; la fonction "profil" définit un ordre total sur l'ensemble des points de vue de la base d'informations de la manière suivante :

- \* un ordre est fixé initialement sur l'ensemble de points de vue de la base d'informations par son concepteur ;
- \* à partir des informations de l'historique de l'utilisateur, en l'occurrence de la liste des concepts appris, on calcule le poids de chacun des points de vue par addition des occurrences de ceux-ci sur l'ensemble des concepts étudiés ; ces poids nous permettent d'ordonner les points de vue propres à l'utilisateur ;
- \* on utilise cet ordre des points de vue pour ordonner la liste des prérequis et des suggestions pour un concept donné ; de cette liste, on supprime tous les concepts qui ne vérifient pas les objectifs de l'utilisateur ;
- \* en cas d'égalité des poids des points de vue, l'ordre par défaut des points de vue est pris en compte.

Ce modèle empirique utilise une approche suivie se référant à une auto-modélisation de l'utilisateur par utilisation de ses traces d'activités (Kerr, 90).

### 2.3. Les fonctionnalités de génération

Elles sont spécifiées autour de la notion de documents. Nous définissons un système d'aide en ligne comme un ensemble de documents (manuels de référence, d'utilisation, de programmation...) manipulés au travers d'une interface homme-machine intelligente. Pour le concepteur du système d'aide (qui n'est pas forcément celui des documents !), nous définissons ces fonctionnalités de génération à partir de celles d'un *lecteur actif* qui s'approprie les documents. Le comportement de ce lecteur vis-à-vis des documents peut être vu comme une activité communicationnelle spéciale ; cette approche identifie les aspects de structuration et d'architecturation d'un document comme les traces d'une activité métalinguistique et logico-cognitive du texte : les traces peuvent se réaliser sous des formes discursives particulières (marques de division, titres, paragraphes, entrées...) ou des propriétés de formatage (mises en évidence typo-dispositionnelles...).

Les différents travaux de recherche de notre équipe sur les pratiques annotatives et les documents électroniques (Evrard et al., 90 ; Hochon et al., 92a ; Virbel et al., 92) nous ont permis de définir la transposition des pratiques traditionnelles de lecture (pratiques graphiques, sélection et extraction de passages, recopie dans des cahiers de notes, structuration, architecturation, annotations...) dans un contexte informatisé, ce qui offre une méthode hautement personnalisée de remodelage d'une documentation source non ciblée. De ces travaux, nous avons retenu les fonctionnalités qui sont pertinentes pour la conception d'une base de connaissances de systèmes d'aide en ligne : hiérarchisation de passages, caractérisation de passages, association de mots-clés et d'informations diverses à des passages, établissement de références croisées entre passages, définition de liens typés entre passages, définition de chemins de lecture. Ce système nous permet donc de définir à partir de documents sources, un ensemble d'entrées caractérisées par des mots-clés et interconnectées entre elles.

### 2.4. Les fonctionnalités de personnalisation et d'appropriation

La personnalisation est caractérisée d'une part, par les fonctionnalités permettant à tout utilisateur d'intégrer sa propre base de connaissances dans le système d'aide et d'autre part, par la possibilité d'individualiser l'environnement d'aide. L'appropriation quant à elle, permet à l'utilisateur d'intégrer des annotations personnelles ou publiques (remarques, suggestions, réactions diverses, commentaires, résumés...) aux bases d'informations existantes. Ces annotations rappellent celles rattachées aux chunks dans certains systèmes hypertextes (Catlin et al., 89). Ces fonctionnalités permettent à un utilisateur du système d'aide d'enrichir de façon personnelle la base d'informations.

## 3. L'ARCHITECTURE LOGICIELLE

Dans le souci d'uniformiser les systèmes d'aide d'un même environnement et de concevoir un système générique, nous avons défini une architecture (Hochon, 92c) qui met en exergue la place du système d'aide dans l'environnement et ses relations avec les autres applications.

Le système d'aide est une application comme toute autre qui joue le rôle de serveur à toutes les autres applications ; sa base de connaissances est donc constituée d'un ensemble de sous-bases, chacune se rapportant à une application précise et pouvant avoir des interactions avec les autres ; comme illustration de cette architecture, citons l'environnement Macintosh avec l'utilisation de HyperCard comme générateur de systèmes d'aide même si la prise en compte du contexte de la demande et les relations entre sous-bases (ici des piles) ne sont pas évidentes à gérer (Jourdan de La Passardière, 90) ou encore NeXTSTEP release 3 (l'environnement des stations NeXT) qui intègre des outils de développement d'aide intelligente.

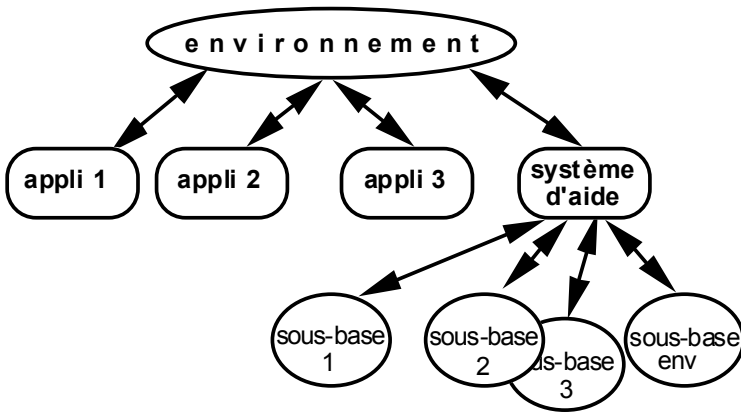


Figure 1 : l'aide en ligne dans l'environnement

L'architecture logicielle que nous avons définie pour notre environnement de développement et d'utilisation de systèmes d'aide en ligne (Hochon, 92b) est représentée par la figure suivante :

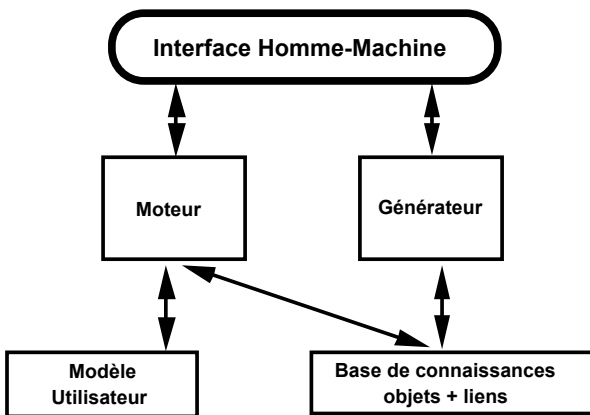


Figure 2 : Architecture logicielle du système

Le générateur permet de définir à partir de la structuration de documents des systèmes d'aide en ligne alors que le moteur est l'application qui contrôle la consultation de ceux-ci. Ces deux applications sont dotées d'une interface utilisateur



sophistiquée rendant le système intelligent ; en effet, le moteur est doté d'un modèle de l'utilisateur (Hochon et al., 92a) permettant de personnaliser la demande de l'utilisateur en tenant compte de ses acquis. Ces applications puisent les informations dans une base de connaissances constituée d'une base de données documentaires et d'une base de données informatives. Nous présentons la structuration de cette base de connaissances par la présentation du système d'information de l'environnement.

#### 4. LE SYSTÈME D'INFORMATION DE L'ENVIRONNEMENT

Dans le domaine des applications informatiques, les informations utilisées pour assister l'utilisateur se trouvent dans les manuels d'utilisation, les manuels de référence, les manuels de programmation, les tutoriels, les aides-mémoires ou les livres de cours. Tous ces documents sont utiles si on se place dans l'optique d'un environnement d'apprentissage où chaque utilisateur (quelque soit son niveau) gère lui-même son apprentissage. Une étude du contenu de quelques-uns de ces documents nous a permis de proposer une structuration des connaissances en différents objets pouvant avoir des caractéristiques diverses et variées (Hochon, 92a). Elle est basée en partie sur les manuels de Le-Lisp et AIDA, ce choix ayant été guidé par notre contribution au projet ELSY (European Lisp SYstem), projet Eurêka ayant pour but de développer un environnement de développement pour Le-Lisp (Eurêka, 89).

