



HAL
open science

Une architecture générique de Systèmes de recommandation de combinaison d'items : application au domaine du tourisme

Romain Picot-Clémente

► To cite this version:

Romain Picot-Clémente. Une architecture générique de Systèmes de recommandation de combinaison d'items : application au domaine du tourisme. Autre [cs.OH]. Université de Bourgogne, 2011. Français. NNT : 2011DIJOS038 . tel-00688994v2

HAL Id: tel-00688994

<https://theses.hal.science/tel-00688994v2>

Submitted on 30 Apr 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Mémoire de Thèse

Présenté pour obtenir

Le grade de Docteur ès Sciences

Mention Informatique

Par

Romain Picot-Clément

Une architecture générique de Systèmes de recommandation de combinaison d'items. Application au domaine du tourisme

Le 07 Decembre 2011

Composition du jury :

Mr. Jacky Akoka, Professeur, CNAM Paris, **Rapporteur**

Mr. Gilles Kassel, Professeur, Université de Picardie, **Rapporteur**

Mr. Jean-Christophe Lapayre, Professeur, Université de Franche-Comté, **Examineur**

Mr. Jean Pallo, Professeur, Université de Bourgogne, **Examineur**

Mr. Christophe Nicolle, Professeur, Université de Bourgogne, **Directeur de thèse**

Mr. Christophe Cruz, Maître de conférences, Université de Bourgogne, **Encadrant de thèse**

Doctorat préparé au sein de l'équipe Systèmes d'Information et Systèmes d'Images, dans l'équipe projet CHECKSEM du laboratoire Le2i de l'Université de Bourgogne, et financé par l'Agence de Développement Touristique Côte-d'Or Tourisme



À mes Parents, Frères et Sœur, et Grands-Mères...

Remerciements

Je souhaite remercier en premier lieu mon directeur de thèse, M. Christophe Nicolle, Professeur des Universités et directeur adjoint du laboratoire Électronique Informatique et Image pour m'avoir accueilli au sein de son équipe. Je lui suis également reconnaissant pour le temps conséquent qu'il m'a accordé, ses qualités pédagogiques et scientifiques, sa franchise et sa sympathie. J'ai beaucoup appris à ses côtés et je lui adresse ma gratitude pour tout cela.

J'adresse de chaleureux remerciements à mon co-encadrant de thèse, M. Christophe Cruz, Maître de conférences du laboratoire Électronique Informatique et Image, pour son attention de tout instant sur mes travaux, pour ses conseils avisés et son écoute qui ont été prépondérants pour la bonne réussite de cette thèse. Son énergie et sa confiance ont été des éléments moteurs pour moi. J'ai pris un grand plaisir à travailler avec lui.

Un grand merci à Mme Florence Mendes, Maître de conférences du laboratoire Électronique Informatique et Image, pour son implication dans le projet, notamment sur la problématique d'optimisation combinatoire. Elle m'a beaucoup appris, j'ai apprécié son enthousiasme et sa sympathie.

Je voudrais remercier les rapporteurs de cette thèse M. Jacky Akoka, Professeur des Universités de l'équipe CNAM-CEDRIC du CNAM Paris, et M. Gilles Kassel, Professeur des Universités, directeur du laboratoire MIS de l'université de Picardie, pour l'intérêt qu'ils ont porté à mon travail.

J'associe à ces remerciements M. Jean Pallo, Professeur des Universités du laboratoire Électronique Informatique et Image, et M. Jean-Christophe Lapayre, Professeur des Universités du LIFC, pour avoir accepté d'examiner mon travail.

J'adresse de sincères remerciements à M. Jean Pierre Rebourgeon, Président de Côte-d'Or Tourisme et vice-président du conseil général pour m'avoir accueilli chaleureusement au sein de l'agence de développement touristique dans le cadre d'un contrat doctoral.

Je désire grandement remercier Mme Pascale Lambert, Directrice de Côte-d'Or Tourisme pour m'avoir choisi sur ce projet et pour m'avoir fait confiance tout au long de ces trois années. Ses remarques, son ouverture d'esprit, sa franchise, sa gentillesse sont autant d'éléments qui m'ont permis d'atteindre les objectifs de l'entreprise dans le cadre du doctorat. Je la remercie pour tout cela.

Bien sûr, atteindre ces objectifs n'aurait pas été possible sans l'aide des membres de Côte-d'Or Tourisme qui ont participé au projet. Je souhaite notamment remercier Mme Christine Legoux, chef de projet TIC ; Mme Isabelle Corond, Chef du pôle Développement et sous-directrice ; Mme Christelle Skowron, Chef du pôle Communication ; M. Pierre Laforêt, Expert TIC et M. Nicolas Romain, administrateur informatique.

Je désire en outre remercier tous les membres de Côte-d'Or Tourisme (incluant ceux précédemment cités) pour leur sympathie, leur amitié. J'ai eu beaucoup de plaisir à travailler avec eux. Le cadre de travail était idéal.

Merci à Laurent Ducarroz, Ingénieur associé au projet CHECKSEM pour son implication dans mes travaux. Il m'a donné des conseils avisés et a été une grande ressource pour le développement de l'application mobile.

Je tiens à remercier tous les membres de l'équipe projet CHECKSEM, pour leur aide et leur bonne humeur. Nous avons partagé de bons moments.

Je souhaite remercier spécialement Jennifer pour son soutien et sa patience tout au long de la thèse.

Enfin, je remercie mes frères et sœur, Nicolas, Jérémy et Luana, ainsi que mes parents, Olivier et Françoise Picot-Clémente pour leur soutien au cours de ces trois années et sans lesquels je n'en serais pas là aujourd'hui.

Résumé

Cette thèse apporte une généralisation du principe de recommandation des systèmes de recommandation. Au lieu de considérer une recommandation comme un item, elle est considérée comme une combinaison constituée de plusieurs items suivant un pattern donné. Une recommandation d'un seul item est alors un cas particulier de ce type de recommandation.

L'architecture de système de recommandation proposé se base sur une architecture dérivée des travaux en systèmes hypermédia adaptatifs. Trois couches sont définies : une couche sémantique, une couche utilisateur et une couche intelligence. La couche sémantique est constituée de deux sous-couches, une sous-couche modélisant le contenu suivant la connaissance générale du domaine et une sous-couche modélisant le contenu suivant la connaissance spécifique à l'application, plus précisément spécifique aux possibles contraintes des utilisateurs dans l'application. Cette deuxième partie permet de prendre en compte le savoir-faire du fournisseur de l'offre dans les propositions du système de recommandation. La couche utilisateur modélise l'utilisateur au sein du système de recommandation et la couche intelligence contient le processus de recommandation. Nous proposons de décomposer ce dernier en deux sous-processus principaux, un processus dit de projection des individus du domaine sur les profils utilisateurs et un processus de recherche combinatoire. Le premier apporte une pondération, appelée note dans la plupart des systèmes de recommandation, donnant les intérêts probables des utilisateurs pour les différents items. Le processus de recherche combinatoire recherche parmi la multitude de combinaisons possibles, une solution convenable (optimale si possible) à proposer à l'utilisateur.

Cette architecture de système de recommandation combinatoire est appliquée au domaine touristique pour l'entreprise Côte-d'Or Tourisme impliquée dans le contrat de recherche. Le but de cette application est de proposer à l'utilisateur un ensemble d'offres touristiques sous forme de séjour. Ce problème touristique amène à la définition formelle d'un problème d'optimisation combinatoire qui est une variante d'un sous-problème du problème de sac à dos. Pour résoudre ce genre de problème, il est nécessaire d'utiliser une métaheuristique afin de tendre vers une bonne solution en un temps raisonnable. Nous présentons un algorithme basé sur le recuit simulé et un algorithme multi-objectif pour la résolution de ce problème.

L'instanciation de chaque couche de l'architecture pour le système touristique est décrite en détail. Enfin, cette thèse présente une application mobile faisant office d'interface utilisateur avec le système de recommandation touristique et elle présente les développements techniques nécessaires à ce projet, étant donné son contexte industriel.

Abstract

This thesis gives a generalization of the recommendation principle of recommender systems. Instead of considering a recommendation as an item, it is considered as a combination of several items following a given pattern. A recommendation of a unique item is then a particular case of this type of recommendation.

The proposed recommender system framework is based on an architecture derived from work in adaptive hypermedia systems. Three layers are defined: a semantic layer, a user layer and an intelligence layer. The semantic layer consists of two sub-layers, a sub-layer modeling the content according to the general knowledge of the domain and a sub-layer modeling the content according to the specific knowledge of the application. This second sub-layer allows taking into account the expertise of the offer's supplier for the proposals of the recommender system. The user layer models the user into the recommender system and the intelligence layer contains recommendation process. We propose to decompose the adaptation into two main processes, a process called projection of domain's individuals on user profiles and a combinatorial research process. The first brings a weight, called the note in most recommender systems, giving probable users' interests on the different items. The combinatorial research process searches among the many possible combinations a suitable solution (optimal if possible) to propose to the user.

This framework of combinatory recommender systems is applied to the domain of tourism for the company Côte-d'Or Tourisme implied in the research contract. The purpose of this application is to offer the user a set of tourism offers in the form of journey. This tourism problem leads to the formal definition of a combinatory optimization problem which is a variant of the knapsack problem. To resolve this kind of problem, it is necessary to use a metaheuristic to tend toward a good solution in a reasonable time. We present an algorithm based on simulated annealing and a multi-objective algorithm to solve this problem.

The instantiation of each layer of the framework for the tourism system is described in detail. Finally, this thesis presents a mobile application serving as a user with the tourism recommender system and it presents the technical developments for this project.

Table des matières

Introduction	21
1 Côte d'Or Tourisme	23
2 Approche.....	25
3 Apport	26
4 Organisation du document	27
Etat de l'art	29
1 Systèmes hypermédia adaptatifs	31
1.1 Architecture des systèmes hypermédia adaptatifs	33
1.2 Représentation du domaine et de l'utilisateur	36
1.3 Conclusion	37
2 Systèmes de recommandation	37
2.1 Systèmes de recommandation basés sur le contenu	40
2.1.1 Techniques de représentation et de recommandation	40
2.1.2 Recommandations basées sur les vecteurs de mots-clefs.....	41
2.1.3 Recommandations basées sur la sémantique	44
2.2 Systèmes de recommandation basés sur une approche collaborative	47
2.2.1 Recommandations basées sur le voisinage	47
2.2.2 Recommandations basées sur un modèle prédictif.....	49
2.3 Conclusion	50
3 Conclusion sur le chapitre.....	51
Approche Sémantique et Combinatoire pour un système de Recommandation Générique	55
1 Discussion	57
2 Couche sémantique	60
2.1 Le Web sémantique	60
2.2 Ontologie.....	61
2.3 Modèle de domaine	62
2.4 Modèle de buts	62
3 Couche utilisateur	64
4 Couche intelligence	65
4.1 Phase de projection.....	65
4.1.1 Formalisation	65
4.1.2 Exemple	66
4.2 Phase de recherche combinatoire	66

4.2.1	Forme de la combinaison.....	67
4.2.2	Pertinence d'une combinaison	67
4.2.3	Problème de recherche de la meilleure combinaison	68
4.3	Récapitulatif/Aperçu de la couche intelligence	68
5	Conclusion.....	69
Problème d'Optimisation.....		71
1	Définition du problème d'optimisation touristique	73
1.1	Item et Poids	73
1.2	Pattern et Combinaison	74
1.3	Dispersion et Pertinence d'une Solution.....	74
1.4	Liens avec d'autres problèmes d'optimisation.....	75
2	Les métaheuristiques.....	76
2.1	Optimisation basée sur le Gradient.....	77
2.2	Les méthodes à état simple	80
2.2.1	Recherche locale et globale	80
2.2.2	Méthode gloutonne.....	83
2.2.3	Recuit simulé.....	83
2.2.4	Méthodes d'acceptation à seuil.....	85
2.2.5	Recherche Tabou	85
2.2.6	Recherche locale itérée	86
2.3	Méthodes avec population	87
2.3.1	Stratégies d'évolution.....	87
2.3.2	L'algorithme génétique.....	88
2.3.3	La colonie de fourmis.....	89
2.4	Méthodes multi-objectifs.....	90
3	Conclusion.....	91
Application au domaine du tourisme		95
1	Implémentation de la couche sémantique	98
1.1	Processus de création du modèle de domaine	98
1.2	Processus de création du modèle de buts	102
2	Implémentation de la couche utilisateur	102
3	Implémentation de la couche intelligence.....	103
3.1	Construction dynamique du pattern.....	103
3.2	Attribution des tolérances de dispersion	104
3.3	Algorithme de recherche combinatoire basé sur le recuit simulé	105
3.4	Benchmark, comparaison avec Hill-Climbing	108

3.5	Algorithme de recherche combinatoire basé sur un algorithme Multi objectif	109
4	Conclusion.....	113
	Développement industriel	115
1	L'offre touristique	117
2	Infrastructure	119
3	Application Mobile/Interface utilisateur	120
3.1	Fonctionnalités.....	120
3.1.1	Définition du profil utilisateur	120
3.1.2	Génération de la combinaison et visualisation.....	122
3.1.3	Régénération d'une combinaison et fonctionnement de la géolocalisation	124
3.2	Aspect technique de l'application iPhone.....	125
4	Calibration/ajustement.....	125
5	Conclusion.....	126
	Conclusion & Travaux Futurs	129
	Bibliographie	135
	Rapports techniques	143
	Publications	145

Liste des figures

Figure 1. Evolution de l'offre web de côte d'or tourisme	25
Figure 2. Technologies de l'hypermédia adaptatif	32
Figure 3. Modèle Dexter	33
Figure 4. AHAM	34
Figure 5. Munich	34
Figure 6. LAOS	35
Figure 7. Classification des différents types de systèmes de recommandation	52
Figure 8. Architecture du SR	59
Figure 9. Ontologie du domaine de la vente de véhicules d'occasion	62
Figure 10. Structure de la couche sémantique	64
Figure 11. Extrait d'une hiérarchie de classes d'une ontologie de domaine	67
Figure 12. Aperçu des différentes interactions de la couche intelligence	69
Figure 13. Positionnement de notre proposition par rapport aux approches existantes	70
Figure 14. Ascension gradient avec une pente négative	78
Figure 15. Point d'inflexion	78
Figure 16. Ascension gradient dépassant le maximum	78
Figure 17. Exploration de X par une approche de recherche locale	80
Figure 18. Tournée du voyageur de commerce	83
Figure 19. Région des solutions Pareto dominée par la solution A, incluant les solutions du bord	91
Figure 20. Front de Pareto des solutions non-dominées	91
Figure 21. Aperçu du système actuel	97
Figure 22. Aperçu du système amélioré	98
Figure 23. Aperçu d'une structure ontologique générée par le programme de traduction	100
Figure 24. Aperçu d'une ontologie après la phase de peuplement	100
Figure 25. Exemple d'enrichissement de l'ontologie touristique	101
Figure 26. Processus de création de l'ontologie de domaine	101
Figure 27. Diagramme fonctionnel de l'algorithme du recuit simulé utilisé	106
Figure 28. Exemple de variation de l'énergie de la combinaison courante par algorithme de recuit simulé	107
Figure 29. Illustration du front de Pareto	111
Figure 30. Modèle conceptuel de donnée des objets touristiques	118
Figure 31. Aperçu des interactions entre serveur, entrepôts de données et interfaces	119
Figure 32. Ecran du carrousel de buts	121
Figure 33. Ecran de contrainte temporelle	121
Figure 34. Ecran de choix du type d'hébergement	122
Figure 35. Ecran de contrainte géographique	122
Figure 36. Visualisation du séjour par liste	123
Figure 37. Visualisation du séjour sur une carte	123
Figure 38. Visualisation du séjour en réalité augmentée	123
Figure 39. Visualisation d'un item touristique	123
Figure 40. Action de régénération totale du séjour	124
Figure 41. Outil de visualisation des items de type restauration	126

Liste des Tableaux

Tableau 1. Aperçu Table descriptive	99
Tableau 2. Aperçu de la vue Hotel, de type Data	99
Tableau 3. Aperçu de la vue Restaurant, de type Data	99
Tableau 4. Aperçu de la vue Hotel_possede_restaurant, de type « relation ».....	99
Tableau 5. Recuit simulé VS Hill-Climbing sur des jeux de données réels et aléatoires	109

Liste des Equations

Équation 1 41
Équation 2 41
Équation 3 42
Équation 4 42
Équation 5 47
Équation 6 48
Équation 7 48
Équation 8 48
Équation 9 48
Équation 10 49
Équation 11 49
Équation 12 49
Équation 13 74
Équation 14 74
Équation 15 75
Équation 16 75
Équation 17 79
Équation 18 79
Équation 19 84
Équation 20 84

Chapitre 1

Introduction

Plan

1	Côte d'Or Tourisme	23
2	Approche	25
3	Apport	26
4	Organisation du document.....	27

Depuis une quarantaine d'années, les technologies ont révolutionné les échanges entre clients et fournisseurs poussant les entreprises à adapter continuellement leurs stratégies de vente. Le secteur du tourisme est l'un des exemples les plus frappants de cette tendance. Dès 1970, des systèmes de réservation sur ordinateurs, à la réservation en ligne d'aujourd'hui, les innovations technologiques ont radicalement transformé les processus et les stratégies du tourisme [1]. De nos jours, avec le développement d'Internet, le nombre d'offres touristiques semble croître de façon exponentielle tandis que le lien entre le client et le fournisseur a été raccourci, en supprimant les intermédiaires. Maintenant, les clients ont la possibilité d'accéder directement aux offres où la stratégie du faible coût semble être devenue le seul critère de qualité. Cependant, ces innovations ont considérablement réduit l'aide apportée par le professionnel du tourisme dans le choix, les services de conseil et de personnalisation de l'offre telle qu'elles existent dans les agences de voyages. La gestion des relations avec la clientèle est une réflexion stratégique fondée sur une analyse attentive et minutieuse du client, en établissant une relation de confiance favorisant l'implication du personnel et des partenaires. Pour mieux analyser les comportements du client au moment du choix, de nouvelles techniques de ciblage comportemental (ou plus simplement de profilage) ont été développées. Ce nouveau domaine du Web marketing consiste à analyser les demandes des utilisateurs pour mieux comprendre et cibler plus efficacement leurs attentes et exigences. Grâce aux recherches d'un internaute, il est possible de connaître non seulement quel type de service il utilise, mais aussi s'il a des enfants ou même s'il a répondu positivement à une newsletter électronique. L'impact de ces nouveaux champs d'études est susceptible de dépasser les enquêtes actuellement utilisées et d'aider à adapter efficacement les demandes des usagers dans le domaine du tourisme.

Le tourisme a une place très importante en France, car il génère plus de 72 milliards d'euros de chiffre d'affaires, représentant plus de 6% du PIB (contre 3.5% pour l'agriculture et agro-industrie, et 1% pour l'industrie automobile). Il est possible de dénombrer environ 200 000 compagnies touristiques et près de 2 millions emplois directs et indirects générés par le tourisme.

1 CÔTE D'OR TOURISME

Côte-d'Or Tourisme, association loi 1901, créée en 1964 est le comité départemental de tourisme du département de la Côte-d'Or. L'association met en œuvre la politique touristique du Conseil Général de la Côte-d'Or, au travers du schéma de développement touristique. Lors de sa session de décembre 2005, l'assemblée départementale adopte le schéma départemental de tourisme pour les années 2006 à 2011. Ce dernier prend en compte les orientations définies par le schéma régional de développement touristique et se veut un outil évolutif d'accompagnement des territoires et des porteurs de projet, dans un contexte de mondialisation et de concurrence forte entre toutes les destinations. Le Conseil Général entretient par ailleurs des rapports privilégiés avec Côte-d'Or Tourisme. Ce partenariat étroit engage le Comité départemental de tourisme de la Côte-d'Or à conduire des actions de développement en relais, en appui, ou en prolongement de la politique départementale. Le schéma départemental de tourisme renforçant le positionnement de Côte-d'Or Tourisme en matière de développement touristique par rapport à la promotion, l'association procède à une modification de ses statuts en 2006 et devient « agence de développement », on ne parle dès lors plus de CDT mais d'ADT.

Dans le cadre du plan d'action « Côte-d'Or Numérique », initié par le Conseil Général de la Côte-d'Or, Côte-d'Or Tourisme a été missionnée pour sensibiliser la filière touristique aux technologies de l'information et de la communication. La première étape de ce programme a consisté en une mission visant à faire participer les professionnels côte-d'oriens à une démarche collective de mobilisation et d'accompagnement. Cette mobilisation a été conduite auprès de 4 000 professionnels et acteurs du tourisme en Côte-d'Or et plus de 70 structures afin de recueillir leurs besoins.

À l'issue de cette étude, un plan d'action a été établi : PISTES21 dont les principaux domaines d'intervention sont :

- La structuration du territoire pour améliorer le travail collaboratif entre les différents acteurs du tourisme en Côte-d'Or et favoriser l'intelligence collective. L'action majeure est la mise en place d'un extranet comportant notamment des fonctionnalités de partage de ressources juridiques.
- Amélioration de la qualité de l'offre en matière de visibilité. L'avènement d'Internet dans le tourisme permet de dépasser les frontières géographiques. De nouveaux clients utilisent ce média pour préparer leur séjour. L'offre présentée sur les sites internet vendant la destination doit par conséquent correspondre aux critères de qualité de ces nouveaux clients, mais également disposer d'un référencement efficace.
- Commercialisation. Si la plupart des sites internet présentent l'offre touristique de manière satisfaisante, en revanche la mise en marché est encore compliquée. La commercialisation du territoire sur Internet passe notamment par de la sensibilisation auprès des producteurs et par la recherche de solutions techniques adaptées au territoire.
- Gestion de la relation client. Ce volet du plan d'action PISTES21 a pour objet d'établir une stratégie à long terme pour la gestion de la relation client, c'est-à-dire la gestion de l'après-séjour. Les TIC sont utilisées ici pour réaliser des segmentations de clientèle ou encore des opérations de communication sur le réseau Internet.
- Collecte/base de données touristique. L'objet ici est la résolution des problèmes liés aux systèmes d'information décrivant l'offre touristique. Ce chantier est le point le plus central du plan d'action. Car, Il est lié à tous les autres. En fait, il est dangereux de vendre des produits touristiques, et de mettre en place des sites internet performants, s'ils ne sont pas soutenus par un back-office solide.

Pour répondre à l'ensemble de ces objectifs, l'ADT Côte-d'Or Tourisme a développé un système d'information collaboratif basé sur des technologies Web. En 2007, un site Web est développé en HTML avec des applets Java. Néanmoins, l'offre touristique n'est pas structurée, les fournisseurs ne peuvent remplir directement le site Web. En 2008, une plateforme Web de dépôt des offres touristiques a été développée pour permettre aux fournisseurs de spécifier leurs différentes offres. Ainsi, environ 4000 fournisseurs ont pu être fédérés dans une base de données relationnelle. Cela a eu pour conséquence de multiplier par trois le trafic du site Web (Figure 1). L'accès à un ensemble complet de données à jour semble être un grand facteur d'attractivité. En 2009, la compagnie souhaite offrir des sites Web dédiés à des publics particuliers. La première étape a été réalisée sous la forme d'une analyse de profilage des utilisateurs. Le site Web est alors devenu un portail fournissant un accès à une variété de sites Web : tourisme de proximité (<http://www.tamtam21.com/>), tourisme dédié au vignoble (<http://www.divine-comedie.com/>), le tourisme nature (<http://www.bouger-nature-en-Bourgogne.com/>). Les nouvelles particularités du Web 2.0 sont disponibles, comme un accès personnalisé, des blogs, des forums, etc. La fréquentation des sites a augmenté d'environ 15% durant 2009. L'ajout de fonctionnalités du Web social à des offres touristiques montre de réelles améliorations pour certains utilisateurs de l'Internet (commentaires, critiques, conseils, ...).

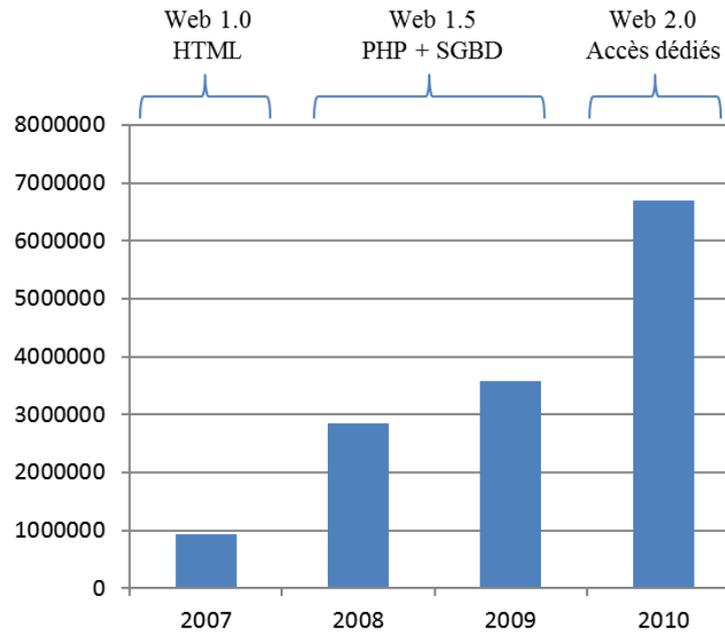


FIGURE 1. ÉVOLUTION DE L'OFFRE WEB DE CÔTE D'OR TOURISME

Depuis, toujours dans une optique de personnalisation à l'utilisateur, l'ADT Côte d'Or Tourisme désire développer un système de gestion de relation client qui permet d'affiner la sélection et l'affichage des offres touristiques. Ce projet doit permettre de transférer le savoir-faire en matière de conseil touristique à l'utilisateur final du site. L'assistance aux choix des clients s'est beaucoup réduite avec l'avancée du Web. Il n'y a pas ou peu de proximité cliente sur le Web. Les offres touristiques sont proposées sous forme de listes sans personnalisation. Pour répondre à cet objectif et résoudre ces contraintes liées au Web, l'ADT Côte d'Or Tourisme a signé un contrat de collaboration avec le CNRS et l'Université de Bourgogne. Le travail de recherche a été confié à l'équipe de recherche Checksem du laboratoire LE2I de l'Université de Bourgogne. Ce travail a pour objectif de construire un système de recommandation touristique permettant de fournir à un utilisateur final une combinaison pertinente d'offres touristiques hétérogènes selon son profil et son contexte de consommation de l'offre touristique. La pertinence de la combinaison doit être assurée par la prise en compte du savoir-faire du fournisseur de l'offre touristique.

2 APPROCHE

Pour atteindre les objectifs fixés par la société, nous avons souhaité étudier les systèmes hypermédia adaptatifs pour comprendre comment fonctionnaient les mécanismes d'adaptation d'un ensemble de données. L'hypermédia adaptatif concerne le développement de système Web où le contenu et la navigation sont dynamiquement adaptés en fonction d'un profil utilisateur. Nous avons pu identifier que l'architecture d'un système hypermédia adaptatif est construite par un empilement de couches. Nous avons aussi identifié une architecture minimum qui peut servir de base à tous types de systèmes hypermédia adaptatifs. Cette architecture est formée de trois couches : le modèle de domaine, le modèle d'adaptation et le modèle utilisateur. Le premier modèle contient l'ensemble des données du système, le second contient les règles d'adaptations et le dernier décrit le profil de l'utilisateur.

Notre recherche nous a permis d'isoler un sous-ensemble des systèmes hypermédia adaptatifs qui sont les systèmes de recommandations. Une étude approfondie de ces systèmes nous a permis d'identifier les principales caractéristiques de ces systèmes et de proposer une classification des principales approches. Toutes les approches proposent des mécanismes pour recommander un item (terme usuel dans le domaine des systèmes de recommandation pour parler d'un élément à recommander) à un utilisateur en se basant sur une analyse du contenu des offres à recommander ou une analyse des avis des utilisateurs. Bien que nous ayons pu identifier certains mécanismes utiles pour notre problème, tels que ceux utilisant des ontologies pour définir la sémantique des offres, aucun projet de recherche identifié ne permet ni de proposer des combinaisons d'offres, ni de mécanismes de recommandation sur le savoir-faire du fournisseur de l'offre.

Le travail de recherche présenté dans ce mémoire a pour principal objectif de développer une architecture générique qui permet la construction d'un système de recommandation qui répond aux limites décrites précédemment (combinaison et savoir-faire). Dans un second temps, ce travail présente l'application de cette architecture pour la construction d'un système de recommandation dédié au domaine du Tourisme et plus spécifiquement aux contraintes de l'agence de tourisme Côte d'Or tourisme.

Pour répondre au problème de combinaison, nous avons étudié le domaine de recherche de l'optimisation combinatoire. Ce domaine nécessite de définir précisément le problème d'optimisation qui doit être étudié. Nous avons défini le problème d'optimisation de l'agence de tourisme Côte d'Or tourisme, puis nous avons étudié les différentes métaheuristiques qui pouvaient répondre à ce problème.

Pour répondre au problème de la modélisation du savoir-faire, nous avons étudié le domaine du web sémantique, puisque les systèmes de recommandations les plus proches de notre problème utilisent des ontologies. Nous avons étendu ces propositions en ajoutant des mécanismes de pondérations et des règles logiques pour répondre aux besoins de la société en matière de modélisation de son savoir-faire métier.

3 APPORT

Dans ce projet, nous avons construit une architecture générique pour la construction d'un système de recommandation qui répond aux besoins de l'entreprise en matière de combinaison d'offres et de modélisation du savoir-faire. Cette architecture est composée d'un ensemble de couches qui associe des ontologies, des règles et pondérations, des métaheuristiques permettant de fournir comme réponse à une recherche une combinaison d'offres, selon un profil utilisateur.

Dans ce projet, nous utilisons des ontologies pour modéliser les connaissances dans le modèle de domaine et pour modéliser des règles logiques qui représentent le savoir-faire de l'agence Côte d'Or Tourisme. L'utilisation de règles logiques et d'un moteur de règles nous a permis de proposer une extraction intelligente de la connaissance en utilisant le profil de l'utilisateur et les relations sémantiques entre concepts. Cet ensemble forme la couche sémantique de notre système.

La prise en compte des critères spatio-temporels dans la caractérisation de l'offre et dans la construction d'une combinaison pertinente d'offres a été étudiée. La pertinence est définie à la fois par des règles métiers et par une proximité géographique. Sur ce plan, le problème d'optimisation combinatoire lié aux objectifs de l'ADT Côte d'Or Tourisme a été identifié. Ce problème a été caractérisé comme une variante d'un problème de sac à dos. Pour répondre à ce problème, nous avons défini des métaheuristiques permettant la génération d'une combinaison pertinente d'items. Les règles identifient les connaissances pertinentes par rapport au profil de l'utilisateur et les algorithmes développent une stratégie pour construire une combinaison d'offres pertinentes. Cet ensemble forme la couche intelligence de notre système, elle est similaire à la couche adaptation d'un système hypermédia adaptatif

La gestion du profil de l'utilisateur, modélisant ses objectifs, goûts et contraintes, est réalisée au sein de la couche utilisateur. Cette couche permet d'identifier le profil et le contexte de consommation de l'offre touristique par l'utilisateur final. Dans ce projet, pour l'agence Côte d'Or Tourisme, cette couche est développée sous la forme d'une application mobile fonctionnant sur les plateformes Apple et Android.

L'ensemble est piloté par des outils de supervision qui représentent une pérennité de la pertinence du système tout au long de son cycle de vie. Ces outils sont utilisés par le personnel de Côte d'Or Tourisme pour définir son savoir-faire et affiner la pondération et l'écriture des règles et des profils selon l'évolution des offres touristiques en Bourgogne.

4 ORGANISATION DU DOCUMENT

Outre l'introduction et la conclusion, ce document est articulé en 5 chapitres : un état de l'art, la présentation d'une architecture générique de système de recommandation, la définition du problème d'optimisation de l'agence de tourisme, l'implémentation de cette architecture pour le domaine spécifique du tourisme et enfin le développement industriel du projet.

Le chapitre 2 est un état de l'art de différentes méthodes et techniques utilisées pour la résolution de notre projet de recherche. La première partie de ce chapitre présente le domaine des systèmes hypermédia adaptatifs. Cette partie met en évidence une architecture nécessaire et suffisante pour la réalisation de systèmes adaptatifs avec une description des couches principales. Trois couches se distinguent, elles représentent l'utilisateur, le domaine et l'adaptation. Le but des systèmes hypermédia adaptatifs est de résoudre le problème de surcharge cognitive (ou surcharge d'informations) et le problème de perte dans l'hyperespace. Un sous-domaine, chargé de résoudre uniquement le premier problème, est appelé « systèmes de recommandation ». Dans ces systèmes, présentés dans la seconde partie de ce chapitre, il existe deux principales méthodes de recommandations. La méthode basée sur le contenu recherche des documents, ou items (terme plus général), selon leurs caractéristiques et selon une requête ou le profil d'un utilisateur. La méthode par filtrage collaboratif recherche des documents en se basant sur les choix d'autres utilisateurs dans le système. Ce chapitre permet d'identifier les limites de ces méthodes par rapport à nos objectifs. Aucune méthode proposée ne répond à notre besoin de combiner plusieurs items selon des critères sémantiques et spatio-temporels pour en faire la recommandation à un utilisateur final. De plus, aucune ne permet au fournisseur de l'offre de traduire son expertise pour la recommandation.

Le chapitre 3 définit formellement un nouveau type de système de recommandation proposant des combinaisons d'individus d'une ontologie de domaine et permettant la prise en compte, sous la forme de règles métiers et de modèle de buts, du savoir-faire du fournisseur de contenus. Cette proposition est une extension des systèmes de recommandation basés sur le contenu utilisant une ontologie associée à des mécanismes de pondération et d'indexation. L'architecture utilisée est issue des systèmes hypermédia adaptatifs, elle est constituée de trois couches : une couche sémantique, une couche intelligence et une couche utilisateur. La constitution des différentes couches est décrite et les différents processus de la couche intelligence permettant la génération des propositions sont expliqués. Pour proposer des recommandations sous forme de combinaisons, le système est constitué de manière à pouvoir utiliser des algorithmes d'optimisation combinatoire adaptés au domaine d'application.

Le chapitre 4 a pour objectif de définir le problème d'optimisation du domaine du tourisme et de proposer des pistes pour la résolution de ce problème en utilisant des métaheuristiques. Ce chapitre définit le problème de combinaison d'objets touristiques hétérogènes comme une variante d'un problème de sac à dos. Ce chapitre présente une étude sur le domaine de l'optimisation combinatoire et présente les concepts nécessaires à la résolution des problèmes d'optimisation. Comme le problème d'optimisation touristique n'est pas résoluble

par des heuristiques classiques en un temps raisonnable, une dernière partie présente les métaheuristiques les plus connues pour résoudre des problèmes d'optimisation combinatoire classiques.

Le chapitre 5 présente l'instanciation au domaine de l'ADT Côte d'Or Tourisme, du système de recommandation générique que nous avons définie. Cette instanciation a pour but de fournir à un utilisateur donné une combinaison d'offres touristiques sous forme de séjour répondant aux critères du problème posé dans le chapitre 4. Le chapitre 5 décrit l'implémentation des différentes couches de notre architecture de recommandation. Tout d'abord, l'implémentation de la couche sémantique est détaillée, incluant principalement le processus de création de l'ontologie de domaine basé sur la base de données de l'entreprise et la connaissance d'experts du tourisme. Ensuite, il est montré l'implémentation de la couche utilisateur contenant les différents paramètres non dépendants du domaine retenus pour la modélisation de l'utilisateur. Enfin, l'implémentation de la couche intelligence est expliquée en se focalisant sur la partie combinatoire, car elle est particulière à l'application touristique à réaliser, contrairement à la partie de pondération qui est définie de manière générique dans le chapitre 3. Trois algorithmes sont testés pour la génération de combinaisons: un algorithme basé sur le recuit simulé, l'algorithme Hill-Climbing et un algorithme multi-objectif.

Le chapitre 6 présente le développement industriel de ces travaux de recherche. Ce chapitre présente la modélisation de l'offre touristique dans la base de données, l'infrastructure d'un point de vue technique et le développement d'une application iPhone. Cette application mobile est découpée en 3 phases principales : une phase de définition du profil de l'utilisateur, une phase de génération et visualisation de la combinaison d'items et des items indépendamment, et enfin, une phase de modification totale ou partielle des offres si besoin. Dans cette partie consacrée aux applications mobiles, la structure technique de l'application iPhone est abordée. Une dernière partie présente un outil de supervision qui permet à l'ADT Côte d'Or Tourisme d'affiner la pertinence des résultats et/ou d'orienter les résultats selon ses besoins.

Chapitre 2

Etat de l'art

Résumé

Ce chapitre est un état de l'art de différentes méthodes et techniques utilisées pour la résolution de notre projet de recherche. La première partie de ce chapitre présente le domaine des systèmes hypermédia adaptatifs. Cette partie met en évidence une architecture nécessaire et suffisante pour la réalisation de systèmes adaptatifs avec une description des couches principales. Trois couches se distinguent, elles représentent l'utilisateur, le domaine et l'adaptation. Le but des systèmes hypermédia adaptatifs est de résoudre le problème de surcharge cognitive (ou surcharge d'informations) et le problème de perte dans l'hyperespace. Un sous-domaine, chargé de résoudre uniquement le premier problème, est appelé « système de recommandation ». Dans ce type de systèmes, présentés dans la seconde partie de ce chapitre, il existe deux principales méthodes de recommandation. La méthode basée sur le contenu recherche des documents, ou items (terme plus général), en se basant sur ses caractéristiques et selon une requête ou le profil d'un utilisateur. La méthode par filtrage collaboratif recherche des documents en se basant sur les choix d'autres utilisateurs dans le système.

Ce chapitre permet, en outre, d'identifier les limites des solutions existantes par rapport à nos objectifs. Aucune ne répond exactement à notre besoin de combiner plusieurs items selon des critères sémantiques et spatio-temporels pour en faire la recommandation à un utilisateur final. De plus, aucune ne permet au fournisseur de l'offre de traduire son expertise pour la recommandation.

Plan

1	Systèmes hypermédia adaptatifs	31
1.1	Architecture des systèmes hypermédia adaptatifs	33
1.2	Représentation du domaine et de l'utilisateur	36
1.3	Conclusion	37
2	Systèmes de recommandation	37
2.1	Systèmes de recommandation basés sur le contenu	40
2.1.1	Techniques de représentation et de recommandation	40
2.1.2	Recommandations basées sur les vecteurs de mots-clefs	41
2.1.3	Recommandations basées sur la sémantique	44

2.2	Systèmes de recommandation basés sur une approche collaborative	46
2.2.1	Recommandations basées sur le voisinage	47
2.2.1.1	Recommandations basées sur le voisinage utilisateur	47
2.2.1.2	Recommandations basées sur le voisinage des items	48
2.2.2	Recommandations basées sur un modèle prédictif	49
2.3	Conclusion	50
3	Conclusion sur le chapitre	51

L'adaptation de l'information à l'utilisateur final est un point clé dans ce projet de recherche. Elle s'inscrit pleinement dans les thématiques de recherche du projet transversal Checksem puisqu'il est considéré comme une forme d'interopérabilité entre des systèmes d'information hétérogènes et un système d'information particulier : l'utilisateur. Cette forme d'interopérabilité a fait l'objet de nombreux travaux de recherche ces 15 dernières années qui ont permis la définition de nombreux systèmes que l'on peut généralement nommer systèmes hypermédia adaptatifs. Dans la suite de ce document, nous verrons que sous cette terminologie, il existe de nombreuses architectures et définitions. Néanmoins, nous identifierons très vite une architecture pivot qui nous servira de « squelette » pour construire notre proposition. Cette architecture est composée de trois modèles : le modèle de domaine qui contient les connaissances du système d'information, le modèle d'adaptation, qui contient les mécanismes d'extraction et de combinaison des connaissances et le modèle utilisateur qui contient la description du profil des utilisateurs.

Après avoir présenté les principaux systèmes hypermédia adaptatifs dans la section 1 de ce chapitre, nous nous intéresserons en particulier à un sous-domaine très spécifique appelé système de recommandation. Ce type de systèmes, moins généralistes que les systèmes hypermédia adaptatifs, se focalise sur la résolution du problème de surcharge cognitive qui est l'un des verrous du domaine d'application de notre projet. La section 2 de ce chapitre présente un état de l'art des systèmes de recommandation.

D'une manière générale, la tendance actuelle de ces systèmes adaptatifs est d'utiliser de la connaissance plutôt que de simples informations stockées dans un système de gestion de base de données. Pour atteindre cet objectif, ils utilisent de plus en plus les technologies du Web sémantique qui devient une base solide pour la création du Web de demain.

1 SYSTEMES HYPERMEDIA ADAPTATIFS

Dans les premières années qui ont suivi sa création, le Web était le même pour tout le monde. Les sites Web présentaient la même information et les mêmes liens à tous les visiteurs, sans prendre en compte leurs objectifs et leur connaissance. Étant donné l'accroissement des informations disponibles sur le Web, la diversité de ses utilisateurs et la complexité des applications Web, l'approche « one-size-fits-all » (la même chose pour tout le monde) du Web a été remise en question. Ainsi est apparue la proposition de systèmes hypermédia adaptatifs.

Selon la vision de Vannevar Bush, l'hypermédia a changé la manière de penser, d'accéder, et de traiter l'information. Le succès du Web a facilité la publication mondiale d'importantes quantités d'informations interconnectées, permettant à des utilisateurs hétérogènes de les parcourir de manière non-linéaire. Néanmoins, la rapide croissance de cet ensemble d'information et l'hétérogénéité de ses utilisateurs montrent un problème important dans les systèmes hypermédia : ils proposent le même contenu des pages et le même ensemble de liens à tous les utilisateurs. Il devient alors évident qu'il est nécessaire d'ajuster les systèmes hypermédia aux utilisateurs pour faciliter leur navigation à travers un espace d'information. Cette nécessité est abordée par les systèmes hypermédia adaptatifs (Brusilovsky, 1996) (De Bra, Aerts, Smits, & Stash, 2002). Les systèmes hypermédia adaptatifs sont devenus très populaires au début des années 1990 pour permettre aux utilisateurs d'accéder à de l'information personnalisée. Ces systèmes sont utiles dans tout domaine où les applications sont utilisées par des utilisateurs aux objectifs et connaissances différents. D'après Brusilovsky (Brusilovsky, 1996), le cœur de l'adaptation peut être défini en posant et répondant à six questions majeures :

- Que peut-on adapter ? (Quoi ?)
- À quoi pouvons-nous nous adapter ? (À quoi ?)
- Pourquoi l'adaptation est-elle voulue ? (Pourquoi ?)
- Où peut-on faire l'adaptation ? (Où ?)

- Quand peut-on faire l'adaptation ? (Quand ?)
- Comment se fait l'adaptation ? (Comment ?)

