Université de Montréal

DIA : un système de recommandation de livres dans un contexte pédagogique

par

Kamal Yammine

Département d'informatique et de recherche opérationnelle

Faculté des Arts et Sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de Maître ès (M. Sc.)
en informatique

Mai, 2005

Université de Montréal
Faculté des études supérieures

Ce Mémoire intitulé :

DIA : un système de recommandation de livres dans un contexte pédagogique

présenté par :
Kamal Yammine

a été évalué par un jury composé des personnes suivantes :

Claude Frasson, président-rapporteur
Esma Aïmeur, directrice de recherche
Yann-Gaël Guéhéneuc, membre du jury

Mémoire accepté le 20 mai 2005

Résumé

L'expansion technologique et l'arrivée des médias de masse ont multiplié et diversifié nos sources d'informations. En conséquence, nous sommes contraints de traiter un nombre croissant de données avant de parvenir à notre but ou à l'information recherchée. Chaque jour, de plus en plus de livres sont publiés et rendus disponibles sur Internet. Toutefois, trouver le bon livre au bon moment peut devenir une tâche fastidieuse et épuisante, particulièrement pour les étudiants. Ce mémoire propose l'utilisation d'un système de recommandation de livres dans un contexte pédagogique, afin de soutenir les étudiants durant la recherche de livres. En utilisant plusieurs méthodes de filtrage, nous avons conçu un système de recommandation de livres appelé DIA (*Discovering Intelligent Agent*). Le système cherche dans la base des livres de la bibliothèque de l'Université de Montréal et renvoie une liste de livres ordonnée selon le modèle de l'étudiant (son style d'apprentissage) et le contenu des livres. L'objectif principal de DIA est d'aider l'étudiant à choisir les livres qui correspondent le mieux à ses champs d'intérêt et son style d'apprentissage.

D'abord, pour chaque domaine couvert, le système emploie un ensemble de mots-clés pour effectuer une première phase d'analyse et de classification des livres de la bibliothèque. La classification se fait par la comparaison des mots-clés du domaine avec la description des livres. La description des livres est extraite à partir des bases de données de la bibliothèque, mais également à l'aide d'une recherche plus extensive sur l'Internet. Cette dernière permet de trouver diverses descriptions telles que la table des matières et la synthèse du livre. Donc, cette première classification des livres se fait par rapport au domaine couvert.

Ensuite, DIA étudie les caractéristiques de chaque étudiant pour mieux répondre à ses requêtes. Le système pose à l'utilisateur (l'étudiant) une série de questions afin d'identifier son style d'apprentissage : visuel, auditif, lecture/écriture ou-et kinesthésique. Ainsi, le système établit un profil personnalisé de l'étudiant et le compare à ceux des autres

étudiants dans le but d'identifier ceux qui partagent des profiles similaires. Un poids (rang) est assigné à chaque livre selon les préférences d'étudiants semblables, pour ensuite retourner l'ensemble des livres ordonnés selon ce poids.

Les tests effectués à l'Université de Montréal ont abouti à des résultats prometteurs. L'évaluation a été réalisée à l'aide de 112 étudiants avec une banque de 1 128 livres couvrant 4 domaines. Ces domaines étant l'intelligence artificielle, les structures de données, la programmation Java et l'apprentissage machine. Les résultats ont montré que les étudiants ont été fortement satisfaits du système.

Mots-clés : Système de recommandation, filtrage collaboratif, filtrage basé sur le contenu, norme Z39.50, recommandation de livres, bibliothèque électronique.

Abstract

Nowadays, the explosive growth of the Internet has brought us such a huge number of books, publications, and documents that students can hardly consider all of them. Finding the right book at the right time is an exhausting and time-consuming task, especially for new students who have diverse learning styles, needs, and interests. Moreover, the growing number of books in one subject can overwhelm students trying to choose the right book. This thesis studies how recommendation systems can be applied in a pedagogical context to help overcome this challenge by ranking books using several filtering methods. Based on these filtering techniques, we have designed and implemented a book recommendation system called Discovering Intelligent Agent (DIA). The system searches the Montreal University library and returns a list of ranked books related to the students' models and to the contents of the books. DIA aims to help learners select books which correspond to their fields of interest and which best match their learning styles.

First, for each covered domain, the system uses a set of computed keywords to carry out an initial analysis and classification of the library books. The classification is made by comparing the domain keywords with the book description extracted from the library databases and from a widespread Internet research to find various descriptions, table of contents, synopses related to the material. At this point the system categorizes the books according to the covered domain.

Then, DIA scans users' characteristics to better respond to their requests. Users are asked a series of questions to assign them in one of four learning style categories: visual, aural, read/write, and kinesthetic. Thus, the system establishes a personalized profile of the learner and compares it with those of other learners to identify the most similar users.

All of these factors are taken into consideration to return responses to the user. A weight (ranking) is assigned to each book according to the way in which it was previously

selected by learners who share similar profiles and the results are presented according to their relevance.

Tests conducted at Montreal University yielded successful results. Evaluations were carried out with 112 students and a bank of 1,128 books was used. This survey showed very promising results. The students were highly satisfied by the system and the achieved objectives.

Keywords: Recommendation systems, collaborative filtering, content-based filtering, Z39.50 protocol, book recommendation, e-libraries.

Table des Matières

Liste des Tableaux

Liste des Figures

À mes parents

Introduction

L'expansion technologique et l'arrivée des médias de masse ont multiplié et diversifié nos sources d'informations. En conséquence, *nous sommes contraints de traiter un nombre croissant de données avant de parvenir à notre but* ou à l'information recherchée. Par exemple, pour nous tenir au courant des événements qui nous entourent, il y a à notre disposition les nouvelles télévisées, les articles de journaux, les sites Web de nouvelles spécialisées, les cotations de la Bourse, les bulletins météorologiques, pour n'en citer que quelques-uns. Par ailleurs, lorsque nous avons besoin d'acheter un produit quelconque, nous nous retournons souvent vers les publicités télédiffusées, les panneaux publicitaires qui enlaidissent nos routes et les soldes dans les magasins. Certains magasins mettent même à notre disposition des circulaires qui arrivent directement sur le pas de notre porte. Tous ces moyens de communication ont pour effet, entre autres, d'augmenter la quantité d'informations que nous avons à analyser avant de faire notre choix.

Cette surcharge d'informations intéresse plusieurs chercheurs, notamment dans les domaines de la psychologie et de l'informatique [Achike, *et al.* 2000], [Borchers, *et al.* 1998], [Stanley, *et al.* 1997], [Fournier 1996]. En 1995, l'Office Québécois de la Langue Française, l'OQLF, a même introduit le terme « **infobésité** » pour désigner cette « *obésité informationnelle* ». Depuis une décennie, ce fléau s'est accentué avec l'arrivée de l'Internet et a commencé à avoir des répercussions de plus en plus sérieuses sur notre quotidien, en nous affectant physiquement et psychologiquement [Stanley, *et al.* 1997], [Fournier 1996].

1.1 Notre motivation

Les étudiants n'échappent pas au problème de l'infobésité. Un étudiant de maîtrise ou de doctorat, par exemple, doit traiter un flot énorme de données provenant de plusieurs sources, notamment des livres, des articles de journaux ou de conférences, des thèses ou mémoires et de l'Internet. Bien que ces flots de données puissent être bénéfiques à un individu, ils peuvent devenir trop lourds et trop encombrants pour être manipulés par une seule et même personne. Ce phénomène peut finir par en décourager certains. Dans [Achike, *et al.* 2000], la surcharge d'informations a été identifiée comme un des facteurs principaux contribuant à de faibles performances académiques de plusieurs étudiants en pharmacologie.

Les conséquences de cette infobésité dans le milieu pédagogique ne s'arrêtent pas là. Un phénomène dénommé « ***Library Anxiety*** » [Battle 2004], [Onwuegbuzie, *et al.* 2000], [Jiao, *et al.* 1999] ou « **l'anxiété des bibliothèques** » existe et qui consiste en un sentiment d'inconfort assez commun chez les étudiants, ressenti dans un environnement bibliothécaire. Ce sentiment a des ramifications cognitives, affectives, physiologiques et comportementales [Mellon 1986] et se rapporte à une disposition émotive caractérisée par une tension, une crainte, des sentiments d'incertitude, des pensées négatives et une désorganisation mentale qui apparaissent seulement quand les étudiants sont dans une bibliothèque ou lorsqu'ils souhaitent s'y rendre [Jiao, *et al.* 1996]. Des recherches ont montré que ce phénomène est présent chez les étudiants de tous les niveaux d'études, que ce soit chez ceux du premier cycle [Mellon 1986] ou des cycles supérieurs [Jiao, *et al.* 1998]. Une des causes principales de ce sentiment serait la taille des bibliothèques et leur complexité [Battle 2004], [Mellon 1986]. Ce problème risque de s'aggraver du fait que les bibliothèques universitaires se doivent de rester à jour et, donc, de se procurer de plus en plus de livres et de ressources pédagogiques.

Pour répondre au problème de surcharge d'informations, la science informatique et plus spécifiquement la communauté de la *recherche et d'extraction d'informations* étudie les manières dont l'ordinateur peut assister l'homme pour l'aider à gérer le surdosage

informationnel [Belkin, *et al.* 1992]. En particulier, ce domaine a développé des techniques de *Recherche d'Informations* (*RI*) et de *Filtrage d'Informations* (*FI*) qui permettent la personnalisation de l'information selon les goûts, les besoins et les préférences de chaque individu. Plus récemment, les *systèmes de recommandations* [Hofmann 2004], [Linden, *et al.* 2003], [Melville, *et al.* 2002], [Sarwar, *et al.* 2000] ont été employés dans le but de ne proposer que l'information pouvant intéresser leurs utilisateurs, en omettant celles qui ne correspondent pas à leur profil. Ceci a pour conséquence de pouvoir réduire la quantité de données à analyser.

Nous pensons que les systèmes de recommandations peuvent être très bénéfiques pour des étudiants qui recherchent des livres de leur bibliothèque universitaire. En effet, cet environnement contient une énorme quantité de ressources, des livres, des périodiques, des journaux scientifiques, des thèses et mémoires, des cédéroms, etc. Même si un effort considérable est consacré à la classification des livres et des ressources de manière générale, l'étudiant fera toujours face à un problème majeur : comment peut-il trouver les ressources qui lui sont le plus appropriées ? Souvent, une base de données est mise à la disposition de l'utilisateur, à partir de laquelle ce dernier peut faire ses recherches. Cette démarche est efficace lorsque l'interrogateur connaît déjà le titre ou les auteurs de l'ouvrage recherché. Si ce n'est pas le cas, l'usager doit entrer une requête constituée de mots-clefs relatifs au sujet. Cette étape exige de l'usager une bonne connaissance des mots-clefs significatifs. De plus, il faut que ces derniers correspondent aux mots réellement utilisés dans la description du livre. Lorsque la requête est envoyée, le système affiche alors toutes les ressources contenant ces mots-clefs dans leurs descriptions. Aucune information sur la pertinence du livre par rapport à l'usager en question n'est offerte.

Dans ce mémoire, nous proposons DIA [Yammine, *et al.* 2004], un système de recommandations de livres. Contrairement à la plupart des systèmes existants qui sont développés dans un contexte commercial, celui-ci est **conçu spécifiquement pour un environnement pédagogique**. Nous étudions dans cette dissertation les enjeux de cette spécificité et les avantages qu'elle amène par rapport aux systèmes existants. Nous

montrons aussi comment ce système parvient à prédire les livres pédagogiques les plus appropriés à un étudiant et nous discutons des caractéristiques et des apports uniques de notre système.

1.2 L'organisation du mémoire

Ce mémoire est divisé en 4 parties : l'état de l'art, l'architecture et la méthodologie, l'implémentation et l'évaluation et la conclusion.

Dans l'état de l'art au chapitre 2, nous passons en revue les systèmes de recommandations et leurs techniques, notamment le filtrage basé sur le contenu, le filtrage collaboratif et la méthode hybride, qui est basée sur l'application de ces deux techniques simultanément.

Au chapitre 3, nous nous intéressons plus spécifiquement aux enjeux de la recommandation de livres pédagogiques. Nous présentons quelques exemples réels de systèmes de recommandations de livres. Puis, nous analysons ces systèmes sous un angle technique et philosophique, dans le but d'évaluer leur adaptabilité à un environnement pédagogique. Finalement, nous discutons de la logique et des défis d'un système de recommandations de livres dédié spécifiquement aux milieux universitaires et nous présentons les caractéristiques générales de DIA [Yammine, *et al.* 2004].

Le chapitre 4 définit l'architecture et les méthodologies appliquées dans DIA. Nous commençons par une discussion du fonctionnement général du système pour ensuite voir en détail chacun de ses composants.

Pour mieux comprendre le système et les choix discutés au chapitre 4, le chapitre 5 se concentre sur la description de scénarios possibles d'utilisation de DIA. En donnant un exemple réel d'utilisation, nous suivons Frédéric, un étudiant recherchant des livres dans le cadre de son cours d'intelligence artificielle.

Le chapitre 6 illustre, dans un premier temps, l'implémentation et la procédure d'évaluation de DIA. Ensuite, nous analysons les résultats des tests effectués et la validation du système.

Finalement, en conclusion, la dernière partie de ce mémoire récapitule les aspects majeurs de ce travail, discute des généralités de DIA, précise les contributions principales de ce mémoire et décrit les limitations du système et les travaux futurs.

Chapitre 2
Les systèmes de filtrage

L'avènement de l'Internet a facilité la diffusion et la propagation de l'information et, ainsi, des milliers de nouveaux documents fleurissent chaque jour dans cet environnement. Cependant, cet avantage a eu comme effet secondaire le « *surdosage informationnel* », d'où une volonté de personnaliser l'information.

Depuis plusieurs années, la science informatique étudie des solutions à ce problème et a développé des techniques de *Recherche d'Informations* (*RI*) et de *Filtrage d'Informations* (*FI*) qui permettent la personnalisation de l'information selon les goûts, les besoins et les préférences de chaque individu. Ce chapitre a pour objectif de présenter le filtrage d'informations et les systèmes de recommandations. Nous l'aborderons d'abord par une description générale et la présentation de ses caractéristiques. La deuxième partie exposera en détail les différents types de filtrage et les algorithmes classiques utilisés. Enfin, la troisième section résumera les notions importantes de ce chapitre.

2.1 Le filtrage d'informations

Le filtrage d'informations est un terme « attribué à une variété de processus permettant l'acheminement de l'information adéquate aux personnes qui en ont besoin » [Belkin, *et al.* 1992] en se basant sur leurs intérêts. Le filtrage se fait par rapport à des descriptions

d'individus, souvent appelées profils et qui représentent généralement l'ensemble des champs d'intérêt à long terme de l'usager.

Les systèmes de filtrage font parvenir, au cours du temps, les documents jugés intéressants et assurent leur acheminement en permanence. En conséquence, le *FI* nous permet d'éviter de procéder régulièrement à des recherches avancées et ainsi économise notre effort et notre temps. Le *FI* est employé dans une multitude de domaines, notamment à but commercial [Bin, *et al.* 2003], [Linden, *et al.* 2003], [Sarwar, *et al.* 2000], culturel [Iwahama, *et al.* 2004], [Chen, *et al.* 2001] ou de divertissement [Gupta, *et al.* 1999].

2.1.1 Le filtrage d'informations et la recherche d'informations

La *RI* est axée essentiellement sur les techniques de *stockage*, *d'indexation* et *d'extraction* des documents textuels. Les systèmes qui utilisent cette approche requièrent, de la part de l'utilisateur, la formulation systématique de son besoin sous forme d'une requête permettant la découverte des documents recherchés. Le besoin informationnel de l'utilisateur est dynamique et temporaire. De plus, les systèmes de recherche d'informations ont tendance à gérer un stockage d'informations relativement statique. Les moteurs de recherche classiques que nous retrouvons sur la Toile (le « Web ») [Kobayashi, *et al.* 2000] ainsi que les bases de données bibliographiques emploient souvent cette méthode.

Contrairement à la *RI*, le *FI* s'effectue sur un flot continu de données. Le filtrage est défini comme l'élimination de données indésirables sur un flux entrant plutôt qu'à la recherche de données spécifiques sur ce flux. Ces systèmes utilisent le profil de l'utilisateur pour filtrer l'information. Ici, le contrôle de l'information est réalisé en se basant sur des intérêts plus statiques et à long terme. Les logiciels qui détectent les courriels non sollicités (le « *Spam* ») sont un exemple de filtrage d'informations [O'Brien, *et al.* 2003].

Bien que la *RI* et le *FI* soient souvent employés ensemble, la science informatique en fait la distinction. En raison de la divergence de leur approche, ces deux techniques

diffèrent sur un certain nombre de points, *Belkin* et *Croft* [Belkin, *et al.* 1992] ont étudié leurs différences, dont voici le résumé :

- la recherche d'informations s'occupe de la collecte et de l'organisation des documents, le filtrage de l'information se préoccupe de la distribution des documents aux personnes qui en ont besoin;
- la recherche d'informations opère sur des données ou sur un ensemble de documents plus ou moins statique ; le filtrage d'informations gère un flux de données assez dynamique;
- la recherche d'informations s'adresse à un public ayant un besoin d'informations ponctuel et dynamique contrairement au filtrage d'informations qui s'intéresse à des individus ayant des objectifs et des intérêts constants et persistants. Ce type de besoin est qualifié de statique ou de long terme;
- l'utilisation d'un profil pour le *FI* permet la personnalisation de l'information pour l'utilisateur et l'adaptation, au fil du temps, à l'évolution de ses intérêts.

2.2 Les types de filtrage

Bien qu'il existe une multitude de façons d'aborder le filtrage, deux techniques sont prédominantes : le *Filtrage Basé sur le Contenu* (*FBC*) et le *Filtrage Collaboratif* (*FC*) [Yu, *et al.* 2004]

2.2.1 Le filtrage basé sur le contenu

Une manière intuitive de filtrer est de proposer à l'utilisateur des produits semblables à ceux qu'il a aimés dans le passé. Le filtrage basé sur le contenu exploite cette idée en recommandant des éléments ayant un contenu similaire à ceux déjà appréciés par l'usager. Cette approche emploie les principes de la *RI* et utilise plusieurs de ses techniques. Dans cette section, nous discutons de la représentation des objets dans un système de *FBC* et nous présenterons deux techniques, le *modèle de l'espace vectoriel* et la *classification bayésienne.*

2.2.1.1 La représentation des objets à filtrer

La représentation des objets peut être vue comme la description de ces derniers afin de permettre au système leur analyse et leur triage. Cette étape est importante si nous souhaitons un filtrage efficace.

Comme la recherche d'informations est surtout adaptée aux documents ou aux données textuelles, il faut représenter les objets sous une forme littérale. Nous nous retrouvons face à deux possibilités :

1. les objets à filtrer sont déjà sous forme littérale, tels que des pages Web, des courriels, ou des articles : leur contenu peut être directement utilisé, d'autant plus qu'il est facilement accessible par le système;
2. les objets sont sous format non textuel, par exemple des films ou de la musique : il est nécessaire de mettre à la disposition du système une description du contenu.

Prenons le cas où nous désirons filtrer des films selon les goûts des utilisateurs en utilisant le filtrage basé sur le contenu. Il est impératif de fournir au système une description de chaque production. Celle-ci peut inclure le résumé du film, les acteurs, le metteur en scène, etc.

2.2.1.2 Le modèle de l'espace vectoriel

Le modèle de l'espace vectoriel [Salton, *et al.* 1986] représente un objet sous forme d'un vecteur de termes, où un poids est associé à chacun des termes. Plus un mot est révélateur, plus sa pondération est élevée. Il existe différentes méthodes pour la détermination des poids. Par exemple, *TF-IDF* « *Term Frequency-Inverted Document Frequency* » [Billsus, *et al.* 1999], [Salton, *et al.* 1986] est une formule statistique, qui assigne un poids à un terme, proportionnel au nombre de ses occurrences dans le contenu et dans une proportion inverse au nombre de documents dans lesquels il figure. La figure 2.2-1 montre un vecteur de termes d'une page Web, où la pondération des mots équivaut à leur fréquence dans le document.

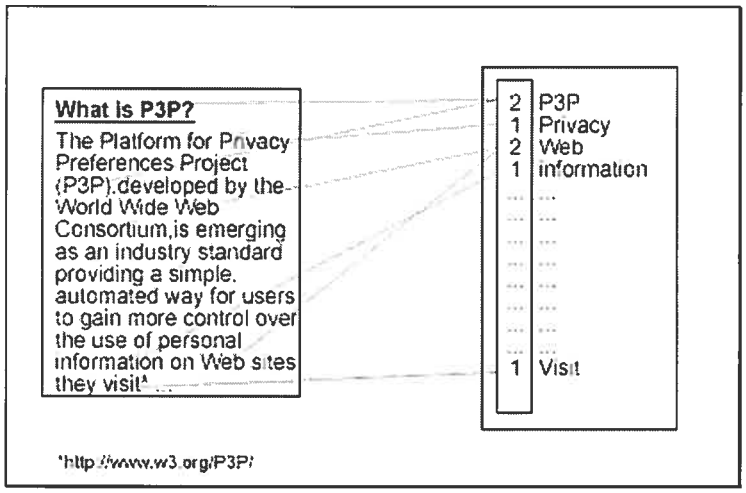


Figure 2.2-1. Le vecteur de termes d'une page Web avec un poids égal à la fréquence du mot

Parallèlement, le profil de l'utilisateur est représenté comme un vecteur de mots-clés décrivant les intérêts de l'usager. Le poids d'un mot-clé est proportionnel au besoin de l'utilisateur. La tâche principale de cette technique est de corréler le vecteur du profil de l'utilisateur avec les vecteurs des objets. En se basant sur cette corrélation, le système fournit à l'utilisateur les objets appropriés, c'est-à-dire ceux qui présentent la plus grande corrélation.

Le profil de l'utilisateur est mis à jour régulièrement à partir des objets visionnés ; si un usager indique qu'un objet l'intéresse, le vecteur de termes de cet objet est ajouté à son profil. Si un mot-clé existe déjà, en augmentant uniquement sa pondération.

2.2.1.3 La classification bayésienne

Le filtrage est traité comme un problème de classification, dont l'objectif est de calculer la probabilité qu'un objet appartienne à une des deux catégories : les objets pertinents (la catégorie positive) ou les objets non pertinents (la catégorie négative) [Mooney, *et al.* 2000]. Cette approche emploie le théorème de Bayes qui stipule : étant donné un élément d_j, la probabilité qu'il appartienne à une classe c peut être calculée comme :

$$P(c \mid d_j) = \frac{P(c)P(d_j \mid c)}{P(d_j)} \tag{2.2-1}$$

où $P(d_j \mid c)$ = la probabilité que l'objet d_j fasse partie de la classe c,

$P(c)$ = la probabilité qu'un objet appartienne à la classe c,

$P(d_j)$ = la probabilité que le document soit choisi.

Hypothèse d'indépendance

Supposons que chaque objet d_j peut être représenté par un ensemble de termes :

$$d_j = \{t_{kj} \mid k = 1,2,\ldots,r\} \tag{2.2-2}$$

L'hypothèse d'indépendance suppose que la présence d'un terme dans un document est indépendante par rapport aux autres termes [Mooney, *et al.* 2000]. En conséquence,

$$P(d_j \mid c) = \prod_{k=1}^{r} P(t_{kj} \mid c) \tag{2.2-3}$$

Finalement, nous calculons le rapport $\frac{P(positif \mid d_j)}{P(négatif \mid d_j)}$. Plus il est élevé pour un document, plus ce document est pertinent pour l'utilisateur.

2.2.1.4 Les limites du filtrage basé sur le contenu

Comme le filtrage basé sur le contenu applique des techniques de *RI*, il hérite plusieurs de leurs faiblesses. [Balabanovic, *et al.* 1997] ont décrit ces faiblesses dont voici les points les plus importants :

- il est difficile de mettre en application ce type de filtrage dans des domaines non concrets, tels que les films, la musique, les restaurants. Dans ces cas, il est compliqué de trouver les mots discriminants adéquats pour bien décrire les objets. Leur représentation, sous une forme textuelle, devient une tâche complexe et peut affecter l'efficacité du système. De plus, comme le filtrage basé sur le contenu ne

s'applique que sur des objets *textuels,* il ignore tout ce qui se rapporte aux multimédias, par exemple les vidéos, les images, ou les graphes et ne peut tirer profit de l'information qu'ils contiennent;

- un autre problème survient lorsque, à la longue, on arrive à une trop grande spécialisation d'un profil (« *over-specialization* »). Le système recommande alors des documents qui ne couvrent qu'une partie d'un domaine, en ignorant les nouveaux objets qui lui sont relatifs, car les mots-clés ne correspondront pas à ceux dans le profil;
- certains aspects importants d'un document, tels que sa qualité, sa clarté, la facilité de sa compréhension, sa complexité ou la véracité de son contenu, ne peuvent être perçus par ces systèmes. Ainsi, l'utilisateur n'a aucune précision sur la qualité des documents analysés et présentés.

2.2.2 Le filtrage collaboratif

Comme son nom l'indique, le *Filtrage Collaboratif* (*FC*) a pour principe de faire profiter un individu particulier de l'expérience collective [Basilico, *et al.* 2004], [Hofmann 2004], [Kai, *et al.* 2004]. Le filtrage ne dépend pas uniquement de l'usager actif mais implique aussi les autres utilisateurs du système. Souvent, les systèmes de filtrage collaboratif comparent les utilisateurs afin d'identifier ceux qui ont une certaine corrélation. Les recommandations sont ensuite calculées en se fondant sur ces similarités. En d'autres mots, ces systèmes recommandent à l'usager les objets appréciés par les utilisateurs qui lui sont similaires.

2.2.2.1 La représentation des préférences des utilisateurs

La collecte des préférences des utilisateurs est une tâche importante dans le processus du filtrage collaboratif. Une approche assez populaire consiste à demander aux usagers de fournir explicitement leurs préférences. Ainsi, lors de leur inscription au système et tout au

long de son utilisation, les utilisateurs notent les éléments à l'aide de « cotes » ou de « votes » numériques. Les cotations de tous les usagers forment la *matrice des cotes*[1].

Par exemple, lorsque le système affiche une liste de recommandations à un utilisateur, ce dernier peut coter la pertinence de chaque objet, suivant une échelle de 1 à 5, où 5 indique que l'objet est très apprécié. Ce retour indique au système les préférences de l'usager et permet l'adaptation des recommandations futures à ces derniers. Les préférences explicites ont l'avantage d'être assez précises et simples à implémenter, mais créent une charge supplémentaire pour les participants.

Une autre approche consiste à estimer les préférences des utilisateurs d'une manière implicite. Les préférences ne sont pas fournies par l'utilisateur directement mais sont déduites en observant ses interactions avec le système [Kelly, *et al.* 2003]. Des travaux suggèrent que le temps passé sur un document [Claypool, *et al.* 2001], [Morita, *et al.* 1994] et la quantité de défilement du document [Claypool, *et al.* 2001] peuvent être employés afin de prédire les préférences des utilisateurs.

2.2.2.2 La différence entre filtrage collaboratif et filtrage basé sur le contenu

Le filtrage basé sur le contenu (*FBC*) calcule la similitude entre les objets en se basant sur leur contenu textuel ou sur leur description. Le système présente à l'utilisateur les objets similaires à ceux qu'il a aimés dans le passé. L'approche du filtrage collaboratif (*FC*) est différente. Elle recommande les articles que d'autres utilisateurs, partageant les mêmes préférences, ont aimés. Donc, plutôt que de calculer la similarité lexicale des objets, on calcule la similarité entre les usagers. D'ailleurs, dans les systèmes de filtrage

[1] Appelée aussi matrice des utilisateurs

« purement » [2] collaboratif, aucune analyse des objets n'est effectuée, seule leur identification est nécessaire [Balabanovic, *et al.* 1997].

Le filtrage collaboratif offre trois avantages par rapport au filtrage basé sur le contenu [Herlocker, *et al.* 1999] :

1. Comme le *FC* ne requiert que l'identificateur des objets, il peut donc supporter les objets dont le contenu n'est pas facilement analysable par des processus automatisés. En conséquence, le filtrage peut être effectué sur pratiquement tout type d'objets, tels que des films, de la musique [Chen, *et al.* 2001] ou même des blagues [Goldberg, *et al.* 2001];
2. la capacité de filtrer des objets en se basant sur la qualité et le goût. En effet, dans le filtrage collaboratif, les utilisateurs déterminent la pertinence, la qualité et l'intérêt des articles, des tâches difficilement réalisables par l'ordinateur. Comme les usagers fournissent eux-mêmes une cote aux objets, ils auront tendance à donner une cote élevée aux articles pertinents et de bonnes qualités;
3. la capacité de recommander des objets à l'usager dont l'utilité était imprévisible. Cette caractéristique est décrite en anglais en tant que « *serendipitous recommendations* ».

Généralement, la tâche principale du filtrage collaboratif consiste à prédire, à partir de la matrice des utilisateurs, la cote d'un usager pour un objet encore non évalué. Plus la cote prédite est élevée, plus le système aura tendance à recommander l'article. Bresse *et al.* [Breese, *et al.* 1998] ont classé les algorithmes du *FC* en deux catégories : les algorithmes *basés sur les utilisateurs* (« *user-based* » ou « m*emory-based* ») et les algorithmes *basés*

---

[2] Le terme purement collaborative a été employé par *Balabanovic et Shoham* [Balabanovic, *et al.* 1997] pour designer les systèmes qui emploient uniquement l'approche collaborative, à la difference des systèmes hybrides qui utilisent en même temps l'approche du *FC* et du *FBC*. Les systèmes hybrides sont discutés plus tard dans ce chapitre.

sur un modèle (« m*odel-based* »). La première classe « opère sur la totalité de la base de données des utilisateurs pour faire leurs prédictions » [Breese, *et al.* 1998] alors que la deuxième analyse cette base de données « pour estimer ou apprendre un modèle, qui sera ensuite utilisé pour les prédictions » [Breese, *et al.* 1998].

2.2.2.3 Les algorithmes basés sur les utilisateurs

Les algorithmes basés sur les utilisateurs essaient d'identifier les participants ayant coté, de façon similaire, les mêmes éléments. Les prédictions des cotes de l'utilisateur actif sont ensuite calculées à l'aide de leurs cotations.

Nous savons que la matrice des utilisateurs est constituée de l'ensemble des votes $v_{i,j}$, représentant la cote que l'usager i a donnée à l'objet j. Soit I_i l'ensemble des votes de l'utilisateur i et $\bar{v}_i$ leur moyenne. Pour prédire la cote $P_{a,j}$ de l'utilisateur a à un objet j encore non évalué, on utilise la somme,

$$P_{a,j} = \bar{v}_a + \kappa \sum_{i=1}^{n} w(a,i)(v_{i,j} - \bar{v}_i) \tag{2.2-4}$$

où n est le nombre total des utilisateurs ayant un poids différent de zéro, κ est un facteur de normalisation tel que la somme des valeurs absolues des poids sera égale à 1. Le poids $w(a,i)$ représente la similarité entre les deux participants et peut être estimé de plusieurs façons, notamment en calculant la *similarité vectorielle* ou la *corrélation* entre l'usager a et i [Breese *et al.*, 1998].

La formule de similarité vectorielle considère les cotes des utilisateurs a et i comme des vecteurs et calcule leur cosinus afin de mesurer leur similarité vectorielle. Pour estimer la corrélation entre deux personnes, nous utilisons la formule de *corrélation de Pearson* décrite comme suit,

$$w(a,i) = \frac{\sum_j (v_{a,j} - \bar{v}_a)(v_{i,j} - \bar{v}_i)}{\sigma_a \times \sigma_i} \qquad (2.2\text{-}5)$$

avec $\sigma_i = \sqrt{\sum_j (v_{i,j} - \bar{v}_i)^2}$, la déviation standard des votes de l'utilisateur *i*.

Ainsi plus les cotes de deux usagers sont similaires, plus $w(a,i)$ tendra vers 1. À l'inverse, plus ils sont différents, plus $w(a,i)$ se rapprochera de -1.

Exemple de prédiction

Prenons le cas ou nous cherchons à recommander des éléments à Ève, il s'agit donc de trouver les objets qui l'intéresseront le plus. Il faut donc prédire ses cotes pour les objets encore non évalués. Le tableau 2.2-1 schématise la problématique.

Tableau 2.2-1. Exemple : la matrice des cotes utilisée pour les estimations.

	Objet A	Objet B	Objet C	Objet D	Objet E
Alice	*1*	*5*	*4*	Ø	*2*
Bob	*3*	*2*	Ø	*5*	*5*
Charlie	*4*	*3*	Ø	*5*	*4*
Dana	Ø	Ø	*3*	Ø	*1*
Ève	*4*	*3*	**?**	*5*	**?**

*L'ensemble vide Ø désigne les éléments encore non évalués par l'usager

Calculons l'estimation pour l'objet E. $\bar{v}_i$, la moyenne des cotes de chaque usager *i* est

$$\bar{v}_{Alice} = \frac{12}{4} = 3 \qquad \bar{v}_{Bob} = \frac{15}{4} = 3.75$$

$$\bar{v}_{Charlie} = \frac{16}{4} = 4 \qquad \bar{v}_{Dana} = \frac{4}{2} = 2 \qquad \bar{v}_{Ève} = \frac{12}{3} = 4$$

Ensuite, nous utiliserons la corrélation de Pearson (équation (2.2-5)) pour calculer le poids $w(a,i)$ entre Ève et les autres utilisateurs du système.

$$w(\grave{E}ve, Alice) = \frac{-2}{\sqrt{2 \times 10}} = -0.447 \qquad w(\grave{E}ve, Bob) = \frac{3}{\sqrt{2 \times 6.75}} = 0.816$$

$$w(\grave{E}ve, Charlie) = \frac{2}{\sqrt{2 \times 2}} = 1$$

D'après ces résultats, Ève a plus de corrélation avec Charlie et Bob (1 et 0.816 respectivement) qu'avec Alice (une corrélation de -0.447). D'ailleurs, en observant les votes des usagers, nous remarquons que Ève, Charlie et Bob partagent les mêmes préférences. Il faut noter que le poids entre Ève et Dana ne peut être calculé, vu qu'elles ne partagent aucune cotation d'un même objet.

Finalement, pour estimer le vote de Ève pour l'objet E, il faut appliquer l'équation (2.2-4) :

$$P_{\grave{E}ve,objet\ E} = 4 + 0.441 \times ((-0.447 \times -1) + (0.816 \times 1.25) + (1 \times 0)) = 4{,}648$$

L'estimation du vote de Ève est de 4,648 (en réalité, Ève ne peut donner que des cotes ayant une valeur entière, dans ce cas elle serait de 5). Ce résultat suit bien la logique du filtrage collaboratif. En effet, la cote estimée pour l'objet E ressemble plus à celles de Bob et de Charlie (avec qui Ève partage une forte corrélation) qu'à celle d'Alice.

Le même calcul est effectué pour l'objet C et l'objet qui a la cote estimée la plus élevée est celui qui correspond le plus aux préférences d'Ève et doit lui être recommandé.

Les limites des algorithmes basés sur les utilisateurs

Les systèmes de filtrage collaboratif basé sur les utilisateurs présentent deux faiblesses majeures [Sarwar, *et al.* 2001] :

1. L'espacement : dans la pratique, les utilisateurs ne cotent que quelques objets. Or, pour calculer la similarité entre deux usagers, cette approche requiert qu'ils aient coté des objets en commun, ce qui n'est souvent pas le cas. En conséquence, il ne serait pas possible de calculer la similitude de plusieurs participants (c'était le cas

de Ève et de Dana dans l'exemple précédent). L'exactitude des recommandations peut en être appauvrie, surtout quand le nombre d'objets est beaucoup plus grand que le nombre d'utilisateurs [Huang, *et al.* 2004].

2. Le passage à l'échelle : dans ces systèmes, la complexité du temps d'exécution augmente considérablement avec le nombre de participants et la quantité des objets [Linden, *et al.* 2003]. Avec des millions d'utilisateurs et d'objets, les sites Web qui utilisent le *FC* (comme amazon.com), souffriraient d'un problème sérieux de performance, d'autant plus que le calcul se fait souvent en temps réel.

2.2.2.4 Les algorithmes basés sur un modèle

Pour adresser ces défis, les chercheurs ont employé des techniques basées sur un modèle. Dans cette approche, un modèle (par exemple, formation de voisinage) est construit hors-ligne et est ensuite utilisé en-ligne, pour la génération des prévisions. Typiquement, le modèle est cher (en temps) à construire mais rapide à exécuter [Breese, *et al.* 1998]. Le modèle peut être déduit à l'aide de plusieurs approches, comme *le modèle des clusters*, la méthode *horting* et *l'approche basée sur les éléments*.

Le modèle des clusters

Cette technique consiste à identifier hors-ligne, les groupes d'utilisateurs qui semblent avoir des préférences communes. Une fois que les clusters sont créés, des prévisions pour un utilisateur individuel peuvent être faites en faisant la moyenne des cotes des autres utilisateurs dans ce cluster. Ainsi, ce modèle traite le filtrage collaboratif comme un problème de classification. Il demande l'estimation d'un certain nombre de paramètres, tels que la probabilité qu'un utilisateur particulier fasse partie de la classe C et la probabilité conditionnelle des cotations étant donné l'adhésion de l'utilisateur à une classe.

Les techniques des clusters produisent habituellement des recommandations moins personnelles que les autres méthodes, et dans certains cas, elles sont moins exactes que les algorithmes basés sur les utilisateurs [Breese, *et al.* 1998].

L'approche horting

Horting [Aggarwal, *et al.* 1999] est une technique basée sur un graphe dans lequel les nœuds représentent les utilisateurs et les arcs entre deux nœuds indiquent le degré de similitude des utilisateurs. Des prévisions sont produites en parcourant le graphe et en combinant les cotes des utilisateurs voisins. Horting diffère des algorithmes basés sur les utilisateurs par le fait que les relations transitives de voisinage peuvent être explorées même si les utilisateurs n'ont pas coté les mêmes éléments.

Les algorithmes basés sur les éléments

Les algorithmes basés sur les éléments analysent d'abord la matrice des cotes pour trouver les relations entre les différents éléments et emploient ensuite ces relations pour calculer les recommandations [Linden, *et al.* 2003], [Sarwar, *et al.* 2001]. Cette approche comporte essentiellement deux tâches critiques :

1. Le calcul de la similitude des éléments.
2. La prédiction des cotes des éléments encore non évalués, en se basant sur ces similitudes.

Calcul de la similitude des éléments

L'approche basée sur les éléments calcule la similarité des éléments par rapport aux votes reçus. En d'autres mots, cette approche mesure la ressemblance de leurs cotes. Ainsi, pour calculer la similarité $sim(i, j)$ entre deux éléments i et j, nous devons comparer l'ensemble des votes de i à ceux de j. Pour cela, nous tenons compte uniquement des participants $u \subset U_{i,j}$ qui ont coté les deux éléments i et j. La figure 2.2-2 schématise une matrice des cotes et identifie l'ensemble $U_{i,j}$ dans le cas où i = *l'objet D* et j = *l'objet E*. Nous obtenons alors $U_{ObjetD,ObjetE} = \{Bob, Carl, Rhéa\}$.

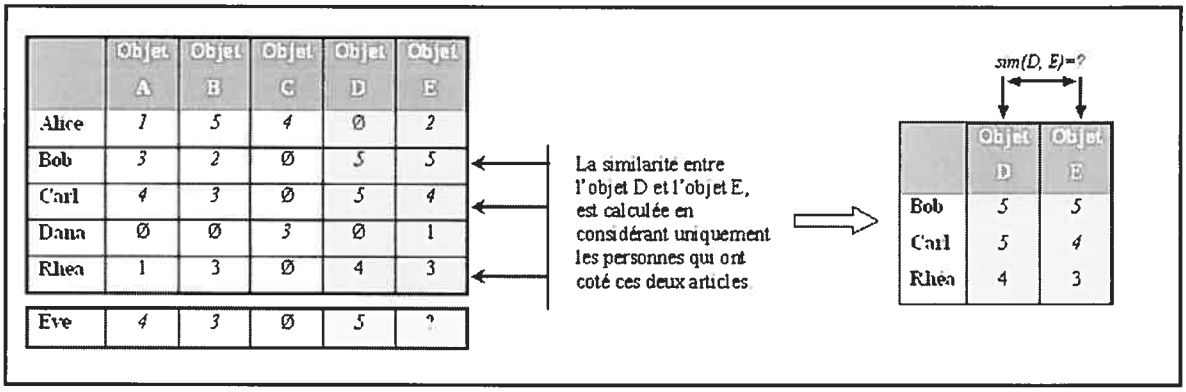


Figure 2.2-2. Approche « *item-based* » : les utilisateurs à considérer lors du calcul de la similitude entre les objets D et E

Une fois que l'ensemble $U_{i,j}$ est trouvé, la similarité entre les éléments peut être calculée de plusieurs façons. Nous présentons trois d'entre-elles : la *similitude basée sur le cosinus*, la *similitude ajustée basée sur le cosinus* et la *corrélation de Pearson-r*.

Similitude basée sur le cosinus

Cette méthode considère les deux objets en tant que deux vecteurs de dimension $|U_{i,j}|$, où chaque vecteur est formé par les cotes des utilisateurs. Dans la figure 2.2-2, les colonnes du tableau représentent chacun des vecteurs. Nous avons alors $\vec{D} = |5 \quad 5 \quad 4|$ et $\vec{E} = |5 \quad 4 \quad 3|$. La similitude entre deux éléments est mesurée en calculant le cosinus de l'angle formé par ces deux vecteurs [Sarwar, *et al.* 2001].

Similitude ajustée basée sur le cosinus

Souvent les utilisateurs ne cotent pas les objets de la même manière. Si l'on prend l'exemple d'un utilisateur qui est plutôt « strict », il peut avoir tendance à attribuer aux éléments des cotes relativement basses par rapport à celles données par un usager beaucoup plus « souple ». Par conséquent, la similitude basée sur le cosinus présente un inconvénient important : les différences dans l'échelle d'évaluation entre les différents utilisateurs ne sont pas prises en considération. La similitude ajustée basée sur le cosinus compense cet

inconvénient en soustrayant la moyenne des cotes de l'utilisateur de chacune de ses propres cotes [Sarwar, *et al.* 2001]. Formellement, la similitude ajustée basée sur le cosinus est donnée par

$$sim(i,j) = \frac{\sum_{u \in U_{i,j}} (R_{u,i} - \overline{R}_u)(R_{u,j} - \overline{R}_u)}{\sqrt{\sum_{u \in U_{i,j}} (R_{u,i} - \overline{R}_u)^2} \sqrt{\sum_{u \in U_{i,j}} (R_{u,j} - \overline{R}_u)^2}} \tag{2.2-6}$$

avec $R_{u,i}$ la cote attribuée à l'élément i par l'usager u
et $\overline{R}_u$ la moyenne des cotes de l'usager u

La corrélation de Pearson-r

Dans ce cas-ci, la similitude entre deux éléments i et j est mesurée en calculant la corrélation de Pearson-r [Sarwar, *et al.* 2001] telle que

$$sim(i,j) = \frac{\sum_{u \in U_{i,j}} (R_{u,i} - \overline{R}_i)(R_{u,j} - \overline{R}_j)}{\sqrt{\sum_{u \in U_{i,j}} (R_{u,i} - \overline{R}_i)^2} \sqrt{\sum_{u \in U_{i,j}} (R_{u,j} - \overline{R}_j)^2}} \tag{2.2-7}$$

avec $R_{u,i}$ la cote attribuée à l'élément i par l'usager u
et $\overline{R}_i$ la moyenne des cotes de l'élément i

Exemple de calcul de la similitude

Calculons la similarité entre l'élément E et les autres éléments de la matrice des cotes de la figure 2.2-2, en utilisant la corrélation de Pearson-r (équation (2.2-7)).

$$U_{objet\ A,\ objet\ E} = \{Alice, Bob, Carl, Rhéa\} \qquad U_{objet\ B,\ objet\ E} = \{Alice, Bob, Carl, Rhéa\}$$
$$sim(objet\ A,\ objet\ E) = \frac{4.5}{5.809} = 0.775 \qquad sim(objet\ B,\ objet\ E) = \frac{-4.5}{4.873} = -0.923$$
$$U_{objet\ C,\ objet\ E} = \{Alice, Dana\} \qquad U_{objet\ D,\ objet\ E} = \{Bob, Carl, Rhéa\}$$
$$sim(objet\ C,\ objet\ E) = \frac{0.5}{0.5} = 1 \qquad sim(objet\ D,\ objet\ E) = \frac{1}{1.157} = 0.866$$

D'après l'équation (2.2-7), l'objet le plus similaire à E est l'élément C, suivi de l'élément D. L'élément B est le moins similaire.

En pratique, toutes les relations entre les paires d'objets doivent être calculées hors-ligne et stockées dans la matrice des relations comme le montre la figure 2.2-3. Cette matrice est ensuite utilisée pour le calcul des prédictions.

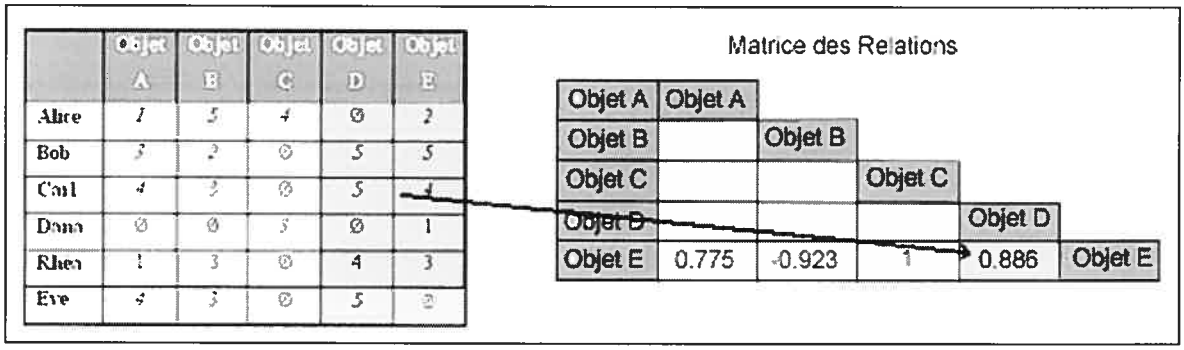


Figure 2.2-3. La génération de la matrice des relations

La prédiction du vote

Une fois que les objets les plus semblables sont identifiés, la prévision $P_{a,j}$ du vote de l'utilisateur a sur l'objet j est calculée en prenant la moyenne pondérée des votes de l'utilisateur a. Ainsi

$$P_{a,j} = \frac{\sum_N (sim(i,N) \times R_{a,N})}{\sum_N (|sim(i,N)|)} \tag{2.2-8}$$

Des expériences suggèrent que les algorithmes basés sur les objets fournissent un temps d'exécution nettement meilleur que des algorithmes basés sur les utilisateurs tout en même ayant des recommandations de qualité comparable [Sarwar, *et al.* 2001].

2.2.2.5 Les limites du filtrage collaboratif

Bien que le filtrage collaboratif résolve plusieurs problèmes reliés au filtrage basé sur le contenu, elle en introduit d'autres :

- le problème du nouvel objet : lorsqu'un objet est introduit dans le système, il n'est pas possible d'en faire la recommandation tant qu'un usager ne l'a pas évalué. Or, si

un élément n'est pas recommandé, il a moins de chance d'être coté par les utilisateurs. Ainsi, le filtrage ne couvrira pas tous les articles et par conséquent des faux négatifs sont présentés [Shahabi 2003];

- le *démarrage à froid* ou « *cold start* » : lors d'un premier déploiement d'un système de *FC*, peu de cotations sont disponibles. Même si l'usager fait l'effort de coter plusieurs objets, le système ne peut lui générer une recommandation car ce processus dépend des cotations des autres utilisateurs;
- s'il existe un usager qui a des préférences non "usuelles" comparées au reste de la population, il aura des recommandations assez pauvres puisqu'il serait difficile de trouver les utilisateurs voisins.

2.2.3 L'approche hybride

Le *FBC* et le *FC* ont chacun leurs avantages et inconvénients. Cependant, il ne faut pas considérer ces deux techniques comme concurrentes l'une à l'autre, mais plutôt comme complémentaires. Plusieurs systèmes de recommandation utilisent une approche *hybride* en combinant deux ou plusieurs approches afin de générer les recommandations. Dans [Burke 2002], une étude approfondie des systèmes hybrides a été effectué. Souvent, on combine le *FC* et le *FBC* dans le but de bénéficier des avantages de chacun d'eux tout en éliminant la majeure partie de leurs inconvénients [Balabanovic, *et al.* 1997].

Il existe différentes façons de combiner les méthodes de collaboration avec celles basées sur le contenu. Une approche consiste à mettre en application des systèmes de *FC* et de *FBC* de façon séparée. Ensuite, les sorties obtenues à partir de chaque système sont réunies dans une recommandation finale en utilisant une combinaison linéaire, où un poids est associé à chacune d'elles [Claypool, *et al.* 1999].

Beaucoup de systèmes hybrides, comme *Fab* [Balabanovic, *et al.* 1997] et la « *collaboration par l'intermédiaire du contenu* » (« *collaboration via content* »), décrite dans [Pazzani 1999], construisent les profils des utilisateurs en se basant sur l'analyse du contenu et comparent les profils résultants pour déterminer les utilisateurs semblables. Les

recommandations s'appuient sur la similarité entre le contenu des objets et le profil de l'utilisateur, tout comme sur un avis favorable des utilisateurs qui ont un profil semblable. Des études effectuées dans [Pazzani 1999] et [Balabanovic, *et al.* 1997] démontrent que les méthodes hybrides peuvent fournir des recommandations plus précises que les approches collaboratives ou celles basées sur le contenu lorsqu'elles sont appliquées séparément.

2.3 Conclusion

Ce chapitre a montré un aperçu des techniques de filtrage existantes et étudiées dans la littérature. Les systèmes de filtrage ont comme objectif la réduction du flot d'information présenté à l'usager, en ne lui présentant que ceux qui correspondent le plus à son profil. Ils sont donc utilisés face au problème de la surcharge d'informations, puisqu'ils permettent le tri des données selon les goûts, les préférences et les champs d'intérêt de chaque individu.

Il faut remarquer qu'une multitude de techniques existent, mais il serait impossible de toutes les présenter dans le cadre de ce mémoire. Nous avons décidé de discuter des deux les techniques les plus prédominantes dans la littérature : le *filtrage basé sur le contenu* et le *filtrage collaboratif*. Nous pouvons dire que chacune de ces techniques a ses avantages et inconvénients. Ces techniques sont considérées comme complémentaires. En effet, la combinaison du filtrage basé sur le contenu et du filtrage collaboratif en une approche hybride nous permet de bénéficier de leurs avantages tout en éliminant la majeure partie de leurs inconvénients, comme le suggère les études que nous avons présentées dans ce chapitre.

Chapitre 3
La recommandation de livres : les enjeux

Après avoir abordé certaines techniques de filtrage, ce chapitre se concentre plus spécifiquement sur la recommandation de livres. Dans cette ère de l'information, de nouveaux titres sont mis sur le marché quotidiennement. Étant donné le nombre croissant de livres, certains lecteurs et plus précisément certains étudiants se demandent comment faire face à cet envahissement. Le filtrage automatisé des livres selon les champs d'intérêt, les goûts et les préférences des apprenants constitue une des nombreuses solutions. Nous commençons par présenter quelques systèmes de recommandation de livres déjà implémentés et étudiés dans la littérature. Ensuite, nous analysons ces systèmes sous plusieurs angles, dans le but d'évaluer leur adaptabilité à un environnement pédagogique. Finalement, nous proposons un Système de Recommandation (*SR*) de livres conçu spécifiquement pour le milieu éducatif.

3.1 Quelques systèmes de recommandation de livres

Nous décrirons trois systèmes dédiés à la recommandation de livres : LIBRA [Mooney, *et al.* 2000], TwinFinder [Hirooka, *et al.* 2000] et le *SR* de amazon.com [Linden, *et al.* 2003]. Notre choix s'est porté sur ces systèmes afin d'offrir un éventail, vu la différence de leurs

approches et du domaine dans lequel ils sont employés. En effet, TwinFinder et le *SR* de amazon.com ont été développés dans le cadre d'un site Web commercial, mais chacun d'eux génère les recommandations selon des approches bien différentes. Le premier emploie celle du filtrage basé sur le contenu, alors que le deuxième applique le filtrage collaboratif. LIBRA quant à lui, utilise l'approche basée sur le contenu, mais il n'a pas été conçu pour un environnement spécifique.

3.1.1 LIBRA

LIBRA (*Learning Intelligent Book Recommending Agent*) [Mooney, *et al.* 2000] est un système de recommandation de livre basé sur le contenu. Il emploie des techniques d'extraction de l'information et un algorithme d'apprentissage machine pour la catégorisation des textes semi-structurés.

3.1.1.1 La représentation des livres

Ce système construit sa base de livres à partir des pages Web de amazon.com. Ainsi, une recherche de sujet sur amazone.com est exécutée pour obtenir la liste des pages Web réservées aux livres du domaine. Quatre catégories ont été couvertes : la fiction littéraire (3 061 titres), la science-fiction (3 813 titres), le mystère (7 285 titres) et la science (6 177 titres). LIBRA télécharge chacune des pages trouvées et emploie un système d'extraction pour retirer les données pertinentes de chaque livre. Ces données sont réparties comme suit : le titre, les auteurs, la synthèse du livre, les critiques officielles, les critiques des clients, les auteurs associés, les titres associés et les termes des sujets.

Les textes extraits sont ensuite transformés en vecteurs de termes; le titre et les auteurs sont ajoutés à la liste des « titres associés » et à celle des « auteurs associés » respectivement. Ensuite, chacune de ces listes est représentée par un vecteur unique alors que la synthèse du livre, les critiques officielles et les critiques des clients sont groupées dans un même vecteur appelé « description ».

3.1.1.2 Le profil de l'utilisateur

Lors de sa première connexion au système, l'utilisateur choisit et évalue un ensemble de livres qui servira d'exemple à l'algorithme de classification, comme le montre la figure 3.1-1. En recherchant des auteurs ou des titres particuliers, l'utilisateur peut éviter de balayer la base de données entière ou d'évaluer des sélections choisies au hasard. Le participant est invité à fournir une cotation entre 1 et 10, dans le but d'indiquer son intérêt pour chaque titre.

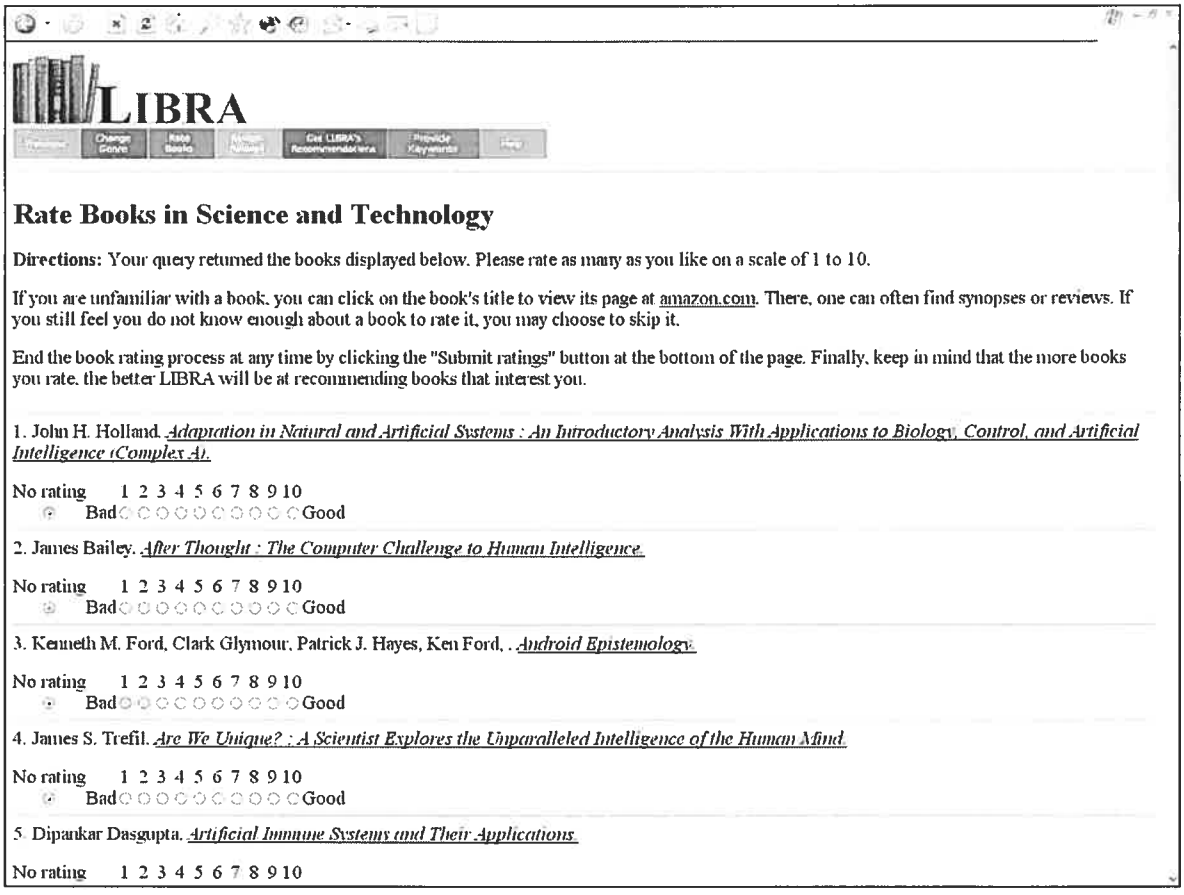


Figure 3.1-1. LIBRA[3] : L'évaluation d'un ensemble de livres générés aléatoirement lors de la première connexion de l'usager

[3] http://titan.cs.utexas.edu:8090/libra/index.jsp

3.1.1.3 Le calcul des recommandations

LIBRA emploie un classificateur bayésien (section 2.2.1.3) étendu pour manipuler plusieurs vecteurs plutôt qu'un seul vecteur simple. Ainsi, LIBRA n'essaye pas de prédire la cote numérique exacte d'un titre, mais plutôt le rang des titres par ordre de préférence. Cette tâche est alors remaniée comme problème binaire probabiliste de catégorisation en vue de calculer la probabilité qu'un livre soit évalué positivement plutôt que négativement. Une cote est considérée positive lorsqu'elle est répartie entre 6 et 10, et négative entre 1 et 5.

3.1.1.4 La mise à jour du profil

Après avoir passé en revue les recommandations, l'utilisateur peut affecter ses propres cotations aux titres proposés pour indiquer si les livres ont été correctement ordonnés. Ce cycle peut être répété plusieurs fois pour permettre au système de produire de meilleurs résultats. La figure 3.1-2 montre la liste des livres suggérés à l'utilisateur. Le participant a la possibilité de fournir un retour au système en évaluant la cohérence des livres proposés. Lors de l'estimation des prochaines recommandations, le système prendra alors en considération ces nouvelles évaluations.

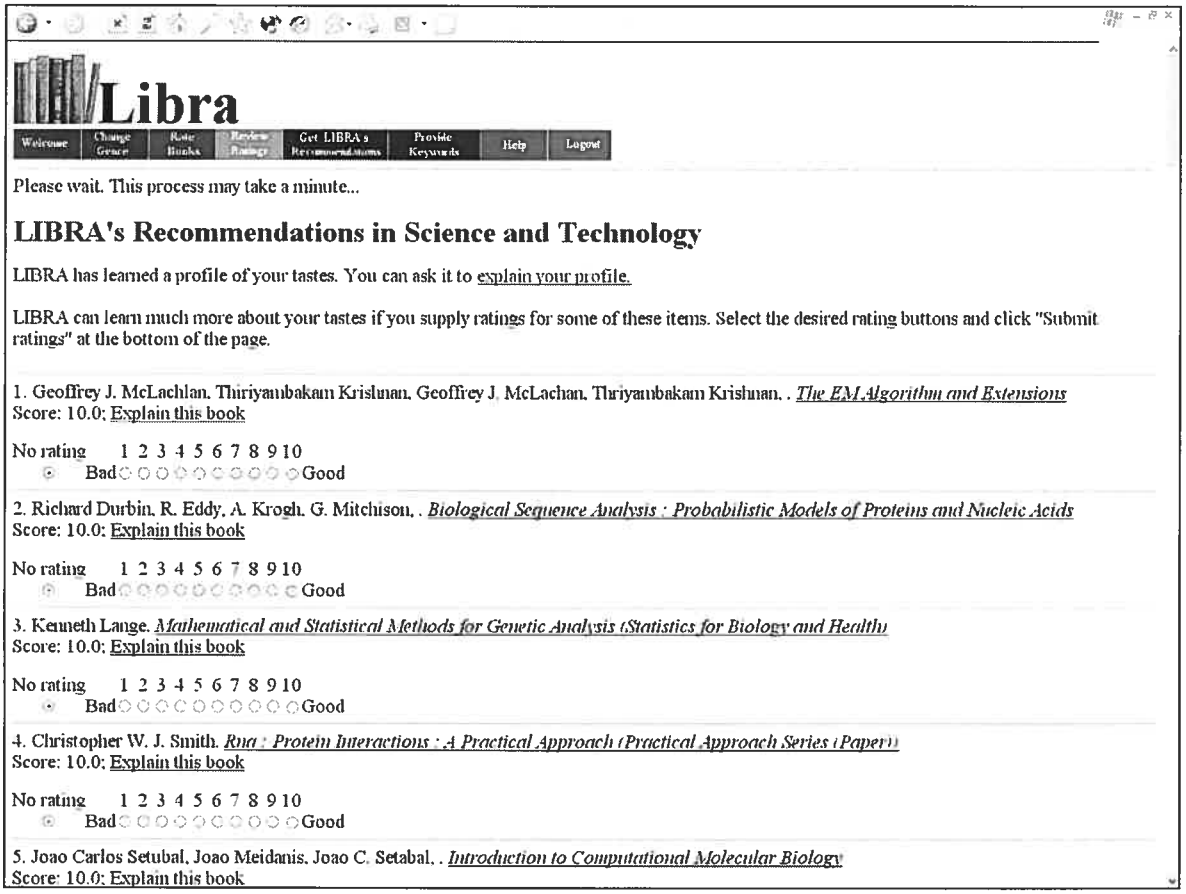


Figure 3.1-2. LIBRA[4]: Génération des recommandations avec une possibilité de retour

3.1.2 TwinFinder

TwinFinder [Hirooka, *et al.* 2000] est un système de recommandation employé par Skysoft [5], une librairie électronique spécialisée dans la vente de livres au Japon. Il utilise l'approche basée sur le contenu en appliquant le modèle vectoriel.

[4] http://titan.cs.utexas.edu:8090/libra/index.jsp

[5] www.skysoft.co.jp

3.1.2.1 La représentation des livres

Skysoft classifie ses livres en 49 catégories telles que Arts, Affaires et Économies, et Ordinateurs en suivant le modèle BISAC [6]. Chacune de ces catégories comporte des sous-catégories plus spécifiques. Pour chacun des livres, TwinFinder utilise le titre, les auteurs, la synthèse et la sous-catégorie dans laquelle il est classifié pour générer un vecteur de termes. Les poids des termes sont calculés à l'aide du *TF-IDF* (voir 2.2.1.2).

3.1.2.2 Le profil de l'utilisateur

TwinFinder présente le profil des clients sous une forme vectorielle. En effet, lorsqu'un client achète un livre, le vecteur des termes de ce livre est ajouté au profil. Cependant, le système considère que les intérêts d'un utilisateur varient d'une catégorie de livres à une autre. Par conséquent, au lieu de représenter le profil par un seul vecteur décrivant tous les intérêts de l'acheteur, TwinFinder construit un vecteur pour chaque catégorie de livres.

3.1.2.3 Le calcul des recommandations

Comme les intérêts des clients sont exprimés sous forme de vecteur, TwinFinder calcule la similarité vectorielle entre le profil de l'utilisateur et le vecteur des termes d'un livre. Une fois la similarité de tous les livres calculée, le système proposera les titres qui partagent la plus grande similitude avec le profil.

Le fait de catégoriser les intérêts des clients permet à TwinFinder d'offrir deux types de recommandations : les recommandations ordonnées « *Order-Matching Method* » (*OMM*) et les recommandations croisées « *Cross-Matching Method* » (*CMM*) [Hirooka, *et al.* 2000].

---

[6] Un standard de catégorisation de livres, voir www.bisg.org/bisac/

3.1.2.4 Order-Matching Method

Cette méthode compare le vecteur de termes d'une catégorie du profil du client avec les livres de cette même catégorie. La figure 3.1-3 illustre ce processus de comparaison. Ainsi, OMM évite de recommander les livres qui partagent quelques termes avec le profil de l'utilisateur, mais qui appartiennent à une catégorie qui ne l'intéresse pas. Par exemple, supposons qu'un client a acheté un livre de recettes de gâteaux et de biscuits, dits en anglais « *cake* » et « *cookies* ». *OMM* utilisera les termes « *cake* » et « *cookies* » dans le but de trouver des livres similaires. Mais comme ce livre appartient évidemment à la catégorie cuisine, TwinFinder ne calculera les similarités que pour les livres qui appartiennent à cette même catégorie. Ceci empêchera le système de proposer des livres d'ordinateurs qui parlent de « cookies », une technique de la programmation Web [Hirooka, *et al.* 2000], pour lesquelles l'utilisateur n'a évidemment aucun intérêt.

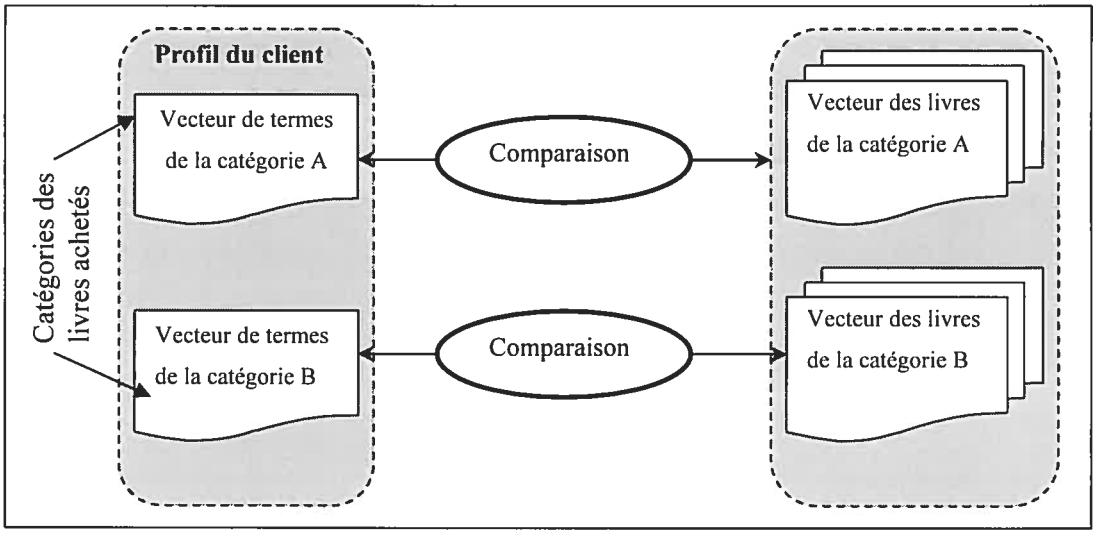


Figure 3.1-3. La comparaison des vecteurs des termes pour les recommandations ordonnées [Hirooka, *et al.* 2000]

3.1.2.5 Cross-Matching Method

Pour faire face à la « surspécialisation », le *CMM* compare le vecteur des termes d'une catégorie aux livres d'une autre catégorie, mais pour laquelle l'utilisateur porte aussi un intérêt, comme le montre la figure 3.1-4.

Cette pratique permet d'exposer de nouveaux intérêts ou tout simplement d'augmenter l'intérêt des clients pour certains livres. Comme l'expliquent les auteurs, un utilisateur qui a acheté des livres de Star Trek (appartenant à la catégorie science-fiction) et des livres de recettes, s'est vu recommander le livre « Star Trek Cookbook ». Ainsi, *CMM* croise deux intérêts du client pour proposer les livres qui les englobent.

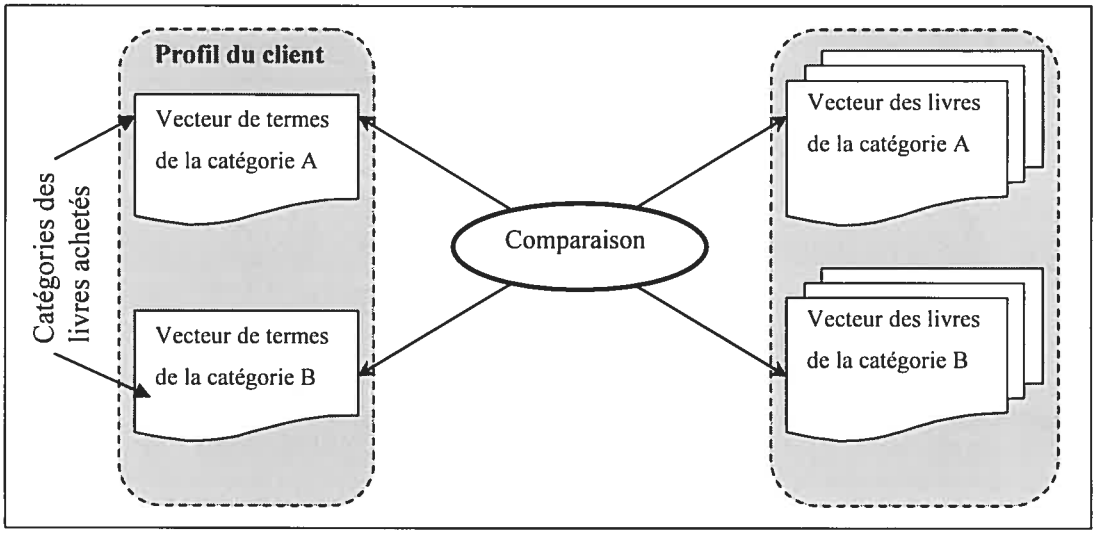


Figure 3.1-4. La comparaison des vecteurs des termes pour les recommandations croisées [Hirooka, *et al.* 2000]

3.1.3 Amazon.com

La librairie en ligne amazon.com est devenue une référence en matière de recommandation de produits. Amazon.com réserve une page Web pour chacun de ses livres. Sur cette page, on retrouve une liste de recommandations basée sur les œuvres fréquemment achetées par les internautes qui se sont procuré le livre en question, comme le montre la figure 3.1-5.

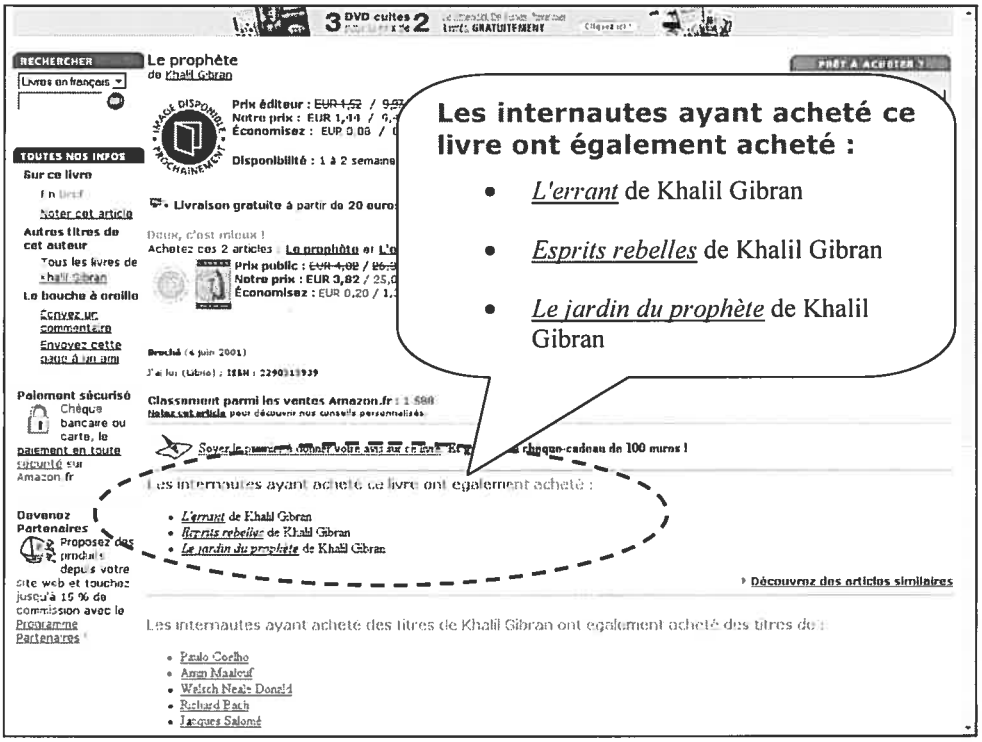


Figure 3.1-5. Amazon.fr : Recommandation sur la page Web d'un livre

Plus récemment, amazon.com a réservé une section spécifique sur son site Web dédiée à la recommandation de produits. Par un clic sur le lien approprié, le client est transporté directement à sa page de recommandations. La figure 3.1-6 est une copie d'écran d'une telle page. Amazon utilise le filtrage collaboratif basé sur les objets (cette technique a été décrite à la section 2.2.2.4) et apparie chacun des livres acheté ou coté par l'utilisateur aux objets semblables pour les combiner ensuite dans cette liste de recommandations. La figure 3.1-7 est une page qui explique au client les raisons pour lesquelles le livre « *Java Development With Ant* » lui a été recommandé.

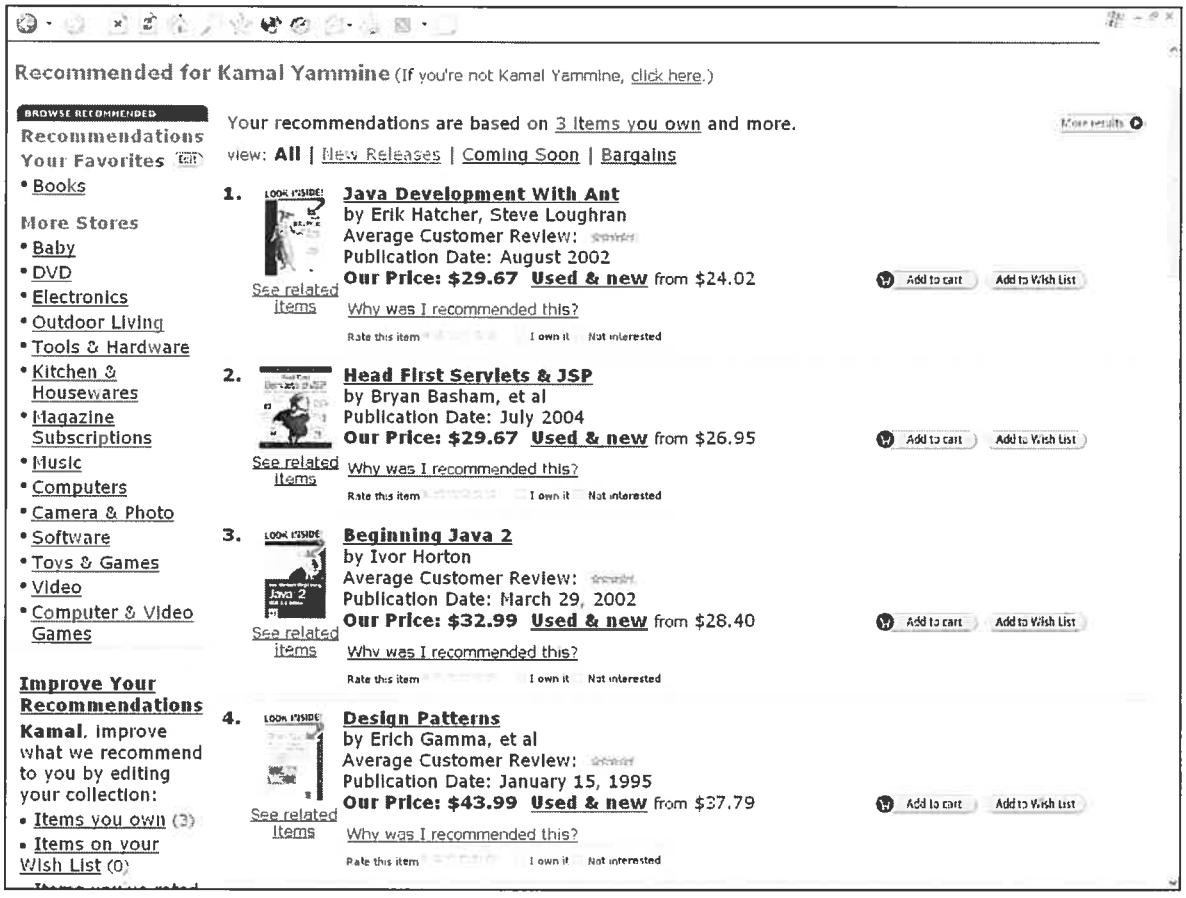


Figure 3.1-6. Amazon.com : exemple de recommandations générées pour un client en utilisant la méthode basée sur les éléments

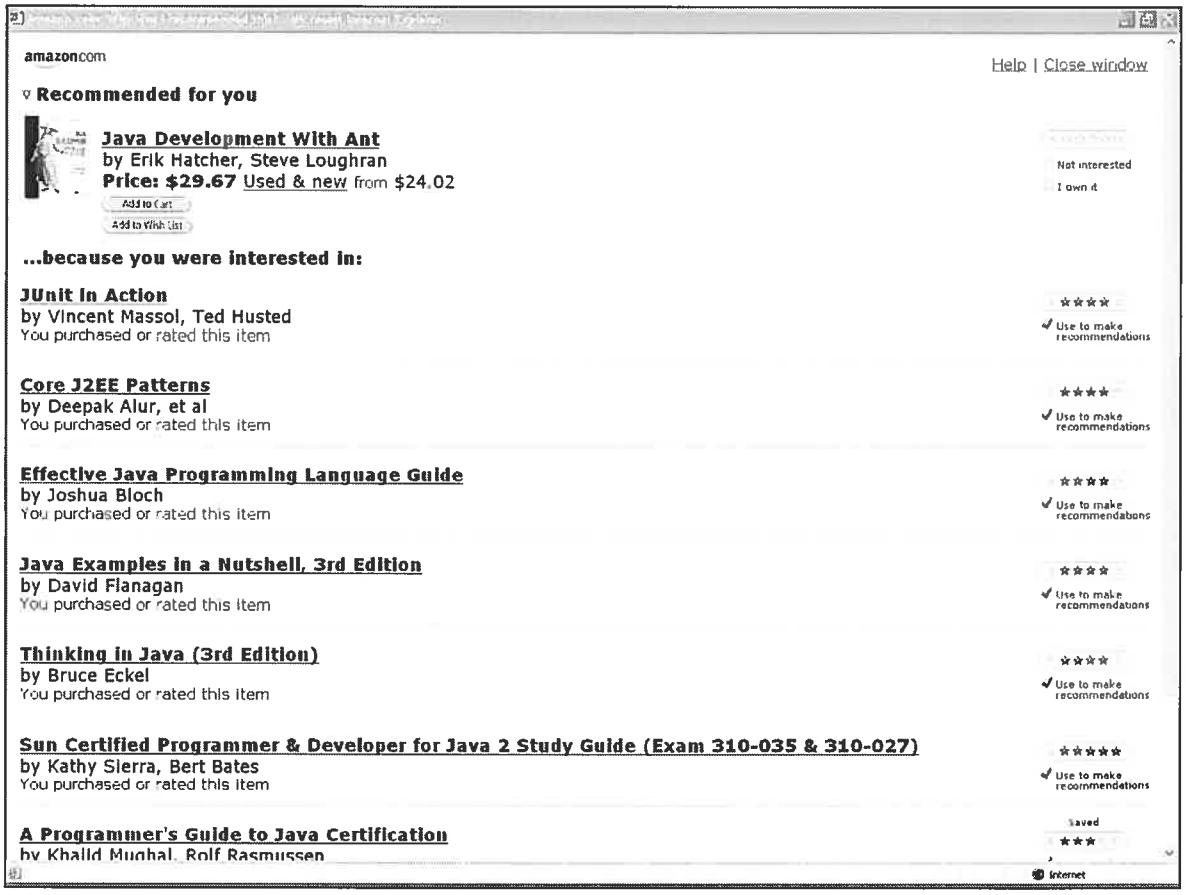


Figure 3.1-7. Amazon.com : l'explication des recommandations au client

Pour déterminer les livres les plus similaires, l'algorithme construit une matrice des objets que l'ensemble des clients tend à acheter ensemble ou à coter de la même manière. Le site Web établit cette matrice en utilisant l'algorithme itératif suivant [Linden, *et al.* 2003] :

```
for each item in product catalog, I1
   for each customer C who purchased I1
      for each item I2 purchased by customer C
         record that a customer purchased I1 and I2
   for each item I2
      compute the similarity between I1 and I2
```

Bien qu'il soit possible de calculer la similitude entre deux livres de diverses manières, [Linden, *et al.* 2003] sont restés un peu flous sur la manière dont Amazon procède. Ils ont cependant décrit brièvement une méthode assez commune, celle de calculer le cosinus vectoriel dans la mesure où un vecteur correspond à un livre et la dimension M du vecteur correspond au nombre de clients qui ont acheté ce livre (voir section 2.2.2.4).

3.2 La recommandation de livres dans un environnement pédagogique

Les systèmes de recommandation ont souvent été étudiés dans le contexte du commerce électronique [Bin, *et al.* 2003], [Hirooka, *et al.* 2000], [Sarwar, *et al.* 2000] ou dans des contextes non spécifiques à l'apprentissage. Mais depuis peu, des recherches ont été effectuées sur l'application des systèmes de recommandation dans le *e-learning*. Dans [Zaïane 2002], le système guide les apprenants en leur proposant des activités en ligne basées sur leurs profils, leurs historiques d'accès et leurs navigations. Dans [Tang, *et al.* 2003], un système de recommandation a été développé dans le but de suggérer des articles selon le profil et le niveau d'expertise du participant.

La recommandation de livres peut être problématique. En effet, les livres sont convoités par une multitude de clientèles ayant des besoins variés. Des personnes s'y intéressent pour le plaisir de lire, d'autres pour approfondir leur culture générale alors que certains, tels que les étudiants, les sollicitent dans le but de perfectionner leurs connaissances dans un cadre pédagogique bien précis. Ainsi, l'utilité des livres ne dépend pas uniquement des goûts, des besoins et des préférences des lecteurs, mais aussi de l'environnement dans lequel ils sont recherchés. Afin de mieux exprimer cette notion, prenons l'exemple de deux personnes qui s'intéressent à l'Intelligence Artificielle (*IA*). L'une d'entre elles désire approfondir ses connaissances générales sur ce domaine alors que l'autre est un étudiant qui cherche des livres pouvant l'assister pour son cours d'*IA*. Il est évident que ces deux personnes ne seront pas intéressées par les mêmes livres. En effet, la première recherche des livres qui couvrent les notions du cours (ou au moins une partie)

alors que pour la seconde, ces mêmes livres seraient peut-être trop techniques ou trop détaillés.

Vu le nombre grandissant de livres d'année en année, les systèmes de recommandation des livres seraient bénéfiques aux étudiants. Bien qu'il en existe plusieurs, à notre connaissance, **aucun d'eux n'a été conçu dans un cadre pédagogique**. Une question peut alors émerger, est-ce que ces systèmes produiront des recommandations qui répondront aux besoins particuliers des étudiants, étant donné les spécificités de l'environnement pédagogique? Pour y répondre, nous analysons dans cette section l'efficacité des systèmes en vigueur dans un tel environnement. Nous abordons ce sujet non seulement d'un point de vue technique, c'est-à-dire l'étude de l'efficacité des algorithmes et des techniques utilisées, mais aussi d'un point de vu philosophique : est-ce que l'approche actuelle de ces systèmes est idéale pour les étudiants?

3.2.1 L'aspect philosophique

L'analyse des systèmes de recommandation dans un cadre spécifique à l'enseignement requiert une certaine réflexion sur plusieurs points et nécessite bien sûr l'implication des experts de plusieurs domaines. Nous initions la discussion en faisant ressortir certains critères importants à considérer lors de la recommandation pédagogique de livres. Nous vérifions si les approches existantes sont efficaces ou idéales pour un étudiant qui recherche des livres sur un domaine précis.

3.2.1.1 Les systèmes de recommandation et le conflit d'intérêts

En commerce électronique, les *SR* sont utilisés, entre autres, « comme une technique de marketing ciblée », comme l'a déclaré *Greg Linden*, le cofondateur et chercheur du groupe de personnalisation de amazon.com [Linden, *et al.* 2003]. Ils peuvent aider à l'augmentation des ventes électroniques de trois façons [Schafer, *et al.* 2001] : changer les navigateurs en acheteurs, permettre l'augmentation des ventes croisées « *cross-sell* » et assurer une plus grande fidélité du client.

Or, peut-il exister un **conflit d'intérêts** lorsque les systèmes de recommandation sont implémentés par un site à but commercial? Cette question est légitime étant donné que c'est le site lui-même qui fournit les recommandations sur les livres qu'il désire vendre. D'ailleurs, *Zacharey Kouwe*, un journaliste qui s'est intéressé aux recommandations de livres dans un de ses articles [Kouwe 2003], a soulevé brièvement cette question dans le *Wall Street Journal*[7] en citant *Don Peterson*, le chef exécutif de *Net Perception inc.* : « De temps en temps, certains marchands utilisent la recommandation personnalisée comme un moyen de se débarrasser de la marchandise excédentaire » [Kouwe 2003]. Ce conflit d'intérêts ne peut-il pas fragiliser l'importance accordée aux besoins des étudiants par rapport aux besoins des commerçants? Il serait prudent de noter que nous soulevons cette question sans vouloir viser aucune des librairies électroniques, qui jusqu'à preuve du contraire, pratiquent la recommandation de leurs produits d'une manière convenable et professionnelle. Mais, nul ne peut contester *qu'un tel conflit porte préjudice aux intérêts des consommateurs en général et des étudiants en particulier.*

3.2.1.2 Les livres recommandés

Nous ne pouvons analyser un système de recommandation sans examiner l'ensemble des objets dont il est question. Évidemment, dans notre cas nous nous intéresserons à l'ensemble des livres couverts par les systèmes actuels. Toujours d'un point de vue subjectif de l'étudiant, nous discuterons deux critères essentiels :

1. Est-ce que la recommandation couvre un ensemble de livres suffisant et adéquat par rapport au domaine qui intéresse l'étudiant?
2. Est-ce que les livres sont facilement accessibles par l'étudiant?

Le premier point fait référence à **la pertinence des livres** présentés. L'étude de ce critère dépend, bien entendu, de chaque système de recommandation. Il existe des librairies,

---

[7] www.wsj.com

telles qu'amazon.com ou Barnes & Noble[8], qui offrent un vaste répertoire de livres sur une multitude de domaines généraux ou spécifiques. Cependant, il est évident que les systèmes de recommandation de ces librairies n'offrent aucun contrôle sur certains critères importants, par exemple, si le livre couvre la totalité ou une partie des notions d'un cours. Ce contrôle est délégué à l'étudiant lui-même.

Le second point que nous discutons est **la facilité d'accès aux livres**. En effet, quel serait l'intérêt de suggérer des livres à l'étudiant, si ces derniers ne lui sont pas facilement accessibles? Les résultats démontrent que c'est avec le temps et l'utilisation du système que nous obtenons les meilleures recommandations [Hofmann 2004], [Kai, *et al.* 2004], [Yu, *et al.* 2004], [Balabanovic, *et al.* 1997], [Konstan, *et al.* 1997]. Ainsi, pour parvenir à des recommandations de plus en plus appropriées, l'utilisateur doit interagir, plusieurs fois, avec le système. Cela requiert du participant l'achat de plusieurs livres afin de trouver ceux qui lui conviennent. Cette solution n'est pas abordable pour un étudiant. Elle entrave un certain accès aux livres recherchés.

3.2.1.3 Les besoins de l'étudiant

Comme les systèmes actuels n'ont pas été conçus dans un contexte pédagogique, le modèle de l'apprenant ne peut être pris en considération. Or, il existe des facteurs qui peuvent influencer directement la pertinence d'un livre par rapport aux besoins d'un étudiant. Par exemple, l'utilité d'un livre dépend, entre autres, de l'expertise de l'étudiant, un critère qui n'est pas considéré dans les systèmes actuels. Comme les besoins de l'étudiant ne sont pas représentés, aucune personnalisation des recommandations, selon le profil de l'étudiant, ne peut être effectuée.

---

[8] www.barnesandnoble.com

3.2.2 L'aspect pragmatique et technique

Dans le chapitre précédent, nous avons discuté de deux approches de filtrage : le *filtrage basé sur le contenu* et le *filtrage collaboratif*. Cette section discute leurs avantages et leurs inconvénients dans un contexte pédagogique.

3.2.2.1 Les avantages du filtrage basé sur le contenu pour le contexte pédagogique

Cette technique suggère les titres dont le contenu se rapproche des livres qui ont intéressé l'usager dans le passé. Contrairement à l'approche collaborative, les recommandations basées sur le contenu découlent directement des interactions de l'utilisateur avec le système. Plus l'étudiant utilise le système, plus les suggestions auront tendance à s'améliorer. Conséquemment, *les efforts de l'usager sont, en quelque sorte, récompensés à juste mesure*. Cependant, ce type de filtrage ne permet pas de contrôler certains critères abstraits tels que la qualité ou la complexité des livres. Le filtrage ne tient pas compte de ces propriétés ou caractéristiques, souvent cruciales pour parvenir à des recommandations convenables aux étudiants.

3.2.2.2 Les avantages du filtrage collaboratif pour le contexte pédagogique

Souvent, les étudiants se fient à l'avis de leurs professeurs, de leurs directeurs de recherche, des bibliothécaires ou de leurs amis pour se procurer les livres dont ils ont besoin. Combien de fois avons-nous entendu dans les couloirs universitaires ces répliques : « pour plus d'informations à ce sujet, je vous conseille de lire ce livre », « cette idée est bien expliquée dans ce livre » ou « le livre recommandé pour ce cours est ...». Le filtrage collaboratif tente d'automatiser ce processus de recommandations du fait qu'il établit ses suggestions à l'aide des jugements des utilisateurs semblables. Ainsi, il offre l'avantage de faire *profiter l'étudiant de l'expérience collective de ses pairs*. De plus, cette approche permet de généraliser le processus de recommandations et de le rendre plus accessible. Elle donne la possibilité à un étudiant de tirer profit de l'expérience d'une personne qu'il ne connaît même pas et de laquelle il lui aurait été difficile d'obtenir l'avis autrement.

Un autre atout introduit par cette technique est la capacité de contrôler les livres par rapport à des critères abstraits. Comme l'analyse des livres est basée sur les évaluations fournies par les autres usagers, la liste des recommandations est constituée des titres ayant obtenu les avis les plus favorables. Or, lorsqu'un utilisateur cote le degré d'appréciation des livres lus, il peut lui être demandé de considérer dans son évaluation non seulement la rigueur du livre par rapport au domaine recherché, mais aussi certaines propriétés telles que la qualité, la facilité (ou la complexité) du livre. Ainsi, le filtrage collaboratif tire profit de la capacité humaine à évaluer des critères subtils ou complexes, et qui, jusqu'à présent, sont difficilement estimés par les machines.

Un défi majeur de cette approche est la corrélation entre le nombre d'évaluations et la qualité des recommandations. Plus la matrice des cotations est dispersée, moins le système sera apte à fournir des suggestions précises et judicieuses. Or, le domaine des livres est caractérisé par un nombre réduit de retours d'un même usager par rapport aux milliers de livres qui sont souvent disponibles dans la base de données [Hirooka, *et al.* 2000], [Mooney, *et al.* 2000]. À titre comparatif, prenons l'exemple d'une base de données contenant 250 000 œuvres différentes[9] et supposons que chaque utilisateur a évalué 100 titres en moyenne. Nous nous retrouvons ainsi avec une matrice d'utilisateurs vide à 99.6% (indépendamment du nombre d'utilisateurs). Dans le domaine pédagogique, ceci ne ferait pas exception. En effet, outre la réticence de certains usagers à évaluer un grand nombre de livres, il existe plusieurs facteurs qui limitent le nombre de livres qui peuvent être évalués, notamment le facteur temporel. En effet, un étudiant se limitera souvent à l'accès de quelques livres, à cause de la durée limitée de son cours. Par conséquent, comme les étudiants ne peuvent coter pertinemment que les livres consultés, cette restriction limitera le nombre d'évaluations qu'ils pourraient faire.

---

[9] Il faut noter que plusieurs des librairies électroniques existantes, comme amazon.com et Barnes & Noble, ont un système de recommandation qui couvre un nombre de produits beaucoup plus imposant. amazon.com par exemple offre des millions de produits distincts dans ses différents catalogues [Linden, *et al.* 2003].

3.2.3 Notre système de recommandation de livres pour un environnement pédagogique

Bien que les systèmes existants puissent aider les étudiants dans la recherche de livre, ils ne comblent pas certains de leurs besoins. Nous proposons dans cette section un système de recommandation appelé DIA (Discovering Intelligent Agent) [Yammine, *et al.* 2004], conçu spécifiquement pour un milieu pédagogique. Ainsi, la structure, les techniques et l'approche générale de DIA devraient mettre au premier plan les besoins et les intérêts particuliers des étudiants.

DIA est un système qui se connecte aux bibliothèques universitaires pour suggérer des livres aux étudiants [Yammine, *et al.* 2004]. Il permet aux utilisateurs de trouver les livres qui sont susceptibles de leur plaire et de leur être utiles. Cependant, DIA n'est pas comme les engins de recherches traditionnels accessibles dans les bibliothèques, car contrairement à ceux-ci, il effectue non seulement des recherches sur la base de livres, mais présente les résultats filtrés selon l'intérêt et le profil de l'apprenant. De plus, DIA profite d'un accès à l'Internet pour augmenter la description disponible d'un livre dans le but de *mieux analyser son contenu* et de *représenter une vue plus complète des livres à l'étudiant.* En effet, bien que le système recommande uniquement les livres des bibliothèques, DIA explore le Web afin de trouver une riche description de chaque livre. Ainsi, il peut trouver un résumé, un préambule, une préface, la table des matières ou plusieurs autres descriptions indispensables. Ce système analyse cette description, filtre les livres et les trie en vue de simplifier la tâche des étudiants. Il limite la quantité d'informations en éliminant certains livres inutiles à un usager, pour ne lui présenter que les livres compatibles avec son profil.

3.2.3.1 La source principale des livres

Comment DIA construira-t-il sa base de livres et comment assurera-t-il la pertinence et le suivi des livres qu'il y inclura? Cette question est essentielle si nous désirons offrir des recommandations appropriées. En effet, nous estimons que dans un contexte pédagogique, la pertinence des livres exploités par le système affecte directement la qualité des

recommandations. Un système qui couvre un ensemble de livres inadéquats est voué à l'échec. DIA forme sa base de données à partir de la bibliothèque universitaire et ce, pour plusieurs raisons. La première est pour une question de cohérence, étant donné que les bibliothèques universitaires sont dédiées, par l'établissement scolaire, à l'approvisionnement des étudiants en ressources pédagogiques nécessaires à leurs cheminements. De plus, le conflit d'intérêts soulevé dans la partie précédente est inexistant dans ce cas, vu l'absence du caractère commercial et lucratif des bibliothèques. Bien au contraire, il y aurait même *une complémentarité et une concordance des intérêts* puisque la bibliothèque et le système de recommandation auront un but commun : permettre l'accès des livres adéquats aux étudiants.

Sur un autre plan, la bibliothèque est dotée, en général, de spécialistes en bibliothéconomie qui s'assurent, entre autres, de la mise à jour des catalogues. Ainsi, la bibliothèque se procure, périodiquement, des nouveautés comme suite aux demandes des professeurs, des chercheurs ou même des étudiants. En utilisant cette base de livres, *DIA gérera un ensemble dynamique de livres, appropriés aux programmes suivis par les étudiants*.

Un autre avantage qui en découle est la facilité d'accès aux livres, que ce soit d'un point de vue géographique (la proximité de la bibliothèque), ou d'un point de vue de « l'abordabilité » des livres procurés.

3.2.3.2 La représentation des utilisateurs

Un point essentiel à la génération de bonnes recommandations personnalisées est la connaissance de l'usager qui recherche ces recommandations. Pour suivre la logique dans laquelle il est conçu, DIA doit avoir une représentation axée « pédagogie » et moins « commerce », contrairement aux systèmes de recommandations de livres existants. Dans notre cas, nous parlerons de **modélisation de l'apprenant** plutôt que de la modélisation des utilisateurs. Ce point sera discuté en profondeur dans le chapitre suivant.

3.2.3.3 La méthode de filtrage

Comme nous avons vu dans la section 3.2.2, le filtrage collaboratif et le filtrage basé sur le contenu ont chacun leurs avantages et leurs inconvénients. Afin de bénéficier de l'apport de chacun d'eux, DIA utilise une approche hybride appelée *le filtrage collaboratif pyramidal* [Abdel-Razek 2004], [Abdel-Razek, *et al.* 2004]. Ainsi, le système emploie le filtrage basé sur le contenu et le filtrage collaboratif dans la génération de ses recommandations. Les détails de cette approche seront discutés dans le chapitre 4.

3.2.3.4 Autres considérations

Ce système de recommandation pourrait être bénéfique, non seulement aux apprenants, mais aussi aux bibliothèques qui désirent personnaliser le service offert aux étudiants. Ainsi, DIA devrait être apte à s'intégrer facilement dans plusieurs environnements pédagogiques, notamment à la bibliothèque, ou sur la page Web d'un cours particulier ou dans un cadre de cours électroniques.

3.3 Conclusion

Dans la littérature et dans la pratique, plusieurs systèmes de recommandation de livres ont été développés. Cependant, aucun d'entre eux n'a été conçu explicitement dans un cadre pédagogique, malgré l'omniprésence du processus de recommandation dans le quotidien des étudiants. Nous proposons DIA [Yammine, *et al.* 2004], un système de recommandation de livres pour les étudiants universitaires. Vu la spécificité de l'environnement pédagogique, DIA diffère des autres systèmes sur plusieurs points : en premier lieu, par l'absence du conflit d'intérêts traditionnellement existant dans les systèmes de recommandations ; deuxièmement, par la recommandation de livres *des bibliothèques*. Ainsi, DIA favorise un certain contrôle sur la qualité des livres recommandés, leur facilité d'accès. Il assure une complémentarité et une concordance avec les objectifs des bibliothèques universitaires.

Chapitre 4
DIA,
architecture et description détaillée

Dans ce chapitre, nous nous intéressons à l'architecture de DIA. Nous commençons par discuter du fonctionnement général du système. Ensuite, nous étudions le détail de chacun de ses trois composants principaux et nous décrivons les méthodologies qu'il utilise.

4.1 Le processus de recommandation de DIA

Avant d'étudier l'architecture détaillée du noyau, nous discutons, d'une manière plus abstraite, du processus de recommandation de DIA.

Essentiellement, la recommandation de livres est divisée en trois phases principales :

1. La phase de la **collecte des données** (*la phase hors ligne*) : cette phase s'occupe de la collecte des données relatives aux livres des bibliothèques. Donc, dans cette phase, une analyse du domaine (comme le cours) est effectuée pour que les livres qui lui sont appropriés soient établis. Les données de ces derniers sont ensuite stockées dans la base qui leur est réservée. Nous dénotons cette étape par phase « hors ligne », puisque ce processus est effectué bien avant la recommandation des

livres. Elle est gérée par l'administrateur du domaine et non pas par l'étudiant lui-même.

2. La **gestion du profil** de l'étudiant : cette phase est divisée en deux tâches fondamentales :
   a) la *modélisation de l'apprenant*, qui consiste à identifier et à représenter correctement le style d'apprentissage de l'étudiant,
   b) la *mise à jour du profil*. Tout au long de l'utilisation du système ou après avoir parcouru la liste des recommandations, l'étudiant peut spécifier ses préférences en sélectionnant les livres qu'il trouve pertinents. Ainsi, DIA stocke toute nouvelle information et suit les changements dans les goûts de l'étudiant afin de mieux adapter les suggestions futures.
3. La phase de **la recommandation** (*la phase en ligne*) : comme son nom l'indique, cette phase est dédiée à l'estimation des recommandations. Ainsi, elle encapsule toutes les tâches qui se rapportent aux calculs des estimations, notamment le calcul des similarités entre les étudiants et la génération d'une liste de livres ordonnés selon le profil de chacun.

La figure 4.1-1 schématise l'architecture générale de DIA et nous montre que chacune de ces phases est définie par un segment du noyau bien précis. Dans la section qui suit, nous parlons, en détail, de la structure de chacun.

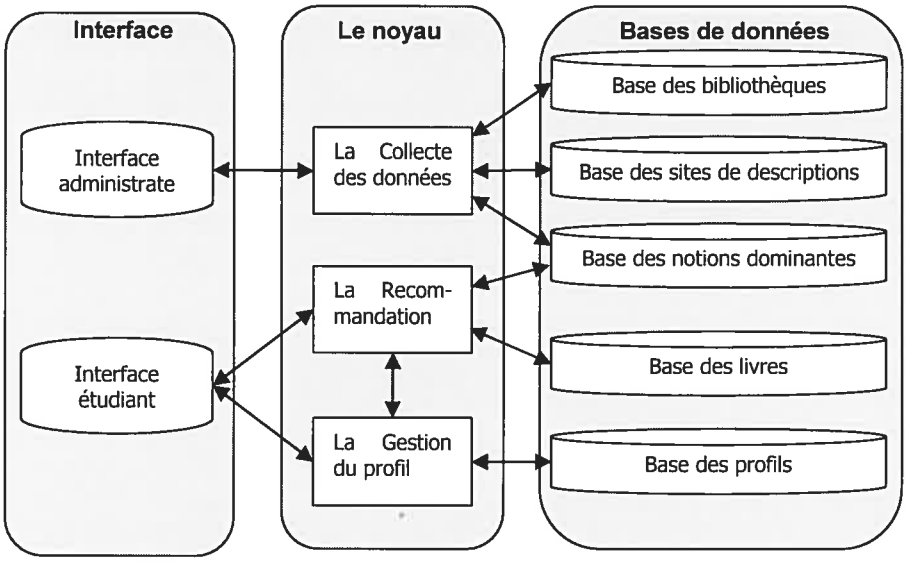


Figure 4.1-1. L'architecture générale de DIA

4.2 Le noyau

Le noyau constitue le principal composant du système. Ce « *tiers* » est dédié principalement aux calculs des recommandations et à toutes les tâches qui s'y rapportent. De plus, le noyau joue le rôle de façade ou d'intermédiaire entre l'interface et les bases de données. En d'autres mots, toutes les données fournies par l'utilisateur passent par le noyau lorsqu'elles doivent être stockées dans les bases de données et vice-versa, les données extraites des bases de données doivent passer par le noyau avant de la présenter à l'utilisateur. Selon le besoin, les données qui transitent par le noyau peuvent être manipulées ou transformées par ce dernier, notamment lors de l'estimation des recommandations. Ainsi, le noyau revêt une importance cruciale pour le système et peut être vu comme le cerveau de DIA.

4.2.1 La collecte des données

La collecte des données consiste essentiellement à trouver l'ensemble des livres qui sont en relation avec le domaine couvert par la recommandation. Elle sert à la reconnaissance des

titres pouvant être utile aux étudiants d'un domaine. L'architecture de cette dernière, présentée à la figure 4.2-1, nous montre qu'elle est composée de trois modules :

- le module *analyse du domaine*, qui a pour rôle d'étudier un domaine (tel qu'un cours) dans le but de discerner ses notions les plus importantes (les notions dominantes);
- le *module gestion des livres* s'occupe de la collecte des données bibliographiques et descriptives de livres couvrant les notions du domaine;
- le *module analyse des livres* compare la description des livres avec celle du domaine afin d'identifier les livres qui lui sont le plus appropriés.

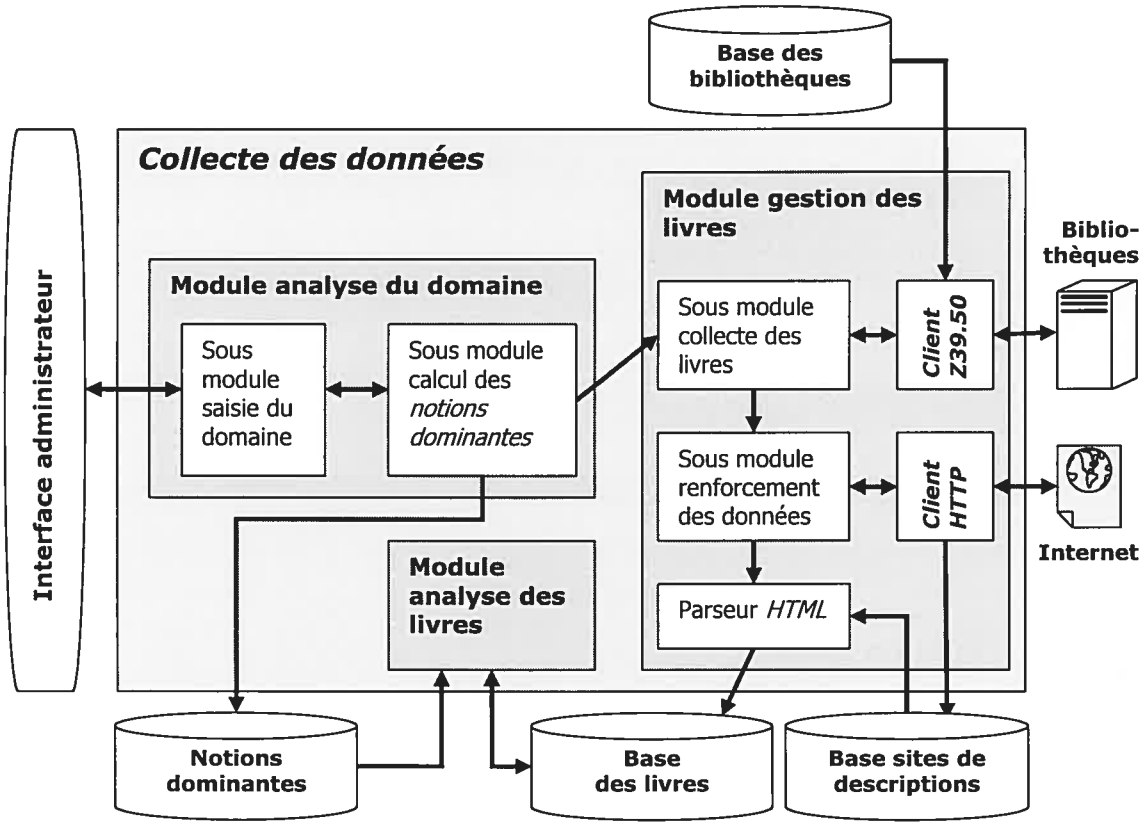


Figure 4.2-1. Architecture détaillée du processus de collecte des données

4.2.1.1 Le module analyse du domaine

Ce module est dédié principalement à l'étude du domaine pour lequel l'administrateur (le professeur) désire offrir des recommandations. La représentation du domaine est un aspect important pour DIA. En effet, pour recommander des livres adéquats, le système doit avoir une certaine *connaissance du domaine* pour lequel il génère des recommandations. Par exemple, si un étudiant souhaite recevoir des suggestions de livres pour son cours de *programmation Java*, il existe sûrement à la bibliothèque une multitude de livres qui traitent de ce sujet. Or, bien qu'ils couvrent tous le même domaine, certains de ces livres ne sont pas « pédagogiquement » convenables pour l'étudiant, car les notions considérées par ces derniers ne correspondent pas à celles du cours en questions. Par conséquent, DIA a besoin d'une bonne *représentation* de la conceptualisation du domaine, lui permettant ainsi de bien connaître les notions importantes de ce dernier. Ce point est essentiel si nous désirons parvenir à des recommandations adéquates.

Ce module interagit directement avec le responsable du domaine afin de construire une base de connaissances des **Notions Dominantes** (*ND*) du domaine [Abdel-Razek 2004], [Abdel-Razek, *et al.* 2003]. Cette étape est effectuée bien avant le processus de recommandation et n'a besoin d'être répétée que si des modifications majeures ont eu lieu au domaine (exemple : un changement dans le programme du cours).

Sous module saisie du domaine

Ce module s'occupe de la saisie des paramètres décrivant le domaine. Afin de faciliter le travail de l'administrateur, il offre une grande flexibilité étant donné qu'il ne requiert que le nom du domaine, les autres paramètres étant facultatifs. Cependant plus l'administrateur offre des descriptions représentatives, mieux l'analyse du domaine se fera et plus le système parviendra à de meilleures recommandations. Les descriptions qui peuvent être fournies sont réparties comme suit :

- **nom du domaine** (*obligatoire*) : le nom du cours pour lequel l'administrateur désire faire des recommandations (exemple, Introduction aux bases des données);

- **description du domaine** (*optionnelle*) : dans le cas d'un cours, souvent, une description de ce dernier est déjà disponible et peut être facilement introduite par l'administrateur;
- **notions relatives** (*optionnelles*) : les mots-clés ou notions importantes du cours ou du domaine;
- **livres de référence** (*optionnels*) : l'administrateur peut inclure aussi le titre et auteur(s), ou l'ISBN des livres de référence du cours. DIA utilisera, entre autres, ces livres pour trouver l'ensemble des livres les plus adéquats au domaine.

Lorsque ces données sont validées, elles sont envoyées au sous module calcul des notions dominantes.

Sous module calcul des Notions Dominantes (ND)

Une des tâches principales de DIA est la représentation d'un domaine sous une forme permettant une classification convenable des livres. Nous utiliserons l'approche du « **Dominant Meaning** » (*DM*) présentée par [Abdel-Razek 2004], [Abdel-Razek, *et al.* 2003]. Cette approche a l'avantage, entre autres, d'être plus performante que le « Bag-of-Words » et plus efficace que la simple utilisation de mots-clés [Abdel-Razek 2004]. Elle permet aussi de représenter un domaine par une hiérarchie de concepts qui lui sont relatifs. Nous commençons par définir le dominant meaning pour ensuite expliquer comment DIA représente ses domaines.

> **Définition :** "*The dominant meaning is a set of keywords that best fit an intended meaning of a target word*" [Abdel-Razek 2004]. Le **sens dominant** est un **ensemble de mots-clés** qui expliquent mieux la signification souhaitée d'un mot cible.

Pour clarifier ce concept, supposons que le mot cible est « *Java* ». Or, le mot Java a trois significations bien connues : le langage de programmation Java, le café Java et Java l'île indonésienne. Ainsi, le terme Java ne suffit pas à la compréhension du domaine ciblé et

donc un ensemble de mots-clés doit être employé pour mieux représenter sa signification désirée. Cet ensemble de mots est appelé « dominant meaning » ou le sens dominant.

Regardons l'aspect formel derrière le sens dominant. Soit C un concept, appelé aussi mot cible. C est représenté par un ensemble de mots W. Pour chaque mot w, tel que $w \in W$, on associe une mesure ou un poids P, appelé « *dominant meaning distance* » ou *distance du sens dominant*. Plus le poids P est grand, plus w représente mieux la signification du concept C. Le sens dominant de C est formé des k mots ayants la mesure P la plus élevée.

Le sens dominant permet la représentation du concept cible sous forme d'un vecteur de mots-clés, où le poids de chaque mot est P. Cette approche permet aussi une représentation hiérarchisée du concept à l'aide du *Dominant Meaning Graph* (*DMG*) [Abdel-Razek 2004], [Abdel-Razek, *et al.* 2003]. Ce graphe est constitué d'un ensemble de nœuds (les mots) et d'un ensemble d'arc (le poids). Les mots de chaque niveau sont reliés par un arc aux mots du niveau supérieur. Chaque arc a un poids non négatif $P_{i,j}$, où $P_{i,j}$ représente la *distance du sens dominant* entre les mots w_i et w_j.

Abdel-Razek *et al.* ont suggéré un modèle mathématique, basé sur l'information lexicologique globale de cooccurrence, permettant la génération du DMG. Nous nous sommes inspirés de ce modèle afin de trouver les *sens dominants* du domaine. Dans notre cas, il serait plus approprié de parler de **notions dominantes** du domaine plutôt que de sens dominants. Rappelons-nous de la problématique : DIA doit représenter les notions importantes d'un domaine pour qu'il puisse rechercher les livres de la bibliothèque qui s'y rapporte. Pour représenter le domaine sous forme hiérarchique de concepts ou de notions, DIA utilise les descriptions introduites par l'administrateur. Un domaine C est défini par un ensemble de description tel que $C = \{D_v \mid v = 1, \ldots, r\}$. Ainsi, $D_v \in$ {nom du domaine, description du domaine, notions relatives, la description des livres de référence}. Il faut noter que la description des livres de référence est trouvée automatiquement par DIA (voir la partie 4.2.1.2).

De plus, chaque description D_v est composée d'un ensemble de mots, tel que $D_v = \{w_j \mid j = 1, \ldots, n\}$. Supposons qu'un domaine C soit symbolisé par les mots w_c^k avec $k = 1, \ldots, m$ (exemple, on a w_c^1 = Java). Pour trouver les notions dominantes de C, nous calculons P_{kj}, la distance du sens dominant entre le mot w_c^k et les mots descriptifs w_j tel que

$$P_{k,j} = P(w_j \mid w_c^k) = \frac{1}{r}\left[\sum_{v=1}^{r} \frac{F(w_j \mid D_v)}{F_c}\right] \quad \forall k = 1,\ldots,m \quad \text{et} \quad \forall j = 1,\ldots,n \qquad (4.2\text{-}1)$$

avec $F(w_j \mid D_v)$ la fréquence du mots w_j dans D_v

avec $F_c = \max_{v=1,\ldots r} \{F(w_c^k \mid D_v)\}$ et $0 \le \max_{v=1,\ldots r} \{F(w_j \mid D_v)\} \le F_c$

[Abdel-Razek 2004]

Ensuite, il est facile d'extraire l'ensemble C_k formé des T notions les plus dominantes du domaine. Ce sont les mots qui ont les T plus grandes valeurs $P_{i,j}$.

4.2.1.2 Le module gestion des livres

Lorsque le domaine est analysé, ce module a pour objectif de recueillir les données des livres présents dans les bibliothèques, afin que DIA puisse construire sa propre base de livres. Ce module possède deux fonctions principales : celle d'effectuer des recherches sur les bases des données des bibliothèques et celle d'en extraire les données bibliographiques. Cependant, ces dernières ne sont pas suffisantes pour permettre des recommandations de qualité, car souvent il n'y a que le titre et la liste de sujets qui nous offrent de l'information sur le contenu des livres. Or, il existe une multitude de sites Web qui offre des descriptions détaillées sur ces derniers (tel que, la table des matières, le résumé, la description, le préambule). Donc, sa vocation secondaire est de parcourir le Web à la recherche d'informations complémentaires pouvant enrichir la description des livres. La figure 4.2-2 schématise les deux fonctions principales de ce module.

Cette manière de faire comporte deux avantages :

- **parvenir à de meilleures recommandations**, puisque le système peut alors *analyser en détail le contenu des livres*.

- Permettre à l'étudiant de faire une **recherche plus centralisée**. En effet, contrairement aux systèmes de recherches traditionnels, plusieurs détails d'un livre sont disponibles. Donc, en un clic de souris, *l'étudiant accède à des informations approfondies* sur le livre sans avoir à les chercher sur les rayons de la bibliothèque. Ceci rend possible une économie d'efforts et de temps.

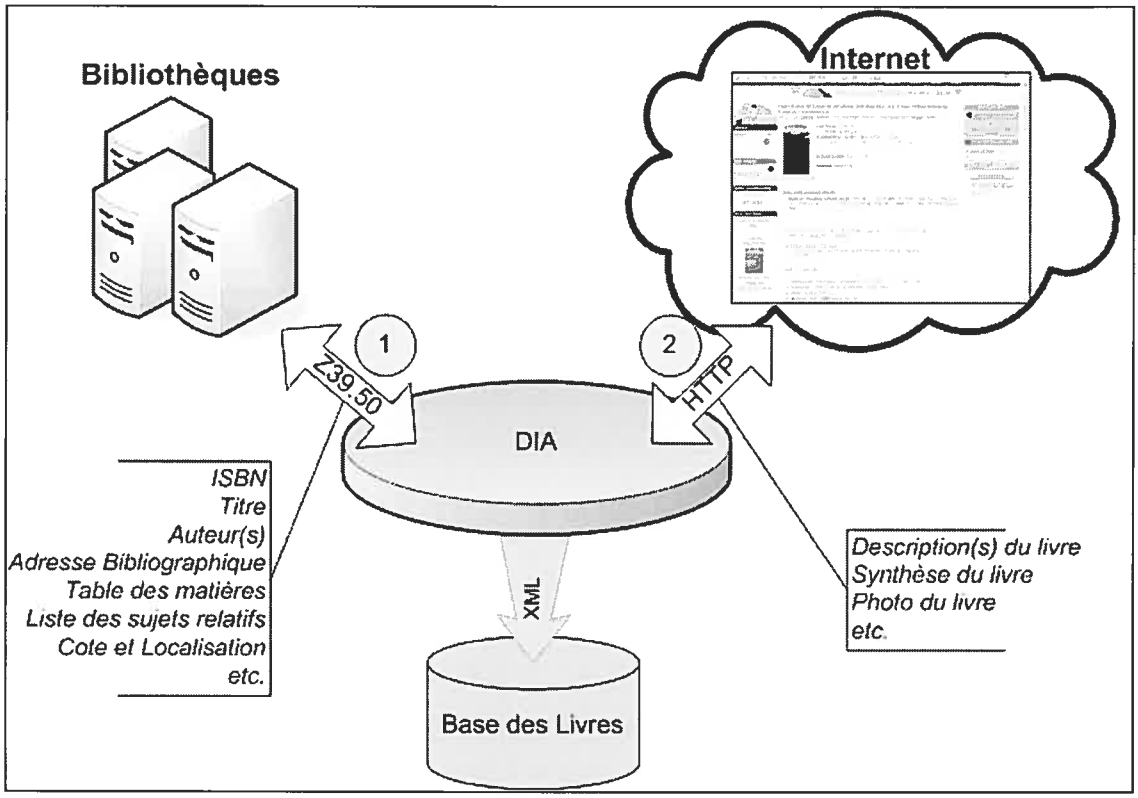


Figure 4.2-2. Le processus de la collecte des données des livres par DIA

Sous module collecte de livres

Ce module prend en entrée une liste de mots-clés représentant les notions importantes du domaine (ou du cours). Ce module s'occupe de la recherche de livres des bibliothèques qui couvrent ces notions. Afin de faciliter l'utilisation du système, ce module est géré par DIA d'une manière autonome. En administrant le client Z39.50 (voir 4.2.1.3), il parcourt les données bibliographiques disponibles dans les bases de la bibliothèque et extrait celles des livres qui couvrent les notions du domaine. Aucune analyse de livres n'est effectuée dans ce module. Son rôle consiste à extraire les livres qui se rapportent aux notions dominantes

du domaine. Donc, une liste contenant les données bibliographiques de livres relatives au domaine est envoyée au sous module renforcement des données dans le but d'augmenter les données disponibles sur un livre.

Sous module renforcement des données

Une fois les données bibliographiques des livres extraites, DIA essaye de chercher sur le Web afin de trouver une description des livres plus approfondie, comme la synthèse du livre, un descriptif général sur le contenu du livre, la table des matières, la couverture du livre (l'image), etc. Ce module fait appel au client HTTP de DIA pour effectuer des recherches sur des sites Web contenant les descriptions des livres. Une liste de ces sites est maintenue dans une base de données. Ainsi, il construit des requêtes pour chaque site Web, les envoie au travers du client HTTP et recueille les résultats pour les envoyer au Parseur (analyseur syntaxique) HTML. Ainsi, ce module prend en entrée une liste de livres (ainsi que leurs données bibliographiques) et retourne une liste de pages Web contenant une description détaillée de chaque livre.

Parseur HTML

Ce module prend en entrée la page Web descriptive de chaque livre et les règles d'extraction du document. La sortie est constituée d'un document *HTML* décomposé en éléments.

Le module de recherche envoie les pages Web trouvées au parseur pour qu'il les décompose en éléments compréhensibles et utilisables par le système. Chaque site Web peut générer des pages descriptives ayant une architecture et une structure différente. Alors, le parseur a besoin de certaines règles qui lui indiqueront comment décomposer le document. Cette étape est assez simple, puisque souvent, les pages d'un même site Web sont générées automatiquement et suivent donc la même structure. Une fois le document décomposé, le parseur le décode de ses balises et envoie les nouvelles données extraites à la base des livres.

4.2.1.3 Le client Z39.50

Beaucoup de bases de données en ligne sont disponibles sur l'Internet mais chacune possède ses propres procédures d'accès, son interface utilisateur, un langage de requêtes spécial et surtout une architecture qui lui est unique. La norme Z39.50 offre une solution à cette situation de diversité et de manque d'interopérabilité [Ludwig, *et al.* 1997]. La norme Z39.50 définit un ensemble de commandes, pouvant **assurer l'indépendance de plateforme et une grande interopérabilité**. Elle définit clairement la syntaxe et la sémantique des requêtes, et permet de rechercher et d'extraire pratiquement n'importe quel genre de données.

Ainsi, la norme Z39.50 a pour objectif de rendre possible l'interrogation simultanée de bases de données hétérogènes et réparties. Cette norme, gérée par la bibliothèque du Congrès[10] et reconnue par la *National Information Standards Organization*[11] (*NISO*), a évolué au fur et à mesure du temps. La première version de la norme Z39.50, connue aussi sous le nom Z39.50-1988, est devenue obsolète lors de son replacement par la version 2 : Z39.50-1992. La version 3, appelée aussi Z39.50-1995, complémente la version précédente tout en lui étant compatible. En 1997, cette norme fut adoptée par l'*Organisation internationale de normalisation*[12] (*ISO*) sous l'appellation *ISO 23950*. La dernière révision en date, Z39.50-2003 [NISO 2003] est une révision des versions 2 et 3.

> "Cette norme définit **un service et un protocole de type client/serveur** pour la recherche d'informations. Elle **spécifie des procédures et des formats** pour qu'un *client* recherche une base de données accessible par un *serveur*, pour extraire des enregistrements de la base de données et pour *exécuter diverses fonctions* relatives à la recherche d'informations. Le protocole s'occupe la communication entre les applications de recherche d'informations du côté client et entre le serveur; il ne se préoccupe pas de l'interaction entre le client et l'utilisateur." [NISO 2003]

---

[10] http://www.loc.gov/z3950/agency/

[11] http://www.niso.org

[12] http://www.iso.org

Vu la complexité du protocole, il serait impossible de couvrir entièrement ses détails dans ce mémoire. Dans cette section, nous présentons la notion de base du Z39.50. Pour une description plus détaillée, la définition de la norme [NISO 2003] est accessible à partir du site Web de la bibliothèque du Congrès[13].

Z39.50 est un protocole de la couche réseau (de la couche application dans le modèle de référence OSI). Il indique des formats et des procédures régissant l'interaction d'un client et d'un serveur. Le client peut demander au serveur de rechercher une base de données et d'identifier les enregistrements correspondants à la requête. Le client peut alors extraire les enregistrements identifiés. Une session Z39.50 typique est constituée de trois transactions : l'*initialisation* (*initialise*), la *recherche* (*search*) et la *présentation* (*present*) [Moore, *et al.* 2000].

Le premier service Z39.50 est l'initialisation. Après qu'une *connexion* soit établie, le client envoie une « **init request** », et le serveur répond avec une « **init response** ». Cet échange de messages (qui est défini dans le *Protocol Data Units*) [NISO 2003], permet l'authentification du client et la négociation de plusieurs paramètres de session. Parmi ceux-ci, nous retrouvons par exemple, la version du protocole qui sera utilisée, la taille des messages et les options à employer dans le service de recherche.

Une fois une *session* établie, le client peut émettre une demande de recherche. Dans cette dernière, le client identifie entre autres, le nom des bases de données à chercher et présente une requête de recherche sous une **forme standardisée**. Comme nous avons indiqué plus haut, un des défis des bases de données est l'indépendance entre la structure des requêtes et l'architecture de la base de données. Par exemple, en SQL, une requête telle que

SELECT *titles*, *author* **FROM** *book_tiles*

[13] http://www.loc.gov/z3950/agency/

dépend directement de la base pour laquelle elle est destinée. Ce qui veut dire que cette requête ne peut pas être valide pour une autre base qui a une architecture différente ou un nom de table différent (il faut que les champs *titles* et *author* et la table *book_title* aient exactement le même nom). Or souvent, nous nous retrouvons avec des bases contenant le même type de données, telles que des données bibliographiques de livres, mais qui ont une architecture différente. Il n'est alors pas possible d'envoyer des requêtes à ces bases sans connaître explicitement l'architecture de chacune d'elle, même si elles doivent certainement avoir des champs en communs. Pour faire face à ce problème, Z39.50 essaye de standardiser les requêtes entre le client et le serveur selon la nature des données recherchées. Il définit donc plusieurs grammaires de requêtes entre autres :

- **Les requêtes de Type-0** sont réservées pour les grammaires privées (définies en dehors de la norme).
- **Les requêtes de Type-1** doivent être supportées par chaque serveur. Ce sont les seuls types de requêtes largement admises. Elles se composent d'un ou de plusieurs *termes de recherche* qui appartiennent à un jeu d'attributs bien défini et qui dépends de la nature des données recherches (données bibliographiques, données géospatiales, etc.). Le serveur est responsable de traduire ces attributs selon la conception logique de la base de données.
- **Les requêtes de Type-2 et de Type-100** emploient les grammaires des langages de commande ISO 8777 et ANSI/NISO Z39.58.
- **Les requêtes de Type-101** sont une extension des requêtes de Type-1 pour soutenir la recherche de proximité.

Il faut noter que, pour notre système, nous sommes intéressés par le jeu d'attributs **Bib-1**[14] car, ces attributs définissent la grammaire des requêtes pour les ressources de type bibliographique.

---

[14] http://www.loc.gov/z3950/agency/

Pour mieux comprendre cette norme, la figure 4.2-3 présente un schéma simplifié d'une session Z39.30.

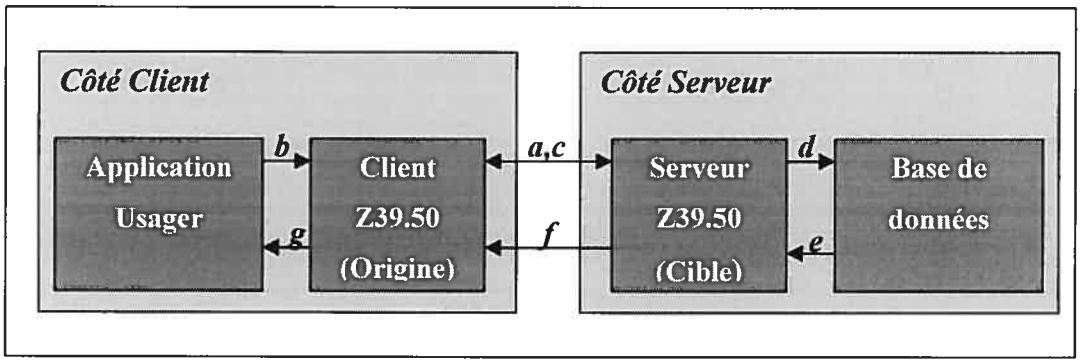


Figure 4.2-3. Le standard Z39.50, un schéma simplifié

a. Une négociation préliminaire à lieu (*le service d'initialisation* entre le client et le serveur) afin d'établir les règles de « l'association » entre les deux systèmes.
b. L'utilisateur fournit au client (à l'origine) les termes recherchés.
c. Le client traduit alors les termes recherchés en une requête ayant une grammaire standardisée à l'aide d'un ensemble d'attributs bien déterminé selon la nature des données et il contacte le serveur (la cible),
d. Le serveur traduit la requête reçue en une requête de recherche spécifique au langage de la base de données du serveur et à son architecture.
e. Le serveur reçoit une réponse relative au nombre de résultats et traduit les résultats dans un format spécifique.
f. Le client reçoit les données correspondantes à la requête.
g. Le résultat est envoyé à l'application de l'usager.

En résumé, la norme Z39.50 repose sur le principe client/serveur. Le client initie le dialogue avec le(s) serveur(s) et ensuite, il agit au nom de l'utilisateur en traduisant le langage d'interrogation d'un système particulier en une forme standardisé. Ce format ne dépend pas de l'architecture de la base de données. Quand une requête Z39.50 atteint le serveur, elle est traduite dans la syntaxe locale de l'engin de recherche. L'engin de recherche fournit les résultats de la recherche, qui sont alors traduits par le serveur dans le

format Z39.50, avant d'être retournés au client. Ainsi, au lieu de chercher la base de données directement, le client un protocole, qui a pour rôle de transformer les requêtes normalisées du client, en requêtes spécifiques à la base de données. La puissance de cette norme découle de la standardisation de la communication entre le client et le serveur. **Elle permet au client d'interroger *simultanément* plusieurs bases de données hétérogènes *et réparties***

- *sans connaître la structure* des données dans les bases interrogées;
- à l'aide *d'une requête unique*;
- où la récupération des données se fait dans *un format normalisé facilement réutilisable.*

Par conséquent, le client Z39.50 permet à DIA de se connecter à n'importe quelle bibliothèque ayant un serveur Z39.50. Or des milliers de bibliothèques universitaires, gouvernementales ou municipales implémentent déjà ce protocole. DIA peut alors consulter et rechercher leurs bases de livres et recommander facilement les titres qu'elles contiennent. Ce point est très important. En effet, grâce au client Z39.50, **DIA a la possibilité de suggérer simultanément des livres à partir de milliers de bibliothèques universitaires**. Par exemple, si Frédéric est un étudiant à l'Université de Montréal, DIA peut lui recommander des livres de la bibliothèque de son université, mais aussi, s'il le désire, ceux de l'Université Concordia, de l'Université McGill et de l'UQAM. En annexe, nous avons joint une liste de quelques universités qui emploient ce protocole, l'adresse des serveurs et le nom des bases de données.

4.2.1.4 Le client HTTP

Le client HTTP prend comme entrées les requêtes à envoyer aux sites Web et retourne les réponses de ces derniers sous forme HTML (les pages Web décrivant le livre). Ce module peut être considéré comme la porte de DIA vers l'Internet. Il est géré directement par le module renforcement des données et ne requiert donc pas l'intervention directe des usagers. Il prend le rôle d'un fureteur (simplifier) lui permettant d'accéder au Web et d'en extraire

l'information recherchée. Il faut noter que ce module n'effectue aucune analyse des documents trouvés, il joue uniquement le rôle de l'intermédiaire entre le système et l'Internet.

4.2.1.5 Le module d'analyse des livres

L'objectif de ce module est de calculer le lien entre chaque livre et le domaine. Après avoir fait l'analyse de ce dernier pour le représenter par ses notions les plus importantes (voir 4.2.1.1) et après avoir extrait une description approfondie des livres (voir 4.2.1.2), DIA étudie le contenu du livre pour identifier ceux qui sont les plus appropriés pour le domaine.

Nous utilisons la **Similarité des Sens Dominants** (*SSD*) [Abdel-Razek 2004] entre un domaine et la description d'un livre pour mesurer leur proximité. Cette mesure consiste à estimer la valeur de la relation qui lie un livre et un domaine. Plus les notions traitées dans le livre se rapprochent de ceux du domaine, plus la *SSD* sera grande.

Supposons que C soit un domaine dont l'ensemble de notions dominantes est $\{w_1, \ldots, w_m\}$. Supposons aussi que $\{d_1, \ldots, d_s\}$ soit la description d'un livre L. Le but est d'évaluer les livres qui offrent le plus grand degré de similarité avec le domaine C. Nous calculons la *Similarité des Sens Dominants* entre C et L, $SSD(L, C)$ d'après [Abdel-Razek 2004] :

$$SSD(L,C) = \frac{1}{s}\sum_{i=1}^{s}\frac{1}{m}\left[\sum_{j=1}^{m}\Theta(w_j, d_i)\right]$$
$$\text{avec} \quad \Theta(w_j, d_i) = \begin{cases} 0 & \text{si } w_j \neq d_i \\ 1 & \text{si } w_j = d_i \end{cases} \qquad (4.2\text{-}2)$$

Lors de cette analyse, la *SSD* de chacun des livres est sauvegardée dans la base des livres ainsi que l'ensemble de mots E, où $E = \{w_1, \ldots, w_m\} \cap \{d_1, \ldots, d_s\}$, c'est-à-dire, les notions du domaine qui sont discutées par le livre.

4.2.2 La gestion du profil

Les recommandations de qualités passent obligatoirement par la connaissance de l'individu pour qui nous les générons. La gestion du profil de l'apprenant consiste à représenter l'étudiant au système pour permettre à ce dernier de suggérer les livres qui correspondent le mieux à son profil. Ce dernier contient un mélange de données statiques et dynamiques. Les données statiques changent peu durant une longue période de temps, telles que le nom de l'apprenant ou son nom d'usager. Par contre, les titres préférés d'un usager et les notions qui l'intéressent sont des exemples de données dynamiques; ils changent et évoluent souvent ou constamment. Or, DIA calcule ses recommandations à partir de ces données changeantes. Par conséquent, les profils des étudiants doivent être mis à jour régulièrement, pour produire des recommandations pertinentes. La gestion du profil se compose de la modélisation de l'apprenant et de la mise à jour du profil, comme nous pouvons constater à la figure 4.2-4.

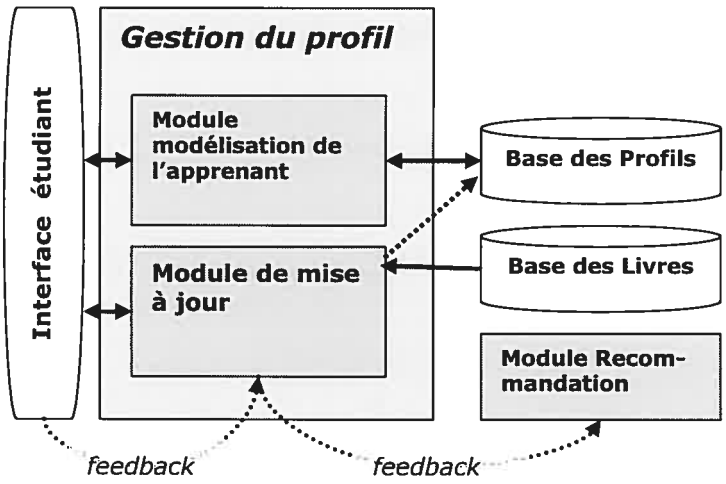


Figure 4.2-4. L'architecture détaillée de la gestion du profil

4.2.2.1 Le module de modélisation de l'apprenant

Avec l'aide de l'étudiant, ce module génère le profil de ce dernier. Ce processus est lancé lors de l'inscription de l'étudiant au système ou peut être exécuté à sa demande si des changements de données doivent être faits. Ce module prend en entrée :

- le nom de l'étudiant;
- le nom de l'université et le niveau d'étude;
- le nom d'usager et le mot de passe;
- le courriel;
- les cours (ou les domaines) pour lesquels il désire recevoir des recommandations.

De plus, ce module a pour rôle de *détecter le style d'apprentissage* de l'apprenant et de le valider avec ce dernier. Nous utilisons la méthode VARK[15], un questionnaire spécialisé de 13 questions capables de détecter les préférences d'apprentissage des utilisateurs. Ces derniers sont réparties en Visuel (*V*), auditif (*A*), écriture/lecture (*R*) et kinesthésique (*K*) et multimodale (qui consiste en toute combinaison de ces styles).

Une fois les données approuvées, ce module génère le profil et le stocke dans la base qui lui est réservée.

4.2.2.2 Le module de mise à jour

Après avoir passé en revue les livres suggérés par le système, ou quand il parcourt la liste des livres relatifs au domaine, un étudiant peut cocher (sélectionner) ses titres préférés. Ceci indique au système que l'étudiant apprécie le livre. Grâce à ce feed-back, DIA met à jour le profil de l'étudiant chaque fois qu'un titre est choisi. Le système ajoute au profil l'ISBN du livre en question et les notions du domaine qu'il couvre (voir 4.2.1.5) comme le montre la figure 4.2-5. Les recommandations suivantes sont basées sur ce profil actualisé.

Ainsi, cette étape peut être répétée plusieurs fois afin d'améliorer les recommandations. Ces sélections permettront au système de superviser les changements de préférences chez l'utilisateur et d'adapter les futures recommandations aux données additionnelles. Il faut noter qu'avec la procédure inverse, c'est-à-dire, lors de la « désélection » d'un livre, DIA retire l'ISBN et les notions importantes du livre du profil de l'étudiant.

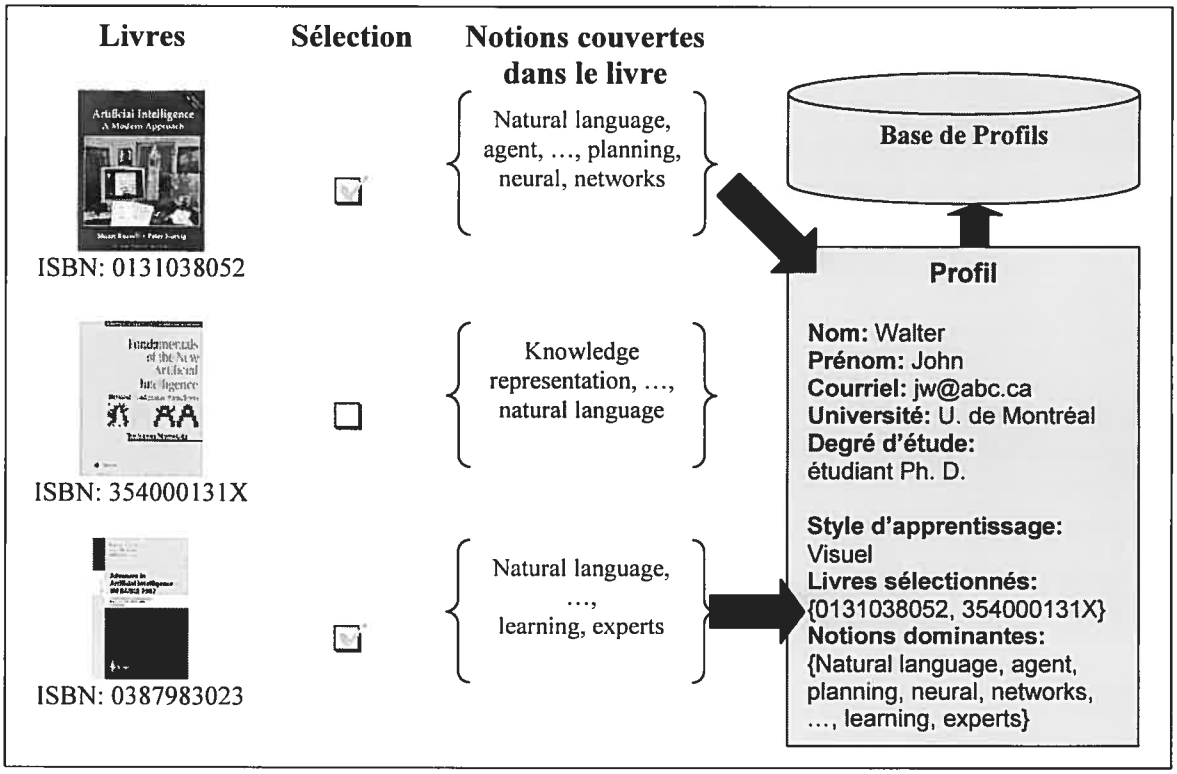


Figure 4.2-5. La mise à jour du profil d'un étudiant lors de la sélection d'un livre apprécié

4.2.3 La recommandation

Nous avons adapté l'approche du **Pyramid Collaborative Filtering Approach** (*PCFA*) (*l'approche de filtrage collaboratif pyramidale*) [Abdel-Razek 2004], [Abdel-Razek, *et al.*

---

[15] © Copyright Version 4.1 (2004) held by Neil D. Fleming, Christchurch, New Zealand and Charles C. Bonwell, Green Mountain Falls, Colorado 80819 U.S.A.

2004], pour notre problème de recommandation de livres. Nous commençons par donner un bref aperçu de cette technique et ensuite, nous montrons comment DIA a adapté cette approche à la collection de livres, lors de la génération des recommandations.

L'approche du PCFA peut être vue comme une pyramide composée de 4 niveaux. À chaque niveau, nous appliquons une technique de filtrage pour éliminer les objets les moins convenables « aux critères » du niveau. Ainsi, nous commençons avec une collection de n objets au bas de la pyramide (niveau 0). Passer d'un niveau à l'autre dépend de 3 techniques de filtrage (une entre chaque niveau). Pour aller du niveau 0 au niveau 1, nous appliquons le *domain model filtering* ou le *filtrage par rapport au modèle du domaine*. En d'autres termes, le passage au niveau supérieur se fait en enlevant les t objets les moins relatifs au domaine particulier. Ainsi, nous nous retrouvons avec un sous-ensemble de k objets, où $k = n - t$. Pour le passage du niveau 1 au niveau 2, nous filtrons les objets selon le modèle de l'usager, c'est-à-dire que nous gardons uniquement les r objets (avec $r \leq k$) qui correspondent le plus avec le profil de l'utilisateur. C'est ce que *Razek et al.* appelle le *user model filtering*. Finalement, pour passer du niveau 2 au dernier niveau, le PCFA filtre les objets restants selon des critères de crédibilité pour ne présenter à l'usager que l'objet le plus crédible. Ce dernier filtrage porte le nom *credibility model filtering*.

En résumé, chaque filtrage se fait selon des critères bien définis : *par rapport au domaine*, *par rapport à l'usager* et *par rapport à la crédibilité*. Pour la recommandation des livres pédagogiques, nous nous sommes inspirés de ce modèle, mais en l'adaptant à notre problématique. Nous avons légèrement modifié les deux premières techniques de filtrages et nous avons introduit l'ordonnancement des livres au lieu du filtrage par rapport à la crédibilité. Nous avons omis cette étape de filtrage pour plusieurs raisons. Les critères de crédibilité de livres sont un sujet complexe qui porte à discussions. Il est assez difficile de trancher sur la crédibilité d'un livre, car cela fait appel à un raisonnement subjectif et donc dépend d'une personne à l'autre. En fait, avant de vérifier si un livre est crédible ou pas, il faut s'intéresser à la question, « *Qu'est-ce qu'un livre crédible?* ». Déjà, les réponses à cette question englobent une multitude d'opinions. Il est clair que la machine ne peut

calculer efficacement ce critère (pour un livre). Le modèle de crédibilité tel que défini dans [Abdel-Razek, *et al.* 2004] est difficilement applicable à notre problème. Mais certains jugeront peut-être que le critère de crédibilité est important dans un système de recommandation de livres pédagogiques. Dans notre application, nous ne négligeons pas ce critère, mais il n'est pas géré directement par DIA. En fait, la crédibilité des livres est prise en considération par le système de deux manières : rappelons que DIA recommande les livres des bibliothèques, ce qui fait que notre collection de livres initiale est déjà « filtrée » par des individus (bibliothécaires et professeurs) ayant une certaine expertise dans le domaine. En effet, ces personnes ont la responsabilité de procurer, pour la bibliothèque, des livres pertinents et à jour. Donc, nous estimons que l'ensemble de livres que DIA doit filtrer, est au départ filtré selon plusieurs critères de pertinence, notamment celui de crédibilité. De plus, le système donne aux usagers le choix de sélectionner leurs livres préférés. Nous posons l'hypothèse que les livres les moins crédibles auront moins de chance d'être sélectionnés que ceux qui le sont et donc leur recommandation serait moins probable. On peut affirmer que bien qu'indirectement, la crédibilité des livres est considérée par le système. Ces deux points ont été discutés à la section 3.2.

Dans la partie suivante, nous expliquons l'architecture du processus de recommandation. Cette dernière s'inspire directement de l'approche pyramidale utilisée. Ainsi, elle est composée de trois modules : *module filtrage par rapport au domaine*, *module filtrage par rapport au modèle de l'apprenant*, *module d'ordonnancement des livres*. Chaque module applique un filtre à un ensemble de livres et envoie la collection résultante au module suivant. Finalement, les livres sont ordonnés selon leurs pertinences à l'utilisateur pour ensuite lui être retournés.

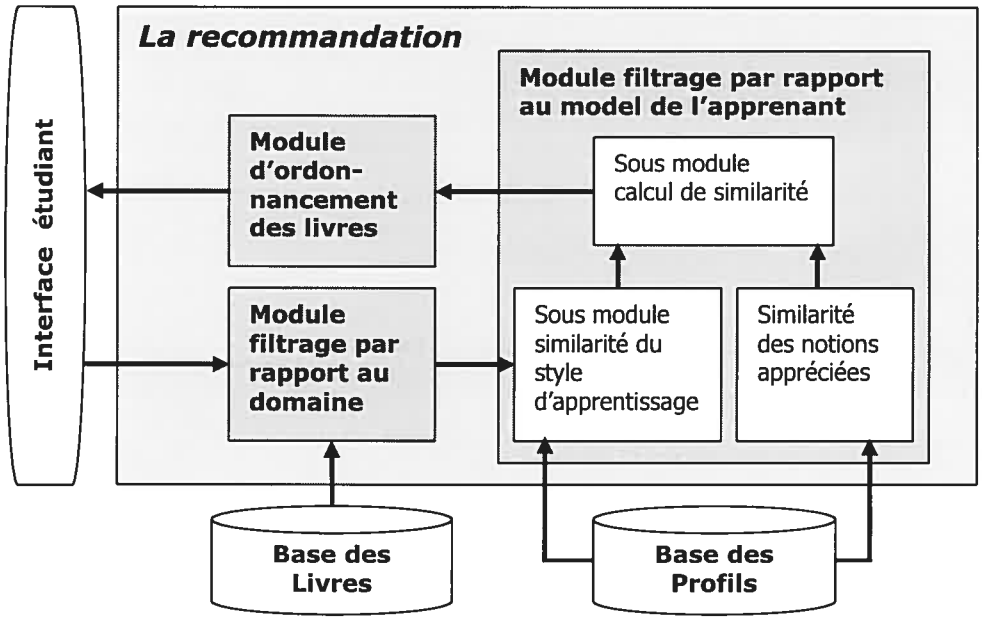


Figure 4.2-6. Architecture détaillée du processus de recommandation

4.2.3.1 Le module filtrage par rapport au domaine

Pour le passage du niveau 0 au niveau 1, nous filtrons la collection de livres selon les notions du domaine. Ainsi, les livres qui ont le plus grand lien avec le domaine sont identifiés. Or, le lien entre chaque livre et le domaine a été calculé durant la phase hors ligne (voir 4.2.1.5) en utilisant la Similarité de leurs Sens Dominants SSD. Donc, ce module classe les livres selon la valeur de SSD et ne considère que les t livres ayant le SSD le plus grand. Il faut noter que ce **filtrage est basé sur le contenu**, vu que les notions importantes des livres sont extraites directement de leur contenu respectif (au chapitre 5, nous montrons que nous avons testé notre système avec $t = 30$).

4.2.3.2 Le module filtrage par rapport au modèle de l'apprenant

Le filtrage par rapport au modèle de l'apprenant applique quant à lui le **filtrage collaboratif**. Donc, ceci requiert le calcul de la similarité entre les utilisateurs du système pour identifier ceux qui sont le plus similaires avec l'usager actif.

Le calcul de cette similarité se fait par rapport à deux critères :

1. La similarité des styles d'apprentissage.
2. La similarité des notions des livres appréciées.

Ainsi, le système fera profiter l'usager actif de l'expérience des autres usagers qui ont les styles d'apprentissage et les notions des livres appréciées le plus similaires avec les siens. Une fois l'ensemble d'étudiants qui lui est le plus similaire identifié, DIA recommandera les livres qu'ils ont appréciés.

Sous module similarité du style d'apprentissage

D'après VARK[16], nous considérons les Styles d'Apprentissage (*SA*) suivants : *visuel* (*V*), *auditif* (*A*), *écriture/lecture* (*R*) et *kinesthésique* (*K*) et *multimodal*. Entre 50% et 70% de toute population appartient à cette dernière catégorie[17]. Par exemple, une personne qui a à la fois le style visuel et le style auditif (*VA*) dominants ou même qui peut partager tous les styles équitablement (*VARK*).

Ce module s'intéresse au calcul de la **similarité du style d'apprentissage** entre deux étudiants. Nous nous sommes inspiré du modèle de [Abdel-Razek 2004], [Abdel-Razek, *et al.* 2004] pour établir la fonction de similarité du style d'apprentissage *LS* entre deux étudiants E_u et E_v. L'équation (4.2-3) définie *LS* comme étant le nombre de styles d'apprentissage dominants commun entre E_u et E_v proportionnel à la somme totale du nombre de leurs styles. Ainsi :

$$LS\left(E_u, E_v\right) = \frac{\left|\{SA_u\} \cap \{SA_v\}\right|}{\left|\{SA_u\} \cup \{SA_v\}\right|} \qquad (4.2\text{-}3)$$

---

[16] © Copyright Version 4.1 (2004) held by Neil D. Fleming, Christchurch, New Zealand and Charles C. Bonwell, Green Mountain Falls, Colorado 80819 U.S.A.

[17] http://www.vark-learn.com/english/index.asp

L'équation (4.2-4) décrit d'une manière plus détaillée la similarité *LS*. Nous avons tronqué cette équation dû aux nombreuses combinaisons possibles. Il n'existe aucune différence entre l'équation (4.2-3) et l'équation (4.2-4), nous voulons montrer une autre manière de décrire cette similarité.

$$LS(E_u, E_v) = \begin{cases} 1 & SA_u = SA_v \\ 1/4 & SA_u \in \{V, A, R, K\} \,\&\, SA_v \in \{VARK\} \\ 1/3 & SA_u \in \{V, A, R, K\} \,\&\, SA_v \in \{VAR, ARK, RKV, VAK\} \\ 1/2 & SA_u \in \{V, A, R, K\} \,\&\, SA_v \in \{VA, VR, VK, AR, AK, RK\} \\ 1/4 & SA_u \in \{VK, AK, RK\} \,\&\, SA_v \in \{VAR\} \\ \ldots & \ldots \\ 0 & sinon \end{cases} \quad (4.2\text{-}4)$$

Le tableau 4.2-1 dresse un exemple concret d'un tel calcul entre 5 étudiants, Frédéric, Samir, Laurent, Karl et Rhéa. Le style d'apprentissage de ces derniers est écrit près de leur nom. D'après ce tableau, nous avons que Frédéric est à la fois visuel et kinesthésique alors que Samir est auditif. Comme ils ne partagent aucun style commun, d'après l'équation (4.2-3) nous avons *LS (Frédéric, Samir)* = 0. D'un autre côté, *LS (Frédéric, Laurent)* = 2/3 vu qu'ils partagent tous les deux les styles visuel et kinesthésique sur les trois qu'ils ont. La similarité entre Frédéric et Karl est de 2/3 aussi vu que Karl est comme Laurent, de type *VAK*. On peut constater que *LS (Frédéric, Rhéa)* = 2/3.

Tableau 4.2-1. Exemple de similarité du style d'apprentissage des étudiants

	Frédéric (VK)	**Samir (A)**	**Laurent (VAK)**	**Karl (VAK)**	**Rhéa (V)**
Frédéric (VK)		0	2/3	2/3	1/2
Samir (A)	0		1/3	1/3	0
Laurent (VAK)	2/3	1/3		1	1/3
Karl (VAK)	2/3	1/3	1		1/3
Rhéa (V)	1/2	0	1/3	1/3	

Il important de noter que la fonction *LS* est *symétrique*, c'est-à-dire que $LS(E_u, E_v) = LS(E_v, E_u)$. De plus, $0 \leq LS(E_u, E_v) \leq 1$.

Similarité des notions appréciées

Comme nous avons vu dans la section 4.2.2.2, DIA permet à l'usager de retourner un feedback sur les livres afin d'indiquer ceux qui sont pertinents. Ce feedback permet à DIA d'extraire les notions d'un domaine qui intéressent le plus chaque étudiant et de les stocker dans le profil. Ce module calcule la similarité des usagers en utilisant comme critère **la similarité des notions préférées** entre les étudiants; ceci permet au système d'identifier ceux qui partagent un intérêt pour les mêmes sujets.

Pour y parvenir, nous utilisons la distance du sens dominant que nous avons déjà vue à la section 4.2.1.5. Cependant, contrairement au module de l'analyse des livres, la similarité n'est pas entre les notions d'un domaine et les notions des livres mais entre les notions qui intéressent deux étudiants. Soit $W^a = \{w_1, \ldots, w_s\}$ l'ensemble des notions préférées de l'étudiant *a* et $W^b = \{w_1, \ldots, w_m\}$ ceux de l'étudiant *b*, la *distance du sens dominant* entre W^a et W^b, définie comme $SN(W^a, W^b)$ est :

$$SN(W^a, W^b) = \frac{1}{s}\sum_{i=1}^{s}\frac{1}{m}\left[\sum_{j=1}^{m}\Theta(w_i^a, w_j^b)\right]$$
$$\text{avec} \quad \Theta(w_i^a, w_j^b) = \begin{cases} 0 & \text{if } w_i^a \neq w_j^b \\ 1 & \text{if } w_i^a = w_j^b \end{cases} \qquad (4.2\text{-}5)$$

Sous module calcul de similarité

Le rôle de ce sous module est d'identifier l'ensemble **des usagers les plus similaires** à l'étudiant actif. Ainsi, ce module prend en entrée le résultat du calcul de la similarité du style d'apprentissage et celui de la similarité des notions appréciées et les utilise pour l'estimation des similarités des étudiants.

Nous nous sommes intéressés sur la manière d'utiliser ces deux similarités afin d'estimer la similarité finale de deux utilisateurs. Nous avons fait face à deux choix. Le

premier étant d'utiliser la similarité des styles d'apprentissage comme coefficient multiplicateur de la similitude des sujets partagés. Cette équation est décrite en (4.2-6).

$$SIM\ (E_a, E_b) = C \times SN\ (W^a, W^b)$$
$$\text{avec}\ \ C = LS(L_a, L_b) \tag{4.2-6}$$

Le coefficient C est croissant, c'est-à-dire que plus le style d'apprentissage de a est similaire a celui de b, plus C sera grand. Il en est de même pour la fonction *SN*. Comme $0 \leq C \leq 1$ et $0 \leq SN \leq 1$, *SIM* est donc proportionnellement grande par rapport à C et *SN* et $0 \leq SIM \leq 1$. Si deux étudiants (E_a et E_b) ont exactement le même style d'apprentissage et sont intéressés par les mêmes notions du domaine, nous aurons *Sim* (E_a, E_b) = 1*1 = 1.

Notre deuxième choix, comme décrit à l'équation (4.2-7), a été de considérer la similitude entre deux étudiants comme la moyenne de leurs deux similarités C et *SN*.

$$SIM\ (E_a, E_b) = \frac{C + SN\ (W^a, W^b)}{2}$$
$$\text{avec}\ \ C = LS(L_a, L_b) \tag{4.2-7}$$

Nous avons décidé d'employer cette dernière équation puisqu'elle présente les mêmes propriétés que l'équation (4.2-6), mais offre, plus d'avantages lors de l'exécution de l'application. Ce point est l'objet d'une discussion au chapitre 5. Nous pouvons facilement affirmer que *SIM* est proportionnellement grande par rapport à C et *SN* et est toujours comprise entre 0 et 1 (vu que $0 \leq C \leq 1$ et $0 \leq SN \leq 1$).

La figure 4.2-7 schématise le calcul de la similarité entre étudiants. Comme Frédéric et Laurent partagent le même style d'apprentissage (*VAK*), nous avons *LS* (*Frédéric, Laurent*) = 1. Étant une similarité des notions qui les intéressent de 0.766, nous avons $SIM(Frédéric, Laurent) = \frac{1 + 0.766}{2} = 0.883$. Pour la similarité entre Frédéric et Rhéa, nous avons $SIM(Frédéric, Rhéa) = \frac{0.333 + 0.788}{2} = 0.561$. D'après ces résultats, nous pouvons affirmer que Frédéric est plus similaire à Laurent qu'à Rhéa.

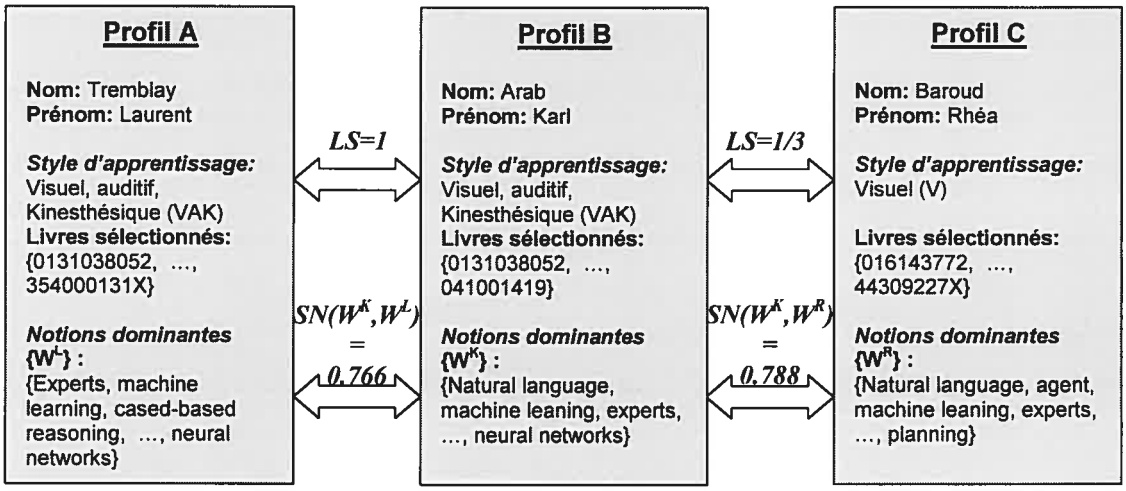


Figure 4.2-7. Le calcul de similarité entre étudiants.

Pour trouver l'ensemble Q_u des K usagers les plus similaires à l'étudiant actif L_u, nous avons introduit l'algorithme (4.2-8). Ainsi, nous calculons la similarité *SIM* entre L_u et tous les autres usagers du système. Ensuite, nous plaçons les K utilisateurs ayants la valeur *SIM* la plus grande dans un ensemble que nous nommerons Q_u.

K most similar Algorithm [Learner L_u]
  for each learner L_v, $L_v \neq L_u$ **compute** $SIM(L_u, L_v)$
  sort the learners in **decreasing** order related to the values of $SIM(L_u, L_v)$ (4.2-8)
  put the K most **similar** learners in a set Q_u

4.2.3.3 Le module d'ordonnancement des livres

Ce module est la dernière étape du processus de filtrage des livres. Une fois que nous avons identifié l'ensemble B_C des livres qui se rapportent le plus au domaine C (voir 4.2.3.1 Le module filtrage par rapport au domaine) et que nous avons construit l'ensemble Q_u des usagers les plus similaires à l'usager actif u (voir 4.2.3.2 Le module filtrage par rapport au modèle de l'apprenant), ce module ordonne les livres de B_C selon les goûts de Q_u. En d'autres mots, ce module applique un filtrage collaboratif pour l'ordonnancement des livres

selon les goûts des étudiants le plus similaires à u. L'ordonnancement se fait selon 3 paramètres en ordre décroissant :

1. le *nombre d'usagers similaires* qui ont aimé le livre;
2. le *nombre total d'usagers* qui ont aimé le livre;
3. la valeur de la *Similarité des Sens Dominants* (*SDD*) entre le livre et le domaine C (Voir 4.2.1.5).

Comme le montre l'algorithme (4.2-9), le premier paramètre à considérer est le *nombre d'usagers similaire*. Si des égalités existent nous regardons le *nombre total d'usagers*. Si des égalités subsistent, nous utilisons le paramètre *SDD*.

```
Book Ranking Algorithm [set of similar users Qu, set of
  filtered books BC]
     for each book h in BC
        get Ph, the list of users that have selected it
     Instantiate TOTALh=| Ph |, SIMILARh=0;
     for each user in Ph
        if user is in Qu
          SIMILARh++;

        Sort the books in decreasing order related to
          the values of SIMILARh. If there are books
          with equal values, give a higher ranking to
          the book with the highest value of TOTALh++.
          If equalities subsist, give a higher ranking
          to books with the hightest value of SDD.
```
(4.2-9)

Supposons que notre ensemble de livres B_C est composé de 4 livres {livre 1, livre 2, livre 3, livre 4} et l'ensemble des étudiants le plus similaire à Frédéric $Q_{Frédéric}$ est {Laurent, Karl, Samir}. Supposons aussi que :

- le livre 1 est apprécié par {Samir, Rhéa, Simon} et *SDD* (*livre 1*, *C*) = *0.877*
  - P_1 = {Samir, Rhéa, Simon}, $TOTAL_1$ = |{Samir, Rhéa, Simon}| = 3, $SIMILAR_1$ = |{Samir}|=1;
- le livre 2 est apprécié par {Samir, Karl} et *SDD* (*livre 2*, *C*) = *0.766*

- P_2 = {Samir, Karl}, $TOTAL_2$ = |{Samir, Karl}| = 2, $SIMILAR_2$ = |{Samir, Karl}| = 2;

- le livre 3 est apprécié par {Samir, Karl, Rhéa} et *SDD (livre 3, C) = 0.883*
  - P_3 = {Samir, Karl, Rhéa}, $TOTAL_3$ = |{Samir, Karl, Rhéa}| = 3, $SIMILAR_3$ = |{Samir, Karl}|= 2;
- le livre 4 est apprécié par {Laurent, Simon, Rhéa} et *SDD (livre 4, C) = 0.910*
  - P_4 = {Laurent, Simon, Rhéa}, $TOTAL_4$ = |{Laurent, Simon, Rhéa}| = 3, $SIMILAR_2$ = |{Laurent}|= 1.

La présentation des livres à l'étudiant *u* se fera sous la forme d'ordonnancement comme présenté par le tableau 4.2-2.

Tableau 4.2-2. Exemple d'ordonnancement des livres selon les paramètres SIMILAR, TOTAL et SDD.

Ordre	Livre	SIMILAR	TOTAL	SSD
1	Livre 3	2	3	0.883
2	Livre 2	2	2	0.766
3	Livre 4	1	3	0.910
4	Livre 1	1	3	0.887

4.3 Les bases de données

Le deuxième tiers de l'architecture de DIA, la base de données, a comme rôle la conservation des données nécessaires au bon fonctionnement du système. Il est composé de quatre tables principales :

- la base des notions dominantes :
  - cette base contient l'information relative au domaine couvert par le système. Ainsi, elle contient sa description telle qu'introduite par l'administrateur du domaine et ses notions importantes (notions dominantes) telles qu'estimées par DIA;

- la base des livres :
  - la base des livres est le lieu de stockage de toutes les données relatives aux livres. Ainsi, tout ce qui se rapporte aux livres comme le titre, les auteurs, le code de localisation à la bibliothèque, le domaine auquel appartient le livre;
- la base de profils des étudiants :
  - la base des profils contient la description et les goûts des étudiants. Le calcule de similarité entre les étudiants est basé entièrement sur ce profil. Ce dernier est construit par le module modélisation de l'apprenant et est constamment mis à jour par un module spécifique à cette tâche. Il contient les données traditionnelles telles que le nom d'usager, le mot de passe, le niveau d'étude, mais aussi le style d'apprentissage de l'étudiant, ses livres appréciés et l'ensemble des notions du domaine qui l'intéressent;
- la base des bibliothèques :
  - cette dernière contient les adresses des serveurs Z39.50 de chaque bibliothèque couverte par les recommandations, ainsi que le nom de leurs bases de données. Un exemple de telles données est disponible en annexe;
- la base des sites de descriptions :
  - elle contient l'adresse Web des sites contenant la description approfondie des livres (comme des librairies en lignes). De plus, elle contient les règles d'extraction de l'information pertinente de leurs pages Web.

4.4 L'interface de l'usager

Ce tiers évoque le côté client. Il est responsable de la présentation des données, de la transmission des événements de l'utilisateur et à l'utilisateur et du contrôle de l'interface de l'usager.

4.5 Conclusion

Dans ce chapitre, nous avons vu l'architecture et la méthodologie de notre système. Nous avons décrit en détail chacun des modules qui le composent, dont voici les points les plus importants :

- DIA adapte l'approche *de filtrage collaboratif pyramidale* [Abdel-Razek 2004], [Abdel-Razek, *et al.* 2004] à sa problématique de génération des recommandations. Nous avons vu que cette approche prend la forme d'une succession de filtrages qui se font selon des critères bien définis :
    - le *premier filtrage* des livres se fait par rapport à leur pertinence au domaine. DIA compare les notions importantes du domaine (calculées à l'aide de la technique du sens dominant) avec la description approfondie de chaque livre. Seuls les livres qui couvrent le plus de notions sont considérés. Nous avons vu aussi comment DIA profite d'un accès à l'Internet pour enrichir la description disponible à la bibliothèque sur un livre à partir des sites de librairies en ligne. Ceci lui permet de mieux analyser le contenu des livres. Ce filtrage est un avantage par rapport aux autres systèmes existants, car il permet au système de générer les premières recommandations sans requérir la participation directe des étudiants. Il contourne ainsi le problème du « *cold start* » (démarrage à froid), un problème assez répandu dans les systèmes de recommandations (voir l'état de l'art 2.2.2.5);
    - le *deuxième filtrage* se fait par rapport à l'usager pour mieux personnaliser la recommandation. Ce filtrage utilise l'approche collaborative pour suggérer les livres appréciés par les étudiants les plus similaires à l'étudiant actif. Pour ce faire, la similarité entre deux étudiants est calculée par rapport à la similitude de leurs styles d'apprentissage et par rapport à la similitude des notions du domaine auxquelles ils s'intéressent.

De plus, nous avons vu que DIA emploie un client Z39.50 lui permettant d'interroger simultanément plusieurs bases de données hétérogènes et réparties :

- sans connaître la structure des bases des données interrogées ;
- à l'aide d'une requête unique ;
- où les données sont représentées dans un format normalisé facilement réutilisable.

Par conséquent, le client Z39.50 permet à DIA de se connecter à n'importe quelle bibliothèque ayant un serveur Z39.50, de consulter la totalité de ses catalogues de livres et de générer des recommandations à partir de toutes ces bibliothèques simultanément. De plus, nous avons vu comment DIA utilise un client HTTP pour chercher des descriptions de livres plus détaillées sur le Web dans le but de permettre une meilleure analyse de livres.

Ce chapitre était donc une présentation assez technique de DIA. Le chapitre suivant introduit un exemple plus concret de l'utilisation du système.

Chapitre 5
DIA, scénarios et exemples réels

Frédéric est un étudiant de l'Université de Montréal, il suit le cours d'intelligence artificielle. Son professeur a voulu offrir un service de recommandation de livres personnalisé et a décidé d'employer DIA. Il a donc enregistré le domaine d'IA. Pour se faire, l'enseignant a fourni au système la description du cours et l'ISBN des deux livres de référence afin de permettre à DIA de faire une analyse du domaine (du contenu du cours). DIA recherche, sur le site Web des libraires électroniques, la description et la table des matières de ces deux livres, puis les utilise avec la description du cours pour identifier les *notions dominantes* du cours. Ensuite, DIA cherche, à l'aide du protocole Z39.50, dans les bases de données de la bibliothèque de l'université, tous les livres qui couvrent ces notions. Il trouve plus de 900 livres. Il est clair que les étudiants, dont notamment Frédéric, n'ont pas les moyens ni le temps de consulter tous ces livres pour trouver ceux qui leurs conviennent le plus.

C'est ici que DIA jouera un rôle primordial, celui d'aider l'étudiant à **identifier rapidement les livres les plus appropriés** en lui suggérant **une liste de livres personnalisée**. DIA considère qu'un livre est convenable pour un étudiant si :

1. le livre est *pertinent pour le cours*, c'est-à-dire, s'il couvre correctement une grande partie des notions du cours suivi;

2. le livre *est intéressant pour l'étudiant* :
    - il couvre les notions du domaine qui intéressent l'étudiant;
    - il aide l'étudiant à la compréhension et l'assimilation de ces notions.

Le système parvient à ces objectifs à l'aide de 3 étapes essentielles. La première étant d'identifier les livres les plus pertinents pour le cours d'IA. Pour cela, DIA recherche sur les sites des librairies en ligne une riche description (résumé, table des matières, etc.) de chacun des 900 livres trouvés à la bibliothèque. Cette description est comparée avec les notions dominantes du cours (extraites précédemment à partir des livres de référence et de la description du cours) pour estimer la similarité des notions dominantes du livre et du cours. Finalement, les 30 livres ayant la plus grande similarité, parmi les 900 identifiés, sont extraits. Pour rendre le processus de recommandation plus rapide, cette étape est effectuée occasionnellement, surtout que les notions d'un cours et le contenu du livre sont statiques et ne changent que rarement. Ce *filtrage basé sur le contenu* est nommé *filtrage par rapport au domaine.* Il faut noter que les 30 livres identifiés sont les mêmes pour tous les étudiants du cours. Ce premier filtrage, étant indépendant de l'étudiant, consistait à l'identification des livres de la bibliothèque les plus pertinents par rapport au cours.

La deuxième étape consiste à trouver, parmi ces 30 livres, ceux qui sont les plus susceptibles de plaire et d'être utiles à Frédéric spécifiquement. Comme ce sont des critères difficilement calculables par la machine, DIA utilise le *filtrage collaboratif* et fait appel à l'expérience des autres étudiants pour la génération de telles recommandations. DIA commence par identifier l'ensemble des étudiants les plus similaires à Frédéric pour ensuite suggérer les livres qu'ils ont préférés. Le système emploie une fonction de similarité *SIM* composée de deux variables pour le calcul de la similitude entre Frédéric et les autres utilisateurs du système : *LS*, la *similarité de leurs styles d'apprentissage* et *SN*, la *similarité des notions qui les intéressent*. Nous étions confrontés à deux alternatives pour le calcul de *SIM*. Ces dernières sont décrites en (5-1) et (5-2). Dans l'une, *LS* est un coefficient multiplicateur de *SN* et alors que dans l'autre *SIM* est la moyenne des 2 similarités *LS* et *SN*. Finalement, notre préférence a été d'appliquer (5-2), vu les avantages que cette

équation nous rapporte. Dans ce chapitre, nous étudierons ces avantages en parcourant tous les scénarios auxquels un étudiant, comme Frédéric, peut faire face.

$$SIM\ (E_a, E_b) = LS(L_a, L_b) \times SN\ (W^a, W^b) \tag{5-1}$$

$$SIM\ (E_a, E_b) = \frac{LS(L_a, L_b) + SN\ (W^a, W^b)}{2} \tag{5-2}$$

5.1 L'ensemble des étudiants similaires

Lors de sa première inscription au système, DIA pose une série de 13 questions à Frédéric afin de déterminer son style d'apprentissage. Le système analyse les réponses de l'étudiant et trouve qu'il est auditif (*A*). Si Frédéric est déjà inscrit au système, il n'a qu'à ouvrir une session à l'aide de son nom d'usager et son mot de passe. DIA essaie ensuite de déterminer les étudiants qui lui sont le plus similaires pour lui suggérer les livres qu'ils ont appréciés. Quatre situations sont possibles :

1. Frédéric ne partage aucune ressemblance avec les autres étudiants, c'est-à-dire que son style d'apprentissage **et** les sujets qui l'intéressent sont uniques.
2. Le style d'apprentissage de Frédéric est unique.
3. Les sujets qui intéressent Frédéric sont uniques.
4. Frédéric partage un style d'apprentissage et des sujets communs avec les autres usagers (*cas régulier*).

5.1.1 Cas 1 : un style d'apprentissage et des sujets uniques

Nous savons que Frédéric a un style d'apprentissage « auditif ». Cependant, personne dans le système ne partage cette même caractéristique. De plus, il est le seul qui s'intéresse à certaines notions du cours pour lesquelles il recherche des livres. Cette situation est peu

probable vu que les styles d'apprentissage sont assez bien repartis dans une population[18], mais peut comme même survenir, notamment si Frédéric est le premier usager du système.

Calcul de similarité et l'ensemble des étudiants les plus similaires à Frédéric

Décrivons cette situation sous une forme formelle. Nous avons :

- $LS(Frédéric, X) = 0 \quad \forall X \in \{\text{étudiants du cours}\}$
- $SN(W^{Frédéric}, W^{X}) \neq 0 \quad \forall X \in \{\text{étudiants du cours}\}$

L'équation (5-1) nous donne :

$$SIM\ (E_a, E_b) = 0 \times 0 = 0$$

Avec l'équation (5-2), nous avons :

$$SIM\ (E_a, E_b) = \frac{0+0}{2} = 0$$

En d'autres mots, si le style d'apprentissage de Frédéric et les sujets qui l'intéressent sont uniques, sa similarité avec tous les étudiants est nulle. Dans ce cas, les deux équations (5-1) et (5-2) nous donnent le même résultat et ne permettent pas la génération de l'ensemble des étudiants les plus similaires à Frédéric. Cependant, la manière dont *DIA* effectue son filtrage de livre lui permet de présenter quand même à l'étudiant une liste de suggestions pertinentes. Ce point sera discuté dans la section 5.1.5.

5.1.2 Cas 2 : un style d'apprentissage unique

Frédéric est auditif. Cependant, aucun autre étudiant de son cours inscrit au système n'a ce style d'apprentissage. Mais, contrairement au cas 1, lui et quelques autres usagers s'intéressent à des sujets communs.

---

[18] http://www.vark-learn.com/english/index.asp

Calcul de similarité et l'ensemble des étudiants les plus similaires à Frédéric

Plus formellement, nous avons :

- $LS(Frédéric, X) = 0 \quad \forall X \in \{\text{étudiants du cours}\}$;
- $\exists X \in \{\text{étudiants du cours}\} \mid SN(W^{Frédéric}, W^{X}) \neq 0$

Comme, d'après l'équation (5-1), la fonction *LS* est un multiplicateur et comme *LS* (*Frédéric*, *X*) est toujours égale à 0 quelque soit l'étudiant *X* (Frédéric a un style d'apprentissage unique), nous avons :

$$SIM\,(Frédéric\ , X) = 0 \times SN(W^{Frédéric}, W^{X}) = 0$$

Ainsi, la similarité entre Frédéric et tout autre utilisateur est égale à 0. Dans ce cas, cette équation a le désavantage de ne pas tenir compte de la similitude des notions appréciées par les utilisateurs. Si par exemple Samir s'intéresse aux mêmes sujets que ceux de Frédéric nous aurons $SN(W^{Frédéric}, W^{Samir}) = 1$. Mais la similarité totale (*SIM*) entre ces deux étudiants est quand même de 0. Cette équation n'est donc pas représentative dans ce cas et ne permet pas au système la construction de l'ensemble des étudiants les plus similaires à Frédéric.

Cependant, pour l'équation (5-2), nous avons :

$$SIM\,(Frédéric\ , X) = \frac{0 + SN(W^{Frédéric}, W^{X})}{2} = \frac{SN(W^{Frédéric}, W^{X})}{2}$$

La similarité entre Frédéric et les autres étudiants inscrits au système est basée sur la similarité des notions (*SN*) du cours qui les intéressent le plus. Bien que la fonction *SN* est divisée par la constante 2 (car on cherche la moyenne des deux similarités), l'ordre des étudiants les plus similaires à l'étudiant actif ne change pas, car cette constante est la même pour tous les étudiants. Contrairement à l'équation précédente, celle-ci nous permet la construction de l'ensemble des usagers les plus similaires à Frédéric, formé des étudiants s'intéressant aux plus grands nombres de sujets communs avec ce dernier.

5.1.3 Cas 3 : un utilisateur qui s'intéresse à des sujets non populaires

Frédéric a un style d'apprentissage assez populaire parmi ses camarades de classe, mais ils ne s'intéressent pas à des sujets communs. Ce cas peut survenir quand c'est la première connexion de l'étudiant au système et qu'il n'a pas eu l'occasion d'indiquer les livres qu'il apprécie. Or, DIA extrait W, l'ensemble des sujets qui intéressent l'étudiant à partir des livres sélectionnés; donc $W^{Frédéric} = \{\emptyset\}$.

Calcule de similarité et l'ensemble des étudiants les plus similaires à Frédéric

Transformons ce cas sous une forme mathématique. Nous avons :

- $SN(W^{Frédéric}, W^{X}) = 0 \quad \forall X \in \{\text{étudiants du cours}\}$;
- $\exists X \in \{\text{étudiants du cours}\} \mid LS(Frédéric, X) \neq 0$

Ce cas ressemble à la situation 2, mais c'est la fonction SN qui est toujours égale à zéro. D'après l'équation (5-1) nous avons :

$$SIM\ (Frédéric,X\ ) = LS(Frédéri\ \ c,X) \times 0 = 0$$

En d'autres mots, la similarité entre l'étudiant actif et les autres utilisateurs est toujours égale à zéro, même lorsqu'ils partagent un style d'apprentissage commun. Contrairement à cette équation qui ne permet pas la génération de l'ensemble des étudiants les plus similaires, l'équation (5-2) donne :

$$SIM(Frédéric,X) = \frac{LS(Frédéric,X)+0}{2} = \frac{LS(Frédéric,X)}{2}$$

On peut donc estimer la similarité entre deux étudiants à partir de la similarité de leur style d'apprentissage. Comme c'est le rang des étudiants qui nous intéresse, la division par 2 ne change pas le résultat et l'ensemble des étudiants les plus similaires à Frédéric est composé de ceux qui ont le style d'apprentissage le plus similaire au sien.

5.1.4 Cas 4 : le cas régulier

Bien que les 3 cas précédents puissent survenir, la situation la plus probable est l'existence d'étudiants ayant un style d'apprentissage similaire à celui de Frédéric et ayant des notions communes qui les intéressent, d'où le nom cas régulier.

Calcul de similarité et l'ensemble des étudiants les plus similaires à Frédéric

Pour être dans cette situation, il faut que :

- $\exists X \in \{\text{étudiants du cours}\} \mid LS(Frédéric, X) = A$ et $SN(W^{Frédéric}, W^{X})$ avec $0 < A \leq 1$ et $0 < B \leq 1$

D'après l'équation (5-1) nous avons :

$$SIM(Frédéric, X) = A \times B$$

Et l'équation (5-2) nous donne :

$$SIM(Frédéric, X) = \frac{A + B}{2}$$

Dans un cas régulier, c'est-à-dire lorsqu'il existe des étudiants X, où $LS\ (Frédéric,\ X)$ et $SN\ (W^{Frédéric},\ W^{X})$ sont différents de zéro, l'équation (5-1) et (5-2) donne les mêmes résultats. En effet, nous cherchons l'ensemble des étudiants ayant le plus de similarité avec Frédéric, donc la valeur numérique absolue retournée par ces deux équations n'importe pas à la problématique, c'est l'ordre de ces valeurs qui est important. Bien que ces deux équations ne retournent pas la même valeur pour un usager X, elles retournent toujours le même ensemble d'étudiants vu que, dans les deux fonctions, plus cet usager est similaire à Frédéric, plus la valeur de *SIM* est grande. Dans un cas régulier, l'équation (5-1) et (5-2) retournent le même ensemble ordonné d'usagers lorsque ces derniers sont triés par rapport à la valeur *SIM* qui leur est associée.

En effet, soit a et b deux variables qui représentent respectivement les fonctions *LS* et *SN*. Nous savons que :

- plus la similarité du style d'apprentissage, entre deux usagers, est grande, plus a sera grand (caractéristique de la fonction *LS*);
- plus la similarité entre leurs sujets appréciés est grande, plus b sera grand. (Caractéristique de la fonction *SN*).

Si entre un étudiant actif (étudiant a) et :

- *l'étudiant 1*, nous avons : $a = A$ et $b = B$ avec $0 < A \leq 1$ et $0 < B \leq 1$
- *l'étudiant 2*, nous avons :
  $a = A + \delta$ et $b = B + \lambda$ avec $0 \leq \delta, 0 < A + \delta \leq 1, 0 \leq \lambda$ et $0 < B + \lambda \leq 1$
- *l'étudiant 3*, nous avons :
  $a = A - \alpha$ et $b = B - \beta$ avec $0 \leq \alpha, 0 < A - \alpha \leq 1, 0 \leq \beta$ et $0 < B - \beta \leq 1$

comme l'équation (5-1) est de la forme $z = a * b$ et est **monotone et croissante** sur l'intervalle]0,1] alors

$$\begin{gathered}(A-\alpha)*(B-\beta) \leq A*B \leq (A+\delta)*(B+\lambda) \\ \Leftrightarrow \\ SIM(\text{étudiant a}, \text{étudiant 3}) \leq SIM(\text{étudiant a}, \text{étudiant 1}) \leq SIM(\text{étudiant a}, \text{étudiant 2})\end{gathered} \tag{5.1-1}$$

Or, l'équation (5-2) qui est de la forme $z = \frac{a+b}{2}$ présente les mêmes caractéristiques de monotonie et de croissance, d'où :

$$\begin{gathered}\frac{(A-\alpha)+(B-\beta)}{2} \leq \frac{A+B}{2} \leq \frac{(A+\delta)+(B+\lambda)}{2} \\ \Leftrightarrow \\ SIM(\text{étudiant a}, \text{étudiant 3}) \leq SIM(\text{étudiant a}, \text{étudiant 1}) \leq SIM(\text{étudiant a}, \text{étudiant 2})\end{gathered} \tag{5.1-2}$$

Comme DIA recherche les étudiants les plus similaires à l'étudiant actif, il s'intéresse à l'ordre des valeurs de *SIM* et non pas aux valeurs mêmes. Chaque étudiant étant associé à la fonction *SIM*, l'ordre des étudiants par rapport à leurs similitudes avec l'étudiant actif est le même dans les deux équations (5.1-1) et (5.1-2)

$$\text{étudiant } 3 \preccurlyeq \text{étudiant } 1 \preccurlyeq \text{étudiant } 2$$

5.1.5 Discussion

Dans un cas régulier, les deux équations (5-1) et (5-2) nous donnent le même résultat. Mais, nous avons privilégié l'équation (5-2), car dans le cas 2 et 3, elle permet le calcul de l'ensemble d'étudiants le plus similaire à l'étudiant actif alors que l'autre équation ne le permet pas (étant donné que dans (5-1) tous les étudiants auront une similarité de 0).

5.2 L'ordonnancement des livres

Cette opération consiste à ordonner les 30 livres extraits plus tôt, selon les préférences des étudiants les plus similaires à Frédéric. Cet ordonnancement est donc différent pour chaque usager.

Lors du calcul de l'ensemble d'étudiants les plus similaires, deux résultats sont possibles (voir section 5.1). Le premier survient si Frédéric est dans le cas 1, DIA ne peut générer alors l'ensemble des étudiants qui lui sont similaire, vu qu'il a des préférences particulières et uniques et dans les cas 2, 3 et 4, le système réussit à identifier cet ensemble.

5.2.1 Cas où l'ensemble des étudiants similaires est vide

Frédéric a un profil singulier parmi les usagers du système et il n'existe pas d'étudiants qui présentent des similitudes avec lui. Ceci peut survenir lorsque Frédéric est le premier usager du système. Ce problème, appelé « cold start » est assez courant dans le filtrage collaboratif. Mais DIA, contrairement à beaucoup de systèmes de recommandation, résout

cette faiblesse en transformant le problème du filtrage collaboratif en problème de filtrage basé sur le contenu.

En effet, DIA a accès à la description du cours et à celles des livres de référence. De plus, durant le processus hors ligne, DIA a identifié les 30 livres les plus « proches » ou les plus relatifs au cours à l'aide de la métrique *SSD*, qui mesure la *Similarité entre des notions dominantes* du livre et ceux du cours. Ainsi, plus la *SSD* du livre se rapproche de 1, plus ce dernier couvre des notions importantes du cours et plus son rang sera élevé pour la recommandation. Le tableau 5.2-1 montre une liste ordonnée des livres qui seront suggérés à Frédéric, même s'il a un profil singulier.

Tableau 5.2-1. Ordonnancement par rapport à **SSD**, la *Similarité des Notions Dominantes* du livre avec ceux du domaine (le cours)

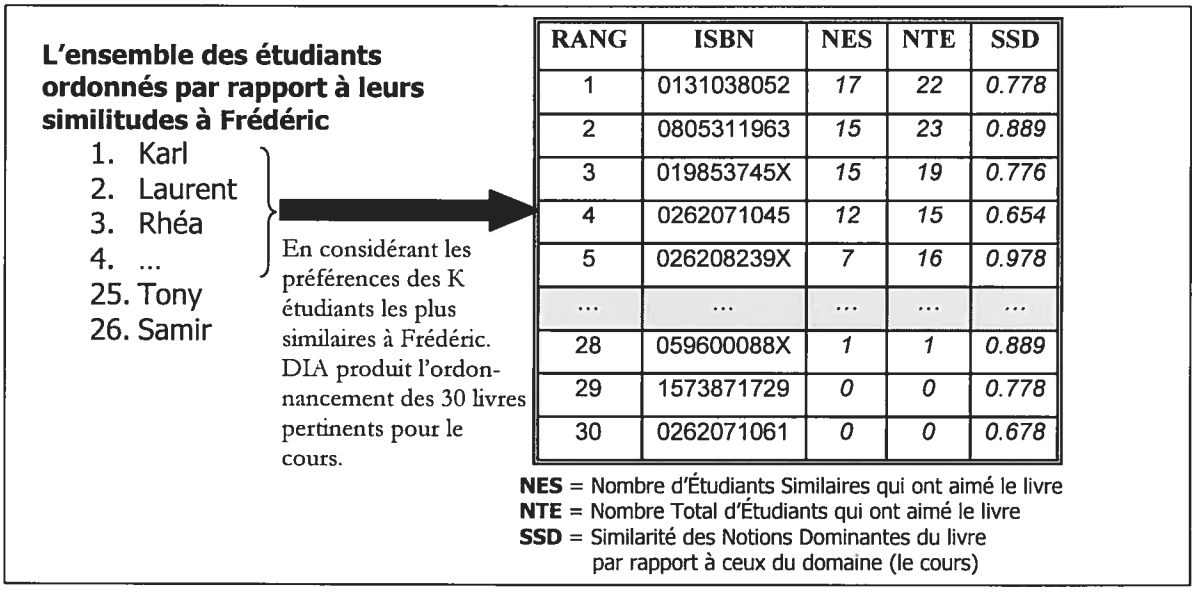
RANG	ISBN	SSD
1	026208239X	*0.978*
2	0805311963	*0.889*
3	059600088X	*0.889*
4	1573871729	*0.778*
5	0131038052	*0.778*
...	...	...
28	019853745X	*0.776*
29	0262071061	*0.678*
30	0262071045	*0.654*

DIA peut donc recommander des livres pertinents, en se basant sur le contenu des livres, même si l'usager n'a pas d'étudiants qui lui sont similaires.

5.2.2 Cas où l'ensemble des étudiants similaires est non vide

Si DIA réussit à déceler l'ensemble des usagers les plus similaires, il utilisera les préférences de ces derniers pour ordonner les livres. Ainsi, plus un livre est apprécié par

l'ensemble des étudiants similaires, plus son rang sera élevé. La figure 5.2-1 montre que le livre dont l'ISBN est 0131038052 a été placé au premier rang vu qu'il a été le plus apprécié parmi les étudiants les plus similaires à Frédéric. Il a été apprécié par 17 étudiants. Il se peut que DIA trouve deux livres ayant été appréciés par le même nombre d'étudiants similaires. Dans ce cas, DIA regarde comme deuxième critère le nombre total d'appréciations. Pour mieux comprendre, regardons le cas des livres ayant comme ISBN = 0805311963 et ISBN = 019853745X. Tous les deux ont été appréciés par 15 étudiants similaires à Frédéric, mais DIA a donné un rang plus élevé au livre apprécié par un plus grand nombre total d'étudiants (similaires et non similaires). Si des égalités surgissent toujours, c'est la similarité des notions dominantes du livre par rapport à ceux du cours (*SSD*) qui fera la différence, comme les livres au 29^{e} et au 30^{e} rang (figure 5.2-1).



RANG	ISBN	NES	NTE	SSD
1	0131038052	*17*	*22*	*0.778*
2	0805311963	*15*	*23*	*0.889*
3	019853745X	*15*	*19*	*0.776*
4	0262071045	*12*	*15*	*0.654*
5	026208239X	*7*	*16*	*0.978*
...	...	...	...	...
28	059600088X	*1*	*1*	*0.889*
29	1573871729	*0*	*0*	*0.778*
30	0262071061	*0*	*0*	*0.678*

Figure 5.2-1. Ordonnancement des livres par rapport aux préférences des usagers les plus similaires

5.3 Conclusion

Cette partie a permis de détailler tous les scénarios possibles pour un étudiant utilisant DIA. De plus, en discutant ces scénarios, nous avons vu les avantages de l'utilisation de l'équation (5-2) par rapport à celle du (5-1).

Finalement, nous avons discuté de l'avantage que présente DIA, celui d'avoir la capacité de recommander des livres à un étudiant, même si ce dernier a des goûts uniques ou si c'est sa première connexion au système.

Chapitre 6
Implémentation et évaluation

Ce chapitre décrit le design de DIA, les technologies employées, le processus d'évaluation du système et les résultats obtenus. Dans un premier temps, nous décrivons les technologies utilisées pour l'implémentation et nous présentons quelques captures d'écran de l'interface de DIA. Ensuite, nous analysons et discutons des résultats de l'évaluation.

6.1 L'implémentation

DIA a été réalisé en Java et utilise plusieurs technologies : Java 2 Platform, Standard Edition (J2SE v1.4.2_04), Java API for XML Processing (JAXP), JavaServer Pages (JSP), Java servlet technology, Java Database Connectivity (JDBC) et XML. De plus, nous avons fait appel à JAFER Toolkit(Java Access For Electronic Resources) [Corfield, *et al.* 2002b, 2002a], une implémentation Java du protocole Z39.50. Le tout a été réalisé sur un environnement Windows NT. La figure 6.1-1 présente ces technologies.

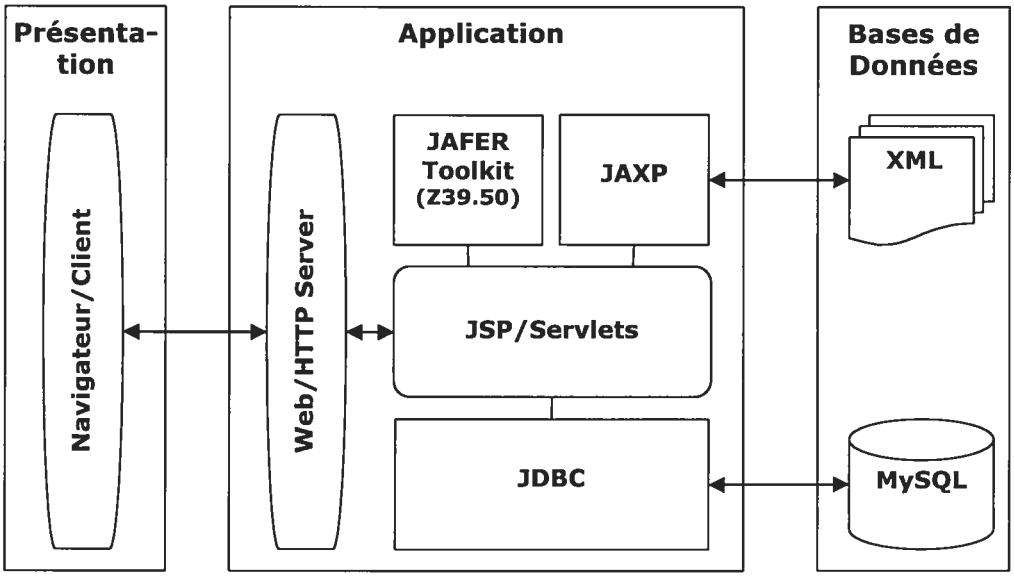


Figure 6.1-1. Les technologies utilisées dans DIA

6.1.1 Présentation

La partie présentation évoque le côté client. Elle est responsable de la présentation des données, de la transmission des événements de l'utilisateur et à l'utilisateur et du contrôle de l'interface de l'usager. Nous avons opté pour une interface Web car elle permet de faciliter l'accès au système (l'usager n'a besoin que d'un navigateur Web) tout en offrant une grande flexibilité pour la présentation. Nous avons donc employé de l'*HTML* et du *Javascript* pour les pages Web. La figure 6.1-2 montre l'interface Web de l'administrateur du domaine. Cette dernière est utilisée pour décrire le domaine pour lequel l'administrateur cherche à générer des recommandations.

Figure 6.1-2. L'interface de l'administrateur du domaine

La figure 6.1-3 donne un exemple d'une recommandation de livres effectuée pour un étudiant. Elle prend la forme d'une liste de livres ordonnés selon leurs pertinences par rapport au profil de l'apprenant. De plus, sur cette interface, nous retrouvons un aperçu des descriptions détaillées de chaque livre afin d'aider l'étudiant à faire son choix. Cette description est extraite automatiquement par DIA à partir de l'Internet (voir 4.2.1.2).

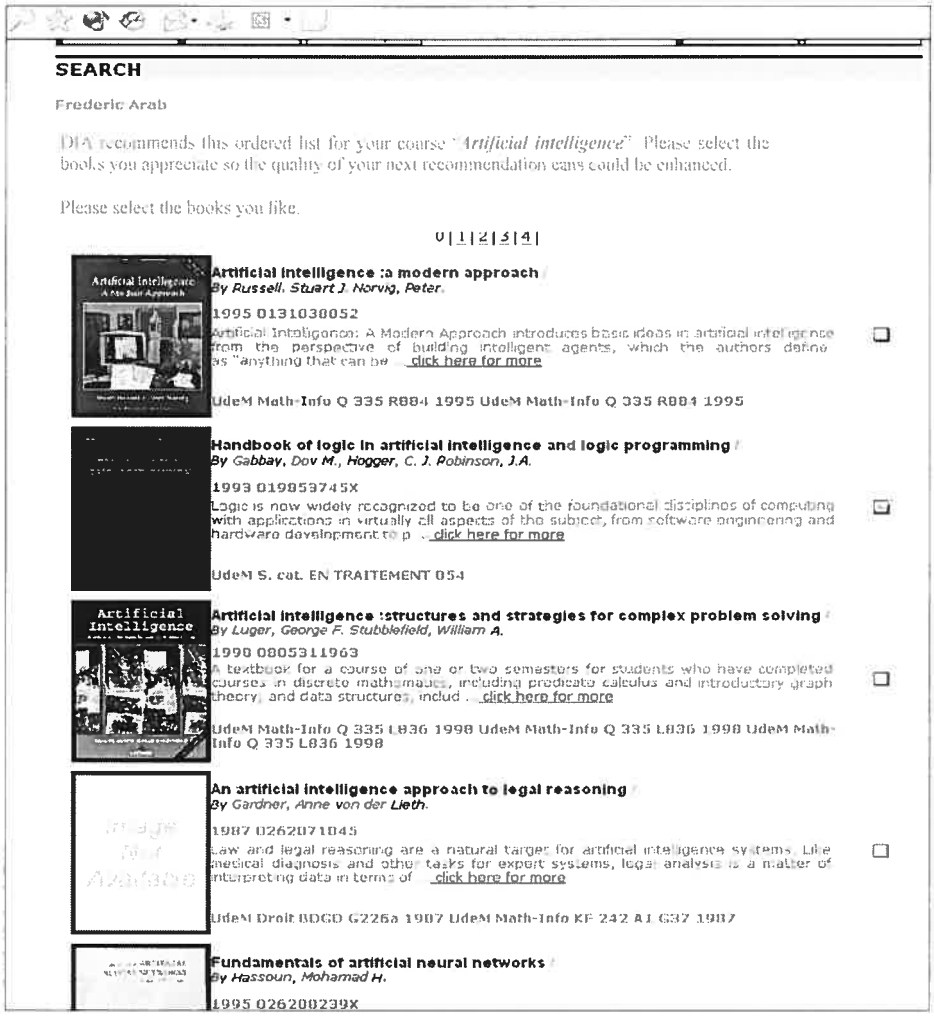


Figure 6.1-3. Recommandation : liste de livres suggérés à l'étudiant

En cliquant sur le lien « click here for more », l'étudiant accède à une description plus complète des livres. La figure 6.1-4 en est un exemple.

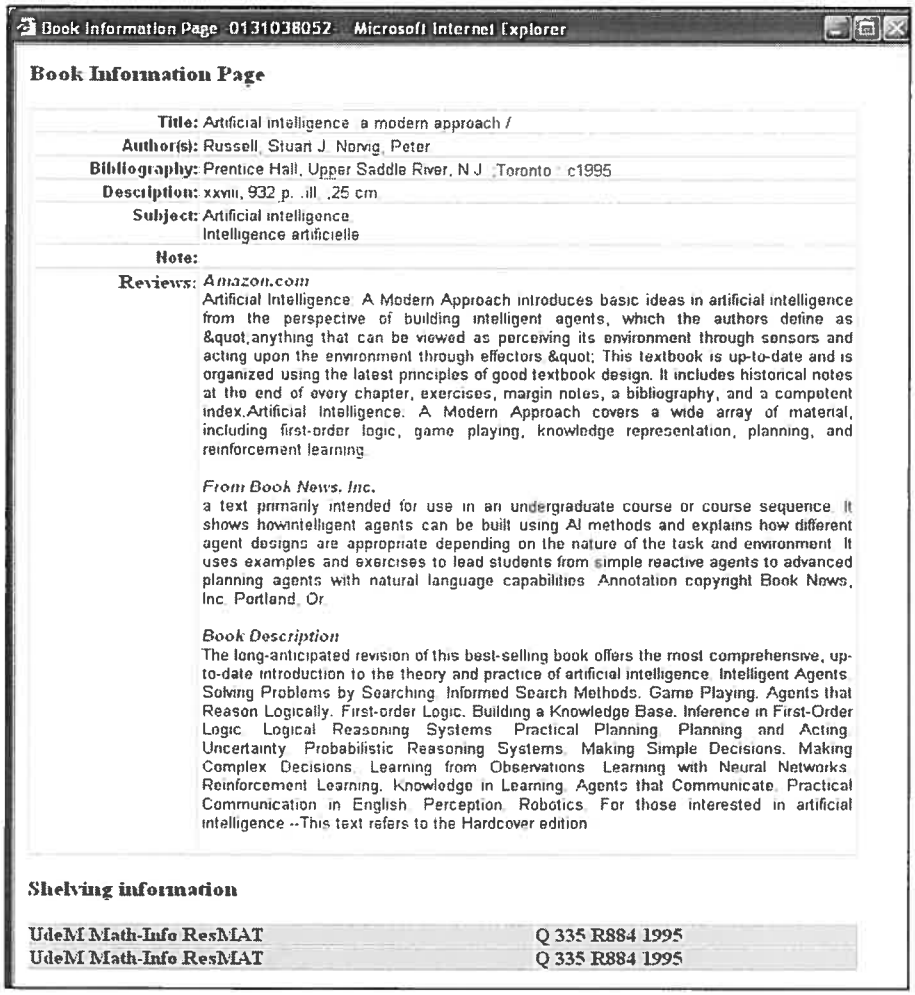


Figure 6.1-4. La description du livre telle que présentée à l'usager

6.1.2 L'application

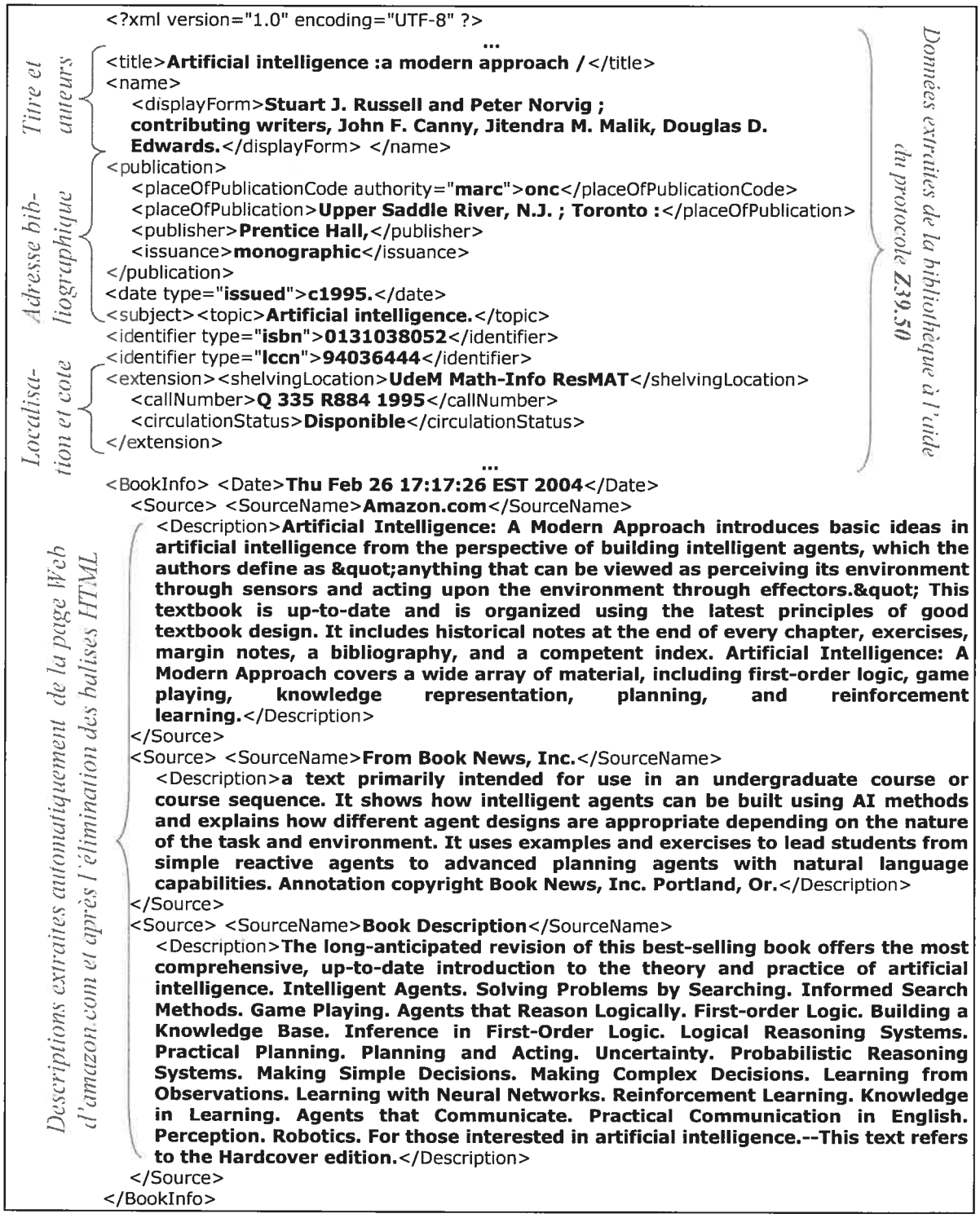
Le fait d'avoir choisi de présenter DIA par une interface Web requiert un serveur Web où la majeure partie des calculs est effectuée. Nous avons utilisé un serveur *Apache Jakarta Tomcat v4.1.30*, la technologie servlets de Java et JSP. De plus, nous avons employé l'API Java pour XML (JAXP) afin que les servlets de l'application puissent gérer, analyser et transformer les fichiers XML de la base de données. Parallèlement, JDBC a assuré la connexion à une base de données MySQL.

Sur un autre plan, nous avons utilisé JAFER Toolkit[19] [Corfield, *et al.* 2002b, 2002a], un outil qui vise à simplifier l'implémentation du standard Z39.50 en permettant d'établir des portails et des émetteurs d'informations sans devoir traiter les complexités techniques de ce protocole.

6.1.3 La base de données

Il y a deux types de bases de données utilisées dans DIA : XML et MySQL. XML est employé pour sauvegarder les données liées aux descriptions des livres (figure 6.1-5), les notions importantes d'un domaine. La base de données MySQL, quant à elle, contient le profil des étudiants et les adresses des serveurs Z39.50 des bibliothèques. Le « DOM parseur », tel que défini dans JAPX, permet la manipulation des fichiers XML en transformant l'arbre XML en une structure de donnée, MySQL est accessible via JDBC avec des requêtes SQL.

[19] http://www.jafer.org/




```
<?xml version="1.0" encoding="UTF-8" ?>
                                    ...
<title>Artificial intelligence :a modern approach /</title>
<name>
   <displayForm>Stuart J. Russell and Peter Norvig ;
   contributing writers, John F. Canny, Jitendra M. Malik, Douglas D.
   Edwards.</displayForm> </name>
<publication>
   <placeOfPublicationCode authority="marc">onc</placeOfPublicationCode>
   <placeOfPublication>Upper Saddle River, N.J. ; Toronto :</placeOfPublication>
   <publisher>Prentice Hall,</publisher>
   <issuance>monographic</issuance>
</publication>
<date type="issued">c1995.</date>
<subject><topic>Artificial intelligence.</topic>
<identifier type="isbn">0131038052</identifier>
<identifier type="lccn">94036444</identifier>
<extension><shelvingLocation>UdeM Math-Info ResMAT</shelvingLocation>
   <callNumber>Q 335 R884 1995</callNumber>
   <circulationStatus>Disponible</circulationStatus>
</extension>
                                    ...
<BookInfo> <Date>Thu Feb 26 17:17:26 EST 2004</Date>
   <Source> <SourceName>Amazon.com</SourceName>
      <Description>Artificial Intelligence: A Modern Approach introduces basic ideas in
      artificial intelligence from the perspective of building intelligent agents, which the
      authors define as &amp;quot;anything that can be viewed as perceiving its environment
      through sensors and acting upon the environment through effectors.&amp;quot; This
      textbook is up-to-date and is organized using the latest principles of good
      textbook design. It includes historical notes at the end of every chapter, exercises,
      margin notes, a bibliography, and a competent index. Artificial Intelligence: A
      Modern Approach covers a wide array of material, including first-order logic, game
      playing, knowledge representation, planning, and reinforcement
      learning.</Description>
   </Source>
   <Source> <SourceName>From Book News, Inc.</SourceName>
      <Description>a text primarily intended for use in an undergraduate course or
      course sequence. It shows how intelligent agents can be built using AI methods
      and explains how different agent designs are appropriate depending on the nature
      of the task and environment. It uses examples and exercises to lead students from
      simple reactive agents to advanced planning agents with natural language
      capabilities. Annotation copyright Book News, Inc. Portland, Or.</Description>
   </Source>
   <Source> <SourceName>Book Description</SourceName>
      <Description>The long-anticipated revision of this best-selling book offers the most
      comprehensive, up-to-date introduction to the theory and practice of artificial
      intelligence. Intelligent Agents. Solving Problems by Searching. Informed Search
      Methods. Game Playing. Agents that Reason Logically. First-order Logic. Building a
      Knowledge Base. Inference in First-Order Logic. Logical Reasoning Systems.
      Practical Planning. Planning and Acting. Uncertainty. Probabilistic Reasoning
      Systems. Making Simple Decisions. Making Complex Decisions. Learning from
      Observations. Learning with Neural Networks. Reinforcement Learning. Knowledge
      in Learning. Agents that Communicate. Practical Communication in English.
      Perception. Robotics. For those interested in artificial intelligence.--This text refers
      to the Hardcover edition.</Description>
   </Source>
</BookInfo>
```

Figure 6.1-5. Un fichier XML qui encapsule la description d'un livre

6.2 L'évaluation

Pour l'évaluation du système, nous avons décidé de couvrir 4 domaines : *artificial intelligence* (intelligence artificielle), *Java programming* (programmation Java), d*ata structure* (structure de données) et *machine learning* (apprentissage machine). Pour chacun de ces domaines, nous avons joué le rôle de l'administrateur en effectuant, à l'aide de DIA, la phase de collecte des données (la phase hors ligne). Durant cette étape, nous avons inscrit les domaines dans le système, en incluant pour chacun d'eux les livres recommandés par les professeurs du domaine. Une fois que nous avons soumis ces données, DIA identifie les notions importantes (*notions dominantes*) de chaque domaine, pour ensuite rechercher les livres de la bibliothèque de l'Université de Montréal qui couvrent ces notions. Par exemple, pour l'*Intelligence Artificielle* DIA a identifié 932 titres, pour la *Programmation Java* 62 titres et 52 titres pour chacun des deux derniers domaines. Ensuite, DIA a recherché le Web pour compléter l'information de tous les titres identifiés.

À partir du moment où cette étape a été achevée, nous avons invité 112 étudiants de l'Université de Montréal à tester le système pour vérifier les recommandations qu'ils reçoivent. Lorsque le test est terminé, l'usager doit répondre à un questionnaire inclus sur le site Web du système. Ce questionnaire est divisé en quatre parties. La première contient des questions d'ordre général, visant à connaître certaines habitudes des participants, par exemple, vérifier si l'étudiant utilise, en général, les livres recommandés par son professeur (livres de référence du cours) et examiner ses habitudes d'emprunts de livres de la bibliothèque. La deuxième partie demande l'avis des étudiants sur le projet que nous avons présenté dans ce mémoire. La troisième partie s'intéresse spécifiquement à la performance de DIA et la dernière partie permet à l'usager d'émettre un commentaire sur le système. De plus, à la fin de chaque test, nous avons discuté avec les participants de leurs expériences et des problèmes rencontrés.

6.2.1 Les résultats

Il faut noter que durant trois semaines, 112 participants, tous cycles confondus, de l'Université de Montréal ont pris part à l'expérimentation. Dans la partie suivante, nous examinons les questions demandées à ces derniers et analysons leurs réponses.

6.2.1.1 Le profil des participants

Nous avons d'abord posé à ces 112 étudiants des questions par rapport à l'usage du processus de recommandation de livres. Les résultats à la figure 6.2-1 suggèrent que, bien que 53% de ces participants demandent peu de suggestions de livres de manière explicite à leurs professeurs ou à leurs proches, une forte majorité d'entre eux (81%) utilise les livres recommandés pour le cours par leurs professeurs.

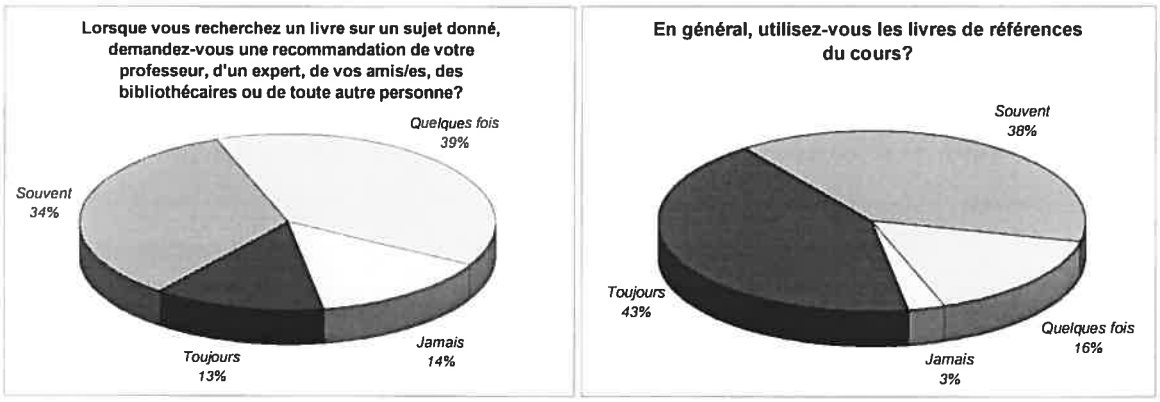


Figure 6.2-1. Graphe : l'utilisation du processus de recommandation de livres

Côté utilisation de la bibliothèque (figure 6.2-2), près de 40% des étudiants disent n'avoir jamais emprunté de livres de leur bibliothèque universitaire et 21% d'entre eux en emprunte souvent. Toujours d'après la même figure, 54% des participants, c'est-à-dire plus d'un étudiant sur deux, trouve une certaine difficulté à choisir un livre de la bibliothèque étant donné le grand nombre de livres disponibles sur un sujet. Ces données suggèrent *l'existence d'un véritable problème « d'infobésité »* ou en d'autres mots, les étudiants ont « trop de ressources » pour savoir quel livre choisir. La description bibliographique retrouvée traditionnellement dans la base de données des bibliothèques, souvent l'unique

description disponible sur un livre, ne facilite point les choix des étudiants. En effet, une bonne majorité (plus de 60%) des utilisateurs ayant testé notre système disent que ces données ne sont pas suffisantes pour les aider à choisir les livres qui leur conviennent le plus (figure 6.2-3).

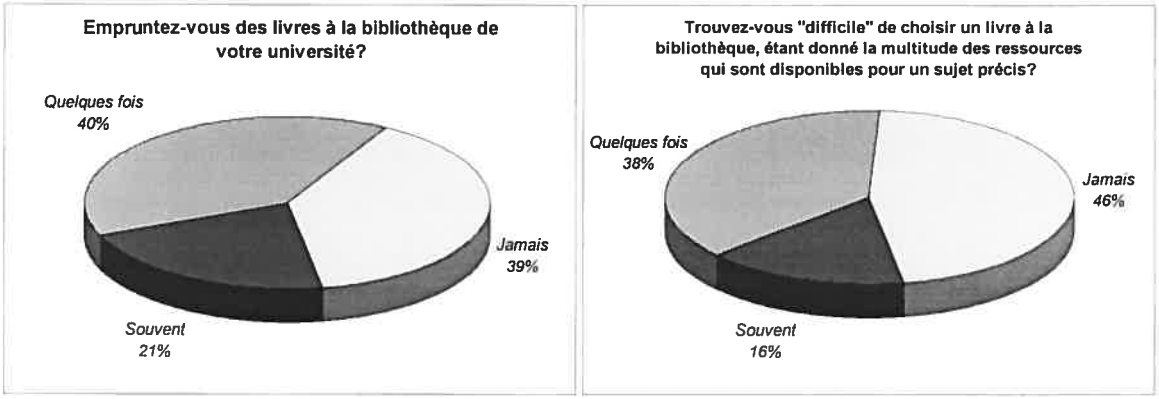


Figure 6.2-2. Graphe : les habitudes d'emprunts de livres de la bibliothèque des étudiants de l'Université de Montréal

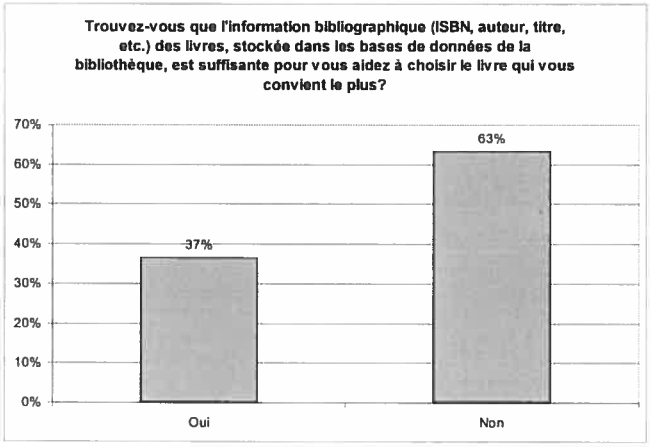


Figure 6.2-3. Graphe : suffisance de descriptions sur les livres

6.2.2 L'évaluation du projet : résultats

Dans cette partie, des questions à choix multiples (« oui », « peut-être », « non ») ont porté sur des aspects généraux du système étant donné que la partie 3 a été dédiée à l'évaluation des fonctionnalités spécifiques de DIA et de certains critères précis. Nous avons demandé aux 112 participants leurs avis sur le projet en général, sa problématique et ses objectifs. Le tableau ci-dessous montre ces questions et dresse leurs résultats.

Tableau 6.2-1. Résultat de la partie 2 du questionnaire

Questions	*Oui*	*Peut-être*	*Non*
Le but de ce projet est-il approprié aux besoins des étudiants?	100%	*nd*	*nd*
Le sujet du projet est-il important?	92%	4%	4%
Le projet adresse-t-il une problématique clairement identifiée?	89%	9%	2%
Le système aide-t-il à l'automatisation du processus de collaboration entre les étudiants?	74%	17%	9%
Le système facilite-t-il la recherche de livres pédagogiques?	78%	16%	6%
Le système peut-il aider les étudiants à identifier les livres qui les intéressent?	76%	13%	11%

Ces résultats sont très positifs. Nous remarquons qu'une forte majorité (9 étudiants sur 10) trouve que le projet est important et qu'il adresse une problématique clairement identifiée. La même proportion trouve qu'il est approprié à leurs besoins. De plus, les trois quarts d'entre eux trouvent que DIA accomplit bien ses objectifs. En d'autres mots, il aide dans l'automatisation de la collaboration entre apprenants, facilite la recherche de livres pédagogiques et aide les étudiants à identifier les livres qui les intéressent.

6.2.3 L'évaluation de DIA : résultats

Cette partie montre les résultats de l'évaluation des aspects précis du système portant sur :

- l'exactitude de l'estimation du style d'apprentissage;
- la pertinence des livres suggérés;

- l'utilité du système;
- la convivialité du système.

Sur la figure 6.2-4, nous remarquons que près de 17% des participants ont été classés comme visuels (V), 12.5% sont bimodales visuels/kinesthésiques (VK) alors que 10.71% ont été identifiés comme visuels/auditifs (VA). Sur cette figure, nous pouvons voir la répartition complète des étudiants selon leur style d'apprentissage tel qu'estimé par le système. En donnant la description de tous les styles d'apprentissage, nous avons demandé aux participants d'indiquer si le système a correctement identifié leurs styles. Environ 67% ont répondu par oui, alors que 12% d'entre eux trouvent que DIA l'a mal identifié.

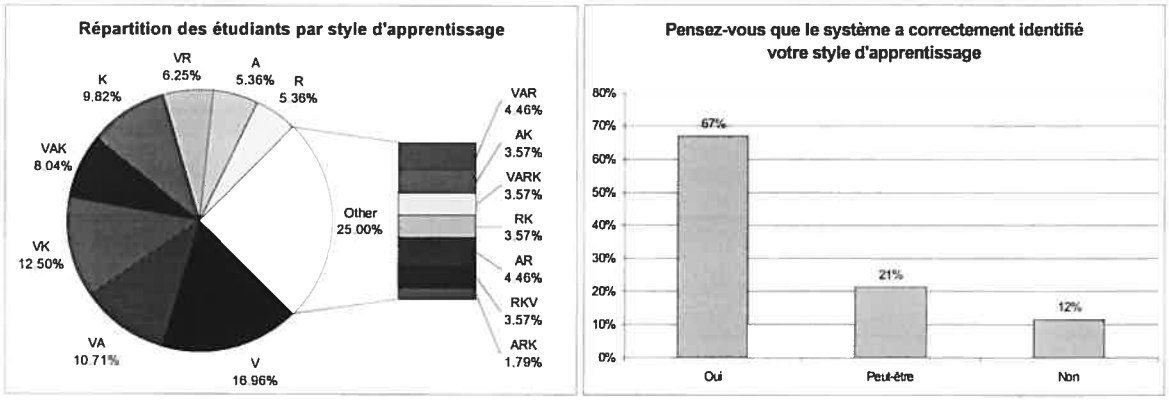


Figure 6.2-4. Graphe : Répartition du style d'apprentissage des étudiants enregistrés au système (V = visuel, A=auditif, K= kinesthésique, R = lecture/écriture)

Ensuite, nous avons proposé une série d'énoncés à l'utilisateur et nous lui avons demandé d'associer une note entre 1 et 10 à chacun d'eux, en considérant 10 comme étant la note maximale qui exprime leur entier accord avec l'affirmation et que 1 représente leur désaccord total.

Sur la figure 6.2-5, on voit que 78% des étudiants ont noté par 7 ou plus la pertinence et la description des livres recommandés et la moyenne des cotes des participants pour ce critère a été de 7.53. Plus de 87% des usagers ont donné une note

supérieure ou égale à 7 pour l'utilité du système. Près d'un étudiant sur trois a été entièrement d'accord (une note de 10) sur le fait que le système est utile aux étudiants.

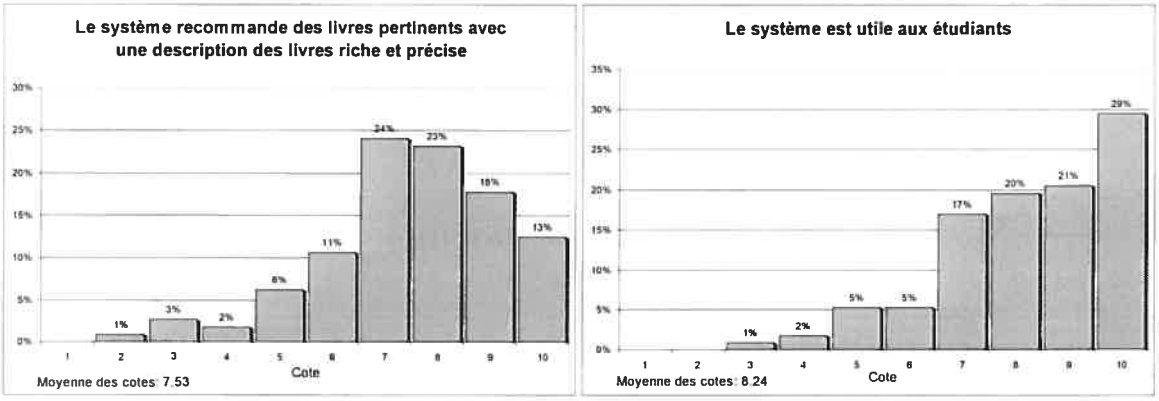


Figure 6.2-5. Graphe : la pertinence et l'utilité du système

Pour la convivialité de l'interface, nous nous sommes intéressés à la qualité de la conception des pages Web du système et à la facilité de leur navigation. Là aussi, nous avons eu des résultats très positifs. Sur la figure 6.2-6, nous pouvons voir que la moyenne de l'évaluation pour le critère de la qualité de la conception a été de 8.02 et 85% des participants ont donné une note supérieure ou égale à 7. De plus, 94% de ces derniers ont noté la facilité de navigation par 7 ou plus.

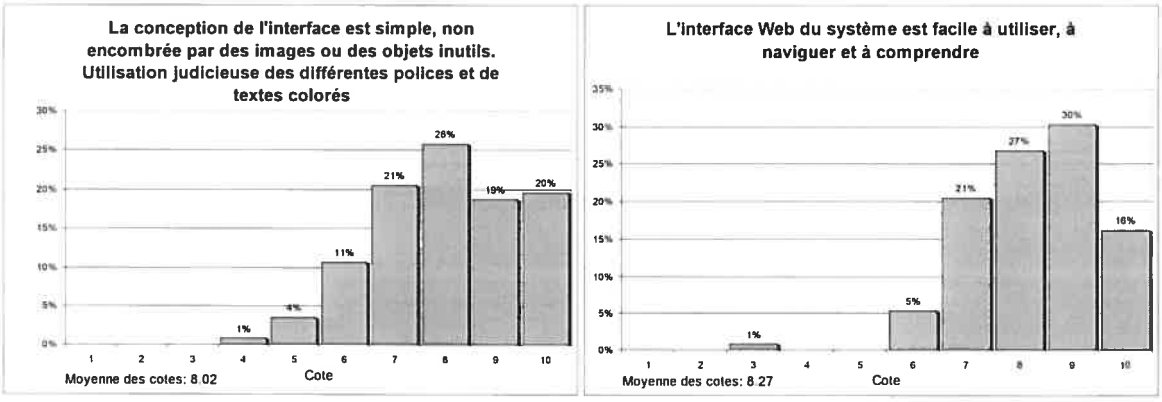


Figure 6.2-6. Graphe : convivialité de l'interface du système

À savoir si les étudiants ont trouvé que le système peut assister la bibliothèque à enrichir et diversifier les services offerts (figure 6.2-7), 87% ont donné une note entre 7 et 10 inclusivement.

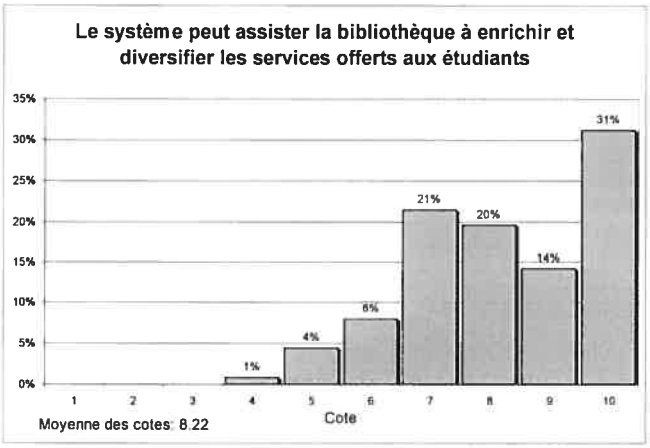


Figure 6.2-7. Graphe : l'utilisation du système à la bibliothèque

Finalement, nous avons demandé aux participants d'attribuer au système une note globale entre 1 et 10. Le résultat, comme nous pouvons le voir sur la figure 6.2-8, a été très positif. La moyenne des notes des 112 participants a été de 8 et 92% d'entre eux ont donné une note supérieure ou égale à 7, avec plus d'un étudiant sur trois ayant adjugé la note de 8.

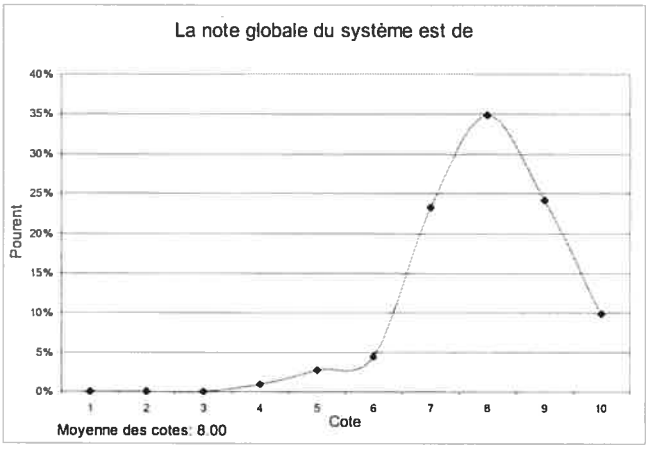


Figure 6.2-8. Graphe : la distribution de la note globale telle qu'évaluée par les étudiants

6.3 Conclusion

Ce chapitre a abordé l'aspect de l'implémentation de DIA. Nous avons discuté des technologies employées pour ensuite nous intéresser à l'évaluation du projet en général et de DIA plus spécifiquement.

Cette évaluation a abouti à des résultats prometteurs. Globalement, elle a montré qu'il existe une certaine concordance, parmi les participants qui ont testé le programme, que ce projet a atteint tous ses objectifs et que DIA réussit à générer des recommandations de livres pédagogiques adéquats. De plus, un consentement favorable a été remarqué nous seulement sur l'efficacité du système, mais aussi sur son utilité pour les étudiants.

Chapitre 7
Conclusion et discussion

Dans le cadre de ce mémoire, nous avons présenté DIA, un système de recommandation de livres pédagogiques, dont le but est de répondre au problème de surcharge d'informations qui touche les étudiants. Dans un questionnaire répondu par 112 étudiants de l'Université de Montréal (tout cycle confondu), 54% des participants, c'est-à-dire plus d'un étudiant sur deux, avoue avoir une certaine difficulté à choisir un livre de la bibliothèque étant donné le grand nombre de livres disponibles sur un sujet. De plus, un phénomène appelé « library anxiety » commence à être de plus en plus observé et étudié, et se retrouve, selon des degrés différents d'intensité, chez la majorité des étudiants. Une des causes de cette anxiété estudiantine serait la grande quantité et diversité de livres et d'informations qui se retrouvent souvent dans les bibliothèques universitaires. DIA est ainsi un outil permettant le filtrage automatisé des livres selon les champs d'intérêt, les préférences et le style d'apprentissage des étudiants.

7.1 DIA : l'approche générale

Nous avons commencé par présenter plusieurs techniques de filtrage étudiées dans la littérature, pour ensuite aborder spécifiquement la recommandation de livres où nous avons présenté quelques systèmes déjà implémentés. Comme, à notre connaissance, aucun système de recommandation de livres existant n'est dédié spécialement à un environnement

pédagogique, mais plutôt à des environnements commerciaux, nous avons étudié les systèmes présentés sous différents angles, dans le but d'examiner leur adaptabilité à un environnement universitaire. Lors de l'analyse de la problématique sous un aspect philosophique, plusieurs points sont ressortis :

- la majorité des systèmes de recommandations de livres sont utilisés dans un contexte commercial (par exemple dans une librairie électronique) comme une technique de marketing ciblé et servent à l'augmentation des ventes électroniques, ce qui soulève un conflit d'intérêts. En effet, étant donné que c'est le site lui-même qui fournit les recommandations sur les articles qu'il désire vendre, certains marchands pourraient être tentés d'utiliser la recommandation personnalisée comme un moyen de se débarrasser de la marchandise excédentaire. Nous pensons, qu'à long terme, ce conflit peut fragiliser l'importance accordée aux besoins des étudiants par rapport aux besoins des commerçants et qu'il porte donc préjudice aux intérêts des consommateurs en général, et des étudiants en particulier;
- de plus, il est évident que les systèmes de recommandation de ces librairies n'offrent aucun contrôle sur certains critères pédagogiques importants, comme par exemple, le fait que le livre couvre la totalité ou une partie des notions d'un cours. Ce contrôle est délégué à l'étudiant lui-même;
- pour parvenir à des recommandations de plus en plus appropriées, l'utilisateur doit interagir, sur une longue période de temps, avec le système. Cela requiert donc de l'utilisateur l'achat de plusieurs livres afin de trouver les livres qui lui conviendront vraiment. Cette solution n'est pas abordable pour un étudiant et donc entrave l'accès aux livres recherchés;
- il est clair que le modèle de l'apprenant n'est pas pris en considération et donc les systèmes actuels ne peuvent répondre à certains besoins spécifiques des étudiants.

Nous nous sommes intéressés aussi à l'aspect pragmatique et technique des systèmes de recommandation en discutant des deux approches de filtrage les plus prédominantes dans la littérature : le filtrage basé sur le contenu et le filtrage collaboratif. Nous avons ainsi dressé

un tableau de leurs avantages et de leurs inconvénients toujours dans un contexte pédagogique.

En nous basant sur les points soulevés plus haut, nous avons proposé DIA, notre système, qui est conçu spécifiquement pour un environnement universitaire. Ainsi, la structure, les techniques et l'approche générale de DIA ont mis les besoins et les intérêts particuliers des étudiants en avant. En particulier, DIA construit sa base de livres à partir de la bibliothèque universitaire, ce qui lui confère plusieurs avantages par rapport aux autres systèmes existants :

- une meilleure cohérence au regard de la problématique, étant donné que les bibliothèques universitaires sont dédiées, par l'établissement scolaire, à l'approvisionnement des étudiants en ressources pédagogiques nécessaires à leurs cheminements;
- l'élimination du conflit d'intérêts qui existe dans les systèmes à but commercial, vu l'absence du caractère lucratif des bibliothèques;
- la complémentarité et la concordance des intérêts entre la bibliothèque et le système de recommandations, entre autres, celles de permettre à l'étudiant un meilleur accès aux livres adéquats;
- la gestion d'un ensemble dynamique de livres et convenant aux programmes suivis par les étudiants. En effet, la bibliothèque est dotée, en général, de spécialistes en bibliothéconomie qui s'assurent, entre autres, de la mise à jour des catalogues par des livres recommandés par les professeurs. Nous considérons aussi que par ce procédé, DIA assure un certain contrôle sur la qualité des livres recommandés;
- un autre avantage est la facilité pour les étudiants d'accéder aux livres, d'un point de vue géographique (la proximité de la bibliothèque) et un point de vue de « l'abordabilité » des livres.

7.2 DIA : l'approche technique

DIA a adapté l'approche du Pyramid Collaborative Filtering Approach (*PCFA*) (*l'approche de filtrage collaboratif pyramidale*) [Abdel-Razek 2004], [Abdel-Razek, *et al.* 2004] à sa problématique de génération des recommandations. L'approche appliquée prend la forme d'une succession de filtrages qui se font selon des critères bien définis.

Le premier filtrage des livres est par rapport à leur pertinence au domaine. DIA compare les notions importantes du domaine (calculées à l'aide de la technique du sens dominant) avec la description approfondie de chaque livre. Seuls les livres qui couvrent le plus de notions sont considérés. Il faut noter que DIA profite d'un accès à l'Internet pour enrichir la description disponible à la bibliothèque sur un livre à partir des sites de libraires en ligne dans le but de mieux analyser son contenu et de représenter une vue plus complète des livres à l'étudiant. Ce premier filtrage basé sur le contenu offre un avantage par rapport aux autres systèmes existants : du fait que la comparaison se fait entre deux instances qui ne dépendent pas de l'étudiant, le système peut générer ses premières recommandations de qualité sans requérir la participation directe des utilisateurs, alors que, la majorité des systèmes de recommandations existants demandent une interaction entre le système et les utilisateurs afin d'étudier leurs préférences avant de générer leurs premières suggestions.

Une fois que DIA a identifié un premier ensemble de livres qui sont liés au domaine, il utilise le filtrage par rapport à l'usager pour mieux personnaliser la recommandation. Ce filtrage utilise sur cet ensemble l'approche collaborative pour suggérer les livres ayant été appréciés le plus par les étudiants les plus similaires à l'étudiant actif. La similarité entre deux étudiants est calculée par rapport à la similitude de leurs styles d'apprentissage et par rapport à la similitude des notions du domaine auxquelles ils s'intéressent le plus. Un avantage de cette approche étant de pouvoir faire profiter l'étudiant de l'expérience collective de ses pairs. De plus, cette approche permet de généraliser le processus de recommandation et de le rendre plus accessible puisqu'elle donne la possibilité à un étudiant de tirer profit de l'expérience d'une personne alors qu'il ne la connaît même pas.

7.3 DIA : Les spécificités

Comme nous avons vu, DIA comporte un client Z39.50 qui permet l'interrogation simultanée de plusieurs bases de données hétérogènes et réparties et ce :

- sans connaître la structure des bases des données interrogées ;
- à l'aide d'une requête unique ;
- où les données sont représentées dans un format normalisé facilement réutilisable.

Par conséquent, le client Z39.50 permet à DIA de se connecter à n'importe quelle bibliothèque ayant un serveur Z39.50 et de consulter la totalité de leurs catalogues de livres et de générer des recommandations à partir de toutes ces bibliothèques simultanément. Or des milliers de bibliothèques universitaires, gouvernementales ou municipales implémentent déjà ce protocole. DIA peut donc, en théorie, consulter et rechercher leurs bases de livres et recommander facilement les titres qu'elles contiennent.

Un désavantage de l'utilisation des bibliothèques comme sources des livres est le peu de descriptions disponibles, qui se résument à des données bibliographiques. Ces données ne sont en général pas assez descriptives pour permettre une bonne analyse des livres. Pour faire face à ce problème, DIA comprend un client HTTP lui permettant de chercher sur la Toile des descriptions de livres plus détaillées.

7.4 Discussion

La spécificité de DIA pour l'environnement pédagogique lui a permis d'acquérir plusieurs avantages par rapport aux besoins des étudiants. En effet, comme nous avons vu dans ce mémoire, l'évaluation du système a abouti à des résultats prometteurs. Globalement, elle a montré qu'il existe un certain accord, parmi les 112 étudiants qui ont testé le système, que ce projet a atteint tous ses objectifs et que DIA réussit à générer des recommandations de livres pédagogiques de façon adéquate. De plus, un consensus favorable a été remarqué sur l'utilité du système pour les étudiants. Mais bien que la spécialisation de DIA pour le

domaine éducatif est avantageuse pour les étudiants, il est évident qu'elle limite ses champs d'application et DIA ne pourra être efficace que dans cet environnement précis.

Bibliographie

[Abdel-Razek 2004] M. Abdel-Razek (2004). "*Multi-Agent Approach Towards Intelligent E-Learning System*". Doctor of Philosophy (Département d'Informatique et de Recherche Opérationnelle (DIRO)), Université de Montréal, Montréal, Qc, Canada.

[Abdel-Razek, *et al.* 2004] M. Abdel-Razek, C. Frasson et M. Kaltenbach (2004). "Building an Effective Groupware System". In *International Conference on Information Technology, IEEE/ITCC '04*, Las Vegas, NV, USA.

[Abdel-Razek, *et al.* 2003] M. Abdel-Razek, C. Frasson et M. Kaltenbach (2003). "Dominant Meanings Classification Model for Web Information". In *Design and Application of Hybrid Intelligent Systems*, Melbourne, Australia, IOS Press.

[Achike, *et al.* 2000] F. I. Achike et C. W. Ogle (2000). "Information overload in the teaching of pharmacology". *Journal of Clinical Pharmacology*, **40**(2): 177-183.

[Aggarwal, *et al.* 1999] C. C. Aggarwal, J. L. Wolf, K.-L. Wu et P. S. Yu (1999). "Horting hatches an egg: a new graph-theoretic approach to collaborative filtering". In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, San Diego, California, United States, ACM Press.

[Balabanovic, *et al.* 1997] M. Balabanovic et Y. Shoham (1997). "Fab: content-based, collaborative recommendation". *Communications of the ACM*, **40**(3): 66-72.

[Basilico, *et al.* 2004] J. Basilico et T. Hofmann (2004). "Unifying collaborative and content-based filtering". In *Twenty-first international conference on Machine learning*, Banff, Alberta, Canada, ACM Press.

[Battle 2004] J. C. Battle (2004). "*The effect of information literacy instruction on library anxiety among international students*". Doctor of Philosophy (Information Science), University of North Texas, Denton, Texas.

[Belkin, *et al.* 1992] N. J. Belkin et W. B. Croft (1992). "Information filtering and information retrieval: two sides of the same coin?" *Communications of the ACM*, **35**(12): 29-38.

[Billsus, *et al.* 1999] D. Billsus et M. J. Pazzani (1999). "A Hybrid User Model for News Story Classification". In *Proceedings of the 7th International Conference on User Modeling*, Banff, Alberta, Canada.

[Bin, *et al.* 2003] X. Bin, E. Aimeur et J. M. Fernandez (2003). "PCFinder: an intelligent product recommendation agent for e-commerce". In *IEEE International Conference on E-Commerce, CEC 2003*, Newport Beach, California, USA.

[Borchers, *et al.* 1998] A. Borchers, J. Herlocker, J. Konstan et J. Reidl (1998). "Ganging up on information overload". *Computer*, **31**(4): 106-108.

[Breese, *et al.* 1998] J. S. Breese, D. Heckerman et C. Kadie (1998). "Empirical Analysis of Predictive Algorithms for Collaborative Filtering". In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, Madison, Wisconsin, USA.

[Burke 2002] R. Burke (2002). "Hybrid Recommender Systems: Survey and Experiments". *User Modeling and User-Adapted Interaction*, **12**(4): 331-370.

[Chen, *et al.* 2001] H.-C. Chen et A. L. P. Chen (2001). "A music recommendation system based on music data grouping and user interests". In *Proceedings of the tenth international conference on Information and knowledge management*, Atlanta, Georgia, USA, ACM Press.

[Claypool, *et al.* 1999] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes et M. Sartin (1999). "Combining Content-Based and Collaborative Filters in an Online Newspaper". In *Proceedings of ACM SIGIR Workshop on Recommender Systems*.

[Claypool, *et al.* 2001] M. Claypool, P. Le, M. Waseda et D. Brown (2001). "Implicit interest indicators". In *Proceedings of the 6th International Conference on Intelligent User Interfaces (IUI '01)*, USA.

[Corfield, *et al.* 2002a] A. Corfield, M. Dovey, R. Mawby et C. Tatham (2002a). "JAFER ToolKit project: interfacing Z39.50 and XML". In *Proceedings of the second ACM/IEEE-CS joint conference on Digital libraries*, Portland, Oregon, USA, ACM Press.

[Corfield, *et al.* 2002b] A. Corfield, M. Dovey, R. Mawby et C. Tatham (2002b). "Z39.50 and XML - Bridging the old and the new". In *The eleventh international World Wide Web Conference, WWW2002*, Honolulu, Hawaii, USA.

[Fournier 1996] J. F. Fournier (1996). "Information Overload and Technology Education". In *Proceedings of the Seventh International Conference of the Society for Information Technology and Teacher Education, SITE '96*, Phoenix, AZ. U.S.A.

[Goldberg, *et al.* 2001] K. Goldberg, T. Roeder, D. Gupta et C. Perkins (2001). "Eigentaste: A Constant Time Collaborative Filtering Algorithm". *Information Retrieval*, **4**(2): 133-151.

[Gupta, *et al.* 1999] D. Gupta, M. Digiovanni, H. Narita et K. Goldberg (1999). "Jester 2.0 (demonstration abstract): collaborative filtering to retrieve jokes". In *Proceedings of the*

22nd annual international ACM SIGIR conference on Research and development in information retrieval, Berkeley, California, United States, ACM Press.

[Herlocker, *et al.* 1999] J. Herlocker, J. Konstan, A. Borchers et J. Riedl (1999). "An algorithmic framework for performing collaborative filtering". In *Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval*, Berkeley, CA.

[Hirooka, *et al.* 2000] Y. Hirooka, T. Terano et Y. Otsuka (2000). "Recommending books of revealed and latent interests in e-commerce". In *Industrial Electronics Society, 2000. IECON 2000. 26th Annual Conference of the IEEE*, Nagoya, Japan.

[Hofmann 2004] T. Hofmann (2004). "Latent semantic models for collaborative filtering". *ACM Transactions on Information Systems (TOIS)*, **22**(1): 89-115.

[Huang, *et al.* 2004] Z. Huang, H. Chen et D. Zeng (2004). "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering". *ACM Transactions on Information Systems (TOIS)*, **22**(1): 116-142.

[Iwahama, *et al.* 2004] K. Iwahama, Y. Hijikata et S. Nishida (2004). "Content-based filtering system for music data". In *International Symposium on Applications and the Internet Workshops, SAINT 2004*, Tokyo, Japan.

[Jiao, *et al.* 1999] Q. G. Jiao et A. J. Onwuegbuzie (1999). "Identifying library anxiety through students' learning-modality preferences." *The Library Quarterly*, **69**(2): 202 - 216.

[Jiao, *et al.* 1998] Q. G. Jiao et A. J. Onwuegbuzie (1998). "Perfectionism and library anxiety among graduate students". *The Journal of Academic Librianship*, **24**(5): 365-371.

[Jiao, *et al.* 1996] Q. G. Jiao, A. J. Onwuegbuzie et A. A. Lichtenstein (1996). " Library Anxiety: Characteristics of 'At-Risk' College Students." *Library & Information Science Research*, **18**(2): 151-163.

[Kai, *et al.* 2004] Y. Kai, A. Schwaighofer, V. Tresp, X. Xiaowei et H. P. Kriegel (2004). "Probabilistic memory-based collaborative filtering". *IEEE Transactions on Knowledge and Data Engineering*, **16**(1): 56-69.

[Kelly, *et al.* 2003] D. Kelly et J. Teevan (2003). "Implicit feedback for inferring user preference: a bibliography". *ACM SIGIR Forum*, **37**(2): 18-28.

[Kobayashi, *et al.* 2000] M. Kobayashi et K. Takeda (2000). "Information retrieval on the web". *ACM Computing Surveys*, **32**(2): 144-173.

[Konstan, *et al.* 1997] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon et J. Riedl (1997). "GroupLens: applying collaborative filtering to Usenet news". *Communications of the ACM*, **40**(3): 77-87.

[Kouwe 2003] Z. Kouwe (2003). "Cranky Consumer: Getting Book Suggestions Online". *The Wall Street Journal*, **issue date:** 29 July 2003.

[Linden, *et al.* 2003] G. Linden, B. Smith et J. York (2003). "Amazon.com recommendations: item-to-item collaborative filtering". *Internet Computing, IEEE*, **7**(1): 76-80.

[Ludwig, *et al.* 1997] A. Ludwig, P. Becker et U. Guntzer (1997). "Interfacing online bibliographic databases with Z39.50". In *International Database Engineering and Applications Symposium, IDEAS '97*, Montreal, Que. Canada.

[Mellon 1986] C. A. Mellon (1986). "Library Anxiety: A Grounded Theory and Its Development." *College & Research Libraries*, **47**(2): 160 - 165.

[Melville, *et al.* 2002] P. Melville, R. J. Mooney et R. Nagarajan (2002). "Content-boosted collaborative filtering for improved recommendations". In *Eighteenth national conference on Artificial intelligence*, Edmonton, Alberta, Canada, American Association for Artificial Intelligence.

[Mooney, *et al.* 2000] R. J. Mooney et L. Roy (2000). "Content-based book recommending using learning for text categorization". In *Proceedings of the fifth ACM conference on Digital libraries*, San Antonio, Texas, United States, ACM Press.

[Moore, *et al.* 2000] J. Moore, S. Cvetkovic, K. Hung et M. Kraner (2000). "The Z39.50 information retrieval standard". *Computing & Control Engineering Journal*, **11**(3): 143-151.

[Morita, *et al.* 1994] M. Morita et Y. Shinoda (1994). "Information filtering based on user behavior analysis and best match text retrieval". In *Proceedings of the 17th Annual International Conference on Research and Development in Information Retrieval*, Ireland.

[NISO 2003] National Information Standards Organization (2003). "*Information Retrieval (Z39.50): Application Service Definition and Protocol Specification*". ANSI/NISO Z39.50-2003 (revision of Z39.50-1995). ISBN: 1-880124-55-6. NISO Press, Bethesda, Maryland, U.S.A.

[O'Brien, *et al.* 2003] C. O'Brien et C. Vogel (2003). "Spam filters: bayes vs. chi-squared; letters vs. words". In *Proceedings of the 1st international symposium on Information and communication technologies*, Dublin, Ireland, Trinity College Dublin.

[Onwuegbuzie, *et al.* 2000] A. J. Onwuegbuzie et Q. G. Jaio (2000). "I'll Go to the Library Later: The Relationship between Academic Procrastination and Library Anxiety". *College & Research Libraries*, **61**(1): 45 - 54.

[Pazzani 1999] M. J. Pazzani (1999). "A Framework for Collaborative, Content-Based and Demographic Filtering". *Artificial Intelligence Review*, **13**(5-6): 393-408.

[Salton, *et al.* 1986] G. Salton et M. J. McGill (1986). "*Introduction to Modern Information Retrieval*". McGraw-Hill, Inc.

[Sarwar, *et al.* 2001] B. Sarwar, G. Karypis, J. Konstan et J. Reidl (2001). "Item-based collaborative filtering recommendation algorithms". In *Proceedings of the tenth international conference on World Wide Web*, Hong Kong, Hong Kong, ACM Press.

[Sarwar, *et al.* 2000] B. Sarwar, G. Karypis, J. Konstan et J. Riedl (2000). "Analysis of recommendation algorithms for e-commerce". In *Proceedings of the 2nd ACM conference on Electronic commerce*, Minneapolis, Minnesota, United States, ACM Press.

[Schafer, *et al.* 2001] J. B. Schafer, J. A. Konstan et J. Riedl (2001). "E-Commerce Recommendation Applications". *Data Mining and Knowledge Discovery*, **5**(1-2): 115-153.

[Shahabi 2003] C. Shahabi (2003). "Web Information Personalization: Challenges and Approaches". In *3nd International Workshop on Databases in Networked Information Systems (DNIS 2003)*, Aizu-Wakamatsu, Japan.

[Stanley, *et al.* 1997] A. J. Stanley et P.S. Clipsharn (1997). "Information Overload - Myth or Reality?" In *Proceedings of the 1997 IEE Colloquium on Information Technology Strategies for Information Overload*, London, UK.

[Tang, *et al.* 2003] T. Y. Tang et G. McCalla (2003). "Towards Pedagogy-Oriented Paper Recommendation and Adaptive Annotations for a Web-Based Learning System". In *the 18th International Joint Conference on Artificial Intelligence, Workshop on Knowledge Representation and Auto-mated Reasoning for E-Learning Systems, (IJCAI '03)*, Acapulco, Mexico.

[Yammine, *et al.* 2004] K. Yammine, M. Abdel-Razek, E. Aïmeur et C. Frasson (2004). "Discovering Intelligent Agent: A Tool for Helping Students Searching a Library". In *Proceedings of the 7th International Conference on Intelligent Tutoring Systems: (ITS 2004)*, Maceió, Alagoas, Brazil, Springer-Verlag.

[Yu, *et al.* 2004] K. Yu, V. Tresp et S. Yu (2004). "A nonparametric hierarchical bayesian framework for information filtering". In *Proceedings of the 27th annual international conference on Research and development in information retrieval*, Sheffield, United Kingdom, ACM Press.

[Zaïane 2002] O. R. Zaïane (2002). "Building a Recommender Agent for e-Learning Systems". In *Proceedings of the 7th International Conference on Computers in Education, (ICCE 2002)*, Auck-land, New Zealand.

Quelques bibliothèques ayant un serveur Z39.50

Ceci n'est qu'un échantillon des nombreuses bibliothèques qui emploient le standard Z39.50. Ces adresses ont été vérifiées valides le **24 janvier 2005**, mais des changements peuvent survenir.

Bibliothèques	Adresse du Serveur	Port	Nom de la BD
Université de Montréal *(Québec)*	atrium.bib.umontreal.ca	210	ADVANCE
McGill University *(Québec)*	aleph.mcgill.ca	210	MUSE
Concordia University *(Québec)*	mercury.concordia.ca	210	INNOPAC
Université Laval *(Québec)*	ariane.ulaval.ca	210	ULAV
University of British Columbia *(Canada)*	dra.library.ubc.ca	210	MARION
Université de Québec *(Québec)*	catalogue.uquebec.ca	210	UQAM
National Library of Canada *(Canada)*	amicus.nlc-bnc.ca	210	CGI
University of California *(USA)*	128.218.15.173	210	INNOPAC
Massachusetts Institute of Technology *(USA)*	library.mit.edu	9909	ALEPH
Université catholique de Louvain *(Belgique)*	bib.sia.ucl.ac.be	3520	DEFAULT
University of North West *(South Africa)*	196.6.221.16	210	INNOPAC
University of Melbourne *(Australie)*	library.unimelb.edu.au	210	INNOPAC
Griffith University *(Australie)*	library.gu.edu.au	21210	GEAC
Göteborg University *(Suède)*	sunda.ub.gu.se	8010	VTLS
Aberdeen University *(UK)*	aulib.abdn.ac.uk	9991	ABN01
Oxford University *(UK)*	library.ox.ac.uk	210	BIBMAST
University of London *(UK)*	qmwlib4.library.qmul.ac.uk	2200	UNICORN