Le modèle de structuration que nous avons conçu se rapproche beaucoup des modèles dictionnaires ; on distingue différents types d'objets manipulés dans ces manuels :

- une *entrée* désigne une commande, une fonction ou une entité atomique du domaine ;
- un *concept* symbolise une idée générale et abstraite développée dans le domaine considéré ; par exemple dans le domaine des traitements de texte, édition, sauvegarde et document sont des concepts ;
- un chapitre est un regroupement logique hiérarchisé d'entrées que nous appelons *thème*, sous-thème ou sur-thème ;
- un *cours* est une description didactique traitant d'un ou plusieurs concepts de base du domaine ; il peut être composé des objets suivants : *exercices*, *solutions*, *exemples* ;
- chaque objet peut être caractérisé par des *critères* qui sont soit des *concepts* de base du domaine, soit des *points de vue*, ceux-ci déterminant une caractéristique du public visé ; des exemples de points de vue sont : débutant, moyen, expert, ou encore analyste, concepteur, programmeur.

Ces différents objets peuvent être vus comme des entrées d'un dictionnaire encyclopédique, décrites par des entités textuelles, graphiques, imagées dont la lecture peut renvoyer à d'autres entrées.

#### 4.1. La structuration de la connaissance

Dans une perspective de généralisation et d'uniformisation, nous avons représenté la connaissance en un ensemble d'objets interconnectés par des liens divers. On distingue deux classes d'objets :

- une *unité logique de documents* (ULD) est la plus petite entité de données que l'on peut manipuler dans ces manuels ; un exemple d'ULD serait un paragraphe, une image ou un graphique ; chaque ULD a un type (texte, graphique, image, son) ; elle est caractérisée par un nom et une localisation (soit un nom de fichier simple ou avec des coordonnées) ;
- une *unité logique d'information* (ULI) est un objet d'information caractérisé par un nom, un type, une liste d'attributs ayant chacun pour valeur une collection d'ULD ou d'autres ULI de types quelconques ; elle représente par exemple une entrée de type concept du domaine, une entrée de type fonction dans le cas du manuel de référence de Le-Lisp ou un sous-chapitre dans le manuel d'utilisation de AIDA.

Cette représentation a l'avantage d'être générique car une entrée d'un type donné peut être représentée en spécialisant l'entrée générique. Les objets que nous avons décrits dans le paragraphe précédent ont été représentés par spécialisation de ces deux classes d'objets.

Le système s'apparente donc à un hypertexte ; les objets sont créés à partir de documents représentant la base de données documentaires. Dans la première partie de nos travaux, nous avons défini un langage de marquage (Hochon et al., 92b) qui est une extension au formateur de texte LaTeX (Lamport, 86) et qui permet de structurer un document en des objets textuels dotés de caractéristiques logico-sémantiques et typo-dispositionnelles (corps, graisse, retraits...). Les caractéristiques logico-sémantiques sont les propriétés permettant de typer les objets (introduction, chapitres, fonctions, mots-clés, familles, titres...) et de les relier (liens de synonymie, hiérarchiques (Daniel-Vatonne, 90), référentiels (DeRose, 89 ; Daniel-Vatonne, 90), de guidage (Zellweger, 89)). Nous avons aussi développé :

- un éditeur structuré permettant au concepteur de spécifier aisément sa base de connaissances (sans connaître les marqueurs) ;
- et un compilateur permettant de générer un dictionnaire de tous les objets spécifiés. Le processus de compilation inhibe les marqueurs insérés dans le document source (notion de marqueurs cachés) et facilite l'accès aux objets informatifs définis (les entrées du dictionnaire) ; cette solution, bien que limitée aux documents de format LaTeX, a permis de valider notre modèle de génération d'aides en ligne.

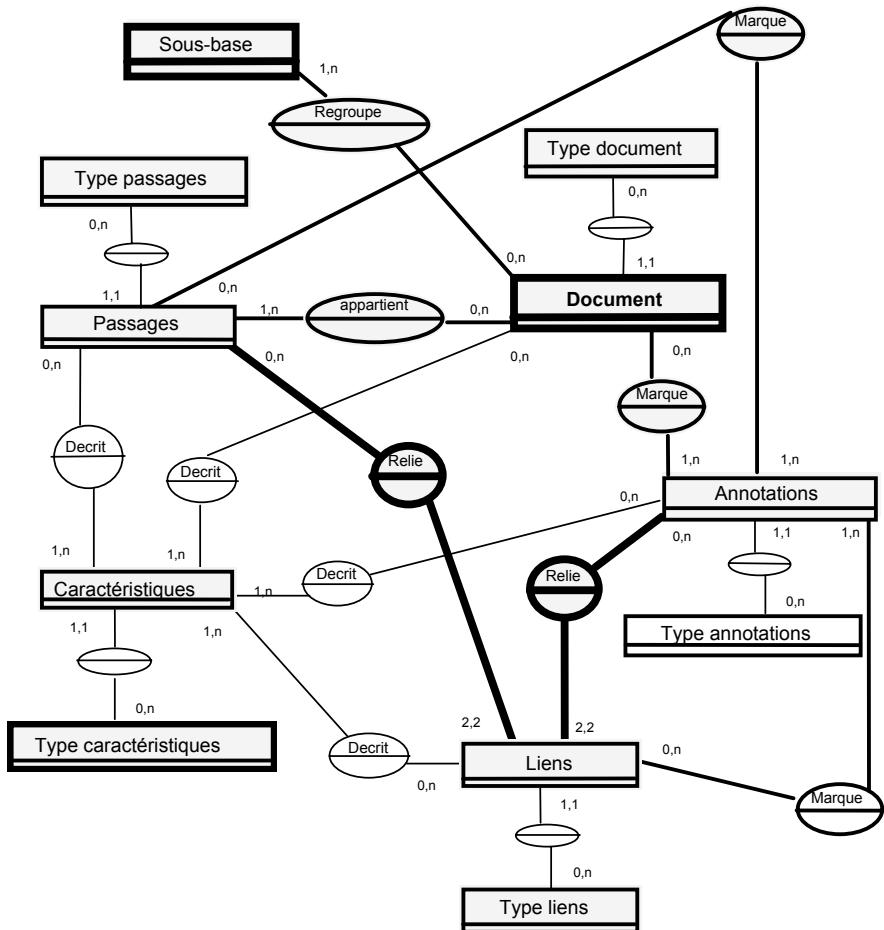
Les documents, le dictionnaire d'objets et le dictionnaire de liens constituent une sous-base de notre système d'aide ; les interactions entre sous-bases distinctes sont gérées au niveau du dictionnaire de liens. Cette sous-base de connaissances est utilisée par le moteur qui dialogue avec l'utilisateur pour répondre aux sollicitations de celui-ci.

Nous étendons actuellement ce système aux documents électroniques (prise en compte d'autres formats) en développant des fonctionnalités de définition automatique d'unités d'information par l'utilisation d'un langage annotational (Virbel et al., 92).

### 4.2. Le modèle conceptuel de la connaissance

Nous formalisons ici sous la forme d'entités et de relations la définition du modèle conceptuel (simplifié) de données de l'environnement. De ce modèle, nous déduisons un schéma d'organisation de base relationnelle en cohérence avec un environnement conçu en termes d'objets.

Ce schéma spécifie les différents objets qui sont manipulés par notre modèle ; il signifie en particulier qu'un document d'un type donné (manuel d'utilisation, manuel de conception détaillée...) est organisé en différents passages dotés de caractéristiques typographiques et reliés entre eux par des liens typés ; à ces objets (document, passage, lien), on peut rattacher des annotations typées (textuelle, imagée, vocale...).