Les architectures des systèmes hypermédia adaptatifs se basent sur ces questions fondamentales. Dans les systèmes hypermédia adaptatifs, plusieurs niveaux d'adaptation ont été établis (Stash, 2007). Selon la classification la plus récente, nous pouvons noter deux niveaux d'adaptation :

- Adaptation du contenu et de la présentation.
- Adaptation des liens de navigation et support de navigation.

En effet, la représentation d'un hypermédia sous la forme d'un graphe, permet d'identifier ces deux types d'adaptations : soit adapter ses nœuds (adaptation du contenu), soit adapter ses arêtes (adaptation de la navigation). Pour chaque niveau d'adaptation, des méthodes et techniques d'adaptation ont été établies. Une méthode est définie comme une notion d'adaptation qui peut être présentée à un niveau conceptuel. Une technique est un moyen pour implémenter une méthode spécifique. Les techniques opèrent sur le contenu actuel des informations et sur la présentation des liens hypertextes. Il peut être possible d'implémenter la même méthode à travers différentes techniques et d'utiliser la même technique pour différentes méthodes. À partir de ce constat, il est possible de résumer les technologies de l'hypermédia adaptatif sous la forme d'une arborescence (Figure 2).

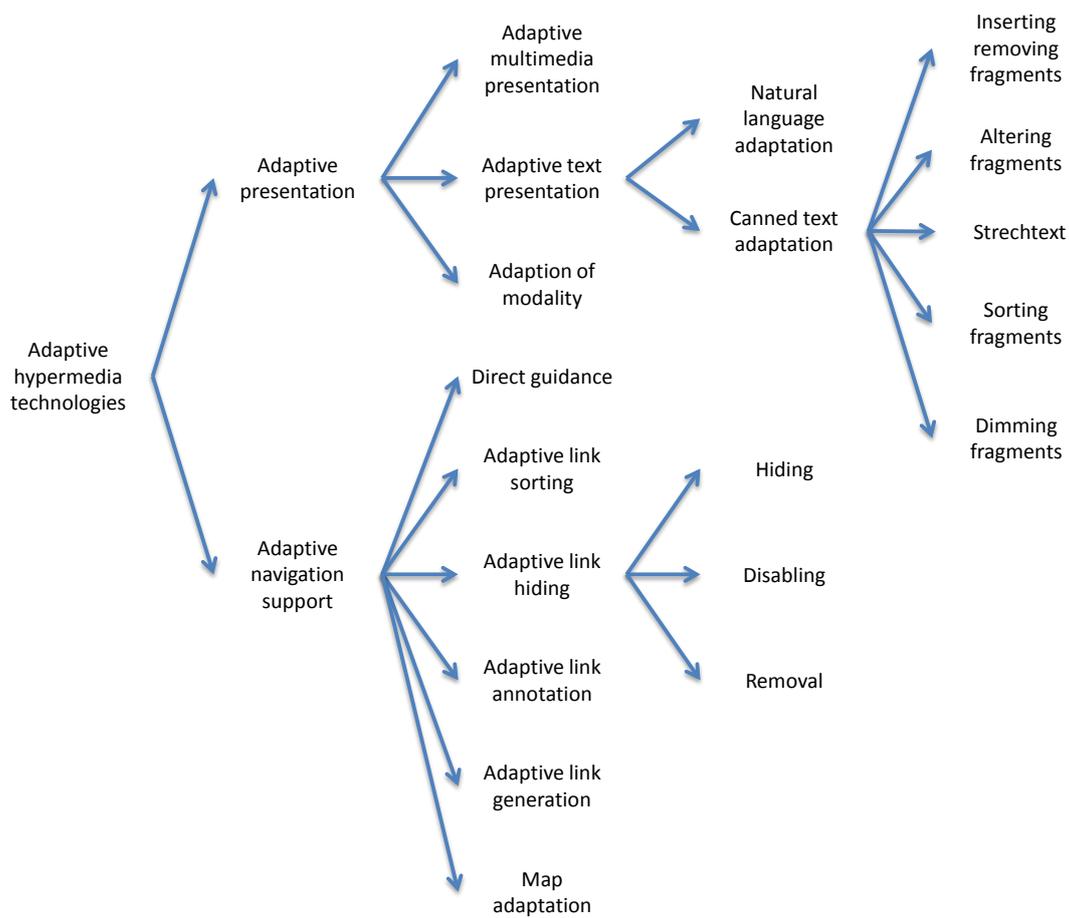


FIGURE 2. TECHNOLOGIES DE L'HYPERMEDIA ADAPTATIF

Plusieurs architectures ont été proposées depuis l'avènement du domaine des systèmes hypermédia adaptatifs. La suite de cette présentation des systèmes hypermédiés adaptatifs est articulée en deux parties. D'abord, les architectures et les modèles les plus connus et influents du domaine sont présentés ; puis, une

comparaison des trois méthodes les plus utilisées pour la modélisation du domaine de connaissance et de l'utilisateur est réalisée.

1.1 ARCHITECTURE DES SYSTEMES HYPERMEDIA ADAPTATIFS

Le modèle Dexter (Halasz & Schwartz, 1990) a été le premier essai pour unifier et clarifier les technologies de l'hypermédia. La principale caractéristique de ce modèle, qui depuis est devenu une référence, est la division de l'application hypermédia en trois couches : composante interne (within-component layer), stockage (storage layer) et exécution (run-time layer). Ces différentes couches sont présentées dans la Figure 3. La première couche représente le contenu comme des éléments indépendants. La seconde couche représente les liaisons des éléments hypertextes d'une application. Et la troisième couche représente le fonctionnement de l'application.

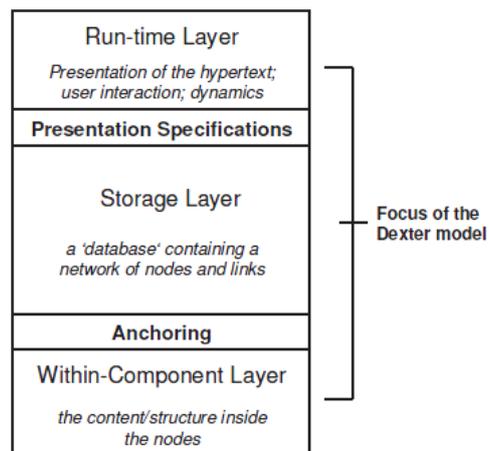


FIGURE 3. MODELE DEXTER

Basé sur le modèle Dexter, un premier modèle de référence pour les applications hypermédia adaptatives dans le domaine de l'e-learning a été défini en 1999, il se nomme AHAM (Adaptive Hypermedia Application Model) (De Bra, Houben, & Wu, 1999). Il a été implémenté dans le système générique AHA! (Adaptive Hypermedia Architecture) (De Bra & Calvi, 1998). Il est basé sur un découpage explicite en trois modèles, repris dans de nombreux autres systèmes depuis : le modèle du domaine, le modèle utilisateur et le modèle d'adaptation (voir la Figure 4). Le modèle de domaine décrit comment le domaine d'application est structuré, à la fois au niveau conceptuel et au niveau des fragments d'information et des pages. Le modèle utilisateur décrit la connaissance de l'utilisateur sur les concepts du modèle de domaine. Le modèle d'adaptation contient des mécanismes d'adaptation de contenus et de liens en se basant sur le modèle utilisateur et le modèle de domaine. Il peut aussi mettre à jour le modèle de l'utilisateur en fonction du comportement de ce dernier. Ce modèle de référence a unifié la communauté de recherche en hypermédia adaptatif et fournit une architecture générique.

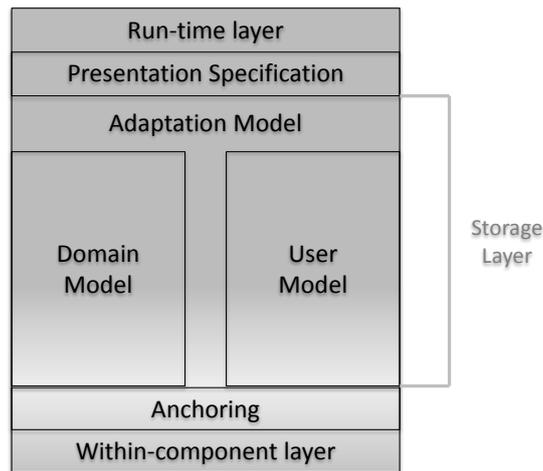


FIGURE 4. AHAM

Ces premiers modèles ont permis de fixer l'idée générale des systèmes hypermédia adaptatifs. Néanmoins, c'est en 2002 avec le modèle Munich (Koch & Wirsing, 2002) qu'une formalisation standard et précise a été réalisée à l'aide des langages UML (Unified Modeling Language) et d'OCL (Object Constraint Language). L'architecture décrite dans le Munich Reference Model (Figure 5) est assez proche de celle du modèle AHAM.

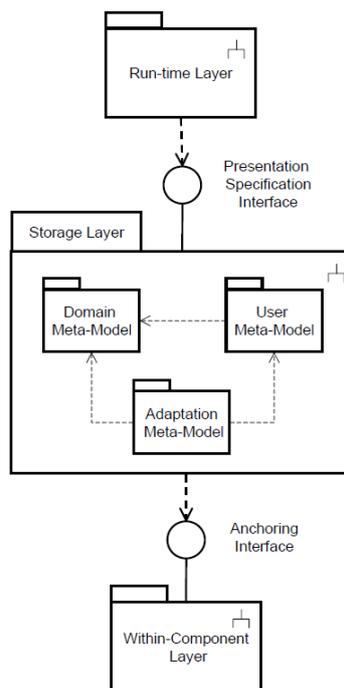


FIGURE 5. MUNICH

Le modèle LAOS (Cristea & de Mooij, 2003), est un modèle pour la création de systèmes hypermédia adaptatifs dans le domaine de l'e-learning. Il décompose les systèmes hypermédia adaptatifs en un plus grand nombre de couches (Figure 6) : un modèle de domaine, un modèle de buts et contraintes, un modèle utilisateur, un modèle d'adaptation et un modèle de présentation.

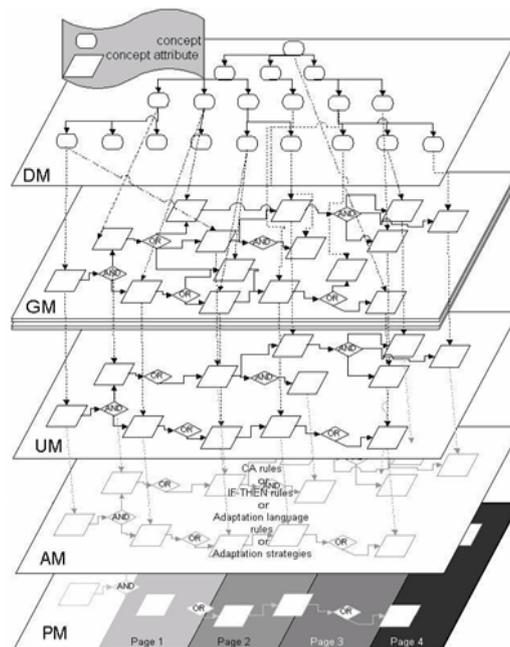


FIGURE 6. LAOS

Le modèle de domaine (DM) est composé de schémas conceptuels composés de concepts, relations et attributs attachés aux concepts. Ce modèle représente les ressources pour l'apprentissage et leurs caractéristiques. Le modèle de buts et des contraintes (GM) filtre, regroupe et restructure le modèle de domaine en fonction des objectifs d'instruction. L'interprétation réelle de cette structure est assurée par le modèle d'adaptation. Le modèle utilisateur (UM) représente l'utilisateur sous la forme d'un schéma. Le modèle d'adaptation (AM) est découpé en 3 niveaux : Un niveau dit "bas niveau" permettant d'agir directement sur les éléments à adapter par les techniques courantes d'adaptation ; un niveau intermédiaire permettant de grouper les éléments du niveau précédent pour créer des mécanismes d'adaptations et de construction, il en résulte un langage de programmation ; un niveau dit "haut niveau", permettant l'utilisation de fonctions. Enfin, le modèle de présentation (PM) prend en compte les propriétés physiques et l'environnement de présentation pour la génération réelle de code sur différentes plateformes. Dans sa construction, le modèle de présentation est similaire au modèle utilisateur. Il combine une surcouche sur le modèle de domaine, le modèle de buts et des contraintes, avec des variables indépendantes et leurs valeurs respectives. Cette structure permet d'attacher à chaque concept un certain type de représentation à l'écran, tout en conservant en même temps des types de représentation indépendants associés aux valeurs courantes du modèle de l'utilisateur.

Malgré ces tentatives de définition d'un modèle générique des systèmes hypermédia adaptatifs, les nouveaux termes, concepts, méthodologies, et prototypes issus des nouveaux systèmes ont engendré de nouvelles situations, montrant des formes d'utilisation qui ne sont pas prises en compte par les modèles. Ces nouvelles formes d'utilisation sont, par exemple, la prise en compte de groupes d'utilisateurs, d'une adaptation de haut niveau ou d'un corpus ouvert.

Il n'y a pas encore unanimité quant à l'architecture idéale des systèmes hypermédia adaptatifs, néanmoins, il est admis que l'adaptation dans une application dépend de trois facteurs majeurs :

- L'application doit être basée sur un modèle de domaine. Ce modèle est la représentation conceptuelle d'un domaine d'application. Il indique les relations entre concepts et la façon dont ils sont connectés aux contenus de présentation en termes de fragments, pages (De Bra, Wu, & Houben, 2000), chapitres, unités d'information (Henze, 2000), ou autres structures encapsulant de l'information sur

un concept. Le modèle de domaine fournit l'information et la structure du domaine qui a besoin d'être adapté, en reliant des concepts au contenu correspondant.

- Un modèle utilisateur doit être créé et gardé à jour pour représenter la connaissance, les intérêts, les préférences, les buts et objectifs, l'historique des actions, le type, le style et autres propriétés qui pourraient être utiles pour l'adaptation. Le modèle utilisateur fournit les données de l'utilisateur et de l'utilisation en utilisant les informations du modèle de domaine. Ainsi, le modèle utilisateur fournit souvent des informations sur les objectifs de l'utilisateur par l'utilisation de la même structure conceptuelle.
- Le système doit adapter la présentation, le contenu de l'information et la structure de navigation en fonction des connaissances, des intérêts, du style de navigation, des buts, des objectifs, etc. de l'utilisateur. Pour répondre à ce besoin, un modèle d'adaptation doit être construit. Il indique la manière dont les relations entre concepts du modèle de domaine affectent la navigation de l'utilisateur et les mises à jour de propriétés (par exemple la façon dont le système devrait guider l'utilisateur vers ou à l'opposé d'informations liées à certains concepts). Ce modèle peut aussi interpréter des contraintes sur une structure de relations du modèle de domaine.

Cette division en modèle de domaine, modèle utilisateur et modèle d'adaptation fournit une séparation claire des parties majeures des systèmes hypermédia adaptatifs. Ces couches sont nécessaires et suffisantes pour la création d'un système hypermédia adaptatif, mais, selon le niveau d'adaptation, de flexibilité et de portabilité désiré, des couches supplémentaires peuvent être ajoutées (modèle de buts, modèle de groupe, modèle d'adaptation de haut niveau, modèle de présentation...).

1.2 REPRESENTATION DU DOMAINE ET DE L'UTILISATEUR

La modélisation du domaine et de l'utilisateur sont intimement liés dans les systèmes hypermédia adaptatifs. Une méthode populaire, utilisée par de nombreux systèmes pour représenter le domaine et l'utilisateur, est la représentation par vecteurs de mots-clefs (Lieberman, 1995) (Kamba, Sakagami, & Koseki, 2007). Cette méthode considère que chaque document et utilisateur est décrit par un ensemble de vecteurs de mots-clefs pondérés. Dans le modèle d'adaptation, les poids sont utilisés pour calculer le degré de similarité entre deux vecteurs et ainsi proposer des documents pertinents à l'utilisateur. Bien que la représentation par vecteur de mots-clefs soit simple et efficace, cette méthode ne permet pas de conserver le sens final du texte, d'où la perte d'une grande quantité d'information durant la phase de représentation.

Pour améliorer cette expressivité, une méthode très connue est la représentation par réseaux sémantiques. Dans ces réseaux sémantiques, chaque nœud représente un concept. Minio et Tasso (Minio & Tasso, 1996) présentent une approche basée sur des réseaux sémantiques où chaque nœud contient un mot particulier du corpus et où des arcs sont créés selon les cooccurrences des mots dans les documents. Chaque document du domaine est représenté de cette façon. Dans des systèmes simples utilisant seulement un seul réseau sémantique pour modéliser l'utilisateur, chaque nœud contient seulement un mot-clef. Des mots-clefs sont extraits des pages pour lesquelles l'utilisateur a montré un intérêt. Ensuite, ils sont traités pour ne garder que les plus pertinents et sont pondérés afin d'éliminer ceux qui ont un poids plus petit qu'un certain seuil. Les mots-clefs sélectionnés sont ensuite ajoutés au réseau sémantique où chaque nœud représente un mot-clef et les arcs représentent les cooccurrences dans les documents. Avec cette méthode, il est possible d'évaluer la pertinence d'un document par rapport à un profil utilisateur. En effet, il suffit de construire un réseau sémantique du document et de le comparer avec le réseau sémantique de l'utilisateur pour classer le document en « intéressant », « inintéressant » ou « indifférent ». Par rapport à l'approche par vecteur de mots-clefs, les réseaux sémantiques permettent une représentation plus complexe et complète des documents et de l'utilisateur. Néanmoins, représenter un document ou un utilisateur par des mots et des relations de

cooccurrences entre ces mots est quelque peu réducteur, cela ne peut exprimer toute la connaissance (de l'utilisateur ou induite du document).

Étant donné ces limitations, une approche plus récente, qui se répand de plus en plus, est l'approche ontologique. Elle est similaire à l'approche par réseau sémantique dans le sens où elle utilise aussi des nœuds et relations entre ces nœuds. Néanmoins, dans les ontologies, les nœuds représentent des concepts au lieu de mots ou ensembles de mots. De plus, les liens ne sont pas seulement des cooccurrences entre des mots, mais ont de nombreuses significations. L'utilisation d'ontologies permet de garder un maximum d'information par rapport aux deux approches précédentes. En effet, la possibilité d'intégrer toutes sortes de relations ou de concepts dans les ontologies permet une représentation complète d'une connaissance, contrairement aux deux précédentes approches dont la structure de représentation limite trop l'expressivité. Dans QuickStep (Middleton, Alani, Shadbolt, & De Roure, 2002), l'ontologie est utilisée pour le domaine de la recherche pour indexer des articles. L'ontologie a été créée par des experts du domaine. Les concepts de l'ontologie sont représentés comme des vecteurs d'exemples d'articles. Les papiers des utilisateurs, à partir de leur liste de publications, sont modélisés comme vecteurs caractéristiques et sont reliés aux concepts en utilisant l'algorithme des plus proches voisins. Chaque concept est pondéré par le nombre de papiers qui lui sont reliés et représente ainsi l'intérêt de l'utilisateur pour ce concept. Les recommandations sont faites à partir des corrélations entre les intérêts de l'utilisateur sur les concepts et les papiers associés à ces concepts. Dans (Cantador & Castells, 2006) et (Sieg, Mobasher, & Burke, 2007), une ontologie prédéfinie est utilisée pour modéliser le domaine. Le profil de l'utilisateur est représenté par un ensemble de concepts pondérés où les poids représentent l'intérêt de l'utilisateur pour ces concepts. Ses intérêts sont déterminés par analyse de son comportement.

1.3 CONCLUSION

Dans cette partie, nous avons tout d'abord présenté les principaux modèles de référence dans le domaine des systèmes hypermédia adaptatifs. Il n'existe pas d'architecture générique pour définir un système hypermédia adaptatif. Néanmoins, il apparaît que trois couches sont nécessaires et suffisantes pour sa création: une couche pour définir le domaine, une couche pour définir l'utilisateur et une pour l'adaptation. Cette division fournit une séparation claire des parties majeures des systèmes hypermédia adaptatifs. Il est à noter que selon le niveau d'adaptation, de flexibilité et de portabilité désiré, des couches supplémentaires peuvent être ajoutées, à l'instar du modèle LAOS (modèle de buts, modèle de présentation).

Ensuite, nous avons identifié trois méthodes majeures de modélisation de la connaissance et de l'utilisateur utilisées dans la littérature. Un consensus sur l'adoption des technologies du Web sémantique pour construire ces modèles semble émerger. Néanmoins, pour répondre aux contraintes de notre projet, le développement d'un système hypermédia généraliste n'est pas adapté.

Dans la suite de ce chapitre, nous présentons une étude sur les systèmes de recommandation qui sont un sous domaine des systèmes hypermédia adaptatifs. Les systèmes de recommandation sont moins généralistes et se focalisent sur la résolution le problème de surcharge cognitive que nous souhaitons résoudre dans le cadre de notre projet.

2 SYSTEMES DE RECOMMANDATION

Les systèmes de recommandation (SR) sont des outils et techniques logiciels fournissant des suggestions d'items à un utilisateur (Burke, 2007) (Mahmood & Ricci, 2009). Les suggestions se rapportent à des processus décisionnels variés. Cela peut être des propositions d'items à acheter, de musique à écouter, ou de news à lire en ligne.

« Item » est le terme général utilisé pour dénoter ce que le système recommande aux utilisateurs. Un système de recommandation se focalise normalement sur un type spécifique d'item (par exemple, des CDs ou news) et en conséquence, son modèle de navigation, son interface graphique, et la technique de recommandation de base utilisée pour générer les recommandations sont tous personnalisés pour fournir des suggestions utiles et efficaces pour un type particulier d'item.

Les systèmes de recommandation sont essentiellement orientés vers les individus qui n'ont pas suffisamment d'expérience personnelle ou de compétences pour évaluer la quantité potentiellement immense d'items alternatifs qu'un site Web, par exemple, peut offrir (Resnick & Varian, *Recommender systems*, 1997). Un bon exemple est un système de recommandation de livres qui aide les utilisateurs dans leur choix d'un livre à lire. Dans le site Web populaire d'Amazon.com, un système de recommandation est utilisé pour personnaliser la boutique en ligne à chaque client (Jannach, 2006). Puisque les recommandations sont habituellement personnalisées, différents utilisateurs ou groupes d'utilisateurs reçoivent des suggestions diverses. En outre, il y a aussi des recommandations non-personnalisées. Celles-ci sont beaucoup plus simples à générer et sont généralement publiées dans des magazines ou des journaux. Des exemples typiques sont les « tops dix » des ventes des livres, CDs, etc. Même s'ils peuvent être utiles et efficaces dans certaines situations, ces types de recommandations non personnalisées ne sont généralement pas abordés par la recherche du domaine des systèmes de recommandation.

Dans leur forme la plus simple, les recommandations personnalisées sont des listes triées d'items. Dans l'accomplissement de ce classement, les systèmes de recommandation tentent de prédire quels sont les produits ou services les plus adaptés aux utilisateurs, basés sur leurs préférences et contraintes. Afin d'effectuer une telle tâche de calcul, les systèmes de recommandation collectent les préférences des utilisateurs, qui sont soit données de façon explicite (par exemple en donnant des notes à des produits), soit de manière implicite, inférées par interprétation des actions de l'utilisateur. Par exemple, un système de recommandation peut considérer la navigation vers la page d'un produit particulier comme un signe implicite de préférence pour les items de cette page.

Le développement des systèmes de recommandation s'est initié à partir d'une observation assez simple: les individus s'appuient souvent sur les recommandations des autres pour la prise de décisions quotidiennes (Mahmood & Ricci, 2009) (McSherry & Mironov, 2009). Par exemple, il est commun de s'appuyer sur ce que nos semblables recommandent lors du choix d'un livre à lire; les employeurs comptent sur les lettres de recommandation pour leurs décisions de recrutement; et pour la sélection de films à regarder, les individus tendent à lire et se fier aux critiques de films qui apparaissent dans leur journal.

En cherchant à mimer ce comportement, les premiers systèmes de recommandation ont appliqué des algorithmes tirant parti des recommandations d'une communauté d'utilisateurs afin de délivrer des recommandations à un utilisateur voulant des suggestions. Les recommandations portaient sur des items que des utilisateurs similaires (avec des goûts similaires) avaient aimés. Cette approche est dite filtrage collaboratif et son raisonnement est basé sur la règle suivante : si des contenus ont intéressé une majorité de membres d'une même communauté, alors ils sont susceptibles d'attirer les autres membres de la communauté.

Étant donné le développement des sites Web de commerce électronique, un besoin important est apparu pour fournir des recommandations issues du filtrage de tout l'éventail des alternatives possibles. Les utilisateurs éprouvaient des difficultés à trouver le choix le plus approprié parmi l'immense variété d'items (produits et services) que ces sites Web offraient.

Sur le Web, la croissance exponentielle des informations disponibles, ainsi que leur forte hétérogénéité et l'introduction rapide de nouveaux services de e-business ont très souvent débordé les utilisateurs, les amenant à prendre de mauvaises décisions. En effet, avoir du choix avec ses implications en termes de liberté,

autonomie, et autodétermination peut devenir excessif. Le sentiment de liberté peut être éventuellement considéré comme une forme de tyrannie (Schwartz, 2004).

Les systèmes de recommandation ont prouvé ces dernières années qu'ils sont un bon moyen pour faire face au problème de surcharge cognitive (ou surcharge d'information). Pour résoudre ce problème, le système de recommandation met en avant des items inconnus qui peuvent être pertinents pour les utilisateurs. Ce niveau de pertinence est déterminé par le système en fonction de connaissances variées (profil de l'utilisateur, contexte de consommation, items disponibles, historique des transactions, feedbacks d'autres utilisateurs sur l'item...). L'utilisateur peut alors parcourir les recommandations. Il peut les accepter ou non et peut fournir, immédiatement ou plus tard, un feedback implicite ou explicite. Toutes les actions et les feedbacks des utilisateurs peuvent être enregistrés dans la base de données et utilisés pour générer de nouvelles recommandations pour les prochaines interactions avec le système.

L'étude des systèmes de recommandation est relativement nouvelle comparée aux recherches sur les systèmes d'information classiques (les bases de données ou moteurs de recherche). Les systèmes de recommandation sont devenus un domaine de recherche indépendant dans le milieu des années 1990s (Goldberg, Nichols, Oki, & Terry, 1992). Ces dernières années, avec le développement du Web et plus particulièrement des plateformes d'e-commerce, l'intérêt pour les systèmes de recommandation a considérablement augmenté. Ils ont un rôle important dans des sites Web très bien côtés tels qu'Amazon.com, YouTube, Netflix, Yahoo, Tripadvisor, Last.fm, et IMDb. De plus, de nombreuses sociétés sont en train de développer et déployer des systèmes de recommandation dans le cadre des services qu'ils fournissent à leurs abonnés. Par exemple, Netflix, le service de location de vidéo en ligne, a décerné un prix d'un million de dollars à l'équipe qui a réussi à améliorer sensiblement les performances de son système de recommandation (Koren, Bell, & Volinsky, 2009). Des conférences et workshops sont dédiés à ce domaine de recherche. Nous pouvons par exemple citer ACM Recommender Systems (RecSys), créé en 2007 qui est devenu rapidement le principal événement annuel dans la recherche sur la recommandation. En outre, des sessions spéciales dédiées aux systèmes de recommandation sont fréquemment incluses dans les traditionnelles conférences sur les bases de données, systèmes d'informations et systèmes adaptatifs. Les tutoriaux sur les systèmes de recommandation sont très populaires dans les conférences en informatique.

Pour avoir un premier aperçu des différents types de systèmes de recommandation, nous pouvons citer une taxonomie fournie dans (Burke, 2007) qui est devenue une référence pour distinguer ces systèmes. Deux approches principales sont définies, celles basées sur le contenu et celles basées sur le filtrage collaboratif.

- **Basée sur le contenu** : Le système recommande des items qui sont similaires à ceux que l'utilisateur a aimés dans le passé. La similarité des items est calculée en se basant sur les caractéristiques associées aux items comparés. Par exemple, si l'utilisateur a noté positivement un film qui appartient au genre « comédie », alors le système peut fournir des recommandations de films de ce genre.
- **Filtrage collaboratif** : L'implémentation la plus simple et originale de cette approche (Schafer, Frankowski, Herlocker, & Sen, 2007) est de recommander à un utilisateur actif les items que d'autres utilisateurs avec des goûts similaires ont aimés dans le passé. La similarité de goût entre deux utilisateurs est calculée en se basant sur la similarité de leur historique de notation. C'est la raison pour laquelle (Schwartz, 2004) se réfère au filtrage collaboratif par l'expression « corrélation de personne à personne ». Le filtrage collaboratif est la technique la plus populaire et la plus répandue dans les systèmes de recommandation.

Dans cette introduction, nous avons brièvement abordé les idées et concepts de bases des systèmes de recommandation. Les principales approches ont été énoncées. Elles sont détaillées dans les parties qui suivent.

2.1 SYSTEMES DE RECOMMANDATION BASES SUR LE CONTENU

Les systèmes de recommandation ont pour objectif d'orienter les utilisateurs d'une manière personnalisée vers des objets intéressants issus d'un large espace d'options possibles. Les systèmes de recommandation basés sur le contenu essaient de recommander des items similaires à ceux aimés par l'utilisateur dans le passé. En effet, le processus principal réalisé par un système de recommandation basé sur le contenu consiste à faire correspondre les attributs d'un profil utilisateur (où les préférences et intérêts sont stockés) avec les attributs des items, dans le but de recommander à l'utilisateur de nouveaux objets intéressants.

La recherche sur les systèmes de recommandation basés sur le contenu prend place à l'intersection de différents domaines de recherche en informatique, notamment la Recherche d'Information (RI) et l'Intelligence Artificielle (Baeza-Yates & Ribeiro-Neto, 1999). En Recherche d'Information, il est considéré que les internautes voulant des recommandations sont engagés dans un processus de recherche d'information. L'utilisateur exprime un besoin ponctuel en donnant une requête (habituellement une liste de mots-clés). Dans les systèmes de filtrage d'information (FI), le besoin est représenté par le profil de l'utilisateur. Les items à recommander peuvent être très différents, en fonction du nombre et du type des attributs utilisés pour les décrire.

En intelligence artificielle, la tâche de recommandation peut être exprimée comme un problème d'apprentissage qui exploite la connaissance passée des utilisateurs. De manière simple, les profils des utilisateurs sont sous la forme de vecteurs de mots-clés et reflètent les intérêts à long terme de l'utilisateur. Souvent, il est préférable que le système apprenne le profil de l'utilisateur plutôt que d'imposer à celui-ci de le fournir. Cela implique généralement l'application de techniques de Machine Learning (ML). Leur but est d'apprendre à catégoriser de nouvelles informations en se basant sur les informations précédemment vues, et qui ont été libellées implicitement ou explicitement comme intéressant ou non par l'utilisateur. Avec ces libellés, les méthodes de Machine Learning sont capables de générer un modèle prédictif qui, étant donné un nouvel item, va aider à décider le degré d'intérêt que peut porter l'utilisateur pour l'item.

Cette partie sur les systèmes de recommandation basés sur le contenu propose un état de l'art de ces derniers en abordant notamment les techniques de représentation des items et les algorithmes de recommandation utilisés.

2.1.1 TECHNIQUES DE REPRESENTATION ET DE RECOMMANDATION

Les items qui peuvent être recommandés aux utilisateurs sont représentés par un ensemble de caractéristiques, aussi appelées attributs ou propriétés. Par exemple, dans une application de recommandation de films, les caractéristiques adoptées pour décrire un film sont: acteurs, directeurs, genres, sujet... Quand chaque item est décrit avec le même ensemble d'attributs, et qu'il y a un ensemble connu de valeurs que les attributs peuvent prendre, l'item est représenté au moyen d'une donnée structurée.

Dans la plupart des systèmes de filtrage basés sur le contenu, les descriptions des items sont des caractéristiques textuelles extraites à partir de pages Web, emails, articles de news ou descriptions de produits. Contrairement aux données structurées, il n'y a pas d'attribut avec des données bien définies. À cause de la naturelle ambiguïté du langage, la construction d'un profil utilisateur par analyse de caractéristiques textuelles engendre de nombreuses complications. Les profils basés sur mots-clés traditionnels ne sont pas capables de capturer la sémantique des intérêts des utilisateurs, car ils sont essentiellement générés par une opération de correspondance de chaînes. Si une chaîne de caractères, ou une variante morphologique est trouvée à la fois dans le profil et dans le document, une correspondance est faite et le document est considéré comme approprié. La correspondance de chaîne souffre des problèmes de polysémie (multiple sens pour un mot) et synonymie (mots différents avec le même sens). La polysémie rend pertinents de mauvais documents et la synonymie ne permet pas au système d'identifier toutes les informations pertinentes si le profil ne contient pas

les mots-clefs exacts du document. La gestion de ces deux problèmes nécessite le développement de techniques d'analyse sémantique.

2.1.2 RECOMMANDATIONS BASEES SUR LES VECTEURS DE MOTS-CLEFS

La plupart des systèmes de recommandation utilisent de simples modèles de recherche, comme la correspondance de mots-clefs ou le Modèle d'Espace Vectoriel (MEV) avec la pondération basique TF-IDF (Term Frequency-Inverse Document Frequency). MEV est une représentation spatiale des documents textuels. Dans ce modèle, chaque document est représenté par un vecteur de dimension n , où chaque dimension correspond à un terme de l'ensemble du vocabulaire d'une collection de documents. Formellement, tout document est représenté par un vecteur poids sur des termes, où chaque poids indique le degré d'association entre le document et le terme. Soit $D = \{d_1, d_2, \dots, d_N\}$ dénotant un ensemble de documents ou corpus, et $T = \{t_1, t_2, \dots, t_n\}$ le dictionnaire, ou l'ensemble des mots du corpus. T est obtenu en appliquant des opérations de traitement du langage naturel, comme l'atomisation (tokenization), l'élimination des mots vides de sens, et la troncature (stemming) (Baeza-Yates & Ribeiro-Neto, 1999). Chaque document d_j est représenté par un vecteur dans un espace vectoriel à n dimensions, tel que $d_j = \{w_{1j}, w_{2j}, \dots, w_{nj}\}$, où w_{kj} est le poids du terme t_k dans le document d_j .

La représentation de documents dans le modèle d'espace vectoriel fait apparaître deux difficultés: la pondération des termes et la mesure de similarité des vecteurs caractéristiques. Le schéma de pondération de terme le plus communément utilisé est la pondération TF-IDF (Term Frequency-Inverse Document Frequency) basée sur des observations empiriques sur le texte (Salton, 1989):

- les termes rares ne sont pas moins pertinents que les termes fréquents (IDF);
- des occurrences multiples d'un terme dans un document ne sont pas moins pertinentes que de simples occurrences (TF);
- des documents longs ne sont pas préférables à des documents courts (normalisation).

En d'autres mots, les termes qui apparaissent fréquemment dans un document, mais rarement dans le reste du corpus ont plus de chance de représenter le sujet du document. De plus, la normalisation des vecteurs résultats empêche les documents trop longs d'avoir plus de chance d'être retrouvés. Ces conjectures sont bien illustrées par la fonction TF-IDF:

ÉQUATION 1

$$\text{TFIDF}(t_k, d_j) = \text{TF}(t_k, d_j) \cdot \log \frac{N}{n_k}$$

où N dénote le nombre de documents dans le corpus, et n_k représente le nombre de documents de la collection dans lesquels le terme t_k apparaît au moins une fois, avec :

ÉQUATION 2

$$\text{TF}(t_k, d_j) = \frac{f_{k,j}}{\max_z f_{z,j}}$$

où le maximum est calculé sur la fréquence $f_{z,j}$ de tous les termes t_z qui apparaissent dans le document d_j . Pour que les poids soient dans l'intervalle $[0,1]$ et que les documents soient représentés par des vecteurs de même longueur, les poids obtenus par la fonction TFIDF() sont généralement normalisés par la normalisation cosinus (Équation 3).

ÉQUATION 3

$$w_{k,j} = \frac{\text{TFIDF}(t_k, d_j)}{\sqrt{\sum_{s=1}^{|T|} \text{TFIDF}(t_k, d_j)^2}}$$

Après la pondération des termes, il faut définir une mesure de similarité des vecteurs caractéristiques. Cette mesure de similarité est requise pour déterminer la proximité entre deux documents. Il existe de nombreuses mesures de similarité, mais la plus largement utilisée est la similarité cosinus:

ÉQUATION 4

$$\text{sim}(d_i, d_j) = \frac{\sum_k w_{ki} \cdot w_{kj}}{\sqrt{\sum_k w_{ki}^2} \cdot \sqrt{\sum_k w_{kj}^2}}$$

Dans les systèmes de recommandation basés sur le contenu s'appuyant sur le modèle d'espace vectoriel, les profils des utilisateurs et les items sont représentés comme des vecteurs de termes pondérés. La prédiction de l'intérêt d'un utilisateur sur un item donné peut être effectuée par calcul de similarité cosinus entre le vecteur de profil et le vecteur de l'item.

De nombreux systèmes de recommandation basés sur les mots-clefs ont été développés en très peu de temps dans de multiples domaines d'applications, comme les actualités, la musique, les films, etc. Chaque domaine présente différents problèmes qui requièrent différentes solutions.

Dans le domaine des systèmes de recommandation Web, des systèmes populaires sont Letizia, Personal WebWatcher, Syskill & Webert, ifWeb, Amalthea et WebMate. Letizia (Lieberman, 1995) est implémenté comme une extension de navigateur Web traquant le comportement de l'utilisateur et construit un modèle personnalisé constitué des mots-clefs liés aux intérêts de l'utilisateur. Il s'appuie sur un feedback implicite pour inférer des préférences de l'utilisateur. Par exemple, ajouter une page en favoris est interprété comme une preuve évidente de l'intérêt de l'utilisateur pour cette page. De manière similaire, Personal WebWatcher (Mladenic, 1999) apprend les intérêts des utilisateurs à partir des pages Web qu'ils visitent, et à partir des documents qui ont un lien hypermédia avec les pages visitées. Il traite les documents visités comme des exemples positifs d'intérêts utilisateurs, et les documents non visités comme des exemples négatifs. Amalthea (Moukas, 1997) utilise des agents spécifiques de filtrage pour assister les utilisateurs dans la recherche d'information. L'utilisateur peut définir les agents de filtrage en fournissant des pages (représentées par des vecteurs pondérés) étroitement liées à leurs intérêts. La même approche est adoptée par Syskill & Webert (Pazzani, Muramatsu, & Billsus, 1996) (Pazzani & Billsus, 1997), qui représente les documents en 128 mots parmi les plus informatifs identifiés. Dans ifWeb (Asnicar & Tasso, 1997), les profils sont représentés sous la forme de réseaux sémantiques pondérés. ifWeb prend en charge le feedback explicite et prend en compte les intérêts et les désintérêts explicites. Un autre aspect intéressant est qu'il incorpore un mécanisme temporel : il attribue un âge aux intérêts exprimés par l'utilisateur. WebMate (Chen & Sycara, 1998) utilise une approche différente pour la représentation des intérêts de l'utilisateur. WebMate assure le suivi des intérêts de l'utilisateur dans différents domaines en construisant le profil de l'utilisateur par apprentissage. Ce profil est constitué de vecteurs de mot-clef qui représentent des exemples positifs d'apprentissage. Un profil de n vecteurs de mot-clef peut représenter correctement jusqu'à n intérêts indépendants des utilisateurs.

Dans le domaine du filtrage des actualités, des systèmes de recommandation notables sont NewT, PSUN, INFOrmer, NewsDude, Daily Learner, et YourNews. NewT (NewsTailor) (Sheth & Maes, 1993) permet aux utilisateurs de fournir un feedback positif ou négatif sur des articles, des parties d'articles, des auteurs ou des sources. Plusieurs agents de filtrage sont formés pour différents types d'information, par exemple, un pour les actualités politiques, un pour le sport, etc. De la même façon, YourNews (Ahn, Brusilovsky, Grady, He, & Syn,

2007), un système plus récent pour l'accès personnalisé aux actualités, garde un profil d'intérêt séparé pour 8 sujets différents (National, Monde, Business, etc.). Le profil d'intérêt de l'utilisateur pour chaque sujet est représenté avec un vecteur de termes prototypes pondérés, extraits de l'historique des actualités vues par l'utilisateur. N articles des dernières actualités visionnées par l'utilisateur sont collectés, et les 100 termes les plus pondérés sont extraits pour générer les vecteurs prototypes finaux. Le système considère des profils à court terme, en ne considérant que les 20 dernières actualités, alors que des profils à long terme utilisent tout ce qui a été vu.

Apprendre des profils à long terme et des profils à court terme est typique des systèmes de filtrage d'actualités. NewsDude (Billsus & Pazzani, 1999) construit un modèle utilisateur à court terme basé sur TF-IDF, et un modèle à long terme basé sur un classificateur Bayésien naïf en s'appuyant sur un ensemble initial d'articles intéressants fournis par l'utilisateur. La source des actualités utilisée est Yahoo! News. De la même façon, Daily Learner (Billsus & Pazzani, 2000), un agent d'apprentissage pour l'accès aux informations en ligne, adopte une approche de construction par apprentissage de deux modèles utilisateurs séparés. Le premier, basé sur un algorithme de classification de texte par plus proche voisin, maintient les intérêts à court terme des utilisateurs, tandis que le deuxième, basé sur un classificateur Bayésien naïf, représente les intérêts à long terme des utilisateurs et s'appuie sur des données collectées sur une longue période de temps.

PSUN (Sorensen & McElligott, 1995) et INFOrmer (Sorensen, O'Riordan, & O'Riordan, 1997) sont deux systèmes utilisant une représentation plus complexe des articles ou profils. PSUN adopte une représentation alternative des articles. Les profils sont construits initialement par présentation au système de quelques articles que l'utilisateur trouve intéressants. Les mots récurrents de ces articles sont enregistrés au moyen de n-grammes (méthode classique issue du domaine du traitement du langage naturel) stockés dans un réseau de mots s'attirant ou se repoussant mutuellement, dont le degré d'attraction est déterminé par le nombre de cooccurrences. INFOrmer utilise un réseau sémantique pour représenter à la fois les profils utilisateurs et les articles. Une technique de propagation d'activation (Collins & Loftus, 1975) est utilisée pour comparer les articles et les profils. De plus, un mécanisme de feedback peut être utilisé pour adapter le comportement du système au changement d'intérêts des utilisateurs. Le modèle de propagation de l'activation consiste en une structure de réseau de données contenant des nœuds interconnectés par des liens, qui peuvent être libellés et/ou pondérés. Le traitement commence par libeller un ensemble de nœuds sources avec des poids d'activation et poursuit en propageant itérativement cette activation aux autres nœuds reliés aux nœuds sources, jusqu'à ce qu'une condition d'arrêt termine ce processus.

Une variété de systèmes basés sur le contenu existe dans d'autres domaines d'application. LIBRA (Mooney & Roy, 2000) implémente une méthode de catégorisation naïve Bayésienne pour la recommandation de livres exploitant la description des produits obtenue des pages Web d'Amazon.com. Citeseer (Bollacker & Giles, 1998) assiste l'utilisateur dans le processus de recherche de littérature scientifique, en utilisant des mots et analysant des citations communes dans les papiers. INTIMATE (Mak, Koprinska, & Poon, 2003) recommande des films en utilisant des techniques de catégorisation de texte sur les synopsis de film obtenus sur IMDb (The Internet Movie Database). Pour avoir des recommandations, l'utilisateur doit noter un certain nombre de films dans six catégories: horrible, mauvais, au-dessous de la moyenne, au-dessus de la moyenne, bon, excellent. De la même façon, Movies2GO (Mukherjee, Jonsdottir, Sen, & Sarathi) apprend les préférences des utilisateurs à partir des synopsis des films notés par l'utilisateur. Ce système intègre des schémas de votes (Straffin, 1980), conçus à l'origine pour permettre à des individus ayant des préférences conflictuelles d'arriver à un compromis acceptable. Ces schémas de votes sont adaptés ici pour gérer les préférences conflictuelles au sein d'un même profil utilisateur.

Dans le domaine de la musique, la technique commune utilisée pour fournir des recommandations est le filtrage collaboratif (par exemple, les systèmes Last.fm ou MyStrands). Le système Pandora (Pandora Internet

Radio) utilise des descriptions manuelles du contenu pour recommander des musiques. Le principal problème de ce type de système est son évolutivité, car le processus d'annotation des musiques est entièrement réalisé manuellement.

De l'analyse des principaux systèmes développés ces 15 dernières années, le plus important à retenir est que la représentation par mots-clefs à la fois pour les items et pour les profils peut donner des résultats précis. Toutefois, pour cela, il est nécessaire qu'un nombre suffisant de preuves d'intérêts des utilisateurs soit disponible. La plupart des systèmes basés sur le contenu sont conçus comme des classificateurs de textes construits à partir d'un ensemble de documents d'apprentissage qui sont soit des exemples positifs, soit des exemples négatifs des intérêts de l'utilisateur. Le problème avec cette approche est le « manque d'intelligence ». Lorsque des caractéristiques plus complexes sont nécessaires, les approches à base de mots-clefs montrent leurs limites. Si l'utilisateur, par exemple, aime « l'impressionnisme français », les approches à base de mots-clefs chercheront seulement des documents dans lesquels les mots « français » et « impressionnisme » apparaissent. Des documents concernant Claude Monet ou Renoir n'apparaîtront pas dans l'ensemble des recommandations, même s'ils sont susceptibles d'être pertinents pour l'utilisateur. Des stratégies de représentation plus avancées sont nécessaires pour que les systèmes de recommandation basés sur le contenu prennent en compte la sémantique associée aux mots.

2.1.3 RECOMMANDATIONS BASEES SUR LA SEMANTIQUE

Cette section présente un aperçu des principales stratégies adoptées pour introduire de la sémantique dans le processus de recommandation. La description de ces stratégies est abordée en tenant compte de plusieurs critères tels que :

- le type de source de connaissance impliquée (lexique, ontologie...);
- les techniques adoptées pour l'annotation ou la représentation d'items;
- le type de contenu inclus dans le profil utilisateur;
- la stratégie de correspondance entre items et profil.

Nous verrons dans la liste des systèmes étudiés que l'usage des ontologies pour décrire la sémantique des objets que l'on souhaite recommander devient petit à petit incontournable. Les systèmes de recommandation basés sur la sémantique évoluent au rythme des méthodes et outils proposés dans le domaine du Web sémantique.

Sitelf a été le premier système à adopter une représentation basée sur le sens des documents pour construire un modèle des intérêts de l'utilisateur (Magnini & Strapparava, 2001). Sitelf est un agent personnel pour un site Web de nouvelles multilingues. La source externe de connaissance impliquée dans le processus de représentation est MultiWordNet, une base de données lexicale multilingue. Chaque news est automatiquement associée à une liste d'ensembles de synonymes (dits synsets) de MultiWordNet en utilisant Word Domain Disambiguation (Magnini & Strapparava, 2000). Le profil utilisateur est un réseau sémantique où les nœuds représentent les synsets trouvés dans les documents lus par l'utilisateur. Durant la phase de correspondance, le système reçoit en entrée la représentation sous forme de synsets d'un document et le modèle utilisateur courant, et il produit en sortie une estimation de la pertinence du document en utilisant la technique Semantic Network Value Technique (Stefani & Strapparava, 1998).

ITR (ITerm Recommender) est un système capable de fournir des recommandations d'items dans plusieurs domaines (films, musique, livres), à condition que les descriptions d'articles soient disponibles sous forme de documents texte (Degemmis, Lops, & Semeraro, 2007) (Semeraro, Basile, de Gemmis, & Lops, 2009). Similairement à Sitelf, ITR intègre la connaissance linguistique au processus d'apprentissage des profils utilisateurs, mais Word Sense Disambiguation est adopté à la place de Word Domain Disambiguation pour obtenir une représentation basée sur le sens des documents. La connaissance linguistique vient exclusivement

de l'ontologie lexicale WordNet. Les items sont représentés selon des vecteurs basés sur des synsets, appelés bag-of-synsets (BOS), qui est une extension du classique bag-of-words (BOW) (Basile, Degemmis, Gentile, Lops, & Semeraro, 2007) (Semeraro, Degemmis, Lops, & Basile, 2007). Dans le modèle BOS, un vecteur de synsets, plutôt qu'un vecteur de mots, correspond à un document. Le profil utilisateur est construit comme un classificateur binaire naïf bayésien de texte, capable de catégoriser un item comme intéressant ou non. Il comprend les synsets qui s'avèrent être plus révélateurs des préférences de l'utilisateur, selon la valeur des probabilités conditionnelles estimées dans la phase d'apprentissage. La correspondance item-profil consiste à calculer la probabilité qu'un item appartienne à la classe « intéressant », par utilisation des probabilités des synsets dans le profil utilisateur.

SEWeP (Semantic Enhancement for Web Personalization) (Eirinaki, Vazirgiannis, & Varlamis, 2003) est un système de personnalisation Web qui utilise à la fois les logs d'utilisation et la sémantique du contenu du site Web dans le but de le personnaliser. Une taxonomie des catégories spécifiques au domaine a été utilisée pour annoter sémantiquement les pages Web, afin d'avoir un vocabulaire uniforme et consistant. Alors que la taxonomie est construite manuellement, le processus d'annotation est réalisé automatiquement. SEWeP, comme SiteIF et ITR, utilise la connaissance lexicale enregistrée dans WordNet pour interpréter le contenu d'un item et pour supporter le processus d'annotation/représentation. Les pages Web sont initialement représentées par des mots-clefs extraits de leur contenu, puis ces mots-clefs sont associés aux concepts de la taxonomie. Étant donné un mot-clef, une mesure de similarité de mot basée sur WordNet est appliquée pour trouver la catégorie la plus proche du mot-clef. SEWeP ne construit pas de profil de l'utilisateur, il découvre plutôt des patterns de navigation. Les catégories qui ont été sémantiquement associées à un pattern sont utilisées par le moteur de recommandation de SEWeP pour étendre l'ensemble des recommandations aux pages caractérisées par les catégories qui semblent intéresser l'utilisateur.

Quickstep (Middleton, Shadbolt, & De Roure, 2004) est un système de recommandation d'articles de recherche académique. Le système adopte une ontologie d'articles de recherche basée sur la classification scientifique du projet DMOZ open directory (DMOZ open directory project) (27 classes utilisées). L'annotation sémantique consiste à les associer avec le nom des classes de l'ontologie, par utilisation d'un classificateur par k plus proches voisins. Des profils d'intérêt sont calculés en corrélant les papiers précédemment parcourus avec leur classification. Les profils utilisateurs contiennent alors un ensemble de sujets et des valeurs d'intérêt dans ces sujets. La correspondance item-profil est réalisée en calculant la corrélation entre les trois premiers sujets intéressants du profil utilisateur et les papiers classifiés dans ces sujets. Foxtrot (Middleton, Shadbolt, & De Roure, 2004) étend le système Quickstep en implémentant une interface de recherche de papier, une interface de visualisation de profil et une notification par email, en plus de la page Web de recommandation. La visualisation du profil est possible, car il est représenté avec des termes ontologiques compréhensibles par l'utilisateur.

Informed Recommender (Aciar, Zhang, Simoff, & Debenham, 2007) utilise les avis des utilisateurs sur les produits pour faire des recommandations. Le système convertit les opinions des clients dans une forme structurée en utilisant une ontologie de traduction, qui est exploitée pour la représentation et le partage de connaissance. L'ontologie fournit un vocabulaire et des relations entre les mots pour décrire le niveau de compétence du client et son expérience sur les produits sur lesquels il donne un avis. Pour cela, l'ontologie contient deux principales parties: qualité de l'avis et qualité du produit, qui formalisent les deux aspects énoncés précédemment. Un processus de traitement de texte automatique associe les phrases dans les avis avec l'ontologie. Le système ne construit pas un profil de l'utilisateur, il calcule plutôt un ensemble de recommandations en se basant sur la requête d'un utilisateur. Prenons l'exemple d'un utilisateur qui demande la qualité d'un produit. Informed Recommender est capable de répondre à une requête et aussi de recommander le meilleur produit selon les caractéristiques de l'utilisateur concerné. Deux aspects sont à noter:

la connaissance ontologique peut modéliser différents points de vue selon lesquels les items peuvent être annotés; de plus les avis des utilisateurs sont du texte libre.

News@hand (Cantador, Bellogin, & Castells, 2008) est un système qui adopte une représentation basée sur une ontologie des caractéristiques des items et des préférences de l'utilisateur pour recommander des news. Le processus d'annotation associe les news avec des concepts appartenant à des ontologies de domaine. News@hand utilise 17 ontologies pour de multiples domaines tels que l'éducation, la culture, la politique, la religion, la science, le sport, etc. La manière dont le processus d'annotation est réalisé n'est pas clairement définie. Les descriptions d'item sont des vecteurs de scores TF-IDF sur l'espace des concepts définis dans les ontologies. Les profils utilisateurs sont représentés dans le même espace. La correspondance item-profil est réalisée par mesure de similarité cosinus.

Un système de recommandation pour de la télévision interactive digitale (Interactive Digital Television) est proposé dans (Blanco-Fernandez, J., G.S.A., Ramos-Cabrer, & Lopez-Nores, 2008). Les auteurs appliquent des techniques de raisonnement empruntées au Web sémantique pour comparer les préférences de l'utilisateur avec les items (programmes TV). Les programmes TV disponibles sont annotés par des métadonnées qui décrivent précisément leurs principaux attributs. Les connaissances sur le domaine de la télévision et les profils utilisateurs sont représentées en utilisant une ontologie OWL. Des profils ontologiques fournissent une représentation formelle des préférences des utilisateurs, permettant de raisonner sur eux et de découvrir de la connaissance supplémentaire sur leurs intérêts. La phase de recommandation exploite la connaissance stockée dans le profil utilisateur pour découvrir des associations sémantiques cachées entre les préférences des utilisateurs et les produits disponibles. Le système traite la connaissance inférée et une technique de propagation est adoptée pour suggérer des produits à l'utilisateur.

Le système JUMP (Basile, de Gemmis, Gentile, Iaquina, & Lops, 2007) (Basile, de Gemmis, Gentile, Iaquina, Lops, & Semeraro, 2007) est capable de délivrer intelligemment une information contextuelle et personnalisée à des travailleurs dans un environnement de travail non routinier. Les besoins des utilisateurs sont représentés sous la forme de requêtes complexes plutôt que sous la forme d'un profil utilisateur. Un exemple de requête complexe est « Je dois préparer un rapport technique pour le projet VIKEF ». L'analyse sémantique des documents et l'analyse des besoins de l'utilisateur sont basées sur une ontologie de domaine où chaque concept est manuellement annoté en utilisant des synsets de WordNet. La correspondance entre les documents et les concepts du domaine est réalisée automatiquement au moyen des procédures Word Sense Disambiguation et Named Entity Recognition, qui exploitent les annotations lexicales dans l'ontologie de domaine. Pour le traitement de la requête donnée en exemple, toutes les instances des concepts « rapport technique » et « projet », et les relations entre ces instances sont extraites de l'ontologie.

L'intérêt majeur pour les connaissances linguistiques est mis en valeur par la large utilisation de WordNet, qui est principalement utilisé pour l'interprétation du contenu en utilisant la désambiguïsation. D'un autre côté, les études décrites précédemment ont montré que le grand potentiel fourni par WordNet n'est pas suffisant pour la compréhension totale des intérêts des utilisateurs et pour leur contextualisation dans le domaine d'application. Une connaissance spécifique au domaine est aussi nécessaire. Les ontologies jouent un rôle fondamental de formalisation du domaine d'application, étant exploitées pour la description sémantique des items et pour la représentation des concepts (classes et instances) et relations (liens hiérarchiques et propriétés) identifiés dans le domaine. En conclusion, toutes les études qui incorporent soit de la connaissance linguistique, soit de la connaissance spécifique au domaine, ou les deux dans les méthodes de filtrage basées sur le contenu fournissent des résultats meilleurs et plus précis comparés aux méthodes traditionnelles basées sur le contenu.

2.2 SYSTEMES DE RECOMMANDATION BASES SUR UNE APPROCHE COLLABORATIVE

Les méthodes de filtrage collaboratif produisent des recommandations d'items à des utilisateurs donnés, en se basant sur des schémas de notations ou d'usage (par exemple, les achats) sans besoin d'information exogène sur les items et les utilisateurs. À la différence des approches basées sur le contenu, qui utilisent les items précédemment notés par un seul utilisateur u , les approches de filtrage collaboratif s'appuient sur les notes de tous les utilisateurs du système. L'idée clef est que la note de u pour un nouvel item i est susceptible d'être similaire à celle donnée par un autre utilisateur v , si u et v ont noté d'autres items d'une manière similaire. De même, u est susceptible de noter deux items i et j de la même façon, si d'autres utilisateurs ont donné des notes similaires à ces deux items. Les approches collaboratives dépassent certaines limitations des approches basées sur le contenu. Par exemple, des items dont le contenu n'est pas défini, ou difficilement définissable peuvent quand même être recommandés aux utilisateurs grâce aux feedbacks des autres utilisateurs. De plus, les recommandations collaboratives sont basées sur la qualité des items évaluée par les utilisateurs, au lieu de s'appuyer sur le contenu qui peut être un mauvais indicateur de qualité. Enfin, au contraire des systèmes basés sur le contenu, le filtrage collaboratif peut recommander des items avec des contenus différents, tant que les autres utilisateurs manifestent leurs intérêts pour ces différents items.

Selon (Bell & Koren, 2007) (Aciar, Zhang, Simoff, & Debenham, 2007) (Basile, de Gemmis, Gentile, Iaquinta, & Lops, 2007) (Cantador, Bellogin, & Castells, 2008), les méthodes collaboratives peuvent être groupées en deux classes générales: les méthodes de voisinages et les méthodes basées sur un modèle.

2.2.1 RECOMMANDATIONS BASEES SUR LE VOISINAGE

Les systèmes de recommandation basés sur le voisinage automatisent le principe de bouche-à-oreille, où l'on se fonde sur l'avis de personnes partageant les mêmes idées ou d'autres sources fiables pour évaluer la valeur d'un item (film, livre, article, album, etc.), selon nos propres préférences.

Ainsi, dans le filtrage basé sur le voisinage, les notes des utilisateurs stockées par le système sont directement utilisées pour prédire les notes pour de nouveaux items. Cela peut être fait de deux manières connues sous le terme de recommandations basées sur les utilisateurs ou recommandations basées sur les items.

2.2.1.1 RECOMMANDATIONS BASEES SUR LE VOISINAGE UTILISATEUR

Les systèmes basés sur le voisinage utilisateur, comme GroupLens (Konstan, Miller, Maltz, Herlocker, Gordon, & Riedl, 1997), Bellcore video (Hill, Stead, Rosenstein, & Furnas, 1995), et Ringo (Shardanand & Maes, 1995), évaluent l'intérêt d'un utilisateur u pour un item i en utilisant les notes de cet item. Ces notes sont données par d'autres utilisateurs, appelés voisins, qui ont des habitudes de notation similaires. Les voisins d'un utilisateur u sont typiquement les utilisateurs v dont les notes sur les items sont les plus proches de celles de u sur ces items. Les plus proches voisins sont les utilisateurs les plus similaires à u dans leur notation. Si on suppose que l'on a une mesure de similarité w_{uv} , pour tout utilisateur $v \neq u$, les k plus proches voisins de u , notés $N_k(u)$, sont les k utilisateurs v avec la plus grande mesure de similarité w_{uv} par rapport à u . Seuls les utilisateurs ayant noté l'item i peuvent être utilisés pour la prédiction. L'ensemble de ces voisins est noté $N_i(u)$. La note r_{ui} de l'utilisateur u sur l'item i peut être prédite par la moyenne des notes r_{vi} de ces voisins:

ÉQUATION 5

$$\hat{r}_{ui} = \frac{1}{|N_i(u)|} \sum_{v \in N_i(u)} r_{vi}$$

Un problème avec cette méthode par moyenne est qu'elle ne prend pas en compte le fait que les voisins peuvent avoir des niveaux différents de similarité. En effet, on peut prédire que la note de l'utilisateur u est

plus à même de se rapprocher de la note des voisins avec une plus grande similarité. Ainsi, une solution à ce problème est de pondérer la contribution de chaque voisin par leur similarité à u. Cependant, si la somme de ces poids ne fait pas 1, les notes prédites peuvent être en dehors des valeurs autorisées. De ce fait, il est courant de normaliser ces poids de telle sorte que la note prédite devienne:

ÉQUATION 6

$$\hat{r}_{ui} = \frac{\sum_{v \in N_i(u)} w_{uv} r_{vi}}{\sum_{v \in N_i(u)} |w_{uv}|}$$

Dans (Breese, Heckerman, & Kadie, 1998), w_{uv} est remplacé par w_{uv}^α , où $\alpha > 0$ est un coefficient amplificateur. Quand $\alpha > 1$, une plus grande importance est donnée aux voisins qui sont les plus proches de u.

Un important défaut de cette équation est qu'elle ne considère pas le fait que des utilisateurs peuvent utiliser des notes différentes pour quantifier le même niveau d'appréciation sur un item. Ce problème est habituellement résolu en convertissant les notes des utilisateurs r_{vi} en notes normalisées $h(r_{vi})$ (Breese, Heckerman, & Kadie, 1998) (Resnick, Iakovou, Sushak, Bergstrom, & Riedl, 1994):

ÉQUATION 7

$$\hat{r}_{ui} = h^{-1} \left(\frac{\sum_{v \in N_i(u)} w_{uv} h(r_{vi})}{\sum_{v \in N_i(u)} |w_{uv}|} \right)$$

Ces méthodes de prédiction basées sur une moyenne de notes du voisinage résolvent essentiellement un problème de régression.

Une autre orientation est la classification. Cela consiste à rechercher la note la plus probable que donnerait un utilisateur u à un item i, en prenant la valeur donnée le plus souvent par les plus proches voisins de u sur cet item et en considérant leur similarité avec u. Le vote v_{ir} donné par les k plus proches voisins de u ayant donné une note $r \in S$ peut être obtenu par la somme des valeurs de similarité des voisins qui ont donné cette note à i:

ÉQUATION 8

$$v_{ir} = \sum_{v \in N_i(u)} \delta(r_{vi} = r) w_{uv}$$

où $\delta(r_{vi} = r)$ est 1 si $r_{vi} = r$, et 0 sinon. Une fois que cela a été calculé pour chaque note possible, la note prédite est simplement la valeur r pour laquelle v_{ir} est le plus grand.

Une méthode de classification qui considère des notes normalisées peut aussi être définie. Soit, S' l'ensemble des valeurs normalisées possibles, la note prédite est :

ÉQUATION 9

$$\hat{r}_{ui} = h^{-1} \left(\arg \max_{r \in S'} \sum_{v \in N_i(u)} \delta(h(r_{vi}) = r) w_{uv} \right)$$

2.2.1.2 RECOMMANDATIONS BASEES SUR LE VOISINAGE DES ITEMS

Alors que les méthodes basées sur le voisinage utilisateur s'appuient sur l'avis d'utilisateurs partageant les mêmes idées pour prédire une note, les approches basées sur les items (Linden, Smith, & York, 2003) (Deshpande & Karypis, 2004) (Last.fm: Music recommendation service, 2009) prédisent la note d'un utilisateur

u pour un item i en se basant sur les notes de u pour des items similaires à i. Dans de telles approches, deux items sont similaires si plusieurs utilisateurs du système les ont notés d'une manière similaire.

Cette idée peut être formalisée comme suit. Soit $N_u(i)$ les items notés par un utilisateur u qui sont similaires à un item i. La note prédite de u pour i peut être obtenue par la moyenne pondérée des notes données par u aux items de $N_u(i)$:

ÉQUATION 10

$$\hat{r}_{ui} = \frac{\sum_{j \in N_u(i)} w_{ij} r_{uj}}{\sum_{j \in N_u(i)} |w_{ij}|}$$

Comme pour la recommandation basée sur le voisinage utilisateur, les différences d'échelle de notation des utilisateurs peuvent être considérées en normalisant les notes avec une fonction h:

ÉQUATION 11

$$\hat{r}_{ui} = h^{-1} \left(\frac{\sum_{j \in N_u(i)} w_{ij} h(r_{uj})}{\sum_{j \in N_u(i)} |w_{ij}|} \right)$$

De plus, il est aussi possible d'utiliser une approche par classification. Dans ce cas, les items j notés par l'utilisateur u votent pour la note à donner à un nouvel item i, et ces votes sont pondérés par la similarité entre j et i. La version normalisée de cette approche peut être exprimée comme suit:

ÉQUATION 12

$$\hat{r}_{ui} = h^{-1} \left(\arg \max_{r \in S'} \sum_{j \in N_u(i)} \delta(h(r_{uj}) = r) w_{ij} \right)$$

2.2.2 RECOMMANDATIONS BASEES SUR UN MODELE PREDICTIF

A la différence des systèmes basés sur le voisinage qui utilisent les notes stockées pour les prédictions, les approches basées sur un modèle utilisent ces notes pour construire un modèle prédictif par apprentissage. L'idée générale est de modéliser les interactions utilisateur-item avec des facteurs représentant des caractéristiques latentes des utilisateurs et items dans le système, comme des classes d'utilisateurs et d'items. Ce modèle est ensuite qualifié à partir des données disponibles, et utilisé plus tard pour prédire les notes des utilisateurs pour de nouveaux items. Les approches basées sur un modèle sont nombreuses, elles incluent le clustering bayésien (Breese, Heckerman, & Kadie, 1998), l'analyse de la sémantique latente (Hofmann, 2003), l'allocation de Dirichlet latente (Blei, Ng, & Jordan, 2003), l'entropie maximum (Zitnick & Kanade, 2004), les machines de Boltzmann (Salakhutdinov, Mnih, & Hinton, 2007), les machines à support vectoriel (Grcar, Fortuna, Mladenic, & Grobelnik, 2006), et la décomposition en Valeur singulière (Bell, Koren, & Volinsky, 2007) (Koren, 2008) (Paterek, 2007).

Ces systèmes transforment à la fois les items et les utilisateurs dans un espace de facteurs latent, permettant de les comparer directement. L'espace latent cherche à expliquer les notes en caractérisant les produits et les utilisateurs sur des facteurs automatiquement déduits des feedbacks utilisateurs. Par exemple, pour des films, les facteurs peuvent mesurer des dimensions évidentes comme la quantité d'action, ou l'orientation pour un certain public; des dimensions moins bien définies telles que la profondeur du développement des personnages; ou des dimensions complètement ininterprétables. Pour les utilisateurs, chaque valeur de facteur mesure à quel point l'utilisateur aime les films sur ce facteur donné.

Les systèmes de recommandation basés sur des modèles de facteurs latents offrent des capacités expressives hautes pour décrire les divers aspects des données. Ainsi, ils tendent à fournir des résultats plus précis que les

systèmes basés sur le voisinage. Cependant, la grande partie de la littérature et des systèmes commerciaux (par exemple, Amazon (Linden, Smith, & York, 2003), TiVo (Ali & van Stam, 2004) et Netflix (Netflix)) sont basés sur le voisinage. Actuellement, il existe beaucoup plus de systèmes de recommandation basés sur le voisinage, car ils sont considérés comme plus faciles et intuitifs à manipuler. Tout d'abord, ils fournissent naturellement des explications plus intuitives du raisonnement derrière les recommandations, ce qui améliore l'expérience utilisateur. Enfin, ils peuvent immédiatement délivrer des recommandations à l'utilisateur en se basant sur les feedbacks qu'il vient juste de fournir.

2.3 CONCLUSION

Les systèmes de recommandation sont considérés comme un sous-ensemble des systèmes hypermédia adaptatifs proposant une solution au problème de surcharge d'information par proposition de recommandations d'items. Le problème de recommandation d'items à un utilisateur a été formalisé et les différents types de systèmes de recommandation et leurs méthodes ont été exposés. Pour synthétiser cette étude, nous avons identifié deux grandes catégories de systèmes de recommandation : les systèmes de recommandation basés sur le contenu et les systèmes de recommandation basés sur le filtrage collaboratif.

L'adoption du paradigme de recommandation basé sur le contenu présente de nombreux avantages par rapport à celui basé sur le collaboratif :

- **Indépendance utilisateur :** Les systèmes de recommandation basés sur le contenu n'utilisent que le comportement de l'utilisateur actif pour construire son profil. Au contraire, l'approche collaborative a besoin de notes d'autres utilisateurs pour trouver les plus proches voisins de l'utilisateur actif et fournir les items aimés par ces voisins.
- **Transparence :** Des explications sur la manière dont le système de recommandation fonctionne peuvent être fournies en listant explicitement les caractéristiques ou les descriptions qui ont engendré l'apparition d'un item dans une liste de recommandations. Au contraire, les systèmes collaboratifs sont des boîtes noires, puisque la seule explication d'une recommandation d'item est que des utilisateurs inconnus avec des goûts similaires ont aimé cet item.
- **Démarrage à froid :** Les systèmes de recommandation basés sur le contenu sont capables de recommander des items pas encore notés par des utilisateurs. Ainsi, ils ne souffrent pas du problème de démarrage à froid, qui affecte les systèmes collaboratifs qui se basent uniquement sur les notes des utilisateurs pour faire des recommandations. Dans ces systèmes, un item qui n'a pas reçu assez de notes des utilisateurs n'est pas recommandé.

Néanmoins, les systèmes basés sur le contenu ont plusieurs inconvénients :

- **Analyse limitée du contenu :** Les techniques basées sur le contenu ont une limite naturelle sur le nombre et le type des caractéristiques qui sont associées, soit automatiquement ou manuellement, aux objets à recommander. La connaissance du domaine est souvent nécessaire. Par exemple, pour de la recommandation de films, le système a besoin de connaître les acteurs et directeurs. Aucune recommandation basée sur le contenu ne peut fournir des suggestions convenables si l'analyse du contenu ne contient pas assez d'information pour discriminer les items que l'utilisateur aime des items qu'il n'aime pas.
- **Sur-spécialisation :** Les systèmes de recommandation basés sur le contenu n'ont pas de méthode inhérente pour trouver quelque chose d'inattendu. Le système suggère des items dont les scores sont hauts lorsqu'ils sont mis en correspondance avec le profil. De ce fait, l'utilisateur se voit recommander des items similaires à ceux recommandés précédemment. Cet inconvénient est aussi appelé le problème du « heureux hasard » pour mettre en avant la tendance des systèmes de recommandation à produire des recommandations avec un degré limité de nouveauté. Pour donner un exemple, quand

un utilisateur a seulement noté un film dirigé par Stanley Kubrick, il lui sera seulement recommandé un seul type de film. Le système de recommandation lui proposera rarement, voire pas du tout, de nouveaux objets, ce qui limite la gamme d'applications pour lesquelles il serait utile.

Par rapport aux méthodes basées sur le contenu, le filtrage collaboratif présente quelques avantages:

- **Indépendance vis-à-vis du contenu:** Contrairement aux systèmes basés sur le contenu, les systèmes collaboratifs n'ont pas besoin de décrire très précisément les items pour de bonnes recommandations. Ces dernières sont faites indépendamment de la représentation des items. Ainsi, elles peuvent s'appliquer à tout type d'items dont le contenu est soit indisponible, soit difficile à analyser.
- **Pas de sur-spécialisation:** Les recommandations des systèmes collaboratifs ne se fondent pas sur la dimension thématique des profils. Ainsi, ils ne sont pas soumis au problème de sur-spécialisation (aussi appelé effet « entonnoir ») tel qu'il existe pour les systèmes de recommandation basés sur le contenu.

Néanmoins, ces systèmes présentent aussi quelques inconvénients:

- **Recommandations limitées:** Lorsque des items n'ont pas encore été notés ou libellés par des utilisateurs, les systèmes collaboratifs ne peuvent pas recommander ces items, car ils se basent sur ces informations pour les recommandations. Cela engendre des problèmes de démarrage à froid.
- **Cas extrêmes:** Le filtrage collaboratif ne peut pas couvrir les cas extrêmes. Si l'échelle des profils possibles est trop petite ou si les utilisateurs ont des goûts qui leur sont uniques, il n'est pas possible d'émettre des correspondances fiables entre profils.

Chaque type de système possède des avantages et inconvénients. Les méthodes collaboratives donnent de meilleurs résultats, qualitativement, que les méthodes basées sur le contenu. Néanmoins, leur principal problème réside dans le fait qu'il faut une base d'utilisateurs ayant déjà fait des choix pour faire des recommandations. Néanmoins, le contenu à proposer n'a pas besoin d'être défini de manière complexe. Les méthodes basées sur le contenu permettent de faire des propositions dès l'initialisation du système. En contrepartie, il faut bien définir le contenu à proposer pour avoir de bonnes propositions.

Il existe de nombreuses propositions combinant les avantages des systèmes de recommandation par le contenu et des systèmes de recommandation par filtrage collaboratif : Fab (Balabanovic, 1997), WebWatcher (Joachims, Freitag, & Mitchell, 1997), P-Tango (Claypool, Gokhale, Miranda, Murnikov, Netes, & Sartin, 1999), ProfBuilder (Wasfi, 1999), PTV (Smith & Cotter, 2000), Content-boosted Collaborative Filtering (Melville, Mooney, & Nagarajan, 2002), Cinema Screen (Salter & Antonopoulos, 2006) et celui proposé dans (Symeonidis, 2008).

Au cours du temps, ces propositions (hybride ou non) intègrent majoritairement les nouvelles techniques de modélisation définies dans le domaine du Web Sémantique et en particulier les ontologies.

3 CONCLUSION SUR LE CHAPITRE

Ce chapitre est un état de l'art de différentes méthodes et techniques qui seront utilisées pour la résolution de notre projet de recherche. La première partie de ce chapitre présente le domaine des systèmes hypermédia adaptatifs. Cette partie met en évidence une architecture nécessaire et suffisante pour la réalisation de systèmes adaptatifs avec une description des couches principales. Trois couches se distinguent, elles représentent l'utilisateur, le domaine et l'adaptation. Le but des systèmes hypermédia adaptatifs est de résoudre le problème de surcharge cognitive (ou surcharge d'informations) et le problème de perte dans l'hyperespace. Un sous-domaine, chargé de résoudre uniquement le premier problème, est appelé « système

de recommandation ». Dans ces systèmes, présentés dans la seconde partie de ce chapitre, il existe deux principales méthodes de recommandation. La méthode basée sur le contenu recherche des items en se basant sur leurs caractéristiques et selon une requête ou le profil d'un utilisateur. La méthode par filtrage collaboratif recherche des documents en se basant sur les choix d'autres utilisateurs dans le système. Ces deux grandes classes de méthodes ne sont pas disjointes, elles sont très souvent combinées pour former des systèmes hybrides (Figure 7).

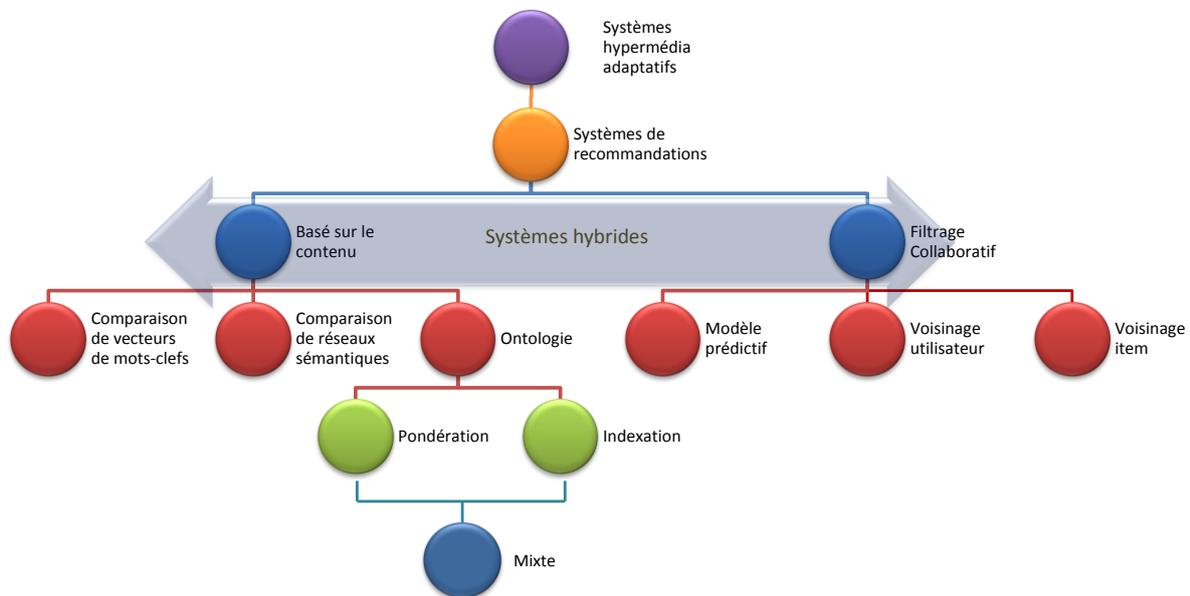


FIGURE 7. CLASSIFICATION DES DIFFERENTS TYPES DE SYSTEMES DE RECOMMANDATION

Au sein des méthodes collaboratives, il existe 3 principales approches. La première utilise des modèles prédictifs basés sur un apprentissage des notes fournies par les utilisateurs sur les items. La deuxième approche se base sur un voisinage utilisateur. Elle cherche à prédire la note sur un item donné en s'appuyant sur les notes émises sur cet item par des utilisateurs partageant des idées proches. Enfin, la troisième approche utilise un voisinage d'items. Elle cherche à prédire la note d'un item donné en fonction des items qui lui ressemblent et qui ont été précédemment notés par l'utilisateur.

Dans les méthodes basées sur le contenu, les 3 principales approches sont une approche par comparaison de vecteurs de mots-clés, une approche par comparaison des réseaux sémantiques et une approche par pondération des classes d'ontologie. Les deux premières approches sont assez proches. Pour ces deux premières méthodes, il est nécessaire que des mots-clés ou textes décrivant les items soient disponibles. La deuxième approche se base sur une modélisation des items et des profils à l'aide de réseaux sémantiques pondérés. Ce sont des réseaux de mots reliés les uns aux autres selon leurs cooccurrences dans des textes. Des comparaisons sont faites entre les réseaux sémantiques des items et celui de l'utilisateur pour fournir des recommandations. La troisième approche consiste à utiliser une ou plusieurs ontologies décrivant un ou plusieurs domaines. Il existe plusieurs manières d'utiliser ces ontologies. Soit, en pondérant les classes de l'ontologie pour représenter les items et/ou les utilisateurs. Soit, en indexant au sein de l'ontologie les items et/ ou les utilisateurs comme des instances. Enfin, il existe de nombreuses méthodes qui proposent une approche dite mixte, qui combine pondération et indexation (Basile, de Gemmis, Gentile, Iaquina, & Lops, 2007) (Blanco-Fernandez, J., G.S.A., Ramos-Cabrer, & Lopez-Nores, 2008) (Middleton, Shadbolt, & De Roure, 2004). Les méthodes de pondération des classes pour représenter les items et les utilisateurs se rapprochent des réseaux sémantiques où la méthode de recommandation consiste à comparer les différentes pondérations (par des distances vectorielles par exemple).

Comparé aux deux autres méthodes, l'ontologie présente l'avantage de désambigüiser la terminologie, elle fixe un certain nombre de règles sémantiques facilitant la formalisation d'un domaine. De plus, un de ses intérêts est qu'elle peut être le support d'inférences logiques afin de déduire de nouvelles connaissances. En outre, un avantage de l'ontologie est sa réutilisabilité. Un système de recommandation dans le même domaine d'application mais avec des processus de recommandations différents peut aussi être basé sur cette ontologie.

Chaque type de système possède des avantages et des inconvénients. Les méthodes collaboratives donnent de meilleurs résultats, qualitativement, que les méthodes basées sur le contenu. Leur principal problème réside dans le fait qu'il faut une base d'utilisateurs ayant déjà fait des choix pour faire des recommandations. Néanmoins, le contenu à proposer n'a pas besoin d'être défini de manière complexe. Les méthodes basées sur le contenu permettent de faire des propositions dès l'initialisation du système. En contrepartie, il faut bien définir le contenu à proposer pour avoir de bonnes propositions.

Les objectifs de notre projet sont de réaliser un système capable de proposer des combinaisons d'items à un utilisateur suivant son profil et se basant sur le savoir-faire du fournisseur. Néanmoins, parmi les différents systèmes de recommandations proposés, aucun ne permet au fournisseur de traduire son expertise pour la recommandation.

Ce type de système se situe clairement dans les systèmes de recommandation basés sur le contenu car, pour pouvoir exprimer pleinement le savoir-faire du fournisseur, il est nécessaire d'avoir une base de connaissance bien formalisée et complète (contenant les concepts considérés par le savoir-faire). L'utilisation d'une ontologie est ainsi de premier intérêt pour modéliser la connaissance du domaine de manière formelle. Le savoir-faire du fournisseur peut alors se baser sur cette formalisation pour sa modélisation.

En outre, le besoin de combiner des items pour la recommandation n'a pas été abordé par les systèmes de recommandation actuels.

Ainsi, en vue de répondre aux objectifs de notre projet, nous proposons dans le chapitre suivant, une architecture de systèmes de recommandation étendant les systèmes de recommandation basés sur une ontologie permettant de fournir une combinaison d'items selon des contraintes spatio-temporelles et tenant compte du savoir-faire du fournisseur.

Chapitre 3

Approche Sémantique et Combinatoire pour un système de Recommandation Générique

Résumé

Le chapitre 3 définit formellement un nouveau type de système de recommandation proposant des combinaisons d'individus d'une ontologie de domaine et permettant la prise en compte, sous la forme de règles métiers et de modèle de but, du savoir-faire du fournisseur de contenus. Cette proposition est une extension des systèmes de recommandation basés sur le contenu utilisant une ontologie associée à des mécanismes de pondération et d'indexation. L'architecture utilisée est issue des systèmes hypermédia adaptatifs, elle est constituée de 3 couches : une couche sémantique, une couche intelligence et une couche utilisateur. La constitution des différentes couches est décrite et les différents processus de la couche intelligence permettant la génération des propositions sont expliqués. Pour proposer des recommandations sous forme de combinaisons, le système est constitué de manière à pouvoir utiliser des algorithmes d'optimisation combinatoire adaptés aux domaines d'applications. En effet, nous verrons dans la suite que les problèmes d'optimisation sont propres au domaine d'application.

Plan

1	Discussion	57
2	Couche sémantique.....	60
2.1	Le Web sémantique	60
2.2	Ontologie	61
2.3	Modèle de domaine	62
2.4	Modèle de buts.....	62
3	Couche utilisateur	64
4	Couche intelligence	65
4.1	Phase de projection	65
4.1.1	Formalisation.....	65

4.1.2	Exemple.....	66
4.2	Phase de recherche combinatoire.....	66
4.2.1	Forme de la combinaison.....	67
4.2.2	Pertinence d'une combinaison.....	67
4.2.3	Problème de recherche de la meilleure combinaison.....	68
4.3	Récapitulatif/Aperçu de la couche intelligence.....	68
5	Conclusion.....	69

Le chapitre 2 nous a permis d'identifier trois éléments essentiels pour notre projet. Au niveau des systèmes hypermédia, une architecture formée de 3 couches minimum est nécessaire et suffisante pour la création d'un système hypermédia adaptatif. Des couches supplémentaires peuvent y être ajoutées selon les besoins d'adaptation et d'évolutivité désirés. Au niveau de la modélisation de l'information, l'utilisation d'ontologies et des technologies relatives issues du Web sémantique sont de premier choix pour formaliser de manière complète et complexe un domaine d'application.

Néanmoins, il est possible d'identifier certains verrous que nous souhaitons lever dans le cadre de la conception d'un système de recommandation. Outre le fait qu'il n'existe pas d'architecture générique sur laquelle s'appuyer pour la création d'un système de recommandation, on constate que lors de la modélisation de la connaissance, aucune distinction explicite n'est faite entre la modélisation de la connaissance générale du domaine et la modélisation de la connaissance du domaine spécifique à l'utilisateur dans une application donnée. Tout est regroupé sous une même notion de modèle de domaine. La séparation de ces deux concepts pourrait s'avérer utile pour la portabilité d'un système de recommandation et pour faciliter la création de ce dernier.

Le modèle LAOS, expliqué dans l'état de l'art, propose un découpage qui peut s'y apparenter, même s'il ne définit pas cette séparation de la même façon. Il considère d'un côté un modèle de domaine représentant des livres du domaine de e-learning et d'un autre côté un modèle de buts et des contraintes. Ce dernier filtre, regroupe, restructure le modèle de domaine en fonction des objectifs d'instruction. Il permet notamment réduire l'espace des propositions possibles et de les formuler différemment suivant un objectif donné.

De plus, dans les systèmes de recommandation, les différents travaux se portent sur des systèmes proposant des items uniques sous forme de liste triée selon la pertinence des items pris individuellement. Ils ne considèrent pas encore la proposition d'items interdépendants ou, en d'autres mots, la proposition de combinaisons d'items.

Nous souhaitons définir un nouveau type de système de recommandation apparenté à la famille des approches basées sur le contenu, qui utilise une ontologie, mais aussi une modélisation du savoir-faire métier sous la forme de règles. Ce nouveau type de systèmes de recommandation se différencie de ceux présentés dans l'état de l'art, car il ne se limite pas à l'extraction d'un item pertinent, mais il a pour objectif la construction de combinaisons pertinentes d'items en fonction de profils définis par les utilisateurs. La combinaison d'items nécessite l'intégration d'algorithmes d'optimisation combinatoire. Néanmoins, nous verrons que chaque application possède son propre problème d'optimisation. Notre architecture devra donc être conçue de manière à accueillir des algorithmes adaptés aux problèmes d'optimisation du domaine étudié.

1 DISCUSSION

Pour la création d'une nouvelle architecture de systèmes recommandations, il faut en premier lieu choisir quel type de système de recommandation il est préférable de réaliser: système basé sur le contenu, collaboratif ou hybride? Ce choix dépend très fortement du niveau de qualité de l'information contenue dans le système d'information du fournisseur et des critères de recommandations des items entre les utilisateurs.

Nous partons du postulat que notre information est modélisée au sein d'une ontologie et par conséquent, la qualité du contenu est avérée. Un système de recommandation basé sur le contenu pourrait être construit à partir de ce postulat. Néanmoins, il ne permettrait pas de représenter le savoir-faire du fournisseur en matière de qualification d'un item unique et en matière de combinaison pertinente des items. Il est donc nécessaire de trouver un moyen qui représente ce savoir-faire sous la forme de règles paramétrables. La qualification des offres dans les systèmes de recommandation par filtrage collaboratif est réalisée par rapport à des notes qui

sont données par les utilisateurs aux items. Nous avons choisi de nous inspirer de cette idée pour permettre de donner à notre architecture des fonctions de pondération des items par rapport aux règles modélisées pour représenter le savoir-faire du fournisseur. Cette solution répond à notre besoin immédiat et de plus, elle possède un caractère évolutif. Cette pondération pourrait être ensuite directement déduite des avis des utilisateurs lors de la sélection des items. Cette transmission de la capacité de pondérer répond à l'une des principales critiques des systèmes de recommandation basés sur du filtrage collaboratif qui nécessitent une phase d'initialisation importante pour collecter les avis d'utilisateurs sur les items.

Pour développer ce système de recommandation, il est nécessaire de construire un système de description de contenu qui assure la pertinence des critères de description des offres. Ce système évolutif de règles et de pondération pourra traduire le savoir-faire du fournisseur, et permettra une qualification constante du contenu tout au long de son cycle de vie, tout en fournissant une combinaison pertinente d'items.

Dans les systèmes adaptatifs, le contenu et les informations relatives sont représentés dans la couche couramment appelée « modèle de domaine » à l'aide d'un ensemble de connaissances relatives au domaine d'application. Les travaux sur les systèmes adaptatifs utilisent de plus en plus les technologies du Web sémantique pour représenter ce modèle, notamment les outils de représentation ontologique.

Dans notre démarche, nous faisons une distinction entre une modélisation générale de la connaissance et une modélisation spécifique de la connaissance suivant les contraintes des utilisateurs dans une application donnée. La modélisation spécifique de la connaissance est spécifique à une application donnée alors que la modélisation générale de la connaissance peut être réutilisée pour une autre application dans le même domaine d'application. Nous garderons le terme « modèle de domaine » pour la modélisation générale de la connaissance, et le terme « modèle de buts », en accord avec la terminologie proposée par LAOS, pour la modélisation spécifique de la connaissance.

Les systèmes de recommandation actuels, basés sur le contenu, proposent des items qui se rapprochent au mieux du profil de l'utilisateur. Cependant, il est assez rare qu'un item suggéré soit en totale adéquation avec ce profil, satisfaisant les différents objectifs (intérêts, connaissances, etc.). L'item est en général « très bon » sur quelques critères du profil et « moyen » sur d'autres. L'idée avancée dans cette thèse est que, dans certains types d'applications, il peut être préférable, au lieu de proposer un seul item ou une liste d'items satisfaisant quelques critères du profil, de proposer une combinaison d'items, qui, lorsqu'ils sont associés les uns aux autres, correspondent complètement au profil de l'utilisateur. L'idée est que même si chaque item pris séparément n'est pas « très bon » dans les différents objectifs des utilisateurs, l'ensemble peut les intéresser, car ils comblent bien tous les objectifs. Ainsi, au lieu de soumettre des listes d'items de même type aux utilisateurs, notre approche consiste à rechercher des combinaisons d'items de types différents à leur proposer.

Pour illustrer l'intérêt de cette approche, imaginons un utilisateur défini par son goût pour la science-fiction, la mer, la gastronomie, les vampires. Un système de recommandation souhaite lui proposer des livres qui lui correspondent. Ce système va afficher une liste de livres qui correspondent aux mieux à tous les critères, cette liste étant triée par pertinence (souvent modélisé par un poids). Dans notre exemple, ce système affiche une liste de trois livres: un livre de science-fiction se déroulant en milieu aquatique et deux livres de cuisine de poissons. Ces livres remplissent au plus deux critères du profil de l'utilisateur, ce qui est satisfaisant. Néanmoins, rien n'est proposé pour le critère « vampires ». Maintenant, si nous proposons une combinaison de trois livres, pour optimiser la pertinence de l'ensemble sur tous les critères, alors il y aurait une grande probabilité qu'il y ait un livre de vampires, un livre de cuisine des poissons et un livre de science-fiction se déroulant en milieu aquatique.

Dans cet exemple, on imagine très bien que la deuxième solution conviendrait mieux à l'utilisateur, car la totalité de ses critères serait prise en compte dans la recommandation. Ainsi, suivant ce raisonnement, nous pensons qu'il peut être pertinent de rechercher des combinaisons d'items plutôt qu'une liste d'items à proposer à l'utilisateur. Ce postulat est fortement lié au domaine du tourisme, où l'organisation d'un voyage nécessite souvent la satisfaction des désirs ou des contraintes des voyageurs sur un ensemble d'offres touristiques.

Un système de recommandation proposant des combinaisons d'items suivant un pattern apparaît comme une généralisation des systèmes de recommandation proposant des listes d'items. L'utilisation de patterns peut amener à un nouveau niveau d'adaptation dans les systèmes de recommandation : adaptation du pattern de combinaison.

Pour répondre aux critères que nous avons définis pour construire ce système de recommandation, nous avons défini une architecture composée de plusieurs couches, en accord avec la structure des systèmes hypermédia adaptatifs. La Figure 8 présente les trois couches principales du système de recommandation générique proposé. La couche sémantique contient la représentation du contenu selon deux niveaux de modélisations. Une modélisation suivant la connaissance générale du domaine (médecine, tourisme, vignoble...) que nous appelons modèle de domaine. Une modélisation suivant l'application qui est une réorganisation de la connaissance suivant les contraintes possibles des utilisateurs dans l'application. Cette dernière couche est appelée modèle de buts. Pour construire ces deux modèles, nous utiliserons des ontologies et des règles logiques. Le modèle de domaine et le modèle de buts étant intimement liés, nous les regroupons dans une même couche dénommée couche sémantique. La couche intelligence est équivalente à la couche modèle d'adaptation des systèmes hypermédia adaptatifs. Elle contient deux principaux types de mécanismes. Tout d'abord des mécanismes de pondération des items selon le profil utilisateur et ensuite des mécanismes de combinatoire pour associer les items pour construire une combinaison pertinente d'items. La couche utilisateur, qui correspond au modèle utilisateur des systèmes hypermédia adaptatifs, contient la définition des profils des utilisateurs.

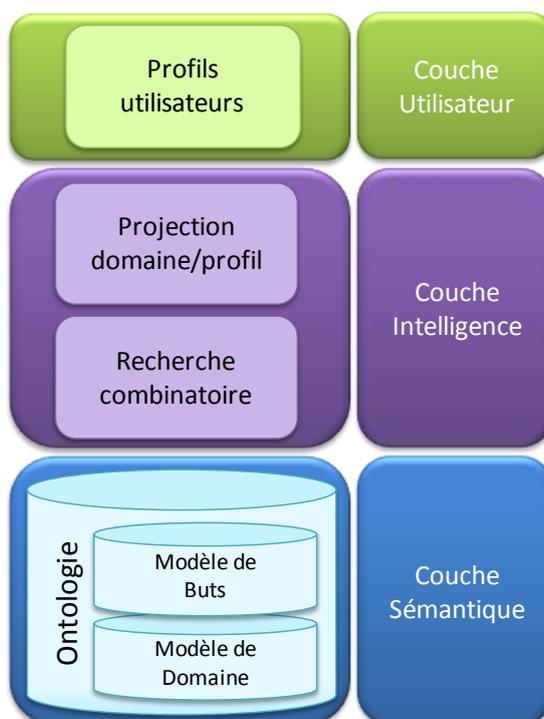


FIGURE 8. ARCHITECTURE DU SR

2 COUCHE SEMANTIQUE

La couche sémantique est la partie qui permet de modéliser le contenu, la connaissance selon le domaine et selon les besoins utilisateurs dans l'application. Dans cette modélisation, l'apport des technologies du Web sémantique est significatif. En effet, l'utilisation d'une ontologie va permettre la modélisation de réseaux complexes de connaissances.

Cette partie présente dans un premier temps le Web sémantique suivi d'une introduction aux ontologies. Ensuite, deux sous-couches de cette couche sémantique sont définies. La première, le modèle de domaine, permet de représenter la connaissance générale du domaine. La deuxième, le modèle de buts, représente la connaissance spécifique à l'application suivant les possibles contraintes des utilisateurs.

2.1 LE WEB SEMANTIQUE

Le Web sémantique a pour objectif d'organiser, de structurer l'immense quantité d'information présente pour que ce volume de données puisse être partageable et traitable par différents processus automatiquement. Le Web sémantique trouve ses origines dans le postulat que le Web est incomplet. En effet, avec l'expansion désordonnée de l'information sur le Web, est apparu le besoin d'organiser et de maintenir les informations afin d'offrir des réponses structurées et plus précises aux demandes des utilisateurs. De plus, le Web a besoin d'être plus compréhensible, que ce soit dans le contexte des utilisateurs ou dans celui des machines, afin d'aider les utilisateurs à trouver la bonne l'information. C'est dans ce contexte qu'est apparue la proposition du Web sémantique, il est attribué à Tim Berners-Lee (Berners-Lee, Hendler, & Lassila, 2001). Le Web sémantique ne constitue pas un Web à part, c'est une extension du Web actuel. Il fait référence au Web de demain comme un large espace d'échange de ressources entre machines et êtres humains permettant une meilleure exploitation qualitative de grands volumes d'informations et de services. L'idée principale du Web sémantique est de structurer syntaxiquement et sémantiquement les données présentes sur le Web pour rendre les informations interopérables et interrogeables par des machines avec un minimum d'intervention de l'utilisateur. Contrairement au Web que nous connaissons actuellement, les tâches de recherche, de construction et de combinaison des résultats n'incomberaient plus complètement aux utilisateurs, mais aux ordinateurs. Ces derniers seraient ainsi dotés de capacités accrues pour accéder aux contenus des ressources et pour effectuer des raisonnements sur ces ressources. De plus, l'idée du Web est que les usagers en soient les contributeurs, c'est en ce sens que les langages développés pour le Web sémantique auront un impact sur ceux qui les utiliseront pour décrire leurs ressources voire leurs services.

La première direction proposée pour développer le Web sémantique était d'encourager les créateurs de pages Web à fournir certaines formes de métadonnées (des données sur les données) à leurs pages Web, comme l'ajout de l'identité de l'auteur d'une page web par exemple. La structure formelle pour ces métadonnées, appelée Resource Description Framework (RDF), a été élaborée par Hayes, un des pionniers de l'intelligence artificielle. RDF est un langage simple pour la création d'assertions sur des propositions (Hayes, 2004). Le concept de base de RDF est celui du « triplet » : toute déclaration peut être composée en un sujet, un prédicat, et un objet. « Le créateur de la page Web est Jack Smith » peut être formulée en `www.u-bourgogne.com/webpage dc:creator "Jack Smith"`. Cette structure a été étendue à celui d'un langage d'ontologie complet décrit par la logique de description. Ce langage ontologique du Web (Web Ontology Language) OWL est ainsi plus expressif que RDF (Welty, Smith, & McGuinness, 2004).

Le paradigme du Web sémantique génère un petit changement, mais un changement fondamental dans l'architecture du Web : une ressource (qui est tout ce qui peut être identifié par une URI) peut se porter sur n'importe quoi. Cela signifie que les URIs, qui étaient anciennement utilisées pour dénoter principalement des pages Web, peuvent dorénavant porter sur toutes choses dont l'existence physique est en dehors du Web (Jacobs & Walsh, 2004). Par exemple, une URI peut, au-delà du fait d'indiquer une page Web sur la tour Eiffel,

indiquer la Tour Eiffel elle-même (même s'il n'y a pas de page Web sur ce monument). Ce qui était, à l'origine, des annotations manuelles de pages Web avec des métadonnées, est devenu le problème de représentation des connaissances à grande échelle et de développement des ontologies.

2.2 ONTOLOGIE

Pour définir une ontologie, il est nécessaire de partir du concept de réseau sémantique. L'idée de réseau sémantique est attribuée à Quillian (Quillian, 1968) pour l'avoir utilisé pour modéliser le fonctionnement de la mémoire. L'idée centrale des réseaux sémantiques est de décrire la réalité sous forme de graphes (réseaux) composés de nœuds qui représentent les concepts, reliés par des arcs, exprimant la relation entre les concepts. Les nœuds et les arcs sont en général étiquetés. Aux premiers sont associés les objets (concept, événement, situation, etc.), aux seconds les relations entre les objets, d'où la désignation de « structure objet-relation ». Les liens les plus spécifiques de ce type de réseau sont les liens « sorte-de ». D'autres liens sont utilisés comme « à pour partie », « instrument pour », etc. Le lien « sorte-de » permet de définir l'une des notions les plus importantes des réseaux sémantiques et de tout autre formalisme intégrant un minimum de logique, qui est la déduction par héritage des propriétés. L'héritage est essentiellement basé sur la transitivité de la relation « sorte-de ». De cette notion d'héritage se dégage le fait qu'il est plus économique d'attacher une caractéristique commune à un ensemble de concepts au niveau le plus haut de la hiérarchie. De cette manière, il n'est pas besoin de stocker en mémoire pour chaque concept l'ensemble des propriétés, et cela augmente le pouvoir d'expression de recherche d'information mais également des commodités de modélisation. Les propriétés qui lui sont propres suffisent, les autres peuvent être déduites. Cette organisation se retrouve dans la programmation orientée objet, où certains objets héritent des propriétés d'autres objets situés au-dessus dans la hiérarchie.

L'usage de la notion de réseau sémantique a été étendu à la représentation des connaissances en particulier pour modéliser les ontologies. On appelle ontologie, un réseau sémantique qui regroupe un ensemble de concepts décrivant complètement un domaine ou une partie d'un domaine. Les ontologies sont utilisées à des fins de définition, de représentation de fait ou de raisonnement. Elles ont pris un essor important avec le développement du Web sémantique. Une ontologie fournit une référence pour la communication entre les machines, mais aussi entre humains et machines en définissant le sens des objets. Outre ses apports évidents par rapport à des approches classiques d'annotation par mots clefs, les ontologies sont utilisées pour le partage et la réutilisation de données, ainsi que pour l'amélioration de la communication. Une ontologie possède des caractéristiques qui favorisent la réutilisation et le partage de données. Cependant, la conception d'ontologies communes occasionne des difficultés. Thomas Gruber, dans (Gruber, 1993) définit trois niveaux de convention pour mettre en place des techniques d'interopérabilité basées sur les communications :

- le format de représentation du langage
- le protocole de communication des agents
- la spécification du contenu du vocabulaire partagé

La communication peut se catégoriser en trois types distincts : communication homme-homme, homme-système ou système-système. Ils possèdent chacun des caractéristiques particulières qui posent des problèmes possiblement résolubles par les ontologies.

Pour synthétiser, les ontologies permettent de standardiser le vocabulaire, d'uniformiser le langage d'échange entre les différents acteurs, de comparer les différents systèmes, de structurer la connaissance pour simplifier l'analyse et la synthèse des connaissances d'un domaine, et de spécifier un contexte. Pour Mitsuru Ikeda (Ikeda, 1998), les ontologies peuvent servir de théorie pour supporter la recherche d'information tout comme les mathématiques apportent une base théorique à la physique. S'appuyant sur les ontologies, le développement du Web sémantique ouvre de nouvelles possibilités et des défis à la conception d'une

génération de systèmes adaptatifs, rendant possible la modélisation du profil des utilisateurs (Razmerita, 2003).

Dans le contexte d'adaptation d'information à l'utilisateur, l'ontologie est devenue une solution incontournable. L'ontologie permet la construction de modèles de connaissance complexes qu'il est possible d'utiliser pour modéliser à la fois les utilisateurs et le domaine de manière intelligible pour tout type d'utilisations et d'acteurs. Basée sur les outils du Web sémantique, l'adaptation du Web à l'utilisateur devrait permettre le développement d'un Web plus évolué, que l'on peut nommer le Web adaptatif.

Dans la couche sémantique de notre proposition, une seule ontologie est utilisée. Cependant, nous considérons deux sous-parties (ontologie de buts et ontologie de domaine) de cette ontologie dans la définition du modèle de buts et modèle de domaine.

2.3 MODELE DE DOMAINE

Le modèle de domaine permet de représenter le contenu selon la connaissance générale du domaine. Il est constitué d'une ontologie dite ontologie de domaine. Cette dernière est une sous-partie de l'ontologie globale utilisée par le système. Une ontologie du domaine est fonctionnelle et orientée objet. Elle est utilisée pour représenter un domaine (les composants informatiques, l'immobilier, le droit, la génétique, le tourisme...) sous forme de base de connaissances. Elle présente les concepts-clés, les attributs, les instances/individus relatifs au domaine. La Figure 9 montre un exemple d'ontologie de domaine du tourisme. L'ontologie de domaine est notée \mathcal{U} et est constituée, de manière simplifiée, d'un ensemble C de concepts, un ensemble R de relations et un ensemble I d'individus.

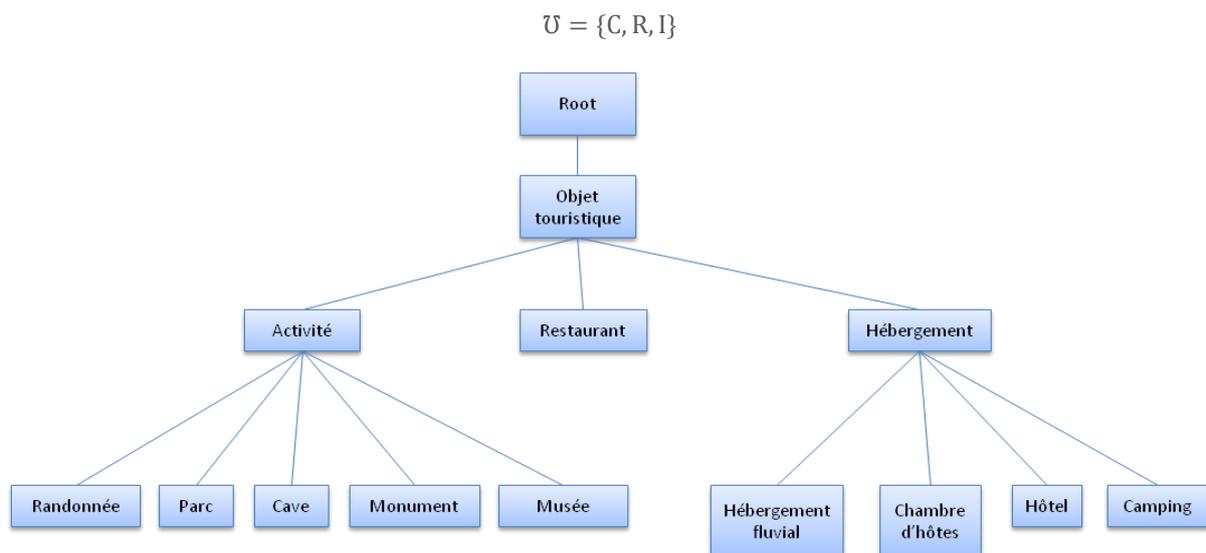


FIGURE 9. EXEMPLE D'ONTOLOGIE DU DOMAINE DU TOURISME

2.4 MODELE DE BUTS

Le modèle de buts représente le contenu selon la connaissance spécifique à l'application et suivant les possibles contraintes des utilisateurs. Le modèle de buts est composé d'une ontologie de buts (la deuxième sous-partie de l'ontologie globale) et d'un ensemble de règles.

L'ontologie de buts est seulement constituée de concepts et relations hiérarchiques entre ces concepts. Aucune instance n'est intégrée originellement. Chaque concept but représente un besoin/intérêt/but possible d'un utilisateur dans l'application considérée. A chaque but est ajoutée une relation de type attribut appelée « poids ».

Le modèle de buts est constitué, outre l'ontologie, d'un ensemble de règles dites règles métier. Elles sont formalisées en logique du premier ordre. Ces règles permettent de réorganiser les individus de l'ontologie de domaine dans les buts de l'ontologie de buts. L'idée est de réorganiser la connaissance générale du domaine de manière spécifique à l'application suivant les possibles contraintes des utilisateurs dans celle-ci.

L'ontologie de buts est notée \mathcal{B} et est constituée d'un ensemble B de concepts buts et d'un ensemble P de relations de type attribut sur ces concepts.

Soit $c_1, c_2 \in C$, des concepts de l'ontologie de domaine ; $r_1 \in R$, une relation de l'ontologie de domaine ; $b_1 \in B$, un concept de l'ontologie de buts ; $p_{b_1} \in P$, une relation de type attribut sur le concept b_1 dans l'ontologie de buts.

En logique du premier ordre, une règle de but peut être de la forme:

$$\forall x, y \in I. c_1(x) \wedge r_1(x, y) \wedge c_2(y) \rightarrow b_1(x) \wedge p_{b_1}(x, 5)$$

Cette règle signifie que tout individu appartenant au concept c_1 qui a une relation r_1 avec un individu du concept c_2 devient un individu du concept but b_1 auquel un poids de 5 est associé entre l'individu et le but.

Une règle de but peut être bien plus complexe, impliquant plus de relations et/ou de concepts.

La figure 10 montre un exemple d'ontologie de buts associé à l'ontologie de domaine de la figure 9.

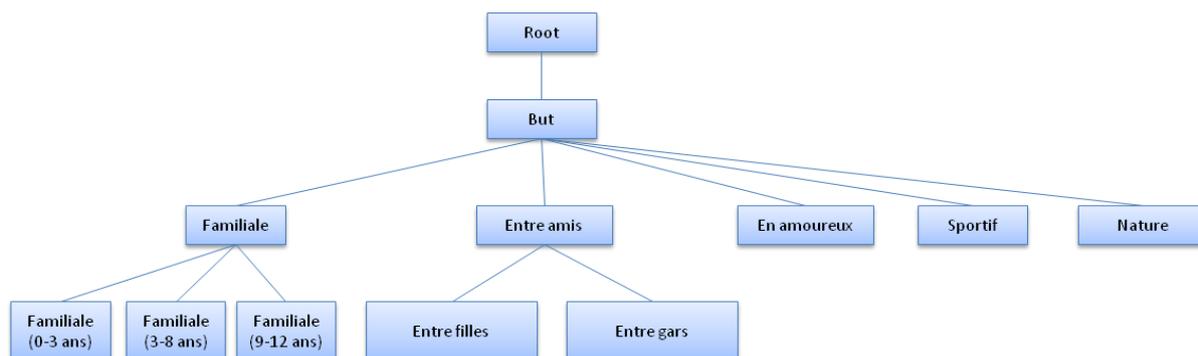


FIGURE 10. EXEMPLE D'ONTOLOGIE DE BUTS

Un exemple de connaissance métier associée au but "entre amis" peut être "Toute activité qui fait intervenir plusieurs participants est une activité entre amis". Sa traduction sous forme de règle est :

$$\forall x, y \in I. activite(x) \wedge plusieurs_participant(x, true) \rightarrow entre_amis(x) \wedge p_{entre_amis}(x, 10)$$

La règle et le poids forment la connaissance métier du fournisseur de contenu pour le but "entre amis". Le but, lui même, est une connaissance métier qu'il apporte.

Ainsi, il y a d'un côté une modélisation générale de la connaissance du domaine et d'un autre une modélisation de cette même connaissance spécifique aux contraintes (buts/intérêts/besoins) des utilisateurs dans le cadre d'une application donnée. La Figure 11 montre un aperçu de la structure de la couche sémantique. On peut y voir une ontologie globale composée de deux sous-parties, l'ontologie de buts et l'ontologie de domaine. L'ontologie de buts et les règles métiers constituent le modèle de buts, et l'ontologie de domaine représente le modèle de domaine.

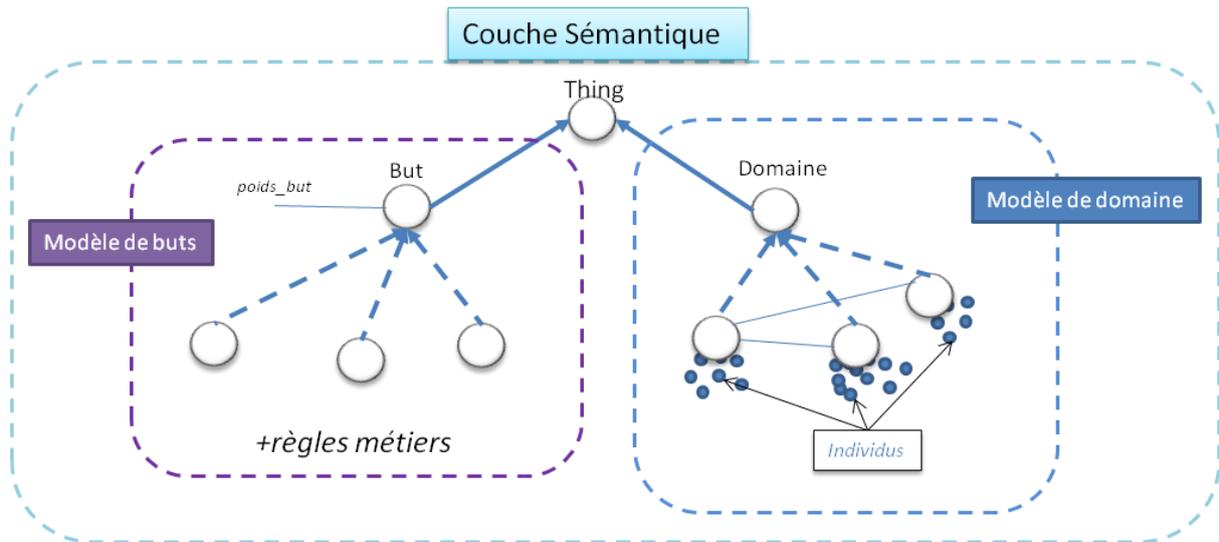


FIGURE 11. STRUCTURE DE LA COUCHE SEMANTIQUE

3 COUCHE UTILISATEUR

La couche utilisateur, aussi appelée modèle de l'utilisateur, permet de représenter les utilisateurs au sein du système de recommandation. Pour ce faire, ce modèle est constitué d'un ensemble de profils utilisateurs, chacun modélisant un utilisateur dans le système.

Le but du profil utilisateur est de stocker toutes données potentiellement utiles sur un utilisateur pour le processus de recommandation. Le profil d'un utilisateur est composé de deux parties principales: une partie indépendante du domaine et une partie dépendante du domaine.

La partie indépendante du domaine, aussi appelée partie statique, concerne toute donnée qui n'a aucun lien avec le domaine. Par exemple, il peut y être défini la géolocalisation, l'âge, la taille, le genre, la situation professionnelle de l'utilisateur.

La partie dépendante du domaine, ou partie dynamique, est constituée des données qui représentent les besoins, intérêts, buts de l'utilisateur en rapport avec le domaine pour l'application donnée. Il y a un lien direct entre le modèle de buts de la couche sémantique et la partie dynamique du modèle utilisateur. En effet, dans le modèle de buts, les concepts buts représentent chaque but possible d'un utilisateur dans l'application. La partie dynamique du profil utilisateur, quant à elle, représente tous les buts d'un utilisateur donné dans l'application. Ainsi, la partie dépendante du domaine est une sélection de concepts buts du modèle de buts.

Soit $C_u = \{p_0, p_1, \dots, p_n\}$, la couche utilisateur constituée de n profils utilisateurs p_i , $0 \leq i \leq n$. Un profil p_i est défini comme suit:

$$p_i = \{\{\text{Static}_{\text{part}}\}, \{\text{Dynamic}_{\text{part}}\}\}$$

$\text{Static}_{\text{part}}$ est la partie indépendante du domaine constituée de m paires attribut-valeur, elle est définie ainsi:

$$\text{Static}_{\text{part}} = \left\{ \left(\begin{array}{c} \text{attribut}_0 \\ \text{valeur}_0 \end{array} \right), \left(\begin{array}{c} \text{attribut}_1 \\ \text{valeur}_1 \end{array} \right), \dots, \left(\begin{array}{c} \text{attribut}_m \\ \text{valeur}_m \end{array} \right) \right\}$$

$\text{Dynamic}_{\text{part}}$ est la partie dépendante du domaine constituée de concepts buts de l'ontologie de buts \mathcal{B} .

$$\text{Dynamic}_{\text{part}} \subset \mathcal{B} \Leftrightarrow \text{Dynamic}_{\text{part}} = \{b_1, b_2, \dots, b_{NbB} / \forall 1 \leq j \leq NbB, b_j \in \mathcal{B}\}$$

Par exemple, un profil utilisateur p_6 peut être:

$$p_6 = \left\{ \left\{ \left(\begin{array}{l} \text{nom} \\ \text{marcel} \end{array} \right), \left(\begin{array}{l} \text{âge} \\ 25 \end{array} \right), \left(\begin{array}{l} \text{genre} \\ \text{masculin} \end{array} \right), \left(\begin{array}{l} \text{lattitude} \\ 58.645 \end{array} \right), \left(\begin{array}{l} \text{longitute} \\ 105.710 \end{array} \right) \right\}, \{ \text{Entre_amis}, \text{Sportif} \} \right\}$$

Tout utilisateur auquel seront proposées des recommandations est modélisé par un profil de ce type dans le système.

Basée sur cette modélisation de l'utilisateur et de la connaissance, une couche intelligence va pouvoir réaliser des recommandations.

4 COUCHE INTELLIGENCE

La couche intelligence correspond au modèle d'adaptation définie dans les systèmes hypermédia adaptatifs. C'est dans cette couche que sont implémentés les différents algorithmes utilisés pour proposer des recommandations aux utilisateurs. Les recommandations sont déduites à partir des profils des utilisateurs. Deux phases principales composent la couche intelligence : une phase de projection des individus du domaine suivant les profils utilisateurs et une phase de recherche combinatoire basée sur le résultat de la projection. La première phase a pour rôle de déduire un vecteur de poids d'intérêts de l'utilisateur pour chaque individu à partir du profil dynamique de l'utilisateur et du poids des individus de l'ontologie dans chaque but. A partir du résultat de la phase de projection, la deuxième phase recherche une combinaison des individus à proposer à l'utilisateur suivant la partie statique du modèle utilisateur.

4.1 PHASE DE PROJECTION

La phase de projection se base sur deux principes: Tout d'abord, chaque concept but est peuplé d'individus de l'ontologie de domaine grâce à des règles métier. Ensuite, les règles attribuent un poids à chaque individu pour chaque but auquel ils sont liés. Ces poids correspondent à la pertinence de l'individu dans le but associé. La partie dynamique du profil étant constituée de k buts, il est possible d'extraire les individus et leur poids associé dans ces buts. Pour chaque individu, la projection consiste à sommer ses poids dans les différents buts du profil utilisateur pour en extraire un seul poids. Ce dernier représente l'intérêt de l'utilisateur pour l'individu considéré. Ainsi, après la phase de projection des individus du domaine par rapport au profil utilisateur, on obtient une liste d'individus avec des poids d'intérêt.

4.1.1 FORMALISATION

Considérons une matrice BI de taille $N \times M$ représentant de manière simplifiée l'ontologie de but. N est le nombre de concepts buts et M est le nombre d'individus du domaine. La matrice relie chaque individu et chaque but par un poids.

$$BI = \begin{pmatrix} P_{11} & \dots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{m1} & \dots & P_{mn} \end{pmatrix}, \text{ où } P_{ij} \text{ est le poids de l'individu } i \text{ dans le but } j.$$

La partie dynamique du profil utilisateur est modélisée par un vecteur de taille N , N étant le nombre de buts.

$$\text{Profil} = \begin{pmatrix} K_1 \\ \vdots \\ K_n \end{pmatrix}, \text{ où } K_j = \begin{cases} 1, & \text{si } j \text{ est un but de l'utilisateur} \\ 0, & \text{sinon} \end{cases}$$

Par produit matriciel entre la matrice BI et le vecteur de profil, nous obtenons un nouveau vecteur que nous appellerons « le projeté de BI sur le profil », noté $\text{Proj}_{BI/\text{Profil}}$.

$$\text{Proj}_{\text{BI}/\text{Profil}} = \text{BI} \cdot \text{Profil} = \begin{pmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{m1} & \cdots & P_{mn} \end{pmatrix} \cdot \begin{pmatrix} K_1 \\ \vdots \\ K_n \end{pmatrix}$$

$$\text{Proj}_{\text{BI}/\text{Profil}} = \begin{pmatrix} P_{11} \times K_1 + P_{12} \times K_2 + \cdots + P_{1n} \times K_n \\ \vdots \\ P_{m1} \times K_1 + P_{m2} \times K_2 + \cdots + P_{mn} \times K_n \end{pmatrix}$$

Au final, la phase de projection engendre un vecteur contenant la somme des poids de chaque individu dans tous les buts choisis par l'utilisateur. Ainsi, nous avons un poids pour chaque individu du domaine représentant l'intérêt de l'utilisateur pour cet individu.

4.1.2 EXEMPLE

Considérons une matrice BI mettant en relation 5 items (activité₁, activité₂, activité₃, activité₄, activité₅) avec 3 buts (Nature, Entre_amis, Sportif) par un poids:

$$\text{BI} = \begin{pmatrix} 5 & 2 & 15 \\ 7 & 10 & 4 \\ 1 & 8 & 13 \\ 4 & 1 & 1 \\ 1 & 6 & 10 \end{pmatrix}$$

Dans cette matrice, les poids donnent un indice de pertinence (convenance) de l'item dans le but correspondant. Par exemple, l'item activité₁ a un poids de 15 dans le but Sportif, ce qui le place comme item le plus pertinent dans ce but, par rapport aux poids des autres items sur celui-ci.

Soit un utilisateur u₁ intéressé par les buts Entre_amis et Sportif. Ainsi, le vecteur de profil de u₁ noté Profil_{u₁} est :

$$\text{Profil}_{u_1} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

Le projeté de BI sur le profil de u₁ représentant l'intérêt de l'utilisateur u₁ sur les items du domaine est :

$$\text{Proj}_{\text{BI}/\text{Profil}_{u_1}} = \begin{pmatrix} 5 & 2 & 15 \\ 7 & 10 & 4 \\ 1 & 8 & 13 \\ 4 & 1 & 1 \\ 1 & 6 & 10 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 17 \\ 14 \\ 21 \\ 2 \\ 16 \end{pmatrix}$$

Ainsi, selon les buts de l'utilisateur u₁ et la matrice BI, l'item censé lui convenir le mieux parmi les 5 items est activité₃. Néanmoins dans notre problème, nous ne cherchons pas à proposer à l'utilisateur l'item qui lui correspond au mieux, mais une combinaison d'items dont l'ensemble lui convient.

4.2 PHASE DE RECHERCHE COMBINATOIRE

Grâce au vecteur projeté obtenu dans la phase de projection précédente, nous avons, pour chaque utilisateur, ses intérêts sur chaque individu du modèle de domaine. Ainsi, la phase suivante va consister à rechercher une meilleure combinaison de ces individus. Pour ce faire, plusieurs questions se posent:

- quelle doit être la forme de la combinaison ? Autrement dit, de quel type d'éléments doit-elle être composée?
- qu'est-ce qui détermine la pertinence d'une combinaison par rapport à une autre?
- quelle méthode utiliser pour renvoyer la meilleure combinaison d'individus ou du moins s'en approcher?

Les deux premières questions amènent à la définition de plusieurs concepts qui seront présentés dans les parties suivantes. La troisième question fait apparaître un problème d'optimisation combinatoire pour lequel il est nécessaire d'utiliser un algorithme d'optimisation. Pour cette recherche combinatoire, les individus du domaine à combiner sont appelés items.

4.2.1 FORME DE LA COMBINAISON

Une combinaison d'items peut prendre plusieurs formes, cela dépend du genre de proposition désirée dans l'application. Pour cela, nous définissons un pattern de combinaison. Le pattern définit le type des différents items, leur position et leur nombre dans la combinaison. Un item peut avoir plusieurs types. Ces derniers correspondent aux classes parentes de l'individu dans l'ontologie de domaine. Par exemple, dans la Figure 12, le type premier de l'item « Four n°405 » est « Four à chaleur tournante », mais aussi, en remontant dans l'arborescence de ses classes parentes dans l'ontologie de domaine, « Four », « Appareil électroménager » et « Thing ».

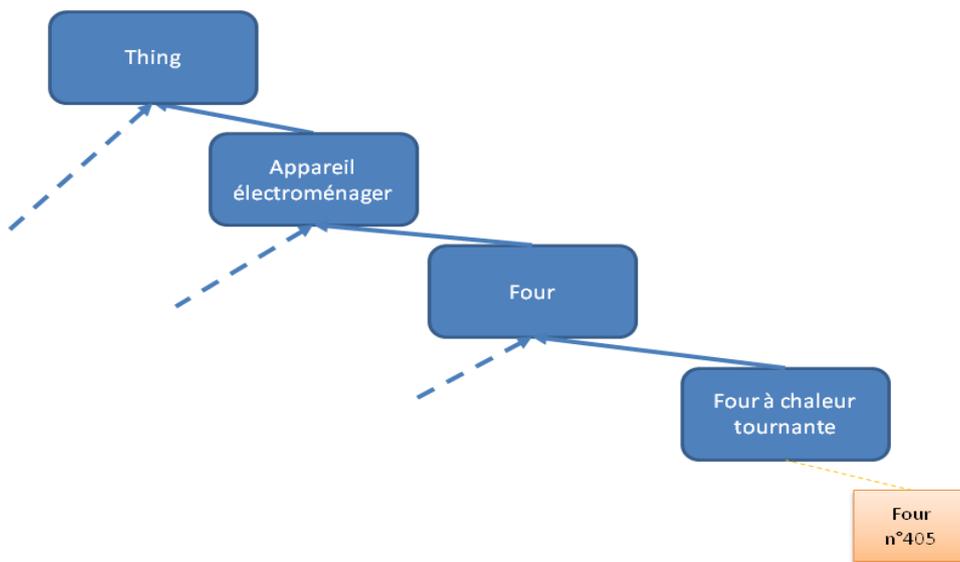


FIGURE 12. EXTRAIT D'UNE HIERARCHIE DE CLASSES D'UNE ONTOLOGIE DE DOMAINE

Ainsi, le pattern de combinaison est défini comme un vecteur montrant la position, le type et le nombre d'items de la combinaison à renvoyer:

$$\text{Pattern}_{\text{combinaison}} = \{\text{Type}_1, \text{Type}_2, \dots, \text{Type}_l\}, l \in \mathbb{N}$$

l est le nombre d'items de la combinaison et Type_e est l'identifiant de classe de l'ontologie de domaine désignant le type de l'item désiré à la position e de la combinaison, $0 < e \leq l$. Le pattern donne donc la forme de toute combinaison à proposer à l'utilisateur dans l'application.

Un exemple de pattern de combinaison peut être :

$$\text{Pattern}_{\text{combinaison}} = \{\text{Four}, \text{Cuisinière}, \text{Meuble de rangement}, \text{Réfrigérateur}, \text{Table}\}$$

4.2.2 PERTINENCE D'UNE COMBINAISON

Le pattern décrit la forme que doit avoir une combinaison. Cependant, comment pouvons-nous comparer deux combinaisons avec des items différents respectant ce pattern? Pour répondre à cette question, il faut définir la notion de pertinence d'une combinaison. Elle représente la pertinence d'une combinaison pour un utilisateur donné. La pertinence d'une combinaison C pour un utilisateur u est représentée par une fonction $f(C, u) = R$, où R est la valeur de pertinence. Plus R est élevé, meilleure est la combinaison pour l'utilisateur u . Le vecteur

projeté de la phase précédente donnant l'intérêt de l'utilisateur pour chaque individu/item du domaine, la fonction donnant la pertinence d'une combinaison de ces items pour un utilisateur donné se base nécessairement sur ce vecteur. Malgré ce constat, il n'est pas possible de définir une fonction générique de pertinence d'une combinaison. La pertinence dépend du domaine d'application. Par exemple, pour une application touristique, il serait possible d'utiliser les coordonnées géographiques des items et de l'utilisateur pour définir une fonction de pertinence. Dans un système de recommandation de produits électroménagers sous forme de cuisines aménagées, la fonction de pertinence peut, par exemple, aussi se baser sur la divergence des couleurs des différents items.

Dans le cas le plus simple, la fonction de pertinence peut simplement être la somme des poids d'intérêts de l'utilisateur pour les items de la combinaison, issue du vecteur projeté. Mais cela peut être beaucoup plus complexe, elle peut être la composition de plusieurs fonctions.

4.2.3 PROBLEME DE RECHERCHE DE LA MEILLEURE COMBINAISON

Suivant le pattern de combinaison et la fonction de pertinence, il est possible de construire des combinaisons et de les comparer. Ainsi, le problème de recherche de la meilleure combinaison pour un utilisateur donné est solvable en triant l'ensemble des combinaisons réalisables (combinaisons qui vérifient le pattern) suivant leur valeur de pertinence pour cet utilisateur. Une combinaison est aussi appelée solution du problème. Néanmoins, sauf cas particulier (nombre d'items faibles, pattern constitué de très peu d'éléments, espace d'items non discret permettant un tri sans énumération de toutes les solutions, ...), trier les solutions suivant leur pertinence revient à toutes les parcourir et à comparer les valeurs de pertinence. Une telle procédure n'est pas envisageable pour trouver la meilleure solution en un temps raisonnable pour des applications de recommandations temps réel. Ce problème est un problème d'optimisation combinatoire. En optimisation combinatoire, on considère un espace discret d'items dans lequel il faut rechercher le meilleur sous-ensemble (combinaison) réalisable selon une fonction « objectif ». Cette fonction « objectif » correspond à la fonction de pertinence évoquée précédemment. Ce genre de problème peut être résolu par des métaheuristiques qui proposent des solutions proches ou égales à la meilleure solution. Contrairement à des algorithmes basiques d'énumération de toutes les solutions possibles, une métaheuristique parcourt l'espace des solutions de manière non-linéaire. Elle possède un caractère semi-aléatoire.

Il n'y a pas de méthode générique pour le problème de recherche de meilleure combinaison. Certains algorithmes se prêtent mieux que d'autres suivant le niveau d'exigence en termes de qualité et de vitesse d'obtention d'une solution ou suivant la complexité de la fonction de pertinence (il peut y avoir plusieurs fonctions à optimiser). On peut aussi vouloir toute une population de bonnes solutions, plutôt qu'une seule solution, cela dépend de l'application. Il existe une multitude de métaheuristiques, les principales seront abordées dans la partie « Les métaheuristiques » du chapitre suivant.

4.3 RECAPITULATIF/APERÇU DE LA COUCHE INTELLIGENCE

La Figure 13 expose un récapitulatif des différentes interactions entre la couche intelligence et les éléments extérieurs à celle-ci. Il est en outre montré dans quelles phases ces éléments sont impliqués pour le processus de recommandation. Le modèle de buts et la partie dynamique du profil de l'utilisateur sont utilisés dans la phase de projection afin d'obtenir un ensemble d'individus du domaine avec des poids d'intérêts. Le modèle de domaine, la partie statique du profil utilisateur, le pattern de combinaison et la fonction de pertinence sont utilisés dans la phase de recherche combinatoire dans le but de générer des recommandations sous forme de combinaisons d'items.

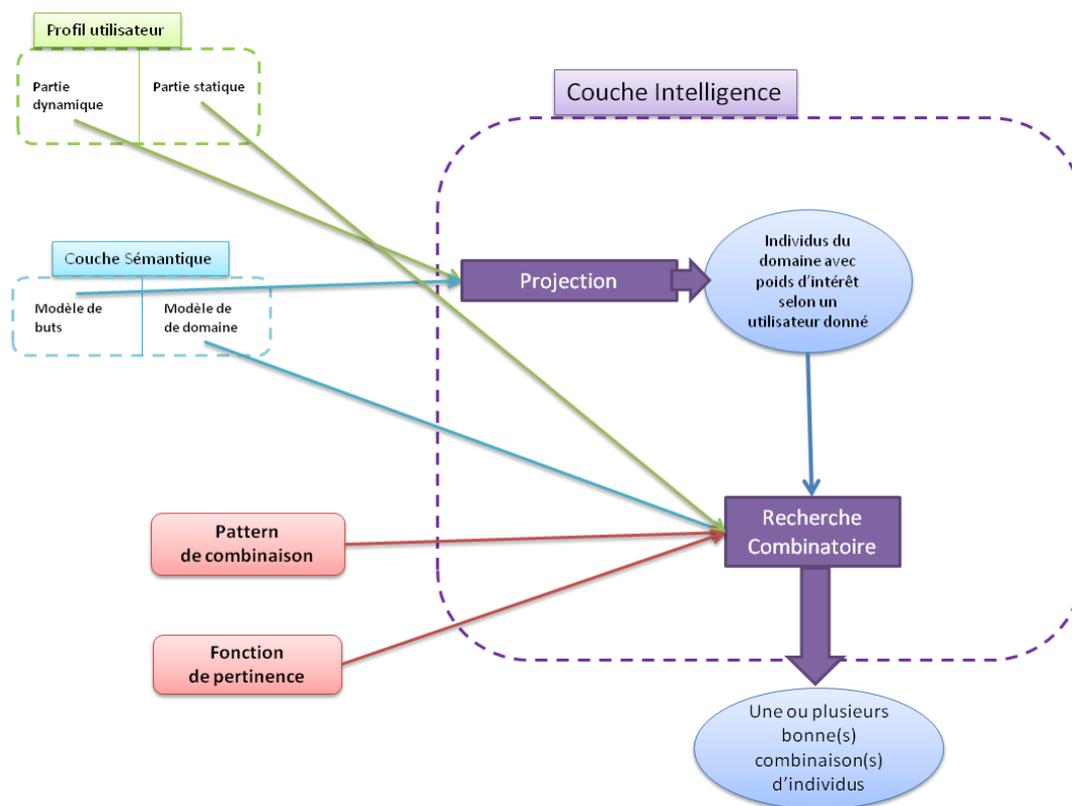


FIGURE 13. APERÇU DES DIFFÉRENTES INTERACTIONS DE LA COUCHE INTELLIGENCE

5 CONCLUSION

Basé sur une étude des limites des systèmes de recommandation existants, ce chapitre définit un nouveau type de système de recommandation proposant des combinaisons d'individus d'une ontologie de domaine et permettant la prise en compte, sous la forme de règles métiers et de modèle de but, du savoir-faire du fournisseur de contenus. Cette proposition est une extension des systèmes de recommandation basés sur le contenu utilisant une ontologie associée à des mécanismes de pondération et d'indexation (Figure 14). L'architecture utilisée est issue des systèmes hypermédia adaptatifs, elle est constituée des 3 couches : une couche sémantique, une couche intelligence et une couche utilisateur. La constitution des différentes couches a été décrite et les différents processus de la couche intelligence permettant la génération des propositions ont été expliqués.

Pour proposer des recommandations sous forme de combinaisons, le système est constitué de manière à pouvoir utiliser des algorithmes d'optimisation combinatoire adaptés aux domaines d'applications. En effet, nous verrons dans la suite que les problèmes d'optimisation sont propres au domaine d'application.

Dans la suite de cette thèse, nous présenterons une implémentation de cette architecture de système de recommandation dans le domaine touristique pour proposer des combinaisons de produits touristiques sous forme de séjours. Pour cela, nous avons dû identifier le problème d'optimisation de cet objectif applicatif. Nous avons identifié ce problème comme une variante du problème du sac à dos. Le chapitre suivant énonce formellement le problème en le situant par rapport au problème du sac à dos, et un état de l'art des principales métaheuristiques utilisables pour résoudre ce genre de problème est proposé.

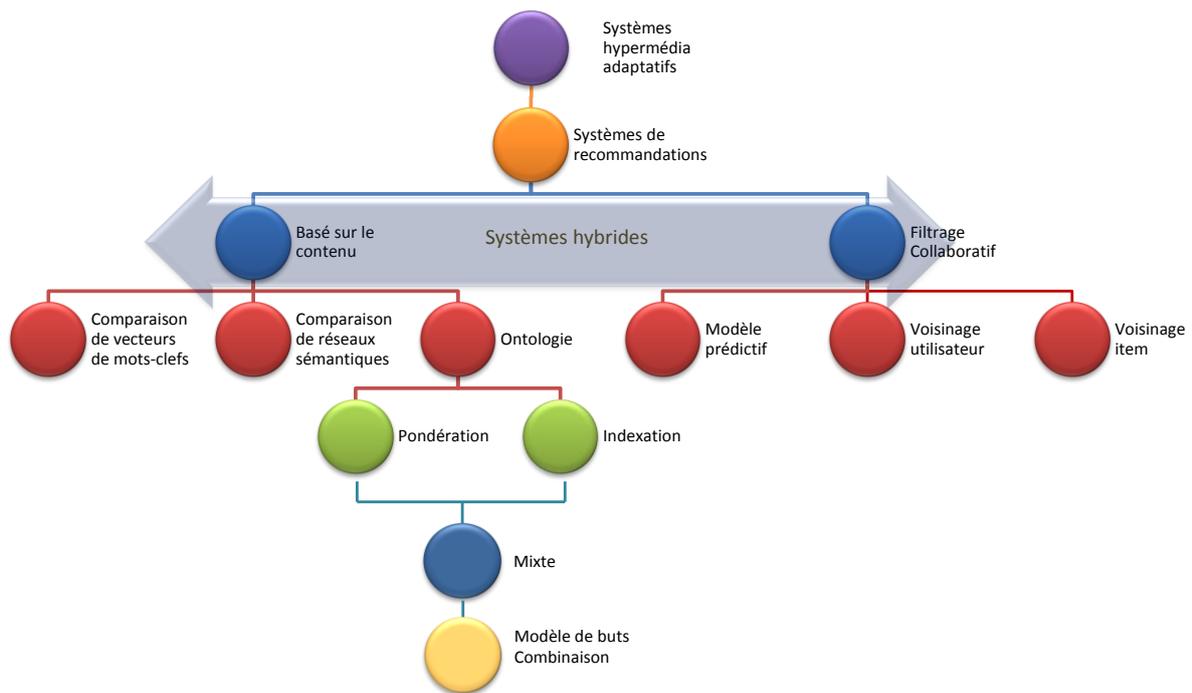


FIGURE 14. POSITIONNEMENT DE NOTRE PROPOSITION PAR RAPPORT AUX APPROCHES EXISTANTES

Chapitre 4

Problème d'Optimisation

Résumé

Le chapitre 4 a pour objectif de définir le problème d'optimisation du domaine du tourisme et de proposer des pistes pour la résolution de ce problème en utilisant des métaheuristiques. Ce chapitre définit le problème de combinaison d'objets touristiques hétérogènes comme une variante d'un problème de sac à dos. Le chapitre 4 présente une étude sur le domaine de l'optimisation combinatoire et définit les concepts nécessaires à la résolution des problèmes d'optimisation. Comme le problème d'optimisation touristique n'est pas solvable par des heuristiques classiques en un temps raisonnable, une dernière partie présente les métaheuristiques les plus connues pour résoudre des problèmes d'optimisation combinatoire.

Plan

1	Définition du problème d'optimisation touristique	73
1.1	Item et Poids.....	73
1.2	Pattern et Combinaison	74
1.3	Dispersion et Pertinence d'une Solution	74
1.4	Liens avec d'autres problèmes d'optimisation	75
2	Les métaheuristiques	76
2.1	Optimisation basée sur le Gradient.....	77
2.2	Les méthodes à état simple.....	80
2.2.1	Recherche locale et globale	80
2.2.1.1	Recherche locale.....	80
2.2.1.2	Recherche globale	81
2.2.2	Recuit simulé	83
2.2.3	Méthodes d'acceptation à seuil	85
2.2.4	Recherche Tabou.....	85

2.2.5	Recherche locale itérée	86
2.2.6	Méthode gloutonne	83
2.3	Méthodes avec population	87
2.3.1	Stratégies d'évolution	87
2.3.2	L'algorithme génétique	88
2.3.3	La colonie de fourmi.....	89
2.4	Méthodes multi-objectifs	90
3	Conclusion	91

Dans le chapitre précédent, l'architecture d'un nouveau type de système de recommandation basé sur le contenu proposant des combinaisons d'items à l'utilisateur et tenant compte du savoir-faire du fournisseur de l'offre a été proposé. Pour notre projet qui consiste à fournir une combinaison d'offres touristiques à l'utilisateur, nous désirons instancier cette architecture. Cependant, la notion de combinatoire implique la définition d'un problème d'optimisation. De ce fait, avant l'instanciation de l'architecture, il est nécessaire d'identifier au préalable le problème d'optimisation qui est posé dans la recherche de combinaisons d'objets touristiques.

Ainsi, ce chapitre définira formellement, dans une première partie, le problème d'optimisation touristique. Les différents concepts seront introduits et le problème sera situé par rapport aux problèmes d'optimisation existants. Ce problème ne pouvant être résolu par des algorithmes classiques en un temps raisonnable, la seconde partie abordera le domaine des métaheuristiques qui permettent d'approcher la solution optimale à un problème donné en un temps raisonnable. Les métaheuristiques les plus connues et les plus utilisées seront présentées. Cette partie permettra d'identifier la ou les méthodes qui se prêtent le plus à l'application et au problème d'optimisation qui nous est posé.

1 DEFINITION DU PROBLEME D'OPTIMISATION TOURISTIQUE

Le problème général qui nous est posé est de trouver un ensemble d'offres touristiques corrélées formant un séjour touristique. Cette partie a pour but de définir formellement le problème d'optimisation qui en découle. Cette partie est articulée en 4 sections. La première section présente les notions d'item et de poids. La deuxième section présente les notions de pattern et de combinaison. La troisième section présente les notions de dispersion et de pertinence. La dernière section met en relief notre problème avec d'autres problèmes de combinatoire approchant.

1.1 ITEM ET POIDS

Un touriste est représenté dans notre système par un profil utilisateur. Ces profils utilisateurs sont stockés dans le modèle des utilisateurs. Dans la suite, nous considérerons seulement un utilisateur, le processus pour produire une solution pouvant être réalisé indépendamment pour chaque utilisateur du système.

Soit Ty , l'ensemble des types d'activités possibles dans le système. Par exemple, un restaurant, un musée, ou un hôtel, sont des types d'activités standards.

Soit I , l'ensemble de tous les éléments que peut proposer le système dans la recommandation finale. Un item $i \in I$ est défini par un vecteur composé d'un nom $i.name$, d'un type $i.type \in Ty$, et de coordonnées géographiques $i.x$ et $i.y$.

$$i = \langle i.name, i.type, i.x, i.y \rangle$$

Le modèle de domaine représente la connaissance du domaine, grâce à l'utilisation d'une ontologie composée de concepts, relations entre concepts, et individus. Les items proposables sont intégrés dans l'ontologie en tant qu'individus. Par commodité, nous réduisons le modèle de domaine à l'ensemble I des items proposables. Le poids d'un item, noté w_i , représente l'intérêt de cet item pour l'utilisateur. Le calcul des poids est réalisé durant la phase de projection abordée dans le chapitre précédent.

Par exemple, soit les items i_0 et i_1 , représentant respectivement le Musée du Louvre et le musée d'Orsay, définis par $i_0 = \langle MuseeduLouvre, Musee, 2.34, 48.86 \rangle$, $i_1 = \langle MuseeOrsay, Musee, 2.43, 47.90 \rangle$ et les poids $w_{i_0} = 11$, $w_{i_1} = 2$. L'intérêt de l'utilisateur pour le Musée du Louvre est beaucoup plus grand que celui pour le musée d'Orsay.

1.2 PATTERN ET COMBINAISON

Lorsque l'utilisateur demande une proposition de séjour, un pattern de solution est établi. Ce dernier définit les types et enchaînements des items autorisés dans une solution.

$$P = (ty_1, ty_2, \dots, ty_{NbTy} / \forall 1 \leq j \leq NbTy, ty_j \in Ty)$$

Par exemple, un pattern de combinaison peut être $P = (Musee, Restaurant, Hotel)$. Notre système doit alors suggérer à l'utilisateur des combinaisons de ces trois éléments: une visite dans un musée, un dîner dans un restaurant, et finalement, une nuit dans un hôtel, chaque item n'étant pas trop éloigné géographiquement des autres.

Une combinaison valide C_k pour un pattern P est un ensemble d'items du modèle de domaine, choisi suivant les contraintes données par le pattern de combinaison.

$$C_k = (i_1, i_2, \dots, i_N), \forall 1 \leq j \leq N, i_j \in I, i_j.type = ty_j, ty_j \in P, 1 \leq k \leq NbC$$

Exemple: la combinaison simplifiée $C_3 = (MuseeduLouvre, RestaurantduPalais, HotelJacques)$ peut être une combinaison valide pour le pattern $P = (Musee, Restaurant, Hotel)$.

Toutes les combinaisons valides pour P peuvent être ordonnées de C_1 à C_{NbC} . Le poids d'une combinaison C_k , noté W_{C_k} , est égale à la moyenne du poids de chaque item de la combinaison. Il représente l'intérêt global de l'utilisateur pour la combinaison, sans considération de la distance entre les items.

ÉQUATION 13

$$W_{C_k} = \frac{\sum_{j=0}^N w_{i_j}}{N}$$

En plus du pattern principal de combinaison, il est possible d'ajouter plus d'exigences par la définition de sous-patterns de combinaisons. Exemple: Soit $p_1 = (Musee, Hotel)$ et $p_2 = (Restaurant, Hotel)$ deux sous-patterns de combinaisons, correspondant au pattern $P = (Musee, Restaurant, Hotel)$. La combinaison $C_3 = (MuseeduLouvre, RestaurantduPalais, HotelJacques)$ peut générer deux sous-combinaisons suivant les deux sous-patterns: $C_{3,1} = (MuseeduLouvre, HotelJacques)$ et $C_{3,2} = (RestaurantduPalais, HotelJacques)$.

1.3 DISPERSION ET PERTINENCE D'UNE SOLUTION

Le problème de recherche de la meilleure combinaison d'items pour un utilisateur donné requiert l'introduction de propriétés géographiques pour distinguer et comparer les solutions. En effet, le poids d'un item n'est pas suffisant pour évaluer la qualité d'une combinaison. Nous définissons une méthode pour évaluer la pertinence des combinaisons en prenant en compte la distance entre les items.

Pour une combinaison donnée C_k , la dispersion $\sigma(C_k)$ est égale à l'écart-type des coordonnées des items de la combinaison.

ÉQUATION 14

$$\sigma(C_k) = \sqrt{\frac{1}{N} \sum_{n=1}^N \left[\left(i_n \cdot x - \frac{1}{N} \sum_{j=1}^N i_j \cdot x \right)^2 + \left(i_n \cdot y - \frac{1}{N} \sum_{j=1}^N i_j \cdot y \right)^2 \right]}$$

Ainsi, pour chaque combinaison, la dispersion nous permet de quantifier la distance géographique entre les items de la combinaison. Une même dispersion peut être considérée différemment par deux utilisateurs. Ainsi,

nous définissons une tolérance de dispersion qui dépend de l'utilisateur concerné et du pattern de combinaison. La tolérance de dispersion, notée $Tol(P)$ est un nombre représentant la tolérance de l'utilisateur en termes de distance géographique sur les items pour le pattern P .

La dispersion modérée est introduite pour considérer de la même façon des combinaisons qui ont des valeurs de dispersion de même ordre selon la tolérance de dispersion de l'utilisateur. Elle est définie comme la partie entière par excès du quotient entre la dispersion de la combinaison et la tolérance de dispersion:

ÉQUATION 15

$$\sigma_{mod}(C_k) = \left\lceil \frac{\sigma(C_k)}{Tol(P)} \right\rceil$$

Ainsi, deux combinaisons avec des valeurs de dispersion variantes seulement de quelques mètres ne seront pas distinguées. Comme pour les combinaisons, chaque sous-combinaison possède une valeur de dispersion et de dispersion modérée. La pertinence d'une combinaison C_k est une agrégation de W_{C_k} et la dispersion modérée de C_k et de ses sous-combinaisons:

ÉQUATION 16

$$\Re(C_k) = W_{C_k} + \frac{PoidsMaxItem}{\sigma_{mod}(C_k) + \sum_{l=0}^{S-1} \sigma_{mod}(C_{k,l})}$$

où S est le nombre de sous-patterns associés au pattern P , $PoidsMaxItem$ le poids maximum que peut avoir un item. Ce dernier paramètre permet de donner une influence de la dispersion équivalente à celle du poids de la combinaison dans la fonction de pertinence.

Définition 1. *Problème touristique: Etant donné un ensemble d'items et patterns pour composer une combinaison valide d'items, le problème de recherche de la meilleure combinaison consiste à trouver pour chaque utilisateur une combinaison avec une pertinence maximale:*

$$\text{maximise } \Re(C_k), \quad 1 \leq k \leq NbC$$

1.4 LIENS AVEC D'AUTRES PROBLEMES D'OPTIMISATION

Ce problème touristique peut être vu comme un problème de « Set Packing » comme dans (Avella, Dauria, & and Salerno, 2006). Les auteurs proposent une variante du problème touristique, qu'ils appellent le problème du tourisme intelligent (Intelligent Tourist Problem). Leur modélisation inclut des items et poids pour les items dépendant des préférences utilisateurs, mais ils prennent seulement en compte un type d'item (activités touristiques). Ils ne suivent ainsi aucun pattern de combinaison. Ils ajoutent des items à la combinaison si la période de temps nécessaire pour l'activité correspond à la période de temps libre du touriste.

Notre problème a également des similarités avec un des principaux problèmes d'optimisation combinatoire appelé le problème de sac à dos ou « Knapsack Problem » (Karp., 1972). Ce problème dérive son nom du problème rencontré par quelqu'un qui est contraint par un sac de taille fixe et qui doit le remplir avec les objets les plus utiles. Chaque item possède un poids et une valeur. Les items mis dans le sac à dos doivent maximiser la valeur sans dépasser un poids total. Une formulation commune de ce problème, avec chaque item différent des autres, est définie comme suit (0-1 Knapsack Problem):

$$\text{maximise } \sum_{i=1}^n v_i x_i$$

avec

$$\sum_{i=1}^n w_i x_i \leq W, x_i \in \{0,1\}$$

où v_i et w_i sont respectivement la valeur et le poids de l'item i . W est le poids maximum du sac à dos. x_i est égale à 1 si i est mis dans le sac à dos, 0 sinon.

Ce problème a été prouvé NP-Complet dans (Lagoudakis, 1996). De nombreuses variantes du problème de sac à dos ont été étudiées : le sac à dos multidimensionnel (MKP), le sac à dos quadratique (QKP), etc. (voir (Martello & Toth, 1990)). Notre problème touristique se rapproche le plus du problème de sac à dos multidimensionnel à choix multiple MMKP (Multi-choice Multi-dimensional Knapsack Problem), dont il est une variante. Dans le MMKP standard, les items sont regroupés en fonction de leur type et seulement un représentant de chaque type doit être choisi. Or, dans notre problème, il ne s'agit pas de sélectionner systématiquement un objet de chaque type, plusieurs items d'un même type peuvent être choisis, en fonction des patterns et sous-patterns de combinaison. MMKP est NP-Difficile, ce qui implique qu'une méthode exacte ne serait pas efficace pour le résoudre, en particulier pour une application de décision temps réel (Chen, Khan, Li, & Manning, 1999).

La partie suivante introduit le domaine de l'optimisation combinatoire et présente plus particulièrement les métaheuristiques les plus connues pour résoudre ce genre de problème.

2 LES METAHEURISTIQUES

Le problème d'optimisation touristique défini dans la partie précédente est un problème d'optimisation combinatoire. L'optimisation combinatoire est le domaine des mathématiques discrètes qui traite de la résolution du problème suivant :

Soit X un ensemble de solutions admissibles. Soit f une fonction permettant d'évaluer chaque solution admissible. Il s'agit de déterminer une solution s^ appartenant à X qui minimise f . L'ensemble X des solutions admissibles est supposé fini et est en général défini par un ensemble C de contraintes.*

Malgré l'évolution permanente des calculateurs et les progrès fulgurants de l'informatique, il existera certainement toujours, pour un problème difficile, une taille critique de X au-dessus de laquelle même une énumération partielle des solutions admissibles devient prohibitive en temps de calcul. Compte tenu de ces difficultés, la plupart des spécialistes de l'optimisation combinatoire ont orienté leur recherche vers le développement de méthodes heuristiques, dites métaheuristiques. Une métaheuristique est souvent définie comme une procédure exploitant au mieux la structure du problème considéré, dans le but de trouver une solution de qualité raisonnable en un temps de calcul aussi faible que possible (Nicholson, 1971).

Bien que l'obtention d'une solution optimale ne soit pas garantie, l'utilisation d'une métaheuristique offre de multiples avantages par rapport à une méthode exacte :

- La recherche d'une solution optimale peut être totalement inappropriée dans certaines applications pratiques en raison de la dimension du problème, de la dynamique qui caractérise l'environnement de travail, du manque de précision dans la récolte des données, de la difficulté de formuler les contraintes en termes explicites ou de la présence d'objectifs contradictoires.
- Lorsqu'elle est applicable, une méthode exacte est souvent beaucoup plus lente qu'une métaheuristique, ce qui engendre des coûts informatiques supplémentaires et des difficultés au niveau du temps de réponse.

- Les principes de recherche qui sont à la base d'une métaheuristique sont en général plus accessibles aux utilisateurs non expérimentés. Le manque de transparence qui caractérise certaines méthodes exactes nécessite une intervention régulière de la part d'un spécialiste voire même du concepteur de la méthode.
- Une métaheuristique peut être facilement adaptée ou combinée avec d'autres types de méthodes. Cette flexibilité augmente considérablement les possibilités d'utilisation des métaheuristicques.

Les métaheuristicques sont utilisées pour trouver des réponses à des problèmes dont on n'a que très peu d'éléments: nous ne savons pas à quoi ressemble la solution optimale, nous ne savons pas comment la trouver de manière raisonnée, il y a très peu de renseignements heuristiques et la recherche par force brute est impensable, car l'espace est trop grand. Néanmoins, si une solution candidate au problème nous est fournie, nous sommes capables de la tester et d'évaluer sa qualité.

Par exemple, imaginons un problème dans lequel nous essayons de trouver un ensemble optimal de comportements pour un robot gardien de but au football. Un simulateur permet de tester chaque ensemble de comportements et nous sommes capables d'y associer une qualité. Dans ce problème, nous avons une idée générale de l'ensemble des comportements d'un robot, mais nous n'avons aucune idée de l'optimal et de la manière de le trouver.

La chose la plus simple à faire dans cette situation est la Recherche Aléatoire: on essaye des ensembles de comportements aléatoires jusqu'à ce que le temps imparti soit dépassé, et le meilleur ensemble trouvé est retourné. Cette méthode est basique et donne généralement de piètres résultats.

Un des algorithmes métaheuristicques les plus simples est la méthode de descente Hill-Climbing: à partir d'un ensemble de valeurs prises aléatoirement, une petite modification aléatoire est effectuée. Si la modification améliore l'ensemble, on la garde, sinon elle est rejetée. Ce processus est effectué autant qu'il est possible. De nombreuses métaheuristicques sont essentiellement élaborées par des combinaisons de cette méthode et de la recherche aléatoire.

Dans la suite, nous proposerons une vue d'ensemble du domaine en expliquant le fonctionnement des principales métaheuristicques. Une première partie abordera l'optimisation basée sur le gradient qui est une méthode traditionnelle mathématique pour trouver le maximum d'une fonction. Elle apporte les bases théoriques à la compréhension du fonctionnement des métaheuristicques. Ensuite, les parties 2.2 et 2.3 expliquent respectivement les méthodes à état simple et les méthodes avec population. Ce sont les deux grandes classes des métaheuristicques existantes. Enfin, une partie est consacrée aux méthodes multi-objectifs dont le but est d'optimiser, non pas une fonction, mais plusieurs fonctions objectif.

2.1 OPTIMISATION BASEE SUR LE GRADIENT

Avant d'aborder les métaheuristicques, il convient d'expliquer la méthode mathématique traditionnelle pour trouver le maximum d'une fonction: l'ascension gradient (Cauchy, 1847). L'idée est d'identifier la pente et de se déplacer vers le haut.

Soit $f(x)$ la fonction à maximiser. On commence avec une valeur arbitraire de x . On lui ajoute répétitivement une petite portion de sa pente $f'(x) : x = x + \alpha f'(x)$, où α est une petite valeur positive. Si la pente est positive, x croît, sinon il décroît. L'algorithme s'arrête lorsque la pente est nulle.

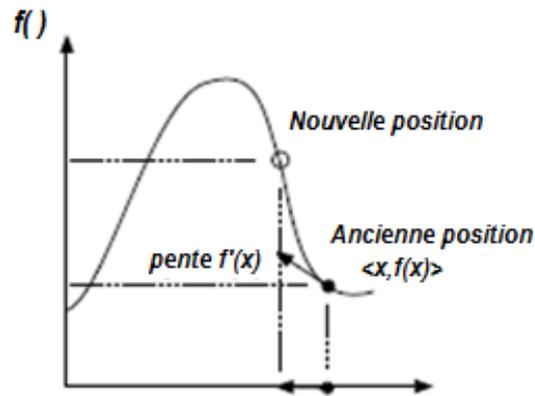


FIGURE 15. ASCENSION GRADIENT AVEC UNE PENTE NEGATIVE

Pour des fonctions à plusieurs dimensions, le principe est le même, mais la pente de la fonction est remplacée par le gradient de la fonction. x devient \vec{x} et $f'(x)$ devient $\nabla f(\vec{x}) = \langle \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \rangle$. En fait, la pente est calculée une fois sur toutes les dimensions.

ALGORITHME 1. ASCENSION GRADIENT

-
- 1: $\vec{x} \leftarrow$ valeur aléatoire initiale
 - 2: **répéter**
 - 3: $\vec{x} \leftarrow \vec{x} + \alpha \nabla f(\vec{x})$ ▷ Pour une dimension: $x \leftarrow x + \alpha f'(x)$
 - 4: **jusqu'à** \vec{x} est la solution idéale ou le temps imparti est atteint
 - 5: **renvoyer** \vec{x}

L'algorithme s'arrête quand le temps autorisé a été dépassé ou dès qu'il trouve la solution idéale (pente=0). Néanmoins, cela implique que l'arrêt s'effectue dès qu'il y a un optimum local ou qu'il y a un point d'inflexion (Figure 16).

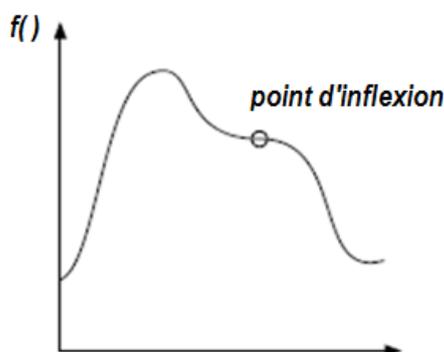


FIGURE 16. POINT D'INFLEXION

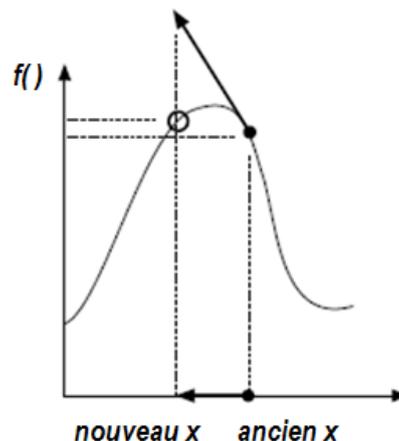


FIGURE 17. ASCENSION GRADIENT DEPASSANT LE MAXIMUM

Un des problèmes de cette méthode est le temps de convergence. En effet, lorsqu'elle est proche du maximum de la fonction, la méthode Gradient peut dépasser le sommet et se placer de l'autre côté de la pente à plusieurs reprises (Figure 17). Ceci est dû à la taille des sauts du Gradient basé sur la pente courante. Si la pente

est très raide, le saut sera très large même si ce n'est pas justifié. Une méthode pour traiter ce problème est d'ajuster la valeur de α à une petite valeur. Néanmoins, cela impliquera beaucoup de temps pour monter la pente. Au contraire, une grosse valeur pour α implique un temps de convergence très long, car la méthode Gradient va constamment dépasser le maximum. Ainsi, il faut trouver la bonne valeur de α . Pour une convergence plus rapide, la méthode Newton propose de considérer la dérivée seconde $f''(x)$ de la fonction, le pas étant défini par l'équation suivante :

ÉQUATION 17

$$x = x + \alpha \frac{f'(x)}{f''(x)}$$

Cette modification amortit α plus on approche d'une pente nulle. Le cas multidimensionnel est plus compliqué pour ce qui est de la seconde dérivée. Elle consiste en une matrice appelée Hessienne $H_f(\vec{x})$ constituée des dérivées partielles sur chaque dimension.

ÉQUATION 18

$$H_f(\vec{x}) = \begin{bmatrix} \frac{\partial}{\partial x_1} \frac{\partial f}{\partial x_1} & \frac{\partial}{\partial x_1} \frac{\partial f}{\partial x_2} & \dots & \frac{\partial}{\partial x_1} \frac{\partial f}{\partial x_n} \\ \frac{\partial}{\partial x_2} \frac{\partial f}{\partial x_1} & \frac{\partial}{\partial x_2} \frac{\partial f}{\partial x_2} & \dots & \frac{\partial}{\partial x_2} \frac{\partial f}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial}{\partial x_n} \frac{\partial f}{\partial x_1} & \frac{\partial}{\partial x_n} \frac{\partial f}{\partial x_2} & \dots & \frac{\partial}{\partial x_n} \frac{\partial f}{\partial x_n} \end{bmatrix}$$

ALGORITHME 2. METHODE NEWTON

1: $\vec{x} \leftarrow$ valeur aléatoire initiale

2: **répéter**

3: $\vec{x} \leftarrow \vec{x} + \alpha [H_f(\vec{x})]^{-1} \nabla f(\vec{x})$ ▷ Pour une dimension: $x \leftarrow x + \alpha \frac{f'(x)}{f''(x)}$

4: **jusqu'à** \vec{x} est la solution idéale ou le temps imparti est atteint

5: **renvoyer** \vec{x}

La méthode Newton converge généralement plus vite que le Gradient grâce à la dérivée seconde et elle permet aussi de savoir si on est sur un maximum, un point d'inflexion ou un minimum. Néanmoins, ces méthodes ont toutes le même problème, elles se bloquent dans des optima locaux. Les algorithmes du Gradient et de Newton sont des algorithmes d'optimisation locale. Plusieurs méthodes peuvent être utilisées pour améliorer ces algorithmes de recherche :

- donner de très grandes valeurs à α pour que l'algorithme dépasse le sommet actuel et atteigne le suivant.
- réaliser la méthode Newton plusieurs fois sur différents points de départ et retourner la meilleure solution trouvée.

L'optimisation basée sur le Gradient suppose qu'il est possible de calculer les dérivées successives de la fonction. Néanmoins, dans la plupart des cas, il n'est pas possible de calculer le gradient, car nous ne savons pas quelle est la fonction. Nous ne pouvons que modifier les éléments d'entrée de la fonction, les tester et

évaluer leur qualité. Ainsi, les métaheuristiques expliquées dans les parties suivantes proposent des solutions alternatives pour tendre vers l'optimum d'une fonction malgré l'impossibilité de calculer le gradient de celle-ci.

2.2 LES METHODES A ETAT SIMPLE

Les méthodes à état simple sont une première classe de métaheuristiques. Elles se caractérisent par le parcours de l'espace des solutions possibles en passant d'une solution à une autre de manière itérative. La plupart des méthodes à état simple sont un compromis entre une recherche locale et une recherche globale de l'espace des solutions possibles. Ces deux méthodes de recherche sont expliquées dans la partie qui suit. Ensuite, les métaheuristiques à état simple les plus représentatives utilisant ces principes seront détaillées. Les différences principales entre ces méthodes se situent au niveau du choix de la solution voisine et au niveau du critère d'arrêt. La recherche peut être clairement interrompue lorsqu'une solution suffisamment proche de la solution optimale est atteinte. Malheureusement, rares sont les problèmes difficiles où la valeur de la solution optimale est connue.

2.2.1 RECHERCHE LOCALE ET GLOBALE

Deux méthodes de recherche se distinguent lors du parcours de l'espace des solutions. Une première méthode est la recherche locale qui parcourt l'espace dans le voisinage de la solution courante. La deuxième méthode est dite recherche globale, elle ne se limite pas au voisinage et garantit un parcours complet de l'espace à terme. Cependant, le temps pour parcourir la totalité de l'espace peut être très grand.

2.2.1.1 RECHERCHE LOCALE

Les méthodes de recherche locale sont des algorithmes itératifs qui explorent l'espace X en se déplaçant pas à pas d'une solution à une autre. Une méthode de ce type débute à partir d'une solution $s_0 \in X$ choisie arbitrairement ou alors obtenue par le biais d'une méthode constructive. Le passage d'une solution admissible à une autre se fait sur la base d'un ensemble de modifications élémentaires qu'il s'agit de définir au cas par cas. Une solution s' est obtenue à partir de s en appliquant une modification élémentaire. Le voisinage $N(s)$ d'une solution $s \in X$ est défini comme l'ensemble des solutions admissibles atteignables depuis s en effectuant une modification élémentaire. Un tel processus d'exploration est interrompu lorsqu'un ou plusieurs critères d'arrêt sont satisfaits. Le fonctionnement d'une méthode de recherche locale est illustré de manière générale dans la Figure 18. Les passages successifs d'une solution à une solution voisine définissent un chemin au travers de l'espace des solutions admissibles. La modélisation d'un problème d'optimisation et le choix du voisinage doivent être effectués de telle sorte qu'il existe au moins un « chemin » entre chaque solution $s \in X$ et une solution optimale s^* . En effet, l'existence de tels chemins permet à la méthode de recherche locale d'atteindre une solution optimale à partir de n'importe quelle solution admissible.

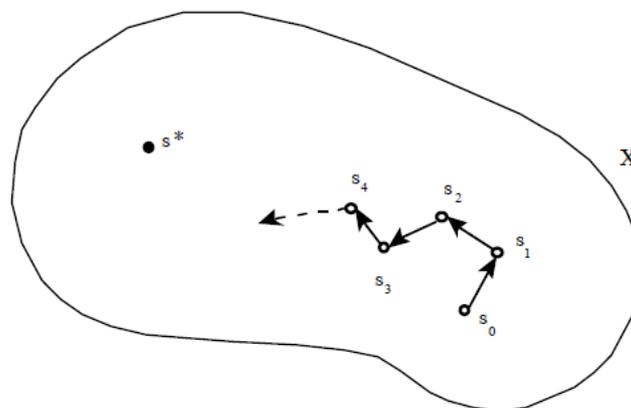


FIGURE 18. EXPLORATION DE X PAR UNE APPROCHE DE RECHERCHE LOCALE

La méthode de descente appelée Hill-Climbing, décrite de manière générique dans l'

Algorithme 3. Hill-Climbing, est un exemple de méthode de recherche locale. Cette technique est proche de la méthode Gradient, sans la nécessité de connaître le gradient. De nouvelles solutions candidates sont testées itérativement dans le voisinage du candidat courant, et elles sont adoptées si elles sont meilleures que la courante. Ce procédé est répété aussi longtemps que la valeur de la fonction objectif diminue. La recherche s'interrompt dès lors qu'un minimum local de f est atteint. La seule réelle différence avec la méthode Gradient est que l'amélioration de la solution courante se base sur une approche stochastique (partiellement aléatoire) pour chercher de meilleurs candidats.

ALGORITHME 3. HILL-CLIMBING

-
- 1: $S \leftarrow$ quelques solutions candidates ▷ Procédure d'initialisation
- 2: **répéter**
- 3: $R \leftarrow$ Modification(Copie(S)) ▷ Procédure de modification
- 4: **si** Qualité(R) > Qualité(S) **alors** ▷ Procédures d'évaluation et de sélection
- 5: $S \leftarrow R$
- 6: **jusqu'à** S est la solution idéale ou le temps imparti est atteint
- 7: **renvoyer** S

Historiquement, les méthodes de descente ont toujours compté parmi les méthodes heuristiques les plus populaires pour traiter les problèmes d'optimisation combinatoire. Toutefois, elles comportent deux obstacles majeurs qui limitent considérablement leur efficacité:

- suivant la taille et la structure du voisinage $N(s)$ considéré, la recherche de la meilleure solution voisine est un problème qui peut être aussi difficile que le problème initial;
- une méthode de descente est incapable de progresser au-delà du premier minimum local rencontré. Or les problèmes d'optimisation combinatoire comportent typiquement de nombreux optima locaux pour lesquels la valeur de la fonction objectif peut être fort éloignée de la valeur optimale.

Pour faire face à ces carences, des méthodes de recherche plus sophistiquées ont été développées au cours de ces vingt dernières années. Ces méthodes acceptent des solutions voisines moins bonnes que la solution courante afin d'échapper aux minima locaux de la fonction f . En règle générale, seule une portion du voisinage courant est explorée à chaque étape.

2.2.1.2 RECHERCHE GLOBALE

Une méthode de recherche globale est une méthode qui peut finalement trouver l'optimum global si on lui laisse suffisamment de temps. Pour le prouver, il suffit de montrer que tout l'espace de recherche serait visité à terme. Les algorithmes de descente ne garantissent pas cela.

Le plus simple algorithme d'optimisation globale est la Recherche Aléatoire :

ALGORITHME 4. RECHERCHE ALEATOIRE

-
- 1: Best \leftarrow quelques solutions initiales aléatoires
- 2: **répéter**

3: $S \leftarrow$ une solution aléatoire
4: **si** Qualité(S)>Qualité(Best) **alors**
5: Best \leftarrow R
6: **jusqu'à** Best est la solution idéale ou le temps imparti est atteint
7: **renvoyer** Best

Cette méthode est l'extrême pour de la recherche globale tandis que le Hill-Climbing est l'extrême pour la recherche locale. Lorsqu'une méthode tend vers la recherche aléatoire, on dit qu'elle réalise de **l'exploration ou diversification**. Au contraire, lorsqu'elle tend vers le Hill-Climbing, on dit qu'elle réalise de **l'exploitation ou intensification**.

Une technique populaire est de coupler les deux méthodes, appelée « Hill-Climbing avec redémarrages aléatoires ». La méthode de descente est lancée plusieurs fois en partant de différentes positions aléatoires, puis la meilleure solution trouvée est gardée.

ALGORITHME 5. HILL CLIMBING AVEC REDEMARRAGES ALEATOIRES

1: T \leftarrow distribution des intervalles possibles de temps
2: S \leftarrow quelques solutions initiales aléatoires
3: Best \leftarrow S
4: **répéter**
5: temps \leftarrow temps aléatoire choisi dans T
6: **répéter**
7: R \leftarrow Modification(Copie(S))
8: **si** Qualité(R)>Qualité(S) **alors**
9: S \leftarrow R
10: **jusqu'à** S est la solution idéale, ou le temps est écoulé, ou le temps total imparti est atteint
11: **si** Qualité(S)>Qualité(Best) **alors**
12: Best \leftarrow S
13: S \leftarrow des solutions aléatoires
14: **jusqu'à** Bestest la solution idéale ou le temps total imparti est atteint
15: **renvoyer** Best

Les méthodes qui seront présentées ci-dessous sont en général beaucoup plus performantes qu'une simple méthode de descente, mais également beaucoup plus coûteuses en termes de ressources informatiques. Leur mise en œuvre doit généralement tenir compte du temps de réponse maximal autorisé par l'utilisateur du

programme. Il convient de signaler pour conclure qu'un effort non négligeable est nécessaire pour ajuster convenablement les paramètres qu'elles font intervenir dans le but de guider efficacement la recherche au travers de l'ensemble X .

2.2.2 METHODE GLOUTONNE

Un algorithme glouton est un algorithme qui cherche à construire une solution optimale pas à pas, sans jamais revenir sur ses décisions, en prenant à chaque étape la solution qui semble la meilleure localement. Pour illustrer ce type de méthodes, il suffit d'imaginer un voyageur de commerce qui doit rendre visite à un ensemble de n clients. Il peut construire sa tournée de la manière suivante : en partant de chez lui, il va chez le client le plus proche (disons $C1$). En quittant $C1$, il va chez le client le plus proche de $C1$ qu'il n'a pas encore rencontré, et ainsi de suite jusqu'à qu'il ait rendu visite à tous ses clients. En quittant le dernier client (disons C_n), il rentre chez lui. Il a ainsi construit la tournée " $M - C1 - C2 - \dots - C_n - M$ " (Figure 19).

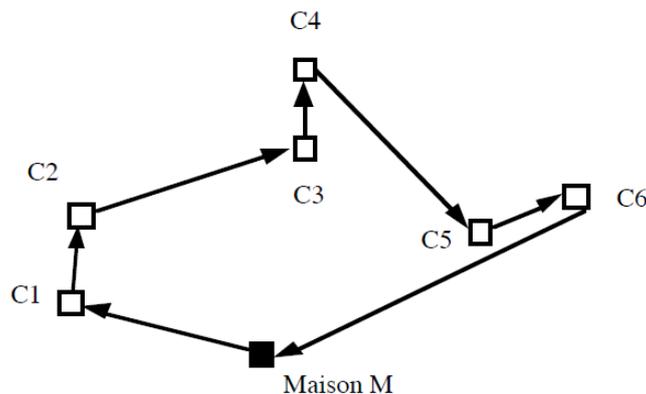


FIGURE 19. TOURNEE DU VOYAGEUR DE COMMERCE

Dans ce type de méthode, l'idée consiste à diminuer la taille du problème à chaque étape, ce qui revient à se restreindre à un sous-ensemble X^k inclus dans X toujours plus petit. Une solution optimale est trouvée lorsque chacun des sous-ensembles considérés contient au moins une solution optimale $s^* \in X$. Malheureusement, rares sont les cas où une telle condition est remplie avec certitude. A chaque étape, la solution courante est complétée de la meilleure façon possible sans tenir compte de toutes les conséquences que cela entraîne au niveau du coût de la solution finale. Dans ce sens, les méthodes gloutonnes sont souvent considérées comme myopes. Ces méthodes se distinguent par leur rapidité en générant une solution admissible sans avoir recours à des techniques hautement sophistiquées. Néanmoins, la qualité de la solution finale obtenue est leur principal défaut.

2.2.3 RECUIT SIMULE

Le recuit simulé a été développé par de nombreux chercheurs dans les années 1980. C'est un dérivé de l'algorithme de Metropolis (Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller, 1953) développé par des scientifiques ayant travaillé sur le projet Manhattan : Nicholas Metropolis, Arianna et Marshall Rosenbluth, et Augusta et Edward Teller en 1953. Leurs travaux ont abouti à un algorithme simple pour simuler l'évolution d'un système physique instable vers un état d'équilibre thermique à une température t fixée. L'état du système physique est caractérisé par la position exacte de l'ensemble des atomes qui le composent. Metropolis et al. utilisent une méthode de Monte Carlo pour générer une suite d'états successifs du système en partant d'un état initial donné. Tout nouvel état est obtenu en faisant subir un déplacement infinitésimal aléatoire à un atome quelconque. Soit ΔE la différence d'énergie occasionnée par une telle perturbation. Le nouvel état est accepté si l'énergie du système diminue ($\Delta E < 0$). Dans le cas contraire ($\Delta E \geq 0$), il est accepté avec une certaine probabilité (Équation 19).

ÉQUATION 19

$$\text{prob}(\Delta E, t) = e^{-\frac{\Delta E}{k_B \times t}},$$

où t est la température du système et k_B une constante physique connue sous le nom de constante de Boltzmann. A chaque étape, l'acceptation d'un nouvel état dont l'énergie n'est pas inférieure à celle de l'état courant est décidée en générant de manière aléatoire un nombre $q \in [0,1[$. Si q est inférieur ou égal à $\text{prob}(\Delta E, t)$, alors le nouvel état est accepté. Autrement l'état courant est maintenu. Metropolis et al. ont montré que l'utilisation répétée d'une telle règle fait évoluer le système vers un état d'équilibre thermique.

Le recuit simulé est une méthode de recherche locale dont le mécanisme de recherche est calqué sur l'algorithme de Metropolis et al. et les principes du recuit thermodynamique. L'idée consiste à utiliser l'algorithme de Metropolis et al. pour des valeurs décroissantes de la température t . Le refroidissement progressif d'un système de particules est simulé en faisant une analogie entre l'énergie du système et la fonction objectif du problème d'une part, et entre les états du système et les solutions admissibles du problème d'autre part. Pour atteindre des états avec une énergie aussi faible que possible, on porte initialement le système à très haute température puis on le refroidit petit à petit. Lorsque la température diminue, les mouvements d'atomes deviennent de moins en moins aléatoires et le système aura tendance à se trouver dans des états à basse énergie. Le refroidissement du système doit se faire très lentement pour avoir l'assurance d'atteindre un état d'équilibre à chaque température t . Lorsqu'aucun état nouveau n'est accepté à une température t donnée, on considère que le système est gelé et on suppose qu'il a atteint un niveau d'énergie minimum.

Ainsi, par rapport à l'algorithme du Hill-Climbing, l'algorithme du recuit simulé varie dans sa manière de choisir quand remplacer s (la solution originelle) par s' (la nouvelle solution qui est une modification élémentaire de s). Si s' est meilleur que s , s' remplace toujours s , mais si s' est moins bon que s , on ne peut remplacer s par s' qu'à une certaine probabilité $P(t, s', s)$:

ÉQUATION 20

$$P(t, s', s) = e^{\frac{\text{Quality}(s') - \text{Quality}(s)}{t}}$$

où $t \geq 0$. Ainsi, l'algorithme descend parfois les pentes (accepte de mauvaises solutions).

Cette équation est doublement intéressante :

- Premièrement, si s' est largement moins bon que s , alors la probabilité tend vers 0. Au contraire si s' est proche de s , alors la probabilité tend vers 1. Ainsi, lorsque s' n'est pas trop mauvais par rapport à s , on peut garder s' à une probabilité raisonnable. Deuxièmement, il y a un paramètre ajustable t . Si t tend vers 0, la probabilité tend vers 0. Si t est très grand, la probabilité se rapproche de 1. L'idée est d'initialiser t à une grande valeur pour bouger à chaque nouvelle solution créée sans quasiment prendre en compte leur qualité au départ, avec un parcours aléatoire de l'espace. Puis t décroît lentement jusqu'à 0, le comportement de l'algorithme revenant alors à celui du Hill-Climbing.

ALGORITHME 6. RECUIT SIMULE

1: $t \leftarrow$ température, initialement une grande valeur

2: $S \leftarrow$ quelques solutions initiales aléatoires

3: Best \leftarrow S

4: **répéter**

5: $R \leftarrow \text{Modification}(\text{Copie}(S))$

6: **si** $\text{Qualité}(R) > \text{Qualité}(S)$ ou un nbre aléatoire pris entre 0 et 1 $< e^{\frac{\text{Qualité}(R) - \text{Qualité}(S)}{t}}$ **alors**

7: $S \leftarrow R$

8: Décroissance de t

9: **si** $\text{Qualité}(S) > \text{Qualité}(\text{Best})$ **alors**

10: $\text{Best} \leftarrow S$

11: **jusqu'à** Bestest la solution idéale, le temps total imparti est atteint, ou $t < 0$

12: **renvoyer** Best

Le taux de décroissance de la température t est appelé taux d'amortissement. Plus cet amortissement est long, plus l'algorithme ressemble à un parcours aléatoire, et plus il réalise de l'exploration. Inversement, plus le taux est bas, plus l'algorithme réalise de l'exploitation.

2.2.4 METHODES D'ACCEPTATION A SEUIL

Les méthodes d'acceptation à seuil sont des méthodes de recherche locale qui dérivent directement de l'algorithme du recuit simulé. La différence essentielle entre ces deux méthodes se situe au niveau de l'acceptation d'une solution de moins bonne qualité à chaque étape. Dans une méthode d'acceptation à seuil, une telle décision est prise de manière déterministe, sans avoir recours aux principes du recuit thermodynamique. On se base uniquement sur une fonction auxiliaire $\gamma(s, s')$ et sur un seuil S qui peut éventuellement faire intervenir la valeur $f(s^*)$ de la meilleure solution trouvée jusque-là. La fonction $\gamma(s, s')$ et le seuil S peuvent être définis de nombreuses manières, ce qui donne lieu à plusieurs variantes pour les méthodes d'acceptation à seuil (Dueck & Scheuer, 1990).

2.2.5 RECHERCHE TABOU

La recherche Tabou de Fred Glover (Glover, 1986) emploie une approche différente d'exploration. Le déplacement d'une solution courante s vers une solution voisine s' est choisi de telle sorte que

$$f(s') = \min_{s'' \in N(s)} f(s'')$$

Tant que l'on ne se trouve pas dans un optimum local, la valeur de la fonction objectif est améliorée à chaque étape. Lorsque l'on atteint par contre un optimum local, la règle de déplacement donnée ci-dessus permet de choisir le moins mauvais des voisins, c'est-à-dire celui qui donne un accroissement aussi faible que possible de la fonction objectif. L'inconvénient que représenterait une méthode basée sur cet unique principe est que si un minimum local s se trouve au fond d'une vallée profonde, il sera impossible de ressortir de celle-ci en une seule itération, et un déplacement de la solution s vers une solution $s' \in N(s)$ avec $f(s') > f(s)$ peut provoquer le déplacement inverse à l'itération suivante, puisqu'en général $s \in N(s')$ et $f(s) < f(s')$; on risque ainsi de « cycler » autour de ce minimum local. De ce fait, la méthode tabou s'appuie sur un historique des candidats récents (liste Tabou) et refuse de retourner à ces candidats pendant un temps défini. Le but est de donner assez de temps à l'algorithme pour lui permettre de sortir d'un minimum local. Un des principaux problèmes de la recherche Tabou est que si l'espace de recherche est très large, il est facile de s'enfermer dans un même voisinage, même si la liste tabou est grande.

Une alternative à la recherche tabou appelée « Recherche Tabou basé sur les caractéristiques » propose de mettre dans la liste tabou, non pas les solutions rencontrées, mais les changements effectués pour arriver à ces solutions. Une condition d'arrêt doit être définie. En général, on se donne un nombre maximum nb_{max} d'itérations entre deux améliorations de la meilleure solution s^* rencontrée. Parfois, lorsqu'il est possible de déterminer une borne inférieure de la fonction objectif, on peut stopper la recherche lorsqu'on atteint une solution s de valeur proche de la borne inférieure.

2.2.6 RECHERCHE LOCALE ITEREE

Cet algorithme est une version plus recherchée de l'algorithme « Hill-Climbing avec redémarrages aléatoires ». Dans la Recherche Locale Itérée (Iterated Local Search ILS), les positions aléatoires de redémarrage du Hill-Climbing sont assez loin du démarrage référence de sorte de tomber sur un nouvel optimum local différent du précédent, mais pas trop loin. Lorsque la Recherche Locale Itérée découvre un nouvel optimum, il choisit soit de le prendre comme démarrage référence soit de garder l'ancien. Si on prend à chaque fois le nouvel optimum, cela revient un peu à un parcours aléatoire (une sorte de méta-exploration). Si on prend seulement les optima qui sont meilleurs que l'optimum courant, cela revient à un Hill-Climbing (une sorte de méta-exploitation). La Recherche Locale Itérée choisit souvent quelque chose d'intermédiaire.

ALGORITHME 7. RECHERCHER LOCALE ITEREE

```

1: T ← distribution des intervalles possibles de temps
2: S ← quelques solutions initiales aléatoires
3: H ← S                                ▷ Point de démarrage de l'optimum local courant
4: Best ← S
5: répéter
6:     temps ← temps aléatoire choisi dans T
7:     répéter
8:         R ← Modification(Copie(S))
9:         si Qualité(R) > Qualité(S) alors
10:            S ← R
11:     jusqu'à S est la solution idéale, ou le temps est écoulé, ou le temps total imparti
        est atteint
12:     si Qualité(S) > Qualité(Best) alors
13:         Best ← S
14:     H ← NewHomeBase(H, S)
15:     S ← Perturbation(H)
16: jusqu'à Bestest la solution idéale ou le temps total imparti est atteint
17: renvoyer Best

```

Le comportement de cet algorithme est identique au Hill Climbing pendant un certain temps ; puis (quand le temps est écoulé), il détermine s'il doit adopter ou non l'optimum local découvert (fonction `NewHomeBase()`) ; puis à partir du nouveau démarrage de référence, il réalise une très grosse modification (fonction `Perturbation()`), qui est idéalement assez large pour aller vers un autre optimum. Les fonctions `NewHomeBase` et `Perturbation` dépendent beaucoup de l'application. Il est possible d'appliquer une approche par recuit simulé ou par recherche Tabou au `NewHomeBase`.

2.3 METHODES AVEC POPULATION

Les méthodes basées sur une population diffèrent des méthodes à états simples du fait qu'elles gardent un échantillon de solutions candidates au lieu d'une seule solution candidate. La plupart de ces méthodes se base sur des concepts de biologie. Une technique particulièrement populaire est connue sous le nom Calcul Evolutionnaire qui s'inspire de l'évolution, de la génétique et de la biologie. Les algorithmes utilisant ces techniques sont appelés Algorithmes Evolutionnaires. Les techniques de calcul évolutionnaires sont généralement des techniques de ré-échantillonnage : de nouveaux échantillons (populations) sont générés ou modifiés en se basant sur les précédents. L'algorithme générationnel évolutionnaire de base construit dans un premier temps une population initiale, puis itère à travers 3 procédures. D'abord, il évalue le fitness (la qualité) de tous les individus dans la population. En second, il utilise l'information sur le fitness pour engendrer une nouvelle population d'enfants. Troisièmement, il associe les parents et enfants pour former la nouvelle génération de population, et le cycle continue.

ALGORITHME 8. L'ALGORITHME GENERATIONNEL EVOLUTIONNAIRE DE BASE

```

1: P ← Population initiale
2: Best ← ∅
3: répéter
4:   EvaluerFitness(P)
5:   pour chaque  $P_i \in P$  faire
6:     si Best = ∅ ou  $Fitness(P_i) > Fitness(Best)$  alors
7:       Best ←  $P_i$ 
8:   P ← Jointure(P, Reproduction(P))
9: jusqu'à Bestest la solution idéale ou le temps total imparti est atteint
10: renvoyer Best

```

2.3.1 STRATEGIES D'EVOLUTION

Les stratégies d'évolution ont été développées par Ingo Rechenberg (Rechenberg, 1973) et Hans-Paul Schwefel (Schwefel, 1995) à l'Université Technique de Berlin. Les stratégies d'évolution emploient une procédure simple pour sélectionner des individus appelée Sélection par Troncature, et utilisent (habituellement) la mutation comme seul opérateur d'amélioration. Le plus simple algorithme issu des stratégies d'évolution est l'algorithme (μ, λ) . Cet algorithme démarre avec une population de λ individus, générés aléatoirement. Puis, il itère la procédure qui suit. Premièrement, il évalue le fitness de tous les individus. Puis, il supprime tous les individus de la population sauf les μ plus forts. Chacun des μ plus forts individus produit λ/μ enfants par des Mutations ordinaires. Au total, λ enfants sont créés. L'opération de jointure est simple : les enfants remplacent les parents. Enfin, la procédure se réitère.

```

1:  $\mu \leftarrow$  nombre de parents sélectionnés
2:  $\lambda \leftarrow$  nombre d'enfants générés par les parents
3:  $P \leftarrow \{\}$ 
4: pour  $\lambda$  fois faire                                 $\triangleright$  Construction de la population initiale
5:    $P \leftarrow P \cup \{\text{un nouvel individu aléatoire}\}$ 
6:  $Best \leftarrow \emptyset$ 
7: répéter
8:   pour chaque  $P_i \in P$  faire
9:     EvaluerFitness( $P_i$ )
10:    si  $Best = \emptyset$  ou  $Fitness(P_i) > Fitness(Best)$  alors
11:       $Best \leftarrow P_i$ 
12:    $Q \leftarrow$  les  $\mu$  individus de  $P$  dont les  $Fitness()$  sont les plus grands     $\triangleright$  Troncature
13:    $P \leftarrow \{\}$                                  $\triangleright$  La jointure est faite en remplaçant  $P$  par les enfants
14:   pour chaque individu  $Q_j \in Q$  faire
15:     pour  $\lambda/\mu$  fois faire
16:        $P \leftarrow P \cup \{\text{Mutation}(\text{Copie}(Q_j))\}$ 
17: jusqu'à Bestest la solution idéale ou le temps total imparti est atteint
18: renvoyer Best

```

L'algorithme (μ, λ) a 3 variables sur lesquelles on peut jouer pour ajuster l'exploration ou l'exploitation.

- La taille de λ contrôle essentiellement la taille de l'échantillon de chaque population. De manière extrême, si λ tend vers l'infini, l'algorithme approche la recherche aléatoire.
- La taille de μ contrôle la sélectivité de l'algorithme, plus la valeur est basse par rapport à λ , plus l'algorithme fait de l'exploitation, étant donné que seulement les meilleurs individus survivent.
- Le degré de mutation : plus les mutations sont bruitées, plus les nouveaux enfants s'éloignent des parents et sont clairement aléatoires, quelle que soit la sélectivité de μ .

Le deuxième algorithme issu des stratégies évolutionnaire est appelé $(\mu + \lambda)$. Il diffère de (μ, λ) sur l'opération de jointure. Dans (μ, λ) , les parents sont simplement remplacés par les enfants dans la génération suivante. Mais dans $(\mu + \lambda)$, la génération suivante consiste des μ parents et des λ nouveaux enfants. Ainsi, chaque génération suivante a une taille de $(\mu + \lambda)$ individus.

2.3.2 L'ALGORITHME GENETIQUE

L'algorithme génétique a été inventé par John Holland (Holland, 1975) à l'université du Michigan dans les années 1970. Il est similaire à la stratégie d'évolution (μ, λ) sur plusieurs aspects : il y a de l'évaluation de fitness, de la sélection et reproduction, et un réassemblage de population. La principale différence est la façon

dont la sélection et la reproduction sont faites : alors que les stratégies d'évolution sélectionnent les parents puis créent les enfants, l'algorithme génétique sélectionne petit à petit peu de parents et génère des enfants jusqu'à ce qu'assez d'enfants soient créés.

ALGORITHME 10. L'ALGORITHME GENETIQUE

```

1: poptaille ← taille de la population désirée                ▷ Basiquement λ
2: P ← {}
3: pour poptaille fois faire
4:     P ← P ∪ {un nouvel individu aléatoire}
5: Best ← ∅
6: répéter
7:     pour chaque Pi ∈ P faire
8:         EvaluerFitness(Pi)
9:         si Best = ∅ ou Fitness(Pi) > Fitness(Best) alors
10:            Best ← Pi
11:     Q ← {}                                                ▷ Différence par rapport à (μ, λ)
12:     pour poptaille/2 fois faire
13:         Parent Pa ← SelectionAvecRemplacement(P)
14:         Parent Pb ← SelectionAvecRemplacement(P)
15:         Enfants Ca, Cb ← Croisement(Copie(Pa), Copie(Pb))
16:         Q ← P ∪ {Mutation(Ca), Mutation(Cb)}
16:     P ← Q                                                  ▷ Fin de la différence
17: jusqu'à Bestest la solution idéale ou le temps imparti est atteint
18: renvoyer Best

```

Classiquement, l'algorithme génétique opère sur des vecteurs de booléens, mais il peut être appliqué à toute sorte de vecteurs.

2.3.3 LA COLONIE DE FOURMIS

Un autre algorithme très connu et largement inspiré de la biologie est l'algorithme de la colonie de fourmis. Pour comprendre le principe, il suffit d'imaginer l'exemple d'une colonie de fourmis qui est à la recherche d'une source de nourriture. Initialement, les fourmis quittent le nid et se déplacent de manière aléatoire. Lorsqu'une fourmi découvre par hasard une source de nourriture, elle informe ses congénères de sa découverte en déposant sur le sol une marque transitoire lors de son retour vers le nid. Cette marque n'est autre qu'une substance chimique, nommée phéromone, qui va guider les autres fourmis vers la même source de nourriture. A leur retour, ces dernières déposent également de la phéromone sur le sol et renforcent ainsi le marquage de la piste qui mène du nid à la source de nourriture découverte. Le renforcement du marquage par

les phéromones de la piste la plus fréquentée optimise la collecte de nourriture. A long terme, les fourmis exploiteront uniquement la source la plus proche, car la trace conduisant aux sources éloignées s'évaporerait et deviendrait indécélable. Cet exemple montre que la colonie de fourmis converge vers une solution optimale alors que chaque fourmi n'a accès qu'à une information locale et qu'elle est incapable de résoudre seule le problème dans un délai raisonnable. L'algorithme par colonie de fourmis a été développé initialement par Colorni et al. (Colorni, Dorigo, & Maniezzo, 1991). C'est une méthode évolutive dont les mécanismes de recherche s'inspirent fortement du comportement collectif d'une colonie de fourmis.

On distingue deux différents types de populations dans cet algorithme. Premièrement, il y a l'ensemble des composants qui engendrent des solutions candidates au problème. Dans le problème du voyageur de commerce, ce serait tous les sommets. L'ensemble de ces composants ne change jamais, mais leur fitness (appelé phéromone) sera ajusté au cours du temps. A chaque génération, une ou plusieurs solutions candidates, appelées sentiers de fourmis, sont construites par sélection des composants un par un, en se basant en partie sur leur phéromone. Cela constitue la deuxième population dans l'algorithme des fourmis: l'ensemble des sentiers. Ensuite, le fitness de chaque sentier est évalué. Pour chaque sentier, chaque composant de ce sentier est alors mis à jour en fonction du fitness: la valeur de fitness est répartie par phéromone sur l'ensemble des composants du sentier.

2.4 METHODES MULTI-OBJECTIFS

Dans toutes les méthodes proposées précédemment, nous ne considérons qu'une seule fonction objectif à optimiser. Néanmoins, de nombreux problèmes nécessitent l'optimisation de plusieurs objectifs à la fois. Par exemple, on peut imaginer un ingénieur qui cherche à construire un immeuble peu cher, grand, résistant aux séismes et avec peu de déperditions énergétiques. Chacune de ces fonctions à optimiser est appelée un objectif.

Pour revenir à un problème plus simple, une méthode naïve consiste à considérer l'ensemble des objectifs dans une seule fonction de fitness où chaque objectif possède un poids selon son importance dans la fonction:

$$\text{Fitness}(i) = \text{faibleCoût}(i) + \frac{1}{10} \text{taille}(i) + \frac{1}{5} \text{résistanceSéisme}(i) + \frac{1}{4} \text{efficacitéEnergétique}(i)$$

Néanmoins, avec ce genre de fonction, on est obligé de définir le degré pour lequel un objectif est meilleur qu'un autre, ce que l'on ne sait pas forcément. De plus, cela peut être impossible si les objectifs sont non linéaires. Lorsqu'on veut optimiser plusieurs objectifs, le cas idéal serait de trouver une solution qui est optimale sur tous les objectifs. Mais la plupart du temps, ce n'est pas le cas, il y a un ensemble de bonnes solutions qui sont des compromis des divers objectifs. L'ingénieur sait qu'il ne peut trouver l'immeuble parfait, mais il pourrait être intéressé par les meilleures options qui lui sont offertes. Il y a de nombreuses façons de définir un ensemble de meilleures solutions, mais la manière prédominante est le front de Pareto de l'espace des solutions candidates.

Soit deux immeubles candidats, M et N. M est dit Pareto dominant de N si M est au moins aussi bon que N dans tous les objectifs, et supérieur à N dans au moins un objectif (voir la Figure 20).

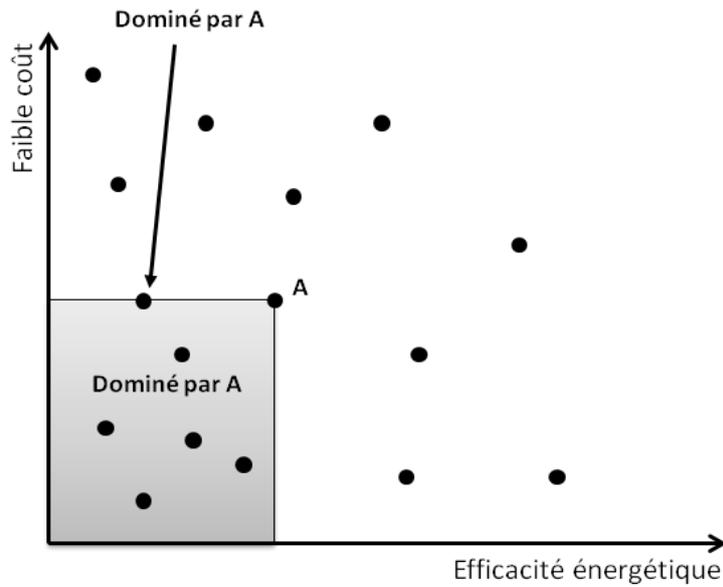


FIGURE 20. REGION DES SOLUTIONS PARETO DOMINEE PAR LA SOLUTION A, INCLUANT LES SOLUTIONS DU BORD

Si M et N sont identiques pour tous les objectifs, ou si M est meilleur que N dans certains objectifs et N meilleur que M dans d'autres objectifs, on ne peut dire qui de M ou N domine l'autre. Dans ce cas, les deux sont intéressants pour l'ingénieur. Ainsi, une autre façon de définir les meilleures solutions est l'ensemble des immeubles qui ne sont dominés par aucun autre. On dit que ces immeubles sont non-dominés. Cet ensemble d'immeubles est le front de Pareto des solutions possibles

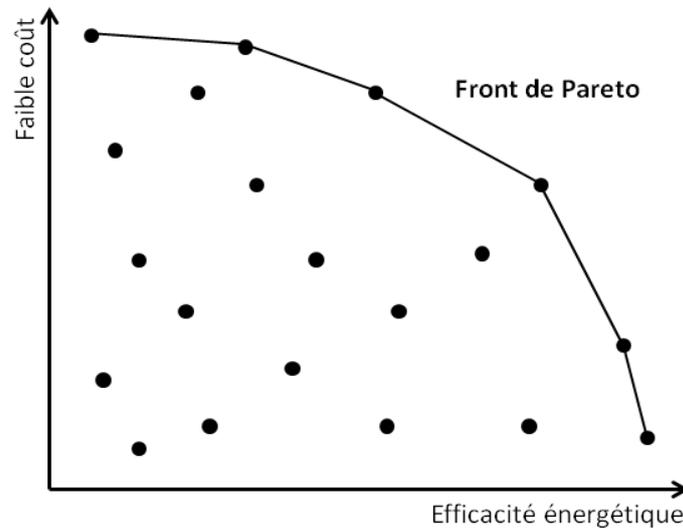


FIGURE 21. FRONT DE PARETO DES SOLUTIONS NON-DOMINEES

Ainsi, le but d'une méthode multi-objectif est de rechercher le front de Pareto de toutes les solutions possibles (Figure 21). Pour ce faire, étant donné que nous recherchons plusieurs bonnes solutions, il est fréquent d'utiliser des métaheuristiques travaillant sur des populations

3 CONCLUSION

Ce chapitre est constitué de deux grandes parties. La première partie a défini le problème d'optimisation combinatoire touristique. Le problème a été défini comme la recherche d'une combinaison d'items qui

maximise une fonction de pertinence. Ces combinaisons sont contraintes par un pattern de combinaison qui contraint l'ordonnement des items dans la combinaison et leur type. De plus, des informations spatiales sur la tolérance des utilisateurs envers la dispersion des coordonnées des items modifient le comportement de la fonction de pertinence. Ce problème d'optimisation touristique est une variante du problème d'optimisation combinatoire MMKP, problème de sac à dos multidimensionnel à choix multiple qui est NP-Difficile.

La deuxième partie a présenté les principales métaheuristiques utilisées pour résoudre des problèmes d'optimisation. Elles ont été regroupées en deux grandes familles, les métaheuristiques à état simple et les métaheuristiques basées sur une population. La première famille recherche la meilleure solution en parcourant l'espace des solutions en passant d'une solution à une autre de manière itérative. Les différences principales entre ces méthodes se situent au niveau du choix de la solution voisine et au niveau du critère d'arrêt. La deuxième famille recherche la meilleure solution en se basant sur un échantillon de solutions candidates au lieu d'une seule solution candidate. Les différences entre ces méthodes résident principalement dans les techniques de sélection des individus de la population, de croisement entre ces individus, de mutations et de l'intégration des nouveaux individus dans l'échantillon de population suivant. De plus, l'ordonnement de ces techniques diverge d'une métaheuristique à une autre.

Les métaheuristiques basées sur une population sont généralement plus coûteuses, en temps de calcul, que les métaheuristiques à état simple pour trouver la meilleure ou une bonne solution. Cependant, un de leurs intérêts est qu'elles engendrent une population de solutions tendant vers l'optimum alors que celles à état simple ne proposent qu'une seule solution. Pour le problème qui nous est posé, et surtout pour l'application à fournir à l'utilisateur final, il n'est pas envisagé, dans un premier temps, de lui fournir plusieurs combinaisons possibles. C'est cette dernière condition qui pourrait justifier l'utilisation de métaheuristiques basées sur une population demandant un plus long temps de calcul.

De ce fait, une métaheuristique à état simple sera choisie pour l'application finale. Celles proposées précédemment possèdent des avantages et des inconvénients à prendre en compte pour ce choix. La méthode de descente Hill-Climbing possède l'avantage de trouver très rapidement une solution, néanmoins c'est un algorithme de recherche locale, il s'enferme dans des optima locaux dont la qualité ne serait pas suffisante pour l'application. Dans la recherche locale itérée, l'utilisation itérative d'une méthode de descente en de multiples points de l'espace des solutions augmente les chances de trouver l'optimum global ou, du moins, de trouver une bonne solution. Cependant, l'appel itératif d'une métaheuristique pour rechercher le nouveau point de démarrage à chaque étape peut augmenter grandement le temps de réponse. La qualité de la solution finale dépend fortement de ce choix. Les méthodes Gloutonnes se distinguent quant à elle par leur rapidité pour proposer une solution. Le principal problème est qu'à chaque étape, la solution courante est complétée de la meilleure façon sans tenir compte des conséquences sur le coût de la solution finale, on dit que c'est une méthode aveugle. Ainsi, cette méthode n'est pas adaptée à notre problème où le choix d'un item est critique pour le choix des items qui suivent dans la combinaison. Les méthodes Tabou, recuit simulé et à seuil peuvent converger vers une bonne solution en un temps raisonnable pour une application quasi temps réel, ce qui est désiré dans le projet. Le principal inconvénient de la méthode Tabou est que si l'espace de recherche est trop large, elle peut s'enfermer dans un même voisinage, même si la liste tabou est grande. Pour les deux autres méthodes, leur inconvénient majeur est la nécessité de définir plus ou moins arbitrairement de nombreux paramètres pour obtenir une bonne, voire la meilleure solution finale. Notre choix s'est porté sur une méthode basée sur le recuit simulé pour l'application à fournir à l'entreprise.

Outre ces deux classes de métaheuristiques, la catégorie des algorithmes multi-objectifs a été abordée. Ces derniers ont été volontairement mis à part du fait qu'ils ne cherchent pas à optimiser une seule fonction objectif, mais plusieurs fonctions objectifs. Du fait de leur temps de calcul important, ils ne seront pas utilisés pour l'application finale qui doit fonctionner en temps réel. Néanmoins, ils nous semblent intéressants pour de

futurs travaux, car il apparaît que la fonction de pertinence définie dans la première partie de ce chapitre est une agrégation de deux fonctions objectif.

Dans le chapitre suivant, nous aborderons l'instanciation des différentes couches définies dans le chapitre 3 pour le projet de système de recommandation fournissant une combinaison d'offre touristique qui nous intéresse. Ainsi, il y sera notamment défini l'algorithme basé sur le recuit simulé utilisé pour l'application temps réel. En outre, un algorithme multi-objectif sera aussi proposé en séparant la fonction de pertinence en deux objectifs à optimiser. En effet, la contrainte de recommandation en temps réel est seulement posée par l'application demandée, mais une autre application plus tolérante temporellement utilisant cet algorithme apporterait des avantages supplémentaires, qui seront expliqués dans le chapitre suivant.

Chapitre 5

Application au domaine du tourisme

Résumé

Le chapitre 5 présente l'instanciation, au domaine de l'ADT Côte d'Or Tourisme, du système de recommandation générique que nous avons défini. Cette instanciation a pour but de fournir à un utilisateur donné une combinaison d'offres touristiques sous forme de séjour suivant la définition du problème défini dans le chapitre 4. Le chapitre 5 décrit l'implémentation des différentes couches de notre architecture de recommandation. Tout d'abord, il est présenté l'implémentation de la couche sémantique, incluant principalement le processus de création de l'ontologie de domaine basé sur la base de données de l'entreprise et la connaissance d'experts du tourisme. Ensuite, l'implémentation de la couche utilisateur est décrite, il y est notamment montré les différents paramètres non dépendants du domaine retenus pour la modélisation de l'utilisateur. Enfin, l'implémentation de la couche intelligence est expliquée en se focalisant sur la partie combinatoire, car elle est particulière à l'application touristique à réaliser, contrairement à la partie de pondération qui est définie de manière générique dans le chapitre 3. Trois algorithmes sont testés pour la génération de combinaisons: un algorithme basé sur le recuit simulé, l'algorithme Hill-Climbing et un algorithme multi-objectif.

Plan

1	Implémentation de la couche sémantique.....	98
1.1	Processus de création du modèle de domaine.....	98
1.2	Processus de création du modèle de buts.....	102
2	Implémentation de la couche utilisateur	102
3	Implémentation de la couche intelligence	103
3.1	Construction dynamique du pattern	103
3.2	Attribution des tolérances de dispersion	104
3.3	Algorithme de recherche combinatoire basé sur le recuit simulé	105
3.4	Benchmark, comparaison avec Hill-Climbing	108
3.5	Algorithme de recherche combinatoire basé sur un algorithme Multi objectif.....	109
4	Conclusion	113

Une architecture et le fonctionnement d'un système de recommandation générique basé sur la sémantique et la combinatoire ont été présentés dans le chapitre 3. Ensuite, dans le chapitre 4, un problème de recommandations touristiques a été défini et une étude sur des méthodes métaheuristiques candidates à la résolution de ce problème a été réalisée. Le chapitre 5 présente une application de cette architecture pour un système de recommandation touristique basé sur le problème d'optimisation touristique exposé dans le chapitre 4. Ce système de recommandation a été créé pour Côte-d'Or Tourisme, l'agence de développement touristique de la Côte-d'Or. La société collecte une très grande quantité d'information sur les offres touristiques par l'intermédiaire d'un site Web où les fournisseurs peuvent renseigner leurs offres. Ces informations sont réunies et centralisées dans une importante base de données. Elle regorge d'informations pertinentes et de connaissances sur le domaine touristique. Ensuite, par l'utilisation de cette base de données, les offres touristiques sont montrées sur un site Web de la même façon pour tous les clients. Elles sont rangées par catégories et affichées sous forme de listes triées (Figure 22).

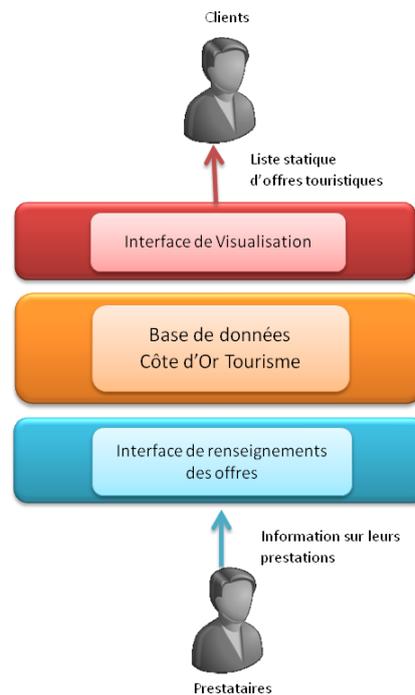


FIGURE 22. APERÇU DU SYSTEME ACTUEL

Pour fournir un contenu adapté à l'utilisateur, des processus doivent être insérés entre la base de données et l'interface de visualisation. La procédure de renseignements des offres touristiques par les prestataires n'est pas modifiée et les données sont toujours stockées dans la base de données. L'interface de visualisation doit être modifiée de sorte d'accepter la nouvelle forme de proposition des offres et doit permettre de récupérer les intérêts du client. Cette interface, qui est une application mobile, est présentée dans le chapitre 6. Un aperçu de la combinaison du système actuel avec l'architecture de système de recommandation proposé précédemment est montré dans la Figure 23.

Ce chapitre est organisé en 3 parties. La première partie explique l'implémentation de la couche sémantique constituée du modèle de domaine et du modèle de buts dans ce cadre touristique et en vue de l'application mobile. La deuxième partie explique l'implémentation de la couche utilisateur en fournissant un exemple de profil utilisateur. Enfin, la troisième partie détaille l'implémentation de la couche intelligence en abordant essentiellement la partie combinatoire, celle-ci n'étant pas générique d'une application à une autre, contrairement à la pondération.

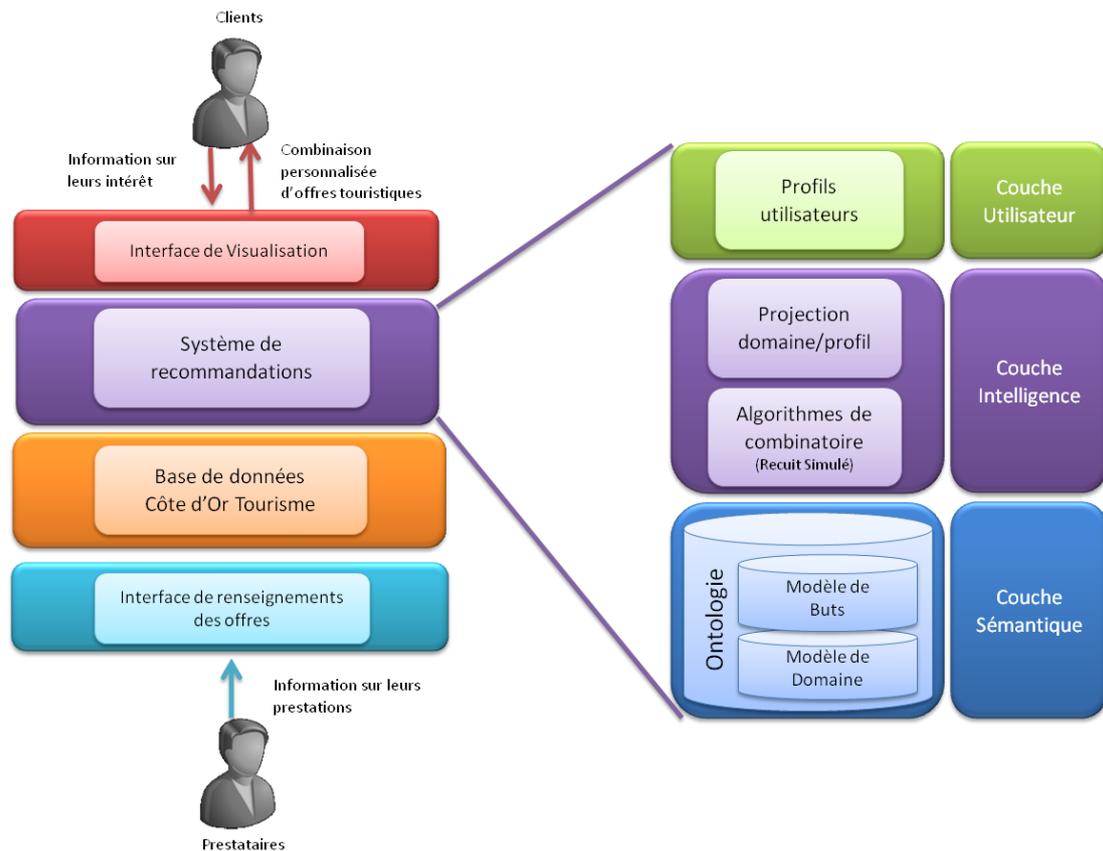


FIGURE 23. APERÇU DU SYSTEME AMELIORE

1 IMPLEMENTATION DE LA COUCHE SEMANTIQUE

La couche sémantique est la partie qui permet de modéliser le contenu, la connaissance selon le domaine et selon les besoins utilisateurs dans l'application. Elle est, d'après sa définition, constituée de deux sous-couches: le modèle de domaine et le modèle de buts. Le premier modèle est une ontologie de domaine peuplée d'individus. Le second est une ontologie de buts peuplée des individus du modèle de domaine par des règles métiers. Nous expliquerons dans une première partie, les processus de création du modèle de domaine à partir de la base de données de Côte-d'Or Tourisme et d'experts du domaine. Puis, une seconde partie, abordera la création du modèle de buts et des règles métiers.

1.1 PROCESSUS DE CREATION DU MODELE DE DOMAINE

Pour la construction de l'ontologie de domaine, nous procédons en quatre étapes. Tout d'abord, nous prenons la base de données comme point de départ, car elle contient la majeure partie de la connaissance et des offres touristiques. Nous sélectionnons alors tous les éléments pertinents de la base de données. Ensuite, la structure de l'ontologie de domaine est construite en prenant comme base cette sélection. Puis, l'ontologie est peuplée d'individus. Enfin, elle est enrichie à l'aide d'experts du domaine.

La première étape consiste à sélectionner, dans la base de données, les tables, attributs et tuples pertinents pour la modélisation du domaine. Pour cela, des vues sont construites sur la base de données à l'aide d'experts de la base et du domaine.

Une table descriptive est définie en parallèle, de sorte de décrire la manière dont doivent être utilisées ces vues pour la construction de l'ontologie. En effet, nous avons distingué deux types différents de vues:

- Les vues Data contenant les informations du domaine
- Les vues Relation mettant en relation les tuples des vues Data. On peut les assimiler à des tables association mettant en correspondance des identifiants de deux tables.

Le Tableau 1 présente un aperçu de la table descriptive. Elle décrit des vues Hotel et Restaurant en type Data et une vue Hotel_possede_restaurant en type Relation. Ensuite, les tableaux 2 et 3 montrent un aperçu de la structure et des tuples des vues Hotel et Restaurant. Enfin, le Tableau 4 montre un aperçu de la vue de type relation Hotel_possede_restaurant, on y voit une relation entre l’hôtel Hot01 et le restaurant Res01.

TABLEAU 1. APERÇU TABLE DESCRIPTIVE

Nom_vue	Type_vue	Vue1_relation	Vue2_relation
Hotel	Data	null	null
Restaurant	Data	null	null
Hotel_possede_restaurant	Relation	Hotel	Restaurant

TABLEAU 2. APERÇU DE LA VUE HOTEL, DE TYPE DATA

Identifiant	Nom	Nombre de chambres
Hot01	La Cloche	40
Hot02	Ibis	25

TABLEAU 3. APERÇU DE LA VUE RESTAURANT, DE TYPE DATA

Identifiant	Nom	Nombre de couverts
Res01	La Cloche	70
Res02	Le bon Pantagruel	32

TABLEAU 4. APERÇU DE LA VUE HOTEL_POSSEDE_RESTAURANT, DE TYPE « RELATION »

Identifiant_1	Identifiant_2
Hot01	Res01

La construction de cette structure a été réalisée manuellement, elle ne sera pas automatisée, car elle ne sera pas amenée à évoluer souvent.

La phase suivante consiste à construire la structure de base de l'ontologie à partir des différentes vues définies. Pour cela, un programme Java utilisant les bibliothèques JDBC et JENA a été créé. La première bibliothèque permet les connexions à la base de données et la deuxième permet la manipulation des composants d'une ontologie.

Le programme parcourt d'abord toute la table descriptive. En fonction du type des vues qui y ont été décrites, le programme va soit créer des concepts de l'ontologie, soit créer des relations entre concepts, associés respectivement aux vues Data et Relation. Cette phase est automatique.

En parcourant la table descriptive Tableau 1, le programme de traduction va générer la structure ontologique présentée dans la Figure 24.

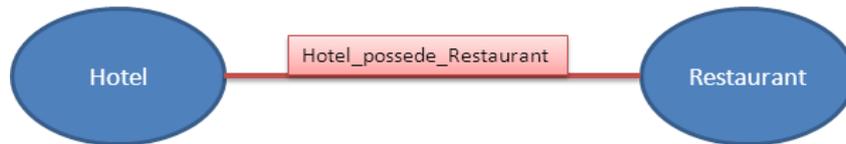


FIGURE 24. APERÇU D'UNE STRUCTURE ONTOLOGIQUE GENEREE PAR LE PROGRAMME DE TRADUCTION

Ensuite, l'étape suivante consiste à instancier concepts et relations de l'ontologie. Pour ce faire, chaque vue présente dans la table descriptive est parcourue par le programme. Pour chaque tuple d'une vue Data, des individus sont ajoutés dans le concept correspondant de l'ontologie. De même, pour chaque tuple d'une vue Relation, des instances de relations (appelées aussi assertions) sont ajoutées pour relier les individus de concepts indiqués par leur identifiant dans l'ontologie. C'est une étape automatique que l'on appelle « peuplement de l'ontologie ». La Figure 25 montre un aperçu d'ontologie peuplée à partir des tuples des vues exposés dans Tableau 2, Tableau 3 et Tableau 4.

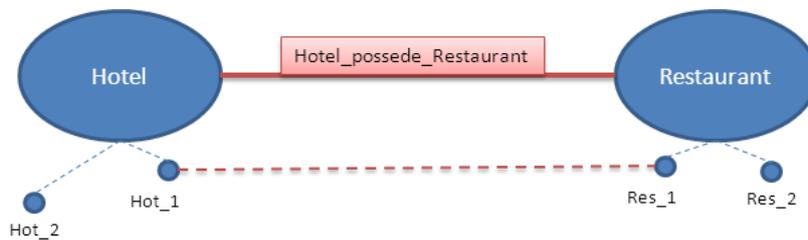


FIGURE 25. APERÇU D'UNE ONTOLOGIE APRES LA PHASE DE PEUPEMENT

Enfin, la dernière phase est appelée « enrichissement de l'ontologie ». Elle consiste en l'ajout de nouvelles connaissances à l'ontologie. Cela peut être l'ajout de concepts, de relations, de contraintes ou encore d'individus. Pour cette opération, le logiciel Protégé est utilisé. Il permet de manipuler simplement une ontologie. La Figure 26 montre un exemple d'enrichissement réalisé sur l'ontologie de domaine touristique. Le concept Hebergement est ajouté comme concept parent hiérarchiquement au concept Hotel. De plus, le concept Objet_touristique est ajouté comme concept parent des concepts Hebergement et Restaurant.

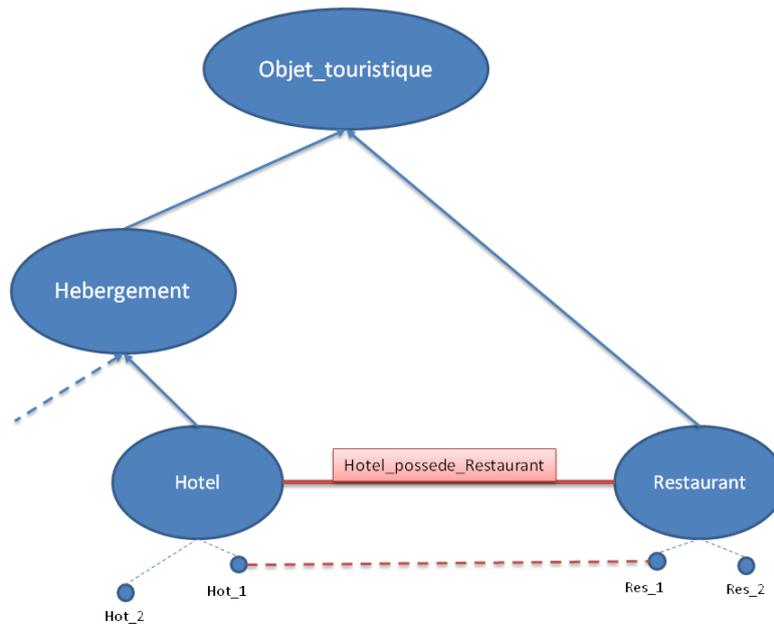


FIGURE 26. EXEMPLE D'ENRICHISSEMENT DE L'ONTOLOGIE TOURISTIQUE

Cette dernière étape manuelle n'est pas réalisée à chaque génération d'une nouvelle ontologie. En effet, le programme est conçu de manière à mettre à jour l'ontologie précédente de manière incrémentielle. Les ajouts manuels sont gardés d'une ontologie à une autre. Les nouveaux concepts/rerelations sont ajoutés lors de la mise à jour, et l'ontologie est repeuplée entièrement par parcours de toutes les vues. Cela évite les interventions manuelles pour la création d'une nouvelle ontologie lorsque sa structure n'a pas changé, ce qui sera généralement le cas. Ainsi, à la suite de ces différentes phases, une ontologie de domaine touristique est créée et enregistrée au format OWL. La Figure 27 récapitule tout le processus de création de l'ontologie de domaine décrit précédemment.

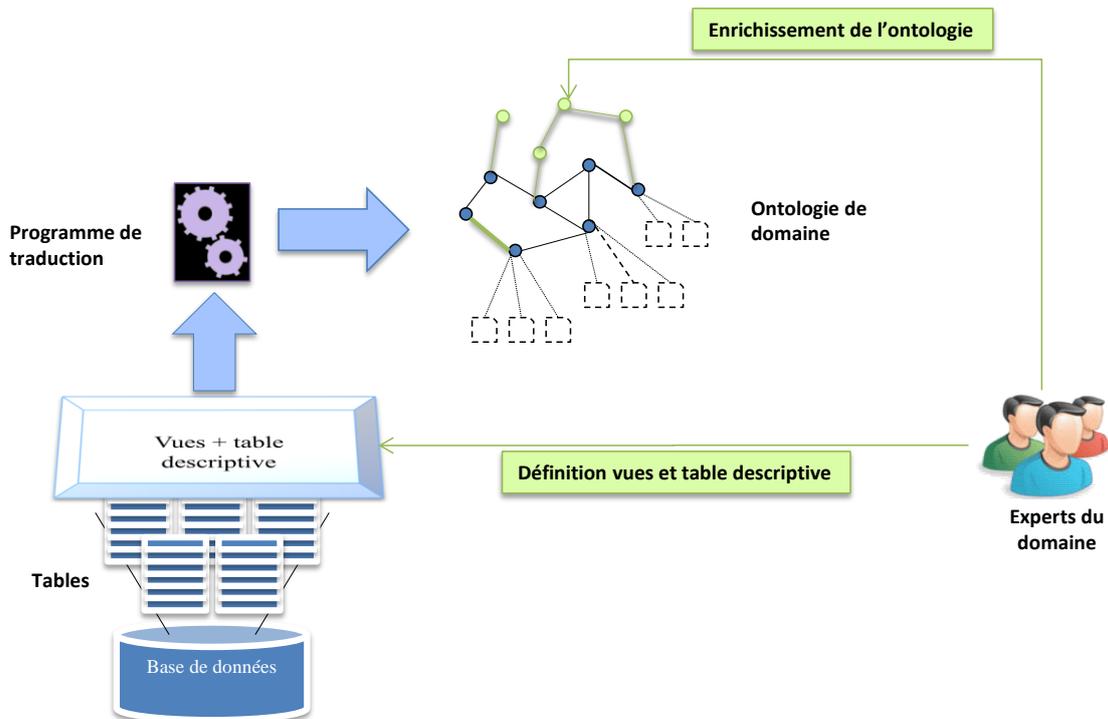


FIGURE 27. PROCESSUS DE CREATION DE L'ONTOLOGIE DE DOMAINE

1.2 PROCESSUS DE CREATION DU MODELE DE BUTS

Le modèle de buts est constitué, d'après sa définition, d'une ontologie de buts composée uniquement de concepts buts, et d'un ensemble de règles du premier ordre (sous forme de clauses de Horn) permettant le peuplement automatique de ces buts à partir des concepts du modèle de domaine.

La première étape consiste à identifier les buts (intérêts/besoins) des utilisateurs dans l'application touristique proposant des combinaisons d'offres touristiques. Pour ce faire, il est nécessaire de faire appel à des experts du domaine capables d'identifier ces différents éléments. Ainsi, nous avons mis en évidence plusieurs concepts dont: « Nature », « Vignoble », « Culturel », « Entre amis », « Handicapé »... Tous ces concepts ont pour concept parent « But » lequel possède un attribut poids qui est hérité à ses concepts enfants (voir Chapitre 3).

Ensuite, pour chaque concept identifié, des règles de premier ordre leur sont associées avec l'aide des experts. Ces règles sont écrites avec le langage SWRL. Elles permettent de réorganiser les individus proposables du domaine selon les buts possibles des utilisateurs dans l'application. Voici un extrait des règles associées au but « Nature » écrites en SWRL :

$$\text{Restaurant}(?x) \wedge \text{possede}(?x, \text{Terrasse}) \rightarrow \text{Nature}(?x) \wedge \text{poids_nature}(?x, 2)$$
$$\text{Objet_touristique}(?x) \wedge \text{zone_geographique}(?x, \text{"Morvan"}) \rightarrow \text{Nature}(?x) \wedge \text{poids_nature}(?x, 10)$$
$$\text{Randonnee_a_pied}(?x) \rightarrow \text{Nature}(?x) \wedge \text{poids_nature}(?x, 20)$$

La première règle peuple le but Nature des individus du concept Restaurant qui possèdent une terrasse et leur attribue un poids de 2 dans ce but. La deuxième règle peuple le but Nature des individus du concept objet_touristique ayant pour zone géographique le Morvan et leur attribue un poids de 10 dans ce but. La troisième règle peuple le but Nature des individus du concept Randonnee_a_pied et leur attribue un poids de 20 dans ce but.

Les individus qui sont considérés par plusieurs règles d'un même but ont leurs poids qui se somment. Ainsi, une randonnée à pied située dans le Morvan aurait un poids de $20 + 10 = 30$ dans le but Nature, suivant les règles précédentes. Toutes les règles définies sont inférées par le moteur d'inférence de l'ontologie. Cette étape de définition des buts et règles demande l'organisation de nombreuses réunions de travail afin d'identifier les buts, intérêts, besoins possibles des utilisateurs dans le système de recommandation touristique. La partie suivante aborde l'implémentation de la couche utilisateur qui permet la modélisation des utilisateurs afin que le système soit capable de leur fournir des propositions adaptées.

2 IMPLEMENTATION DE LA COUCHE UTILISATEUR

La couche utilisateur contient tous les profils utilisateurs représentant les différents utilisateurs au sein du système. Comme vu dans le chapitre 3, le profil est constitué de deux parties: une partie dite dépendante du domaine et une partie non-dépendante du domaine.

Dans la partie dépendante du domaine, sont présents les différents buts du modèle de buts qui correspondent à l'utilisateur courant. C'est un sous-ensemble des concepts buts du modèle de buts. Un exemple de partie dépendante du profil d'un utilisateur peut être :

$$\{\text{Famille, Nature, Culturel, Bien être}\}$$

où chaque terme désigne le nom d'un concept de l'ontologie de buts de la couche sémantique.

Dans la partie non dépendante du domaine, sont considérées des caractéristiques de l'utilisateur qui vont contraindre de manière plus ou moins forte les propositions. Ainsi, elle contient sous forme de paires attribut-valeur :

- la position géographique de l'utilisateur;
- la tolérance de l'utilisateur sur la dispersion géographique des offres;
- le nombre de jours du séjour;
- le type d'hébergement désiré si besoin ;
- la date du séjour.

Un exemple de partie non dépendante du profil d'un utilisateur peut être:

$$\left\{ \left(\begin{array}{l} \text{position}_{\text{latitude}} \\ 61.0541^\circ \end{array} \right), \left(\begin{array}{l} \text{position}_{\text{longitude}} \\ 58.1221^\circ \end{array} \right), \left(\begin{array}{l} \text{tolérance}_{\text{dispersion}} \\ 15 \text{ km} \end{array} \right), \left(\begin{array}{l} \text{nb}_{\text{jours}} \\ 2 \end{array} \right), \left(\begin{array}{l} \text{type}_{\text{hébergement}} \\ \text{Gîte} \end{array} \right), \left(\begin{array}{l} \text{date}_{\text{début}_{\text{séjour}}} \\ 15/07/2011 \end{array} \right) \right\}$$

Basée sur cette modélisation de l'utilisateur et la couche sémantique, la couche intelligence dont l'implémentation est expliquée dans la partie suivante peut fournir des recommandations à l'utilisateur.

3 IMPLEMENTATION DE LA COUCHE INTELLIGENCE

Dans la couche intelligence, les différents algorithmes utilisés pour proposer des recommandations aux utilisateurs sont implémentés. Dans le mécanisme de proposition d'une combinaison, deux phases se distinguent: la phase de projection et la phase de recherche combinatoire.

La phase de projection ne considère que la partie dépendante du domaine des profils utilisateurs. Pour chaque utilisateur, cette phase engendre un vecteur montrant ses intérêts sur chaque individu du modèle de domaine sous forme de poids (voir Chapitre 3 pour plus de détails). Elle est suivie par une phase de recherche combinatoire. Cette dernière consiste à rechercher une combinaison des individus du modèle de domaine pondérés, issus de la phase précédente. Cette section décrit la partie combinatoire, car celle-ci est particulière à l'application touristique à réaliser.

Pour cette phase combinatoire, plusieurs mécanismes sont à définir pour obtenir une recommandation touristique suivant la formalisation du problème posé dans le chapitre 4. Tout d'abord, il est nécessaire de définir le pattern et les sous-patterns de combinaison contraignant la combinaison d'items à fournir à l'utilisateur. Cela fait l'objet d'une première partie. Puis, la partie suivante explique l'attribution des tolérances de dispersion à ces patterns. Ensuite, l'algorithme de recherche combinatoire basé sur le recuit simulé est exposé, suivi de quelques benchmarks de comparaison de performances suivant la fonction de pertinence \mathfrak{R} (voir chapitre 4). L'algorithme basé sur le recuit simulé est comparé à un algorithme Hill-Climbing. Enfin, une dernière partie détaille l'utilisation d'un algorithme multi-objectif pour réaliser cette recherche combinatoire. Ce dernier algorithme n'est pas utilisé pour l'application mobile finale présentée dans le chapitre 6, mais peut être intéressant pour une future application ne demandant pas une réponse quasi temps réel.

3.1 CONSTRUCTION DYNAMIQUE DU PATTERN

Avant toute chose, la phase de recherche combinatoire détermine quel sera le pattern de combinaison à proposer. Pour cela, le pattern d'une combinaison est construit de base comme ceci :

$$P = \text{Activité}_0, \text{Restaurant}_1, \text{Activité}_2, \text{Activité}_3, \text{Restaurant}_4, \text{Hébergement}_5$$

De plus, des sous-patterns de combinaison sont définis (les indices permettent de situer les éléments du pattern utilisés par les sous-patterns):

$$p_1 = \text{Activité}_0, \text{Restaurant}_1, \text{Activité}_2, \text{Activité}_3, \text{Hébergement}_5$$

$$p_2 = \text{Restaurant}_4, \text{Hébergement}_5$$

Différentes informations issues de la partie non dépendante du domaine du profil utilisateur vont être ensuite considérées pour modifier ce pattern. Le pattern est construit dynamiquement.

En premier lieu, le type d'hébergement peut être modifié suivant le désir de l'utilisateur spécifié dans son profil. Par exemple, il peut être contraint au type Hôtel (concept fils du concept Hébergement dans l'ontologie de domaine), générant les patterns suivants :

$$P = \text{Activité}_0, \text{Restaurant}_1, \text{Activité}_2, \text{Activité}_3, \text{Restaurant}_4, \text{Hotel}_5$$

$$p_1 = \text{Activité}_0, \text{Restaurant}_1, \text{Activité}_2, \text{Activité}_3, \text{Hotel}_5$$

$$p_2 = \text{Restaurant}_4, \text{Hotel}_5$$

Puis, le nombre de jours du séjour va définir le nombre d'offres à proposer dans une combinaison. Si le nombre de jours est égal à 1, le pattern initial n'est pas changé. Sinon, les activités et restaurants vont être dupliqués pour chaque jour supplémentaire. Seul l'hébergement reste unique.

Ainsi, soit i , le nombre de jours du séjour, $i \in \mathbb{N}^*$. La taille du pattern est définie comme ceci:

$$\text{card}(P) = (i \times 5) + 1$$

Par exemple, si on considère un séjour de 3 jours, le pattern de la combinaison sera constitué de 16 objets et construit comme suit:

$$P = \text{Activité}_0, \text{Restaurant}_1, \text{Activité}_2, \text{Activité}_3, \text{Restaurant}_4, \\ \text{Activité}_5, \text{Restaurant}_6, \text{Activité}_7, \text{Activité}_8, \text{Restaurant}_9, \\ \text{Activité}_{10}, \text{Restaurant}_{11}, \text{Activité}_{12}, \text{Activité}_{13}, \text{Restaurant}_{14}, \text{Hotel}_{15}$$

Les sous-patterns de combinaison seront :

$$p_1 = \text{Activité}_0, \text{Restaurant}_1, \text{Activité}_2, \text{Activité}_3, \text{Hébergement}_{15}$$

$$p_2 = \text{Activité}_5, \text{Restaurant}_6, \text{Activité}_7, \text{Activité}_8, \text{Hébergement}_{15}$$

$$p_3 = \text{Activité}_{10}, \text{Restaurant}_{11}, \text{Activité}_{12}, \text{Activité}_{13}, \text{Hébergement}_{15}$$

$$p_4 = \text{Restaurant}_4, \text{Hébergement}_{15}$$

$$p_5 = \text{Restaurant}_9, \text{Hébergement}_{15}$$

$$p_6 = \text{Restaurant}_{14}, \text{Hébergement}_{15}$$

De ce fait, le nombre de jours et le type d'hébergement agissent sur la forme de la combinaison à fournir à l'utilisateur. À présent, il est nécessaire d'attribuer des tolérances de dispersion à ces patterns. La partie suivante explique le principe d'attribution des tolérances.

3.2 ATTRIBUTION DES TOLERANCES DE DISPERSION

Comme vu dans le chapitre 4, à chaque pattern et sous-patterns de combinaison doit être attribuée une tolérance de dispersion. Pour ce faire, nous utilisons la tolérance de l'utilisateur sur la dispersion des offres exprimée en Km dans son profil. Cette dernière est traduite en degré pour correspondre à l'expression de la longitude et latitude de chaque item. Puis, elle est attribuée à la tolérance de dispersion du pattern de combinaison. Pour les sous-patterns de combinaison, les tolérances de dispersion associées sont proportionnelles à la tolérance de dispersion du pattern de combinaison.

Soit un utilisateur dont la tolérance de dispersion traduite en degré vaut λ dans son profil utilisateur. En considérant les patterns de l'exemple précédent, la tolérance de dispersion pour le pattern de combinaison pour cet utilisateur est $tol(P) = \lambda$ et les tolérances de dispersion pour les sous-patterns sont :

$$tol(p_1) = tol(p_2) = tol(p_3) = \frac{90}{100} \lambda$$

$$tol(p_4) = tol(p_5) = tol(p_6) = \frac{10}{100} \lambda$$

La faible proportion $\frac{10}{100} \lambda$ appliquée à la tolérance de dispersion entre le deuxième restaurant de chaque jour et l'hébergement est justifiée par le fait qu'il est souhaitable que le système de recommandation propose un restaurant proche de l'hébergement pour la soirée. Cette notion de proximité est subjective, c'est pourquoi elle dépend de la tolérance de dispersion de l'utilisateur.

3.3 ALGORITHME DE RECHERCHE COMBINATOIRE BASE SUR LE RECUIT SIMULE

En se basant sur le pattern (le terme pattern générique englobe le pattern de combinaison et les sous-patterns) et les tolérances de dispersion, une métaheuristique peut alors se charger de rechercher la combinaison optimale d'items du domaine, par maximisation de la fonction de pertinence.

Nous utilisons un algorithme basé sur le recuit simulé pour chercher la combinaison optimale. Cet algorithme a la capacité de converger vers une bonne solution, ou la meilleure solution en un temps raisonnable pour une application quasi temps réel. Il a été prouvé dans (Geman & Geman, 1984) que le recuit simulé convergeait vers l'optimum global si le refroidissement était suffisamment lent (voir le Chapitre 4 pour plus de détail).

L'algorithme du recuit simulé possède de grandes similitudes avec l'algorithme Hill-Climbing. Voici un rappel de ce en quoi consiste ce dernier :

- On commence tout d'abord par trouver une solution au problème, une première combinaison, on calcule alors son énergie (inverse de la fonction de pertinence).
- Ensuite, on transforme légèrement la solution et on calcule la nouvelle énergie. Si cette dernière est inférieure à la précédente alors on prend cette nouvelle combinaison comme solution courante, sinon on garde l'ancienne.
- On continue ainsi jusqu'à ne plus pouvoir générer de nouvelle combinaison ayant une énergie inférieure à la combinaison courante.

Le principal problème de cette méthode réside dans le fait qu'il y a une grande probabilité pour tomber dans un minimum local, c'est-à-dire une combinaison qui, tout en n'ayant pas l'énergie minimale, sera telle qu'aucune légère transformation ne permettra d'obtenir une combinaison avec une meilleure énergie.

C'est à ce niveau que la technique du recuit simulé se distingue, elle permet en effet de sortir de certains minima locaux. La méthode consiste à introduire une erreur volontaire pour sortir de ces minima en acceptant, de temps en temps (avec une certaine probabilité), une combinaison de moins bonne qualité. Son acceptation sera déterminée aléatoirement en tenant compte de la différence entre les énergies et d'un paramètre appelé température. Ce dernier permet de prendre en compte le fait que plus le processus d'optimisation est avancé, moins il est prêt à accepter une solution plus coûteuse. L'algorithme fonctionne comme suit. Au départ, il choisit une combinaison initiale aléatoire d'items suivant le pattern défini précédemment. Cette combinaison a une énergie E_0 , appelée énergie initiale.

L'énergie $\mathcal{E}(C_k)$ d'une combinaison valide C_k est l'inverse de la fonction de pertinence $\mathfrak{R}(C_k)$ appliquée à cette combinaison:

$$\mathcal{E}(C_k) = \frac{1}{\mathfrak{R}(C_k)}, \text{ avec } 1 \leq k \leq \text{NbC}$$

Il s'agit alors de minimiser l'énergie. Plus l'énergie est basse, meilleure est la combinaison. Une variable T , appelée température, décroît par palier au cours du temps. A chaque niveau de température, un certain nombre de modifications aléatoires sont testées sur la combinaison courante. Lorsque ce nombre de modifications est atteint, on dit que le système est à l'état d'équilibre. Un coût d_f est associé à chaque modification; il est défini comme la différence entre l'énergie de la combinaison après et avant la modification. Un coût négatif signifie que l'énergie courante a une énergie plus petite que la précédente (qui est donc meilleure par définition), elle est alors gardée. Au contraire, un coût positif représente une « mauvaise » modification. Néanmoins, elle peut être gardée à une certaine probabilité (taux d'acceptation t_a) qui dépend du niveau de température courante et du coût. Plus la température est grande, plus la probabilité l'est aussi. Ainsi, au cours du temps, le nombre de changements autorisés diminue dû à la décroissance de la température, jusqu'à ce qu'il n'y ait plus aucune modification acceptable. Au final, on dit que le système est gelé, et la combinaison courante devient la combinaison finale à présenter à l'utilisateur. Le taux d'acceptation est défini comme ceci:

$$t_a = e^{-\frac{d_f}{T_h}}$$

où T_h est la température au niveau h , $h \in \mathbb{N}$. Ce choix de l'exponentiel pour le taux d'acceptation s'appelle règle de Metropolis.

Cet algorithme est décrit par le diagramme fonctionnel de la Figure 28.

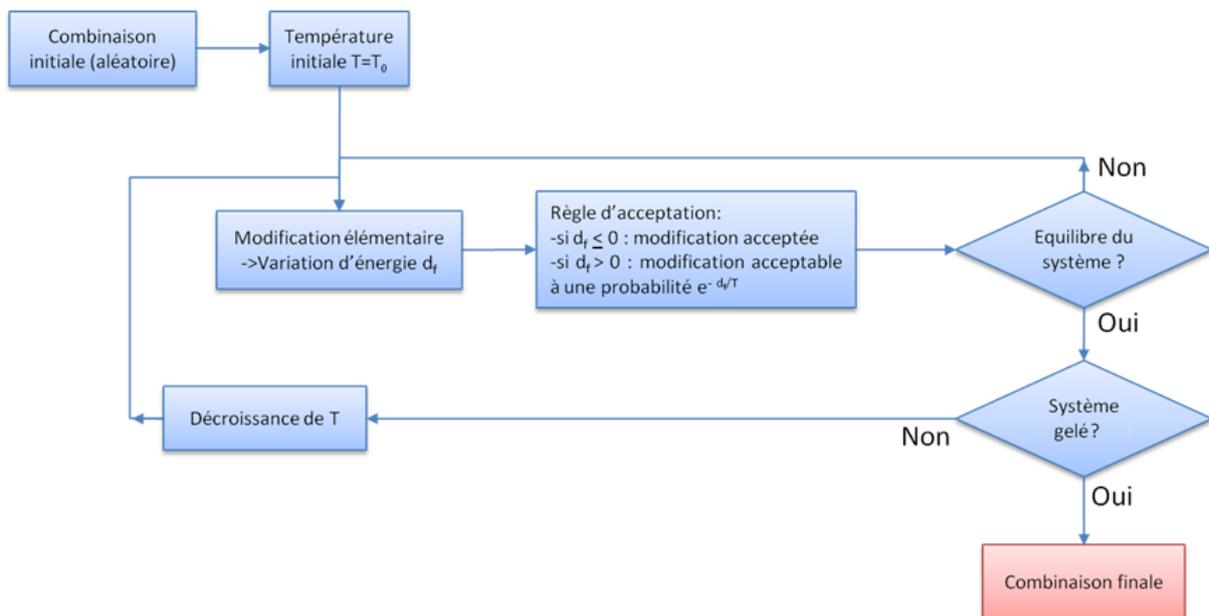


FIGURE 28. DIAGRAMME FONCTIONNEL DE L'ALGORITHME DU RECUIT SIMULE UTILISE

La Figure 29 montre un exemple de variation de l'énergie de la combinaison courante au cours du temps. Au début, l'amplitude des variations de l'énergie est grande, car de nombreuses solutions sont acceptées étant donné la température du système. Puis, cette amplitude diminue au fil du temps, dû au refroidissement du

système (baisse de la température). Enfin, lorsqu'il n'y a plus de variation de cette énergie, le système est gelé et la meilleure solution trouvée (qui est généralement celle pour laquelle le système est gelé) est renvoyée.

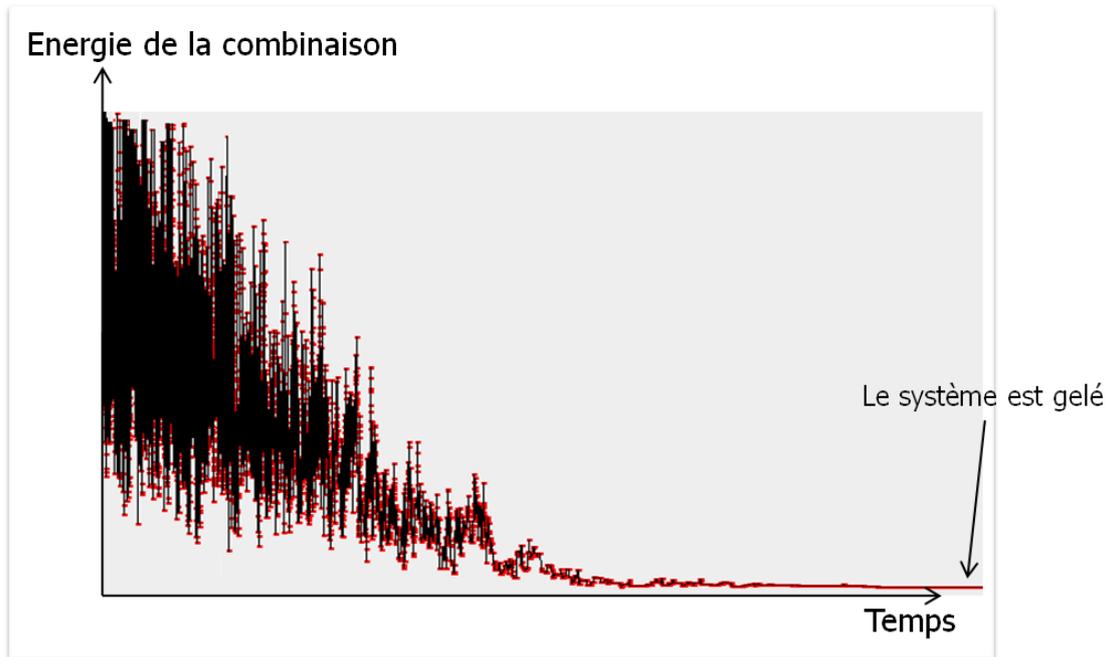


FIGURE 29. EXEMPLE DE VARIATION DE L'ENERGIE DE LA COMBINAISON COURANTE PAR ALGORITHME DE RECUIT SIMULE

Une décroissance géométrique de la température à chaque palier a été choisie :

$$T_h = g(T_{h-1}) = coef \times T_{h-1} = coef^h \times T_0$$

Si nous voulions privilégier la qualité de la solution finale, nous lui donnerions une décroissance logarithmique, en sacrifiant le temps d'exécution. Néanmoins, étant donné l'application temps réel que l'on souhaite en avoir et du fait qu'une « bonne » solution qui n'est pas l'optimum est tout à fait acceptable, cette décroissance exponentielle est pertinente.

Il est possible de jouer sur de nombreux paramètres du recuit simulé pour modifier la pertinence des solutions et le temps de réponse.

Tout d'abord, il faut déterminer la température initiale T_0 qui va décroître au cours du temps. Cette température peut être choisie empiriquement, mais la difficulté est que si elle est trop grande à l'initialisation, toutes les modifications au départ seront acceptées quel que soit le coût associé, ce qui est une perte de temps non négligeable. Si elle est trop faible au départ, le parcours de l'espace des solutions sera trop limité. Il faut trouver le juste milieu.

Une façon de trouver cette valeur consiste à générer un certain nombre de transformations coûteuses et à calculer la variation moyenne (coût moyen) $d_{f_{moy}}$. Un taux d'acceptation t_a à respecter au départ est choisi,

nous le fixons à 0.9. Ainsi, la valeur de T_0 est : $T_0 = \frac{d_{f_{moy}}}{\ln \frac{1}{t_a}}$ (car $t_a = e^{-\frac{d_f}{T_h}}$).

Ensuite, il convient de définir le coefficient de décroissance de la température entre les paliers et le nombre d'itérations des modifications par palier. Le coefficient de décroissance est compris entre 0 et 1. Plus il est grand, plus la décroissance est lente et plus l'algorithme met du temps à converger vers une solution. En contrepartie, cela lui permet de ne pas se bloquer trop tôt dans un minimum local. Une valeur de 0.6 a été choisie suite à de nombreux tests, il permet d'obtenir de bons résultats en un temps raisonnable.

Le nombre d'itérations par palier définit le nombre de modifications réalisées par palier de température. Lorsque ce nombre est atteint, on dit que le système est en équilibre statistique. C'est alors que le changement de palier (décroissance de température) se fait. Ce nombre d'itérations est fixé à 2000.

Enfin, le critère d'arrêt de l'algorithme est important. On dit que le système est gelé lorsque plus aucune modification n'est acceptable. En pratique, certains arrêtent l'algorithme lorsqu'un certain seuil de température est atteint ou lorsque le nombre de paliers maximaux de température a été dépassé. Une autre méthode consiste à arrêter l'algorithme lorsqu'aucun changement n'a été autorisé sur un nombre donné d'itérations. C'est cette dernière méthode que nous utilisons, elle garantit quasiment que nous nous trouvons dans un minimum (local ou global). Le nombre maximal d'itérations autorisé sans que cela ne génère de modification avant de stopper l'algorithme est égal au nombre maximal d'itérations par palier défini précédemment.

La définition arbitraire de certains paramètres du recuit simulé est vue comme le principal inconvénient de cet algorithme. Nous proposons une méthode pour définir ces paramètres sans tâtonnement dans le chapitre 7 (travaux futurs).

De plus, un autre inconvénient est qu'il est possible (même si c'est très rare) que l'algorithme rencontre une meilleure solution que la solution finale durant sa recherche. Néanmoins, c'est cette dernière qui est normalement renvoyée dans l'algorithme originel. Pour parer à ce problème, dans notre système, la meilleure solution rencontrée au cours du temps est gardée en mémoire.

Pour avoir un ordre d'idée des résultats fournis par cet algorithme, une comparaison avec l'algorithme du Hill-Climbing est réalisée dans la partie suivante.

3.4 BENCHMARK, COMPARAISON AVEC HILL-CLIMBING

Afin de justifier l'utilisation d'une métaheuristique plus complexe qu'une méthode de descente comme le Hill-Climbing, quelques benchmarks ont été réalisés sur des données réelles et aléatoires pour comparer les deux méthodes.

Dans le jeu de données réelles, 4960 items sont présents, composés de 3714 activités, 1008 restaurants et 238 hébergements.

Dans le jeu de données aléatoires, 30000 items sont présents, composés de 10000 activités, 10000 restaurants et 10000 hébergements. Les coordonnées des items sont définies aléatoirement dans le rectangle englobant la Côte-d'Or. Les poids des items sont aussi définis aléatoirement et compris entre 0 et 1000.

Le pattern et les sous-patterns sont construits comme ceci :

$$P = \text{Activité}_0, \text{Restaurant}_1, \text{Activité}_2, \text{Activité}_3, \text{Restaurant}_4, \text{Hébergement}_5$$

$$p_1 = \text{Activité}_0, \text{Restaurant}_1, \text{Activité}_2, \text{Activité}_3, \text{Hébergement}_5$$

$$p_2 = \text{Restaurant}_4, \text{Hébergement}_5$$

Et les tolérances de dispersion associées, pour un utilisateur donné, sont :

$$\text{tol}(P) = 0.1$$

$$\text{tol}(p_1) = 0.09$$

$$\text{tol}(p_2) = 0.01$$

Le Tableau 5 compare les résultats obtenus par l'algorithme du recuit simulé avec l'algorithme Hill-Climbing. Il présente l'énergie moyenne et les temps moyens obtenus sur les données aléatoires et les données réelles. Ces moyennes sont réalisées sur 100 itérations des algorithmes.

Tableau 5. Recuit simulé VS Hill-Climbing sur des jeux de données réels et aléatoires

	Jeux de données aléatoires		Jeux de données réelles	
	Energie moyenne× 10 ⁶	Temps moyen (ms)	Energie moyenne× 10 ⁶	Temps moyen (ms)
Recuit simulé	64.13	404	65.29	211
Hill-Climbing	106.16	7	93.24	6
Meilleure énergie trouvée × 10⁶	62.70		65.25	

Notre contrainte de génération de proposition quasi temps réel pour l'application mobile correspond à une tolérance fixée arbitrairement à 500 ms. Ainsi, étant donné des temps de calcul inférieurs à ce seuil pour les deux méthodes, les deux sont acceptables. Cependant, le tableau comparatif montre que l'énergie moyenne des combinaisons trouvées par le recuit simulé est meilleure (plus petite) que celle trouvée par le Hill-Climbing. Cette différence de pertinence se retrouve lors des propositions à l'utilisateur. De ce fait, l'utilisation de l'algorithme du recuit simulé est justifiée pour l'application mobile.

Malgré ce choix, si la contrainte de temps était réduite, l'utilisation d'une méthode multi-objectif pourrait aussi s'avérer intéressante. Même si cette méthode ne sera pas utilisée pour l'application mobile, elle est définie dans la partie suivante afin d'envisager des applications futures.

3.5 ALGORITHME DE RECHERCHE COMBINATOIRE BASE SUR UN ALGORITHME MULTI OBJECTIF

Dans le chapitre 4, nous avons défini une fonction de pertinence, donnant la pertinence d'une combinaison C_k pour un utilisateur donné:

$$\mathfrak{R}(C_k) = W_{C_k} + \frac{PoidsMaxItem}{\sigma_{mod}(C_k) + \sum_{l=0}^{S-1} \sigma_{mod}(C_{k,l})}$$

où W_{C_k} est le poids de la combinaison, $\sigma_{mod}(C_k)$ et $\sigma_{mod}(C_{k,l})$ la dispersion modérée de la combinaison et de ses sous-combinaisons selon le pattern et les sous-patterns et $PoidsMaxItem$ le poids maximum que peut avoir un item (voir chapitre D pour plus de détails).

Cette fonction est une somme pondérée de la forme $f(C_k) = r_1 O_1(C_k) + r_2 O_2(C_k)$, où $r_1 = 1$ et $r_2 = PoidsMaxItem$, et $O_1(C_k) = W_{C_k}$ et $O_2(C_k) = \frac{1}{\sigma_{mod}(C_k) + \sum_{l=0}^{S-1} \sigma_{mod}(C_{k,l})}$. Pour ce problème, nous désirons maximiser O_1 et O_2 .

Le choix des valeurs de r_1 et r_2 est arbitraire. Il détermine l'influence d'un des deux objectifs (O_1 et O_2) par rapport à l'autre dans $f(C_k)$. Étant donné les différences d'échelle qu'il y a entre O_1 et O_2 , nous avons choisi de les mettre à un niveau équivalent d'influence en donnant à r_2 la valeur $PoidsMaxItem$. Cependant, les valeurs choisies ne sont peut être pas pertinentes pour tout utilisateur, peut-être qu'un des deux objectifs doit-être mis plus en avant que l'autre ou que ce choix doit-être laissé à l'appréciation de l'utilisateur final.

Dans ce contexte, l'idée de l'optimisation multi-objectif est de considérer, non pas une fonction à optimiser, mais plusieurs objectifs (fonctions) à optimiser.

Ainsi, pour résoudre notre problème touristique, nous allons chercher à maximiser O_1 et O_2 indépendamment. La différenciation des solutions ne se fait alors plus sur un simple classement par valeur de pertinence. En effet, si on considère deux combinaisons X_1 et X_2 dont les valeurs respectives pour les objectifs O_1 et O_2 sont (20, 50) et (35, 40), il n'est pas possible de dire laquelle des deux est préférable par rapport à l'autre, car X_1 est meilleure sur l'objectif O_1 et X_2 est meilleure sur l'objectif O_2 .

Le concept de domination de Pareto est alors défini. On dit qu'une solution X_1 domine une solution X_2 si elle est au moins aussi bonne que X_2 sur tous les objectifs et meilleure sur au moins un objectif.

Dans l'espace des solutions possibles, toutes celles qui ne sont dominées par aucune autre font partie de ce qu'on appelle le front de Pareto. Le but de l'optimisation multi-objectif est de rechercher ce front de Pareto. Il n'y a non plus une seule bonne solution au problème, mais un ensemble de bonnes solutions possibles.

Ainsi, outre l'utilisation de l'algorithme du recuit simulé expliqué précédemment, nous avons utilisé un algorithme multi-objectif pour répondre au problème touristique.

L'algorithme utilisé se base sur les principes d'un algorithme évolutionnaire ($\mu + \lambda$). Dans un premier temps, une population est générée aléatoirement. Une population est constituée de ($\mu + \lambda$) individus. Chaque individu est une combinaison réalisable d'items, et chaque item est appelé gène. Puis une sélection par tournois des individus (solutions) est effectuée, μ individus sont sélectionnés. Les individus alors sélectionnés sont croisés et engendrent λ enfants sur lesquels des mutations peuvent se produire à une probabilité donnée. Enfin, une nouvelle population est considérée, elle contient les μ individus parents sélectionnés et les λ individus enfants; les autres individus de la population précédente sont supprimés. L'algorithme se répète jusqu'à ce qu'un nombre de générations donné soit atteint.

Pour la sélection par tournoi, il faut pouvoir discriminer les individus les uns par rapport aux autres. Le rang de Pareto est alors introduit. Il permet de donner à chaque individu un rang définissant son éloignement par rapport au front de Pareto de la population courante. Les individus du front de Pareto ont un rang de 1. Si on enlève ce front et que l'on recherche le nouveau front de Pareto des individus restant, nous obtenons un nouvel ensemble d'individus auxquels on attribue le rang de 2. Et ainsi de suite.

L'algorithme utilisé pour calculer le front de Pareto d'une population G est le suivant:

ALGORITHME 11. CALCUL DU FRONT DE PARETO

-
- 1: $G \leftarrow \{G_1, \dots, G_m\}$ Ensemble d'individus sur lequel calculer le front
 - 2: $O \leftarrow \{O_1, \dots, O_n\}$ Les objectifs utilisés pour l'évaluation
 - 3: $F \leftarrow \{ \}$ ▷ Le front
 - 4: **pour** chaque individu $G_i \in G$ **faire**
 - 5: $F \leftarrow F \cup \{G_i\}$
 - 6: **pour** chaque individu $F_j \in F$ différent de G_i **faire**
 - 7: **si** F_j Pareto Domine G_i suivant O **alors**
 - 8: $F \leftarrow F - \{G_i\}$

-
- 9: **stopper** la boucle **pour**
- 10: **sinon si** G_i Pareto Domine F_j suivant **Oalors**
- 11: $F \leftarrow F - \{F_j\}$
- 12: **renvoyer** Best

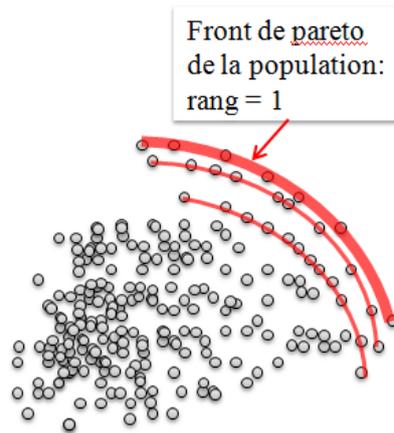


FIGURE 30. ILLUSTRATION DU FRONT DE PARETO

Pour engendrer une répartition plus équilibrée de la population sur le front, le concept de rareté (sparsity) d'un individu est également utilisé. On dit qu'un individu est dans une région éparsée si les individus les plus proches dans son rang sont assez éloignés. La rareté d'un individu peut être calculée à l'aide de la distance Manhattan sur chaque objectif, entre le voisin gauche et le voisin droit le long de son rang de Pareto. Les individus situés aux extrémités reçoivent une valeur de rareté infinie.

Nous basons alors la sélection par tournoi sur la rareté des individus lorsque la sélection se fait sur deux individus d'un même rang, et sur le rang de Pareto si la sélection se fait sur deux individus de rangs différents.

ALGORITHME 12. ALGORITHME DE SELECTION PAR TOURNOI PAR TRI LEXICOGRAPHIQUE NON-DOMINE ET UTILISATION DE LA RARETE

-
- 1: $P \leftarrow$ Population avec des rangs de Pareto assignés
- 2: Best \leftarrow individu pris de manière aléatoire dans P
- 3: $t \leftarrow$ taille du tournoi, $t \geq 1$
- 4: **pour** i allant de 2 à t **faire**
- 5: Next \leftarrow individu pris de manière aléatoire dans P
- 6: **si** RangPareto(Next) < RangPareto(Best) **alors** \triangleright Un plus petit rang est meilleur
- 7: Best \leftarrow Next
- 9: **sinon si** RangPareto(Next) = RangPareto(Best) **alors**
- 10: **si** Rareté(Next) > Rareté(Best) **alors** \triangleright Le plus rare est meilleur

11: Best ← Next

12: **renvoyer** Best

Une fois les μ individus sélectionnés, $\frac{\lambda}{2}$ croisements aléatoires entre ces individus sont effectués. Chaque croisement génère deux individus recevant un mélange des gènes de leurs parents. Pour savoir quels gènes doivent être échangés lors d'un croisement, nous leur attribuons une probabilité de 0.5.

Par exemple, soit les individus (Hotel48, Restaurant10, Activité24) et (Hotel36, Restaurant58, Activité51) à croiser. Après calcul des probabilités sur chaque gène pour connaître les positions des gènes à échanger dans les combinaisons, les positions 2 et 3 sont retenues. Cela engendre alors deux individus enfants dont les gènes sont : (Hotel48, Restaurant58, Activité51) et (Hotel36, Restaurant10, Activité58).

La phase suivante appelée mutation consiste à introduire des anomalies génétiques au sein des enfants générés. En effet, une probabilité de mutation de 0.01 est définie pour chaque gène des enfants. Si mutation il y a, un gène est pris aléatoirement dans l'ensemble des gènes possibles (items du modèle de domaine) à cette position.

La phase de croisement est une phase d'intensification alors que la phase de mutation est une phase de diversification.

La nouvelle génération est alors constituée des μ individus sélectionnés et des λ enfants issus des croisements. À chaque génération, les phases de sélection, croisements et mutations sont effectuées faisant tendre la population vers le front de Pareto de l'espace possible des solutions.

Nous arrêtons cet algorithme lorsque ψ générations ont été engendrées. On obtient alors un nombre N d'individus qui tendent vers le front de Pareto ou qui appartiennent à ce dernier. Ces individus sont les combinaisons possiblement proposées à l'utilisateur par une application.

ALGORITHME 13. ALGORITHME MULTI-OBJECTIF BASE SUR UN ALGORITHME EVOLUTIONNAIRE ($\mu + \lambda$)

1: $P \leftarrow \{P_1, \dots, P_m\}$ ▷ Population initiale de $m = \mu + \lambda$

2: EvaluerFitness(P) ▷ Calcul des valeurs dans les objectifs

3: $R \leftarrow \langle \dots \rangle$ Rangs de Pareto de P

4: **pour** chaque rang $R_i \in R$ **faire**

5: Calculer Rareté des individus dans R_i

6: BestFront ← front de Pareto de P

7: **répéter**

8: $Q \leftarrow \{ \}$

9: **pour** $\lambda/2$ fois **faire** ▷ Reproduction: sélection + croisement + mutation

10: Parent $P_a \leftarrow$ SelectionParTournoi(P) ▷ algorithme 12, taille tournoi: 2

11: Parent $P_b \leftarrow$ SelectionParTournoi(P)

12: Enfants $C_a, C_b \leftarrow$ Croisement(Copie(P_a), Copie(P_b))

```

13:      Q ← Q ∪ {Mutation(Ca), Mutation(Cb)}
14:  EvaluerFitness(Q)
15:  Q ← Q ∪ P
16:  P = {}
17:  R ← Calculer Rangs de Pareto de Q
18:  BestFront ← front de Pareto de Q
19:  pour chaque rang Ri ∈ R faire
20:      Calculer Rareté des individus dans Ri
21:      si ||P|| + ||Ri|| ≥ m alors
22:          P ← P ∪ les m – ||P|| individus les plus rares de Ri
23:          stopper la boucle pour
24:      sinon
25:          P ← P ∪ Ri
26: jusqu'à un nombre de générations ψ est atteint
27: renvoyer BestFront

```

L'utilisation de cet algorithme multi-objectif a l'avantage de fournir une multitude de bonnes solutions par rapport aux méthodes à état simple comme le recuit simulé. L'utilisateur pourrait alors se voir proposer toutes les combinaisons du front de Pareto trouvé et ce serait à lui de décider laquelle il préfère, en favorisant soit un écart-type faible des coordonnées des offres, soit un grand poids (et les intermédiaires bien sûr). Cela donnerait des possibilités supplémentaires certaines. Cependant, comme évoqué plus haut, le temps de calcul de ce front de Pareto est important (de l'ordre de 5 secondes pour une combinaison d'une journée) et n'est pas envisagé pour l'application mobile développée. Néanmoins, cette métaheuristique a un grand potentiel pour une application plus tolérante temporellement.

4 CONCLUSION

Ce chapitre a montré l'instanciation au domaine de l'ADT Côte d'Or Tourisme, du système de recommandation générique que nous avons défini. Cette instanciation a pour but de fournir à un utilisateur donné une combinaison d'offres touristiques sous forme de séjour suivant la définition du problème défini dans le chapitre 4. Dans ce chapitre, nous avons décrit l'implémentation des différentes couches de notre architecture de recommandation.

Tout d'abord, il a été présenté l'implémentation de la couche sémantique, incluant principalement le processus de création de l'ontologie de domaine basé sur la base de données de l'entreprise et la connaissance d'experts du tourisme. Ensuite, l'implémentation de la couche utilisateur a été décrite, il y a été notamment montré les différents paramètres non dépendants du domaine retenus pour la modélisation de l'utilisateur. Enfin, l'implémentation de la couche intelligence a été expliquée en se focalisant sur la partie combinatoire, car elle est particulière à l'application touristique à réaliser, contrairement à la partie de pondération qui est définie de manière générique dans le chapitre 3. Trois algorithmes ont été testés pour la génération de combinaisons: un

algorithme basé sur le recuit simulé, l'algorithme Hill-Climbing et un algorithme multi-objectif. Les deux premiers ont été comparés et il en ressort que le recuit simulé est supérieur, en terme de qualité de la solution finale. Son temps de génération est plus long que celui du Hill-Climbing mais cette différence est largement négligeable selon notre tolérance de temps pour l'application mobile. Ainsi, l'utilisation de l'algorithme basé sur le recuit simulé est justifiée. L'algorithme multi-objectif n'est pas comparable avec les autres métaheuristiques testées, car il travaille sur une population de solutions engendrant des temps de réponse plus longs. De plus, il décompose la fonction de pertinence en deux objectifs à optimiser indépendamment, ce qui fait qu'il y a une multitude de solutions optimales (les solutions qui appartiennent au front de Pareto).

Le développement d'une application mobile fournissant une combinaison d'items touristiques adaptés à l'utilisateur est le but industriel final de ce projet. Cette application est expliquée dans le chapitre suivant. C'est une interface utilisateur communiquant avec le système de recommandation défini dans ce chapitre. Néanmoins, c'est cette application qui a orienté le modèle de buts, les patterns et le choix de l'algorithme d'optimisation. Outre la démonstration de l'application mobile réalisée, le chapitre suivant expose la méthode utilisée pour calibrer et ajuster les propositions. De plus, la partie technique de la réalisation du serveur et de l'application est présentée.

Chapitre 6

Développement industriel

Résumé

Ce chapitre présente le développement industriel des travaux de recherche présentés dans ce mémoire. Il décrit la modélisation de l'offre touristique dans la base de données, l'infrastructure d'un point de vue technique et le développement de l'application mobile. Cette dernière est découpée en 3 phases principales : une phase de définition du profil de l'utilisateur, une phase de génération et visualisation de la combinaison d'items et des items indépendamment, et enfin, une phase de modification totale ou partielle des offres si besoin. Dans cette partie consacrée aux applications mobiles, la structure technique de l'application iPhone est abordée. Une dernière partie présente un outil de supervision qui permet à l'ADT Côte d'Or Tourisme d'affiner la pertinence des résultats et/ou d'orienter les résultats selon ses besoins.

Plan

1	L'offre touristique.....	117
2	Infrastructure	119
3	Application Mobile/Interface utilisateur	120
3.1	Fonctionnalités	120
3.1.1	Définition du profil utilisateur	120
3.1.2	Génération de la combinaison et visualisation	122
3.1.3	Régénération d'une combinaison et fonctionnement de la géolocalisation	124
3.2	Aspect technique de l'application iPhone	125
4	Calibration/ajustement	125
5	Conclusion	126

Le chapitre précédent a présenté une implémentation de l'architecture proposée dans le chapitre 3 afin de fournir des recommandations optimisées d'offres touristiques aux utilisateurs. La mise en place de ce système de recommandation a demandé un temps important consacré au développement de l'infrastructure pour son intégration. De plus, une application mobile faisant office d'interface utilisateur pour dialoguer avec le système de recommandation a été réalisée.

Dans un premier temps, pour comprendre pourquoi la construction de l'ontologie se base principalement sur la base de données, ce chapitre aborde dans une première partie la structuration de l'offre touristique centralisée par l'entreprise dans la base de données. Puis, une deuxième partie expose l'infrastructure d'un point de vue technique, il y est notamment abordé les langages, les structures, les modes de communication utilisés par le serveur hébergeant le système de recommandation. Cette partie a pour objectif de présenter comment notre proposition s'est insérée dans le système d'information existant à l'ADT Côte d'Or Tourisme. Ensuite, les développements de l'application mobile sont présentés, en montrant plus particulièrement ses fonctionnalités et sa structuration technique. Enfin, un outil de supervision et des méthodes ont été développés afin d'ajuster et calibrer les résultats de recommandations. Ils sont expliqués dans la dernière partie.

1 L'OFFRE TOURISTIQUE

L'offre touristique peut être définie comme l'ensemble des prestations pouvant être consommées par des touristes. Afin de donner une image fidèle de l'offre touristique, il convient tout d'abord de définir les concepts liés aux objets touristiques. Tout d'abord, certaines propriétés sont communes à tous les objets : identification, représentation spatiale, moyens de communication, informations millésimées, périodes d'ouverture, contacts humains. D'autres sont en revanche spécifiques à certains types d'objets ; les équipements et les tarifs. Les différents types d'objets touristiques sont : Hôtels, Hébergements pour jeunes et scolaires, Visites pour jeunes et scolaires, Activités de loisirs, Randonnées pédestres, Hébergements collectifs, Parcs et jardins, Loueurs de bateaux, Marchés, Routes touristiques, Fermes auberges, Salles de séminaires, Forfaits touristiques, Comités du tourisme réceptif et agences réceptives, Animations, Campings, Confréries, Chambres d'hôtes, Grands vins, Artisans, Galeries d'art, Circuits, Chefs, Randonnées à vélo, Offices de tourisme, Etapes à vélo, Meublés de tourisme classés, Séjours clés en main, Restaurants, Produits du terroir, Fêtes et manifestations, Sites et monuments historiques, Musées, Evènements.

Un objet touristique est généralement d'un seul type, mais peut cependant en posséder plusieurs. Par exemple, un hôtel-restaurant est un objet de type restaurant et de type hôtel. Un objet touristique peut également être plusieurs fois du même type. Ainsi, un hôtel peut posséder plusieurs restaurants ou plusieurs salles de séminaire. Ensuite, des relations hiérarchiques peuvent exister entre objets. Par exemple, une randonnée à vélo est composée d'étapes à vélo. Ces étapes peuvent être consommées indépendamment de la randonnée et ont donc une existence propre dans le système en tant qu'objets. De même, un évènement est composé de plusieurs fêtes ou manifestations. Enfin, de nouveaux types d'objets touristiques sont régulièrement ajoutés en fonction des travaux portant sur la structuration de l'offre touristique.

Un objet touristique est commercialement identifié par son nom, il possède une adresse, un code postal, et est relié à une localité ou à une zone géographique plus large. Un descriptif commercial détaillé de l'objet touristique complète l'identification. Le protocole d'identification informatique utilisé est celui utilisé par la norme Tourinfrance. C'est un identifiant décomposé de la manière suivante : soit l'identifiant **FMABOU0210002331**, les caractères 1 à 3 désignent le type d'objet touristique, ici FMA pour fête et manifestations. Les caractères 3 à 6 donnent le code de la région, ici Bourgogne. Les caractères 7 à 9 indiquent le code du département, ici Côte-d'Or. Les caractères 10 à 11 donnent le code du propriétaire de la fiche, ici 00 pour Côte-d'Or Tourisme, site centralisateur des objets touristiques pour le département de la Côte-d'Or. Enfin,

les caractères 12 à 16 sont un numéro lié à l'ordre de création des objets, ici c'est le 2331^{ème} objet créé par Côte-d'Or Tourisme.

En fonction de son ou de ses types, un objet touristique est caractérisé par les données spécifiques ayant une dépendance fonctionnelle directe avec l'identifiant d'objet. Les objets de type « hôtels » possèdent notamment des informations relatives à leur capacité d'accueil, en nombre de chambres, tandis que les campings ont un nombre d'emplacements pour tentes, un nombre de chalets, un nombre de mobile homes, etc.

Un des critères principaux dans le choix d'une prestation touristique réside dans la caractérisation de celui-ci par rapport à des référentiels d'équipements ou de services assimilés. Des listes d'équipements, de services, d'activités proposées, de thématiques de visites ou encore de modes de paiement doivent donc être mises en relation avec les objets touristiques.

Exemple d'équipements des hôtels : Ascenseur, bar, bibliothèque, boulodrome, boutiques, borne interactive, borne internet, cabine téléphonique/point phone, discothèque, garage privé, jardin, jeux pour enfants, local matériel fermé, minigolf, minitel, parc, parking privé, piano-bar, piscine, piscine couverte, restaurant, salon, salle de sport, salon de télévision, sauna, hammam, jacuzzi, squash, salle de remise en forme, tennis, terrasse.

Un objet touristique peut posséder plusieurs équipements ou services de ce type. Un équipement ou service peut être présent dans plusieurs objets touristiques différents. De plus, pour chaque prestation, il est indiqué un tarif minimum, un tarif maximum ainsi que les conditions d'application du tarif.

La Figure 31 montre une vision d'ensemble de ce qu'est l'offre touristique dans la base de données de Côte-d'Or Tourisme.

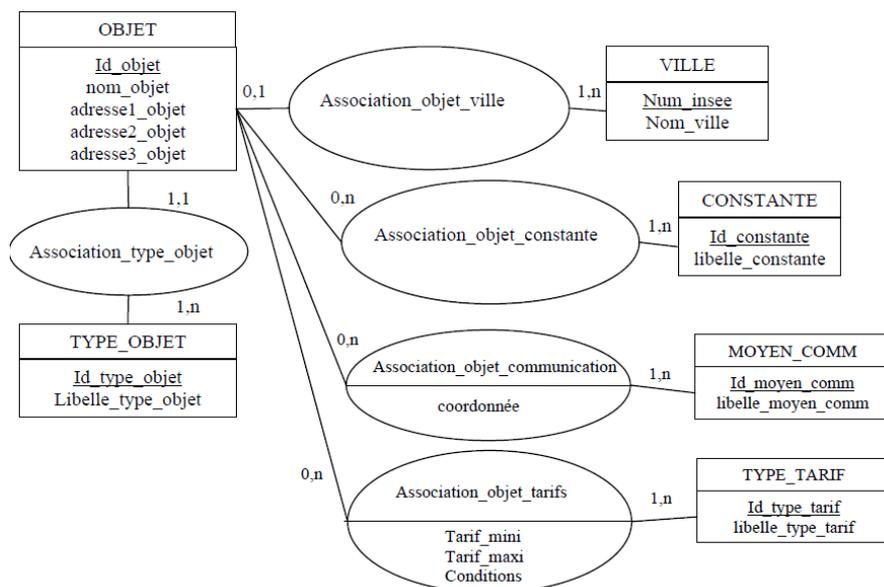


FIGURE 31. MODELE CONCEPTUEL DE DONNEES DES OBJETS TOURISTIQUES

Cette définition complexe des objets touristiques dans la base de données explique pourquoi le processus de construction de l'ontologie de domaine de la couche sémantique montrée dans le chapitre 5 est basé en grande partie sur celle-ci. Pour des raisons de performances, notamment dues à la taille importante de l'ontologie lorsque toutes les informations relatives aux objets touristiques y sont présentes, seulement certaines d'entre elles ont été sélectionnées pour la création de l'ontologie de domaine. Celles qui sont sélectionnées par des vues sont celles utilisées dans les règles métiers du modèle de buts.

Comme l'a montré la Figure 23, le système de recommandation touristique se greffe à l'environnement actuel de Côte-d'Or Tourisme. Son introduction nécessite néanmoins la création d'une nouvelle interface pour recueillir des informations d'utilisateurs afin de construire leur profil et pour leur fournir des recommandations touristiques. Avant d'aborder les spécificités techniques et fonctionnalités de l'interface, la partie suivante s'attarde sur les différents langages, les structures et les modes de communication entre les différents composants (interface, serveurs, base de données) pour la mise en place complète du système.

2 INFRASTRUCTURE

Cette partie présente l'ensemble de l'infrastructure d'un point de vue technique en abordant le langage et la structure du serveur, le stockage des données et les modes de communication.

La Figure 32 montre un aperçu des interactions entre serveur, bases de données et interfaces. Le serveur (au centre) contient l'implémentation du système de recommandation exposé dans le chapitre 5. Une interface est utilisée pour récolter des informations sur l'utilisateur. Ces informations sont transmises au serveur afin que le système de recommandation construise le profil de l'utilisateur. Ensuite, l'ontologie est utilisée par le système de recommandation pour générer des recommandations suivant ce profil. Elles sont affichées sur l'interface de l'utilisateur. Les informations descriptives des offres sont récupérées directement dans la base de données, n'étant pas toutes présentes dans l'ontologie.

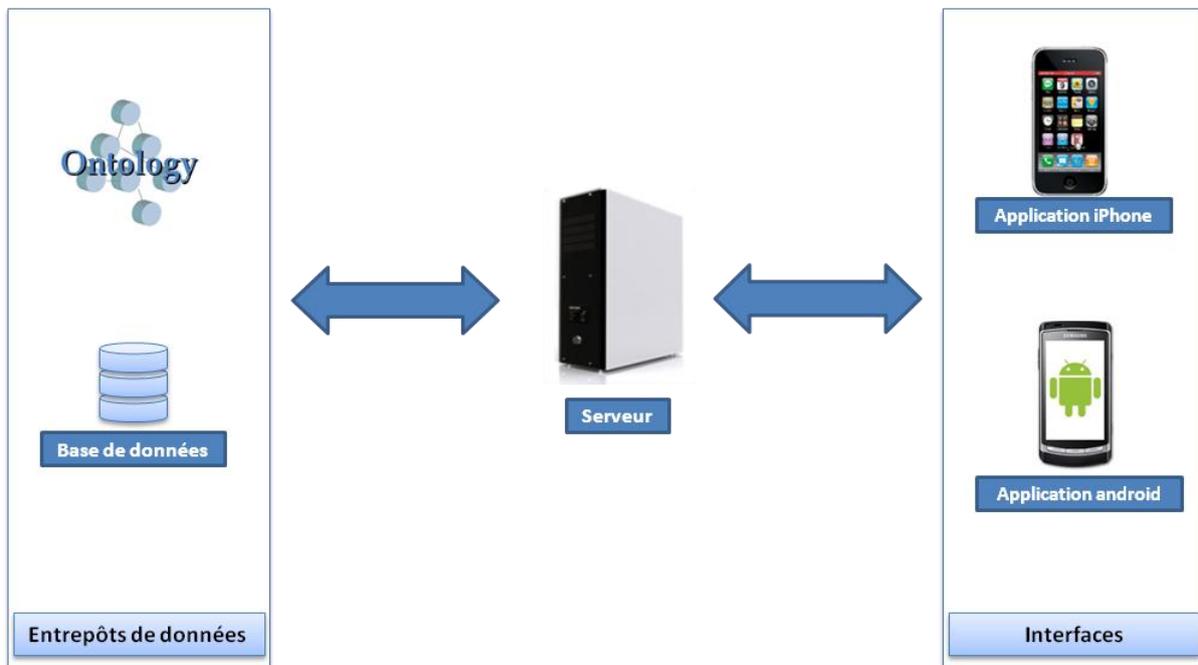


FIGURE 32. APERÇU DES INTERACTIONS ENTRE SERVEUR, ENTREPOTS DE DONNEES ET INTERFACES

Afin de pouvoir accéder en tous lieux et à tous moments aux données touristiques, le système de recommandation est hébergé sur un serveur OVH distant capable de supporter la montée en charge. Le langage utilisé pour coder cette implémentation de l'architecture est le Java. Le langage Java s'est imposé, en raison de la bibliothèque Jena, existante uniquement dans ce langage. Cette bibliothèque qui permet de manipuler des ontologies est la plus répandue et la plus documentée.

De plus, un des intérêts de Java est sa facilité d'intégration au Web. Apache Tomcat a été utilisé pour réaliser des servlets Java. Ces servlets réagissent comme une page PHP, c'est-à-dire que le code est interprété par le serveur à la demande du client.

Dans le code du serveur, l'implémentation de l'architecture et la gestion des données ont été séparés. Il y a ainsi deux packages différents :

- Un package contenant l'architecture en couches du système de recommandation.
- Un package contenant tout ce qui concerne l'application et la communication avec celle-ci.

La plupart des classes gérant des données sont des singletons, permettant d'éviter de dupliquer en mémoire des informations réutilisables.

Dans la Figure 32, on peut voir deux types de stockage de données. Le premier est une base de données stockant les informations propres aux applications mobiles, telles que les données utilisateurs, les séjours enregistrés, les offres favorites, etc. De plus, dans cette base de données sont présentes les données utilisées par Côte-d'Or Tourisme pour définir l'offre touristique expliquée dans la partie 1 de ce chapitre. Le système de gestion de base de données SQLServer est utilisé pour cette base de données.

Le second type de stockage de données, même si ce terme est un peu réducteur, est une ontologie. Cette ontologie est chargée en mémoire par le système de recommandation à son démarrage. Comme définie précédemment, l'ontologie modélise le contenu suivant la connaissance générale du domaine et suivant la connaissance spécifique aux contraintes possibles des utilisateurs dans les applications (voir chapitre 3). Cette ontologie est décrite avec le langage OWL, basé sur la logique de description. C'est un standard pour la description d'ontologie.

En ce qui concerne les méthodes de communication, dans un but de flexibilité et de compatibilité, le langage XML couplé aux requêtes HTTP pour la communication entre les interfaces (applications iPhone et Android) et le serveur est utilisé. Le serveur reçoit ainsi une requête HTTP de type GET, et renvoie un fichier XML.

La partie suivante décrit les applications mobiles en expliquant leurs fonctionnalités et les aspects techniques de l'application iPhone.

3 APPLICATION MOBILE/INTERFACE UTILISATEUR

L'introduction du système de recommandation demande la création d'une nouvelle interface pour dialoguer avec le serveur. Elle doit recueillir des informations d'utilisateurs afin de permettre la construction de leur profil et doit leur fournir les recommandations touristiques.

Pour cela, une application mobile a été créée se chargeant d'être l'interface avec les utilisateurs. Nous avons développé l'application mobile pour iOS (système d'exploitation des iPhones). Puis, une version sous le système d'exploitation Android proche de la version iOS a été réalisée par une société externe.

Cette partie présente dans un premier temps les fonctionnalités de l'application mobile. Nous considérons les deux versions de l'application comme une seule application, du fait qu'elles sont identiques en termes de fonctionnalités. Seul l'aspect esthétique diffère quelque peu. Puis, il sera abordé le côté technique du développement de l'application iPhone. Nous n'entrerons néanmoins pas dans le détail du code

3.1 FONCTIONNALITES

L'utilisation de l'application mobile peut être découpée en 3 phases principales : une première phase de définition du profil de l'utilisateur, une phase de visualisation de la combinaison d'items et des items indépendamment, et enfin, une phase de modification totale ou partielle des offres si besoin.

3.1.1 DEFINITION DU PROFIL UTILISATEUR

Dans cette application, l'accent n'a pas été mis sur le processus de profilage de l'utilisateur. Il n'y a pas de profilage implicite de l'utilisateur par analyse comportementale, tout se fait de manière explicite en lui demandant ses désirs et contraintes pour la proposition touristique.

Dans une première étape, il est demandé à l'utilisateur de définir la partie dynamique de son profil utilisateur. Pour ce faire, les différents buts du modèle de buts lui sont proposés dans un carrousel sous forme d'icônes qu'il peut sélectionner (Figure 33).



FIGURE 33. ECRAN DU CARROUSEL DE BUTS

Ensuite, l'utilisateur peut définir la partie statique de son profil. Cela débute par le nombre de jours du séjour désirés. Par défaut, ce nombre est fixé à 1 jour et est limité à 3 jours maximum. Il est aussi possible, mais pas obligatoire, de définir une date d'arrivée, ce qui permet la proposition d'offres événementielles se déroulant durant le séjour (Figure 34).



FIGURE 34. ECRAN DE CONTRAINTE TEMPORELLE

Un écran suivant (voir Figure 35) permet de spécifier que l'on désire ou non un hébergement, dont il est possible de spécifier son type (Hôtel, Camping, Chambre d'hôtes, Gîte, ...). Par défaut, aucun hébergement

n'est proposé, car, d'après les experts touristiques, la plupart des visiteurs de Côte-d'Or sont hébergés par des connaissances (amis, familles, ...).



FIGURE 35. ECRAN DE CHOIX DU TYPE D'HEBERGEMENT

Enfin, un dernier écran (Figure 36) propose à l'utilisateur de définir une position géographique de sorte d'orienter la combinaison d'offres vers la localisation spécifiée. L'adresse peut être entrée manuellement ou le téléphone mobile de l'utilisateur peut être géolocalisé. Par défaut, aucune position géographique n'est présente et la combinaison d'offres peut se situer n'importe où en Côte-d'Or. Outre cette position géographique, l'utilisateur peut aussi définir sa tolérance de dispersion des offres au sein d'une combinaison. Cette tolérance est exprimée en kilomètres, elle est par défaut fixée à 5Km, mais peut aussi prendre les valeurs suivantes: 15 Km, 50 Km, Toute la Côte-d'Or.



FIGURE 36. ECRAN DE CONTRAINTE GEOGRAPHIQUE

3.1.2 GENERATION DE LA COMBINAISON ET VISUALISATION

Après la définition du profil utilisateur, la génération de la combinaison d'offres touristiques peut être actionnée par l'utilisateur.

Ce dernier se voit alors proposer un ensemble d'offres affichées suivant un pattern construit dynamiquement à l'aide du profil de l'étape précédente. Ainsi, six offres touristiques sont proposées par jour du séjour, auxquelles s'ajoute un hébergement si demandé

Les items du séjour peuvent être visualisés de plusieurs façons. Ils sont tout d'abord proposés sous forme d'une liste triée par jour du séjour dans laquelle figure le nom des prestations touristiques et leur lieu (Figure 37). Il est aussi possible de les visualiser sur une carte géographique pour chaque jour du séjour (Figure 38). De plus, une visualisation en réalité augmentée des destinations touristiques est possible (Figure 39).



FIGURE 37. VISUALISATION DU SEJOUR PAR LISTE

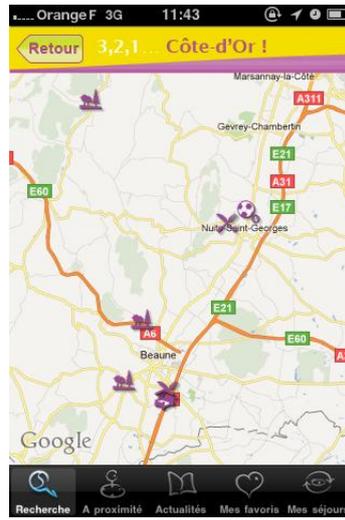


FIGURE 38. VISUALISATION DU SEJOUR SUR UNE CARTE

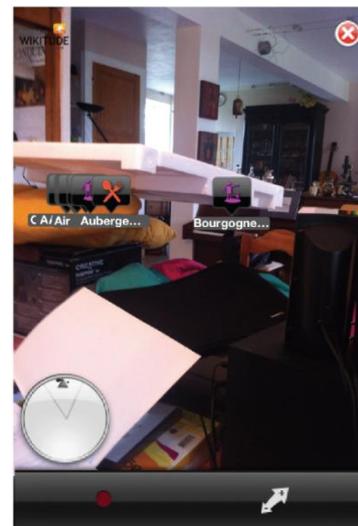


FIGURE 39. VISUALISATION DU SEJOUR EN REALITE AUGMENTEE

Outre cet aperçu global des items par jour, ils peuvent être visualisés en détail de manière indépendante (Figure 40).



FIGURE 40. VISUALISATION D'UN ITEM TOURISTIQUE

La combinaison ou seulement certains éléments de cette dernière peuvent ne pas convenir à l'utilisateur. Il a alors la possibilité de faire une régénération partielle ou totale de la combinaison.

3.1.3 REGENERATION D'UNE COMBINAISON ET FONCTIONNEMENT DE LA GEOLOCALISATION

Lorsqu'il se voit proposer une combinaison, l'utilisateur a la possibilité de régénérer totalement la combinaison. Cela consiste à réaliser de nouveau une recherche combinatoire à partir du profil de l'utilisateur. Cependant, les offres de la précédente combinaison ne sont plus considérées pour cette nouvelle recherche. Elles sont placées dans un tableau d'offres, dit tableau tabou qui contient toutes les prestations non désirées par l'utilisateur lors de la recherche combinatoire.

Outre cette régénération totale, l'utilisateur a la possibilité de ne régénérer que partiellement la combinaison en sélectionnant les offres qu'il ne souhaite pas garder pour la suivante. Pour la recherche combinatoire, cela consiste à fixer certaines offres dans la combinaison et à ne réaliser ce processus que sur les éléments non figés de celle-ci. Ainsi, étant donné que la fonction de pertinence d'une combinaison se base en partie sur les coordonnées géographiques des offres, la combinaison nouvellement proposée sera nécessairement contrainte géographiquement par les offres fixées. Les prestations non gardées de la précédente combinaison ne peuvent être proposées dans la nouvelle combinaison, car elles sont ajoutées à la liste tabou des offres à ne pas considérer. La Figure 41 montre l'action de régénération totale du séjour.

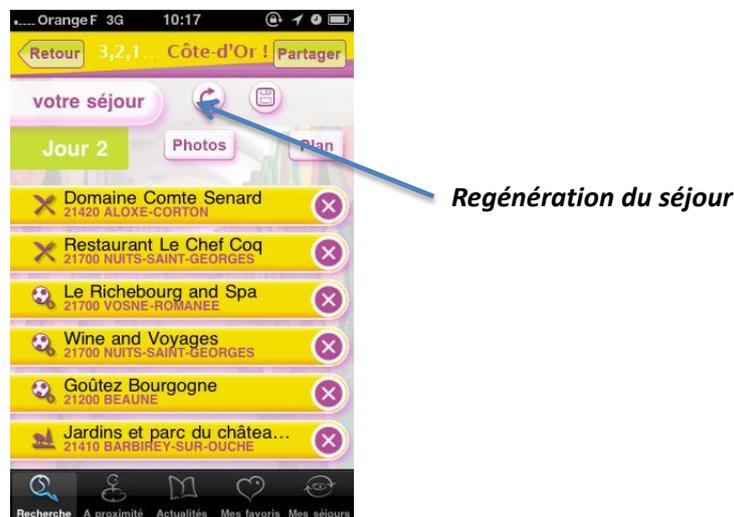


FIGURE 41. ACTION DE REGENERATION TOTALE DU SEJOUR

Nous avons abordé la possibilité pour l'utilisateur de spécifier une localisation géographique, de sorte de lui proposer une combinaison d'offres proche de cette position. Ce processus est un cas particulier de régénération partielle d'une combinaison. En effet, afin de rechercher une combinaison d'offres proches d'une certaine position géographique, un nouvel item est ajouté à la combinaison. Le poids de cet item vaut 1 et ses coordonnées sont celles de la position géographique spécifiée par l'utilisateur. Alors, un nouvel élément de type « Géoloc » est ajouté au pattern de combinaison. Par exemple, pour un séjour de 1 jour avec un hébergement de tout type et une géolocalisation, le pattern sera ainsi:

(Activité, Restaurant, Activité, Activité, Restaurant, Hébergement, Géoloc)

La recherche combinatoire s'effectue sur tous les éléments de la combinaison sauf sur le dernier qui reste fixe. Du fait de l'impossibilité de changer le dernier élément de la combinaison et la dépendance géographique des offres les unes par rapport aux autres par la fonction de pertinence, la combinaison finale sera logiquement orientée vers la position géographique de l'utilisateur.

La partie suivante s'attarde sur l'aspect technique du développement de l'application sans toutefois entrer dans les détails du code.

3.2 ASPECT TECHNIQUE DE L'APPLICATION IPHONE

L'application iPhone est développée sur Mac OS X 10.6 à l'aide de Xcode et InterfaceBuilder version 3.1.4. Xcode est muni d'un émulateur iPhone qui permet de tester les applications iPhone directement sur le Mac. Notre application utilise l'API Cocoa qui permet de développer des applications Mac et, dans notre cas, des applications iPhone. Il est basé sur le langage Objective-C, un langage orienté objet basé, comme le C++ sur le C ANSI.

Pour rendre l'application plus modulable, elle a été séparée en différentes couches : la couche de données, la couche métier, les managers, les éléments graphiques, et l'interface graphique.

Tout au long de son déroulement, l'application communique avec le serveur qui, à l'aide de Tomcat 6.20, permet la génération de documents XML grâce à des servlets Java. C'est la couche de données qui va effectuer la tâche de communication avec le serveur. L'application envoie par l'intermédiaire de la couche de données une requête HTTP/GET à ce serveur qui générera un document XML en fonction de cette requête. La couche de données est ensuite capable de parser la réponse.

La couche métier est l'intermédiaire entre la couche de données et l'interface graphique. Elle comporte toutes les classes du domaine. La couche métier utilise la couche de données afin de récupérer les données nécessaires à l'application et également afin de récupérer les données de l'utilisateur pour les transmettre au serveur.

Les managers sont les classes qui gèrent les différents éléments de l'interface. Il y a quatre managers. Le premier manager gère les vues de l'application, le deuxième gère les alertes, le troisième se charge des images, et le dernier gère le design. Ces classes sont des singletons qui vont permettre d'accéder à tous les éléments de l'application depuis n'importe quelle interface.

Afin d'avoir une application ergonomique, il a été nécessaire de définir nos propres éléments d'interface. Cette couche regroupe ces différents éléments. Deux types de carrousels ont notamment été réalisés. Le premier permet de présenter l'ensemble des critères utilisés pour définir l'utilisateur et le second permet d'afficher les items sous forme de diaporama d'images. Un pop-up personnalisable a aussi été réalisé.

L'interface graphique est la dernière couche de l'application, elle utilise toutes les couches précédemment citées. Elle comporte une classe pour chaque vue de l'application.

4 CALIBRATION/AJUSTEMENT

Après son implémentation, le système de recommandation peut nécessiter quelques ajustements pour fournir des propositions plus pertinentes d'après les experts du tourisme. De plus, l'entreprise peut vouloir mettre en avant certaines offres. C'est pourquoi un outil de supervision a été développé et des méthodologies ont été définies pour ajuster les propositions faites par le système.

En fonction d'un profil type d'utilisateur choisi, l'outil de supervision permet d'afficher la liste des items triés par poids d'intérêt selon ce profil. Il est possible d'afficher les règles qui entrent en jeu dans la pondération de chaque item.

Ainsi, les experts peuvent identifier les items qui ont un poids trop important ou trop faible pour ces profils types. Dans ce cas, cela peut indiquer plusieurs défaillances:

- des règles sont mal définies, avec des poids non pertinents et doivent être modifiées;
- de nouvelles règles doivent être introduites;
- de nouveaux buts doivent être introduits.

Identifier les défaillances est un travail que seuls les experts sont capables de réaliser, de par leurs connaissances approfondies sur les possibles buts des utilisateurs dans l'application, les règles logiques qui peuvent être associées à ces buts et les pertinences des items dans les buts.

Ainsi, les administrateurs peuvent modifier les règles, buts, et poids afin d'améliorer la pondération des items, ce qui est fondamental pour de bonnes recommandations de combinaisons.

La Figure 42 montre un exemple d'utilisation de l'outil de supervision pour visualiser les items « Restaurant » et leur poids associé dans les différents buts. Ils sont triés par buts et par poids. Dans l'exemple, la somme des poids (colonne « total ») des restaurants dans le but « Entre amis » défini par le code « amis » est affichée.

Type de stats:	<input type="text" value="tourisme KM"/>	<input type="button" value="Selectionner type"/>
Libelle stats:	<input type="text" value="synthèse restauration"/>	<input type="button" value="Lancer la recherche"/>

Nombre de résultats obtenus :3442

id	societe	codepostal	localite	code	total	pays_touristique
RESBOU0210000018	La Table de l'Hostellerie	21140	SEMUR-EN-AUXOIS	amis	19	PAYS D'AUXOIS
RESBOU0210000027	Restaurant de la Poste	21440	SAINT-SEINE-L'ABBAYE	amis	19	PAYS SEINE ET TILLES
RESBOU0210000052	Restaurant Le Chef Coq	21700	NUITS-SAINT-GEORGES	amis	19	COTE/HAUTES COTES DE NUITS
RESBOU0210000149	A ARCHIVER - Le Chevreuil	21190	MEURSAULT	amis	19	COTE/HAUTES COTES DE BEAUNE
RESBOU0210000169	Hostellerie de Levernois	21200	LEVERNOIS	amis	19	COTE/HAUTES COTES DE BEAUNE
RESBOU0210000215	Hostellerie du Vieux Moulin	21420	BOULLAND	amis	19	COTE/HAUTES COTES DE BEAUNE
RESBOU0210000228	Les Oenophiles	21000	DIJON	amis	19	DIJON ET SA REGION
RESBOU0210000233	Les Jardins de La Cloche	21000	DIJON	amis	19	DIJON ET SA REGION
RESBOU0210000241	Le Fleury	21200	BEAUNE	amis	19	COTE/HAUTES COTES DE BEAUNE
RESBOU0210000265	Le Pré aux Clercs JP et Alexis Billoux	21000	DIJON	amis	19	DIJON ET SA REGION
RESBOU0210000552	Le Grill du Castel	21320	CHATEAUNEUF EN AUXOIS	amis	19	PAYS D'AUXOIS
RESBOU0210000565	Le Renaton	21200	BEAUNE	amis	19	COTE/HAUTES COTES

FIGURE 42. OUTIL DE VISUALISATION DES ITEMS DE TYPE RESTAURATION

De manière extrême, la société peut vouloir promouvoir certains items par rapport à d'autres dans certains buts. Ces items sont appelés « Les incontournables du but » et doivent avoir un poids plus important que les autres items non incontournables dans le but donné. Alors, une modification manuelle directe du poids de l'item dans ce but est possible. Dans ce cas, les règles et leur poids associés pour cet item dans ce but ne sont plus pris en compte pour la pondération et c'est ce poids défini manuellement qui est considéré.

Une telle modification fait que, lorsqu'un utilisateur choisira ce but pour constituer son profil, l'item dont le poids a été augmenté aura une plus grande probabilité de sortie dans la combinaison d'items.

Ces ajustements a posteriori permettent d'affiner la pertinence des résultats selon les experts, ou, tout du moins, d'orienter les résultats selon le désir de l'entreprise.

5 CONCLUSION

Le développement industriel des travaux de recherches s'est déroulé sur 18 mois. Le premier élément développé a été l'ontologie et le chargement de l'ontologie avec les données de la base de données de l'ADT Côte d'Or Tourisme. La complexité de la modélisation des objets touristiques dans la base de données a

fortement influencé la construction de l'ontologie. La structure de l'ontologie ne devait pas être trop éloignée de la modélisation de l'offre touristique pour que l'administrateur puisse la comprendre et la faire évoluer. Ensuite, nous avons développé l'imbrication de l'architecture de recommandation avec le système d'information touristique (langage, flux de données, modes de communication, serveur). Nous avons ensuite développé un prototype d'application mobile faisant office d'interfaces entre l'utilisateur et le système de recommandation. L'utilisation de l'application mobile qui peut être découpée en 3 phases principales : une phase de définition du profil de l'utilisateur, une phase de génération et visualisation de la combinaison d'items et des items indépendamment, et enfin, une phase de modification totale ou partielle des offres si besoin. Dans cette partie consacrée aux applications mobiles, la structure technique de l'application iPhone a été abordée. L'application sous Android a été développée par une société extérieure.

Enfin, une dernière partie a présenté l'outil de supervision développé pour que l'ADT Côte-d'Or Tourisme puisse affiner la pertinence des résultats et/ou orienter les résultats selon ses besoins. Les outils utilisés pour cela ne sont actuellement qu'au stade de tests et sont voués à évoluer tant au niveau de l'interface que de la méthodologie.

Chapitre 7

Conclusion & Travaux Futurs

Le travail de recherche présenté dans ce mémoire concerne l'étude d'un nouveau type de systèmes de recommandation. Notre objectif est de ne plus considérer une recommandation comme un simple item, mais comme une combinaison constituée de plusieurs items suivant un pattern donné. Une recommandation d'un seul item est alors un cas particulier de ce type de recommandation. Outre les possibilités supplémentaires en termes d'interdépendance des items dans une combinaison, la notion de pattern offre un niveau supplémentaire d'adaptation à l'utilisateur. Le pattern de combinaison peut être modifié dynamiquement au cours du temps suivant le profil de l'utilisateur courant.

Le système de recommandation proposé se base sur une architecture dérivée des travaux en systèmes hypermédia adaptatifs. Nous avons identifié un ensemble minimum de trois couches pour modéliser un système hypermédia adaptatif : une couche pour modéliser la connaissance du domaine, une couche pour modéliser les profils de l'utilisateur et une couche pour réaliser l'adaptation. