

Université de Montréal

# **UMAKE : Adaptation et recommandation d'outils d'aide d'un quiz pour l'auto-évaluation**

par

Moez Mabrouk

Département d'Informatique et de Recherche Opérationnelle  
Faculté des Arts et des Sciences

Mémoire présenté à la Faculté des études supérieures  
en vue de l'obtention du grade de Maîtrise ès Sciences  
en Informatique

juin, 2006

© Moez Mabrouk, 2006



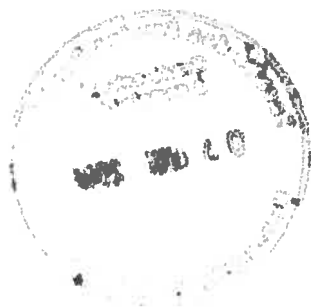
QA

76

U54

2006

V. 038



## AVIS

L'auteur a autorisé l'Université de Montréal à reproduire et diffuser, en totalité ou en partie, par quelque moyen que ce soit et sur quelque support que ce soit, et exclusivement à des fins non lucratives d'enseignement et de recherche, des copies de ce mémoire ou de cette thèse.

L'auteur et les coauteurs le cas échéant conservent la propriété du droit d'auteur et des droits moraux qui protègent ce document. Ni la thèse ou le mémoire, ni des extraits substantiels de ce document, ne doivent être imprimés ou autrement reproduits sans l'autorisation de l'auteur.

Afin de se conformer à la Loi canadienne sur la protection des renseignements personnels, quelques formulaires secondaires, coordonnées ou signatures intégrées au texte ont pu être enlevés de ce document. Bien que cela ait pu affecter la pagination, il n'y a aucun contenu manquant.

## NOTICE

The author of this thesis or dissertation has granted a nonexclusive license allowing Université de Montréal to reproduce and publish the document, in part or in whole, and in any format, solely for noncommercial educational and research purposes.

The author and co-authors if applicable retain copyright ownership and moral rights in this document. Neither the whole thesis or dissertation, nor substantial extracts from it, may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms, contact information or signatures may have been removed from the document. While this may affect the document page count, it does not represent any loss of content from the document.

Université de Montréal  
Faculté des études supérieures

Ce mémoire intitulé :

UMAKE : Adaptation et recommandation d'outils d'aide d'un quiz pour l'auto-évaluation

Présenté par :  
Moez Mabrouk

a été évalué par un jury composé des personnes suivantes :

Claude Frasson, président-rapporteur  
Esma Aïmeur, directrice de recherche  
Philippe Langlais, membre du jury

Mémoire accepté le 15 juin 2006

## Résumé

De nos jours, plusieurs systèmes basés sur le Web ont fait leur apparition, tels que les systèmes éducationnels sur Internet. Ces systèmes offrent non seulement un environnement d'apprentissage permettant d'acquérir des connaissances à distance, mais ils s'assurent également qu'elles ont été bien assimilées en ayant recours à des tests ou des quiz permettant de les évaluer. Parfois, ces tests offrent une aide pour que l'apprenant comprenne la question en cours mais l'aide proposée est identique pour tous. Nous présentons ici UMAKE, *User Modeling For Adapted Knowledge Evaluation*, un système qui assiste l'apprenant lors de son auto-évaluation en lui recommandant une aide personnalisée afin de lui permettre de mieux comprendre, et ainsi de mieux répondre à la question qui lui est posée. UMAKE recommande l'aide la plus bénéfique à l'apprenant selon son style d'apprentissage (auditif, visuel et kinesthésique). Pour ce faire, UMAKE se base sur « la commutation », une technique de recommandation hybride qui utilise le filtrage collaboratif et le filtrage à base de connaissances. De plus, des technologies d'hypermédia adaptatif, telles que l'adaptation de la navigation et l'adaptation de la présentation, sont également employées.

**Mots-clés :** auto-évaluation de l'apprenant, recommandation d'aide personnalisée, filtrage collaboratif, filtrage à base de connaissances, adaptation de la navigation, adaptation de la présentation.

## Abstract

Nowadays, many web-based systems have emerged, namely educational ones. Such systems not only offer a learning environment for distance knowledge acquisition but also evaluate the learner to ensure that the knowledge has been well assimilated using tests or quizzes. In an effort for the learner to understand the question at hand, few systems can sometimes propose helps during the test. However, the offered help is not personalized and thus the same for all the learners. We present UMAKE, **U**ser **M**odeling **F**or **A**dapted **K**nowledge **E**valuation, a system to assist the learner during his auto-evaluation process by recommending him personalized helps to fully understand the questions. UMAKE recommends the most beneficial help based on the learner's learning style (auditory, visual and kinesthetic). In order to do so, UMAKE employs switching, a hybrid recommendation technique based on both collaborative and knowledge-based filtering. Moreover, adaptive hypermedia technologies such as adaptive navigation support and adaptive presentation are also used.

**Keywords:** Learner's auto-evaluation, personalized help recommendation, collaborative filtering, knowledge-based filtering, adaptive navigation support, adaptive presentation.

# Table des matières

<b>Chapitre 1 : Introduction</b> .....	1
1.1. Notre approche .....	2
1.2. Plan du mémoire.....	5
<b>Chapitre 2 : Hypermédia adaptatif</b> .....	6
2.1. Généralités .....	6
2.2. Adaptabilité et adaptativité.....	7
2.2.1. Adaptabilité .....	8
2.2.1. Adaptativité .....	8
2.3. Modèle de référence pour les applications d’hypermédia adaptatif.....	9
2.4. Classification des systèmes d’hypermédia adaptatif .....	10
2.4.1. Domaines d’application.....	11
2.4.2. Critères d’adaptation.....	15
2.4.3. Technologies d’adaptation.....	17
2.4.3.1. Adaptation de la présentation .....	18
a) Texte conditionnel .....	19
b) Variantes .....	20
c) Texte « élastique ».....	23
d) Techniques de génération de langage naturel.....	26
2.4.3.2. Adaptation de la navigation.....	29
a) Annoter les liens .....	29
b) Trier les liens .....	35
c) Guider l’apprenant.....	37
d) Cacher les liens.....	39
e) Ajouter des liens dynamiques.....	42
<b>Chapitre 3 : Évaluation de l’apprenant</b> .....	44
3.1. Généralités .....	44
3.2. Cycle de vie d’une question.....	45
3.2.1. L’étape de la préparation .....	46
3.2.2. L’étape de la livraison .....	48

3.2.3.	L'étape de l'évaluation .....	51
3.3.	Présentation de quelques systèmes d'évaluation .....	53
3.3.1.	Le système ALE .....	53
3.3.1.1.	La préparation et le stockage des tests.....	54
a)	Les tests de choix.....	54
b)	Les textes à trous .....	54
c)	Les questions d'association .....	55
3.3.1.2.	La livraison .....	55
3.3.1.3.	L'évaluation et le feedback.....	55
3.3.2.	Le système SAPZAS .....	56
3.3.2.1.	La préparation.....	56
3.3.2.2.	La livraison .....	57
3.3.2.3.	L'évaluation.....	58
3.3.3.	Le système AHyCo.....	58
3.3.3.1.	La préparation.....	59
3.3.3.2.	La livraison .....	59
3.3.3.3.	L'évaluation.....	60
3.3.4.	Le système SIETTE.....	60
3.3.4.1.	La préparation.....	61
3.3.4.2.	La livraison .....	62
3.3.4.3.	L'évaluation.....	62
<b>Chapitre 4 :</b>	<b>Les systèmes de recommandation.....</b>	<b>63</b>
4.1.	Introduction .....	63
4.2.	Les techniques de recommandation.....	64
4.2.1.	Le filtrage collaboratif.....	65
4.2.1.1.	Algorithmes fondés sur la mémoire ( <i>memory-based</i> ) .....	66
4.2.1.2.	Algorithmes à base de modèles ( <i>model-based</i> ) .....	68
4.2.1.3.	Les limites du filtrage collaboratif.....	68
4.2.2.	Le filtrage basé sur le contenu.....	69
4.2.3.	Le filtrage démographique.....	70
4.2.4.	Le filtrage basé sur l'utilité.....	70
4.2.5.	Le filtrage à base de connaissances .....	71



4.3.	Les systèmes hybrides .....	71
4.3.1.	Pondération (Weighted).....	71
4.3.2.	Commutation (Switching) .....	71
4.3.3.	Technique mixte (Mixed).....	72
4.3.4.	Combinaison de caractéristiques (Features combination).....	72
4.3.5.	Cascade (Cascade).....	72
<b>Chapitre 5 : Conception et méthodologie</b> .....		<b>74</b>
5.1.	Objectifs.....	74
5.2.	Approche de UMAKE .....	74
5.2.1.	Fonctionnement général .....	74
5.2.2.	Fonctions principales accessibles aux utilisateurs.....	76
5.2.3.	Techniques de recommandation .....	77
5.3.	Méthodologie de UMAKE .....	77
5.3.1.	Style d'apprentissage .....	80
5.3.2.	Évaluation des aides .....	81
5.4.	Architecture de UMAKE.....	84
5.4.1.	Module de gestion des profils.....	87
5.4.2.	Module de gestion des réponses .....	87
5.4.3.	Recommandeur d'aide .....	88
5.4.3.1.	Filtrage collaboratif .....	89
5.4.3.2.	Filtrage à base de connaissances .....	99
5.4.4.	Base des profils.....	104
5.4.4.1.	Données statiques .....	104
5.4.4.2.	Données dynamiques .....	106
5.4.5.	Structuration du domaine d'application .....	110
5.4.6.	Base des concepts et questions associées .....	110
5.5.	Comparaison.....	112
5.6.	Conclusion.....	117
<b>Chapitre 6 : Implémentation et validation</b> .....		<b>118</b>
6.1.	Implémentation de UMAKE .....	118
6.1.1.	Les Servlets et les pages JSP (Java Server Pages).....	118

6.1.1.1.	Architecture client-serveur .....	119
6.1.1.2.	Les servlets .....	119
6.1.1.3.	Technologie XML .....	122
6.2.	Domaine d'application .....	122
6.3.	Environnement d'un utilisateur .....	123
6.3.1.	Page d'accueil de UMAKE .....	123
6.3.2.	Enregistrement de l'utilisateur.....	125
6.3.3.	Outils membres.....	127
6.3.3.1.	Statistiques globales .....	128
6.3.3.2.	Questionnaire du style d'apprentissage .....	129
6.3.3.3.	Mot de passe .....	129
6.3.3.4.	Voisinage .....	130
6.3.3.5.	Aide .....	131
6.3.4.	Auto-évaluation de l'utilisateur .....	133
6.4.	Validation de UMAKE .....	144
6.4.1.	Mise en contexte .....	144
6.4.2.	Style d'apprentissage .....	146
6.4.3.	Qualité de la recommandation.....	150
6.4.3.1.	Population associée à la technique CF .....	151
6.4.3.2.	Population associée à la technique KB.....	151
6.4.3.3.	Population associée à la technique HY.....	152
6.4.3.4.	Comparaison des trois techniques .....	153
6.4.4.	Idée de la recommandation d'aide : utile ou non?.....	154
6.4.5.	Feedback des utilisateurs .....	155
6.4.6.	Conclusion .....	157
<b>Chapitre 7 : Conclusion .....</b>		<b>158</b>
<b>Bibliographie.....</b>		<b>162</b>

## Liste des tableaux

<b>Tableau 3.1:</b> Les étapes du cycle de vie d'une question [Brusilovsky & Miller 99] .....	46
<b>Tableau 4.1 :</b> Les différentes techniques de recommandation [Burke 02] .....	65
<b>Tableau 5.1 :</b> Matrice des votes des apprenants pour les différentes aides des quatre questions (premier exemple) .....	93
<b>Tableau 5.2 :</b> Matrice des votes des apprenants pour les différentes aides des quatre questions (deuxième exemple) .....	98
<b>Tableau 5.3 :</b> Exemple de données stockées dans le profil apprenant pour les trois aides d'une question i .....	101
<b>Tableau 5.4 :</b> comparaison des systèmes.....	113
<b>Tableau 6.1 :</b> Nombre d'utilisateurs par technique de filtrage employée .....	145
<b>Tableau 6.2 :</b> Nombre de fois où les utilisateurs ont accepté leur style d'apprentissage évolué.....	149
<b>Tableau 6.3 :</b> Les recommandations suivies pour la population associée à la technique CF .....	151
<b>Tableau 6.4 :</b> Les recommandations suivies pour la population associée à la technique KB.....	152
<b>Tableau 6.5 :</b> Les recommandations suivies pour la population associée à la technique HY .....	153
<b>Tableau 6.6 :</b> La comparaison de la recommandation dans les trois techniques.....	153
<b>Tableau 6.7 :</b> L'utilisation de l'aide et son impact sur la réponse donnée .....	154
<b>Tableau 6.8 :</b> Le feedback des utilisateurs .....	156

## Liste des figures

<b>Figure 2.1</b> : La structure des applications d'hypermédia adaptative [Wu et al. 01].....	10
<b>Figure 2.2</b> : Le continuum des systèmes d'hypermédia adaptatif [Brusilovsky 96] .....	14
<b>Figure 2.3</b> : Les techniques d'adaptation dans les systèmes d'hypermédia adaptatif [Brusilovsky 01] .....	18
<b>Figure 2.4a</b> : La carte de Sienne en mode graphique dans AVANTI [Fink et al. 98].....	22
<b>Figure 2.4b</b> : La « carte de Sienne » pour les personnes ayant des troubles de vision dans AVANTI [Fink et al. 98].....	22
<b>Figure 2.5</b> : Une page du système PUSH .....	24
<b>Figure 2.6</b> : Exemple de procédure de dépannage dans ADAPTS.....	25
<b>Figure 2.7</b> : Une page adaptative du cours « Rückenschule » [Kobsa et al. 01].....	26
<b>Figure 2.8</b> : Une présentation de texte à un utilisateur s'intéressant aux styles de bijoux [Kobsa et al. 01] .....	27
<b>Figure 2.9</b> : Une présentation de texte à un visiteur n'ayant aucun intérêt spécifique [Kobsa et al. 01] .....	28
<b>Figure 2.10</b> : L'annotation dans ISIS-TUTOR [Brusilovsky 04].....	30
<b>Figure 2.11</b> : L'annotation des liens dans ELM-ART [Brusilovsky 04].....	32
<b>Figure 2.12</b> : L'annotation des liens dans InterBook. En outre, la fenêtre de la recommandation de l'aide (à gauche) utilise la génération (ajout dynamique) et le tri des liens [Brusilovsky 04].....	33
<b>Figure 2.13</b> : L'annotation dans ACE [Specht & Oppermann 98].....	34
<b>Figure 2.14</b> : La première image présente l'annotation des liens dans une page de ISIS-TUTOR. La deuxième présente la même page, mais en plus de l'annotation des liens, ceux qui ne sont pas prêts à être étudiés (fond gris clair avec écriture noire dans la première image) sont supprimés [Brusilovsky 04].....	41
<b>Figure 5.1</b> : Architecture générale de UMAKE.....	86
<b>Figure 5.2</b> : Exemple d'un profil utilisateur initial.....	105
<b>Figure 5.3</b> : Exemple d'un profil d'utilisateur durant une session d'auto-évaluation ....	107
<b>Figure 5.4</b> : Représentation hiérarchique du domaine d'application.....	110

<b>Figure 5.5</b> : Exemple de la structuration d'une question associée à un concept donné .	111
<b>Figure 6.1</b> : La communication dans l'architecture client-serveur.....	119
<b>Figure 6.2</b> : Le cycle de vie d'une servlet .....	120
<b>Figure 6.3</b> : Le cycle de vie de JSP .....	121
<b>Figure 6.4</b> : Illustration d'une partie du domaine d'application.....	122
<b>Figure 6.5</b> : La page d'accueil de UMAKE (sans détails).....	123
<b>Figure 6.6</b> : La page d'accueil de UMAKE (avec détails) .....	124
<b>Figure 6.7</b> : La première étape de l'enregistrement.....	125
<b>Figure 6.8</b> : La deuxième étape de l'enregistrement .....	126
<b>Figure 6.9</b> : Le style d'apprentissage déterminé par UMAKE.....	127
<b>Figure 6.10</b> : Les apprenants visuels .....	127
<b>Figure 6.11</b> : La page principale de UMAKE .....	128
<b>Figure 6.12</b> : L'outil <i>Overall Summary</i> .....	129
<b>Figure 6.13</b> : L'outil <i>Change Password</i> .....	130
<b>Figure 6.14</b> : L'outil <i>Neighbourhood</i> .....	131
<b>Figure 6.15</b> : L'outil <i>Help</i> .....	132
<b>Figure 6.16</b> : Le choix du nombre des questions dans UMAKE.....	133
<b>Figure 6.17</b> : Le tutorial pour l'aide visuelle.....	134
<b>Figure 6.18</b> : Exemple de questions pour le concept « tri rapide » .....	135
<b>Figure 6.19</b> : L'évaluation des aides dans UMAKE (évaluation nécessaire).....	136
<b>Figure 6.20</b> : L'évaluation des aides dans UMAKE (évaluation non nécessaire).....	137
<b>Figure 6.21</b> : La page des résultats et des statistiques .....	138
<b>Figure 6.22</b> : La solution, sans détails, à une question répondue incorrectement .....	139
<b>Figure 6.23</b> : La solution, avec détails, à une question répondue incorrectement (la première raison pour l'utilisation du filtrage à base de connaissances) .....	140
<b>Figure 6.24</b> : La solution, avec détails, à une question répondue incorrectement (la deuxième raison pour l'utilisation du filtrage à base de connaissances).....	141
<b>Figure 6.25</b> : La solution, avec détails, à une question répondue correctement (la troisième raison pour l'utilisation du filtrage à base de connaissances).....	142
<b>Figure 6.26</b> : La solution, avec détails, à une question répondue correctement (l'utilisation du filtrage collaboratif) .....	143

<b>Figure 6.27</b> : La fin de l'auto-évaluation.....	144
<b>Figure 6.28</b> : Évolution du style d'apprentissage pour la population associée à la technique KB.....	146
<b>Figure 6.29</b> : Évolution du style d'apprentissage pour la population associée à la technique HY.....	147
<b>Figure 6.30</b> : Évolution du style d'apprentissage en moyenne pour les trois populations .....	148
<b>Figure 6.31</b> : L'utilisation de l'aide et son impact sur la réponse donnée.....	155
<b>Figure 6.32</b> : Le formulaire du <i>feedback</i> des apprenants.....	156

## Remerciements

Je voudrai en premier lieu adresser mes sincères remerciements à ma directrice de recherche, professeure Esma Aïmeur : elle a su me guider lors de l'accomplissement de mon projet, m'a facilité la tâche en fournissant de précieux conseils et a toujours été disponible pour moi, avec des encouragements permanents.

Par cet ouvrage, je tiens également à témoigner toute mon affection et mon attachement à

### Ma famille :

Ma chère mère Bornia Mabrouk qui m'a toujours apporté soutien et réconfort, et sans qui ce travail n'aurait pas été possible.

Mon cher père Mohamed Mabrouk qui a toujours été là pour moi.

Ma sœur Abir et sa mignonne petite fille Assil ainsi que ma sœur Maroua.

Mes tantes Dalila, Ibtissam, Ferdaws et Jamila qui m'ont beaucoup soutenu et encouragé.

Je vous remercie de votre immense aide et de votre présence à mes côtés. Que Dieu vous protège et vous garde en bonne santé.

### Mes amis :

Hicham Hage, Pierre Chalfoun, Soumaya Chaffar et Anita Saleman, qui se sont avérés d'une aide sans égale pour l'accomplissement de mon projet de maîtrise. Je tiens particulièrement à remercier Mohamed El Golli, Anis Mansour, Walid Aloui, Faiza Kacem, Sirine Tlili et Anissa Aissa qui m'ont fourni soutien et amitié jour après jour.

### Les membres du laboratoire Héron :

Sébastien Gams, Flavien Serge Mani Onana, Emmanuel Blanchard, Hicham Hage, Pierre Chalfoun, Soumaya Chaffar, Anita Saleman, Narimel Bendakir, Zakia Lakhli et Sihem Benlizidia, qui m'ont apporté un grand soutien et m'ont fourni un environnement de travail propice.

Je remercie également les membres de l'honorable jury qui ont bien voulu évaluer mon travail, les professeurs Claude Frasson et Philippe Langlais.

Un remerciement spécial va à ma chère Tasnouma pour son support moral, sa patience, son encouragement constant et son excellente cuisine tout au long de cette aventure.

## Chapitre 1 : Introduction

Au cours de ces dernières années, l'Internet a pris une dimension vraiment considérable. Elle évolue avec la masse d'informations mises en ligne qui ne cesse d'augmenter au fil des jours. Nous voyons de plus en plus de systèmes disponibles sur le *net* dont les plus évolués offrent maintenant des services personnalisés à l'utilisateur. En effet, la personnalisation est devenue un aspect important qui prend de plus en plus d'ampleur. Parmi ces systèmes, nous pouvons citer les systèmes de recommandation comme Amazon<sup>1</sup> qui suggère à l'utilisateur un item susceptible de lui plaire en se basant sur les items déjà appréciés par ce dernier, et aussi les systèmes éducationnels basés sur le Web comme ELM-ART [Weber & Brusilovsky 01], qui permet à l'utilisateur d'apprendre le langage de programmation LISP d'une façon personnalisée, et ce en lui suggérant, à partir d'un cours donné, un chemin optimal à travers les différents cours proposés pour atteindre son objectif. En général, les systèmes éducationnels basés sur le Web offrent à l'utilisateur un environnement d'apprentissage qui lui permet d'acquérir plusieurs connaissances dans un domaine spécifique. Ils se basent sur des technologies d'hypermédia adaptatif (voir chapitre 2) pour réaliser la personnalisation [Burke, Mobasher & Bhaumik 2005; Freyne & Smyth 2005; Suryavanshi *et al.* 05]. Mais au fil des années, il y a un autre aspect qui est devenu important et qui est, de plus en plus, pris en compte. Il s'agit là de l'aspect « évaluation ». En effet, les systèmes offrant un environnement d'apprentissage, ne se contentent pas seulement de faire parvenir la connaissance à l'apprenant, mais ils essaient aussi de s'assurer que ce dernier a bien assimilé et compris cette information. Après lui avoir présenté la connaissance, ils essaient généralement de l'évaluer à travers des tests et des *quiz*<sup>2</sup> (voir chapitre 3) et se basent sur la performance réalisée dans ces tests pour juger s'il a bien appris, assimilé et compris la connaissance étudiée ou non, c'est-à-dire s'il a, bel et bien, atteint son objectif ou non [Brusilovsky & Miller 99; Pesin 03; Pirotte 02].

---

<sup>1</sup> [http : //www.amazon.com](http://www.amazon.com)

<sup>2</sup> Quiz veut aussi dire test.



Dans les systèmes les plus évolués, ces tests ne sont pas statiques. Ils sont générés dynamiquement, adaptés aux niveaux de connaissances des apprenants (ils portent sur un concept donné) et se basent essentiellement sur le paramètre « difficulté » pour sélectionner les questions. En effet, la prochaine question proposée à l'apprenant dépend de la réponse donnée à la question courante : s'il répond incorrectement à la question courante, la question suivante sera plus facile, sinon elle sera plus difficile [Weber & Brusilovsky 01; Guzman *et al.* 05 ; Gejji 01].

Dans la section suivante, nous allons définir la problématique concernant cet aspect et présenter notre approche.

## 1.1. Notre approche

Dans cette partie, nous nous basons sur un petit scénario afin de définir la problématique et introduire notre approche : imaginons une société appelée *3DWorld* qui développe des applications 3D et *Alice* une employée de cette société. Elle fait partie des premières personnes à y travailler depuis sa fondation. Mais au fil des années, *3DWorld* s'est forgé un grand nom dans le domaine, a connu la réussite, a grandi et il y a de plus en plus de grands projets qui s'offrent à elle. *3DWorld* a commencé à faire du remaniement dans son personnel afin de maintenir cette réussite et être à la hauteur de ses projets: recruter du personnel qualifié avec des critères sélectifs stricts, laisser les personnes jugées capables d'être performantes dans les grands projets d'avenir et remercier les autres. Toutefois, durant ce processus de remaniement, *3DWorld* se montre reconnaissante envers l'ancien personnel, jugé non capable de donner le plus dans l'avenir, et leur donne encore une chance de continuer à travailler dans la société : ceux qui veulent faire partie des projets futurs de la société, doivent suivre une formation avancée en infographie afin d'approfondir leurs connaissances et d'obtenir un certificat. Le nombre de places étant vraiment limité, 5 employés seulement vont rester travailler au sein de la société. *Alice* fait partie de ce type de personnel. Elle a des connaissances en infographie qui lui permettaient de participer au développement des petites applications, mais ces connaissances ne lui permettront sûrement pas de participer convenablement dans les grands projets. Elle veut saisir l'opportunité pour garder son travail, alors elle

cherche sur Internet un organisme qui offre une formation ou des cours avancés dans ce domaine et trouve l'université  $x$  de renommée internationale qui permet à ses étudiants de faire des formations à distance. Chaque formation contient une multitude de cours en ligne et permet de passer des tests d'évaluation (ou d'auto-évaluation) à la fin de chaque cours. Parfois, un étudiant ne peut passer au cours suivant qu'après avoir réussi les tests du cours actuel. En effet, il y a certains cours qui demandent de maîtriser un ou plusieurs pré-requis avant d'être étudiés. Par exemple, le cours gradué d'infographie qui concerne *Alice*, nécessite des connaissances en structures de données et demande comme préalable ce cours. *Alice* se doit donc de le réussir afin d'accéder au dernier cours avancé de sa formation. Pour ce faire, elle doit passer le test d'évaluation du cours des structures des données.

Après le test, *Alice* se montre confiante : d'une part, elle a bien révisé pour ce test et a couvert toute la matière du cours, et d'autre part elle est persuadée qu'elle a bien répondu à toutes les questions du test et qu'elle va le réussir. Mais à sa grande stupéfaction, le résultat du test est totalement différent : elle ne l'a pas réussi et elle doit reprendre le cours parce qu'elle « n'a pas les connaissances minimales nécessaires pour le réussir » et « ne l'a pas bien assimilé ». Quand *Alice* a consulté la solution relative à toutes les questions, elle s'aperçoit qu'elle *connaît* toutes les réponses, mais qu'elle a simplement *mal compris* les questions.

Ce premier scénario nous montre comment *Alice* risque de perdre le travail qu'elle aime tant parce qu'elle a été *mal évaluée*<sup>3</sup> et que parce qu'elle n'a pas été *assistée* durant son évaluation. Elle va perdre du temps à refaire le cours et les 5 places proposées par *3DWorld* ne seront probablement plus vacantes. De même, la société risque de perdre une employée pouvant jouer un grand rôle dans la réussite des grands projets car elle a beaucoup de potentiel.

---

<sup>3</sup> Nous voulons dire par personne mal évaluée, une personne ayant les connaissances nécessaires pour répondre correctement à une question donnée, mais qu'elle a répondu incorrectement car simplement elle ne l'a pas comprise.

Envisageons le scénario différemment. Cette fois-ci, la plateforme de l'université  $x$  utilise un logiciel interne pour l'évaluation et l'auto-évaluation des apprenants, qui est UMAKE. Tous les apprenants qui utilisent UMAKE pour la première fois doivent d'abord s'enregistrer (voir chapitre 5, section 5.2.1). A la fin de la phase de l'enregistrement, UMAKE détermine le style d'apprentissage de l'apprenant grâce à un questionnaire psychotechnique [URL 1]. *Alice* se connecte à UMAKE et passe à la phase d'évaluation pour faire le test des structures des données. Pour chaque question du test, elle se voit proposer trois aides qui correspondent aux trois principaux styles d'apprentissage (auditif, visuel, kinesthésique) [URL 1]. Le test est présenté sous forme d'une question à la fois afin d'alléger la page, vu la richesse du matériel éducationnel présenté (le texte de la question, les réponses proposées, les trois types d'aide associés...). Afin de permettre à *Alice* de rester concentrée au maximum sur chaque question et maximiser les chances d'avoir des résultats significatifs, UMAKE lui présente les aides adaptées aux différents styles d'apprentissage. Il lui *recommande* aussi l'aide appropriée afin de lui permettre de mieux comprendre la question. Par exemple, supposons qu'elle utilise pour la première fois UMAKE, qu'elle se trouve devant la question *quelle est la définition du quicksort?* et qu'*Alice* assimile mieux une information en la visualisant, UMAKE lui recommandera alors l'aide visuelle qui fait tourner devant elle l'algorithme du quicksort sur un jeu de données. *Alice* donnera probablement la bonne réponse *choisir un élément du tableau comme pivot, partitionner les éléments restants par rapport au pivot et trier récursivement les partitions*. Suite à ce raisonnement, elle finira par réussir le test, obtenir son certificat sans perdre de temps et garder son travail.

Cette deuxième variante du scénario montre que grâce à UMAKE, *Alice* a été *bien assistée* lors de son évaluation et *a été évaluée équitablement*. De plus, elle a eu de meilleures chances pour finir à temps sa formation et garder son travail. Afin de développer UMAKE, nous avons focalisé nos efforts sur les moyens permettant à l'apprenant de comprendre la question posée dans un contexte d'évaluation donné grâce aux recommandations. A chaque question sont associées trois aides différentes *adaptées* aux trois principaux styles d'apprentissage : auditif, visuel, kinesthésique (voir chapitre 5,

section 5.3.1). UMAKE recommande une aide appropriée à l'apprenant selon une technique hybride : *la commutation* basée sur le filtrage collaboratif et le filtrage à base de connaissances. Quand l'apprenant utilise une aide donnée, il est souvent requis de donner une évaluation à cette aide. La première technique de filtrage [Breese *et al.* 98; O'Sullivan *et al.* 02] se base sur les différentes évaluations (votes) données par les apprenants aux différentes aides, tandis que la deuxième technique [Burke 02] infère les préférences de l'apprenant courant en fonction de son comportement et de sa performance pour lui recommander une aide appropriée (voir section 5.4.3 dans la section 5).

Concrètement, notre système peut être défini comme étant un système de recommandation et d'adaptation d'outils d'aide d'un quiz pour l'auto-évaluation.

## 1.2. Plan du mémoire

Ce mémoire est organisé de la façon suivante : dans le deuxième chapitre, nous présentons les différentes technologies de l'hypermédia adaptatif, plus particulièrement l'adaptation du support de navigation et l'adaptation de la présentation que nous utilisons pour indiquer l'aide recommandée et pour présenter la solution des questions à l'apprenant de façon personnalisée. Le troisième chapitre décrit le cycle de vie d'une question dans les systèmes éducationnels basés sur le Web et la façon avec laquelle est réalisée l'évaluation de l'apprenant dans ces systèmes. Le quatrième chapitre propose un état de l'art sur les systèmes de recommandation, dans lequel les principales techniques de filtrage, y compris les techniques hybrides, sont abordées. Au cinquième chapitre, l'architecture de UMAKE et les différents modules qui le composent, sont décrits. Le sixième chapitre présente l'implémentation et la validation du système. Finalement, ce mémoire est clôturé par une conclusion et des perspectives futures.

Notons que dans la suite de ce mémoire, nous allons utiliser, de façon interchangeable, le mot « utilisateur » pour désigner « apprenant », le mot « test » pour désigner « quiz » et le mot « vote » pour désigner « évaluation ». De plus, l'utilisation du masculin sera privilégiée pour des fins de clarté.

## Chapitre 2 : Hypermédia adaptatif

Dans ce chapitre, nous allons essentiellement présenter les différentes techniques employées dans les systèmes d'hypermédia adaptatif. Mais nous allons d'abord passer par un aperçu de l'état de l'art de ce domaine de recherche qui est hybride et qui couvre à la fois l'hypermédia et le modèle apprenant [Brusilovsky 01].

### 2.1. Généralités

Rappelons qu'un modèle apprenant inclut normalement tous les aspects reliés au comportement et aux connaissances de l'utilisateur (nom, âge, sexe, préférences, choix faits précédemment...) et qui ont des répercussions sur ses performances et sur son apprentissage [Wenger 87]. Toutefois, il est toujours considéré comme une approximation à cause des comportements contradictoires de l'utilisateur qui peuvent être dus à la fatigue ou au manque de concentration, et à la nature de l'apprentissage humain qui est dynamique et non monotone. Ceci est expliqué par le fait que des hypothèses faites à un moment donné dans le modèle apprenant peuvent devenir incorrectes à l'avenir suite à l'acquisition de nouvelles connaissances [Cristea & De Bra 02, Dufour 98, Mitrovic *et al.* 96].

Un système d'hypermédia adaptatif construit un modèle des buts, des préférences et des connaissances de chaque utilisateur. En plus, le système utilise ce modèle et le met constamment à jour dans toute interaction avec l'utilisateur afin de s'adapter à ses besoins [Brusilovsky 01, Dara-Abrams 02]. Par exemple, une encyclopédie électronique adaptative devrait personnaliser le contenu d'un article pour augmenter les connaissances et les intérêts déjà existants de l'utilisateur [Milosavljevic 97]. Un second exemple est celui d'un musée virtuel adapte la présentation de chaque objet visité en fonction de l'itinéraire suivi par l'utilisateur dans le musée [Oberlander *et al.* 98].

Par contre, les systèmes d'hypermédia traditionnels présentent toujours les mêmes pages et le même ensemble de liens<sup>4</sup> à tous les utilisateurs. Citons comme exemple une encyclopédie électronique statique présentant à des lecteurs, de différentes connaissances et intérêts, les mêmes informations et les mêmes liens aux articles relatifs à leur recherche. Nous pouvons également mentionner un musée virtuel statique qui offre la même visite guidée et la même description à tous les visiteurs ayant différents objectifs et différentes connaissances.

D'un point de vue historique, l'année 1996 a été le tournant pour ce domaine de recherche. Nous avons pu voir de nouveaux projets réalisés, ainsi que plusieurs workshops en liaison avec l'hypermédia adaptative. Brusilovsky [Brusilovsky 01] explique ce grand progrès par deux facteurs principaux :

- Le premier est l'utilisation croissante du Web par une population très diversifiée. Ceci donne toujours un nouveau défi à relever et accélère les recherches pour pouvoir fournir des systèmes adaptables basés sur le Web, entre autres, aux connaissances de chaque personne qui les utilise.
- Le deuxième facteur est la consolidation et l'accumulation des expériences dans ce domaine. En effet, la plupart des articles publiés à partir de 1996 se basent sur les anciennes recherches et proposent donc de nouvelles techniques ou des extensions de techniques suggérées antérieurement, ce qui n'était pas le cas avant cette année. Ceci pourrait constituer une indication que l'hypermédia adaptatif a atteint plus de maturité durant cette période [Brusilovsky 01].

## 2.2. Adaptabilité et adaptativité<sup>5</sup>

Afin de bien décrire les systèmes hypermédias, il est utile de distinguer entre les équipements adaptables et les équipements adaptatifs d'un système.

---

<sup>4</sup> Le mot « lien » veut dire URL dans ce chapitre.

<sup>5</sup> Nous utilisons le mot en français « adaptativité » pour désigner le mot anglais « adaptivity ».

### **2.2.1. Adaptabilité**

Un système hypermédia adaptable est un système qui permet à l'utilisateur de fixer explicitement ses préférences ou de fournir un profil en remplissant un formulaire. L'aspect, le contenu ou la fonctionnalité de certains composants d'une page du système peuvent changer selon ces données [Cristea & De Bra 02; De Bra 98]. Le fait d'avoir un bouton pour changer les couleurs en est un petit exemple. Un autre exemple, d'ordre plus complexe, est d'offrir des cours alternatifs pour les étudiants handicapés. A cet égard, nous pouvons mentionner le fait de lire à haute voix les données ou de fournir une version avec des caractères plus gros pour les étudiants ayant une altération de la vue [Cristea & De Bra 02]. Ces informations sont enregistrées dans le modèle apprenant. Leur présentation est alors adaptée à ce modèle qui est mis à jour seulement sur une demande explicite de l'utilisateur [De Bra 98].

### **2.2.1. Adaptativité**

Un système adaptatif est un système qui construit un modèle apprenant en observant la navigation de l'utilisateur. La présentation est adaptée au modèle d'utilisateur qui est constamment mis à jour pendant que l'utilisateur consulte l'information [Dara-Abrams 02; De Bra 98].

Les auteurs de [Cristea & De Bra 02] définissent le terme adaptativité comme étant la capacité actuelle d'un système à s'adapter automatiquement à des nouvelles conditions (habituellement déduites du modèle apprenant). Comparée à l'adaptabilité, l'adaptativité représente une étape plus avancée vers l'intelligence artificielle.

Naturellement, la plupart des systèmes d'hypermédia adaptatif sont adaptatifs et adaptables pour répondre au besoin d'initialiser le modèle d'utilisateur, ou de permettre à des utilisateurs de modifier le modèle si nécessaire [Dara-Abrams 02; De Bra 98].

## 2.3. Modèle de référence pour les applications d'hypermédia adaptatif

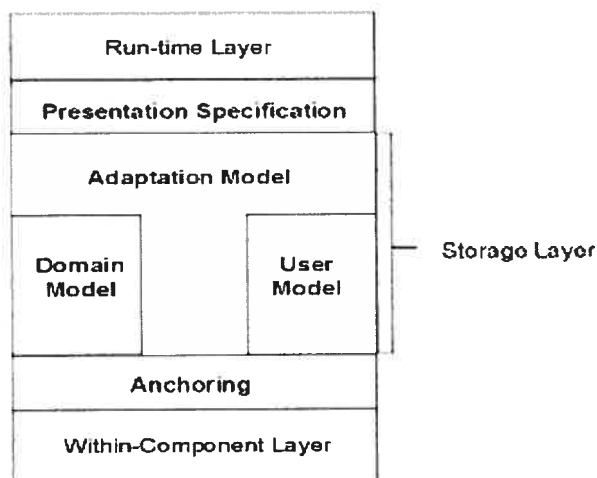
Wu *et al.* [Wu *et al.* 01] ont développé un modèle de référence pour l'architecture des applications d'hypermédia adaptatif (voir Figure 2.1). D'après ce modèle, une application adaptative comprend trois composantes essentielles [Kavcic *et al.* 02]:

- Un modèle du domaine qui représente comment l'information contenue dans cette application est structurée (en utilisant les concepts et les relations entre ces concepts) [Wu *et al.* 01]. Un exemple illustrant la structuration du domaine est [Abdel Razek *et al.* 03] où le domaine de connaissances est représenté comme une hiérarchie de concepts. Ainsi, chaque cours se décompose en quelques concepts, chaque concept se compose en cinq fragments (*background*, définitions, problèmes, exemples et exercices), chaque fragment est défini par trois morceaux (*chunks*) et chaque morceau se compose d'unités atomiques telles que du texte, des images, de l'audio et de la vidéo.
- Un modèle apprenant qui représente les préférences, les connaissances, les objectifs, l'historique de navigation et d'autres aspects pertinents de l'utilisateur (voir section 2.4.2). Les connaissances de l'utilisateur sont représentées en fonction des concepts du modèle de domaine [Dara-Abrams 02; Wu *et al.* 01].
- Un modèle d'adaptation qui permet au système de faire l'adaptation du contenu et des liens en se basant sur le modèle apprenant et le modèle du domaine. Il consiste en des règles d'adaptation définissant à la fois le processus de génération adaptative de l'information et celui de la mise à jour du modèle apprenant [Wu *et al.* 01].

Dans les applications d'hypermédia adaptatif, le *modèle apprenant* représente la relation entre l'utilisateur et le *modèle du domaine* et joue un rôle central dans l'adaptation. Il observe les interactions de l'utilisateur avec le système afin de déterminer la connaissance de ce dernier des concepts du domaine d'application. Le *modèle d'adaptation* spécifie la façon avec laquelle les connaissances de l'utilisateur modifient la



présentation des informations. Le moteur d'adaptation utilise les règles d'adaptation pour manipuler les *indices visuels des liens* et pour générer les *spécifications de présentation* [Dara-Abrams 02; Wu *et al.* 01] (pour un concept atomique, un champ booléen spécifiant si ce concept est à cacher ou à montrer selon De Bra *et al.* [De Bra *et al.* 99]).



**Figure 2.1 :** La structure des applications d'hypermédia adaptative [Wu *et al.* 01]

## 2.4. Classification des systèmes d'hypermédia adaptatif

Dans le cadre de la présente recherche, nous présentons et expliquons les principales dimensions des systèmes d'hypermédia adaptatif tel que classifiés par Brusilovsky [Brusilovsky 96] :

- Les domaines d'application : ce sont les différents domaines dans lequel un système d'hypermédia adaptatif peut être utile.
- Les critères d'adaptation : ce sont les différentes caractéristiques de l'utilisateur qui peuvent être une source d'adaptation pour le système.
- Les technologies d'adaptation : elles permettent d'adapter le contenu et les liens dans un système hypermédia aux besoins de l'utilisateur.

### 2.4.1. Domaines d'application

L'hypermédia adaptatif est utilisé dans six domaines principaux qui sont l'éducation, l'information en ligne, l'aide en ligne, la recherche d'information, l'institution et la gestion des vues personnalisées dans les espaces d'information [Brusilovsky 01].

L'hypermédia éducationnel constitue un des plus anciens et des plus populaires domaines d'application pour l'hypermédia adaptatif. La plupart des systèmes hypermédia éducationnels limitent la grandeur de l'hyperespace<sup>6</sup> en représentant un cours spécifique ou un sujet particulier à apprendre [Dara-Abrams 02]. Les connaissances des différents utilisateurs peuvent varier considérablement et la connaissance d'un seul utilisateur peut croître très rapidement. Les utilisateurs novices, ne connaissant rien du sujet, ont besoin d'aide pour naviguer et trouver leur chemin dans l'hyperespace sinon ils peuvent se perdre dans les petits hyperespaces des systèmes d'hypermédia éducationnel. Les techniques d'hypermédia adaptatif combinées au modèle apprenant sont très utiles pour résoudre ce genre de problèmes [Brusilovsky 96].

Les systèmes d'information en ligne permettent un accès référencé et adapté à l'information pour les utilisateurs avec différents niveaux de connaissances du sujet [Brusilovsky 96]. Tout comme les systèmes éducationnels, ils cherchent à satisfaire les besoins d'un grand nombre d'utilisateurs différents. Ils doivent comprendre l'objectif de l'utilisateur, ce qui est généralement difficile à déterminer – excepté dans les circonstances où l'utilisateur le spécifie explicitement [Dara-Abrams 02]. Les encyclopédies électroniques font partie de cette catégorie de systèmes. Elles peuvent tracer le comportement de navigation de l'utilisateur, déduire ses intérêts et lui offrir une liste des articles les plus pertinents [Hirashima *et al.* 98].

Les systèmes d'aide en ligne sont similaires aux systèmes d'information en ligne. Ils fournissent des informations qui sont adaptées aux besoins des différents utilisateurs. Mais ces systèmes ne sont pas indépendants. En effet, ils font partie intégrante de leurs

---

<sup>6</sup> Hyperespace est un ensemble d'informations accessible et navigable par un apprenant.

propres systèmes d'application et ont pour but d'aider les utilisateurs de ces derniers. Ces systèmes d'application sont la plupart du temps en relation avec l'informatique comme les environnements de programmation et les systèmes experts [Brusilovsky 96].

Les trois premiers domaines d'application d'hypermédia adaptatif sont les plus anciens, et donc les plus répandus. Notons que la plupart des recherches et des projets sont réalisés dans le domaine de l'éducation.

Les systèmes de recherche de documents combinent les techniques traditionnelles de recherche de documents avec les accès hypertextes aux documents à partir des mots clés. A l'aide de mesures de similarité et d'autres techniques, ces systèmes doivent calculer les liens dans l'hyperespace, contrairement aux systèmes d'information en ligne dans lesquels le concepteur fournit lui-même les liens. Un autre point qui les distingue des systèmes d'information en ligne est que leurs utilisateurs sont le plus souvent des professionnels dans différents domaines qui utilisent le système chaque jour dans leur travail avec des objectifs différents [Brusilovsky 96]. Les systèmes d'hypermédia adaptatif de recherche de documents ont pour but d'aider les utilisateurs à localiser les informations dont ils ont besoin en leur recommandant des liens et en limitant leurs choix de navigation [Dara-Abrams 02].

L'institution est un domaine d'application relativement nouveau pour l'hypermédia adaptatif. Les systèmes d'information institutionnels fournissent en ligne toutes les informations nécessaires au bon déroulement du travail et des tâches quotidiennes dans une institution. Un exemple de système d'information institutionnel est le système d'information de l'hôpital décrit dans [Vassileva 96]. Ce genre de systèmes utilise maintenant des bases de données reliées entre elles dans un hyperespace pouvant être très large. La plupart des employés d'une institution n'ont besoin d'accéder qu'à un sous-ensemble de cet hyperespace en fonction de leurs professions et de leurs objectifs courants de travail. En outre, les nouveaux employés ont besoin d'aide pour naviguer comme c'est le cas des novices dans les systèmes hypermédia éducationnels [Dara-Abrams 02].

La gestion des vues personnalisées dans les espaces d'information constitue un domaine d'application assez nouveau pour les systèmes d'hypermédia adaptatif. Le Web offre une très grande quantité d'informations et de services en ligne qui constituent un hyperespace « illimité ». Puisque les utilisateurs n'ont besoin que de sous-ensembles de cet espace de travail dans leur quotidien, et pour éviter de se perdre dans celui-ci à cause de sa complexité, ces systèmes définissent des vues personnalisées en utilisant des techniques d'hypermédia adaptatif. Ces vues personnalisées (ou sous-ensembles d'informations) sont sélectionnées en se basant sur les objectifs de l'utilisateur, ses intérêts et ses connaissances [Dara-Abrams 02].

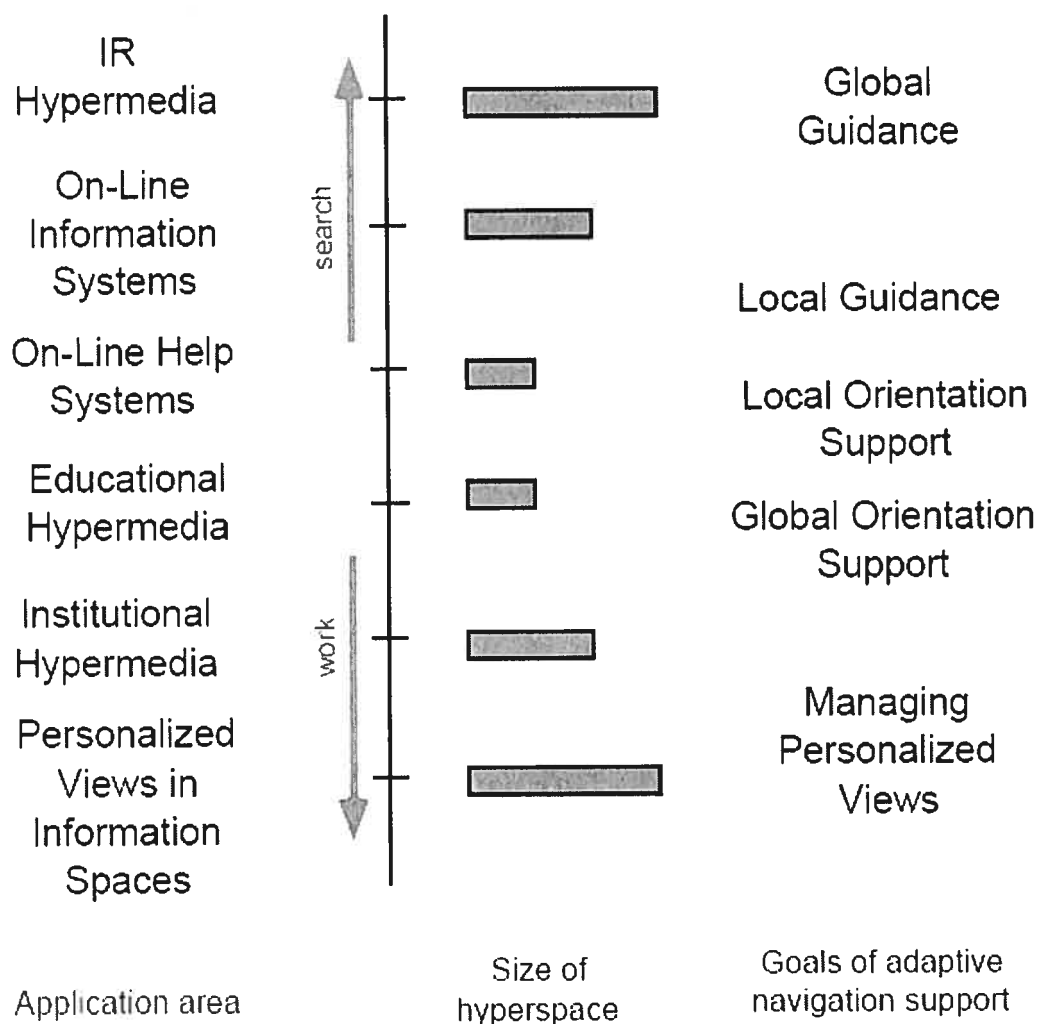
Nous remarquons que ces domaines d'application ne sont pas mutuellement exclusifs, il y a des similarités entre eux et ils partagent quelques problèmes. Dans ce contexte, Brusilovsky [Brusilovsky 96] a placé ces domaines d'application sur un continuum (voir Figure 2.2) et les a classés par paires comme suit :

- La recherche de documents et l'information en ligne.
- L'information ou l'aide en ligne et l'hypermédia éducationnel.
- L'hypermédia éducationnel et l'hypermédia institutionnel.
- L'hypermédia institutionnel et la gestion des espaces d'information.

Dans la Figure 2.2, les systèmes d'hypermédia adaptatif et d'aide en ligne sont situés au milieu du continuum. Ces domaines sont considérés comme des domaines traditionnels pour l'hypermédia adaptatif. Et ces systèmes utilisent les caractéristiques classiques de l'hypermédia comme les liens contextuels, les index, les visites guidées etc., tandis que les systèmes situés aux extrémités du continuum n'utilisent que quelques unes des caractéristiques traditionnelles de l'hypermédia.

Remarquons aussi que la taille de l'hyperespace augmente quand nous nous éloignons du centre. Les systèmes d'hypermédia adaptatif situés aux extrémités, tels que les systèmes de recherche de documents et les systèmes de gestion des vues personnalisées dans les espaces d'information, ont des problèmes reliés à la taille de l'hyperespace et sont donc similaires. Les systèmes qui sont au-dessus du centre du

continuum sont de plus en plus orientés recherche (par exemple la recherche de documents). Par contre, les systèmes qui sont situés au-dessous du centre du continuum sont plus orientés travail et se concentrent sur la tâche principale de l'application (par exemple la gestion des vues personnalisées). La partie droite du diagramme donne les principaux objectifs de l'adaptation de la navigation tout au long du continuum [Dara-Abrams 02].



**Figure 2.2** : Le continuum des systèmes d'hypermédia adaptatif [Brusilovsky 96]

## 2.4.2. Critères d'adaptation

Les systèmes d'hypermédia adaptatif incluent le modèle apprenant suivant lequel se fait l'adaptation. Avant 1996, ces systèmes prenaient en compte seulement cinq paramètres du modèle [Brusilovsky 01, Brusilovsky 96, Dara-Abrams 02] :

- Connaissances de l'utilisateur : elles constituent la source principale d'adaptation pour la plupart des techniques d'adaptation de la présentation. Elles varient d'un utilisateur à un autre et sont fonction du temps. Un système d'hypermédia adaptatif devrait reconnaître ces changements et mettre à jour le modèle apprenant relatif à cet utilisateur [Dara-Abrams 02].
- Objectifs de l'utilisateur : ce sont les buts que l'utilisateur cherche à atteindre en se servant du système. Plusieurs techniques d'adaptation au niveau des liens tiennent compte de ce paramètre pour adapter le chemin que l'utilisateur doit suivre dans le système [Dara-Abrams 02].
- Connaissances antérieures : ce sont les données pertinentes de cet utilisateur comme sa profession, son expérience de travail dans des secteurs relatifs et son point de vue sur le sujet. Certains systèmes d'hypermédia adaptatif utilisent ces données pour faire de l'adaptation au niveau du contenu ou des liens [Dara-Abrams 02].
- Expérience de l'utilisateur avec l'hyperespace : ce paramètre indique le degré de familiarité de l'utilisateur avec l'hyperespace et la facilité avec laquelle il devrait naviguer dans ce dernier. En effet, dans certains cas, il peut être familier avec le sujet mais pas avec la structure de l'hyperespace [Dara-Abrams 02].
- Préférences : ce sont les informations que l'utilisateur doit fournir au système comme le fait d'utiliser certains liens ou regarder des segments particuliers d'une page [Abdel Razek *et al.* 03] puisque le système ne peut pas déduire de telles préférences. Dans la plupart des systèmes de recherche de documents, les préférences sont les seules informations présentes dans le modèle apprenant [Dara-Abrams 02].

Après 1996, Kobsa *et al.* [Kobsa *et al.* 01] ont suggéré de faire la distinction entre l'adaptation aux données de l'utilisateur, celle aux données de l'utilisation et celle aux données de l'environnement.

Les données de l'utilisateur incluent les cinq paramètres définis ci-haut en plus des deux suivants :

- Intérêts de l'utilisateur : c'est un paramètre qui n'est pas nouveau. Il n'était pas pris en compte par les anciens systèmes d'hypermédia adaptatif. Toutefois, avec le développement des systèmes de recherche de documents basés sur le Web, cette situation a changé. En effet, ces systèmes utilisent, en parallèle, les intérêts de l'utilisateur à long terme et sa recherche courante à court terme pour améliorer le filtrage des informations et les recommandations [Brusilovsky 01]. Ces dernières peuvent être également faites en demandant à des utilisateurs d'évaluer les items avec lesquels ils sont familiers, c'est-à-dire d'exprimer leurs intérêts. Ensuite, le système essaiera de trouver d'autres utilisateurs avec des évaluations similaires et recommandera les items que ces derniers ont évalués positivement [Kobsa *et al.* 01] (voir chapitre 4 : Les systèmes de recommandation).
- Traits individuels de l'utilisateur : c'est un groupe de paramètres qui caractérisent l'utilisateur en tant qu'individu, comme par exemple les facteurs cognitifs et les styles d'apprentissage. Les traits individuels sont stables. Ils ne changent pas ou ils ne peuvent changer qu'après une longue période de temps. Ils ne peuvent être déterminés qu'avec des tests psychologiques spéciaux. Il y a des travaux surtout dans les systèmes hypermédia éducationnels où on essaie de faire de l'adaptation par rapport au style d'apprentissage [Specht and Opperman 98; Gilbert and Han 99]. Cependant, il n'est toujours pas clair quels aspects du style d'apprentissage méritent d'être modélisés, ni ce qui peut être fait différemment pour des utilisateurs avec différents styles [Brusilovsky 01].

Les données d'utilisation sont des données qui concernent l'interaction de l'utilisateur avec le système comme les sélections ou les évaluations que fait ce dernier au cours de son utilisation du système [Kobsa *et al.* 01]. Elles ne peuvent pas être affectées

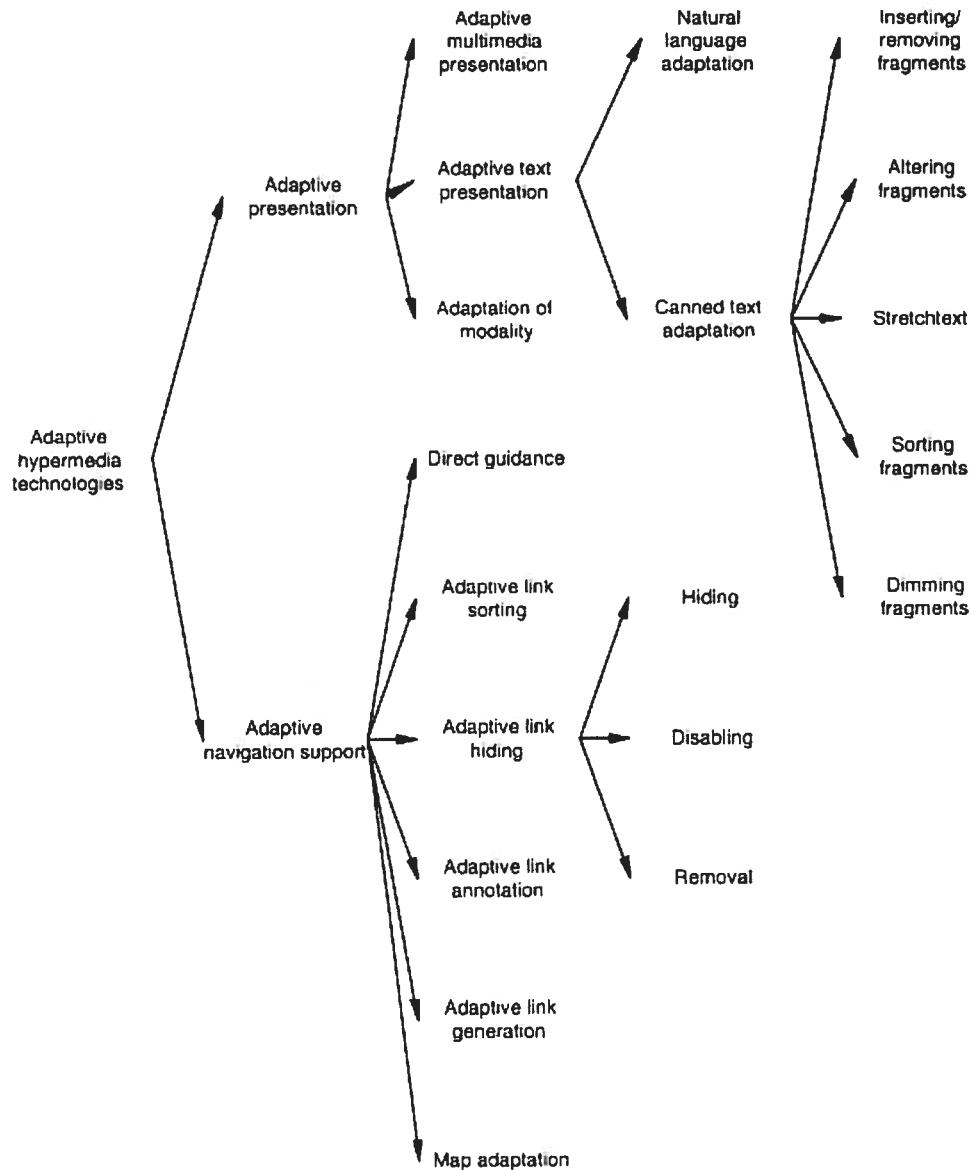
aux caractéristiques d'utilisateur mais peuvent servir dans la prise des décisions d'adaptation [Brusilovsky 01].

Les données d'environnement comportent tous les aspects de l'environnement de l'utilisateur. L'adaptation à l'environnement de l'utilisateur est un nouveau type d'adaptation qui est né avec les systèmes basés sur le Web. Les utilisateurs du côté serveur d'une même application Web peuvent virtuellement être n'importe où et peuvent utiliser des environnements logiciels et matériels différents tels que la version du navigateur, la version de la machine virtuelle de Java installée sur ce navigateur, la bande passante disponible pour leur connexion, la vitesse du processeur, les périphériques d'entrée/sortie [Kobsa *et al.* 01]. Certains systèmes courants d'hypermédia adaptatif ont suggéré quelques techniques pour s'adapter à certains de ces paramètres et à l'endroit de l'utilisateur. La simple adaptation à ces paramètres matériels et logiciels concerne généralement la sélection du matériel et des médias (image, vidéo, etc.) pour la présentation du contenu [Joerding 99]. Les données concernant l'emplacement de l'utilisateur peuvent aussi être utiles pour filtrer des informations, adapter les formats de présentation, ou faire des recommandations en se basant sur les connaissances géographiques [Kobsa *et al.* 01]. L'adaptation à l'emplacement de l'utilisateur a été utilisée avec succès par plusieurs systèmes d'information en ligne comme l'a déjà fait SWAN [Garletti *et al.* 99], un système pour le filtrage d'information pour la marine.

### **2.4.3. Technologies d'adaptation**

Brusilovsky [Brusilovsky 01] a défini deux principales classes d'adaptation dans l'hypermédia : *l'adaptation de la présentation et l'adaptation du support de navigation*. Chaque classe contient des techniques d'adaptation (voir Figure 2.3). Nous allons discuter et présenter la plupart d'entre elles.





**Figure 2.3 :** Les techniques d'adaptation dans les systèmes d'hypermédia adaptatif  
[Brusilovsky 01]

### 2.4.3.1. Adaptation de la présentation

L'adaptation de la présentation est connue aussi sous le nom d'adaptation du contenu. Elle vise à adapter d'abord le contenu d'une page hypermédia en fonction des différentes informations stockées dans le modèle apprenant, ensuite présenter la page à l'utilisateur en se basant sur ces informations. Elle est utilisée essentiellement dans les

systemes d'information en ligne, les systemes d'aide en ligne et les systemes hypermedia educationnels [Dara-Abrams 02], mais elle ne cesse de se repandre egalement dans de nouveaux domaines tels que le commerce electronique [Kobsa *et al.* 01]. D'apres des etudes empiriques, l'adaptation de la presentation peut ameliorer la comprehension du contenu [Brusilovsky & Maybury 02].

Dans cette section, nous allons presenter quelques techniques utilisees pour l'adaptation de la presentation. Ces techniques peuvent varier entre celles qui montrent certaines parties d'une page avec certaines conditions jusqu'a celles qui utilisent du traitement de langage naturel.

#### a) **Texte conditionnel**

La technique consiste a diviser une page en parties de texte ayant des conditions. Quand la page va etre presentee a l'utilisateur, ces conditions seront evaluees pour voir si le texte conditionnel doit etre inclus [Dara-Abrams 02; Stern 01].

C'est une technique tres simple a developper et a mettre en application. L'important est de decider a quels instants montrer un morceau de texte. Mais cette technique n'est pas flexible : chaque morceau conditionnel de texte doit etre associe a une regle au lieu d'associer toute la classe de texte a une regle generale. Par exemple, prenons la regle generale « si l'apprenant ne connait pas un concept, les exemples de ce concept doivent etre montrés ». Elle ne peut pas etre implementee dans le cas du texte conditionnel, il faut avoir une regle pour chaque exemple textuel d'une page qui permet de montrer ce texte si l'apprenant n'a pas compris le concept [Stern 01].

Le systeme AHA! utilise cette technique [De Bra 98]. En effet, chaque page Web contient des informations codees et des informations conditionnees par des regles de type « si-alors » indiquant si ces informations doivent etre fournies. Les conditions de ces regles sont comparees au modele apprenant. Si une regle est evaluee a « vrai », alors le texte de la partie « alors » sera inclus dans la page, autrement ce sera le texte de la partie « sinon » (si elle existe).

## b) Variantes

La technique des variantes consiste à avoir plusieurs façons de décrire un morceau de contenu. Le modèle apprenant sert à distinguer ces façons et choisir les plus correctes [Stern 01].

Nous identifions principalement deux approches qui fonctionnent selon ce principe :

- Variantes de page : elle consiste à faire différentes versions de toutes les pages en lesquelles se produit l'adaptation. Cette approche est relativement simple et facile d'un point de vue technique puisque l'adaptation au moment de l'exécution est ramenée au choix de la bonne page [Brusilovsky 96; Dara-Abrams 02; Kobsa *et al.* 01]. Cependant, elle est également encombrante puisqu'une page complètement nouvelle doit être écrite pour chaque variation des adaptations locales pouvant se produire. Elle manque notamment de flexibilité puisque beaucoup de pages doivent être modifiées si une seule adaptation locale change [Kobsa *et al.* 01].
- Variantes de fragment : elle consiste à faire des variantes pour chaque morceau adaptatif<sup>7</sup> de page. Au moment de l'exécution, les fragments appropriés sont inclus dans un cadre de page statique. Cette approche est plus difficile que les variantes de page d'un point de vue technique puisqu'elle nécessite une génération dynamique des pages Web au moment de l'exécution [Kobsa *et al.* 01].

Le système tutorial intelligent ANATOM-TUTOR [Beaumont 98] utilise les deux approches. Le modèle apprenant est basé sur des stéréotypes<sup>8</sup>, mais peut être personnalisé. Quand l'utilisateur démarre la première fois la composante hypertexte, un texte de départ sera d'abord choisi en se basant sur son stéréotype, ensuite personnalisé en se basant sur son modèle personnel (variantes de page). Ce texte de départ se compose

---

<sup>7</sup> Morceau adaptatif d'une page : c'est le morceau de page qui va changer d'un apprenant à l'autre.

<sup>8</sup> Le modèle apprenant est basé sur des classes déjà définies.

d'unités d'une ou de plusieurs phrases. Les unités sont ordonnées selon l'information qu'elles présentent et selon le niveau de connaissance de l'utilisateur. Si elles sont connues par l'utilisateur, elles peuvent être supprimées. Si elles sont nécessaires, elles peuvent être ajoutées (variantes de fragment) [Kobsa *et al.* 01; Stern 01].

Le système d'information pour touristes AVANTI [Fink *et al.* 98] prend en considération les besoins des handicapés moteurs et de ceux qui présentent des troubles de la vision, et recommande seulement des actions que ces utilisateurs peuvent réellement exécuter. Il utilise la méthode des variantes pour personnaliser la présentation des pages Web. Chaque page contient des éléments statiques et des éléments optionnels associés à des règles d'adaptation qui prennent en compte le modèle apprenant et qui déterminent lesquels de ces éléments optionnels doivent être inclus. En outre, chaque page contient des règles pour mettre à jour le modèle apprenant en fonction de ses interactions avec la page. Dans la Figure 2.4a, AVANTI présente une carte de la cité de Sienne qui inclut les places intéressantes telles que les musées, les restaurants, les hôtels... La même page est présentée à des personnes ayant des troubles de vision dans la Figure 2.4b. L'image a été remplacée par du texte. La carte visuelle donne aux utilisateurs une idée générale des différents endroits et leur permet de savoir quels itinéraires prendre pour s'y rendre. La « carte » pour les personnes ayant des troubles de vision contient des descriptions textuelles de tous les endroits importants par rapport à l'emplacement de la « Piazza del Campo » un point de référence en Sienne [Kobsa *et al.* 01; Fink *et al.* 98; Stern 01].

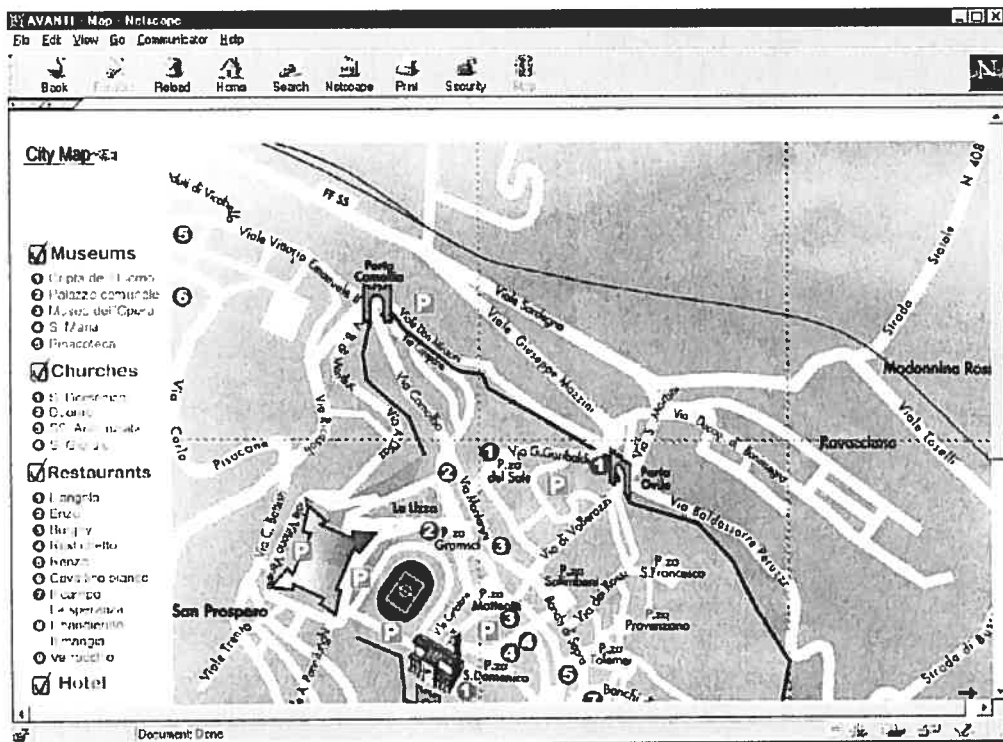


Figure 2.4a : La carte de Sienne en mode graphique dans AVANTI [Fink *et al.* 98]

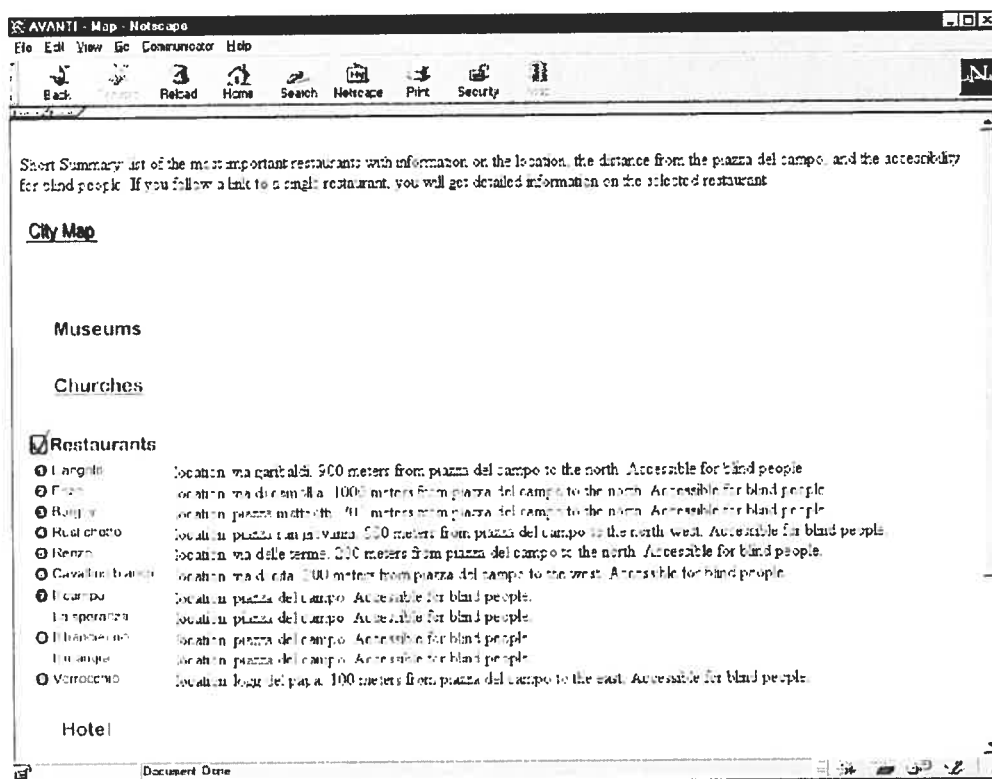


Figure 2.4b : La « carte de Sienne » pour les personnes ayant des troubles de vision dans AVANTI [Fink *et al.* 98]

La méthode des variantes est utilisée aussi pour présenter des informations personnalisées concernant les traitements du cancer [Bental *et al.* 00]. Le domaine est organisé par sujets et par texte concernant ces sujets. Ces derniers sont choisis en se basant sur ce qui est pertinent pour le traitement du patient, sa maladie, et son état actuel (*current time frame*). Si le patient choisit un sujet, les textes concernant ce dernier sont choisis en se basant à la fois sur le dossier médical du patient et, si possible, sur les notes qui fournissent les détails des conditions du patient. Les plans des textes associés aux sujets indiquent quels textes choisir ainsi que la manière dont ils devraient être groupés pour donner lieu à une présentation cohérente [Stern 01].

En utilisant la technique des variantes, un système peut essentiellement présenter la même information d'une façon personnalisée à une variété d'utilisateurs. Un tel système peut donc être très général et flexible. Il peut utiliser des règles générales dont on ne peut faire usage avec le texte conditionnel. Néanmoins, il peut aussi être très intensif en termes de construction de domaine, ce qui implique la nécessité de construire les variantes et les règles qui le présentent. De plus, si le système fait une évaluation incorrecte des utilisateurs, la présentation d'une variante inappropriée peut en résulter. Dans ces cas-là, il doit leur fournir les moyens pour accéder aux autres variantes [Stern 01].

### c) Texte « élastique »

Le texte « élastique » est une technique qui permet à l'utilisateur d'étendre ou de réduire du texte en cliquant dessus avec la souris. Généralement, la taille du texte « élastique » est toujours très petite. En effet, cette technique remplace un mot ou une expression clefs par des informations supplémentaires sur cette expression. Dans les systèmes hypermédias personnalisés, ce texte peut être étendu ou réduit automatiquement par le système en fonction du modèle apprenant. Toutefois, elle reste une technique très flexible du moment que les utilisateurs peuvent aussi adapter le contenu de la page manuellement si l'adaptation faite par le système est inappropriée, ce qui constitue un avantage. Ainsi, si le système décide de cacher ou de présenter incorrectement des informations, les utilisateurs peuvent contourner cette décision erronée. Elle peut être

considérée comme un cas spécial des variantes des fragments qui sont à la fois sous le contrôle du système et sous le contrôle de l'utilisateur. Mais comme les techniques des variantes, elle est très intensive et toutes les informations qui peuvent être incluses ou présentées à l'utilisateur doivent être écrites [Dara-Abrams 02; Kobsa *et al.* 01; Stern 01].

Dans le système PUSH [Höök *et al.* 98], toutes les informations pertinentes d'un objet sont présentées sur une seule page et une partie du contenu peut être cachée (voir Figure 2.5). En se basant sur la tâche de recherche d'informations de l'utilisateur, le système décide quelles informations montrer ou cacher. Ainsi, l'utilisateur peut manipuler les informations présentées en ouvrant/fermant les sous-sections [Stern 01].

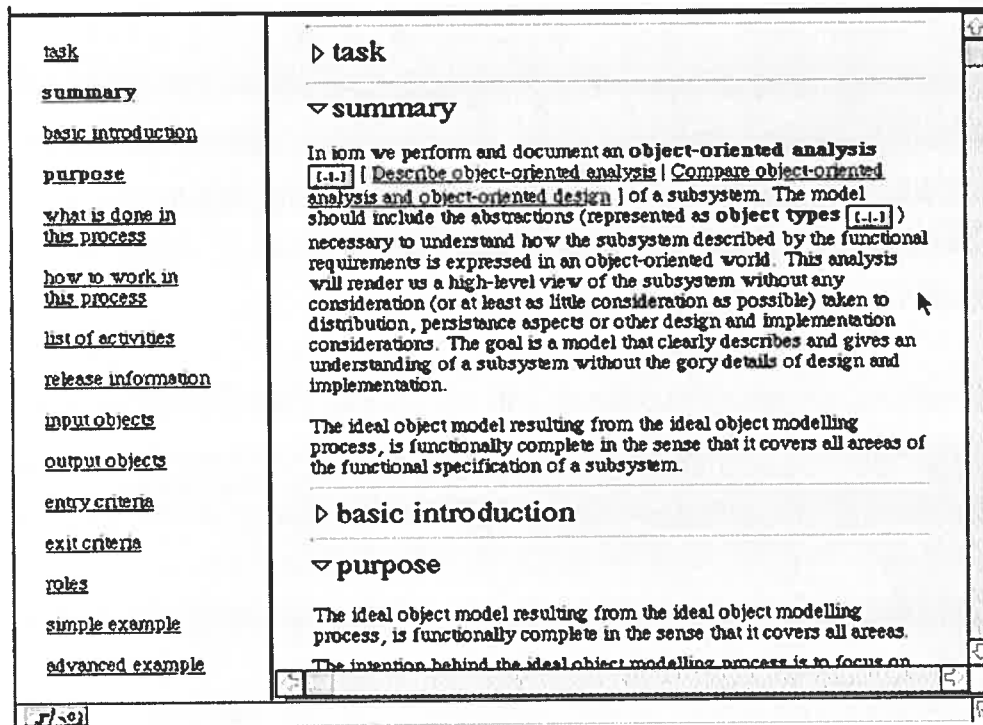


Figure 2.5 : Une page du système PUSH<sup>9</sup>

<sup>9</sup> Cette image a été tirée de cette adresse Web, accédée en Janvier 2005: [www.ags.uni-sb.de/~cullrich/lehre/elearning/aha.ppt](http://www.ags.uni-sb.de/~cullrich/lehre/elearning/aha.ppt), disponible également sur <http://www-etud.iro.umontreal.ca/~mabroukm/aha.ppt>

Le système ADAPTS [De Bra 99] utilise aussi la technique du texte « élastique ». En effet, les paragraphes de la procédure de dépannage (troubleshooting) sont étendus ou réduits en se basant sur la connaissance de la tâche par l'utilisateur ou sur le fait que l'information contenue dans le paragraphe n'est pas pertinente pour le contexte actuel [Stern 01] (voir Figure 2.6).

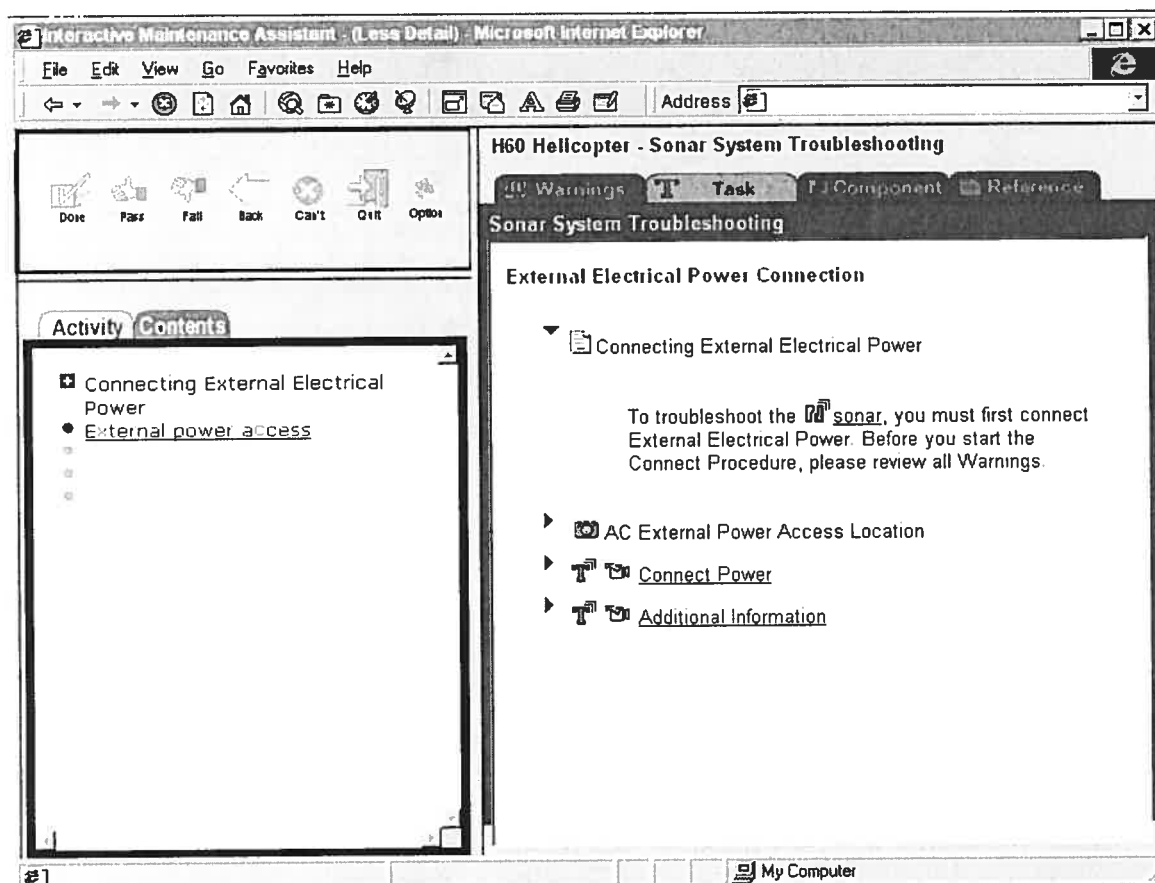


Figure 2.6 : Exemple de procédure de dépannage dans ADAPTS<sup>10</sup>

Dans la Figure 2.7 présentant une page du système KN-AHS [Kobsa *et al.* 97], il est supposé que l'utilisateur ne connaît pas le mot technique « physiologische Krümmungen ». Par conséquent, des informations additionnelles sont insérées automatiquement pour lui expliquer ce mot. Ces informations consistent en un fragment

<sup>10</sup> Cette image a été tirée de cette adresse Web, accédée en Janvier 2005: [www.ags.uni-sb.de/~cullrich/lehre/elearning/aha.ppt](http://www.ags.uni-sb.de/~cullrich/lehre/elearning/aha.ppt), disponible également sur <http://www-etud.iro.umontreal.ca/~mabroukm/aha.ppt>



optionnel de page contenant un petit graphique et un texte « élastique » explicatif (il s'agit du paragraphe marqué par \*\*) [Kobsa *et al.* 01].

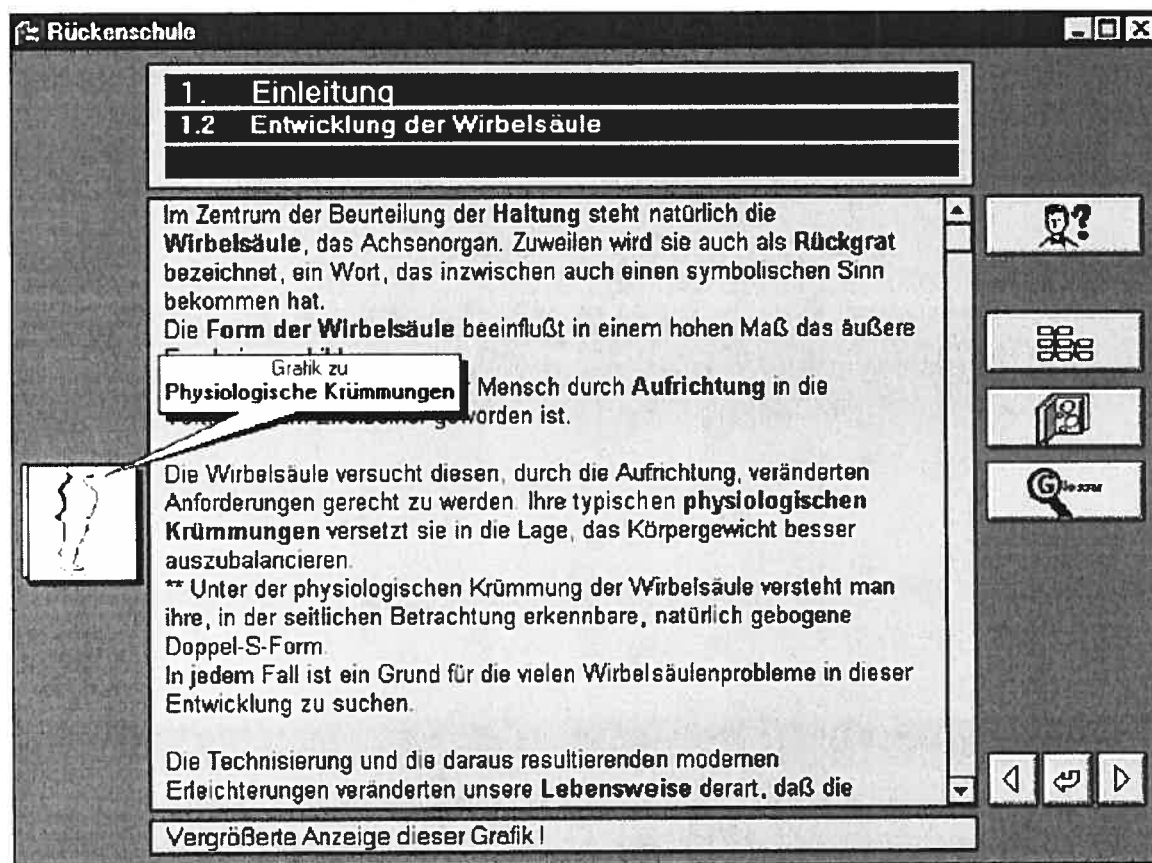


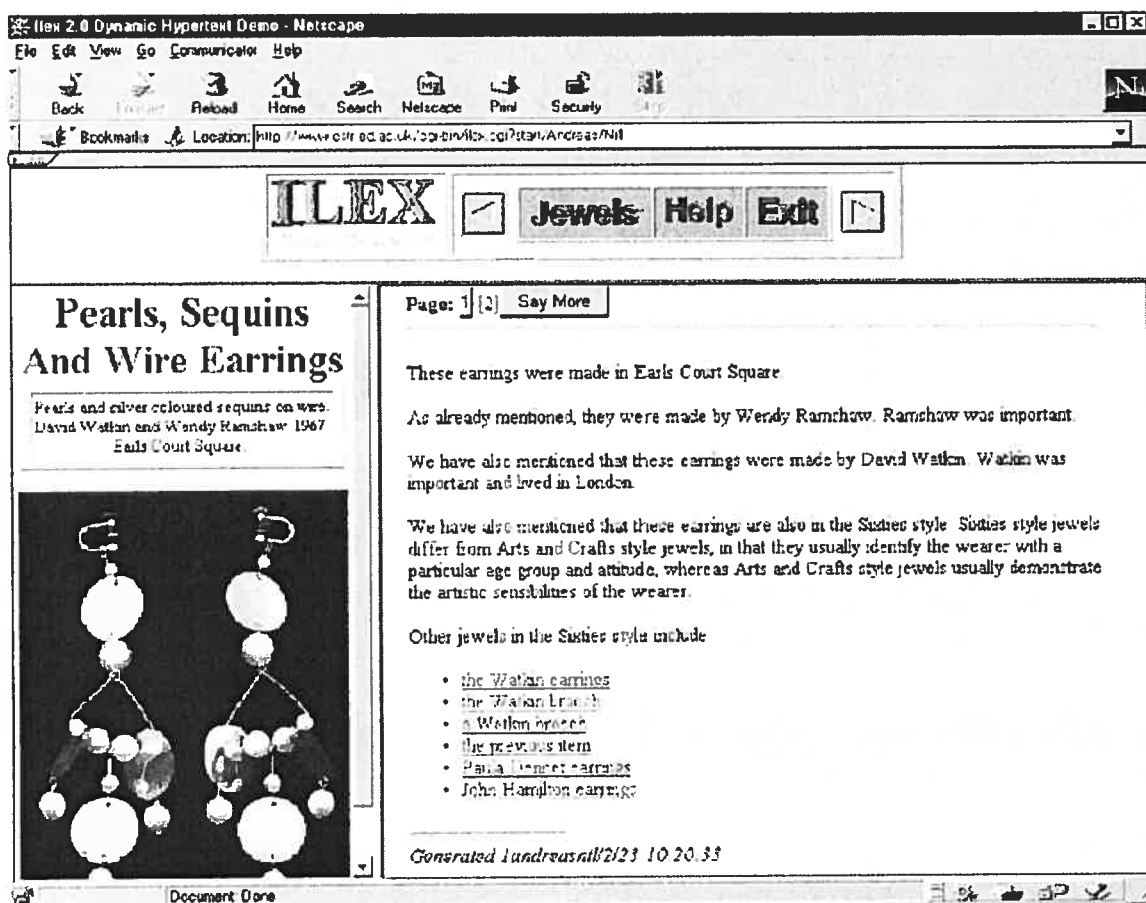
Figure 2.7 : Une page adaptative du cours « Rückenschule » [Kobsa *et al.* 01]

#### d) Techniques de génération de langage naturel

Nous pouvons utiliser des techniques de génération de langage naturel pour créer différentes descriptions textuelles pour différents utilisateurs [Kobsa *et al.* 01].

Une approche simple est d'avoir des modèles de textes avec des cases qu'on peut remplir, par exemple, avec des descriptions de complexités différentes en fonction du niveau d'expertise de l'utilisateur [Ardissono & Goy 00a]. ILEX [Milosavljevic & Oberlander 98] emploie cette approche. En effet, selon les intérêts des utilisateurs, ce système fournit des descriptions personnalisées des objets de la galerie des bijoux du 20ème siècle des musées nationaux d'Écosse. Le système observe la façon avec laquelle

l'utilisateur s'en sert et ajuste la présentation en conséquence [Stern 01]. Les Figures 2.8 et 2.9 montrent deux descriptions qui sont accordées à deux utilisateurs d'intérêts différents [Kobsa *et al.* 01].



**Figure 2.8 :** Une présentation de texte à un utilisateur s'intéressant aux styles de bijoux [Kobsa *et al.* 01]

Le système ARIANNE [De Carolis *et al.* 98] a été conçu pour présenter les directives médicales à des différents groupes d'utilisateurs tels que les patients, les médecins et les étudiants [Stern 01]. Il utilise des techniques de génération de langage naturel pour produire dynamiquement des pages hypertextes adaptées à l'objectif de l'utilisateur courant, à ses connaissances et à sa profession [Kobsa *et al.* 01].

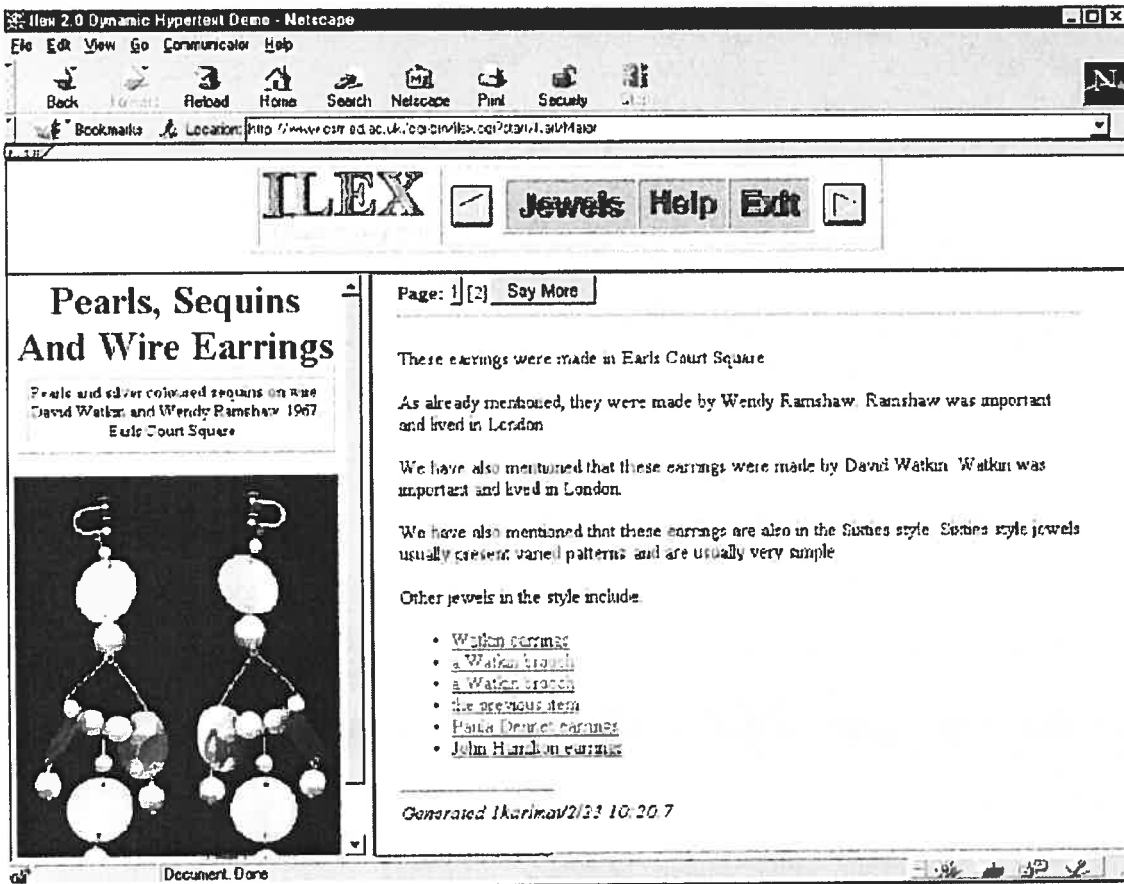


Figure 2.9 : Une présentation de texte à un visiteur n'ayant aucun intérêt spécifique  
[Kobsa *et al.* 01]

HYLITE+ [Bontcheva & Wilks 05] est un système qui génère de l'hypertexte dynamiquement en se basant sur les techniques de génération de langue naturelle. Il fournit des explications de style « encyclopédie » pour tous les termes des deux domaines suivants : l'informatique et la chimie. L'utilisateur interagit avec le système par le biais du navigateur pour spécifier le terme qu'il veut chercher. Le système fournit l'explication du terme avec des liens hypertextes. L'utilisateur peut obtenir d'autres informations en suivant les liens hypertextes ou spécifiant une autre requête.

Le système PEBA-II [Milosavljevic & Oberlander 98] fournit différentes descriptions et comparaisons d'animaux en se basant sur des connaissances taxonomiques selon lesquelles l'utilisateur est considéré comme un novice ou un expert [Kobsa *et al.* 01; Stern 01].

La génération de langage naturel procure les occasions les plus flexibles et les plus sophistiquées pour l'adaptation. Des phrases et des paragraphes peuvent être construits individuellement pour fournir l'information désirée. Cependant, puisque ses techniques sont sophistiquées, elles restent difficiles à implémenter [Stern 01]. La génération de langage naturel semble être également un complément prometteur à la technique du texte « élastique ». Les pages hypermédias qui utilisent le texte « élastique » local doivent être syntaxiquement et sémantiquement correctes dans toutes les combinaisons possibles d'expansion de ce texte [Kobsa *et al.* 01].

#### **2.4.3.2. Adaptation de la navigation**

L'adaptation de navigation est aussi connue sous le nom d'adaptation au niveau des liens. Elle vise à aider les apprenants à trouver leur cheminement à travers l'hypermédia en adaptant les fonctionnalités et l'apparence des liens aux objectifs, aux connaissances et à d'autres caractéristiques de l'utilisateur. Elle peut ainsi accroître la vitesse de la navigation et de l'apprentissage [Brusilovsky & Maybury 02]. Elle encourage la navigation non séquentielle et permet aux utilisateurs d'atteindre un meilleur niveau de connaissances [Brusilovsky 04].

Une étude faite par Weber et Brusilovsky [Weber & Brusilovsky 01] a montré que les utilisateurs occasionnels utilisent plus longtemps le système s'il fournit l'adaptation de navigation.

Dans cette section, nous allons présenter la plupart des techniques utilisées pour l'adaptation de la navigation qui sont employées dans les systèmes développés jusqu'à ce jour. Toutefois, nous devons distinguer si le lien fait partie ou non d'un contexte (le cas d'un mot dans une phrase ou d'un élément dans un modèle c'est à dire le mot, le symbole ou l'icône qui symbolisent le lien dans la page source) [Kobsa *et al.* 01].

##### **a) Annoter les liens**

Cette technique peut être utilisée pour les deux types de liens : contextuels et non contextuels. Elle indique à l'utilisateur la pertinence d'un lien en modifiant son apparence

ou en affichant un symbole coloré [Brusilovsky 04]. Cette technique est très utile pour améliorer la compréhension et réduire le nombre d'étapes pour atteindre un objectif [Dara-Abrams 02].

L'annotation des liens peut être adaptative ou non. Lorsque les liens qui ont déjà été visités changent de couleurs, c'est l'annotation non adaptative et elle est très connue de la plupart des navigateurs Web. Par contre, les systèmes d'hypermédia adaptatif utilisent des couleurs et des codes de symboles différents pour annoter les liens d'une manière personnalisée.

Dans ISIS-TUTOR [Brusilovsky & Pesin 94], les couleurs des liens changent en fonction du niveau de connaissances de l'utilisateur. Le fond gris foncé (avec écriture blanche) mentionne que l'information est déjà connue ou acquise, le fond gris clair (avec écriture blanche) veut dire que l'information en question est prête à être étudiée et le fond gris clair (avec écriture noire) indique que l'information n'est pas celle qu'il faudra étudier en ce moment (voir Figure 2.10).

Доступные тени	
+ 1 Общий вид формата	2 Арифметические выражения
3 Удаление пустых строк	4 Безусловный переход на новую строку
+ 5 Переход на новую строку	6 Выбор позиции в строке
7 Печать пробелов	+ 8 Вывод поля
9 Понятие MFN	10 Безусловный литерал
11 Арифметическая функция L	12 Арифметическая функция Mfn
13 Арифметическая функция Val	14 Арифметическая функция Rsum
15 Арифметическая функция Rmin	16 Арифметическая функция Rmax
17 Арифметическая функция Ravr	18 Совмещение % и #
19 Совмещение / и #	20 Условный литерал
21 Повторяющийся литерал	22 Вывод MFN
23 Строковые выражения	24 Префиксный условный литерал
25 Суффиксные литералы	26 Нуль-литералы
27 Повторяющийся литерал с +	28 Префиксный повторяющийся литерал
29 Установка режима вывода	30 Совмещение условных литералов и %
31 Совмещение условных литералов с #	32 Совмещение условных литералов с /
33 Совмещение условных литералов с C	34 Совмещение условных литералов с X
35 Совмещение условных литералов с M	36 Режимы L,U в команде M.
37 Режим H в команде M	38 Режим D в команде M
39 Режим P в команде M	40 Строковая функция F
41 Строковая функция Ref	42 Строковая функция S
43 Программы пользователя format	44 Выражения отношения
Enter - изучить F4-практ F6-учи F8-инд.задач F9-назад PgDn-след.стр.	
+ Хорошо изучен	Изучен
Можно изучать	Не готов

Figure 2.10 : L'annotation dans ISIS-TUTOR [Brusilovsky 04]

ISIS-TUTOR était une source d'inspiration par la suite pour le développement de ELM-ART [Brusilovsky, Schwarz & Weber 96], le premier système hypermédia basé sur le Web utilisant l'annotation des liens. Il sert à enseigner le langage de programmation LISP. Il utilise à la fois des icônes (des bulles rouges, vertes et jaunes) et des polices différentes (italique, gras et régulier) pour indiquer à l'utilisateur l'état du contenu du lien.

En métaphore avec les couleurs des feux de circulation, la couleur rouge veut dire que le contenu derrière le lien n'est pas prêt à être étudié, tandis que la couleur verte indique les lectures recommandées. D'autres couleurs comme le jaune ou le blanc indiquent d'autres états dans l'apprentissage comme le manque de nouvelles connaissances derrière le lien en question [Brusilovsky 04]. La Figure 2.11 montre l'annotation adaptative des liens dans ELM-ART [Weber & Brusilovsky 01].

INSPIRE [Papanikolaou *et al.* 03] assiste les apprenants durant leur navigation dans le domaine de connaissances et les oriente en utilisant la métaphore des feux de circulation. Afin de proposer un chemin optimal à l'apprenant dans ce domaine, les différentes couleurs permettent de distinguer les unités recommandées par le système, l'unité courante et les différents pré-requis.

InterBook [Brusilovsky *et al.* 98] est la première plateforme permettant de créer et de délivrer des cours adaptatifs sur le Web [Brusilovsky 04]. Il utilise à la fois les annotations adaptatives et le guidage direct pour orienter les étudiants dans le cours. Les annotations sont basées sur l'historique, les connaissances de l'étudiant et les pré-requis du cours. Elles sont faites à partir de quatre bulles de couleurs différentes et trois polices différentes [Stern 01] (voir Figure 2.12).

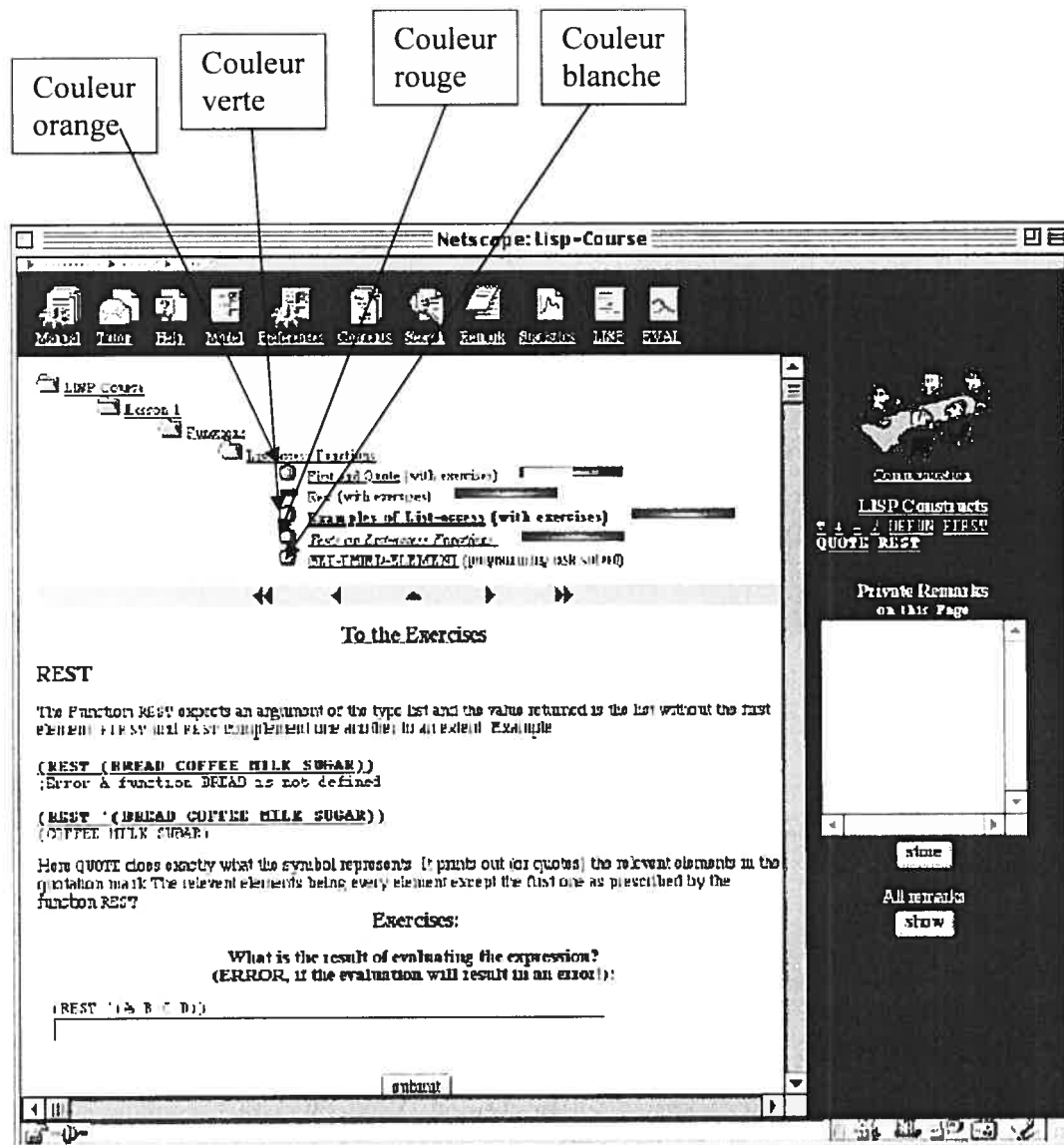
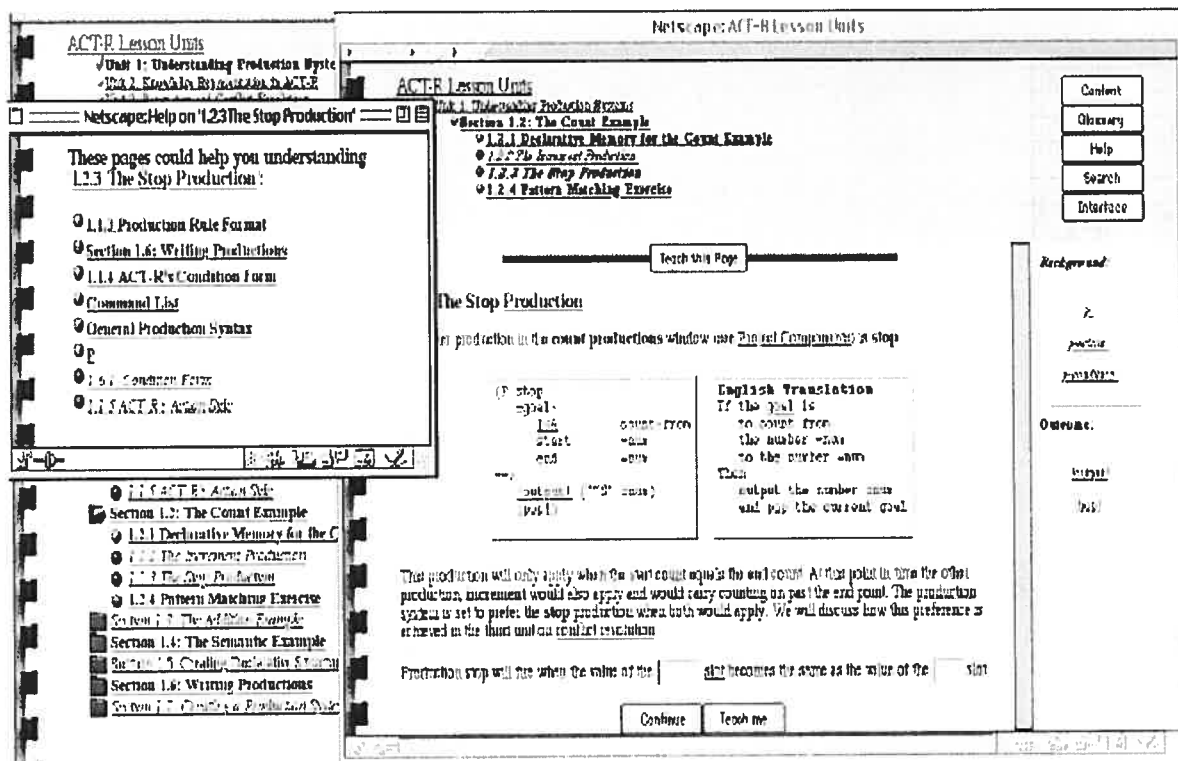


Figure 2.11 : L'annotation des liens dans ELM-ART [Brusilovsky 04]

Le KBS-HyperBook [Henze & Nejd1 01] utilise aussi l'annotation des liens à travers la métaphore des feux de circulation introduite précédemment. Les annotations concernent des liens entre les unités de l'information, qui sont les parties sémantiques du cours. Elles indiquent si un item d'information (sorte de mot-clé ou mot concept) est déjà connu, suggéré ou trop difficile [Stern 01].



**Figure 2.12** : L'annotation des liens dans InterBook. En outre, la fenêtre de la recommandation de l'aide (à gauche) utilise la génération (ajout dynamique) et le tri des liens [Brusilovsky 04]

Dans ACE [Specht & Oppermann 98], l'annotation des liens est basée sur l'état ou niveau des connaissances de l'utilisateur et un modèle pédagogique des unités du domaine. Elle est faite à l'aide de bulles colorées : la couleur rouge indique un concept que l'étudiant n'est pas encore prêt à étudier, la couleur verte indique un concept prêt à être étudié, la couleur orange indique les unités pour lesquelles il ne manque pas des prérequis mais qui ne sont pas recommandées. En outre, les liens pointant sur les concepts que l'étudiant a déjà visités ont l'icône de crochet (voir Figure 2.13).



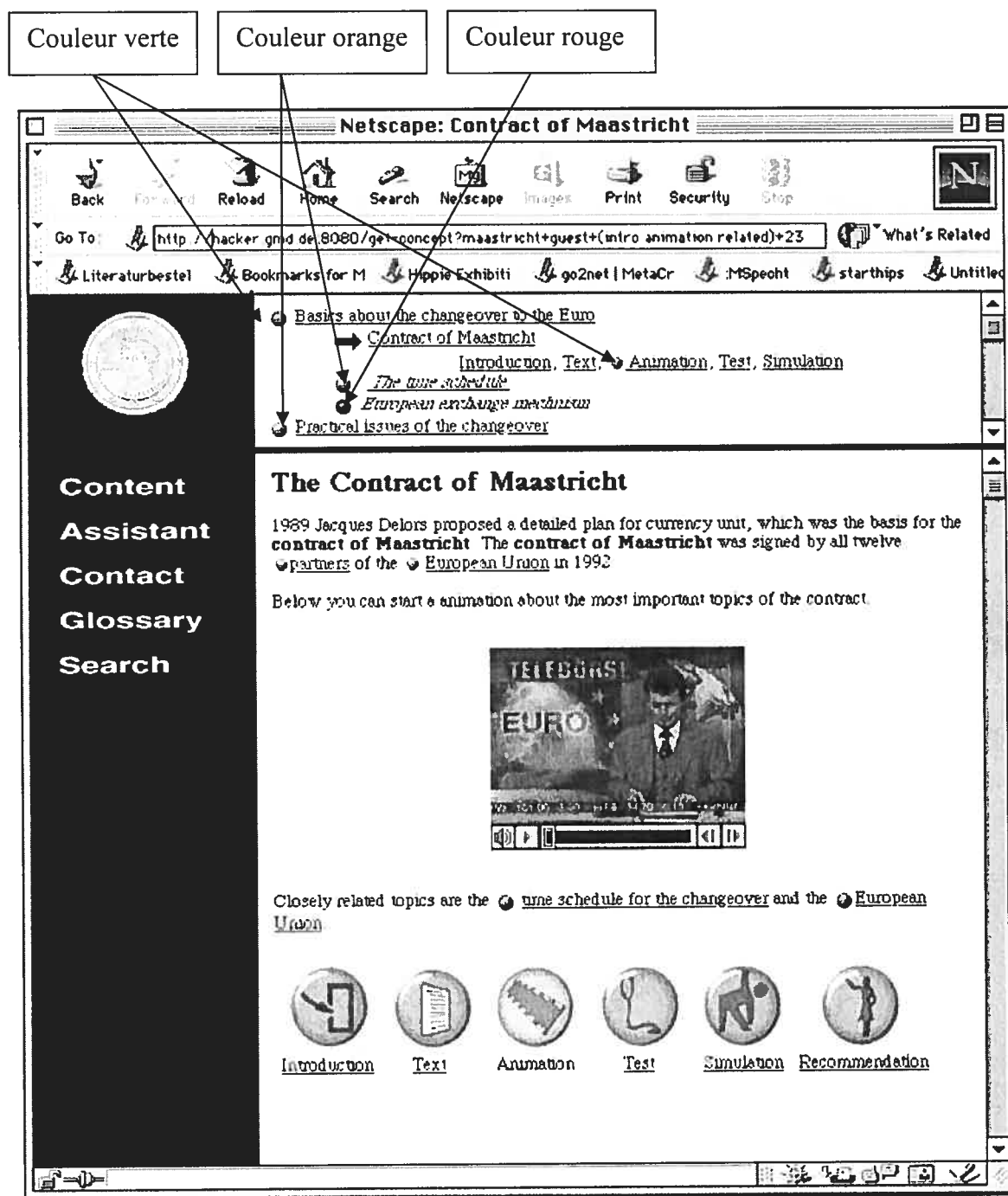


Figure 2.13 : L'annotation dans ACE [Specht & Oppermann 98]

L'annotation peut être utilisée avec toutes les formes possibles de liens. Cette technologie garde un ordre stable des liens. Elle est généralement plus puissante que celle qui cache les liens parce que cette dernière peut distinguer seulement deux états pour les nœuds, à savoir les pertinents et les non pertinents, tandis que les applications existantes

de l'annotation peuvent distinguer jusqu'à six états [Brusilovsky 04]. L'annotation peut aider les utilisateurs à choisir le lien à suivre. En outre, si ces recommandations sont incorrectes, l'utilisateur peut les ignorer puisque les liens restent actifs même s'ils ne sont pas recommandés [Stern 01].

Pour toutes ces raisons, l'annotation adaptative s'est développée plus tard et est devenue la plus utilisée. Cependant, les annotations peuvent également donner lieu à des confusions, surtout si l'utilisateur ne sait pas comment l'annotation a été calculée. De même, une annotation pourrait avoir différentes significations dans différentes circonstances. Si des annotations doivent être employées, elles doivent être simples dans leur présentation. Ainsi, les utilisateurs peuvent rapidement les voir et comprendre les recommandations [Stern 01].

#### **b) Trier les liens**

Cette technique vise à ordonner les liens d'une page particulière selon leur pertinence pour l'utilisateur (du plus recommandé au moins recommandé) en se basant essentiellement sur le modèle apprenant.

L'un des premiers systèmes à avoir utilisé cette technique est HYPERFLEX [Kaplan *et al.* 93]. Le feedback est donné à l'utilisateur en triant les liens par ordre de pertinence par rapport à la page courante et au but spécifié. Si l'utilisateur n'accepte pas le classement donné par le système ou le modifie, le système raffinerait le modèle apprenant pour améliorer graduellement les résultats de recherche [Kobsa *et al.* 01].

ALICE [Kavcic 02] est un système éducationnel basé sur le Web, conçu pour un domaine d'enseignement arbitraire. Il utilise une technique appelée « insertion automatique des liens » qui combine deux techniques d'adaptation de navigation notamment celle qui cache les liens et celle qui trie les liens. L'algorithme sélectionne dynamiquement les liens qui mènent aux unités les plus appropriées à apprendre après l'unité courante. Ce sont les liens les plus proches des points de vue contenu et niveau de difficulté.

Ardissono *et al.* 99 [Ardissono *et al.* 99] ont pu trier les articles des nouvelles de façon individualisée. Le système utilise le modèle apprenant pour sélectionner les nouvelles appropriées, déterminer le niveau de détails des nouvelles et des annonces, et générer dynamiquement la page. Le modèle apprenant est mis à jour dynamiquement en se basant sur certains événements comme sauter des nouvelles ou des sections, suivre un lien pour avoir plus de détails, supprimer quelques détails, etc. Le système prend aussi en considération le nombre de connexions aux sections par semaine ou par mois [Stern 01].

Cette technique est utilisée aussi pour présenter des listes de classement (ranking lists) d'items recommandés comme des films [Alspector *et al.* 97] ou des équipements techniques [Ardissono & Goy 00b]. Elle peut également être utilisée en se basant sur la fréquence d'utilisation comme dans l'historique de la plupart des navigateurs Web [Kobsa *et al.* 01].

Elle permet aux utilisateurs de contourner les jugements du système puisque les liens restent actifs comme c'est le cas pour l'annotation des liens. En outre, ils peuvent rapidement voir les recommandations puisque les liens les plus suggérés sont en haut de la liste. Par contre, avec l'annotation des liens, ils doivent chercher toute la liste et lire toutes les annotations pour trouver les meilleurs liens. Elle peut réduire significativement le temps de navigation, accélérer la recherche pour les utilisateurs et diminuer leurs erreurs dans les applications hypermédias orientées recherche [Brusilovsky 04; Kobsa *et al.* 01].

Cependant, avec le tri des liens, il y a parfois un problème avec les recommandations basées sur la popularité. En effet, les liens les plus populaires (en fonction du nombre de visites) sont toujours en haut de la liste et ils continuent à être les plus visités même s'ils ne sont pas réellement les mieux recommandés pour l'utilisateur. Un lien intéressant peut être enterré au milieu de la liste et ce lien ne pourra pas devenir populaire en raison de cet emplacement. Ainsi, des utilisateurs ne seront jamais dirigés vers des liens plus intéressants [Stern 01].

Cette technique a un autre inconvénient, c'est qu'elle rend l'ordre des liens instable. En effet, elle peut le modifier à chaque fois que l'utilisateur entre la page en question.

Elle peut être utilisée avec les liens non contextuels<sup>11</sup>, rarement avec les pages d'index car elles gardent habituellement un ordre stable des liens, mais jamais avec les liens contextuels.

Pour plusieurs de ces raisons, cette technique est utilisée en conjonction avec la génération des liens (ajout des liens dynamiques) pour présenter des nouveaux liens à l'utilisateur [Brusilovsky 04].

### **c) Guider l'apprenant**

Le guidage est une technique qui fournit à l'utilisateur une vue particulière de l'hyperespace ce qui lui permet de mieux décider où aller. Avec cette technique, une séquence de liens ou un itinéraire sont habituellement générés pour que l'utilisateur puisse les suivre dans l'hyperespace.

Le système ADAPTS [De Bra 99] utilise plusieurs techniques d'adaptation, dont le guidage, pour aider les techniciens à accomplir leurs tâches. Le système présente étape par étape les directives adaptées à l'expérience de l'utilisateur dans le dépannage. En effet, il présente une liste adaptative de contrôles qui suggère un itinéraire optimal pour accomplir la tâche en question. Cette liste de contrôles indique l'état courant de l'accomplissement de la tâche. A l'aide des annotations, on peut savoir l'étape courante ou les étapes complétées ou restantes de la tâche [Stern 01].

Les buts d'Opperman et de Specht dans [Specht & Opperman 98] étaient de garder l'apprenant dans un itinéraire optimal en se basant sur ses connaissances et d'adapter la séquence des unités d'études (concepts et sections) à ses intérêts en lui présentant par exemple des tests concernant les connaissances pré-requises et manquantes

---

<sup>11</sup> Rappelons qu'un lien contextuel peut être le cas d'un mot dans une phrase ou d'un élément dans un modèle c'est à dire le mot, le symbole ou l'icône qui symbolisent le lien dans la page source

pour étudier l'unité en question. S'il s'avère qu'il n'est pas en mesure de les réussir, le système lui recommandera de travailler sur ces pré-requis avant l'unité choisie. Le système calcule aussi la prochaine meilleure unité d'études (*learning unit*) pour l'étudiant, et adapte dynamiquement la stratégie d'enseignement à son style préféré en tenant compte de son succès avec ce style. Pour cela, le système garde différentes informations concernant l'apprenant qui sont mises à jour à l'aide d'un dialogue direct avec ce dernier. Ces informations contiennent ses préférences pour la langue, pour les médias et pour les paramètres de l'interface [Specht & Opperman 98; Stern 01].

Il existe une approche plus directe pour guider l'apprenant dans un cours en utilisant le « guidage direct » qui est la plus simple des techniques de l'adaptation de navigation. Il suggère le prochain meilleur lien pour l'utilisateur en fonction de ses objectifs, ses connaissances et d'autres paramètres représentés dans le modèle apprenant. Il est très souvent appliqué dans les systèmes éducationnels d'hypermédia adaptatif. C'est une méthode très flexible pour la présentation du matériel d'enseignement (*learning material*) car le nœud destination du bouton « prochain » n'est pas connecté directement au nœud courant mais il peut être calculé dynamiquement pendant l'exécution en tenant compte des dernières actions de l'utilisateur [Brusilovsky 04; Kobsa *et al.* 01].

Cette technique est utilisée dans iMANIC [Stern 01] à la fin d'un cours pour présenter à l'étudiant le prochain cours, un résumé ou un quiz. Elle est utilisée aussi dans ALICE [Kavcic 02] pour fournir aux apprenants une navigation linéaire leur permettant de se déplacer à partir de l'unité courante vers l'unité suivante ou la précédente.

Une étude d'ELM-ART [Weber & Brusilovsky 01] a montré que le guidage direct marche mieux avec les utilisateurs qui ont peu de connaissances sur le sujet, tandis que l'annotation adaptative est plus utile pour les utilisateurs qui ont de bonnes connaissances sur le sujet.

Cependant, il ne fournit pas de soutien pour les utilisateurs qui ne veulent pas suivre les suggestions du système. Il est utile mais il faut l'utiliser en conjonction avec l'une des autres techniques d'adaptation de navigation [Brusilovsky 04].

La technique du « guidage » est similaire à celle de « tri des liens » car elle peut présenter les liens à l'utilisateur avec un ordre de préférence. Toutefois, la technique de « tri des liens » ne considère pas les liens comme un itinéraire complet, elle les présente individuellement et ainsi, après que l'utilisateur suit le premier lien, il ne saura peut-être pas le reste de l'itinéraire. De ce point de vue, le guidage peut être plus bénéfique puisqu'il fournit à l'utilisateur des suggestions plus compréhensibles en terme d'itinéraire à suivre plutôt que le « prochain » lien.

Néanmoins, le guidage, et spécialement le guidage direct, peut-être entaché de confusion pour les utilisateurs parce qu'ils peuvent percevoir que la structure de l'hyperespace est entrain de changer pour eux. Par exemple, si un utilisateur est entrain de voir pour la première fois le sujet A, le système peut le guider en lui proposant de suivre l'itinéraire : sujet B, sujet C, sujet D. Toutefois, la deuxième fois que l'utilisateur consulte le sujet A, il se verra proposer un autre itinéraire totalement différent du premier, ne contenant aucun des sujets B, C et D puisqu'il les a déjà consultés [Stern 01].

#### **d) Cacher les liens**

L'objectif de cette technique est de restreindre l'espace de navigation en cachant, supprimant ou désactivant les liens qui mènent à des pages non pertinentes pour l'utilisateur. Par exemple, une page peut être considérée comme non pertinente si elle ne répond pas à l'objectif de l'utilisateur ou si elle présente des données que l'utilisateur n'est pas encore en mesure de comprendre [Brusilovsky 04]. Donc, les liens cachés existent dans la structure de l'hyperespace mais sont simplement indisponibles jusqu'à ce que le système (ou le tuteur) décide que l'utilisateur est prêt à consulter les pages auxquelles mènent ces liens. De cette façon, cette technique évite aux utilisateurs de faire face à la complexité de tout l'hyperespace.

Les systèmes d'hypermédia éducationnel constituent le principal domaine d'application pour cette technique. En effet, c'est un moyen simple pour implémenter l'approche la plus populaire dans le domaine de l'éducation : pour enseigner un cours à un étudiant, on commence normalement par lui présenter des notions préliminaires

simples et ensuite, on introduit de nouvelles notions plus compliquées au fur et à mesure qu'il progresse dans le cours [Brusilovsky 04].

Les premiers systèmes d'hypermédia adaptatif comme ISIS-TUTOR [Brusilovsky & Pesin 94] utilisent une méthode simple pour cacher les liens qui consiste à les enlever simplement de la page (voir Figure 2.14).

A travers le système AHA [De Bra & Calvi 98], De Bra et Calvi ont suggéré et implémenté d'autres variantes de cette technique. Ces variantes, comme par exemple la désactivation des liens, sont devenues plus populaires puisqu'elles laissent le mot qui représente le lien intact et désactivent ou cachent juste l'indicateur visuel du lien.

Le système présenté dans [Pilar da Silva *et al.* 98] utilise cette technique. Les liens dans l'hypermédia sont typés et pondérés. Les nœuds sont aussi typés et peuvent être soit des documents, soit des concepts. Il peut y avoir des liens entre des concepts ou bien entre des concepts et des documents. Quand un utilisateur consulte un document, son niveau de connaissances pour le concept approprié sera mis à jour si le niveau de difficulté du document est supérieur à celui du concept. Les liens pour les documents pertinents sont accessibles si leur niveau de difficulté est considéré comme approprié pour l'utilisateur, sinon ils seront non visibles [Stern 01].

Specht et Oppermann [Specht & Oppermann 98] utilisent aussi cette technique dans leur système. Ils ajoutent des éléments de l'interface et les tâches qui manipulent ces éléments sont introduites à l'apprenant au fur et à mesure en se basant sur leur difficulté et les objets déjà appris. Les textes dans ACE peuvent contenir des mots-clés pour des concepts relatifs. Si le texte est présenté pour la première fois à l'apprenant, il ne contiendra aucun hyperlien pour ces concepts qui ne sont pas encore appris par l'étudiant. Ensuite, après que l'apprenant travaille avec le système, ACE lui présentera le texte dans lequel tous les mots-clés des concepts appris et prêts à être appris sont des liens. En outre, il désactive les liens pour les unités qui ne sont pas prêtes à être étudiées.

Доступные темы	
+ 1 Общий вид формата	2 Арифметические выражения
3 Удаление пустых строк	4 Безусловный переход на новую строку
+ 5 Переход на новую строку	6 Выбор позиции в строке
7 Печать пробелов	+ 8 Вывод поля
9 Понятие MFN	10 Безусловный литерал
11 Арифметическая функция L	12 Арифметическая функция MFn
13 Арифметическая функция Val	14 Арифметическая функция Result
15 Арифметическая функция Rmin	15 Арифметическая функция Rmax
17 Арифметическая функция Bavg	18 Сообщение % и %
19 Совмещение / и #	20 Условный литерал
21 Повторяющийся литерал	22 Вывод MFN
23 Строковые выражения	24 Префиксный условный литерал
25 Сuffixный литерал	25 Null-литерал
27 Повторяющийся литерал с +	28 Префиксный повторяющийся литерал
29 Установка режима вывода	30 Совмещение условных литералов и %
31 Совмещение условных литералов с #	32 Совмещение условных литералов с /
33 Совмещение условных литералов с 0	34 Совмещение условных литералов с X
35 Совмещение условных литералов с M	36 Режим L, O в команде M
37 Режим H в команде M	38 Режим D в команде M
39 Режим F в команде M	40 Строковая функция F
41 Строковая функция Ref	42 Строковая функция S
43 Программы пользователя Format	44 Выражения отношения
Enter - изучить F4-практ F6-учи F8-вид задач F9-назад F10-след стр.	
+ Хорошо изучен	Изучен
Можно изучать	Не готов

Доступные темы	
+ 1 Общий вид формата	2 Арифметические выражения
3 Удаление пустых строк	4 Безусловный переход на новую строку
+ 5 Переход на новую строку	6 Выбор позиции в строке
7 Печать пробелов	+ 8 Вывод поля
9 Понятие MFN	10 Безусловный литерал
13 Арифметическая функция Val	20 Условный литерал
21 Повторяющийся литерал	22 Вывод MFN
27 Повторяющийся литерал с +	28 Префиксный повторяющийся литерал
29 Установка режима вывода	52 Размещение первой строки поля
53 Выбор длины фрагмента поля	54 Выбор смещения фрагмента поля
55 Вывод подполя	56 Повторяющиеся группы
Enter - изучить F4-практ F6-учи F8-вид задач F9-назад	
+ Хорошо изучен	Изучен
Можно изучать	

Figure 2.14 : La première image présente l'annotation des liens dans une page de ISIS-TUTOR. La deuxième présente la même page, mais en plus de l'annotation des liens, ceux qui ne sont pas prêts à être étudiés (fond gris clair avec écriture noire dans la première image) sont supprimés [Brusilovsky 04]

A partir de ces quelques exemples, nous pouvons constater que cette technique est employée pour empêcher des utilisateurs de voir le matériel pour lequel ils ne sont pas prêts. Si le système peut correctement le juger, alors cette technique sera bénéfique pour les utilisateurs puisqu'ils comprendront bien ce qui leur sera présenté. Cependant, si le système se trompe au sujet du niveau de connaissances de l'utilisateur, alors que ce dernier est prêt pour voir ce matériel, il n'y aura aucune manière pour l'atteindre puisque les liens ont été enlevés. Par conséquent, contrairement à plusieurs techniques qu'on a



présentées précédemment, les utilisateurs dans ce cas ne pourront pas revenir sur les erreurs du système.

#### e) **Ajouter des liens dynamiques**

Contrairement à d'autres techniques d'adaptation de la navigation telles que l'annotation et le tri des liens qui adaptent la présentation de liens, cette technique crée de nouveaux liens dans une page [Brusilovsky 04]. Elle permet d'avoir une structure dynamique de l'hyperespace basée sur les différences individuelles. Ainsi, la structure de l'hyperespace sera différente pour chaque utilisateur [Stern 01].

Brusilovsky [Brusilovsky 04] distingue trois types de cette technique : le premier consiste à découvrir de nouveaux liens utiles entre les documents et les ajouter de manière permanente aux liens existants. Le deuxième est la génération de liens pour la navigation basée sur la similarité. Le dernier est la recommandation dynamique des liens pertinents. Il insiste aussi sur le fait que les deux premiers types ne sont pas adaptatifs contrairement au troisième.

InterBook [Brusilovsky *et al.* 98] est l'un des premiers systèmes à utiliser cette technologie. Dans une fenêtre d'aide, le système propose à l'étudiant de visiter plusieurs pages qui peuvent l'aider à comprendre le cours (voir Figure 2.12). Les liens vers ces pages sont annotés et générés dynamiquement dans cette fenêtre.

Dans le KBS-HyperBook [Henze & Nejd1 01], les cours se composent de plusieurs lectures qui consistent en des unités de texte et plusieurs cours appartiennent à un groupe de cours. Un plan du contenu est réalisé par l'auteur de la ressource de l'information. Pour chaque page HTML, ce plan est l'ensemble des items d'information (mots-clés ou mots concepts). A l'aide de ces plans, le système calcule dynamiquement les liens aux informations relatives. Les pages relatives sont celles dont les plans ne sont pas disjoints avec celui de la page courante [Stern 01].

Le guide de musée HIPS [Oppermann & Specht 00] constitue une autre application de l'ajout des liens où le système introduit automatiquement des liens pour les

peintures proches de l'utilisateur. Des liens pour les peintures éloignées sont également introduits si le système infère un intérêt de l'utilisateur dans leurs sujets, leurs peintres ou la période du temps durant laquelle elles ont été peintes en se basant sur les données des peintures que l'utilisateur a déjà visité [Kobsa *et al.* 01].

Le fait d'ajouter des liens dynamiques peut transformer un espace statique en un espace personnalisé du moment que les liens sont calculés en fonction de paramètres personnels. Cependant, comme les liens sont calculés dynamiquement, la structure de l'hypermédia peut changer tant que l'utilisateur utilise le système. Ceci peut-être très désorientant pour l'utilisateur surtout s'il s'attend à ce que les liens d'un nœud à l'autre existent toujours [Stern 01].

## Chapitre 3 : Évaluation de l'apprenant

Dans ce chapitre, nous allons présenter l'évaluation de l'apprenant. Il s'agit là d'un aspect utilisé de plus en plus dans l'apprentissage à distance. Nous allons détailler le cycle de vie d'une question dans un système éducationnel basé sur le Web et nous finirons par présenter quelques systèmes ayant recours à cet aspect.

### 3.1. Généralités

L'évaluation de l'apprenant est un aspect pédagogique assez important dans l'apprentissage à distance. Cet aspect est pris en compte par la plupart des environnements d'apprentissage surtout ceux qui sont dans le domaine de l'éducation. En général, il permet au tuteur d'évaluer les connaissances de l'apprenant. Selon le moment de son exécution par rapport à l'apprentissage et l'objectif recherché, nous pouvons classer l'évaluation dans une des quatre classes suivantes [Pirotte 02] :

- Diagnostique : elle se fait au début de l'apprentissage et permet au professeur d'évaluer les capacités de l'apprenant (exemple : les pré-tests).
- Formative : elle accompagne l'apprentissage et permet au professeur de situer l'apprenant, et à l'apprenant de se situer par rapport à un objectif donné.
- Formatrice : elle se fait avant, pendant et après l'apprentissage. Elle vise à apprendre aux apprenants à s'auto-évaluer et leur permet d'évoluer dans leur apprentissage.
- Sommative : elle se fait en fin d'apprentissage. Elle vise à mesurer les compétences acquises ou non acquises de l'apprenant de deux manières :
  - Normative : par rapport aux autres apprenants.
  - Critériée : par rapport à des critères.

L'évaluation est généralement réalisée à travers des tests et des quiz. Chaque test ou quiz constitue une suite de questions qui peuvent être évaluées comme correctes, incorrectes ou partiellement correctes (ou incomplètes). Le type d'une question dépend

du type de la réponse prévue. Ainsi, il existe plusieurs types de questions dont les questions booléennes, les questions à choix multiples (avec une seule réponse ou avec plusieurs réponses), les questions d'association, les textes à trous et les questions de pointage où la réponse est une partie d'une figure [Brusilovsky & Miller 99].

### 3.2. Cycle de vie d'une question

Dans l'éducation basée sur le Web, Brusilovsky et Miller [Brusilovsky & Miller 99] ont analysé le cycle de vie d'une question. Ils l'ont divisé en 3 grandes étapes : la préparation, la livraison et l'évaluation. Chaque grande étape se compose elle-même de petites étapes (voir Tableau 3.1).

*La livraison* concerne la vie active d'une question. Quant aux autres étapes (*la préparation et l'évaluation*), elles concernent respectivement ce qui la précède et ce qui la suit.

La vie d'une question commence au moment de sa *création*. Les systèmes d'éducation basés sur le Web offrent souvent à l'auteur une technologie et un outil pour la création des questions et toutes les questions créées sont *stockées* dans le système. La vie active d'une telle question commence dès qu'elle est *sélectionnée pour être présentée* dans un test ou quiz.

Ensuite, le système *livre* la question en la *présentant* et en *fournissant une interface* pour la réponse de l'étudiant qui va être évaluée par la suite.

Lors de l'étape de *l'évaluation*, le système *évalue* la réponse (correcte, incorrecte ou partiellement correcte), fournit le *feedback*, *note* la question et *enregistre* la performance de l'étudiant.

Les systèmes d'éducation basés sur le Web qui utilisent ce genre d'outils, diffèrent de par le type et la qualité de l'aide fournie dans chacune des étapes décrites un peu plus haut.

**Tableau 3.1:** Les étapes du cycle de vie d'une question [Brusilovsky & Miller 99]

<b>Avant la vie active</b> <i>(Préparation)</i>	<b>Pendant la vie active</b> <i>(Livraison)</i>	<b>Après la vie active</b> <i>(Évaluation)</i>
<ul style="list-style-type: none"> <li>• Créer</li> <li>• Stocker</li> <li>• Sélectionner</li> </ul>	<ul style="list-style-type: none"> <li>• Présenter</li> <li>• Interagir</li> <li>• Avoir la réponse</li> </ul>	<ul style="list-style-type: none"> <li>• Évaluer</li> <li>• Noter et enregistrer</li> <li>• Fournir le feedback</li> </ul>

### 3.2.1. L'étape de la préparation

Une question est généralement créée par des enseignants et des développeurs. Elle a normalement les composantes suivantes : le texte de la question, un ensemble des réponses possibles, une indication qui permet de connaître quelles sont les réponses correctes, le type de l'interface pour la présentation et un feedback spécifique pour chacune des réponses possibles. En outre, l'auteur peut fournir des données supplémentaires comme les mots-clés, le poids de la question, la partie du cours sur laquelle porte le test, le nombre d'essais etc. Elles peuvent être utilisées aussi bien pour choisir et présenter une question particulière que pour noter ou évaluer la réponse [Brusilovsky & Miller 99].

Les options pour la création dépendent toujours de la technologie utilisée pour stocker une question dans le système. Il existe deux façons de stocker une question : *format de présentation* ou *format interne*. Les questions stockées sous un format de présentation sont aussi appelées *questions statiques* car elles sont stockées en tant qu'une partie d'un code HTML (comme un formulaire HTML) et un système d'éducation basé sur le Web ne peut que les présenter telles qu'elles sont [Brusilovsky & Miller 99].

Le fait de stocker une question dans un format interne veut dire la stocker comme un enregistrement de base de données. Les composantes de la question (texte, réponses et feedback) représentent les champs de cet enregistrement. La question sera générée à partir de son format interne et présentée à l'étudiant au moment de la livraison. Ce

deuxième type de format est plus flexible que le premier. En effet, la même question peut être présentée sous des formes différentes (question à choix multiples ou texte à trous) ou avec des caractéristiques d'interface différentes (avec des boutons radios ou une liste de sélections). L'ordre des options dans les questions à choix multiples peut être mélangé ce qui fournit un niveau important d'individualisation [Carbone & Schendzielorz 97]. De plus, d'un point de vue pédagogique, c'est très utile et ça diminue la possibilité de fraude. Il existe deux manières de créer les questions dans un format interne : avec une forme basée sur l'interface graphique de l'utilisateur (*GUI : Graphical User Interface*) ou avec un langage de balises spécial pour les questions [Campos Pimentel, dos Santos Junior & de Mattos Fortes 1998]. Chacune de ces approches a des avantages et inconvénients mais la plus populaire reste la première qui est utilisée par la plupart des systèmes commerciaux avancés comme WebCT<sup>12</sup> et Blackboard<sup>13</sup>.

Il y a des équipes dont les recherches portent sur les bases de données de questions dans un format interne. Elles essaient d'avancer dans trois directions principales :

- Les questions paramétriques : le fait d'utiliser des questions paramétriques permet de créer un nombre illimité de tests à partir d'un même ensemble de questions, ainsi que d'éliminer les fraudes [Brusilovsky & Miller 99].
- Les méta-données des questions : la connaissance de données supplémentaires concernant une question (par exemple le type, les mots-clés, le poids, la partie du cours sur laquelle elle porte) peut aider le système à générer, à la demande de l'auteur ou du système lui-même, des quiz ou des tests personnalisés. En effet, à un endroit particulier du cours, et en se basant sur ce dont l'étudiant aura besoin à ce moment là, l'auteur peut spécifier certains paramètres pour le quiz (le nombre total des questions, difficultés...) et le système produira à la demande un quiz personnalisé.
- L'ordonnancement adaptatif des questions : un modèle apprenant de recouvrement (*overlay*) peut représenter séparément la connaissance des

---

<sup>12</sup> <http://www.webct.com>

<sup>13</sup> <http://www.blackboard.com>

différents concepts. A l'aide de ce modèle, ELM-ART [Weber & Brusilovsky 01] a pu fournir des questions et des tests adaptés au niveau des connaissances de l'étudiant et réduire le nombre de questions nécessaires pour évaluer les connaissances de ce dernier.

### 3.2.2. L'étape de la livraison

Un des plus importants paramètres dans cette étape est la technologie d'interaction employée pour avoir la réponse de l'étudiant. Elle détermine les options de livraison et influence la création et l'évaluation. Nous pouvons distinguer les liens HTML, les formulaires HTML/CGI, les langages des scripts, les plug-in, le langage java etc.

Les liens HTML sont une technologie d'interaction simple qui permet de présenter à l'étudiant l'ensemble des réponses possibles comme une liste de liens HTML, chaque lien étant connecté à une page de feedback particulière. Le problème de cette technologie est que les questions sont difficiles à créer (car la logique de la question doit être liée à l'hypertexte du cours) et qu'elle ne prend compte que de deux types de questions : les questions booléennes (oui/non) et les questions à choix multiples avec une seule réponse. Cette technologie était utilisée dans les premiers systèmes éducatifs basés sur le Web quand des technologies plus sophistiquées, comme la CGI (en anglais, *Common Gateway Interface*), le javascript ou le java, n'avaient pas encore vu le jour.

Cependant, plusieurs systèmes commerciaux utilisent, de nos jours, une technologie qui est une combinaison des formulaires HTML et des scripts d'évaluation compatibles avec la CGI. En effet, les formulaires HTML peuvent bien présenter les principaux types de questions :

- Les questions booléennes et les questions à choix multiples avec une seule réponse peuvent être représentées à l'aide de boutons radios, des listes de sélection ou des menus pop-up.
- Les questions à choix multiples avec plusieurs réponses peuvent être représentées avec plusieurs listes de sélection ou des cases à cocher.

- Les textes à trous peuvent être implémentés avec des champs de texte.
- Les questions plus complexes comme les questions d'association peuvent être représentées avec les formulaires.

En outre, des champs cachés peuvent être utilisés dans les formulaires pour contenir des informations supplémentaires dont le script CGI a besoin.

Les technologies employées du côté serveur, comme celle-ci (formulaires/CGI), ont plusieurs avantages. En effet, les tests peuvent être réalisés avec des outils de création HTML. Les informations nécessaires à l'évaluation du test (les paramètres des questions, les réponses, le feedback) sont stockées du côté client sans risque de tricherie (car un système bien développé n'aura besoin que de l'identifiant du test et l'identifiant de l'étudiant pour évaluer un test). Les fonctions d'évaluation (par exemple : l'enregistrement des résultats, la notation, la production du feedback) sont faciles à implémenter et sont généralement exécutées par un même script d'évaluation du côté serveur [Brusilovsky & Miller 99].

Toutefois, cette technologie a aussi des inconvénients puisqu'elle ne peut être utilisée que pour présenter les principaux types de questions. Des questions plus avancées, par exemple les questions interactives dans lesquelles il y a des activités de « glisser-déposer » (*drag-and-drop*), ne peuvent pas être implémentées avec une technologie utilisée seulement du côté serveur. De plus, les questions basées sur les scripts CGI ne fonctionnent pas quand la connexion avec le serveur est coupée ou très lente.

JavaScript est une technologie utilisée aussi pour la livraison et l'évaluation des questions. L'interface fournie par cette technologie est similaire à celle fournie par la première, mais en plus elle peut présenter des questions plus avancées et interactives telles que la sélection d'un fragment pertinent dans un texte. Avec JavaScript, toutes les données nécessaires à l'évaluation et au feedback de la question, tout comme le programme de l'évaluation, doivent être stockées comme une partie du texte de la question dans le même fichier, ce qui signifie que la question en JavaScript est autonome



et autosuffisante. Toutefois, cela signifie aussi que les étudiants peuvent accéder au code source de la question et le modifier. Donc, JavaScript reste un meilleur choix pour l'auto-évaluation (sans avoir à noter l'étudiant). De toute façon, avec seulement du JavaScript, on ne peut pas enregistrer les notes et les évaluations de l'étudiant. C'est pour cette raison qu'il vaudrait mieux utiliser une technologie hybride qui emploie du JavaScript avec une autre technologie du côté serveur, comme par exemple les CGI. Avec une telle technologie hybride, JavaScript pourra présenter plus de types de questions tout en laissant l'évaluation et l'enregistrement des résultats aux CGI pour des raisons de sécurité comme c'est le cas dans WebCT<sup>14</sup>.

Les technologies de « plug-in » permettent d'atteindre un niveau plus élevé de liberté dans les interfaces. Par exemple, le « plug-in » de Shockwave<sup>15</sup> permet de faire fonctionner des présentations multimédia réalisées avec des outils Macromedia. C'est une technologie puissante qui permet de fournir des *animations interactives* donc elle permet de délivrer une variété de questions développées avec des outils Macromedia avec le moindre effort. Cependant, elle a le même inconvénient que JavaScript : l'enregistrement des résultats des évaluations nécessite la connexion avec le serveur.

Au niveau le plus élevé, nous trouvons la technologie Java qui est un langage de programmation complet désigné pour être intégré à la fonctionnalité du navigateur et de l'Internet, et combinant la connectivité de la technologie formulaire/CGI et l'interactivité de Shockwave et de JavaScript. L'interface de la question peut être développée avec Java, et la question peut en même temps communiquer avec le navigateur et n'importe quel objet de l'Internet que ce soit une autre application Java ou un serveur. Cependant, développer les interfaces de questions avec Java est plus compliqué que les développer avec la technologie formulaire/CGI. Java constitue aussi une solution pour implémenter une variété de questions qu'on ne peut pas implémenter avec la technologie

---

<sup>14</sup> <http://www.webct.com>

<sup>15</sup> <http://www.macromedia.com/shockwave/>

formulaire/CGI comme les questions de pointage<sup>16</sup>, les questions graphiques<sup>17</sup> et d'autres types de questions spécialisées [Brusilovsky & Miller 99].

### 3.2.3. L'étape de l'évaluation

Comme nous l'avons déjà mentionné, le choix de la technologie d'interaction influence considérablement les options de l'évaluation. Nous voulons dire par *évaluation* le moment où une réponse est jugée correcte, incorrecte ou partiellement correcte. Normalement, les réponses correctes et les réponses incorrectes sont fournies lors de la création de la question, donc l'évaluation est soit liée à la question comme c'est le cas des questions à choix multiples avec une seule réponse, soit réalisée par une comparaison simple (texte à trous par exemple). Cependant, dans certains domaines, les réponses correctes ne sont pas forcément égales littéralement à la réponse correcte stockée comme c'est le cas pour les mots désordonnés, les nombres réels et les expressions algébriques simples. Dans ces cas, un simple programme de comparaison (peut être écrit avec JavaScript sinon avec Java ou avec un autre programme du côté serveur utilisant l'interface CGI s'il y a des calculs avancés à faire) s'avère nécessaire. Parfois, il faudra même un « expert du domaine » comme l'interpréteur Lisp pour le langage de programmation Lisp dans ELM-ART [Weber & Brusilovsky 01]. Cet « expert du domaine » doit être exécuté du côté serveur avec une passerelle compatible CGI.

Concernant le feedback, on peut informer simplement l'étudiant de l'état de sa réponse. On peut aussi donner la bonne réponse et fournir un feedback personnalisé. Ce genre de feedback peut informer l'étudiant de ce qui est juste dans la réponse correcte, ce qui est mauvais dans la réponse incorrecte ou partiellement incorrecte, motiver l'étudiant et fournir des informations ou des liens pour y remédier [Brusilovsky & Miller 99].

Tout feedback personnalisé est créé et stocké avec la question. Si les métadonnées de la question comportent les concepts ou les sujets évalués, on peut fournir un bon feedback puisqu'on connaît les connaissances manquantes et on sait où les trouver.

---

<sup>16</sup> L'utilisateur fournit une réponse dans ce genre de questions quand il pointe sur un élément donné.

<sup>17</sup> L'utilisateur manipule des cartes, des graphes ou des figures dans ce genre de questions.

Ce qui veut dire que la puissance du feedback « dépend » en quelque sorte de la technologie de création et de stockage. La quantité d'information, fournie comme feedback, dépend du contexte. Par exemple, dans un contexte d'auto-évaluation, il vaudra mieux que l'étudiant reçoive du feedback le plus possible car celui-ci constitue une importante source d'apprentissage. Par contre, dans une situation d'évaluation stricte, l'étudiant peut ne pas obtenir ni la réponse correcte, ni l'information que sa réponse est correcte. Dans ce cas, le seul feedback qu'il obtient est le nombre de ses réponses correctes dans tout le test, ce qui réduit considérablement ses chances de tricher, et en même temps ses chances d'apprendre. Cependant, plusieurs systèmes essaient de rendre l'évaluation moins stricte en fournissant plus de feedback et en essayant de vaincre les fraudes par d'autres moyens [Brusilovsky & Miller 99]. Toutefois, nous pouvons combiner apprentissage et évaluation stricte en utilisant des technologies plus avancées comme les questions paramétriques [Mornar *et al.* 03] et la génération des tests basée sur la connaissance (*knowledge based test generation*) [Weber & Specht 97]. Ces technologies permettent de générer un nombre illimité de questions et dans ce cas le système peut fournir un bon feedback sans favoriser la tricherie [Brusilovsky & Miller 99].

Dans un contexte d'auto-évaluation, générer un bon feedback après le test est un objectif en soi à atteindre absolument puisque l'étudiant a besoin de voir ses résultats pour comprendre et apprendre. Par contre, dans un contexte d'évaluation stricte, *noter l'étudiant et enregistrer ses performances* pour une future utilisation constituent les derniers objectifs à atteindre pour un système d'éducation basé sur le Web. En effet, dans ce contexte, au sein d'une université par exemple, les notes sont importantes pour les professeurs, les administrateurs de cours et les étudiants. Dans les premiers systèmes éducatifs basés sur le Web, le professeur avait lui-même à traiter les résultats des tests, les noter, enregistrer les notes et s'assurer que toutes les parties impliquées peuvent accéder à ces données selon la politique de l'université. Cette option est facile à implémenter et ne nécessite aucune connaissance de nouvelles technologies, comme Java, de la part du professeur. C'est pour cette raison qu'elle est encore utilisée comme alternative dans plusieurs systèmes [Brusilovsky & Miller 99].

Cependant, un bon système d'éducation basé sur le Web devrait être capable de noter un test automatiquement et enregistrer les résultats dans une base de données. Il devrait aussi fournir un accès strict aux étudiants, aux professeurs et aux administrateurs selon les politiques de l'université. Par exemple, un étudiant ne devrait pas pouvoir voir les notes des autres étudiants et un professeur devrait pouvoir changer les notes assignées automatiquement. Ces notes peuvent être stockées dans des fichiers structurés sur lesquels il y a des droits d'accès. Elles peuvent aussi être stockées dans des bases de données. La technologie de bases de données peut fournir plusieurs options pour voir les notes et d'autres données concernant la performance de l'étudiant comme le temps mis pour finir le test en question. De plus, cette technologie permet de générer facilement plusieurs statistiques, notamment les résultats de l'étudiant dans plusieurs tests. La comparaison des statistiques est importante pour les professeurs et pour les étudiants puisque, dans ce contexte virtuel, la communication entre étudiants et entre étudiant et professeur reste limitée. Par exemple, si nous comparons la moyenne de la classe avec les notes individuelles de l'étudiant, nous pourrions déterminer son rang dans la classe ou bien si un professeur compare les notes de la classe pour les différents tests et questions, il pourra trouver les questions très simples, les questions très difficiles et les questions créées incorrectement [Brusilovsky & Miller 99].

### **3.3. Présentation de quelques systèmes d'évaluation**

Il existe plusieurs systèmes d'évaluation pour l'apprenant. Cependant, dans cette partie, nous allons en présenter juste quelques uns des plus récents.

#### **3.3.1. Le système ALE**

ALE [Pessin 03] est un environnement intégré d'apprentissage adaptatif qui fournit plusieurs outils d'évaluation de l'étudiant. En effet, il y a différents types de tests, de quiz, d'exercices et de questionnaires : test de choix (d'image ou de texte), textes à trous, questions d'association, les questions d'ordonnement. Le test de choix présente plusieurs alternatives d'images ou de textes et demande à l'utilisateur d'en choisir une ou plusieurs. Les autres types de test ont déjà été définis au début de ce chapitre. Cet

environnement permet de les créer et de les présenter d'une manière flexible. Dans les trois sections suivantes, nous décrivons les différentes étapes du cycle de vie des tests dans ALE (préparation, livraison, évaluation).

### **3.3.1.1. La préparation et le stockage des tests**

Un test dans ALE est stocké dans un *format interne* et l'interface de création de test reflète bien sa structure. Chaque test, indépendamment de son type, possède des options spécifiques que l'auteur pourra modifier comme la difficulté, le nombre d'essais permis et le type de feedback. Le feedback montre à l'utilisateur si la réponse est correcte ou lui donne des références à un objet d'apprentissage pouvant l'aider à trouver la solution. Les auteurs dans ALE peuvent faire facilement des quiz en réutilisant les autres tests et questions déjà existants qui ont été créés par tous les auteurs, c'est ce que Pesin, Brusilovsky et Miller ont appelé *la flexibilité du temps de création* [Pesin 03; Brusilovsky & Miller 99].

#### **a) Les tests de choix**

Les tests de choix contiennent une liste de choix ou de réponses. Chaque choix doit avoir son texte ou image, une indication s'il est correct, un feedback à donner dans le cas d'une mauvaise sélection et un objet d'apprentissage chargé dont la connaissance permet de sélectionner la réponse donc il est responsable en quelque sorte de la bonne ou mauvaise sélection. Ces deux derniers paramètres (feedback et objet chargé) sont optionnels et peuvent aider à la fois le système et l'utilisateur. En effet, le système pourra déterminer les objets d'apprentissage que l'utilisateur connaît bien ou non. Ce dernier pourra se voir orienté ou donné une astuce pour lui expliquer pourquoi sa réponse était fausse [Pesin 03].

#### **b) Les textes à trous**

Les questions de textes à trous consistent généralement en un ensemble ordonné de fragments de textes et de champs vides. Chaque champ doit avoir du texte comme

réponse correcte ou des variantes de ce texte si la réponse n'est pas unique, un feedback pour les mauvaises réponses et un objet d'apprentissage chargé [Pesin 03].

### **c) Les questions d'association**

Les questions d'association consistent à avoir plusieurs groupes contenant des textes, à partir desquels l'utilisateur choisit ceux qui vont ensemble. L'auteur (le créateur) doit d'abord spécifier le nombre de ces groupes (taille de l'association), ensuite les éditer : chaque membre du groupe doit avoir la valeur du texte correspondant qui sera affiché, le feedback pour les associations incorrectes et un objet chargé. L'auteur doit aussi spécifier les combinaisons correctes en mettant les membres de chaque groupe dans l'ordre approprié.

#### **3.3.1.2. La livraison**

Dans ALE, un test peut être choisi pour être présenté au moment où l'utilisateur navigue dans la structure du cours ou à travers les hyperliens entre les objets d'apprentissage. Les tests sont stockés dans la base de données des objets d'apprentissage, donc le système peut former une variété de collections de tests en se basant sur la structure du cours, les méta-données des tests (ou questions) et le niveau de connaissances de l'utilisateur courant. Le système peut générer un quiz en se basant sur les questions stockées dans une unité d'apprentissage (et de ses sous-unités) ou dans tout le cours ce qui peut aider l'utilisateur à tester ses connaissances concernant une unité d'apprentissage particulière avant ou après l'avoir étudiée. A l'aide du modèle apprenant, ALE peut aussi fournir une séquence adaptative de questions pour chaque utilisateur en se basant sur ses connaissances, ses objectifs et ses préférences [Pesin 03].

#### **3.3.1.3. L'évaluation et le feedback**

Le feedback montre si la réponse est correcte et fournit des liens à des objets d'apprentissage chargés pour toutes les réponses incorrectes. Une réponse est reconnue automatiquement par le système en utilisant les données fournies par l'auteur de la question. L'étudiant peut soumettre une nouvelle réponse si le nombre d'essais n'est pas

atteint et si c'est permis par le professeur. Le résultat est stocké dans le modèle apprenant. Le statut des objets d'apprentissage chargés fournis par l'auteur change dans le modèle : les objets qui sont en charge des réponses correctes sont marqués comme *maîtrisés* (non maîtrisés dans le cas des réponses incorrectes). De cette façon, ALE peut construire un modèle de connaissances pour chaque étudiant en observant ses réponses pour chaque test. Ce modèle est utilisé pour réaliser l'annotation adaptative où l'état de tous les objets d'apprentissage et le pourcentage des tests maîtrisés pour toutes les unités sont affichés à côté des objets. Il est aussi utilisé pour le séquençement adaptatif qui permet de recommander le prochain meilleur objet d'apprentissage. La présentation adaptative, qui montre des blocs conditionnels dépendamment de la connaissance des autres objets d'apprentissage, est également réalisée à l'aide de ce modèle. Ces informations peuvent aider aussi bien le professeur que l'étudiant pour vérifier le niveau de connaissances de ce dernier [Pesin 03].

### **3.3.2. Le système SAPZAS**

SAPZAS est un système qui a pour objectif de tester les étudiants et évaluer leur progrès. En langue Croate, SAPZAS veut dire système pour l'apprentissage et les tests automatisés. Il comprend deux composantes : la première s'occupe de l'administration de l'étudiant et génère les questions et les tests, et la deuxième est une application Web de Microsoft ASP.NET C# qui s'occupe de l'évaluation et de l'analyse à travers l'Internet [Mornar *et al.* 03].

Dans les sections suivantes, nous allons décrire comment ce système réalise chacune des étapes du cycle de vie d'une question.

#### **3.3.2.1. La préparation**

Le système utilise les formulaires de Microsoft Access comme interface de création pour les questions des tests. Toutes les informations concernant les tests et les étudiants sont stockées dans une base de données Microsoft SQL Server 2000. Les tests peuvent contenir quelques types de questions qui ont été décrits un peu plus haut dans ce chapitre : les questions booléennes (vrai/faux), les questions à choix multiples avec une

seule réponse correcte et les questions à choix multiples avec plusieurs réponses correctes. Le nombre des choix proposés pour ces deux derniers types de questions est arbitraire. Comme nous le savons déjà, l'inconvénient majeur de ces types de questions est que l'étudiant « chanceux » pourra choisir la bonne réponse sans la connaître réellement. Dans ce système, le problème est traité en assignant un score négatif aux réponses fausses dont la valeur est supérieure à la probabilité de deviner la bonne réponse, ce qui fait que si l'étudiant ne connaît pas la réponse, il vaut mieux pour lui de ne pas répondre du tout. Dans SAPZAS, on peut aussi créer des questions paramétriques, ce qui permet à l'auteur de créer un nombre illimité de tests à partir d'un même ensemble de questions pour diminuer les chances de fraude. En général, chaque question doit avoir un texte (la donnée de la question avec quelques fragments hypermédias comme les icônes,...), un poids, un ensemble des réponses possibles, l'indice de la réponse correcte (respectivement l'indice de la fonction d'évaluation de la réponse correcte dans le cas des questions paramétriques). A part ces données, l'auteur doit spécifier aussi un intervalle de valeurs permises pour chacun des paramètres dans le cas des questions paramétriques. Quant à la fonction d'évaluation, elle est écrite dans un langage de script et elle évalue chaque réponse candidate en fonction des paramètres générés arbitrairement. La donnée et les réponses possibles de la question sont groupées dans un objet structuré et stockées dans la base de données. Cet objet structuré est, par la suite, partitionné automatiquement en des fragments HTML et stocké dans le serveur Web pour augmenter la performance du système. Durant la phase de l'apprentissage, la position de la réponse correcte changera toujours car l'ordre des fragments des réponses est créé arbitrairement. Les tests sont également créés arbitrairement et on ne doit spécifier que la structure du test lors de cette phase (le nombre total des questions et le nombre de questions de chaque leçon devant être introduites dans le test) [Mornar *et al.* 03].

### **3.3.2.2. La livraison**

Dans SAPZAS, l'environnement utilisé pour délivrer les tests est développé à l'aide de la technologie de Microsoft ASP.NET C# [Mornar *et al.* 03]. Le test est un ensemble de pages Web. Chaque page correspond à une question. Les pages Web sont générées en se basant sur le contenu de la base de données. Les questions et la séquence



des réponses proposées sont générées arbitrairement. L'utilisateur sélectionne la réponse et navigue à travers les pages du test en question en utilisant les éléments (les hyperliens et les boutons) de l'interface Web présentée. A la fin du test, l'étudiant clique sur le bouton intitulé « Grading » pour soumettre ses réponses et les stocker dans la base de données avant l'étape de l'évaluation [Mornar *et al.* 03].

### 3.3.2.3. L'évaluation

Au cours de cette étape, chaque réponse de l'étudiant est évaluée à correcte, incorrecte ou partiellement correcte et la base de données sera mise à jour en fonction de ces évaluations. Les résultats des tests concernant les leçons individuelles sont séparément affichés à l'étudiant. Le feedback est une source importante d'apprentissage. Pour cette raison, le système informe l'étudiant de l'exactitude de sa réponse pour chaque question. Notons que les résultats des tests sont utilisés pour noter les étudiants des cours : programmation, algorithmes et structures de données de la faculté du génie électrique et des calculs à Zagreb en Croatie (*Faculty of Electrical Engineering and Computing*) [Mornar *et al.* 03].

### 3.3.3. Le système AHyCo

AHyCo (*Adaptive Hypermedia Courseware*) est un système d'hypermédia éducatif basé sur le Web ayant pour objectif la mise à jour du modèle apprenant à travers les tests fournis pour son auto-évaluation [Mornar *et al.* 03]. AHyCo comprend :

- Un modèle du domaine décrivant la structure du domaine comme étant un ensemble de concepts (des leçons et des *tests*) ayant des relations pré-requises.
- Un modèle apprenant dans lequel est stockée la connaissance de l'étudiant concernant les concepts.
- Un modèle d'adaptation contient les règles définissant la façon avec laquelle le modèle du domaine et le modèle de l'apprenant seront combinés pour réaliser *l'adaptation de navigation*.

Dans AHyCo, nous trouvons les mini-tests ou les quiz utilisés pour modifier (orienter) la navigation dans un même module et les tests finaux utilisés pour pouvoir naviguer entre les modules. En effet, les mini-tests sont employés pour vérifier les connaissances de l'étudiant et pour corriger le modèle apprenant quand ce dernier navigue dans le module. Cependant, pour passer d'un module à un autre, il faudra que les connaissances de l'étudiant soient supérieures au niveau de connaissances minimum acceptable pour ce module défini par l'auteur ou le professeur, ou alors il faudra que l'apprenant réussisse le test final concernant le module en question [Mornar *et al.* 03].

Dans les sous sections suivantes, nous allons décrire le cycle de vie d'une question dans le système AHyCo tout en faisant une comparaison avec celui du système SAPZAS, les deux systèmes étant développés par la même équipe de recherche [Mornar *et al.* 03] à Zagreb en Croatie.

#### **3.3.3.1. La préparation**

Les types des questions et le processus de création dans AHyCo sont les mêmes que ceux dans SAPZAS. Toutefois, l'auteur doit créer le contenu des leçons et les modules ou le graphe basé sur les relations pré-requises entre les leçons. L'auteur doit aussi spécifier le poids de la leçon, le graphe des pré-requis pour les modules et le niveau de connaissances minimum acceptable pour chaque module [Mornar *et al.* 03].

#### **3.3.3.2. La livraison**

AHyCo a les mêmes composantes que le système SAPZAS, mais l'environnement des tests est étendu à un environnement d'apprentissage. Une leçon de AHyCo est présentée sous forme de pages Web en fonction des règles d'adaptation et des données enregistrées dans le modèle apprenant afin de l'adapter aux connaissances précédentes de l'étudiant. En bas de la page présentant une leçon, nous trouvons des hyperliens qui mènent à d'autres leçons ou à des mini-tests. Tous ces hyperliens sont fonctionnels, donc l'étudiant peut suivre celui qu'il veut. Toutefois, chaque lien est annoté avec une couleur selon le type du concept correspondant (concept principal, recommandé, non recommandé ou déjà appris). Une fois que l'étudiant aura appris toutes les leçons d'un

module, il pourra choisir quelques mini-tests pour s'auto-évaluer avant de faire le test final [Mornar *et al.* 03].

### 3.3.3.3. L'évaluation

Dans cette étape, le processus d'évaluation dépend du type de test (mini-test ou test final). Toutefois, dans tous les cas, le modèle apprenant est mis à jour. La connaissance de l'étudiant concernant la leçon a une valeur qui varie de -1 (il ne connaît pas la leçon) à 1 (il connaît la leçon). Au début, avant que l'étudiant ne fasse aucun test, toutes les leçons ont comme valeur initiale 0 dans le modèle apprenant. Cette valeur est augmentée ou diminuée en fonction de la réponse donnée (correcte ou incorrecte) et par rapport à l'ancienne valeur et le poids de la question dans le modèle du domaine. La leçon est considérée comme apprise si cette valeur est supérieure au niveau de connaissances minimum acceptable correspondant défini par l'auteur, sinon l'étudiant devrait revoir cette leçon plus tard après avoir appris les autres leçons du module. Dans le cas d'un test final, la connaissance du module a une valeur qui est déterminée en fonction des niveaux de connaissance et les poids de chaque leçon dans le module. Après avoir fait le test final d'un module, l'étudiant pourra :

- Réussir complètement le test et passer à un autre module.
- Réussir partiellement le test et passer à un autre module, mais il aura des leçons à revoir.
- Échouer dans le test, ce qui le poussera à le refaire s'il veut passer à un autre module.

Notons que ce système est utilisé au département des sciences de l'information de la faculté de philosophie à Zagreb en Croatie [Mornar *et al.* 03].

### 3.3.4. Le système SIETTE

SIETTE [Guzmán *et al.* 05] est un des plus récents systèmes basés sur le Web qui permettent d'évaluer l'apprenant. Il comprend un ensemble d'outils permettant aux

professeurs de créer des tests et d'analyser la performance de l'apprenant après son évaluation.

SIETTE fournit à l'apprenant des tests adaptatifs. Nous entendons par test adaptatif, un test qui s'adapte à la performance de l'apprenant au fur et à mesure que ce dernier avance dans son évaluation. Dans ce test, la question suivante posée à l'apprenant dépend de la réponse de ce dernier à la question courante : si sa réponse est correcte, la question suivante sera un peu plus difficile, sinon elle sera plus facile que la question courante.

#### **3.3.4.1. La préparation**

La préparation de ce genre de tests nécessite un effort spécial de la part du professeur car il doit se conformer à certaines règles qui garantissent la notion d'adaptabilité du test. Par exemple, il doit s'assurer que tous les items (ou questions) soient indépendants les uns des autres, c'est-à-dire qu'ils ne contiennent aucun indice permettant de répondre correctement à un autre item du même test. De plus, un professeur doit disposer d'un grand nombre de questions différentes avec des difficultés variables. Pour ces raisons, SIETTE comprend un environnement auteur qui permet de créer ce genre de tests et quelques outils permettant de s'assurer que l'ensemble des items vérifient les propriétés citées ci-haut.

A l'aide de l'éditeur de tests, un professeur peut définir plusieurs sujets. Chaque sujet peut-être divisé en sous sujets, etc. Les sous sujets sont reliés à leurs parents (sujets) par des relations d'agrégation. Chaque item peut-être assigné à un sous sujet particulier pour évaluer la connaissance de l'apprenant concernant ce dernier. Il peut être une question vrai/faux, à choix multiples avec une ou plusieurs réponses correctes... De plus, pour chaque item, le professeur peut définir le texte de la question, l'ensemble des réponses correctes, l'ensemble de réponses incorrectes possibles, etc. [Guzmán *et al.* 05].

### **3.3.4.2. La livraison**

Dans SIETTE, les apprenants peuvent faire des tests d'auto-évaluation ou des tests d'évaluation. Tous les tests, indépendamment de leur type, sont livrés à travers une interface Web, un item à la fois.

### **3.3.4.3. L'évaluation**

Si l'apprenant est entrain de faire un test d'auto-évaluation, après chaque question la correction correspondante lui sera affichée avec un feedback optionnel. Néanmoins, les auteurs du papier [Guzmán *et al.* 05] n'ont ni spécifié le type du feedback, ni donné un exemple du feedback.

## Chapitre 4 : Les systèmes de recommandation

De nos jours, la quantité d'informations disponibles en ligne devient de plus en plus importante. Les personnes qui naviguent sur le net dans le but de rechercher une information spécifique, ont du mal à se retrouver dans cette masse de connaissances importante et sont souvent submergées par les données mises à leur disposition, parmi lesquelles elles doivent faire leurs choix. Afin de remédier à tous ces problèmes, les systèmes de recommandation ont été mis au point depuis plusieurs années. Ils ont pour but de faciliter la tâche aux utilisateurs et de les aider à faire le choix qui leur convient. En effet, ils recommandent des items en se basant sur les besoins et les intérêts de l'utilisateur, déterminés à partir de son profil.

Dans ce chapitre, nous allons parler des techniques de recommandation les plus utilisées.

### 4.1. Introduction

Comme mentionné ci-haut, un système de recommandation permet à un utilisateur donné d'avoir un service adapté à son profil : il peut lui fournir des recommandations personnalisées ou le guider à choisir des produits (ou des services) qui lui sont utiles et qui répondent le mieux à son besoin [Burke 02]. Amazon.com est un des sites commerciaux les plus connus qui se basent sur des techniques de recommandation pour suggérer à l'utilisateur des items de façon personnalisée. En effet, Amazon.com se base sur les caractéristiques des items que l'utilisateur trouve intéressants, pour lui suggérer d'autres items semblables susceptibles de lui plaire. Citons également DIA [Yasmine *et al.* 04], un système qui se base sur la similarité des profils des étudiants avec d'autres usagers pour leur recommander des livres.

Dans la section suivante, nous allons décrire les différentes techniques de recommandation.

## 4.2. Les techniques de recommandation

En 2002, Burke [Burke 02] a classé les techniques de recommandation en fonction de plusieurs critères : les données sur lesquelles est basée la recommandation et l'utilisation qui en découle. En général, une technique de recommandation peut être considérée comme une fonction qui prend en entrée les données du système et les interactions de l'utilisateur avec ce dernier. Elle combine ces deux paramètres d'entrée et fournit, en sortie, des recommandations ou des suggestions. Ainsi, Burke a distingué cinq principales techniques de recommandation : le filtrage par contenu, le filtrage collaboratif, le filtrage démographique, le filtrage basé sur l'utilité et le filtrage basé sur les connaissances, appelé aussi filtrage à base de connaissances. Par contre, dans un récent article publié en 2005 [Adomivicius & Tuzhilin 05], les auteurs ont classé les techniques de recommandation en trois grandes catégories : le filtrage par contenu, le filtrage collaboratif et l'approche hybride qui utilise ces deux techniques de filtrage.

Toutefois, nous nous baserons dans ce chapitre sur l'approche de Burke car, à notre avis, elle est plus générale et couvre tous les systèmes de recommandation à l'aide des cinq techniques regroupées dans le tableau 4.1 ci-dessous :  $I$  est l'ensemble des « items » ou objets sur lesquels des recommandations peuvent être faites,  $U$  est l'ensemble des utilisateurs du système de recommandation. L'utilisateur  $u$  est l'utilisateur cible du système pour lequel les recommandations doivent être calculées et  $i$  est un des objets de l'ensemble  $I$ .

Tableau 4.1 : Les différentes techniques de recommandation [Burke 02]

Technique	Background	Informations nécessaires	Fonctionnement
<b>Collaborative</b>	Ratings from $U$ of items in $I$ .	Ratings from $u$ of items in $I$ .	Identify users in $U$ similar to $u$ , and extrapolate from their ratings of $i$ .
<b>Content-based</b>	Features of items in $I$ .	$u$ 's ratings of items in $I$ .	Generate a classifier that fits $u$ 's rating behaviour and use it on $i$ .
<b>Demographic</b>	Demographic information about $U$ and their ratings of items in $I$ .	Demographic information about $u$ .	Identify users that are demographically similar to $u$ , and extrapolate from their ratings of $i$ .
<b>Utility-based</b>	Features of items in $I$ .	A utility function over items in $I$ that describes $u$ 's preferences.	Apply the function to the items and determine $i$ 's rank.
<b>Knowledge-based</b>	Features of items in $I$ . Knowledge of how these items meet a user's needs.	A description of $u$ 's needs or interests.	Infer a match between $i$ and $u$ 's need.

Nous allons présenter chacune des techniques décrites dans ce tableau et en discuter les avantages et les limites.

#### 4.2.1. Le filtrage collaboratif

Le filtrage collaboratif est la plus connue et la plus utilisée des techniques de recommandation. Il identifie les similarités entre plusieurs utilisateurs en se basant sur les votes que ces derniers ont effectués sur des objets du système. Ces votes représentent le degré d'appréciation de l'objet par l'utilisateur [O'Sullivan *et al.* 02]. Concrètement, le vote se présente comme une évaluation, qui peut-être textuelle où le système sollicite l'utilisateur afin que ce dernier exprime son opinion par écrit. L'évaluation peut aussi être numérique, il s'agit d'attribuer une note au produit sur une échelle numérique (de 1 à 5, de 1 à 10, etc.).



Le filtrage collaboratif comporte trois étapes principales, à savoir :

- L'utilisateur effectue un vote, ou une évaluation concernant les différents objets proposés par le système.
- Le système fait une étude de similarité entre les différents utilisateurs en basant la fonction de similarité sur les votes. Si deux utilisateurs ont fourni des votes semblables vis-à-vis d'un produit donné, le système les considère comme proches, et prévoit des votes similaires de leur part sur d'autres produits.
- De nouvelles recommandations sont faites par le système, et ce en fonction des utilisateurs ayant le plus grand degré de similarité dans les votes.

Deux classes principales d'algorithmes de filtrage collaboratif ont été identifiées par Breese *et al.* [Breese *et al.* 98] : les algorithmes basés sur la mémoire (*memory-based*) et ceux basés sur les modèles (*model-based*).

#### 4.2.1.1. Algorithmes fondés sur la mémoire (*memory-based*)

Les algorithmes basés sur la mémoire utilisent les profils utilisateurs afin de générer de nouvelles prédictions.

Généralement, les algorithmes fondés sur la mémoire procèdent selon deux étapes :

- Rechercher les utilisateurs ayant voté les mêmes éléments de façon similaire à l'utilisateur cible.
- Calculer des prédictions sur les objets (ou items) que l'utilisateur cible n'a pas encore évalués, et ce en fonction des votes des utilisateurs similaires.

Deux techniques permettant de calculer la similarité entre deux usagers ont été citées par Breese *et al.* [Breese *et al.* 98]. Il s'agit de *la similarité vectorielle* et *la corrélation de Pearson*.

Le modèle vectoriel utilise une technique de recherche d'informations permettant de mesurer les degrés de similarités entre deux documents. Les documents en question

sont représentés par des vecteurs de mots-clés. La similarité entre deux vecteurs est calculée en vertu de la formule du cosinus. Cette technique est utilisée dans le cas du filtrage collaboratif dans la mesure où l'utilisateur joue le rôle d'un document  $A$ . Les noms des objets sont considérés comme des mots-clés et les votes  $v_{A,j}$  comme le poids donné par l'utilisateur  $A$  à l'objet  $j$ . Le poids est ici défini comme la fréquence de l'occurrence du mot-clé dans le document. Ainsi, la similarité vectorielle entre deux usagers  $A$  et  $B$  est calculée par la formule du cosinus présentée ci-bas (voir Formule 4.1):

$$\cosinus(A, B) = \sum_{j=1}^{|I|} \frac{v_{A,j}}{\sqrt{\sum_{j=1}^{|I|} v_{A,j}^2}} \times \frac{v_{B,j}}{\sqrt{\sum_{j=1}^{|I|} v_{B,j}^2}}$$

$|I|$  : nombre d'items communs entre  $A$  et  $B$  votées par  $v$ .

$v_{A,j}$  : vote de  $A$  pour l'item  $j$ .

$v_{B,j}$  : vote de  $B$  pour l'item  $j$ .

#### Formule 4.1 : Le calcul du cosinus

La corrélation de Pearson est une des techniques utilisées dans les systèmes de recommandation qui permettent de calculer la similarité entre deux usagers [Resnick *et al.* 94]. La corrélation entre deux personnes  $A$  et  $B$  est définie par la formule suivante :

$$w(A, B) = \frac{\sum_j (v_{A,j} - \bar{v}_A)(v_{B,j} - \bar{v}_B)}{\sqrt{\sum_j (v_{A,j} - \bar{v}_A)^2 \sum_j (v_{B,j} - \bar{v}_B)^2}}$$

$j$  : nombre d'objets ayant été voté à la fois par  $A$  et  $B$

$v_{A,j}$  : vote de  $A$  pour l'item  $j$ .

$\bar{v}_A$  : moyenne des votes de  $A$ .

#### Formule 4.2 : La corrélation de Pearson

La prédiction du vote sur l'objet  $j$  par l'utilisateur  $A$  est calculée par la formule suivante:

$$p_{A,j} = \bar{v}_A + \frac{\sum_{i=1}^n w(A,i)(v_{i,j} - \bar{v}_i)}{\sum_{i=1}^n |w(A,i)|}$$

$n$  : nombre d'utilisateurs présents dans le voisinage de  $A$ , ayant déjà voté l'objet  $j$ .

$v_{i,j}$  : vote de l'utilisateur  $i$  pour l'objet  $j$ .

$\bar{v}_i$  : moyenne des votes de l'utilisateur  $i$ .

**Formule 4.3 :** Le calcul de la prédiction de l'utilisateur  $A$  pour l'objet  $j$

#### 4.2.1.2. Algorithmes à base de modèles (*model-based*)

Cette technique est basée sur les profils des utilisateurs afin d'estimer ou d'apprendre un modèle, qui sera utilisé par la suite pour faire des prédictions pour un utilisateur donné. L'algorithme à base de modèles construit un modèle de l'utilisateur en se basant sur ses votes antérieurs : il génère la probabilité qu'un utilisateur  $A$  effectue un vote  $v_{A,j}$  sur l'objet  $j$ . La construction du modèle est généralement réalisée à l'aide des algorithmes d'apprentissage machine comme le modèle des clusters et le modèle de réseaux Bayésiens.

Dans ce mémoire, nous avons essentiellement travaillé avec les algorithmes basés sur la mémoire qui ont été décrits avec plus de détails dans la section précédente.

#### 4.2.1.3. Les limites du filtrage collaboratif

Les limites majeures du filtrage collaboratif sont dues au calcul de corrélation entre les utilisateurs. Celui-ci étant basé sur les votes donnés à des objets communs, deux types de problèmes peuvent survenir [Burke 02]:

- Le problème du « nouvel utilisateur » réside dans le fait que ce dernier a peu ou pas de votes, ce qui rend difficile sa comparaison aux autres utilisateurs du système. Cet utilisateur voit ses chances d'avoir des recommandations pertinentes, amoindries.
- Le problème du « nouvel objet » provient du fait que ce dernier n'a pas reçu beaucoup de votes. Sa recommandation par le système devient moins probable.

Ces deux problèmes sont connus sous le nom de démarrage à froid ou *Cold Start*.

Dans le cas d'un système contenant beaucoup d'objets et un nombre d'utilisateurs relativement petit, un des problèmes qui peuvent survenir est celui de l'espacement ou *Sparsity problem* [Sarwar *et al.* 01]. Ce problème découle du fait que peu d'utilisateurs peuvent voter les mêmes objets et ainsi, du manque de corrélation entre les utilisateurs qui peut en résulter.

Un système utilisant le filtrage collaboratif a tendance à recommander les produits les plus populaires. Ainsi, si un utilisateur a des goûts différents de ceux de la masse des autres usagers, il devient difficile de lui générer de bonnes recommandations. Il s'agit du problème de *Gray Sheep* [Claypool *et al.* 99].

Le problème de performance, notamment pour les algorithmes basés sur les utilisateurs, est à évoquer. La complexité du temps de calcul augmente avec le nombre d'utilisateurs et le nombre d'objets [Sarwar *et al.* 01].

#### **4.2.2. Le filtrage basé sur le contenu**

Dans un système qui utilise le filtrage basé sur le contenu, les recommandations sont effectuées en tenant compte des objets que l'utilisateur a apprécié auparavant. Le système lui propose ainsi des objets au contenu similaire. Ceci nécessite la représentation des objets de sorte qu'ils puissent être comparés. La description des objets doit ainsi contenir des caractéristiques semblables.

Le filtrage basé sur le contenu présente comme principal avantage le fait qu'il est facile à implémenter.

Les limitations d'un tel mode de filtrage résident en premier lieu dans la représentation des objets. Il faut pouvoir comparer de façon adéquate les objets recommandés avec ceux qui sont enregistrés dans le profil de l'utilisateur. En deuxième lieu, il fournit souvent des recommandations statiques vu que l'ensemble de caractéristiques ne change pas. De plus, si le contenu de l'objet est vague, ce type de filtrage peut s'avérer très inefficace.

### **4.2.3. Le filtrage démographique**

Les systèmes de recommandation basés sur le filtrage démographique visent la classification des clients par catégories, et ce en tenant compte de leurs données démographiques. Les recommandations sont ainsi faites à partir de ces catégories.

Ce type de filtrage n'impose pas aux utilisateurs d'évaluer un certain nombre d'items avant de pouvoir générer des recommandations significatives. Il ne demande pas non plus de garder l'historique des votes des utilisateurs, ce qui permet d'éviter les inconvénients de ce système de votes, à savoir, le démarrage à froid.

Cependant, les informations démographiques nécessaires, peuvent s'avérer difficiles à recueillir. Il s'agit en premier lieu du problème de temps pris par l'utilisateur pour fournir ces informations, mais aussi de la nature de confidentialité des ces dernières.

### **4.2.4. Le filtrage basé sur l'utilité**

Le filtrage basé sur l'utilité a comme caractéristique qu'il nécessite qu'une fonction d'utilité soit attribuée à chaque utilisateur. Un système basé sur cette technique doit ainsi calculer l'utilité de chaque item avant de pouvoir faire des recommandations à l'utilisateur. Les fonctions d'utilité utilisent des informations sur les items. La principale difficulté est de trouver la fonction d'utilité. Mais l'avantage est que cette fonction peut ne pas dépendre de l'objet.

#### **4.2.5. Le filtrage à base de connaissances**

Il s'agit là d'une approche qui prend en considération les besoins et les préférences des utilisateurs pour en déduire des recommandations. La particularité est que les connaissances dont il est question sont « fonctionnelles », c'est-à-dire que ces systèmes connaissent la façon dont un objet particulier peut satisfaire un utilisateur donné. Par exemple, le système « Entrée » [Burke 02] utilise ses connaissances sur la cuisine pour déduire des similarités entre des restaurants.

Les systèmes de recommandation basés sur le filtrage à base de connaissances sont faciles à implémenter et ne nécessitent pas d'espace de stockage additionnel.

Les inconvénients de tels systèmes sont qu'ils s'adaptent difficilement à un changement brusque de la connaissance, comme le nom d'un objet.

### **4.3. Les systèmes hybrides**

Les systèmes de recommandation hybrides consistent à combiner, au moins, deux techniques de filtrage afin d'améliorer la performance du système, et ce en essayant de contourner les problèmes posés par une seule technique [Burke 02]. Dans la partie suivante, nous définissons les différentes techniques hybrides.

#### **4.3.1. Pondération (Weighted)**

Les résultats ou les votes qui sont générés par différentes techniques de recommandation sont combinés de façon à ce qu'une seule recommandation en résulte.

#### **4.3.2. Commutation (Switching)**

C'est une technique qui permet de faire le choix d'un modèle de recommandation entre plusieurs, et ce en se basant sur plusieurs critères. La détermination de la technique appropriée dépend de la situation. Le système se doit alors de définir les critères de

commutation, ou les cas où l'utilisation d'une autre technique est recommandée. Ceci permet au système de connaître les points forts et les points faibles des techniques de recommandation qui le constituent [Burke 02].

### **4.3.3. Technique mixte (Mixed)**

Cette technique est apte à donner à l'utilisateur des recommandations qui proviennent de plusieurs techniques [Smyth & Cotter 99]. L'utilisation simultanée du filtrage collaboratif et du filtrage par contenu est un exemple de technique mixte. Cette utilisation simultanée permet d'éviter les problèmes posés par le filtrage collaboratif, à savoir, le démarrage à froid. Aussi, le filtrage basé sur le contenu permet d'obtenir des recommandations sur de nouveaux objets, et ce en se basant sur leurs descriptions respectives.

### **4.3.4. Combinaison de caractéristiques (Features combination)**

Il s'agit d'une technique où les caractéristiques des informations qui sont fournies par les différentes méthodes de recommandation sont combinées, afin de permettre l'utilisation d'une technique unique sur l'ensemble des données. Le filtrage collaboratif utilise les votes des utilisateurs comme source de données. Ces données, lorsque ajoutées aux informations utilisées par le filtrage basé sur le contenu, peuvent produire des recommandations fiables [Burke 02].

### **4.3.5. Cascade (Cascade)**

Cette technique hybride se fait selon deux techniques :

- Une première technique permet de générer un ensemble de candidats potentiels.
- Une deuxième technique permet de raffiner les recommandations.

Cette méthode a pour avantage que la deuxième technique peut servir uniquement dans le cas de figure où les recommandations générées par la première nécessitent une discrimination additionnelle. Si la première technique génère peu de recommandations,

ou si ces recommandations sont ordonnées afin de permettre une sélection rapide, la deuxième technique ne sera pas utilisée non plus [Burke 02].



## Chapitre 5 : Conception et méthodologie

Au cours de ce chapitre, nous allons rappeler brièvement les objectifs de notre système, présenter notre approche, décrire l'architecture de notre système UMAKE et enfin, expliquer la méthodologie adoptée pour atteindre nos objectifs.

### 5.1. Objectifs

Parfois, dans un contexte d'évaluation, un apprenant n'arrive pas à comprendre la question qui lui est posée. De ce fait, il a tendance à répondre incorrectement. Pourtant, s'il l'avait bien comprise, il aurait pu être évalué de façon équitable et aurait pu avoir une meilleure chance de répondre correctement. Nous avons voulu focaliser nos efforts sur les moyens permettant à l'apprenant de comprendre la question [Mabrouk & Aïmeur 05]. Pour répondre à cet objectif, nous avons développé une approche qui consiste à recommander de l'aide sur chaque question. Cette aide est recommandée en se basant sur une technique hybride appelée la « commutation » [Burke 02]. Elle se base sur le filtrage collaboratif et le filtrage à base de connaissances. En outre, l'aide est adaptée en se basant sur plusieurs critères que nous verrons en détails plus tard dans ce chapitre.

### 5.2. Approche de UMAKE

Nous présentons dans cette partie le fonctionnement global et les techniques de recommandation utilisées dans notre système.

#### 5.2.1. Fonctionnement général

UMAKE opère selon deux phases essentielles : une phase d'enregistrement et une phase d'auto-évaluation.

Afin de pouvoir utiliser UMAKE, tous les utilisateurs doivent s'enregistrer et créer un compte. La *phase d'enregistrement* comporte deux étapes. Lors de la première étape, l'utilisateur doit fournir des informations générales comme son nom, son sexe, son adresse email et son mot de passe. Lors de la deuxième étape de l'enregistrement, l'utilisateur se verra proposer un questionnaire psychotechnique [URL 1]. Ce questionnaire comporte 14 questions auxquelles chaque nouvel utilisateur se doit de

répondre afin que UMAKE détermine son style d'apprentissage selon la procédure indiquée [URL 1]. Lorsque l'utilisateur voudra ultérieurement utiliser le système, il se connectera à l'aide de son nom d'utilisateur (son adresse email) et de son mot de passe. Il n'aura plus à remplir de nouveau le formulaire, ni à répondre au test psychotechnique.

Quand un utilisateur existant se connecte à UMAKE (avec son nom et son mot de passe), il sera acheminé vers la page principale qui est le cœur du système. En effet, à partir de cette page, l'utilisateur a accès aux différents sujets disponibles pour son auto-évaluation. De plus, dans cette page, il peut accéder à plusieurs « outils membre » qui lui permettent de changer son mot de passe, de refaire le test psychotechnique, de consulter des statistiques globales, de consulter son voisinage (voir section 5.2.3) ou encore de consulter de l'aide.

Quant à la *phase d'auto-évaluation*, celle-ci commence quand l'utilisateur choisit un sujet sur lequel il veut être auto-évalué parmi ceux disponibles dans le système. Avant de commencer le test, l'utilisateur doit choisir le nombre de questions constituant ce test et peut, s'il le veut, consulter un tutorial pour chaque type d'aide. Ensuite, nous lui présentons le test, une question à la fois. Toutes les questions sont à choix multiple. Pour chacune, nous lui proposons trois aides qui correspondent aux trois styles d'apprentissage principaux (voir section 5.3.1) : auditif, visuel et kinesthésique. Les personnes auditives comprennent mieux les idées et les concepts en *écoutant* les informations [URL 2]. Les personnes visuelles préfèrent utiliser des *images*, des *couleurs* et des *cartes* pour organiser les informations et communiquer avec les autres [URL 2]. Quant aux personnes kinesthésiques, elles préfèrent apprendre *par elles mêmes* en essayant différentes approches [URL 2]. Afin de fournir à l'utilisateur une aide personnalisée, UMAKE lui recommandera une de ces trois aides. Cependant, le système le laisse libre d'utiliser l'aide qui lui convient le mieux. S'il se trouve en difficulté, il peut utiliser une ou plusieurs aide(s) et il se verra parfois obligé d'évaluer les aides utilisées (selon un mécanisme qui sera décrit ultérieurement dans la section 5.3.2) avant de passer à la prochaine question. UMAKE enregistre ces évaluations et les utilise afin d'améliorer et raffiner ses recommandations futures. A la fin du test, UMAKE lui présente un tableau récapitulatif de sa performance durant cette session d'auto-évaluation, à partir duquel il

peut voir les solutions aux questions, et un autre tableau récapitulatif de sa performance globale. Notons toutefois que UMAKE présente les solutions aux questions pour chaque apprenant de façon *personnalisée*. En effet, lors de cette étape, il peut connaître (voir chapitre 6 : Implémentation et validation) :

- Quelles aides il a utilisées pour répondre à la question courante.
- Quelle aide nous lui avons recommandé pour cette question.
- Quelle méthode de recommandation nous avons utilisé et la raison pour laquelle nous avons employé cette méthode.

Nous lui présentons aussi son style d'apprentissage tel qu'il a été observé et interprété par le système en fonction de son comportement (voir section 5.3.3). Ce style est un mélange des trois styles principaux (auditif, visuel et kinesthésique). Il peut consulter une brève définition et quelques informations caractérisant ce style. Il donnera par la suite son avis positif ou négatif en se basant sur ces informations (voir section 5.3.1).

## **5.2.2. Fonctions principales accessibles aux utilisateurs**

A partir de la description faite précédemment, nous pouvons dégager les fonctions principales du système. Chaque utilisateur peut :

- Changer son mot de passe quand il le veut par mesure de sécurité.
- Refaire le test psychotechnique en se concentrant davantage avant de répondre à chaque question si le style d'apprentissage déterminé initialement ne l'a pas trop convaincu.
- Consulter des statistiques générales reflétant sa performance globale dans tous les tests faits antérieurement.
- Consulter les usagers constituant son voisinage et les différentes mesures de similarité avec chacun d'eux.
- Consulter l'aide qui lui explique chacune des options qui se présentent à lui dans la page principale.
- Choisir le nombre de questions qui composeront le test d'auto-évaluation.
- Accéder au petit tutorial proposé pour savoir manipuler chaque type d'aide.

- Faire des tests d'auto-évaluation sur les différents sujets disponibles.
- Utiliser une ou plusieurs des aides proposées pour chaque question d'un test d'auto-évaluation.
- Fournir des évaluations pour les aides utilisées.
- Consulter la solution finale et les statistiques à la fin du test.
- Mettre à jour son profil en donnant son accord ou son désaccord pour le nouveau style d'apprentissage présenté par le système.

### 5.2.3. Techniques de recommandation

La recommandation est faite selon une technique hybride appelée *la commutation* qui combine *le filtrage collaboratif* et *le filtrage à base de connaissances*. En effet, afin de fournir une bonne recommandation, nous commutons entre ces deux techniques en fonction de la situation.

La première technique implique la prise en compte des autres utilisateurs du système. Elle consiste à accumuler leurs votes sur les différents items (aides), identifier ceux ayant voté les items en commun et offrir des recommandations en se basant sur des comparaisons faites entre l'utilisateur cible et ces utilisateurs.

Dans la deuxième technique, la recommandation des items est basée sur des inférences concernant les préférences de l'utilisateur et ses besoins.

Nous verrons dans la section 5.4.3 les circonstances dans lesquelles nous appliquons chacune de ces deux techniques.

## 5.3. Méthodologie de UMAKE

Comme nous l'avons déjà mentionné, UMAKE est un système basé sur le web [URL 5] qui permet de fournir *dynamiquement* un quiz aux apprenants et de les aider à comprendre les questions posées en leur *recommandant* les aides appropriées. Les aides recommandées sont *adaptées* en se basant sur plusieurs critères (voir section 5.4.3) :

- Le style d'apprentissage de l'utilisateur : dans le cas où son style n'est pas simple (par exemple auditif, visuel ou kinesthésique), mais est une composition de deux ou de trois de ces styles principaux (par exemple auditif et visuel, auditif et kinesthésique, visuel et kinesthésique ou encore auditif, visuel et kinesthésique), nous lui recommandons l'aide qui correspond au style d'apprentissage dominant.
- Les préférences de l'utilisateur comme les différents types d'aide qu'il a aimés pour les questions précédentes.
- Les préférences des autres utilisateurs ayant des goûts similaires (ceux qui ont répondu aux mêmes questions et voté pour les mêmes aides).
- La performance de l'utilisateur en fonction des différents types d'aide qu'il a utilisés précédemment.

L'adaptation dans un système qui assiste les apprenants dans leur auto-évaluation prend en considération deux aspects : l'aspect éducationnel et l'aspect technologique.

**L'aspect éducationnel** touche essentiellement deux volets : la planification du contenu ou la *sélection* des questions et des aides, et sa livraison ou *la présentation* de ces questions et de ces aides à l'utilisateur (voir chapitre 3).

Nous voulons souligner que le processus de *sélection des questions* n'est pas aléatoire : bien que l'objectif principal de UMAKE reste la recommandation à l'utilisateur d'une aide appropriée qui l'aide à mieux comprendre la question posée, nous nous assurons aussi du fait que les questions incluses dans son test ne soient pas choisies au hasard. En effet, UMAKE génère *dynamiquement* les quiz aux utilisateurs. Pour ce faire, nous avons développé un processus de sélection qui se déclenche instantanément dès que l'utilisateur choisit le nombre de questions qui constitueront le test qu'il va passer. Ce processus détermine l'ensemble des questions à inclure dans le test en utilisant une stratégie particulière. Cette stratégie permet d'avoir un ensemble contenant un mélange de questions « nouvelles » et « anciennes » à l'utilisateur. Nous voulons dire par questions « nouvelles », les questions que l'utilisateur n'a jamais eu dans un test auparavant. Tandis que les questions « anciennes », sont les questions que l'utilisateur a

eues au moins dans un test lors d'une de ses précédentes sessions d'auto-évaluation. Notre stratégie permet de choisir un sous-ensemble des questions « anciennes », auxquelles l'utilisateur a échoué, autrement auxquelles il a répondu incorrectement, et un sous-ensemble des questions « nouvelles ». Nous avons inclus le premier sous-ensemble en pensant que l'utilisateur sera plus motivé à se rattraper dans ces questions comme toute personne qui sera motivée et contente lorsqu'on lui donne la chance de se reprendre dans une de ses erreurs dans la vie. En se basant sur les critères cités ci-dessus, l'ensemble final des questions sera constitué d'un certain pourcentage  $P_{Anciennes}$  de questions « anciennes » et d'un certain pourcentage  $P_{Nouvelles}$  (sachant que  $P_{Nouvelles} = 100\% - P_{Anciennes}$ ) de questions « nouvelles ». Nous avons initialisé  $P_{Anciennes}$  à 20%, et ainsi,  $P_{Nouvelles}$  sera initialisé à 80%. Le choix de ces valeurs est totalement arbitraire et nous pouvons les modifier après certains tests du système s'il s'avère nécessaire.

A chaque question sont associés trois types d'aide qui correspondent aux principaux styles d'apprentissage (voir section 5.3.1). Donc, le processus de sélection des questions engendre aussi la sélection des aides associées.

Concernant *la présentation de ces questions et de ces aides*, nous avons livré le test à l'apprenant une question à la fois pour plusieurs raisons, dont les plus importantes sont :

- Alléger la page vue la richesse du matériel éducationnel présenté (le texte de la question, les réponses proposées, les trois types d'aide associés...).
- Permettre à l'utilisateur de rester concentré au maximum sur chaque question avant de répondre et maximiser les chances d'avoir des résultats naturels et significatifs. En effet, si nous lui avons présenté tout le quiz en une seule page, il est susceptible qu'il soit perturbé par les questions qu'il jugera difficiles dès le départ, ou encore par la consultation de toutes les aides. Ainsi, il ne pourra pas faire le test convenablement et naturellement (sans parler des facteurs externes sur lesquels nous n'avons aucun contrôle et qui peuvent, aussi, le perturber).

Quant à l'**aspect technologique**, il concerne la technologie employée qui nous a permis de réaliser l'aspect éducationnel décrit ci-dessus et les différents objectifs du système comme l'emploi des principales techniques d'adaptation telles que *l'adaptation de la présentation* et *l'adaptation de la navigation*. Nous allons les présenter de façon détaillée dans le chapitre 6.

### 5.3.1. Style d'apprentissage

Nous avons utilisé deux méthodes pour déterminer le style d'apprentissage de l'utilisateur et ses préférences concernant la façon d'apprendre et d'assimiler : les questionnaires et l'observation de son comportement et de ses choix durant son interaction avec le système.

Dans UMAKE, le style d'apprentissage de l'utilisateur est initialisé à travers la soumission d'un *questionnaire psychotechnique* développé par Neil. D. Fleming avec Charles. C. Bonwell [URL 1]. Ainsi, quand l'utilisateur se connecte pour la première fois au système, il répond à un questionnaire lors de la deuxième étape de la phase d'enregistrement tel qu'il a été décrit dans la section 5.2.1. Ce questionnaire sert à déterminer son style d'apprentissage qui sera classé dans une des catégories suivantes (auditive, visuelle, kinesthésique, auditive et visuelle, auditive et kinesthésique, visuelle et kinesthésique, auditive visuelle et kinesthésique). Le système lui présente le style d'apprentissage déterminé et lui procure la possibilité de consulter les informations qui caractérisent chaque style. Par exemple, les apprenants kinesthésiques sont les apprenants qui apprennent par eux-mêmes à travers le déplacement, le toucher... Ils apprennent mieux en explorant activement le monde physique autour d'eux [URL 2].

A la fin de chaque test, UMAKE présente à l'utilisateur son style d'apprentissage évolué tel qu'il a été perçu par le système en fonction de *ses choix*, de *ses évaluations* données à chaque type d'aide utilisée et de *sa performance* (voir section 5.4.3). Ce dernier aura, ainsi, la possibilité d'exprimer son avis concernant le style proposé : s'il est d'accord avec ce style ou non. En fonction de cet avis (positif ou négatif), le profil sera

mis à jour et les recommandations futures seront basées sur ce nouveau style d'apprentissage.

### 5.3.2. Évaluation des aides

Pour déterminer la façon avec laquelle un utilisateur doit voter dans UMAKE ou plus précisément les moments pendant lesquels il doit voter, nous avons développé un mécanisme de vote dépendant de plusieurs variables. Ce mécanisme doit être *un compromis* entre ces deux faits :

- Moins l'utilisateur vote pour les différents types d'aide et plus souvent il doit donner des évaluations.
- Moins un type d'aide donné a des évaluations et plus souvent les utilisateurs doivent lui attribuer des notes.

Après mûre réflexion, nous avons décrit le mécanisme sous la forme d'une condition  $C$  détaillée dans la Formule 5.1 ci-dessous :

$$C : x - y_{ij} \geq t$$

$x$  : moins la personne vote, plus  $x$  augmente.

$y_{ij}$  : plus les utilisateurs votent pour l'aide  $j$  de la question  $i$ , plus  $y_{ij}$  augmente.

$t$  : une valeur seuil qui sera définie dans la Formule 5.2

**Formule 5.1** : La condition  $C$  définissant la façon de voter dans UMAKE

Au départ, les variables  $x$  et  $y_{ij}$  seront initialisées à zéro pour tout  $i$  et  $j$ . Imaginons la situation du tout premier utilisateur qui se connecte pour la première fois sur notre système,  $y_{ij}$  sera égale à zéro pour toutes les aides qu'il va utiliser et  $x$  augmentera chaque fois qu'il ne vote pas pour les aides qu'il a utilisées. Par exemple, après deux questions pendant lesquelles l'apprenant n'a pas voté, la condition  $C$  sera



évaluée comme suit à :  $2 - 0 \geq t$ . Donc, si nous souhaitons que l'utilisateur vote toujours après deux questions au cours des premières utilisations du système, nous pourrions par exemple fixer la valeur de  $t$  à 2.

Mais, nous pensons aussi que la valeur de  $t$  ne doit pas rester à deux pour toujours, car au fur et à mesure que plusieurs apprenants utilisent le système, nous allons accumuler de plus en plus de votes pour que UMAKE fonctionne correctement et efficacement. Ceci aura pour conséquence le fait qu'à partir d'un certain moment les apprenants n'auront plus à voter aussi souvent qu'ils le faisaient au début de l'utilisation du système parce que le système apprend au fur et à mesure. Par conséquent, la valeur de la variable  $t$  doit s'adapter et évoluer au fur et à mesure de l'utilisation du système et dépendra du nombre de votes déjà accumulés pour les différents types d'aide que l'utilisateur va pouvoir utiliser durant son test. Suite à cette réflexion, nous avons élaboré la Formule 5.2 :

$$t = c_1 + \left[ \frac{\sum_{q=1}^n \sum_{k=1}^3 y_{qk}}{c_2} \right]$$

$c_1$  : une constante qui définit la valeur initiale de  $t$ .

$c_2$  : une constante qui va influencer à quel point le nombre de votes accumulés dans le système va changer la fréquence de demande de vote.

$y_{qk}$  : la variable  $y$  de la  $k$ ème aide de la  $q$ ème question ( $q$  variant entre 1 et  $n$  et  $n$  étant le nombre des questions présentes dans le test).

$\sum_{q=1}^n \sum_{k=1}^3 y_{qk}$  : c'est la somme des valeurs  $y_{qk}$  de toutes les aides présentes dans le test courant.

**Formule 5.2** : la détermination de la valeur de la variable  $t$

Nous allons continuer, dans ce qui suit, le même raisonnement adopté pour notre exemple précédent. Alors, dans ce cas, nous définissons les valeurs des constantes  $c_1$  et  $c_2$  comme suit :

- $c_1 = 2$ , ce qui signifie que lors des premières utilisations du système, les apprenants devront voter toutes les deux questions en moyenne.
- $c_2 = 10$ , ce qui signifie que si les aides présentées dans ce test ont accumulé au moins 10 votes,  $t$  augmentera de 1, la condition  $C$  sera réévaluée et l'apprenant courant devra voter moins souvent qu'il devait le faire si le nombre des votes accumulés était moins que 10. Le fait de fixer  $c_2$  à 10 correspond à tenir compte des votes accumulés et changer la condition par tranche de 10.

Cependant, pour rendre ce mécanisme de vote plus *robuste* et *moins déterministe*, nous avons introduit la condition  $C$  dans un mécanisme probabiliste. Pour cela, nous avons conçu la Formule 5.3 qui montre comment nous calculons la probabilité de demander à l'apprenant de voter pour une aide :

$$Pr\ obaDemanderVote = \frac{1}{4 + \left\lfloor \frac{NbVotesTotal}{1000} \right\rfloor} + \begin{cases} \frac{1}{2} & \text{si } x - y \geq t \\ 0 & \text{sinon} \end{cases}$$

**Formule 5.3 :** Le calcul de la probabilité de demander un vote à l'apprenant dans  
UMAKE

Nous remarquons que la probabilité de demander un vote à l'apprenant augmente si la condition  $C$  est vérifiée. La variable  $t$  est déjà calculée à l'aide de la Formule 5.2.

Cette façon de faire consiste à générer d'abord un nombre aléatoire entre 1 et 100. Ensuite, si ce nombre est inférieur ou égal à la probabilité de demander un vote, nous allons demander à l'apprenant de fournir un vote. Dans le cas contraire, il ne sera pas obligé de voter. Statistiquement, les chances de donner des votes seront ainsi *mieux réparties*. De plus, cette façon de faire permet d'éviter à l'apprenant de « voter toujours au même endroit ». Ainsi, même si celui-ci connaît bien les valeurs des variables  $x$ ,  $y$  et  $t$ , cela ne lui permettra pas de prédire à coup sûr à l'avance les moments au cours desquels il va devoir voter.

Nous remarquons aussi que la probabilité de base  $\frac{1}{4 + \left\lfloor \frac{NbVotesTotal}{1000} \right\rfloor}$  dépend du

nombre total de votes. En effet, cette probabilité tient compte des votes accumulés dans le système et diminue par tranche de 1000. Par exemple, si nous avons accumulé 1000 votes, nous allons lui demander de voter seulement une fois toutes les cinq questions en moyenne au lieu d'une fois toutes les quatre questions. Nous avons choisi d'utiliser le chiffre 4 dans cette probabilité de base parce que nous avons voulu que l'apprenant vote une fois toutes les quatre questions quand la condition  $C$  n'est pas vérifiée. De plus, les quiz peuvent comporter trois, cinq ou sept questions, donc quatre est un chiffre moyen pour les trois quiz.

En quelque sorte, nous pouvons donc dire que le mécanisme de demande de vote dans UMAKE est *probabiliste et non pas déterministe*.

## 5.4. Architecture de UMAKE

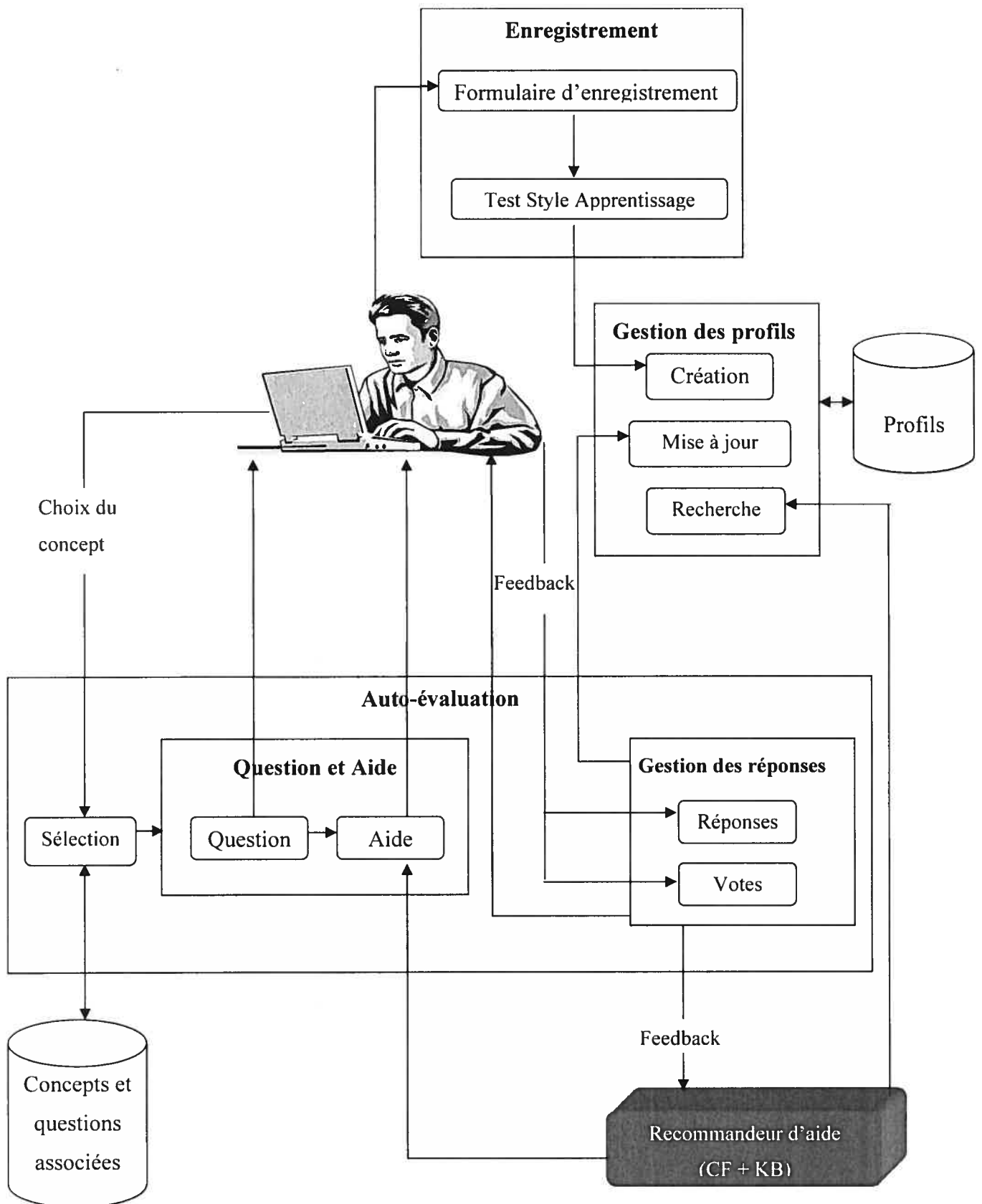
Dans cette section, nous présentons l'architecture générale de UMAKE (voir Figure 5.1) qui se compose de plusieurs modules comme le module de *gestion des profils* et le module de *gestion des réponses*. La fonction principale est réalisée par le *recommandeur d'aide* qui veille à recommander l'aide appropriée à l'utilisateur. Comme nous le voyons sur cette architecture et comme nous l'avons déjà expliqué dans la section

5.2.1, le système procède par deux phases : la phase d'*enregistrement* et la phase d'*auto-évaluation*.

Dans la partie qui suit, nous allons définir les différents composants et modules nécessaires au fonctionnement de UMAKE.

Notons que dans la Figure 5.1, le recommandeur d'aide utilise deux techniques :

- Le filtrage collaboratif CF.
- Le filtrage à base de connaissances KB.



**Figure 5.1 :** Architecture générale de UMAKE

### 5.4.1. Module de gestion des profils

Le module de gestion des profils permet de manipuler les profils des utilisateurs. Les principales fonctions assurées par ce module sont :

- La *création* des profils des utilisateurs lors de la phase de l'enregistrement où ils doivent remplir un simple formulaire avec leurs données démographiques et répondre à un test qui sert à déterminer leurs styles d'apprentissage. La phase d'enregistrement est effectuée une seule fois pour chaque nouvel utilisateur. Les anciens utilisateurs déjà enregistrés sont exemptés et passent directement à la phase d'auto-évaluation.
- La *mise à jour* des profils des utilisateurs déjà enregistrés :
  - S'ils sont en phase d'auto-évaluation, ce module ajoute les questions auxquelles ils ont déjà répondu, leurs réponses, les différentes aides utilisées, les votes correspondants et le style d'apprentissage évolué déterminé par le système en fonction de ces nombreux paramètres.
  - S'ils sont en train de manipuler les « outils membres », ce module peut aussi changer le mot de passe et modifier le style d'apprentissage initial enregistré lors de la phase d'enregistrement.
- La *recherche* des profils lors de la phase d'auto-évaluation et plus précisément lors de la recommandation. Cette recherche est généralement effectuée par le recommandeur d'aide selon plusieurs critères que nous verrons dans la section 5.4.3.

### 5.4.2. Module de gestion des réponses

Le module de gestion des réponses fonctionne généralement lors de la phase de l'auto-évaluation. En effet, il enregistre les réponses de l'utilisateur pour chaque question et les évaluations qu'il a fournies aux aides utilisées et les envoie au :

- Module de gestion des profils pour faire la mise à jour du profil de l'utilisateur en question.

- Recommandeur d'aide pour qu'il prenne compte de ces données et les utilise pour raffiner et améliorer ses recommandations.

### 5.4.3. Recommandeur d'aide

Le recommandeur d'aide est la pièce maîtresse de notre système. En effet, celui-ci, comme son nom l'indique, exécute la fonction principale du système qui est la recommandation des aides appropriées pour les utilisateurs. Il se base sur les données enregistrées dans le profil de l'apprenant, ainsi que son comportement observé lors des sessions d'auto-évaluation pour déterminer l'aide à recommander. En fait, ce comportement permet de diagnostiquer certaines des caractéristiques de l'apprenant comme ses préférences pour les aides proposées. Il constitue la base pour les prochaines recommandations. Au fur et à mesure que l'utilisateur interagit avec le système, la recommandation est adaptée en fonction de ses actions et de son pourcentage de succès dans les tests.

Afin de réaliser convenablement cette fonction, le recommandeur d'aide emploie une technique de recommandation hybride qui *commute* entre deux techniques de filtrage, à savoir : le filtrage collaboratif et le filtrage à base de connaissances. Généralement, c'est la première technique qui est appliquée en premier lieu, mais l'emploi de la technique appropriée dépend toujours de la situation. Nous avons défini les critères de commutation et nous présenterons les cas d'utilisation de chaque technique dans les deux sections suivantes.

#### ❖ Pseudo-code de l'algorithme hybride HY :

Dans cette partie, nous présentons un exemple de pseudo-code pour notre algorithme hybride HY. Toutes les étapes de l'algorithme du filtrage collaboratif CF et de l'algorithme à base de connaissances KB, employées dans HY, seront détaillées dans les sections 5.4.3.1 et 5.4.3.2 au fur et à mesure que nous décrivons ces deux algorithmes.

Notons que cet algorithme est exécuté avant l'affichage de chaque question  $q$  du quiz à l'utilisateur  $u$  afin déterminer l'aide à recommander pour  $q$ .

```

Si utilisateur u n'a fait aucune question jusqu'à maintenant
|
| Recommander l'aide qui correspond à son style d'apprentissage
| initial dominant selon le filtrage à base de connaissances KB.
|
Sinon
|
| Appliquer CF pour calculer les prédictions correspondantes aux trois
| types d'aide.
|
| Si q est nouvelle
|
| | Si les deux plus hautes prédictions sont égales
| |
| | | Appliquer KB pour recommander l'aide qui a la plus haute
| | | utilité.
| |
| | Sinon
| |
| | | Recommander l'aide qui a la plus haute prédiction.
| |
| | Sinon
| |
| | | Comparer les prédictions calculées avec les anciens votes que
| | | u a donné aux différentes aides lors des sessions d'auto-
| | | évaluation précédentes et déterminer les trois maxima.
| | |
| | | Si les deux plus hautes valeurs de ces maxima sont égales,
| | |
| | | | Appliquer KB pour recommander l'aide qui a la plus haute
| | | | utilité.
| | |
| | | Sinon
| | |
| | | | Recommander le type d'aide qui correspond à la plus haute
| | | | valeur de ces maxima.
| | |
| | | Fin si
| |
| | Fin si
|
| Fin si
|
Fin si

```

#### 5.4.3.1. Filtrage collaboratif

Lors de ce filtrage, nous récoltons les différents votes de l'utilisateur pour les aides utilisées selon le mécanisme *probabiliste* qui était décrit dans la section « Évaluation des aides » (voir section 5.3.2). En effet, l'utilisateur peut exprimer son opinion concernant chaque aide utilisée en fournissant des votes pour cette aide. Chaque vote se fait selon une échelle d'entiers variant de 1 à 5 et exprimant si l'aide correspondante a été utile pour l'utilisateur, c'est-à-dire si elle l'a aidé à comprendre la question ou non : 1 pour indiquer que cette aide ne l'a pas aidé du tout et 5 pour dire que



cette aide lui a été très bénéfique. UMAKE tiendra compte *immédiatement* de ces votes afin d'améliorer et de raffiner les recommandations futures.

Nous appliquons la méthode du cosinus qui consiste d'abord à définir le voisinage de l'utilisateur cible en recherchant des corrélations et des similarités entre les différents utilisateurs par rapport à leurs votes pour les aides des différentes questions dans leurs profils, et à générer les recommandations pour cet utilisateur en se basant sur ce voisinage.

Avant de présenter un exemple détaillé de la façon avec laquelle nous appliquons la technique du filtrage collaboratif, nous voulons souligner le fait que nous avons modifié un peu l'algorithme original pour l'adapter à notre système et pour qu'il prenne en considération tous les cas d'utilisation de UMAKE. Nous savons tous que cet algorithme essaie de prédire les votes de l'utilisateur cible pour les différents items proposés en se basant sur son voisinage et lui recommande l'item qui a eu la plus haute prédiction. Cependant, cet algorithme suppose dès le départ que l'utilisateur cible ne connaît pas ces items et donc, ne les a pas évalué avant. Or, dans UMAKE, l'utilisateur peut être confronté à des questions qu'il a déjà faites dans des tests antérieurs, mais auxquelles il a répondu incorrectement (voir section 5.3). Pour ces questions, il est probable qu'il ait utilisé certaines aides et qu'il les ait évalué par la suite.

Les évaluations fournies au cours de ses réponses antérieures à ces questions constituent une source de connaissances qui nous permettent de connaître ses préférences pour les différentes aides associées et définir celles qu'il trouve plus au moins utiles. Afin de ne pas les omettre, nous avons modifié cet algorithme de la façon suivante : au début, nous l'appliquons normalement et nous effectuons nos prédictions pour les trois types d'aide proposés. Ensuite, nous vérifions si notre utilisateur a déjà fait cette question auparavant ou non.

Dans le cas où il a déjà fait cette question dans le passé, nous ajoutons une étape dans laquelle nous comparons chaque prédiction avec la moyenne des évaluations antérieures, données pour le type d'aide correspondant en se basant sur son profil (voir

section 5.4.4). Pour chaque type d'aide, nous déterminons le maximum de ces deux valeurs. Et enfin, nous recommandons l'aide qui a la plus haute valeur de ces nouveaux maxima (voir Exemple 2).

Dans le cas contraire, nous continuons d'appliquer l'algorithme normalement en recommandant l'aide qui a eu la plus haute prédiction (voir Exemple 1).

De cette façon, nous concevons un algorithme *général* qui englobe l'algorithme du filtrage collaboratif. Ce nouvel algorithme prend en compte non seulement les goûts des usagers similaires et ceux de l'utilisateur courant pour les différentes aides des questions auxquelles il a déjà répondu auparavant, mais aussi de son *historique* concernant la question courante et notamment ses anciens votes pour les différentes aides associées lors des sessions d'auto-évaluation antérieures.

❖ Pseudo-code de l'algorithme général du filtrage collaboratif CF :

Dans cette partie, nous présentons un exemple de pseudo-code pour notre algorithme du filtrage collaboratif CF. Notons que cet algorithme est exécuté avant l'affichage de chaque question  $q$  du quiz à l'utilisateur  $u$  afin déterminer l'aide à recommander pour  $q$ .

Si utilisateur  $u$  n'a fait aucune question jusqu'à maintenant,  
 | Recommander l'aide qui correspond à son style d'apprentissage  
 | initial dominant selon le filtrage à base de connaissances KB.

Sinon

- a) Calculer la similarité de  $u$  avec les autres usagers (voir Formule 5.4).
- b) Déterminer le voisinage de  $u$ .
- c) Calculer les prédictions correspondantes à chaque type d'aide de la question  $q$  (voir Formule 5.5).

Si  $q$  est nouvelle,

| Recommander l'aide qui a la plus haute prédiction selon le  
 | filtrage collaboratif CF.

Sinon

- a) Comparer les prédictions calculées avec les anciens votes que  $u$  a donné aux différentes aides lors des sessions d'auto-évaluation précédentes et déterminer les trois maxima.
- b) Recommander l'aide qui a le maximum de ces nouvelles valeurs.

Fin si

Fin si

#### ❖ Exemple 1 :

Imaginons que nous avons un ensemble d'utilisateurs de UMAKE. Durant leur auto-évaluation, chacun a répondu à quatre questions au maximum. Nous avons aussi les votes de chacun pour les aides utilisées. Nous regroupons ces votes dans une matrice « utilisateurs  $\times$  objets » (dans notre cas, les objets sont les différentes aides associées aux questions). Ensuite, nous appliquons la méthode du cosinus qui consiste à définir le voisinage d'un utilisateur donné, et à lui générer les recommandations en se basant sur son voisinage.

**Tableau 5.1** : Matrice des votes des apprenants pour les différentes aides des quatre questions (premier exemple)

		Question 1			Question 2			Question 3			Question 4		
		Aide 1	Aide 2	Aide 3	Aide 1	Aide 2	Aide 3	Aide 1	Aide 2	Aide 3	Aide 1	Aide 2	Aide 3
Utilisateurs	Mike	3	1			4	3				2		5
	Dave									5		3	
	Alice	4	2			3					?	?	?
	Bob		4		5	1		1	3			4	
	Marc	5	2	1	2	5		4		3	4	3	2
	Julie	2	3			2						5	

Le Tableau 5.1 est un exemple qui représente une matrice de votes de six apprenants pour les différentes aides des quatre questions. Chaque question a trois types d'aide différents (Aide 1, Aide 2 et Aide 3) qui correspondent respectivement aux aides auditive, visuelle et kinesthésique.

Nous remarquons que les votes dans cette matrice varient bien de 1 à 5. Chaque vote correspond à une note donnée par l'utilisateur à l'aide qu'il a utilisée avant de répondre à la question correspondante.

Nous remarquons aussi qu'il y a des cases vides dans la matrice. Ces cases s'expliquent par trois cas :

- Soit l'utilisateur en question a répondu à la question courante sans utiliser l'aide correspondante et donc, il n'avait pas à l'évaluer.
- Soit il a répondu à la question courante en utilisant l'aide correspondante à la case vide, mais il n'avait pas à l'évaluer (voir section 5.3.2).
- Soit il n'a pas répondu à cette question, c'est-à-dire cette question n'a été incluse dans aucun de ses tests d'auto-évaluation.

Les utilisateurs, Mike pour la question 3 et Dave pour les questions 1 et 2, font partie de ces deux derniers cas.

Dans cet exemple, notre objectif est de déterminer quel type d'aide il faudrait recommander à Alice pour la question 4. Pour cela, nous nous basons d'abord sur l'hypothèse que les trois types d'aide de cette question sont recommandables (ce qui est toujours vrai) et nous chercherons à prévoir les votes qu'Alice pourrait donner respectivement à chacune de ces aides en se basant sur son voisinage qui sera déterminé à l'aide de la Formule 5.4. Dans le Tableau 5.1, les « ? » indiquent donc les prédictions que nous cherchons à calculer.

➤ *Déterminer le voisinage d'Alice :*

Pour déterminer le voisinage d'Alice, nous utilisons la formule du cosinus suivante (voir Formule 5.4) :

$$\cosinus(A, B) = \sum_{j=1}^{ll} \frac{v_{A,j}}{\sqrt{\sum_{j=1}^{ll} v_{A,j}^2}} \times \frac{v_{B,j}}{\sqrt{\sum_{j=1}^{ll} v_{B,j}^2}}$$

$ll$  : nombre d'items communs entre A et B votées par  $v$ .

$v_{A,j}$  : vote de A pour l'item  $j$ .

$v_{B,j}$  : vote de B pour l'item  $j$ .

#### Formule 5.4 : Le calcul du cosinus

Dans notre cas, les items communs sont les aides utilisées en commun pour les questions évaluées en commun. La Formule 5.4 nous sert à calculer la similarité qui existe entre Alice et les différents autres utilisateurs. Nous obtiendrons donc les résultats suivants :

$$\cosinus(Alice, Mike) = \sum_{j=1}^3 \frac{v_{Alice,j}}{\sqrt{\sum_{j=1}^2 v_{Alice,j}^2}} \times \frac{v_{Mike,j}}{\sqrt{\sum_{j=1}^2 v_{Mike,j}^2}} = \frac{4}{\sqrt{29}} \times \frac{3}{\sqrt{26}} + \frac{2}{\sqrt{29}} \times \frac{1}{\sqrt{26}} + \frac{3}{\sqrt{29}} \times \frac{4}{\sqrt{26}} = 0.91$$

$$\cosinus(Alice, Dave) = 0$$

$$\cosinus(Alice, Bob) = \sum_{j=1}^2 \frac{v_{Alice,j}}{\sqrt{\sum_{j=1}^2 v_{Alice,j}^2}} \times \frac{v_{Bob,j}}{\sqrt{\sum_{j=1}^2 v_{Bob,j}^2}} = \frac{2}{\sqrt{13}} \times \frac{4}{\sqrt{17}} + \frac{3}{\sqrt{13}} \times \frac{1}{\sqrt{17}} = 0.73$$

$$\cosinus(Alice, Marc) = \sum_{j=1}^3 \frac{v_{Alice,j}}{\sqrt{\sum_{j=1}^2 v_{Alice,j}^2}} \times \frac{v_{Marc,j}}{\sqrt{\sum_{j=1}^2 v_{Marc,j}^2}} = \frac{4}{\sqrt{29}} \times \frac{5}{\sqrt{54}} + \frac{2}{\sqrt{29}} \times \frac{2}{\sqrt{54}} + \frac{3}{\sqrt{29}} \times \frac{5}{\sqrt{54}} = 0.98$$

$$\cosinus(Alice, Julie) = \sum_{j=1}^3 \frac{v_{Alice,j}}{\sqrt{\sum_{j=1}^2 v_{Alice,j}^2}} \times \frac{v_{Julie,j}}{\sqrt{\sum_{j=1}^2 v_{Julie,j}^2}} = \frac{4}{\sqrt{29}} \times \frac{2}{\sqrt{17}} + \frac{2}{\sqrt{29}} \times \frac{3}{\sqrt{17}} + \frac{3}{\sqrt{29}} \times \frac{2}{\sqrt{17}} = 0.90$$

La similarité entre Alice et Dave est nulle parce qu'ils n'ont pas eu des votes en commun. Nous pouvons maintenant sélectionner les utilisateurs qui possèdent un coefficient de similarité avec Alice supérieur à un certain seuil  $k$ . Si nous prenons  $k=0.9$ , par exemple, nous ne retiendrons que Mike, Marc et Julie. Ces trois utilisateurs constituent le voisinage d'Alice. Notons, toutefois, que le choix de  $k$  est vraiment arbitraire.

➤ *Générer les recommandations :*

Afin de calculer les différentes prédictions d'Alice pour chaque type d'aide de la « Question 4 », nous utilisons la Formule 5.5 suivante :

$$\text{Pr}_{A,j} = \bar{v}_A + \frac{\sum_{i=1}^l (v_{i,j} - \bar{v}_i) \times \cosinus(A,i)}{\sum_{i=1}^l |\cosinus(A,i)|}$$

$\text{Pr}_{A,j}$  : Prédiction d'un utilisateur  $A$  pour un item  $j$ .

$v_i$  : Évaluations effectuées par les  $l$  utilisateurs ayant évalué l'item  $j$  et qui sont présents dans le voisinage de  $A$ .

$\bar{v}_i$  : Moyenne des évaluations pour l'utilisateur  $i$

**Formule 5.5 :** Le calcul de la prédiction d'un utilisateur  $A$  pour un item  $j$

Nous commençons par calculer la moyenne des évaluations pour chaque utilisateur faisant partie du voisinage d'Alice, y compris Alice :

$$\bar{v}_{Alice} = \frac{4+2+3}{3} = \frac{9}{3} = 3$$

$$\bar{v}_{Mike} = \frac{3+1+4+3+2+5}{6} = \frac{18}{6} = 3$$

$$\overline{v}_{Marc} = \frac{5+2+1+2+5+4+3+4+3+2}{10} = \frac{31}{10} = 3.1$$

$$\overline{v}_{Julie} = \frac{2+3+2+5}{4} = \frac{12}{4} = 3$$

Nous pouvons maintenant calculer la prédiction d'Alice pour chacun des types d'aide :

$$\Pr_{Alice,Help1} = 3 + \frac{(2-3) \times 0.91 + (4-3.1) \times 0.98}{|0.91 + 0.98|} = 2.98 \quad (\text{Les votes de Mike et de Marc}$$

ont participé à prédire le vote d'Alice pour le type d'aide « Aide 1 »).

$$\Pr_{Alice,Help2} = 3 + \frac{(3-3.1) \times 0.98 + (5-3) \times 0.9}{|0.98 + 0.9|} = 3.9 \quad (\text{Les votes de Marc et de Julie ont}$$

participé à prédire le vote d'Alice pour le type d'aide « Aide 2 »).

$$\Pr_{Alice,Help3} = 3 + \frac{(5-3) \times 0.91 + (2-3.1) \times 0.98}{|0.91 + 0.98|} = 3.39 \quad (\text{Les votes de Mike et de Marc}$$

ont participé à prédire le vote d'Alice pour le type d'aide « Aide 3 »).

Nous en déduisons que le deuxième type d'aide « Aide 2 » lui sera plus utile que les deux autres types d'aide puisqu'il a eu la plus haute prédiction. Dans ce cas, notre recommandation pour Alice sera donc la deuxième aide qui est *l'aide visuelle*.

#### ❖ Exemple 2 :

Dans cet exemple, nous avons les mêmes données que l'exemple précédent, sauf qu'ici Alice a répondu *incorrectement* à la « Question 4 » au cours d'une session d'auto-évaluation antérieure. Elle a utilisé et évalué quelques aides correspondantes à cette question (voir les notes à côté des « ? » dans le Tableau 5.2). Nous voulons lui recommander une aide appropriée pour cette question :



**Tableau 5.2 :** Matrice des votes des apprenants pour les différentes aides des quatre questions (deuxième exemple)

		Question 1			Question 2			Question 3			Question 4		
		Aide 1	Aide 2	Aide 3	Aide 1	Aide 2	Aide 3	Aide 1	Aide 2	Aide 3	Aide 1	Aide 2	Aide 3
Utilisateurs	Mike	3	1			4	3				2		5
	Dave									5		3	
	Alice	4	2			3					?	2?	5?
	Bob		4		5	1		1	3			4	
	Marc	5	2	1	2	5		4		3	4	3	2
	Julie	2	3			2						5	

Nous appliquons d'abord le même algorithme pour prédire les votes d'Alice pour les trois types d'aide, nous aurons les mêmes prédictions que l'exemple précédent :

$$\Pr_{\text{Alice, Help1}} = 3 + \frac{(2-3) \times 0.91 + (4-3.1) \times 0.98}{|0.91 + 0.98|} = 2.98$$

$$\Pr_{Alice,Help2} = 3 + \frac{(3 - 3.1) \times 0.98 + (5 - 3) \times 0.9}{|0.98 + 0.90|} = 3.9$$

$$\Pr_{Alice,Help3} = 3 + \frac{(5 - 3) \times 0.91 + (2 - 3.1) \times 0.98}{|0.91 + 0.98|} = 3.39$$

Ensuite, pour chaque type d'aide, nous déterminons les prédictions finales de notre algorithme général qui sont les valeurs maximales résultantes de la comparaison de la prédiction calculée, avec le vote correspondant donné par Alice antérieurement :

$$\Pr_{Finale Alice,Help1} = 2.98$$

$$\Pr_{Finale Alice,Help2} = \max(\Pr_{Alice,Help2}, 2) = \Pr_{Alice,Help2} = 3.9$$

$$\Pr_{Finale Alice,Help3} = \max(\Pr_{Alice,Help3}, 5) = 5$$

Nous recommandons par la suite l'aide qui a la plus haute valeur parmi les prédictions finales. En effet, l'*aide kinesthésique* était non seulement appréciée et jugée utile par l'utilisateur lors d'une évaluation antérieure ( $\overline{v_{Alice,Help3}} = 5$ ), mais elle avait aussi une bonne prédiction dans l'algorithme initial ( $\Pr_{Alice,Help3} = 3.39$ ), proche de la valeur de la prédiction maximale ( $\Pr_{Alice,Help2} = 3.9$ ). Et de cette façon, nous pouvons constater que notre algorithme général considère à la fois, les goûts communs des utilisateurs, leurs préférences, leurs historiques concernant la question courante ainsi que les aides associées au cours des sessions d'auto-évaluations précédentes.

#### 5.4.3.2. Filtrage à base de connaissances

Dans notre méthodologie, nous appliquons le filtrage à base de connaissances dans un des cas suivants :

- Si l'utilisateur courant est devant une nouvelle question à laquelle personne de son voisinage n'a répondu avant.

- Si l'utilisateur courant fait partie des premiers utilisateurs du système, c'est ce qu'on appelle le démarrage à froid du système.
- S'il y a égalité après l'application du filtrage collaboratif, c'est-à-dire quand celui-ci ne permet pas de trancher sur le type d'aide qu'il faut recommander à l'utilisateur courant.

Dans le deuxième cas, afin de remédier au démarrage à froid du système, nous nous basons sur le style d'apprentissage initial de l'utilisateur (voir section 5.2.1) pour lui recommander l'aide appropriée.

Dans le premier et le troisième cas, nous *inférons* ou nous *devinons* les préférences de l'apprenant en utilisant *son style d'apprentissage évolué* ou *son comportement* (les types d'aides choisis, les votes correspondants, les réponses...) tel qu'il est perçu et enregistré par le système. Par exemple, même s'il utilise plus des aides auditives, ceci ne veut pas dire forcément qu'il est auditif, nous lui recommandons l'aide qui a contribué à avoir plus de réponses *correctes*. Nous évaluons les items les plus susceptibles de satisfaire les besoins et les préférences de l'utilisateur.

Pour un apprenant donné, afin de trouver le type d'aide le *plus utile* parmi ceux qu'il a déjà utilisés, nous devons déterminer le type d'aide qui l'a le plus aidé et qui a contribué à avoir le plus grand nombre de réponses correctes.

Pour cette raison, nous trouvons dans chaque profil utilisateur et pour chaque type d'aide utilisé un champ *indC*, indiquant si cette aide a contribué à répondre correctement ou non : *indC* sera évalué à 1 dans le cas favorable et à 0 dans le cas contraire. Le Tableau 5.3 est un exemple des différentes données que nous stockons dans le profil utilisateur, pour chaque question. Parmi ces données, nous trouvons évidemment le champ *indC*.

**Tableau 5.3 :** Exemple de données stockées dans le profil apprenant pour les trois aides d'une question  $i$

	<i>Auditive</i>	<i>Visuelle</i>	<i>Kinesthésique</i>
<i>Nb.Util</i>	NU1	NU2	NU3
<i>Nb.Votes</i>	NV1	NV2	NV3
<i>Moyenne_Votes</i>	3.5	4.5	2
<i>indC</i>	1	0	0

Afin de déterminer le type d'aide le plus utile pour un utilisateur donné, nous calculons les valeurs des trois utilités :  $U_{AideAuditive}$ ,  $U_{AideVisuelle}$  et  $U_{AideKinesthésique}$  selon la Formule 5.6 :

$$U_{AideAuditive} = \frac{\sum_{i=1}^n indC\_Auditive_i \times Moyenne\_Auditive_i}{Compteur\ Re\ p_i}$$

$$U_{AideVisuelle} = \frac{\sum_{j=1}^n indC\_Visuelle_j \times Moyenne\_Visuelle_j}{Compteur\ Re\ p_j}$$

$$U_{AideKinesthésique} = \frac{\sum_{k=1}^n indC\_Kines_k \times Moyenne\_Kines_k}{Compteur\ Re\ p_k}$$

**Formule 5.6 :** Le calcul des utilités  $U_{AideAuditive}$ ,  $U_{AideVisuelle}$  et  $U_{AideKinesthésique}$

$indC\_Auditive_i$  et  $Moyenne\_Auditive_i$  représentent respectivement les variables  $indC$  et  $Moyenne$  pour l'aide auditive de la  $i^{ème}$  question stockée dans le profil apprenant.

$indC\_Visuelle_j$  et  $Moyenne\_Visuelle_j$  représentent respectivement les variables  $indC$  et  $Moyenne$  pour l'aide visuelle de la  $j^{ème}$  question stockée dans le profil apprenant.

$indC\_Kines_k$  et  $Moyenne\_Kines_k$  représentent respectivement les variables  $indC$  et  $Moyenne$  pour l'aide kinesthésique de la  $k^{ème}$  question stockée dans le profil apprenant.

Les variables  $Compteur\ Re\ p_i$ ,  $Compteur\ Re\ p_j$  et  $Compteur\ Re\ p_k$  représentent la même variable  $Compteur\ Re\ p$  et correspondent respectivement au nombre de réponses fournies pour les questions  $i$ ,  $j$  et  $k$ .

Les variables  $i$ ,  $j$  et  $k$  sont des compteurs qui varient de 1 à  $n$ ,  $n$  étant le nombre de questions auxquelles l'apprenant a répondu et qui sont stockées dans son profil.

Nous remarquons que chacune des utilités  $U_{AideAuditive}$ ,  $U_{AideVisuelle}$  et  $U_{AideKinesthésique}$  dépend :

- ✓ Des votes de l'apprenant pour le type d'aide correspondant. Ces votes sont représentés par la variable  $Moyenne$ .
- ✓ Du fait que cette aide a déjà contribué à donner des réponses correctes ou non. Ce fait est représenté par la variable  $indC$ .
- ✓ Du nombre de fois pour lesquelles il a répondu à une question. Ce nombre est représenté par la variable  $CompteurRep$ .

Nous divisons par le nombre de fois pour lesquelles l'apprenant a répondu à chaque question pour normaliser l'utilité. L'utilité d'une aide, ayant permis à l'apprenant de répondre correctement, dépend essentiellement du nombre d'essais effectués au total pour donner la réponse correcte. En effet, une aide, qui a contribué à guider l'apprenant à répondre correctement après six essais, n'est *logiquement* pas plus utile qu'une aide qui a contribué à une réponse correcte après deux essais seulement.

Le maximum des utilités  $U_{AideAuditive}$ ,  $U_{AideVisuelle}$  et  $U_{AideKinesthésique}$  correspond au type d'aide que nous recommandons en premier lieu à l'utilisateur, mais nous tenons compte aussi des valeurs de ces trois utilités et nous les sauvegardons dans le profil selon leur contribution dans le style d'apprentissage. Pour cette raison, nous calculons les trois

pourcentages  $P_{Auditif}$ ,  $P_{Visuel}$  et  $P_{Kinesthésique}$  qui représentent la composition du style selon les trois axes *auditif*, *visuel* et *kinesthésique* dans la Formule 5.7 :

$$P_{Auditif} = \frac{U_{AideAuditive}}{U_{AideAuditive} + U_{AideVisuelle} + U_{AideKinesthésique}}$$

$$P_{visuel} = \frac{U_{AideVisuelle}}{U_{AideAuditive} + U_{AideVisuelle} + U_{AideKinesthésique}}$$

$$P_{Kinesthésique} = \frac{U_{AideKinesthésique}}{U_{AideAuditive} + U_{AideVisuelle} + U_{AideKinesthésique}}$$

**Formule 5.7 :** Le calcul des pourcentages  $P_{Auditif}$ ,  $P_{Visuel}$  et  $P_{Kinesthésique}$

Nous sauvegardons les trois pourcentages  $P_{Auditif}$ ,  $P_{Visuel}$  et  $P_{Kinesthésique}$  dans le profil apprenant comme étant son *style d'apprentissage évolué*. D'une part, ils donnent une idée sur la composition et l'évolution du style d'apprentissage présenté à l'apprenant à la fin de son auto-évaluation. D'autre part, ils permettent de recommander l'aide la plus utile à l'utilisateur dans chacun des cas cités au début de cette section.

❖ Pseudo-code de l'algorithme à base de connaissances KB :

Dans cette partie, nous présentons un exemple de pseudo-code pour notre algorithme à base de connaissances KB. Notons que cet algorithme est exécuté avant l'affichage de chaque question  $q$  du quiz à l'utilisateur  $u$  afin déterminer l'aide à recommander pour  $q$ .

Si utilisateur  $u$  n'a fait aucune question jusqu'à maintenant  
 | Recommander l'aide qui correspond à son style d'apprentissage  
 | initial dominant selon le filtrage à base de connaissances KB.

Sinon  
 | Calculer les utilités correspondantes aux trois types d'aide (voir  
 | Formule 5.6).  
 | Si la somme des trois utilités est différente de 0  
 | a) Calculer les pourcentages correspondants aux trois styles  
 | d'apprentissage (voir Formule 5.7).  
 | b) Mettre à jour le style d'apprentissage évolué de  $u$ .  
 | c) Recommander le type d'aide qui correspond au plus haut  
 | pourcentage.

Sinon  
 | Recommander l'aide qui correspond à son style d'apprentissage  
 | évolué dominant.

Fin si

Fin si

#### 5.4.4. Base des profils

Le module de gestion des profils doit constamment mettre à jour le profil utilisateur pour que le recommandeur puisse améliorer ses recommandations pour tous les utilisateurs du système.

Dans cette partie, nous présentons la composition d'un profil utilisateur et l'utilité de ses différentes données. Un profil contient des données statiques et des données dynamiques.

##### 5.4.4.1. Données statiques

Ces données ne changent pas durant une longue période de temps comme certaines informations générales données par l'utilisateur lors de l'étape 1 de la phase d'enregistrement (voir section 5.2.1). Ce sont des données personnelles comme le nom, le prénom, l'âge, le genre et l'adresse e-mail (voir Figure 5.2). L'adresse e-mail représente son nom utilisateur par lequel il est identifié par le système. Cette adresse e-mail doit être donc unique.

```

<?xml version="1.0" encoding="UTF-8" ?>
- <userprofile>
  <firstname>Jeanne</firstname>
  <lastname>D'arc</lastname>
  <age>25</age>
  <sex>Female</sex>
  [REDACTED]
  <password>jeanne</password>
  <concepts />
- <comportement>
  - <styleAppInit>
    <auditoryInit>33.33%</auditoryInit>
    <visualInit>33.33%</visualInit>
    <kinestheticInit>33.33%</kinestheticInit>
  </styleAppInit>
  - <styleAppEvol>
    <auditoryEvol>33.33%</auditoryEvol>
    <visualEvol>33.33%</visualEvol>
    <kinestheticEvol>33.33%</kinestheticEvol>
  </styleAppEvol>
  <x>0</x>
</comportement>
<recSuivie>0</recSuivie>
<compteurQues>0</compteurQues>
- <feedback>
  <posFeedback>0</posFeedback>
  <negFeedback>0</negFeedback>
</feedback>
</userprofile>

```

**Figure 5.2 :** Exemple d'un profil utilisateur initial

La Figure 5.2 représente la structure XML d'un profil utilisateur initial qui vient de s'enregistrer dans le système. Dans cet exemple, nous pouvons voir les données personnelles décrites un peu plus haut dans cette section (nom, prénom, âge, genre et adresse e-mail). Cet utilisateur n'a encore fait aucune session d'auto-évaluation. C'est pour cette raison que les styles d'apprentissage, initial « styleAppInit » et évolué « styleAppEvol », ont les même valeurs et que les compteurs de recommandation suivie « recSuivie », de questions « compteurQues » et des feedbacks « posFeedback » et « negFeedback » sont initialisés à zéro. Nous allons les décrire de façon détaillée dans la section qui suit.



#### 5.4.4.2. Données dynamiques

Ces données peuvent changer durant la connexion de l'utilisateur au système. En effet, l'utilisateur peut utiliser les « outils membre » qui s'offrent à lui dans la page principale pour changer son mot de passe ou son style d'apprentissage initial (voir section 5.2.1). Et durant une session d'auto-évaluation, d'autres données vont être modifiées (voir Figure 5.3), elles concernent les questions auxquelles une réponse a été apportée, les aides utilisées, les votes correspondants...

La Figure 5.3 ci-dessous représente la structure XML d'un profil utilisateur évolué qui est en cours de session d'auto-évaluation. Cet utilisateur est en train de faire un test, il vient de répondre à la question « q13 ». Il y a plusieurs attributs qui ne figuraient pas ou qui n'avaient pas les mêmes valeurs dans le profil présenté dans la Figure 5.2. Ces attributs représentent les données dynamiques. Elles changent au fur et à mesure que l'utilisateur utilise le système.

```

- <styleAppEvol>
  <auditoryEvol>33.33%</auditoryEvol>
  <visualEvol>33.33%</visualEvol>
  <kinestheticEvol>33.33%</kinestheticEvol>
</styleAppEvol>
<x>1</x>
- <itemEvaluation>
  <idQues>q13</idQues>
  <aideRec>visual</aideRec>
- <aidesUtil>
  - <usedAuditory>
    <nbUsedAuditory>1</nbUsedAuditory>
    <nbVotesAuditory>0</nbVotesAuditory>
    <averageVotesAuditory>0</averageVotesAuditory>
    <indCAuditory>1</indCAuditory>
  </usedAuditory>
  - <usedVisual>
    <nbUsedVisual>0</nbUsedVisual>
    <nbVotesVisual>0</nbVotesVisual>
    <averageVotesVisual>0</averageVotesVisual>
    <indCVisual>0</indCVisual>
  </usedVisual>
  - <usedKinesthetic>
    <nbUsedKinesthetic>0</nbUsedKinesthetic>
    <nbVotesKinesthetic>0</nbVotesKinesthetic>
    <averageVotesKinesthetic>0</averageVotesKinesthetic>
    <indCKinesthetic>0</indCKinesthetic>
  </usedKinesthetic>
  </aidesUtil>
  <nbTrueReponses>1</nbTrueReponses>
  <nbFalseReponses>0</nbFalseReponses>
  <compteurRep>1</compteurRep>
</itemEvaluation>
</comportement>
<recSuivie>0</recSuivie>
<compteurQues>0</compteurQues>
- <feedback>
  <posFeedback>0</posFeedback>
  <negFeedback>0</negFeedback>
</feedback>

```

**Figure 5.3 :** Exemple d'un profil d'utilisateur durant une session d'auto-évaluation

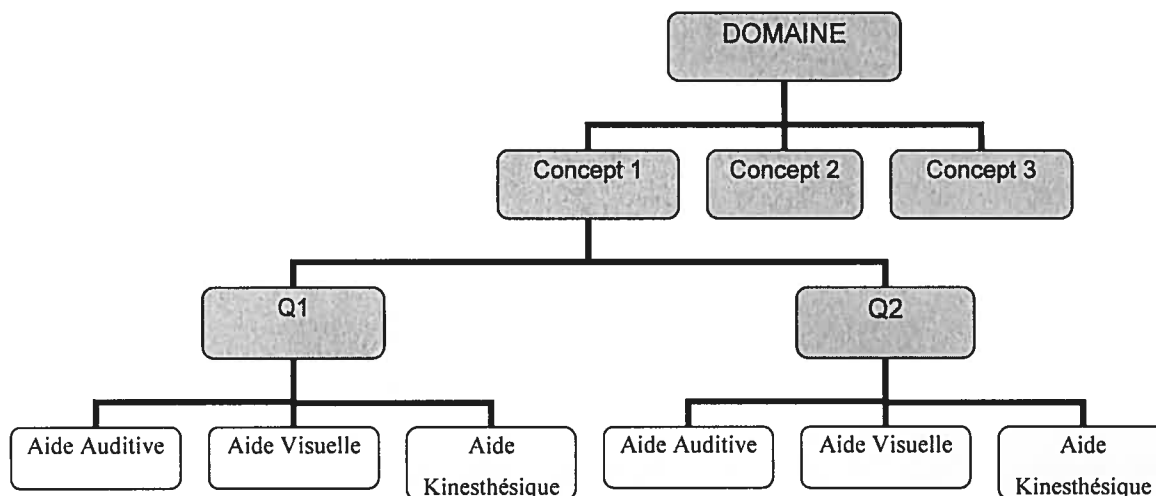
Dans cette partie, nous allons expliquer la fonctionnalité de chacun de ces attributs, ainsi que les données qu'ils représentent dans le profil utilisateur :

- **Concepts** : cet attribut contient la liste des concepts sur lesquels l'apprenant a été évalué. Par exemple, le concept « tri rapide » ou « tri par sélection ».
- **Comportement** : Cet ensemble d'attributs contient des données relatives au comportement de l'utilisateur tel qu'il est analysé par le système. Il se compose de :
  - i. **StyleAppInit** : il représente le style d'apprentissage initial de l'utilisateur déterminé par le système à l'aide du questionnaire de style d'apprentissage. C'est une composition des trois principaux styles d'apprentissage (voir section 5.3.1).
  - ii. **StyleAppEvol** : il représente le style d'apprentissage évolué de l'utilisateur, qui est déterminé par le système au fur et à mesure que l'utilisateur interagit avec lui (voir section 5.4.3).
  - iii. **x** : cet attribut représente la variable  $x$  décrite dans le mécanisme de vote adopté dans UMAKE (voir section 5.3.2).
  - iv. **itemEvaluation** : il représente toutes les données relatives aux différents items (ou questions) sélectionnés pour l'utilisateur durant son auto-évaluation (voir section 5.4.3). Chaque *item/question* contient :
    1. **idQuestion** : c'est l'identifiant de la question à laquelle l'utilisateur vient de répondre au cours de son auto-évaluation. Par exemple, « q13 » désigne la troisième question stockée dans la base de questions du premier concept « tri rapide ».
    2. **AideRec** : cet attribut représente l'aide recommandée à l'utilisateur courant par le système selon la technique de filtrage hybride.
    3. **AidesUtil** : c'est un ensemble d'attributs qui définissent les propriétés des *aides utilisées* par l'utilisateur pour répondre à cette question. Pour chacun des trois types d'aide proposés pour cette question, nous pouvons enregistrer les données suivantes (elles sont représentées dans le même ordre dans la Figure 5.3) :
      - a. le nombre de fois où l'utilisateur a consulté cette aide, représenté par l'attribut qui commence par *nbUsed*.

- b. le nombre de fois où l'utilisateur a fourni une évaluation pour cette aide, représenté par l'attribut qui commence par *nbVotes*.
  - c. La moyenne des votes que l'utilisateur a fournies pour cette aide représentée par l'attribut qui commence par *averageVotes*.
  - d. Un indice qui permet de savoir si l'utilisateur, lorsqu'il a utilisé cette aide, a pu répondre correctement (indice égal à 1) ou non (indice égal à 0). Cet indice est représenté par l'attribut qui commence par *indC*.
4. *nbTrueReponses* : cet attribut représente un compteur pour les bonnes réponses de l'utilisateur à cette question.
  5. *nbFalseReponses* : cet attribut représente un compteur pour les mauvaises réponses de l'utilisateur à cette question.
  6. *compteurRep* : cet attribut représente un compteur pour le nombre de fois pour lesquelles l'utilisateur a répondu à cette question.
- v. ***recSuivie*** : cet attribut représente un compteur du nombre de fois où l'utilisateur a suivi la recommandation du système. Nous considérons qu'un utilisateur a suivi la recommandation proposée par le système en se basant sur plusieurs critères que nous présenterons en détail dans la section 6.4.3 du chapitre 6.
  - vi. ***compteurQues*** : cet attribut représente un compteur du nombre des questions auxquelles l'utilisateur a répondu.
  - vii. ***posFeedback*** : cet attribut représente un compteur pour le nombre de fois où l'utilisateur a été d'accord avec le style d'apprentissage évolué présenté par le système à la fin de chaque test d'auto-évaluation (voir section 5.3.1).
  - viii. ***negFeedback*** : cet attribut représente un compteur pour le nombre de fois où l'utilisateur n'a pas été d'accord avec le style d'apprentissage évolué présenté par le système à la fin de chaque test d'auto-évaluation (voir section 5.3.1).

### 5.4.5. Structuration du domaine d'application

Nous avons choisi de structurer le domaine d'application de manière hiérarchique. La Figure 5.4 présente la hiérarchisation du domaine d'application. Nous voulons souligner aussi que le système peut s'appliquer à plusieurs domaines (voir chapitre 6, section 6.2).



**Figure 5.4 :** Représentation hiérarchique du domaine d'application

A chaque concept du domaine d'application sont associées plusieurs questions et à chaque question sont associées les trois aides auditive, visuelle et kinesthésique parmi lesquelles UMAKE va recommander la plus appropriée à l'utilisateur.

### 5.4.6. Base des concepts et questions associées

Cette base comporte plusieurs concepts disponibles qui peuvent être des sujets de tests d'auto-évaluation pour l'utilisateur. Comme nous l'avons vu dans la section précédente, à chaque concept sont associées plusieurs questions. Dans cette partie, nous présentons la composition de chaque question de la base des questions et l'utilité de ses différentes données (voir Figure 5.5).

```

- <q13>
  <text>When sorting N records, Quicksort has worst-case cost?</text>
  <nb_responses>4</nb_responses>
- <responses>
  <response1>O(log N)</response1>
  <response2>O(N)</response2>
  <response3>O(Nlog N)</response3>
  <response4>O(N^2)</response4>
</responses>
<solution>3</solution>
<AHint>q3-auditory.html</AHint>
<VidHint>q3-visual.html</VidHint>
<KinHint>q3-kinesthetic.html</KinHint>
<yAuditory>20</yAuditory>
<yVisual>17</yVisual>
<yKinesthetic>13</yKinesthetic>
</q13>

```

**Figure 5.5 :** Exemple de la structuration d'une question associée à un concept donné

La Figure 5.5 représente la structure XML d'une question associée au concept « tri rapide ». Toute question dans la base comporte les données suivantes :

- La donnée ou le texte de la question, désigné par l'attribut *text*, qui va être présentée à l'utilisateur dans une page HTML lors de son auto-évaluation.
- Le nombre de choix ou de réponses, désigné par l'attribut *nb\_responses*, que comporte la question.
- Le texte des différentes réponses associées à cette question et qui vont être proposées à l'utilisateur puisque toutes les questions sont à choix multiple. Ces réponses sont désignées par les attributs *response<sub>i</sub>*, *i* variant de 1 à 4 dans la Figure 5.5.
- La position de la bonne réponse ou de la solution désignée par l'attribut *solution*.
- Les chemins d'accès pour les différents types d'aide correspondants à la question courante, désignés respectivement par les attributs *AHint*, *VidHint* et *KinHint*.
- La variable *y* présentée dans la section « Évaluation des aides » (voir section 5.3.2). Chacun des attributs *yAuditory*, *yVisual* et *yKinesthetic* représentent la variable *y* correspondante à chacun des trois types d'aide proposés.

## 5.5. Comparaison

Dans cette section, nous comparons notre système avec d'autres systèmes que nous avons déjà décrits dans notre état de l'art (voir Tableau 5.4). Nous nous basons sur les critères suivants :

- Gestion du démarrage à froid : ce critère permet de savoir si le système arrive à fonctionner correctement lors de sa première utilisation.
- Proposition d'aide : Ce critère permet de savoir si le système assiste les apprenants dans leur évaluation en leur fournissant de l'aide.
- Génération du test/quiz : dynamique, c'est-à-dire le test n'est pas le même pour tous les apprenants, ou statique, c'est-à-dire le test ne change pas dépendamment de l'apprenant.
- Algorithmes utilisés pour la sélection des questions : Ce critère permet de savoir comment le système sélectionne les questions qui constitueront le test pour l'apprenant.
- Algorithmes utilisés pour les aides : Ce critère permet de savoir comment le système sélectionne une aide appropriée pour chaque apprenant.
- Solutions aux questions : Ce critère permet de savoir si le système propose à l'apprenant une solution pour chaque question du test.
- Feedback de l'utilisateur : Nous voulons savoir par ce critère si l'apprenant peut donner son avis sur ce que le système lui présente comme questions, aides ou résultats, et si le système prend en considération cet avis lors des évaluations suivantes.
- Feedback du système : Ce critère permet de savoir si le système fournit un feedback à l'apprenant concernant ses réponses et sa performance.
- Source d'adaptation ou base de la recommandation : Ce critère permet de savoir sur quelles données du profil apprenant, le système se base pour adapter les tests ou pour recommander l'aide appropriée (voir chapitre 2 « hypermédia adaptatif » et chapitre 4 « systèmes de recommandation »).

- Techniques d'adaptation : Ce critère permet de connaître les techniques d'adaptation employées par le système.
- Adaptabilité : Ce critère permet de savoir jusqu'à quel point le système est adaptable à l'apprenant, c'est-à-dire quelles sont les options que l'apprenant peut modifier et que le système prendra en compte. Nous distinguons ici le modèle apprenant du profil apprenant. Le premier nous permet de représenter ce que nous assumons être vrai sur l'apprenant, c'est en quelque sorte le moule de celui-ci. Le second contient les informations construites grâce au modèle, c'est en quelque sorte l'objet moulu.

**Tableau 5.4 : comparaison des systèmes**

	<b>ELM-ART [Weber &amp; Brusilovsky 01].</b>	<b>SIETTE [Guzman <i>et al.</i> 05]</b>	<b>Web-based practice quiz system [Gejji 01]</b>	<b>UMAKE</b>
<b>Gestion du démarrage à froid</b>	Ce n'est pas mentionné.	Ce n'est pas mentionné.	Ce n'est pas mentionné.	Oui
<b>Proposition d'aide</b>	Oui	Non	Non	Oui
<b>Génération du test/quiz</b>	Dynamique	Dynamique	Dynamique	Dynamique
<b>Algorithmes utilisés pour la sélection des questions</b>	La prochaine question proposée dépend de la réponse donnée à la question courante.	La prochaine question proposée dépend de la réponse donnée à la question courante.	La prochaine question proposée dépend de la réponse donnée à la question courante.	Mélange de questions « nouvelles » et « anciennes ».
<b>Algorithmes utilisés pour les aides</b>	Chaque aide relative à une question mène à la page de cours correspondante.	Pas d'algorithme	Pas d'algorithme	Technique de filtrage hybride : filtrage collaboratif et filtrage à base de connaissances.
<b>Solutions aux questions</b>	Oui	Oui	Pas de solution proposée.	Oui



Suite du Tableau 5.4 :

	<b>ELM-ART</b> [Weber & Brusilovsky 01].	<b>SIETTE</b> [Guzman <i>et al.</i> 05]	<b>Web-based practice</b> quiz system [Gejji 01]	<b>UMAKE</b>
<b>Feedback de l'utilisateur</b>	Non	Non	Non	Oui, il donne des évaluations aux aides utilisées et, son accord ou désaccord pour le style d'apprentissage évolué.
<b>Feedback du système</b>	Oui	La solution est affichée après chaque question avec un feedback optionnel.	On informe l'utilisateur de sa réponse après chaque question et on lui propose de faire une question plus difficile s'il a répondu juste, sinon une question plus facile.	Statistiques, solution proposée, style d'apprentissage évolué, méthode utilisée pour la recommandation.
<b>Source d'adaptation ou base de la recommandation</b>	Niveau de connaissances de l'utilisateur et ses préférences.	Pas d'adaptation.	Pas d'adaptation.	Style d'apprentissage de l'utilisateur, ses réponses, ses préférences et sa performance.
<b>Techniques d'adaptation</b>	Adaptation de la présentation, adaptation de la navigation, ordonnancement adaptatif des cours.	Pas d'adaptation.	Pas d'adaptation.	Adaptation de la présentation et adaptation de la navigation.
<b>Adaptabilité</b>	Manipulation directe du modèle apprenant.	Pas d'adaptabilité.	Pas d'adaptabilité.	Manipulation de quelques attributs du profil apprenant.

❖ Interprétation du tableau :

Comme nous l'avons déjà expliqué dans la section 5.4.3, UMAKE arrive à gérer le démarrage à froid grâce à l'initialisation du style d'apprentissage de l'apprenant dès la phase d'enregistrement.

Parmi les systèmes présents dans le tableau, il n'y a que ELM-ART [Weber & Brusilovsky 01] et UMAKE qui proposent de l'aide aux apprenants. A la fin de l'exercice, ELM-ART présente à l'apprenant ses réponses fausses et, à côté de chaque réponse fausse, il y a un bouton d'aide qui mène à la page du cours qui traite le concept visé par la question (sachant que ELM-ART est un système d'apprentissage complet qui comprend aussi des outils d'évaluation pour évaluer le niveau de connaissances des apprenants), tandis que UMAKE assiste les utilisateurs au cours de leur auto-évaluation. Pour chaque question, il leur propose trois types d'aide, et leur recommande un seul parmi les trois en utilisant une technique de filtrage hybride *la commutation* (voir section 5.4.3).

La génération du test/quiz est dynamique pour tous les systèmes cités dans ce tableau : UMAKE suit une stratégie particulière pour sélectionner les questions constituant le test pour l'apprenant en se basant sur le concept de questions « nouvelles » et questions « anciennes » (voir section 5.3), alors que les trois autres systèmes se basent sur le paramètre « difficulté » pour sélectionner les questions constituant le test. En effet, dans ces systèmes, la prochaine question proposée à l'apprenant dépend de la réponse donnée à la question courante.

Parmi les quatre systèmes cités dans ce tableau, seul le système de Gejji [Gejji 01] ne présente pas la solution à l'apprenant. Après chaque question, il affiche juste la nature de la réponse de ce dernier (correcte ou incorrecte), et lui propose de faire une question plus difficile ou plus facile selon cette réponse. Tandis que SIETTE [Guzman *et al.* 05], propose la solution après chaque réponse de l'apprenant avec un feedback optionnel. ELM-ART propose la solution pour toutes les questions à la fin du test avec une note additionnelle dans le cas d'une réponse fausse de l'apprenant, lui expliquant pourquoi sa

réponse était incorrecte. A la fin de chaque test, UMAKE présente une variété d'informations à l'apprenant. Pour chaque question, il lui présente le temps mis pour répondre à cette question, la nature de sa réponse avec la possibilité de consulter la bonne réponse, les aides utilisées, l'aide recommandée, quelle méthode le système a utilisé pour lui recommander cette aide (filtrage collaboratif ou filtrage à base de connaissances) et la raison pour laquelle il a utilisé cette méthode (voir section 5.4.3). UMAKE présente aussi à l'apprenant des statistiques concernant sa performance dans ce test, des statistiques concernant sa performance globale (pourcentage de réussite global, pourcentage d'échec global, combien de fois il a utilisé chaque type d'aide, combien de fois il a suivi la recommandation du système...) et comment son style d'apprentissage a évolué durant son interaction avec le système. UMAKE reste aussi *le seul système* qui permet à l'apprenant de donner son opinion concernant les aides utilisées (les différentes évaluations données à ces aides) et le style d'apprentissage évolué (si l'apprenant en est convaincu ou non). UMAKE tient compte de cette opinion pour mettre à jour le profil apprenant et améliorer ses recommandations futures.

ELM-ART s'appuie sur le niveau de connaissances de l'apprenant et ses préférences comme base d'adaptation pour appliquer l'adaptation de la présentation, celle de la navigation (l'annotation des liens selon la métaphore des feux de circulation décrite dans la section 2.4.3.1-a du chapitre 2) et l'ordonnancement adaptatif des cours (à partir du cours courant, le meilleur prochain cours, requis pour atteindre l'objectif de l'apprenant, est accessible par le lien « suivant »). De plus, l'utilisateur peut accéder directement à son modèle apprenant et le modifier selon son estimation de son niveau de connaissances. Par contre, UMAKE fait de l'adaptation en se basant sur le style d'apprentissage, les réponses, les préférences et la performance de l'apprenant. L'adaptation de la navigation est réalisée par le biais de l'annotation des liens (l'aide recommandée est colorée d'une couleur différente). La présentation de la solution est adaptée à la réponse donnée par l'utilisateur. En effet, pour chaque question, nous présentons les réponses alternatives (n'oublions pas que toutes les questions sont à choix multiple) tout en marquant, parmi ces choix, la bonne réponse et la réponse de l'apprenant d'une façon différente dépendamment de sa nature (correcte ou incorrecte).

De façon personnalisée, nous lui présentons aussi la méthode adoptée pour lui recommander l'aide appropriée, avec la raison pour laquelle nous avons appliqué cette méthode dans son cas à lui. Dans UMAKE, l'utilisateur a la possibilité de modifier quelques aspects de son profil comme le mot de passe ou le style d'apprentissage initial ou évolué (voir section 5.2.1).

## 5.6. Conclusion

Au cours de ce chapitre, nous avons vu l'architecture et expliqué la méthodologie de notre système UMAKE. Nous avons présenté en détails les différents modules qui composent cette architecture et les différentes étapes de notre méthodologie. Le système que nous avons conçu, permet d'aider un apprenant donné à bien comprendre la question qui lui est posée. Au fait, nous lui recommandons une aide bien définie parmi 3 proposées en se basant sur une technique hybride HY qui commute entre deux techniques de filtrage : le filtrage collaboratif CF et le filtrage basé sur la connaissance KB. Nous utilisons aussi certaines techniques d'adaptation dans UMAKE comme l'*adaptation de présentation* pour présenter la solution d'une question donnée, et l'*adaptation de navigation* pour indiquer l'aide recommandée pour chaque question. Nous montrons ces différentes techniques en détails dans le chapitre 6.

Dans le chapitre suivant, nous proposons de voir le détail de l'implémentation et les technologies utilisées dans UMAKE. Ce sera, donc, une présentation plus concrète du système.

## Chapitre 6 : Implémentation et validation

Dans ce chapitre, nous parlerons de l'environnement de développement de notre système, notamment des outils et des langages de programmation utilisés. Nous donnerons par la suite des scénarios d'utilisation de UMAKE en termes de captures d'écran.

### 6.1. Implémentation de UMAKE

UMAKE a été développé sous le système d'exploitation Windows XP et le serveur web Apache Tomcat, version 5.0.30. Nous avons utilisé le langage de programmation JAVA (J2SE version 1.4.2) sous l'environnement de développement NetBeans, version 4.1. Nos données sont stockées sous forme de fichiers XML.

Plus précisément, les technologies utilisées pour le développement de UMAKE sont les suivantes :

- Java Servlet version 2.4 et JSP (Java Server Pages) version 2.0, sous le serveur Tomcat version 5.0.30.
- La technologie DOM pour parser les données des fichiers XML.
- HTML et Javascript pour les interfaces.

Ces technologies sont décrites brièvement dans la section qui suit.

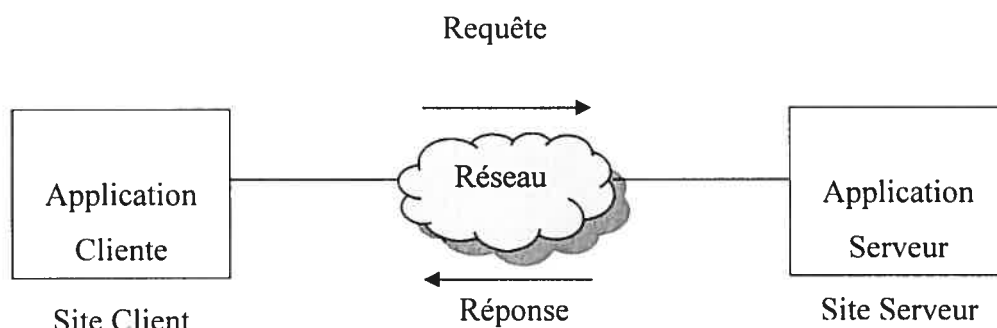
#### 6.1.1. Les Servlets et les pages JSP (Java Server Pages)

Au fil de ces dernières années, beaucoup de sites web se sont développés. Par conséquent, plusieurs technologies et langages de programmation liés au Web sont apparus. Ils permettent d'améliorer la gestion et le design des sites web : HTML, ASP, PHP, ASP, ASP.NET, etc.

Nous allons introduire le fonctionnement des Servlets et des pages JSP, liés au langage de programmation JAVA, et permettant de développer des applications client-serveur dynamiques.

#### 6.1.1.1. Architecture client-serveur

Dans l'architecture client-serveur, la communication est réalisée à travers un dialogue entre un client et un serveur (voir Figure 6.1). En effet, le client émet des requêtes et reçoit les réponses. C'est le serveur qui reçoit ces requêtes, les traite et émet les réponses.



**Figure 6.1** : La communication dans l'architecture client-serveur

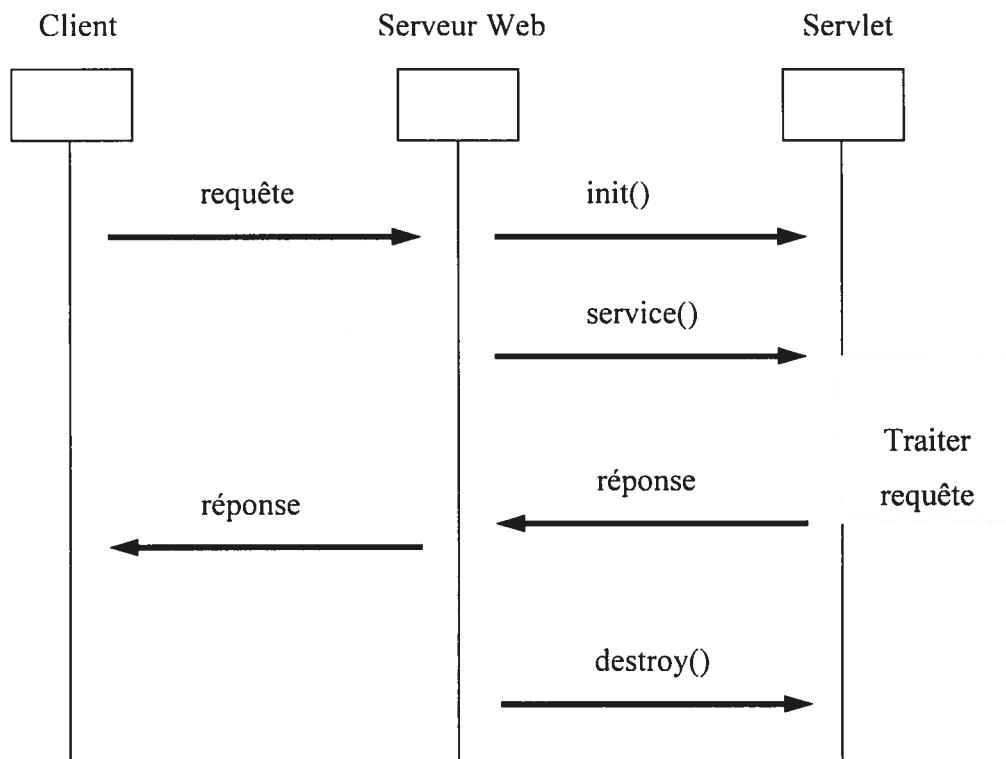
Par conséquent, un site Web doit permettre l'échange d'informations entre le client, situé du côté navigateur, et le serveur Web. La communication peut se faire selon plusieurs protocoles, dont le protocole HTTP (HyperText Transfer Protocol). Lorsque le client saisit une URL (Uniform Resource Locator) dans son navigateur, une requête http est envoyée au serveur Web, c'est-à-dire que le client interroge le serveur Web. Le serveur Web renvoie ensuite une réponse basée également sur le protocole http.

#### 6.1.1.2. Les servlets

Une servlet Java est une classe qui a pour but de générer une réponse à une sollicitation d'un service. Étant donné qu'elle est écrite en Java, elle est *multi-plateforme* : elle est portable entre les serveurs et les différents systèmes d'exploitation. Une servlet peut être chargée automatiquement lors du démarrage du serveur Web ou lors

de la première requête du client. Une fois chargées, les servlets restent actives dans l'attente d'autres requêtes du client. Elles sont aussi performantes parce qu'elles sont exécutées dans des *threads* et ne sont pas limitées au protocole http. En effet, il existe plusieurs types de servlets selon le protocole de communication entre le client et le serveur Web. Les servlets que nous avons utilisées, sont les servlets *HttpServlet*.

La classe de base d'une servlet est *GenericServlet*. Le serveur Web interagit avec la servlet en utilisant une de ces trois méthodes : *init*, *service* et *destroy* (voir Figure 6.2).

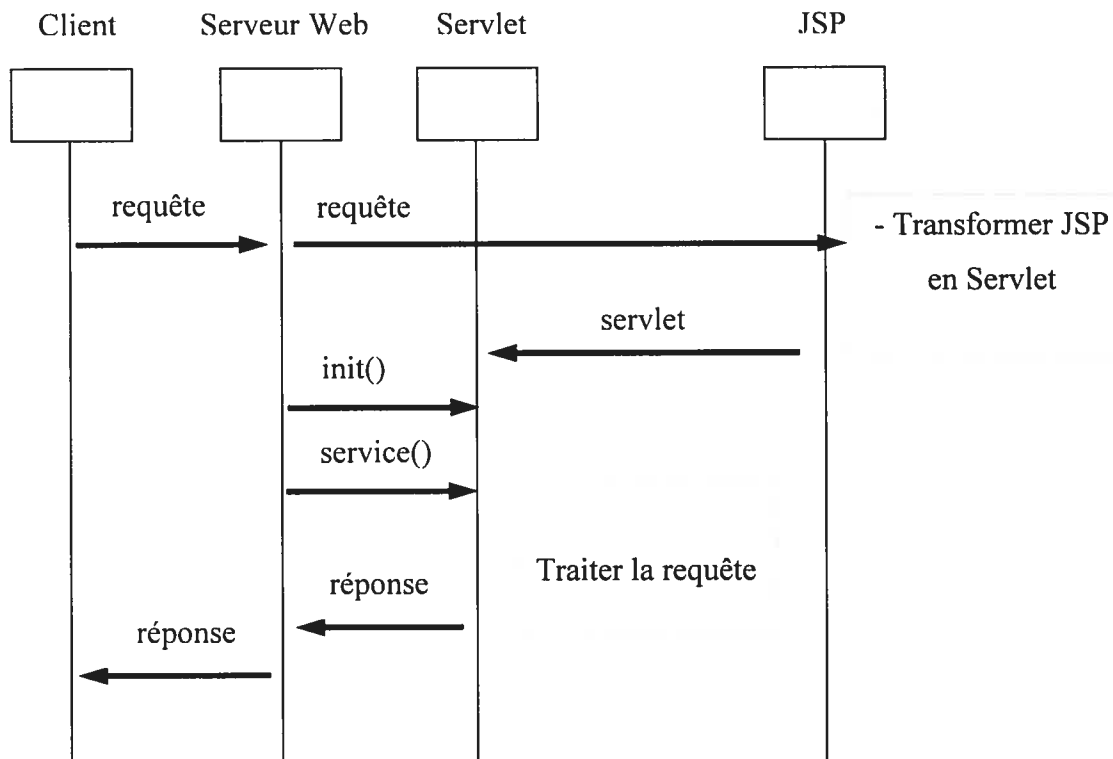


**Figure 6.2** : Le cycle de vie d'une servlet

La méthode *init* est appelée par le serveur s'il n'existe aucune instance de la servlet. Elle s'exécute une seule fois au début pour initialiser l'état de la servlet, configurer les ressources externes, etc. La méthode *service* est la méthode principale, elle est définie dans la classe `javax.servlet.GenericServlet`. Elle peut être appelée zéro ou plusieurs fois durant la vie de la servlet car plusieurs utilisateurs peuvent demander ce service (c'est-à-dire la servlet). Cette méthode prend deux paramètres de types

*ServletRequest* et *ServletResponse*. L'objet *ServletRequest* permet d'accéder aux différentes informations concernant la requête du client, et l'objet *ServletResponse* permet de produire la réponse envoyée par le serveur au client. La méthode *destroy* est utilisée par le serveur juste avant de se fermer pour permettre à la servlet de se défaire de ses liens vers des ressources externes.

La technologie JSP (Java Server Pages) est souvent combinée avec des servlets dans une même application. Elle facilite la création des pages Web dynamiques, en permettant l'intégration de code HTML statique et du code JAVA. Les pages JSP sont automatiquement compilées sous forme de servlets avant d'être exécutées (voir Figure 6.3).



**Figure 6.3 :** Le cycle de vie de JSP



### 6.1.1.3. Technologie XML

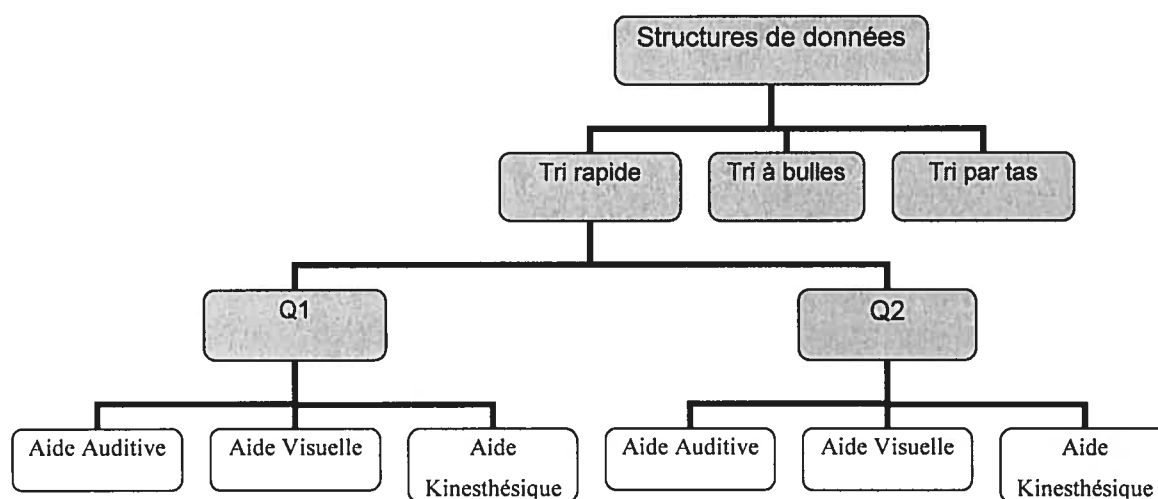
XML (eXtensible Markup Language) est un format de description des données, fondé sur le langage HTML et utilisé pour l'échange structuré de documents sur Internet. En effet, c'est un ensemble de règles de syntaxe qui permet de représenter, grâce à des balises, des données dans des documents de type texte.

Nous utilisons XML dans UMAKE pour stocker les informations de l'application : profils des utilisateurs, les questions et les concepts, etc.

## 6.2. Domaine d'application

Le domaine d'application que nous avons choisi est ``structures de données``. UMAKE permet donc aux utilisateurs de faire des tests d'auto-évaluation, et les aide à mieux comprendre chaque question posée en leur recommandant une aide appropriée. Chaque test peut porter sur un des plusieurs concepts du domaine structures de données, comme le tri rapide « Quick sort », le tri à bulles, etc.

En Figure 6.4, nous pouvons voir une partie de la hiérarchisation du domaine. Un utilisateur peut donc choisir un concept, parmi plusieurs disponibles, pour commencer son auto-évaluation.



**Figure 6.4 :** Illustration d'une partie du domaine d'application

Nous allons maintenant voir les différentes étapes par lesquelles un utilisateur doit passer pour accéder aux différentes fonctionnalités du système, puis nous montrerons des captures d'écran de ces fonctionnalités.

## 6.3. Environnement d'un utilisateur

Dans cette section, nous présentons les composants de l'environnement de l'utilisateur lorsque celui-ci se connecte au système.

### 6.3.1. Page d'accueil de UMAKE

La Figure 6.5 présente la page d'accueil du système. Dans cette page, l'utilisateur déjà enregistré, saisit son nom d'utilisateur, son mot de passe et clique sur le bouton *Sign In* pour se connecter au système et, éventuellement accéder aux différentes fonctionnalités offertes aux membres (voir sections 6.3.3 et 6.3.4). Dans le cas où il n'est pas enregistré, il pourra le faire en cliquant sur le lien *Sign Up* (voir section 6.3.2).

# Welcome to UMAKE

**Signing in to UMAKE**

---

User Name:

Password:

---

**Don't have a user name?**

Signing up is easy.

[Sign Up](#)

**About UMAKE**

UMAKE (User Modeling for Adapted Knowledge Evaluation) is a Web-Based system which can provide the learners with dynamic quizzes. It enables them to understand the questions: UMAKE assists them while they are being evaluated and recommends to them the most suitable "helps" for each question in the quiz. We employed a hybrid technique of recommendation based on the collaborative filtering and the knowledge-based filtering. The different helps tools are adapted to the learning style of the learner. [Click here for more details](#)

Figure 6.5 : La page d'accueil de UMAKE (sans détails)

Dans la partie droite de la page d'accueil, nous donnons une brève définition de UMAKE (voir Figure 6.5). Les utilisateurs peuvent en connaître davantage sur UMAKE en cliquant sur le lien *Click here for more details*. Une définition plus détaillée s'affichera dans la même page, en dessous de la définition générale (voir Figure 6.6). Cette option est utile aussi pour les utilisateurs qui ne savent pas par où commencer. Elle leur explique les différentes étapes nécessaires pour faire une auto-évaluation.

## Welcome to UMAKE

**Signing in to UMAKE**

---

User Name:

Password:

---

**Don't have a user name?**

Signing up is easy.

[Sign Up](#)

**About UMAKE**

UMAKE (User Modeling for Adapted Knowledge Evaluation) is a Web-Based system which can provide the learners with dynamic quizzes. It enables them to understand the questions. UMAKE assists them while they are being evaluated and recommends to them the most suitable "helps" for each question in the quiz. We employed a hybrid technique of recommendation based on the collaborative filtering and the knowledge-based filtering. The different helps tools are adapted to the learning style of the learner.

UMAKE operates in two essential phases: registration phase and auto-evaluation phase.

In order to use UMAKE, all the users must register and create an account. The registration phase is composed of two parts. In the first part, the user must provide general information such as his name, email address and password. The second part of the registration is the psychotechnical questionnaire. This questionnaire is composed of 14 questions that new user must answer in order for UMAKE to determine his learning style.

When an existing member logs into UMAKE, he will be presented with the main page which is the center point of the UMAKE system. From the main page, the user has access to the available topics for the auto-evaluation. Moreover, from the main page, the user has access to several "member tools" such as: change his password, redo the learning style questionnaire or consult his overall performance summary.

The auto-evaluation phase begins when the member chooses a topic. Before starting the auto-evaluation test, the user must select the number of questions in the test. The test will be presented to him one question at a time. For each question, he is provided with three hints with respect to the three main learning styles. UMAKE will recommend one of the hints to get a personalized help. If the user uses one or more helps, he might be required to evaluate the used help(s) before passing to the next question. UMAKE records these evaluations and uses them in order to improve future recommendations. At the end of the test, UMAKE presents a summary of the user's performance during this auto-evaluation session, where the user can view the solution to the answers, and an overall summary of his performance. [Click here to hide the details](#).

**Figure 6.6 :** La page d'accueil de UMAKE (avec détails)

### 6.3.2. Enregistrement de l'utilisateur

Dans UMAKE, un nouvel utilisateur peut s'enregistrer en deux étapes faciles. Lors de la première étape, il saisit dans un simple formulaire des informations générales comme son nom, son sexe, son adresse email et son mot de passe (voir Figure 6.7).

## Registration

### Step 1: General Information

Fields marked with an asterisk \* are required.

* First Name:	Sara
* Last Name:	Doe
Age:	22
* Gender:	Female ▼
* E-mail (Username):	sara.d@hotmail.com
* Password:	●●●●●●
* Re-type your password:	●●●●●●



[Go to Step 2](#)

**Figure 6.7 :** La première étape de l'enregistrement

Lors de la deuxième étape, il se verra proposer un questionnaire psychotechnique [URL 1] qui comporte 14 questions (voir Figure 6.8).

# Registration

## Step 2: Learning Style

Please answer these 14 questions. Select the choice that best describes you.

### Question 1

**If I have to learn how to do something, I learn best when I:**

- Watch someone show me how
- Hear someone tell me how
- Try to do it myself

### Question 2

**When I read, I often find that I:**

- Visualize what I am reading in my minds' eye
- Read out loud or hear the words inside my head.
- Mess and try to feel Fidget and try to feel the content.

...

### Question 13

**If someone were verbally describing something to me, I would:**

- Try to visualize what she was saying.
- Enjoy listening but want to interrupt and talk myself
- Become bored if her description got too long and detailed.

### Question 14

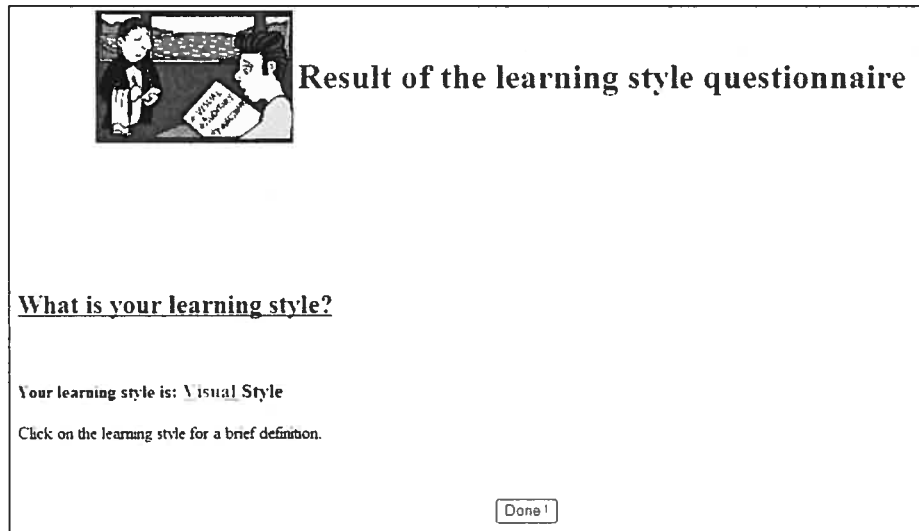
**When trying to recall names, I remember:**

- Faces but forget names.
- Names, but forget faces.
- events, but not names or faces.

Finish

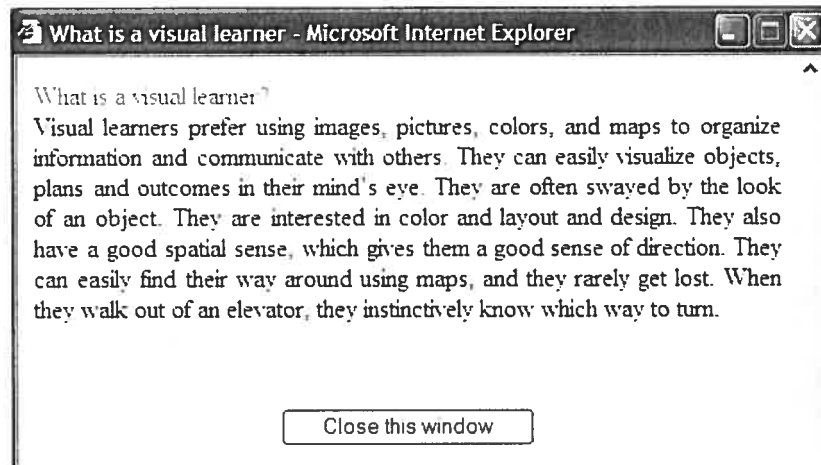
**Figure 6.8 :** La deuxième étape de l'enregistrement

Ce questionnaire permet à UMAKE de déterminer le style d'apprentissage de l'utilisateur (voir section 5.3.1). Quand celui-ci soumet ses réponses, le système lui présente le style d'apprentissage déterminé (voir Figure 6.9).



**Figure 6.9** : Le style d'apprentissage déterminé par UMAKE



De plus, il lui procure la possibilité de consulter les informations qui caractérisent son style. Quand il clique sur le lien approprié, une fenêtre pop up s'ouvre et affiche une brève définition (voir Figure 6.10).



**Figure 6.10** : Les apprenants visuels

### 6.3.3. Outils membres

Une fois enregistré, l'utilisateur peut accéder à la page principale du système (voir Figure 6.11).

<p>Welcome Sara Doe</p>	 <p data-bbox="654 653 1230 716">"The next big killer application on the internet is going to be education. Education over the internet is going to be so big it is going to make e-mail usage look like a rounding error."</p> <p data-bbox="873 741 1008 804">John Chambers CEO Cisco Systems</p>
<p>Available topics</p> <ul style="list-style-type: none"> <li><a href="#">Overview</a></li> <li><a href="#">Basic Set</a></li> <li><a href="#">Concepts</a></li> <li><a href="#">Help Set</a></li> </ul>	
<p>Member Tools</p>	
<ul style="list-style-type: none"> <li><a href="#">Overall Summary</a></li> <li><a href="#">Learning Style Questionnaire</a></li> <li><a href="#">Change Password</a></li> <li><a href="#">My Profile</a></li> <li><a href="#">Help</a></li> </ul>	
	

**Figure 6.11** : La page principale de UMAKE

Cette page est divisée en deux parties. La partie gauche est le menu de navigation de l'utilisateur. Elle présente les différents sujets (ou concepts) disponibles pour son auto-évaluation et plusieurs outils membre. La manipulation de la plupart de ces outils se fait dans la partie droite de la page. Au chargement de la page principale, la partie droite présentera une photo et une citation (choisie arbitrairement parmi plusieurs enregistrées dans le système). Nous détaillons chacun des outils offerts aux membres dans la partie qui suit :

### 6.3.3.1. Statistiques globales

L'outil *Overall Summary* permet à l'apprenant de consulter des statistiques globales concernant sa performance dans les tests, les aides utilisées et le nombre de fois

pour lesquelles il a suivi notre recommandation durant son interaction avec le système (voir Figure 6.12).

The screenshot shows a web interface for a user named Sara Doe. The main content area is titled "Statistics" and contains the text "An overall summary of your performance". Below this is a table with the following data:

Your overall success percentage	75.2%
Your overall failure percentage	24.75%
Number of times you used an auditory help	8
Number of times you used a visual help	6
Number of times you used an kinesthetic help	6
Number of times you follow our recommendation	2

At the bottom of the main content area is a button labeled "Go to the Main Page". The left sidebar contains a "Welcome Sara Doe" message, "Available topics" (with links for Home, Home 2, Home 3, Home 4), "Member Tools" (with links for Home, Home 2, Home 3, Home 4), and a cartoon character of a boy carrying books.

**Figure 6.12 :** L'outil *Overall Summary*


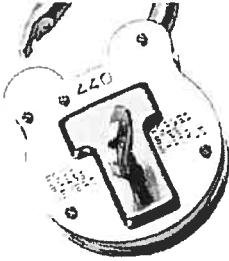
### 6.3.3.2. Questionnaire du style d'apprentissage

L'outil *Learning Style Questionnaire* permet à l'apprenant de réinitialiser son style d'apprentissage initial détecté lors de la deuxième étape de son enregistrement. A l'aide de cet outil, il refait le questionnaire du style d'apprentissage présenté dans la Figure 6.8 [URL 1].

### 6.3.3.3. Mot de passe

L'outil *Change Password* permet à l'apprenant de changer son mot de passe (voir Figure 6.13). Quand il se connectera au système la prochaine fois, il va devoir utiliser son nouveau mot de passe.



<p>Welcome Sara Doe</p> <hr/> <p>Available topics</p> <hr/> <p><a href="#">Quickstart</a></p> <p><a href="#">Building Site</a></p> <p><a href="#">Merge Post</a></p> <p><a href="#">Hello Site</a></p> <hr/> <p>Member Tools</p> <hr/> <p><a href="#">Create a Summary</a></p> <p><a href="#">Learning Some Quickstart</a></p> <p><a href="#">Change Password</a></p> <p><a href="#">Neighbourhood</a></p> <p><a href="#">Help</a></p> 	<h2>Password Modification</h2> <p>Fields marked with an asterisk* are required.</p> <p>* New Password: <input type="password"/></p> <p>* Re-type your password: <input type="password"/></p> <p><input type="button" value="Submit"/></p> 
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Figure 6.13** : L'outil *Change Password*

#### 6.3.3.4. Voisinage

Comme nous l'avons déjà détaillé dans le chapitre précédent, la recommandation dans UMAKE est basée sur une technique de filtrage hybride qui emploie le filtrage collaboratif et le filtrage à base de connaissances. Le premier filtrage est réalisé en se basant sur le voisinage de l'utilisateur (voir section 5.4.3). Pour cette raison, nous lui avons procuré la possibilité de le consulter à partir de la page principale du système, à l'aide de l'outil *Neighbourhood*. Dans la partie droite de cette page, il verra des informations lui expliquant la notion de « voisinage ». En bas de la page, il trouvera un tableau affichant les membres constituant son voisinage avec la mesure de similarité qu'il a avec chacun d'eux (voir Figure 6.14).

Welcome Sara Doe

---

Available topics

[Question](#)

[Feedback](#)

[Help](#)

---


Member Tools

[Profile](#)

[Settings](#)

[Logout](#)

---



## Neighbourhood

Hi Sara Doe,


Each member in UMAKE has his own neighbourhood, which means a set of other members who have almost the same tastes (what they like and what they dislike)

UMAKE computes your similarity with each member registered based on the set of questions he has done and the evaluations given to the related different helps

The set of members with whom you have a similarity superior to a value threshold (initialized by the system) constitute your neighbourhood.

UMAKE recommends to you the appropriate help based on this neighbourhood.

Members in your neighbourhood	Similarity
Andy Jason	0.62
Roy Keane	0.75
Moez Mabrouk	0.76



Return to the Main Page

**Figure 6.14 :** L'outil *Neighbourhood*

### 6.3.3.5. Aide

L'apprenant, qui se trouve dans la page principale, peut consulter une aide lui expliquant la signification de chacun des outils offerts et lui indiquant comment procéder pour faire un test d'auto-évaluation (voir Figure 6.15).



<p>Welcome Sara Doe</p>	<h2 style="text-align: center;">Help</h2> <p>Hi <i>Sara Doe</i>,</p> <p>You are now in the main page of UMAKE. On the left hand side of this page, you have the navigation menu which offers you several options.</p> <p>Under "Available Topics", you have the list of various topics in which you can do an auto-evaluation test.</p> <p>The "Member Tools" offer you a set of options to view and maintain your account:</p> <p>you can change your password, redo the learning style questionnaire, consult the overall summary of your performance, to view which members are in your neighbourhood or display this help</p> <p>When you choose topic for the auto-evaluation test, you must first select the number of questions in the test. The test will presented to you one question at a time. For each question, if you need some help, you can click on one of the three available hints. UMAKE will recommend you one of the hints, displayed in a different color. On some occasions, you will be required to evaluate the used help(s) before passing to the next question. UMAKE records these evaluations and uses them in order to <u>improve future recommendations</u>. At the end of the test, UMAKE will present you a summary of your performance during this auto-evaluation session, where you can view the solution to the answers, and an overall summary of your performance</p> <div style="text-align: center;">  </div> <p style="text-align: center;"><input type="button" value="Hide this help"/></p>
<p>Available topics</p> <ul style="list-style-type: none"> <li><a href="#">Quick Sort</a></li> <li><a href="#">Bubble Sort</a></li> <li><a href="#">Merge Sort</a></li> <li><a href="#">Heap Sort</a></li> </ul>	
<p>Member Tools</p>	
<ul style="list-style-type: none"> <li><a href="#">My Account</a></li> <li><a href="#">Learning Style Questionnaire</a></li> <li><a href="#">Change Password</a></li> <li><a href="#">Neighbourhood</a></li> <li><a href="#">Help</a></li> </ul> <div style="text-align: center;">  </div>	

Figure 6.15 : L'outil *Help*

### 6.3.4. Auto-évaluation de l'utilisateur

La phase d'auto-évaluation commence quand l'utilisateur choisit un des concepts disponibles (voir section 5.2.1). Il commence par sélectionner le nombre des questions qui composeront son test (voir Figure 6.16).

Welcome Sara Doe ([sara.d@hotmai.com](mailto:sara.d@hotmai.com))

## Auto-Evaluation test on: Quicksort

Please select the size of the auto-evaluation test:  questions


For each question, you'll be provided with three hints  related to the three main learning styles.


If you need some help, click on one of the hints.


UMAKE recommend you to click on the coloured one to get a personalized help.

Disable the PopUp blocker to see the help.

**Tutorial (How to use the hints?):**


Auditory


Visual


Kinesthetic

**Figure 6.16 :** Le choix du nombre des questions dans UMAKE

Dans la même page, il a accès à un petit tutorial qui lui explique comment manipuler chaque type d'aide dans le test. La Figure 6.17 montre un exemple de tutorial pour l'aide visuelle.

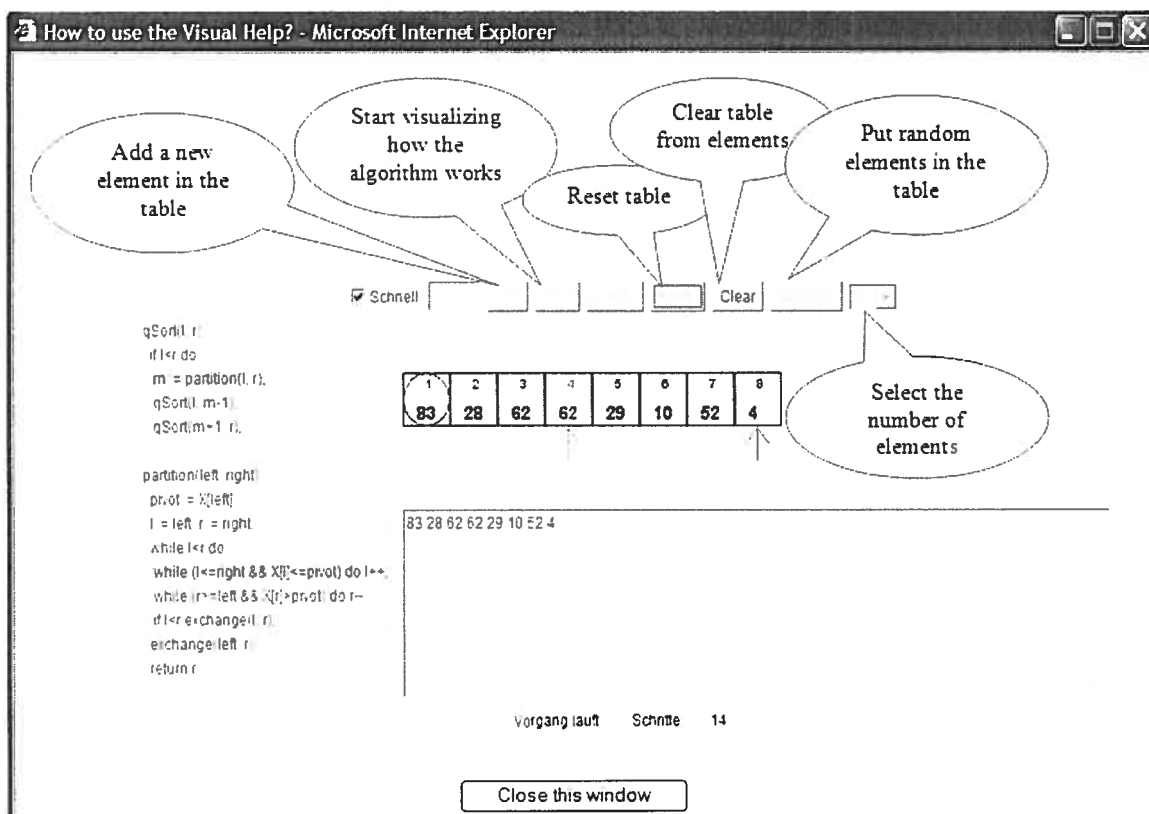



Figure 6.17 : Le tutorial pour l'aide visuelle

Quand il appuie sur le bouton *Start!* pour démarrer le test (voir Figure 6.16), les questions seront choisies selon la stratégie décrite dans la section 5.3. Nous lui présentons le test, une question à la fois. Pour chaque question, il y a trois types d'aide proposés : auditif, visuel et kinesthésique (voir Figure 6.18). Il lui suffit de cliquer sur un des liens pour accéder au type correspondant. Par exemple, l'aide visuelle déroule devant lui l'algorithme sur un ensemble de données. Il va pouvoir voir comment il fonctionne étape par étape avec ces données. Par contre, s'il utilise l'aide kinesthésique, il aura la possibilité d'introduire ses propres données avant de dérouler l'algorithme. Ainsi, l'aide kinesthésique lui permet d'essayer à son gré plusieurs jeux de données, sur lesquels il pourra dérouler l'algorithme. S'il est plus kinesthésique que visuel ou qu'auditif, cette aide l'aidera à mieux comprendre l'algorithme. L'aide auditive, quant à elle, lui présente un exemple accompagné du son. L'algorithme y est déroulé. Pour chaque étape, il entend l'explication qui lui est fournie de façon vocale. Il peut utiliser les aides qui lui conviennent le mieux. Toutefois, nous lui recommandons le type approprié selon

plusieurs critères détaillés dans la section 5.4.3. Nous utilisons la technique *annotation des liens* de la technologie *adaptation de la navigation* pour lui indiquer l'aide suggérée par le système (voir chapitre 2, section annotation des liens). Le lien correspondant est coloré d'une couleur différente (voir Figure 6.18).

Welcome Sara Doe ([sara.d@acme.com](#))






## Quicksort Auto-Evaluation

**Question 1: Why is quick sort known as "divide and conquer" algorithm?**

- Because it doesn't sort the entire list it is given.
- Because it sorts the entire list at once.
- Because it divides the list it is given into smaller lists and then sorts those smaller lists.
- Because it uses arithmetic division to speed the sorting process.


**Hints:**

-  [Auditory](#)
-  [Visual](#)
-  [Kinesthetic](#)

**Figure 6.18** : Exemple de questions pour le concept « tri rapide »

Si l'utilisateur a eu recours à une ou plusieurs des aides proposées pour répondre à la question, il se verra *parfois* obligé de les évaluer dans la page qui suit, selon le mécanisme décrit dans la section 5.3.2 (voir Figure 6.19). Dans ce cas, le bouton *Submit* ne sera activé qu'après avoir entré toutes les évaluations, ce qui lui permettra de continuer l'auto-évaluation.




Welcome Sara Doe ([sara.d@hotmail.com](mailto:sara.d@hotmail.com))



## Evaluation of helps

Please provide an evaluation of the help's :

Note: Providing an evaluation will enable the submit button.

	Auditory	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5
	Visual	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5
	Kinesthetic	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5

**Legend:**

1:	Not helpful
2:	Neutral
3:	Somewhat helpful
4:	Helpful
5:	Very helpful


**Figure 6.19 :** L'évaluation des aides dans UMAKE (évaluation nécessaire)

Dans le cas contraire, le bouton *Submit* sera toujours actif et il pourra continuer son auto-évaluation sans avoir à entrer les évaluations correspondantes (voir Figure 6.20). Les évaluations sont des entiers entre 1 et 5 comme le montrent Figure 6.19 et Figure 6.20 :

- La note 1 signifie que l'aide correspondante n'a pas aidé l'apprenant à comprendre la question courante.
- La note 2 signifie que l'apprenant est neutre vis-à-vis de cette aide.
- La note 3 signifie que l'aide correspondante a aidé un peu l'apprenant à comprendre la question courante.


- La note 4 signifie que l'aide a aidé l'apprenant à comprendre la question courante.
- La note 5 signifie que l'aide a beaucoup aidé l'apprenant à comprendre la question courante.


Welcome Sara Doe ([sara.d@botna.com](mailto:sara.d@botna.com))



## Evaluation of helps

Please provide an evaluation of the help(s):

 Visual  1  2  3  4  5

 Kinesthetic  1  2  3  4  5

**Legend:**


1: Not helpful  
 2: Neutral  
 3: Somewhat helpful  
 4: Helpful  
 5: Very helpful

**Figure 6.20 :** L'évaluation des aides dans UMAKE (évaluation non nécessaire)

A la fin du test, UMAKE présente un tableau récapitulatif de sa performance durant cette session d'auto-évaluation et un tableau récapitulatif de sa performance globale. Pour chaque question, il peut consulter le temps mis pour répondre, la nature de sa réponse, son pourcentage de succès, son pourcentage d'échec et peut accéder à la solution correspondante. Le deuxième tableau comporte d'autres informations générales comme son pourcentage de succès global, son pourcentage d'échec global, le nombre de fois pour lesquelles il a utilisé chaque type d'aide et le nombre de fois pour lesquelles il a suivi notre recommandation (voir Figure 6.21).




Welcome Sara Doe ([1313.07.1020001.com](#))



## Results and statistics

A summary of your performance during this evaluation session:

Questions	Elapsed time (sec) 	Nature of your answer	Solution
Question 1	106	True	<a href="#">View Solution 1!</a>
Question 2	14	False	<a href="#">View Solution 2!</a>
Question 3	7	True	<a href="#">View Solution 3!</a>
<b>Your success percentage in this test</b>		66.67%	
<b>Your failure percentage in this test</b>		33.33%	

An overall summary of your performance:

<b>Your overall success percentage</b>	66.67%
<b>Your overall failure percentage</b>	33.33%
<b>Number of times you used an auditory help</b>	8
<b>Number of times you used a visual help</b>	6
<b>Number of times you used an kinesthetic help</b>	6
<b>Number of times you follow our recommendation</b>	2

From the learning style test, you were 0% Auditory, 100% Visual and 0% Kinesthetic.  
 But after observing your behaviour through your utilization of the system, we can predict that you are 33.33% Auditory, 33.33% Visual and 33.33% Kinesthetic.

Would you like to save your updated learning style?  Yes  No

[Save Opinion](#)


**Figure 6.21** : La page des résultats et des statistiques

En bas de la page, UMAKE lui affiche son style d'apprentissage évolué tel qu'il a été interprété par le système en fonction de *ses choix*, de *ses évaluations* données à chaque type d'aide utilisée et de *sa performance* (voir chapitre 5, sections 5.3.1 et 5.4.3). L'apprenant peut, à tout moment, revoir une brève définition de chaque style en cliquant

sur le lien correspondant. Le système lui propose aussi de mettre à jour son profil selon ce style d'apprentissage évolué (voir Figure 6.21).

Si l'apprenant veut voir la solution d'une question donnée, il lui suffit de cliquer sur le bouton *View Solution* correspondant. Une fenêtre pop up s'ouvrira lui affichant les données de la question, la réponse correcte, sa réponse et les types d'aide qu'il a utilisés (voir Figure 6.22).

Welcome Sara Doe ([View Profile](#))



## Quicksort Auto-Evaluation: Solution for the question 2

**Question 2:** When sorting  $n$  records, Quicksort has average-case cost:

$O(\log n)$

$O(n)$

$O(n \log n)$

$O(n^2)$

**!** Hints: For this question, you used the *Auditory*, *Visual* and *Kinesthetic* hints.

**Legend:**

The right answer

Your answer


Do you want to know how you've been recommended? [Click here for details](#)

**Figure 6.22 :** La solution, sans détails, à une question répondue incorrectement

Nous avons ajouté à chaque solution des informations concernant la méthode employée pour la recommandation, une brève explication de cette méthode, et la raison de son emploi. Nous utilisons la technique du *texte élastique* de la technologie *adaptation de la présentation* pour lui afficher ces informations (voir chapitre 2, section 2.4.3.1-a). En effet, il suffit à l'apprenant de cliquer sur le lien *Click here for more details*, à côté de la question *Do you want to know how you've been recommended?*, pour que les informations recherchées s'affichent en dessous de cette question. De plus, elles sont

adaptées pour chaque apprenant. Elles dépendent de ses *choix*, de ses *évaluations* données à chaque type d'aide utilisée et de *sa performance* (voir chapitre 5, sections 5.3.1 et 5.4.3), réalisés avant qu'il ne réponde à la question courante dans le test.

Welcome Sara Doe ([View Profile](#))



## Quicksort Auto-Evaluation: Solution for the question 2

Question 2: When sorting  $n$  records, Quicksort has average-case cost:

$O(\log n)$   
  $O(n)$   
  $O(n \log n)$   
 ...

**!** **Hints:** For this question, you used the *Auditory*, *Visual* and *Kinesthetic* hints.

**Legend:**

The right answer  
 Your answer

Do you want to know how you've been recommended?

We recommended you the auditory help related to the Knowledge-Based Filtering Method which determines the most useful help (the one you liked the most and helped you to answer correctly most of times) based on your own profile.

We used this method because the predictions computed related to the Collaborative Filtering Method didn't allow us to decide which help is the best for you. [Click here to hide the details](#)

Close this window


**Figure 6.23 :** La solution, avec détails, à une question répondue incorrectement (la première raison pour l'utilisation du filtrage à base de connaissances)

Par exemple, pour une question donnée, nous pouvons utiliser la méthode du filtrage à base de connaissances afin de recommander une aide, pour l'une des raisons suivantes :

- La méthode du filtrage collaboratif ne nous a pas permis de trancher sur le type d'aide à recommander (voir Figure 6.23).

- Aucun utilisateur du voisinage de l'utilisateur courant n'a répondu à cette question avant (voir Figure 6.24).
- L'utilisateur courant fait partie des premiers utilisateurs du système, c'est ce qu'on appelle le démarrage à froid du système (voir Figure 6.25).

We come Sarz Doe [View Profile](#)



## Quicksort Auto-Evaluation: Solution for the question 1

**Question 1:** What is the definition of the quicksort?

- It picks an element from the array (the pivot), partitions the remaining elements into those greater than and less than this pivot, and recursively sorts the partitions.
- It splits the items to be sorted into two groups, recursively sorts each group, and merges them into a final sorted sequence. Runtime is  $O(n \log n)$ .

**Hints:** For this question, you used 0 hints.

**Legend:**

- The right answer
- Your answer

Do you want to know how you've been recommended?

We recommended you the auditory help related to the Knowledge-Based Filtering Method which determines the most useful help the one you liked the most and helped you to answer correctly most of times based on your own profile.

We used this method because no one in your neighbourhood faced this question before.

[Click here to learn the details.](#)

**Figure 6.24 :** La solution, avec détails, à une question répondue incorrectement (la deuxième raison pour l'utilisation du filtrage à base de connaissances)

Welcome Sara Doe ([Logout](#)) ([Account](#))



## Quicksort Auto-Evaluation: Solution for the question 1

Question 1: When sorting  $N$  records, Quicksort has worst-case cost?

- $O(N)$   
  $O(N^2)$   
  $O(N \log N)$   
  $O(N^3)$

**!** **Hints:** For this question, you used the *Auditory*, *Visual* and *Kinesthetic* hints

**Legend:**

The right answer / your answer

Do you want to know how you've been recommended?

We recommended you the auditory help related to the Knowledge-Based Filtering Method which determines the most useful help (the one you liked the most and helped you to answer correctly most of times) based on your own profile.

We used this method because you were one of the first members who logged in UMAKE.


[Click here to hide the details](#)

Close this window

**Figure 6.25 :** La solution, avec détails, à une question répondue correctement (la troisième raison pour l'utilisation du filtrage à base de connaissances)

La Figure 6.26 montre un exemple des informations affichées quand nous avons utilisé la méthode du filtrage collaboratif pour recommander une aide appropriée à l'apprenant : nous lui expliquons la méthode de filtrage employée et nous lui affichons les membres de son voisinage, sur lequel nous nous sommes basés, avec les mesures de similarité correspondantes.

Welcome Sara Doe [View my profile](#)



## Quicksort Auto-Evaluation: Solution for the question 1

Question 1: When sorting  $N$  records, Quicksort has worst-case cost?

$O(N^2)$   
  $O(N)$   
  $O(N \log N)$   
  $O(N^3)$

**Hints:** For this question, you used the *Auditory*, *Visual* and *Kinesthetic* hints.

**Legend:**

The right answer (our answer)

Do you want to know how you've been recommended?

We recommended you the auditory help based on the predictions computed with respect to the Collaborative Filtering Method.

These predictions depend essentially on your past evaluations and the evaluations given by each member in your neighbourhood

Members in your neighbourhood	Similarity
<i>Soumaya Chaffar</i>	<i>0.7</i>
<i>Hicham Hage</i>	<i>0.75</i>
<i>Pierre Chalfoun</i>	<i>0.8</i>

[Click here to hide the details](#)

Close this window

**Figure 6.26 :** La solution, avec détails, à une question répondue correctement (l'utilisation du filtrage collaboratif)

Après avoir consulté les différentes solutions, l'apprenant peut cliquer sur le bouton *Save Opinion* situé en bas de la page des résultats et des statistiques (voir Figure 6.21) pour passer à la page finale du système. Cette page lui présente plusieurs options comme refaire le même test, faire un autre test, refaire le même test mais avec les questions désordonnées (pour voir s'il a bien assimilé ses erreurs) ou retourner à la page principale (voir Figure 6.27). S'il opte pour la troisième option, il sera redirigé vers la page où il doit sélectionner le nombre de questions qui composeront son test.

Welcome Sara Doe ([sara.d@university.com](#))

## End of the Auto-Evaluation

Your profile has been updated successfully.

Now, please choose one of these options to continue:

- Repeat the same quiz
- Take another quiz
- Repeat the same quiz with shuffled questions
- Return to the Main Page

Thank you for using UMAKE, we hope that our system was helpful for you and that you'll do better next time.

**Figure 6.27** : La fin de l'auto-évaluation

## 6.4. Validation de UMAKE

Dans cette section, nous présentons la validation de notre système et les différents résultats obtenus.

### 6.4.1. Mise en contexte

Le domaine d'application est les structures de données et plus précisément le concept tri rapide « Quick sort ». De ce fait, nous avons choisi, pour la version préliminaire de UMAKE, des étudiants en informatique. La plupart finissent leur bac cette année ou sont en maîtrise. Nous avons donc des utilisateurs quasiment du même niveau.

Comme nous l'avons mentionné précédemment, UMAKE emploie une technique de filtrage *hybride (HY)* pour déterminer l'aide à recommander à l'apprenant. Cette technique commute entre le *filtrage collaboratif (CF)* et le *filtrage à base de connaissances (KB)*.

Pour des fins d'évaluation, UMAKE a été testé avec chacune de ces trois techniques (CF, KB et HY) sur trois populations différentes (voir Tableau 6.1). Il y a eu un total de 68 personnes qui ont participé à l'évaluation du système pendant une période d'environ 5 jours. Ces personnes ont été départagées aléatoirement sur les trois techniques pour constituer les trois populations.

**Tableau 6.1** : Nombre d'utilisateurs par technique de filtrage employée

	<b>Nombre de personnes</b>
<b>Technique CF</b>	24
<b>Technique KB</b>	24
<b>Technique HY</b>	20
<b>Nombre total de personnes</b>	68

Notre base de questions contenait 12 questions, 4 aides auditives, 7 aides visuelles et 6 aides kinesthésiques. Les aides sont réparties uniformément sur les questions tirées des notes du cours de structures de données IFT2010 de l'université de Montréal et de SparkNotes [URL 4], un site qui propose des notes de cours en ligne et des quiz associés pour différentes disciplines telles que les mathématiques, l'histoire, etc.

Les quiz proposés peuvent comporter 3 questions, 5 questions ou 7 questions (voir Figure 6.16).

L'analyse des résultats a été faite à partir des données récupérées dans les fichiers log en format XML enregistrés pour chaque utilisateur.

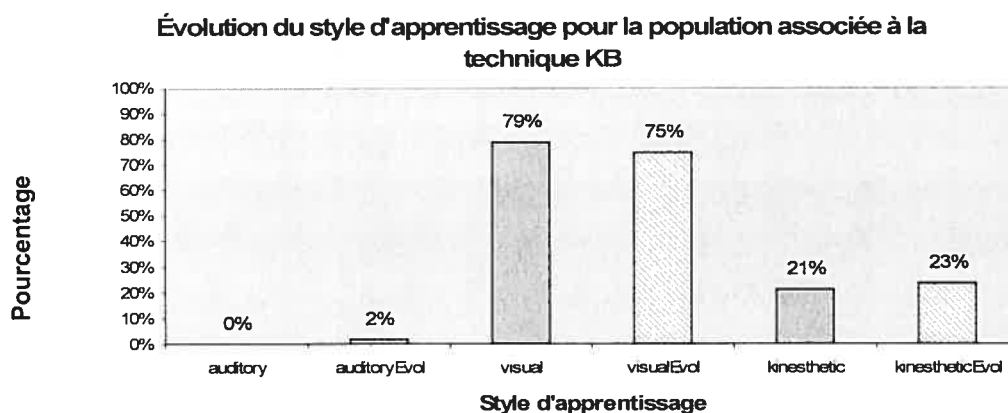
*Hypothèse* : L'objectif principal de cette évaluation est de vérifier si nous avons fait le bon choix d'opter pour la technique hybride. Est-ce que cette technique nous permet d'obtenir de meilleurs résultats par rapport aux deux autres techniques mentionnées précédemment ou non ?



## 6.4.2. Style d'apprentissage

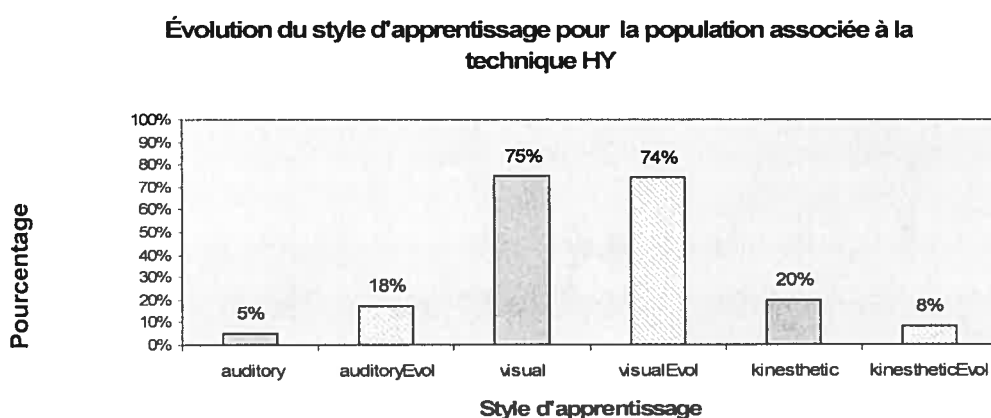
Un utilisateur doit s'enregistrer avant d'accéder aux différentes fonctionnalités de UMAKE. Lors de la deuxième étape de l'enregistrement, il répond à un questionnaire psychotechnique tiré de [URL 1] afin que le système puisse initialiser son style d'apprentissage (voir chapitre 5, section 5.3.1). Durant les tests ou les quiz, UMAKE observe ses choix, ses préférences, les notes attribuées aux différentes aides, sa performance, etc. et détermine son style d'apprentissage évolué (voir chapitre 5, section 5.4.3).

Nous avons établi un histogramme concernant l'évolution du style d'apprentissage pour chaque population associée aux techniques de recommandation appliquées KB et HY (voir Figures 6.29 et 6.30). Les styles d'apprentissage n'ont pas changé pour la population associée à CF. C'est pour cette raison que nous n'avons pas établi un histogramme pour cette population. Ceci s'explique par le fait que l'algorithme CF se base sur les votes des apprenants similaires pour recommander l'aide appropriée à l'apprenant courant (voir chapitre 5, section 5.4.3.1), mais en aucun cas il ne permet de vérifier si tel ou tel type d'aide a été plus bénéfique à l'apprenant durant ses sessions d'évaluation et de mettre à jour son style d'apprentissage évolué en fonction de ce résultat.



**Figure 6.28** : Évolution du style d'apprentissage pour la population associée à la technique KB

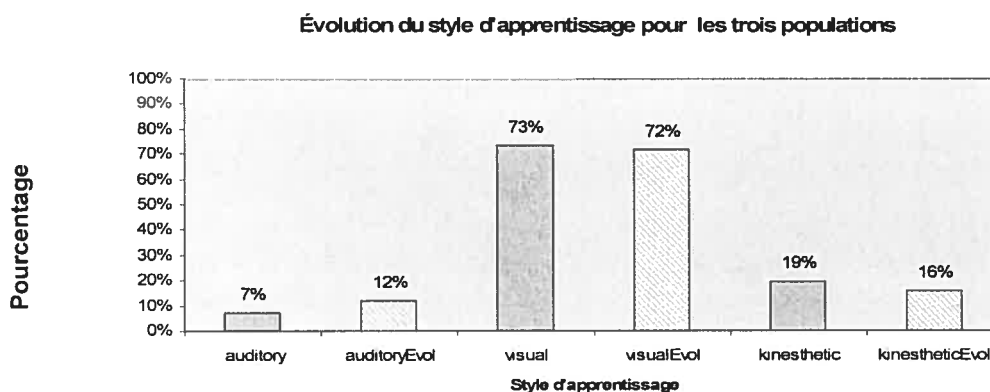
La Figure 6.28 montre qu'il n'y a presque pas de changement dans le style d'apprentissage des apprenants associées à la technique KB. Cette technique permet de déterminer l'aide qui a été la plus bénéfique et la plus utile à l'apprenant en se basant sur la nature de ses réponses (correctes ou incorrectes) obtenues en utilisant le type d'aide correspondant durant les sessions d'auto-évaluation précédentes et actuelle. Le fait qu'il n'y ait pas un grand changement dans le style d'apprentissage pourrait signifier que l'apprenant utilise le plus souvent l'aide qui est adaptée à son style initial et que ce type d'aide lui a été bénéfique.



**Figure 6.29 :** Évolution du style d'apprentissage pour la population associée à la technique HY

La Figure 6.29 montre qu'il y a quelques changements dans le style d'apprentissage des apprenants associés à la technique HY. Ces changements sont de 13% pour le style auditif, de 1% pour le style visuel et de 12% pour le style kinesthésique. La technique hybride se base sur les deux techniques : filtrage collaboratif et filtrage à base de connaissances. C'est la dernière technique qui met à jour le style d'apprentissage évolué en se basant sur l'aide la plus bénéfique et la plus utile à l'apprenant (voir Figure 6.28). Donc normalement, nous n'aurions pas dû avoir des changements importants dans le style d'apprentissage dans cette figure pour les mêmes raisons qu'on a évoquées pour la population associée à la technique KB. L'explication la plus plausible pour ces changements est qu'il ne s'agit pas de la

même population de personnes dans le sens que les deux échantillons de population se comportent différemment par rapport aux aides présentées.



**Figure 6.30 :** Évolution du style d'apprentissage en moyenne pour les trois populations

La Figure 6.30 montre l'évolution du style d'apprentissage en moyenne pour les trois populations. Nous remarquons qu'il y a des changements presque insignifiants pour les trois styles d'apprentissage. Ceci pourrait signifier éventuellement que le questionnaire psychotechnique [URL 1], qui est proposé à l'apprenant lors de son enregistrement dans le système (voir chapitre 5, section 5.2.1), détermine exactement le style d'apprentissage. De plus, ceci pourrait également signifier qu'un apprenant donné passe rarement d'une catégorie à une autre.

Nous voulons souligner aussi que pour les trois populations, il y a peu d'utilisateurs qui sont kinesthésiques, de même pour les utilisateurs auditifs. Toutefois, la plupart sont visuels.

Rappelons qu'à la fin de chaque test, UMAKE présente à l'utilisateur son style d'apprentissage évolué tel qu'il a été déterminé par le système (voir chapitre 5, section 5.4.3.2). Dans ce cas, l'utilisateur peut choisir de mettre à jour son style d'apprentissage en fonction de ce style ou non. Le Tableau 6.2 présente, par population, le nombre des cas dans lesquels l'utilisateur a été en accord avec le style d'apprentissage évolué présenté par UMAKE.

**Tableau 6.2 :** Nombre de fois où les utilisateurs ont accepté leur style d'apprentissage évolué

	En accord avec le style d'apprentissage évolué			
	Oui	%	Non	%
<b>CF</b>	23	95,83%	1	4,17%
<b>KB</b>	15	55,56%	12	44,44%
<b>HY</b>	16	76,19%	5	23,81%
<b>Moyenne</b>		<b>75,86%</b>		<b>24,14%</b>

Pour ne pas tomber dans la confusion, nous voulons clarifier que pour une population donnée, le nombre qui est dans la colonne « Oui » représente le nombre de tests à la fin desquels les apprenants de cette population ont accepté leur style d'apprentissage évolué. Respectivement, la colonne « Non » représente le nombre de tests à la fin desquels l'apprenant de cette population n'a pas accepté son style d'apprentissage évolué. Donc, si nous sommes ces deux nombres, nous trouvons le nombre total des tests effectués par cette population (un apprenant peut effectuer plusieurs quiz).

D'après le tableau ci-dessus, le style d'apprentissage évolué a été accepté en moyenne dans 75,86% des cas indépendamment de la population, ce qui constitue une bonne moyenne. Cela signifie que globalement, UMAKE détermine bien le style d'apprentissage de l'apprenant quand celui-ci interagit avec le système. Ce style était accepté presque à l'unanimité par la population (95,83%) associée à la technique CF, ce qui est normal vu que le style évolué est le même que le style initial. Il était accepté aussi par les populations associées à KB (55,56%) et à HY (76,19%) avec des pourcentages assez bons compte tenu des changements mentionnés précédemment entre les deux styles d'apprentissage dans ces populations (voir Figures 6.30 et 6.31). Ceci semble indiquer que les changements de style apportés par la technique Hybride, surtout lorsqu'ils sont relativement significatifs, sont mieux acceptés que les changements de style non significatifs apportés par la technique KB.

Remarquons toutefois qu'il y a une nette différence entre les pourcentages de la population associée à KB (55,56%) et la population associée à HY (76,19%) malgré que

ce soit la technique KB qui détermine le style évolué dans la technique HY. Nous n'avons trouvé aucune explication valable sauf le fait qu'il ne s'agit pas de la même population pour les deux, les personnes étant sensiblement différentes et conséquemment réagissent différemment aux questions et aides proposées.

### 6.4.3. Qualité de la recommandation

Dans cette partie, nous allons déterminer le nombre des cas où l'apprenant suit la recommandation du système. Nous considérons qu'un apprenant a suivi notre recommandation s'il utilise le type d'aide recommandé et si par la suite, il lui attribue une note supérieure ou égale à 3. Selon la légende présente dans les Figures 6.20 et 6.21, la note 3 signifie qu'il a apprécié l'aide recommandée et qu'elle lui a été, d'une certaine manière, bénéfique et utile.

Si l'apprenant n'utilise qu'une ou plusieurs aides autres que l'aide recommandée, cela signifie qu'il n'a pas suivi notre recommandation. S'il répond à la question sans utiliser aucune aide, ce cas est représenté par la colonne « aucune » dans les Tableaux 6.3, 6.4 et 6.5. Dans ce cas, nous ne pouvons pas affirmer s'il a suivi notre recommandation ou non, alors nous l'ignorons dans le calcul du pourcentage *effectif* des recommandations suivies (représenté par la colonne « % effectif »). Mais, nous en tenons compte dans le calcul du pourcentage total des recommandations suivies (représenté par la colonne « % total ») à titre d'information. Par exemple, nous pouvons analyser la première ligne du Tableau 6.3 de la façon suivante : pour la population CF, nous avons recommandé en total 43 aides de type visuel, il y a 18 cas où notre recommandation a été suivie par les apprenants et 12 cas où les apprenants n'ont utilisé aucune aide parmi les trois proposées. Par conséquent, nous aurons 13 cas où les apprenants n'ont pas suivie notre recommandation pour ce type d'aide.

Dans les sections suivantes, nous allons présenter le pourcentage effectif des recommandations suivies par technique et par type d'aide. Ensuite, nous comparerons les résultats obtenus par les trois techniques.

### 6.4.3.1. Population associée à la technique CF

Dans le Tableau 6.3, nous avons enregistré les données par type d'aide recommandé concernant la qualité de la recommandation pour la technique CF.

**Tableau 6.3** : Les recommandations suivies pour la population associée à la technique CF

	suivie	total	aucune	% total	% effectif
<b>V</b>	18	43	12	41,86%	<b>58,06%</b>
<b>A</b>	3	8	3	37,50%	<b>60,00%</b>
<b>K</b>	13	25	2	52,00%	<b>56,52%</b>
<b>Moyenne</b>				43,79%	<b>58,20%</b>

La technique CF se base sur les votes pour les différentes aides utilisées de l'apprenant courant et des apprenants similaires pour déterminer l'aide à recommander. Nous remarquons qu'en utilisant cette technique, nos recommandations pour l'aide visuelle, l'aide auditive et l'aide kinesthésique sont suivies respectivement dans 58,06%, 60,00% et 56,52% des cas (voir Tableau 6.3). Nous constatons de ces chiffres que cette technique donne des résultats globalement bons. La recommandation pour l'aide auditive a été suivie le plus avec un pourcentage de 60,00%. Toutefois, notons qu'il y a peu de recommandations fournies pour ce type d'aide avec un total de 8 recommandations, ce qui est tout à fait normal parce qu'il y a peu d'utilisateurs auditifs.

### 6.4.3.2. Population associée à la technique KB

Dans le Tableau 6.4, nous avons enregistré les données par type d'aide recommandé concernant la qualité de la recommandation pour la technique KB.

**Tableau 6.4 :** Les recommandations suivies pour la population associée à la technique KB

	suivie	total	aucune	% total	% effectif
<b>V</b>	15	45	19	33,33%	<b>57,69%</b>
<b>A</b>	2	7	2	28,57%	<b>40,00%</b>
<b>K</b>	18	31	6	58,06%	<b>72,00%</b>
<b>Moyenne</b>				39,99%	<b>56,56%</b>

La technique KB permet de recommander l'aide qui a été la plus bénéfique et la plus utile à l'apprenant en se basant sur la nature de ses réponses (correctes ou incorrectes) obtenues en utilisant le type d'aide correspondant durant les sessions d'auto-évaluation précédentes et actuelle. Cette technique permet d'obtenir des résultats au-delà de la moyenne pour l'aide visuelle et l'aide kinesthésique avec un pourcentage de 57,69% pour la première et un pourcentage de 72,00% pour la deuxième (voir Tableau 6.4). Toutefois, l'aide auditive n'a pas connu le même succès : sa recommandation a été suivie dans 40,00% des cas. Ce type d'aide reste le moins recommandé avec un total de seulement 7 recommandations. Ceci est expliqué par le fait qu'il y a un bas pourcentage d'utilisateurs auditifs.

#### 6.4.3.3. Population associée à la technique HY

Comme nous l'avons mentionné précédemment, la technique HY commute entre la technique CF et la technique KB afin de déterminer l'aide à recommander. La technique CF est appliquée dans la plupart des cas en premier lieu, et si elle ne permet pas de trancher sur le type d'aide à recommander, nous faisons appel à la technique KB (voir chapitre 5, section 5.4.3).

D'après le Tableau 6.5 ci-dessous, nous constatons que nos recommandations pour les différents types d'aide, calculées avec la technique HY, ont été suivies dans la plupart des cas avec 75,00% pour l'aide visuelle, 66,67% pour l'aide auditive et 80,00% pour l'aide kinesthésique. La recommandation pour l'aide auditive reste toujours la moins suivie. De plus, elle n'a été recommandée que 5 fois. Ceci est expliqué par les mêmes raisons qu'on a évoquées pour les deux autres techniques de filtrage.

**Tableau 6.5 :** Les recommandations suivies pour la population associée à la technique HY

	suivie	total	aucune	% total	% effectif
<b>V</b>	12	41	25	29,58%	<b>75,00%</b>
<b>A</b>	2	5	2	40,00%	<b>66,67%</b>
<b>K</b>	7	28	19	25,00%	<b>80,00%</b>
<b>Moyenne</b>				31,53%	<b>73,89%</b>

#### 6.4.3.4. Comparaison des trois techniques

Le Tableau 6.6 est un tableau récapitulatif qui permet de comparer la qualité de la recommandation dans les trois techniques.

**Tableau 6.6 :** La comparaison de la recommandation dans les trois techniques

	CF	KB	HY	Moyenne effective
<b>V</b>	58,06%	57,69%	75,00%	63,59%
<b>A</b>	60,00%	40,00%	66,67%	55,56%
<b>K</b>	56,52%	72,00%	80,00%	69,51%
<b>Moyenne par technique</b>	58,20%	56,56%	73,89%	62,88%

Ce tableau montre que la technique hybride HY a des meilleurs résultats par rapport à ses deux concurrentes que ce soit globalement ou même par type d'aide. En effet, nous remarquons que la recommandation fournie par la technique HY a été suivie en moyenne dans 73,89% des cas, ce qui représente un bon résultat. Cependant, elle a été suivie en moyenne dans 58,20% et 56,56% des cas respectivement pour la technique CF et la technique KB.

Le Tableau 6.6 nous permet aussi de connaître d'autres résultats généraux. En effet, la recommandation de UMAKE a été suivie en moyenne dans 62,88% des cas durant la période d'évaluation, et ce indépendamment de la technique utilisée et du type d'aide recommandé. Nous pouvons également tirer de ce tableau la moyenne globale de la recommandation par type d'aide : la recommandation a été suivie dans 63,59% des cas visuels, 55,56% des cas auditifs et 69,51% des cas kinesthésique. L'aide auditive reste



toujours la moins recommandée pour les mêmes raisons citées dans les trois sections précédentes.

#### 6.4.4. Idée de la recommandation d'aide : utile ou non?

Après avoir vérifié que notre recommandation ait été suivie dans la plupart des cas par les apprenants, nous avons voulu vérifier également si l'idée de « recommandation d'aide » dans un quiz est utile ou non. Cette idée est originale, nouvelle et a été la base de la conception de UMAKE.

Pour ce faire, nous avons écarté du nombre total de réponses, soit 233 réponses (moyenne de 3,43 questions par apprenant), celles où l'apprenant n'a choisi aucune aide. Nous avons donc conservé un total de 143 réponses parmi lesquelles l'apprenant aurait pu utiliser l'aide recommandée ou pas.

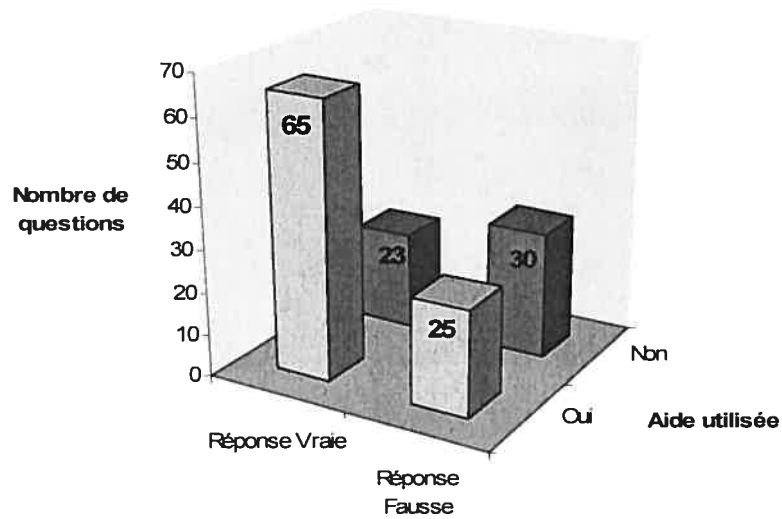
Les résultats obtenus sont représentés par le Tableau 6.7 et la Figure 6.31 ci-dessous.

**Tableau 6.7 : L'utilisation de l'aide et son impact sur la réponse donnée**

	Aide utilisée			
	Oui	%	Non	%
<b>Réponse Vraie</b>	65	45,45%	23	16,08%
<b>Réponse Fausse</b>	25	17,48%	30	20,98%

Remarquons que les apprenants ont donné plus de réponses correctes en utilisant l'aide recommandée que le cas contraire. De fait, le premier cas est représenté sur la Figure 6.31 par 65 réponses, tandis que le deuxième est représenté par 23 réponses.

### Impact de la recommandation de l'aide dans UMAKE




**Figure 6.31 :** L'utilisation de l'aide et son impact sur la réponse donnée

Nous pouvons également constater que les apprenants ont eu moins de réponses fausses avec l'utilisation de l'aide recommandée que sans l'avoir utilisée. En effet, nous avons obtenu 25 réponses dans le premier cas et 30 réponses dans le deuxième.

D'après ces chiffres, nous pouvons dire que le fait de recommander de l'aide pour les questions a un impact positif sur la nature des réponses données par la suite. Mais il faudrait le vérifier encore avec une expérience à une plus grande échelle pour l'affirmer.

#### 6.4.5. Feedback des utilisateurs

Dans le menu de navigation de la page principale, nous avons ajouté, pour des buts d'évaluation, un lien *Feedback*. A travers ce lien, l'apprenant exprime son avis concernant l'usage et l'objectif du système et peut ajouter des commentaires au besoin (voir Figure 6.32).

<p>Welcome Sara Doe</p> <hr/> <p>Available topics</p> <p><a href="#">Quality</a></p> <p><a href="#">Efficiency</a></p> <p><a href="#">Usability</a></p> <p><a href="#">Helpdesk</a></p> <hr/> <p>Member Tools</p> <p><a href="#">Create Account</a></p> <p><a href="#">Manage My Profile</a></p> <p><a href="#">Change Password</a></p> <p><a href="#">Forgot Password</a></p> <p><a href="#">Help</a></p> <p><a href="#">Sign Out</a></p> 	<h2>Feedback</h2> <p>Thank you for answering the following questions. Your feedback is important for us.</p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 60%;"></th> <th style="width: 5%;">1</th> <th style="width: 5%;">2</th> <th style="width: 5%;">3</th> <th style="width: 5%;">4</th> <th style="width: 5%;">5</th> <th style="width: 15%;">Legend</th> </tr> </thead> <tbody> <tr> <td>S1: The system's goal is clear</td> <td style="text-align: center;"><input type="radio"/></td> <td style="text-align: center;"><input type="radio"/></td> <td style="text-align: center;"><input type="radio"/></td> <td style="text-align: center;"><input type="radio"/></td> <td style="text-align: center;"><input checked="" type="radio"/></td> <td>1 Totally disagree</td> </tr> <tr> <td>S2: The system's usage is intuitive</td> <td style="text-align: center;"><input type="radio"/></td> <td style="text-align: center;"><input type="radio"/></td> <td style="text-align: center;"><input type="radio"/></td> <td style="text-align: center;"><input type="radio"/></td> <td style="text-align: center;"><input checked="" type="radio"/></td> <td>5 Totally agree</td> </tr> </tbody> </table> <p>Comments</p> <div style="border: 1px solid black; height: 40px; width: 100%;"></div> <p style="text-align: center;"><input type="button" value="Send Feedback"/></p>		1	2	3	4	5	Legend	S1: The system's goal is clear	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	1 Totally disagree	S2: The system's usage is intuitive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	5 Totally agree
	1	2	3	4	5	Legend																
S1: The system's goal is clear	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	1 Totally disagree																
S2: The system's usage is intuitive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	5 Totally agree																

**Figure 6.32 :** Le formulaire du *feedback* des apprenants

L'avis de l'apprenant est une note entre 1 et 5 : 1 pour signifier son désaccord concernant l'usage intuitif et l'objectif du système et 5 pour indiquer qu'il est entièrement d'accord avec l'usage intuitif et l'objectif clair du système.

Nous avons récolté les différentes notes concernant ces deux questions et nous avons calculé leurs moyennes (voir Tableau 6.8).

**Tableau 6.8 :** Le feedback des utilisateurs

	Moyenne des notes
L'objectif du système est clair	4,7
L'usage du système est intuitif	4,5

D'après ce tableau, nous avons obtenu une bonne moyenne (4,5/5 et 4,7/5) pour les deux questions. De plus, nous avons reçu des commentaires de satisfaction et d'encouragement très positifs.

#### **6.4.6. Conclusion**

Cette évaluation préliminaire nous permet de vérifier notre hypothèse de départ concernant le choix de la technique de recommandation : il s'est avéré que nous avons fait le bon choix en optant pour la technique HY. En effet, en comparant cette technique avec CF et KB, nous avons pu voir qu'elle fournit une recommandation de meilleure qualité.

Ces différentes analyses nous permettent également de répondre à quelques interrogations concernant l'intérêt d'un tel système, à savoir si le fait de recommander de l'aide est utile ou non. Les résultats obtenus nous laissent penser que c'est le cas.

## Chapitre 7 : Conclusion

Notre objectif principal, avant même la conception du système décrite tout au long de ce mémoire, était de *recommander* à l'apprenant l'aide *appropriée* pour une question donnée. Le fait de *générer dynamiquement des quiz* fait partie du domaine de l'évaluation de l'apprenant. Par contre, le fait de proposer des aides *adaptées* aux différents styles d'apprentissage, et d'utiliser des technologies telles que *l'adaptation de la présentation* et *l'adaptation du support de la navigation* fait partie des travaux de l'hypermédia adaptatif (voir chapitre 2, section 2.4.3). Que voulons-nous dire par « aide appropriée »? Quelle aide serait « appropriée » à l'apprenant? Ou en d'autres termes, quelle aide lui serait la plus bénéfique? Durant la conception de notre système, nous avons voulu focaliser nos efforts sur les moyens permettant à l'apprenant de comprendre une question. Dans notre approche, nous avons pris en considération le fait que chaque personne a une façon particulière d'apprendre et d'assimiler les connaissances, c'est ce que nous appelons le style d'apprentissage (voir chapitre 5, section 5.3.1). Nous avons utilisé deux méthodes pour déterminer le style d'apprentissage de l'apprenant : les questionnaires et l'observation de son comportement et de ses choix durant son interaction avec le système. UMAKE a pour but de recommander à l'apprenant une aide qui lui fera mieux comprendre la question posée. A travers ce mémoire, nous avons détaillé notre approche pour atteindre ces objectifs.

Afin de répondre à ces objectifs, UMAKE utilise une technique de recommandation hybride qui *commute* entre *le filtrage collaboratif* et *le filtrage à base de connaissances*. L'algorithme du filtrage collaboratif *modifié* est une version adaptée à notre contexte de l'algorithme original : il prend en considération non seulement les goûts des apprenants similaires et ceux de l'apprenant courant pour les différentes aides proposées pour les questions auxquelles il a déjà répondu auparavant, mais aussi son *historique* concernant la question courante et notamment ses anciens votes pour les différentes aides associées lors des sessions d'auto-évaluation antérieures (voir chapitre 5, section 5.4.3.1). En effet, comme nous l'avons mentionné dans la section 5.3 du chapitre 5, un test peut aussi comprendre, à part les questions nouvelles pour l'apprenant,

des questions auxquelles il a répondu incorrectement lors des sessions d'auto-évaluation antérieures. Quant à l'algorithme du filtrage à base de connaissances, il tient compte du *comportement* de l'apprenant durant son interaction avec le système : les types d'aide utilisés, les votes correspondants et sa performance dans chaque question (voir chapitre 5, section 5.4.3.2). UMAKE applique cette technique de recommandation hybride en *temps réel* afin de déterminer l'aide à recommander avant d'afficher chaque question du quiz. Notre méthodologie a été vérifiée et les techniques employées ont montré qu'elles répondent à notre problématique (voir chapitre 6, section 6.4). En effet, le Tableau 6.6 montre que la recommandation fournie par la technique hybride a été suivie dans 73,89% des cas par l'apprenant (contre 58,20% pour la technique CF et 56,56% pour la technique KB). De plus, les résultats obtenus dans la section 6.4.4 du chapitre 6 montrent que l'intérêt d'un tel système est démontré : l'idée *originale et nouvelle* de « recommander de l'aide » pour les questions s'est avérée utile. De fait, sur l'ensemble des aides recommandées, 45,45% de celles utilisées ont permis à l'apprenant de répondre correctement aux questions. Par contre, l'utilisation d'une aide non recommandée a permis de répondre correctement aux questions dans seulement 16,08% du temps (voir Tableau 6.7).

Après avoir comparé notre système avec ELM-ART [Weber & Brusilovsky 01], SIETTE [Guzman *et al.* 05] et avec le système de Gejji [Gejji 01] (voir chapitre 5, section 5.5), UMAKE s'est distingué par le fait qu'il est le seul système à laisser l'apprenant donner un feedback (ses évaluations pour les différents types d'aide utilisés et son avis concernant le style d'apprentissage évolué). De plus, il présente une variété d'informations pour l'apprenant à la fin de chaque test : à part la nature de chaque réponse (correcte ou incorrecte), l'apprenant peut aussi connaître le temps mis pour répondre à chaque question, la solution et les aides utilisées correspondantes. Il peut même connaître la technique avec laquelle UMAKE lui a recommandé telle aide pour telle question avec une explication de la raison de l'emploi de cette technique (rappelons que UMAKE *commute* entre le filtrage collaboratif et le filtrage à base de connaissances). UMAKE l'informe également comment son style d'apprentissage a évolué durant son interaction avec le système.

Nous pouvons *généraliser* la méthodologie de UMAKE à n'importe quel domaine : il suffit que les questions et les aides associées soient construites pour que le système réponde efficacement. UMAKE travaille avec trois aides, mais ce nombre peut-être augmenté sans problème. De plus, le langage de programmation utilisé est *multiplateforme*, donc nous pouvons greffer UMAKE dans différents systèmes d'apprentissage. En définitif, il existe de nombreuses applications de ce système qui restent à exploiter, notamment le domaine académique et le domaine de la conduite automobile.

Dans le premier domaine, UMAKE peut, non seulement être utilisé par des étudiants lors de leur évaluation (ou auto-évaluation) en ligne concernant des cours offerts par l'université, mais aussi par des étudiants du niveau secondaire ou primaire. Dans le cas des étudiants de l'université, comme nous l'avons vu dans la partie validation, la plupart des étudiants qui ont utilisé UMAKE l'ont apprécié puisqu'il semble les avoir aidés à mieux comprendre les questions et ainsi obtenir de meilleurs résultats. Dans le deuxième cas, nous savons que l'utilisation des ordinateurs dans le milieu secondaire ou primaire est une pratique commune. Généralement, les étudiants de ces milieux, surtout au primaire, pourraient trouver utile le questionnaire psychotechnique mais pourraient incorrectement s'évaluer. L'utilisation de UMAKE pourrait ainsi s'avérer très bénéfique autant pour l'enseignant que pour l'étudiant. D'une part, UMAKE permettra ultimement à l'étudiant de connaître son style d'apprentissage initial, mais surtout de suivre l'évolution de celui-ci. D'autre part, UMAKE permettra aussi à l'enseignant de surveiller l'évolution du style d'apprentissage de ses étudiants et d'adapter sa méthode d'apprentissage en fonction de ces styles.

Dans le deuxième domaine, UMAKE peut également servir d'outil d'évaluation ou d'auto-évaluation pour les gens qui veulent apprendre le code de la route. Imaginons qu'une question présente une situation donnée sur la route et demande à l'apprenant si cette situation est légale ou non. Si l'apprenant est auditif, il peut utiliser l'aide auditive qui lui présentera la situation par le biais d'une explication sonore. S'il est visuel, il peut utiliser l'aide visuelle qui lui présentera une image de la situation. S'il est kinesthésique,

il pourra simuler la situation lui-même en ajoutant des éléments sur la route (camions, voitures, motos...) et ainsi, il comprendra mieux la question.

Dans UMAKE, à chaque question sont associés trois types d'aide différents correspondants aux trois principaux styles d'apprentissage. Il est utile de signaler que le fait de fabriquer trois aides différentes pour chaque question requiert un temps non négligeable. Le fait que UMAKE utilise des aides multimédias pourrait pénaliser l'apprenant au niveau du temps requis pour le chargement de ces aides compte tenu de la taille de celles-ci.

Nous pourrions étendre les fonctionnalités de UMAKE en ajoutant un suivi personnalisé par courriel afin de permettre à l'apprenant de connaître régulièrement certains détails de son profil comme son style d'apprentissage évolué et sa performance globale dans les différents tests qu'il a effectués. Il pourra aussi être informé des dernières mises à jour du système comme des dernières aides et les derniers concepts qui ont été ajoutés.

Dans le futur, nous comptons également expérimenter UMAKE sur plus d'utilisateurs et varier les types de population.



## Bibliographie

[Abdel Razek *et al.* 03] Abdel Razek M., Frasson C., Kaltenbach M. *Web Course Self-Adaptation*. The 2003 IEEE/WIC International Conference on Intelligent Agent Technology (IAT 2003), October, 2003, Halifax, Canada, pp. 614-617.

[Adomivicius & Tuzhilin 05] Adomivicius, G., Tuzhilin, A. *Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions*. IEEE Transactions on knowledge and data engineering, Vol. 17, No. 6, June, 2005, pp. 734-749.

[Alspector *et al.* 97] Alspector, J., Kolcz, A., Karunanithi, N. *Feature-based and clique-based user models for movie selection: a comparative study*. User Modeling and User-Adapted Interaction, Vol. 7, No. 4, 1997, pp. 279–304.

[Ardissono *et al.* 99] Ardissono, L., Console, L., Torre, I. *Exploiting User Models for Personalizing News Presentations*. In Proceedings of the 2<sup>nd</sup> Workshop on Adaptive Systems and User Modeling on the WWW, 1999, Canada, pp. 13-20.

[Ardissono & Goy 00a] Ardissono, L., Goy, A. *Dynamic generation of adaptive web catalogs*. In Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, August, 2000a, Trento, Italy, pp. 5–16.

[Ardissono & Goy 00b] Ardissono, L., Goy, A. *Tailoring the interaction with users in web stores*. User Modeling and User-Adapted Interaction, Vol. 10, No. 4, 2000b, pp. 251–303.

[Beaumont 98] Beaumont, I. *User Modeling in the Interactive Anatomy Tutoring System ANATOM-TUTOR*. In Adaptive Hypertext and Hypermedia, P. Brusilovsky, A. Kobsa and J. Vassileva, Eds. Kluwer Academic Publishers, The Netherlands, 1998, Ch. 4, pp. 91-116.

[Bental *et al.* 00] Bental, D., Cawsey, A., Pearson, J., Jones, R. *Adapting Web-Based Information to the Needs of Patients with Cancer*. In Proceedings of Adaptive Hypermedia and Adaptive Web-Based Systems, August, 2000, Trento, Italy, pp. 27-37.

[Bontcheva & Wilks 05] Bontcheva, K., Wilks, Y. *Tailoring Automatically Generated Hypertext*. User Modeling and User Adapted Interaction, No. 15, 2005, pp. 135-168.

[Breese *et al.* 98] Breese, J., Heckerman, D., Kadie, C. *Empirical Analysis of predictive algorithms for collaborative filtering*. In Proceedings of the 14<sup>th</sup> Conference on Uncertainty in Artificial Intelligence, UAI 98, 1998, pp. 43-52.

[Brusilovsky 04] Brusilovsky, P. *Adaptive navigation support: From adaptive hypermedia to the adaptive Web and beyond*. The Psychology Journal, Vol. 2, No. 1, 2004, pp. 7-23.

[Brusilovsky 01] Brusilovsky, P. *Adaptive Hypermedia*. User Modeling and User-Adapted Interaction, Vol. 11, 2001, pp. 87-110.

[Brusilovsky 96] Brusilovsky, P. *Methods and techniques of adaptive hypermedia*. User Modeling and User-Adapted Interaction, 6(2-3), 1996, pp. 87-129.

[Brusilovsky *et al.* 98] Brusilovsky, P., Eklund, J., Schwarz, E. *Web-based education for all: A tool for developing adaptive courseware*. In Proceedings of Seventh International World Wide Web Conference, WWW98, April, 1998, Brisbane, Australia, pp. 291-300.

[Brusilovsky & Maybury 02] Brusilovsky, P., Maybury, M., T. *From adaptive hypermedia to the adaptive web*. Communications of the ACM, Vol. 45, No. 5, May, 2002, pp. 30-33.

[Brusilovsky & Miller 99] Brusilovsky, P., Miller, P., L. *Web-Based Testing for Distance Education*. In Proceedings of World Conference of the WWW, Internet and Intranet, AACE, WebNet 99, October, 1999, Honolulu, Hawaii, USA, pp. 149-155.

[Brusilovsky & Pesin 94] Brusilovsky, P., Pesin, L. *ISIS-Tutor: An adaptive hypertext learning environment*. In Proceedings of JCKBSE94, Japanese-CIS Symposium on knowledge-based software engineering, May, 1994, EIC, Pereslavl-Zalesski, Russia, pp. 83-87.

[Brusilovsky *et al.* 96] Brusilovsky, P., Schwarz, E., Weber, G. *ELM-ART: An intelligent tutoring system on World Wide Web*. In Intelligent Tutoring Systems. Lecture Notes in Computer Science, Vol. 1086, (Proceedings of Third International Conference on Intelligent Tutoring Systems, ITS 96, Montreal, June, 1996) Berlin: Springer Verlag, pp. 261-269.

[Burke 02] Burke R. *Hybrid recommender systems: surveys and experiments*. *User Modeling and User-Adapted Interaction*, **12** (4), November 2002, pp. 331–370.

[Burke, Mobasher & Bhaumik 2005] Burke, R., Mobasher, B., Bhaumik, R. *Limited knowledge shilling attacks in collaborative filtering systems*. In Proceedings of 3rd International Workshop on Intelligent Techniques for Web Personalization (ITWP 2005), 19th International Joint Conference on Artificial Intelligence (IJCAI 2005), August, 2005, Edinburgh, Scotland, pp. 17–24.

[Campos Pimentel *et al.* 98] Campos Pimentel, M., d. G., dos Santos Junior, J., B., de Mattos Fortes, R., P. *Tools for authoring and presenting structured teaching material in the WWW*. In Proceedings of World Conference of the WWW, Internet, and Intranet, AACE, WebNet 98, November, 1998, Orlando, FL, pp. 194-199.

[Carbone & Schendzielorz 97] Carbone, A., Schendzielorz, P. *Developing and integrating a Web-based quiz generator into the curriculum*. In Proceedings of World Conference of the WWW, Internet and Intranet, AACE, WebNet 97, November, 1997, Toronto, Canada, pp. 90-95.

[Claypool *et al.* 99] Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin, M. *Combining Content-Based and Collaborative Filters in an online Newspaper*.

In Proceedings of AMC SIGIR Workshop on Recommender Systems, August 1999, Berkeley, CA.

[Cristea & De Bra 02] Cristea, A., De Bra, P. *ODL Education Environments Based on Adaptivity and Adaptability*. In Proceedings of the AACE Elearn 2002 conference, October, 2002, Montreal, Canada, pp. 232-239.

[Dara-Abrams 02] Dara-Abrams, B., P. *Applying Multi-Agent Adaptive Hypermedia to Online Learning*. Phd Thesis. Union Institute & University Graduate College, March, 2002, 270 pages.

[De Bra 99] De Bra, P. *ADAPTS: Adaptive Hypermedia for a Web-based Performance Support System*. In Proceedings of the 2<sup>nd</sup> Workshop on Adaptive Systems and User Modeling on the WWW, June, 1999, Canada, pp. 41-47.

[De Bra 98] De Bra, P. *Adaptive Hypermedia on the Web: Methods, Technology and Applications*. In Proceedings of the AACE World Conference of the WWW, Internet & Intranet, WebNet 98, November, 1998, Orlando, FL, pp. 220-225.

[De Bra & Calvi 98] De Bra, P., Calvi, L. *AHA! An open Adaptive Hypermedia Architecture*. *The New Review of Hypermedia and Multimedia* 4, 1998, pp. 115-140.

[De Bra *et al.* 99] De Bra, P., Houben, G., J., Wu, H. *AHAM: A Dexter-based Reference Model for Adaptive Hypermedia*. In Proceedings of ACM, Hypertext 99, February, 1999, Darmstadt, Germany, pp. 147-156.

[De Carolis *et al.* 98] De Carolis, B., de Rosis, F., Andreoli, C., Cavallo, V., De Cicco, M., L. *The dynamic generation of hypertext presentations of medical guidelines*. *The New Review of Hypermedia and Multimedia* 4, 1998, pp. 67-88.

[Dufour 98] Dufour, C. *Le modèle apprenant dans les architectures de systèmes tutoriels intelligents SAFARI et TrainingOffice*. Mémoire de M.Sc, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, 1998, 126 pages.

[Fink *et al.* 98] Fink, J., Kobsa, A., Nill, A. *Adaptable and Adaptive Information Provision for All Users, Including Disabled and Elderly People*. *The New Review of Hypermedia and Multimedia* 4, 1998, pp. 163-188.

[Freyne & Smyth 05] Freyne, J., Smyth, S. *Communities, collaboration and cooperation in personalized web search*. In *Proceedings of 3rd International Workshop on Intelligent Techniques for Web Personalization (ITWP 2005)*, 19th International Joint Conference on Artificial Intelligence (IJCAI 2005), August, 2005, Edinburgh, Scotland, pp. 73–80.

[Garlatti *et al.* 99] Garlatti, S., Iksal, S., Kervella, P. *Adaptive on-line information system by means of a task model and spatial views*. In *Proceedings of Second Workshop on Adaptive Systems and User Modeling on the World Wide Web*, Toronto and Banff, Canada. Computer Science Report 99-07, Eindhoven University of Technology, 1999, pp. 59-66.

[Gejji 01] Gejji, R., R. *Web-based Programmed Learning via PERL/CGI*. The American Society for Engineering Education Conference, ASEE 2001, April, 2001, Cleveland, Ohio, USA.

[Gilbert & Han 99] Gilbert, J., E., Han, C., Y. *Arthur: Adapting Instruction to Accommodate Learning Style*. In *Proceedings of World Conference of the WWW and Internet, WebNet 99*, October, 1999, Honolulu, HI, pp. 433-438.

[Guzmán *et al.* 05] Guzmán, E., Conejo, R., García-Hervás, E. *An Authoring Environment for Adaptive Testing*. *Educational Technology & Society*, 8 (3), 2005, pp. 66-76.

[Henze & Nejd1 01] Henze, N., Nejd1, W. *Adaptation in open corpus hypermedia*. In *International Journal of Artificial Intelligence in Education*, Vol. 12, No. 4, 2001, pp. 325-350.

[Hirashima *et al.* 98] Hirashima, T., Matsuda, N., Nomoto, T., Toyoda, J. I. *Context-sensitive filtering for browsing in hypertext*. In *Proceedings of the 3rd International*

Conference on Intelligent User Interfaces, IUI 98, January, 1998, San Francisco, California, USA, pp. 119-126.

[Höök *et al.* 98] Höök, K., Karlgren, J., Waern, A., Dahlbäck, N., Janson, C., G., Karlgren, K., Lemaire, B. *A Glass Box Approach to Adaptive Hypermedia*. In Adaptive Hypertext and Hypermedia, P. Brusilovsky, A. Kobsa, and J. Vassileva, Eds. Kluwer Academic Publishers, The Netherlands, 1998, Ch. 4, pp. 143-170.

[Joerding 99] Joerding, T. *A temporary user modeling approach for adaptive shopping on the Web*. In Proceedings of Second Workshop on Adaptive Systems and User Modeling on the World Wide Web, Toronto and Banff, Canada. Computer Science Report 99-07, Eindhoven University of Technology, 1999, pp. 75-79.

[Kaplan *et al.* 93] Kaplan, C., Fenwick, J., Chen, J. *Adaptive hypertext navigation based on user goals and context*. User Modeling and User-Adapted Interaction, Vol. 3, No. 3, 1993, pp. 193-220.

[Kavcic *et al.* 93] Kavcic, A., Privosnik, M., Marolt, M., Divjak, S. *Educational Hypermedia: An Evaluation Study*. In Proceedings of the 11th IEEE Mediterranean Electrotechnical Conference, MEleCon 2002, May, 2002, Cairo, Egypt, pp. 80- 84.

[Kobsa *et al.* 01] Kobsa, A., Koenemann, J., Pohl, W. *Personalised hypermedia presentation techniques for improving online customer relationships*. The Knowledge Engineering Review, Vol. 16, No. 2, 2001, pp. 111-155.

[Kobsa *et al.* 97] Kobsa, A., Nill, A., Fink, J. *Adaptive Hypertext and Hypermedia Clients of the User Modeling System BGP-MS*. In Intelligent Multimedia Information Retrieval, M. Maybury, Ed. MIT Press, 1997, Boston, MA, pp. 339 - 356.

[Mabrouk & Aïmeur 05] Mabrouk, M., Aïmeur, E. *Adaptation d'outils d'aide d'un quiz pour l'auto-évaluation*. 4<sup>ème</sup> Colloque annuel DIVA 05, Développement, intégration et évaluation des technologies de formation et d'apprentissage, Mai, 2005, Montréal, Canada.

[Milosavljevic 97] Milosavljevic, M. *Augmenting the user's knowledge via comparison*. In Proceedings of 6th International Conference on User Modeling, UM 97, June 2-5, 1997, Chia Laguna, Sardinia, Italy, pp. 119-130.

[Milosavljevic & Oberlander 98] Milosavljevic, M. and Oberlander, J. *Dynamic Electronic catalogues: helping users to help themselves*. In Proceedings of the 9th ACM Conference on Hypertext and Hypermedia (HT 98), June, 1998, Pittsburgh, PA, USA, pp. 123-131.

[Mitrovic *et al.* 96] Mitrovic, A., Djordjevic-Kajan, S., Stoimenov, L. *INSTRUCT: Modeling Students by Asking Questions*. User Modeling and User-Adapted Interaction, Vol. 6, No. 4, 1996, pp. 273-302.

[Mornar *et al.* 03] Mornar, V., Hoic-Bozic, N., Zokovic D., P. *Approaches to Online Testing in Web-based Educational Systems*. In Proceedings of IEEE Region 8 EUROCON 2003: Computer as a tool, September, 2003, IEEE 2003, Ljubljana, Slovenia, pp. 343-346.

[O'Sullivan *et al.* 02] O'Sullivan, D., Wilson, D., Smyth, B. *Improving case-based recommendation, a collaborative filtering approach*. In Proceedings of the 6th European Conference on Advances in Case-Base Reasoning, ECCBR 02, September, 2002, Aberdeen, Scotland, Royaume-Uni, pp. 278-291.

[Oberlander *et al.* 98] Oberlander, J., O'Donnell, M., Mellish, C., Knott, A. *Conversation in the museum: experiments in dynamic hypermedia with the intelligent labeling explorer*. The New Review of Multimedia and Hypermedia 4, 1998, pp. 11-32.

[Oppermann & Specht 00] Oppermann, R., Specht, M. *A context-sensitive nomadic information system as an exhibition guide*. In Proceedings of the Handheld and Ubiquitous Computing Second International Symposium, HUC 2000, September, 2000, Bristol, UK, pp. 127-142.

[Papnikolaou *et al.* 03] Papanikolaou, K., A., Grigoriadou, M., Kornilakis, H., Magoulas, G., D. *Personalizing the Interaction in a Web-based Educational Hypermedia System: the case of INSPIRE*. User Modeling and User-Adapted Interaction, Volume 13, 2003, pp. 213-267.

[Pesin 03] Pesin, L. *Knowledge testing and Evaluation in the integrated Web-Based Authoring and Learning Environment*. In Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies, ICALT 03, July, 2003, Athens, Greece, pp. 268-269.

[Pilar da Silva *et al.* 98] Pilar da Silva, D., Van Durm, R., Duval, E., Olivié, H. *Concepts and Documents for Adaptive Educational Hypermedia: a Model and a Prototype*. In the 2<sup>nd</sup> Workshop on Adaptive Hypertext and Hypermedia, HYPERTEXT 98, June, 1998, Pittsburg, USA, pp. 35-43.

[Pirotte 02] Pirotte, C. *Module d'évaluation*. Stage de formation aux techniques multimédia organisé par le département de Mathématiques et l'Unité d'Appui Multimédia de la Faculté des Sciences de Namur. Septembre 2002 – Janvier 2003, Namur, Belgique.

[Resnick *et al.* 94] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. Riedl, J. *GroupLens: an open architecture for collaborative filtering of netnews*. In Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW 94, October, 1994, Chapel Hill, Caroline du Nord, Etats-Unis, pp. 175-186.

[Sarwar *et al.* 01] Sarwar, B., Karypis, G., Konstan, J., Reidl, J. *Item-based collaborative filtering recommendation algorithms*. In Proceedings of the 10<sup>th</sup> International Conference on World Wide Web, May, 2001, Hong Kong, pp. 285-295.

[Smyth & Cotter 99] Smyth, B., Cotter, P. *Surfing the digital wave: generating personalised television guides using collaborative, case-based recommendation*. In Proceedings of the 3<sup>rd</sup> International Conference on Case-based Reasoning, , Juillet, 1999, Munich, Allemagne, pp. 561-571.



[Specht & Oppermann 98] Specht, M., Oppermann, R. *ACE - Adaptive Courseware Environment*. *The New Review of Hypermedia and Multimedia* 4, 1998, pp. 141-161.

[Stern 01] Stern, M., K. *Using Adaptive Hypermedia and Machine Learning to Create Intelligent Web-Based Courses*. Graduate School of the University of Massachusetts Amherst, Ph.D degree, September, 2001, 280 pages.

[Suryavanshi *et al.* 05] Suryavanshi, B., Shiri, N., Mudur, S. *A fuzzy hybrid collaborative filtering technique for web personalization*. In Proceedings of 3rd International Workshop on Intelligent Techniques for Web Personalization (ITWP 2005), 19th International Joint Conference on Artificial Intelligence (IJCAI 2005). August, 2005, Edinburgh, Scotland.

[Vassileva 96] Vassileva, J. *A Task-Centred (sic) Approach for User Modeling in a Hypermedia Office Documentation System*. *User Modeling and User-Adapted Interaction*, 6(2-3), 1996.

[Weber & Brusilovsky 01] Weber, G., Brusilovsky, P. *ELM-ART: An adaptive versatile system for Web-based instruction*. In *International Journal of Artificial Intelligence in Education*, Vol. 12, No. 4, 2001, pp. 351-384.

[Weber & Specht 97] Weber, G., Specht, M. *User modeling and adaptive navigation support in WWW-based tutoring systems*. *User Modeling: Proceedings of the Sixth International Conference, UM 97*, June, 1997, Sardinia, Italy, pp. 289-300.

[Wenger 87] Wenger, E. *Artificial Intelligence and Tutoring Systems*. Morgan Kaufmann Publishers, Inc., Los Altos, CA, 1987, 486 pages.

[Wu *et al.* 01] Wu, H., De Kort, E., De Bra, P. *Design Issues for General-Purpose Adaptive Hypermedia Systems*. In Proceedings of the 12th ACM Conference on Hypertext and Hypermedia, August 14-18, 2001, Aarhus, Danmark, pp. 141-150.

[Yammine *et al.* 04] Yammine K., Abdel Razek M., Aïmeur E. et Frasson C. *Discovering intelligent agent: a tool for helping students searching a library*. In Proceedings of the 7<sup>th</sup>

International Conference on Intelligent Tutoring Systems, ITS2004, Août /Septembre, 2004, Maceio, Alagoas, Brésil, pp. 720-729.

## Sites Web

[URL 1] <http://www.vark-learn.com/english/index.asp> accédé en Février 2005. Droits d'auteur à Neil D. Fleming, Christchurch, Nouvelle-Zélande et Charles C. Bonwell, Green Mountain, Colorado, États-Unis.

[URL 2] <http://www.ldpride.net/learningstyles.MI.htm#Kinesthetic%20Learners>

[URL3] <http://www.sciences.fundp.ac.be/uam/stages2002/Dreamweaver/seminaires/seminaires/seminaire13-12.ppt>

[URL 4] <http://www.sparknotes.com/cs/sorting/review/quiz.html>

[URL 5] <http://www-etud.iro.umontreal.ca:8080/~mabroukm/>