### 4.3. La mise en oeuvre

Nous utilisons des fichiers de documents et un système de gestion de base de données relationnelles pour mémoriser tous les objets spécifiés sur les documents. La base de connaissances est mise en oeuvre par une interface utilisateur évolutive, conviviale, cohérente à toutes les applications intégrées dans le système d'aide et dotée d'un modèle de l'utilisateur. Utilisant la multi-modalité des interfaces à

manipulation directe (texte, graphique...) et des techniques graphiques associées à des outils de désignation (souris, menus, fenêtrage...), cette interface met à la disposition de l'utilisateur toutes les différentes fonctionnalités décrites plus haut. Son adéquation à l'utilisateur est rendue possible grâce à une modélisation des connaissances et des traces d'activité de l'utilisateur : chemins parcourus, annotations... (Hochon et al., 92a).

Le système d'aide permet de guider l'utilisateur, ce qui est un apport considérable par rapport aux systèmes hypertextes existants ; il lui permet aussi d'intégrer ses propres documents.

## **5. CONCLUSIONS ET PERSPECTIVES**

Le modèle de système d'aide intelligente que nous avons présenté dans ce papier peut être intégré dans de nombreux environnements (environnement de programmation, systèmes d'exploitation...) et utilisé dans de nombreuses applications informatiques (langages de programmation, générateurs de systèmes-experts, ateliers logiciels, traitements de texte...). Une application dans le domaine des environnements de programmation a été réalisée. Actuellement nous développons une extension de ce modèle au texte, à l'image et au son (par la prise en compte des annotations imagées ou vocales) dans la réalisation d'un atelier de lecture active.

L'originalité de ce système vient d'une part du fait qu'il s'intègre parfaitement dans un environnement informatique, et d'autre part du fait qu'il soit un générateur basé sur les pratiques annotatives d'un auteur ; son interface homme-machine pour la consultation de documents prend en compte le profil de l'utilisateur afin de s'adapter à celui-ci : l'utilisateur n'a donc pas beaucoup d'effort à faire pour s'adapter au système d'aide !

Le système utilise un modèle hypertexte regroupant texte informel, connaissance logico-sémantique et typo-dispositionnelle, ce qui permet une grande rapidité d'accès à l'information. Par contre, la conception d'une base de connaissances pour une application donnée nécessite une organisation qui demande beaucoup plus d'effort que dans la rédaction de documents papier compte tenu de l'aspect non linéaire de cette base documentaire. Ici, il s'agit de penser comme l'utilisateur et non comme le concepteur de l'application ! La communication par voie de textes met en jeu des aspects originaux ne pouvant pas être simplement dérivés de la communication orale ou non langagière (ie : graphique, gestuelle...). L'extension de l'informatisation à l'ensemble des opérations supportées par des textes (exemple : création d'un cours à partir de documents et livres divers) engendre un renouvellement majeur pour la réflexion théorique ; elle constitue d'autre part un contexte d'expérimentation inédit dans un vaste terrain d'applications directement parties prenantes dans les technologies cognitives.

Au niveau des perspectives, nous envisageons d'étendre les fonctionnalités d'appropriation du système en généralisant la notion d'appropriation dynamique : indexation automatique, parseurs, création automatique d'annotations, exécution de ces annotations... ainsi que les fonctionnalités de guidage en utilisant des modèles théoriques de l'utilisateur.

Ces recherches devraient permettre de concevoir des outils avancés pour la consultation intelligente de grandes bases documentaires : manuels intelligents, livres, dictionnaires et encyclopédies électroniques, documents techniques (Evrard et al., 90) et donc répondre à la problématique de la lecture assistée par ordinateur.

## 6. BIBLIOGRAPHIE

- (Catlin et al., 89) : Catlin T., Bush P., Yankelovich N., *Internote : Extending a hypermedia framework to support annotative collaboration*, Hypertext'89 Proc., Pittsburg Penn, ACM Press, Novembre 1989.
- (Daniel-Vatonne, 90) : Daniel-Vatonne M.C., *Hypertextes : des principes communs et des variations*, TSI, Vol. 9, Numéro 6, 1990, 475-492.
- (DeRose, 89) : DeRose S.J., *Expanding the notion of links*, Hypertext'89 Proc., Special Issue SIGCHI Bulletin, November 1989, 249-257.
- (Eurêka, 89) : *Eurêka : Proposal for the development of a European Lisp System (ELSY)*, Version 6, January 1989.
- (Evrard et al., 90) : Evrard F., Pascuel E., Virbel J., *Outils personnalisés de Modélisation de l'utilisateur dans la gestion électronique de documents*, Rapport interne IRIT, Toulouse, Octobre 1990.
- (Frasson, 91) : Frasson C., *Systèmes tutoriels intelligents : état et perspectives en Amérique du Nord*, Génie Educatif, n° 1, Mai 1991.
- (Guez, 87) : Guez S., *Représentation des connaissances pour un système d'aide intelligent*, Les Systèmes Experts et leurs Applications, Avignon 87, EC2 Eds, 1987.
- (Hochon, 92a) : Hochon JC., *MAYDAY : Un système générique d'aide intelligente à l'écran*, rapport ADERMIP, Toulouse, Février 1992.
- (Hochon, 92b) : Hochon JC., *Un modèle de systèmes d'aide intelligente pour un environnement informatique*, Actes du 1er Colloque Africain sur la Recherche en Informatique CARI'92, Yaoundé, Cameroun, Octobre 1992.
- (Hochon, 92c) : Hochon JC., *Une nouvelle génération d'aides intelligentes à l'écran : une architecture et un modèle*, Cinquièmes journées internationales Le Génie Logiciel et ses Applications TOULOUSE 92, Eds Jean-Claude Rault EC2, Toulouse, Décembre 1992.
- (Hochon et al., 92a) : Hochon JC., Tayrac P., Evrard F., *MAYDAY : Un système d'aide intelligente pour des applications informatiques*, Actes du congrès INFORSID 92, Clermont-Ferrand, Mai 1992.
- (Hochon et al., 92b) : Hochon JC., Tayrac P., Evrard F., *MAYDAY : Un générateur d'aide intelligente pour des systèmes de commandes*, Actes du congrès ERGO-IA'92, Biarritz, Octobre 1992.
- (Jourdan de La Passardière, 90) : Jourdan de La Passardière B., *Apports des nouvelles technologies de l'Informatique à l'EAO*, rapport de recherche MASI 90.35, Juin 1990.

- (Katzeff, 90) : Katzeff C., *System demands on Mental Models for a Fulltext Database*, International Man-Machine Studies, 32, 1990.
- (Kearsley, 88) : Kearsley, G., *Online Help Systems : design and implementation*, Human Computer Interaction series, Eds Ablex Publishing Corporation, 1988.
- (Kerr, 90) : Kerr S., *Wayfinding in an Electronic Database : the Relative Independence of Navigational Cues vs Mental Models*, Information Processing & Management, 26, 4, 1990.
- (Lamport, 86) : Lamport L., *A document preparation system : LaTeX user's guide & reference manual*, Addison-Wesley Publishing Company, 1986.
- (Meirieu, 89) : Meirieu P., *Apprendre... oui, mais comment*, E.S.F., 1989.
- (Nicaud et al., 88) : Nicaud J.F., Vivet M., *Les tuteurs intelligents : réalisations et tendances*, TSI vol 7 - n° 1 (numéro spécial : Applications de l'Informatique à la formation), Dunod-AFCET, 1988.
- (Richard, 83) : Richard J.F., *Logique du fonctionnement et logique de l'utilisation*, rapport de recherche n° 202, INRIA, Avril 1983.
- (Shirk, 88) : Shirk H.N., *Technical Writers as Computer Scientists : the challenge of Online Documentation*, Text, ConText and HyperText : writing with and for the computer, Edward Barret Eds, The MIT Press, 1988.
- (Virbel et al., 92) : Virbel J., Evrard F., Pascual E., *Langage d'annotation*, Rapport final du contrat de recherche 9007732, Conseil régional Midi-Pyrénées, Toulouse, Juin 1992.
- (Zellweger, 89) : Zellweger P.T., *Scripted Documents : a Hypermedia Path Mechanism*. Prec. Hypertext'89, special issue SIGCHI Bulletin, 1989, pp 1-14.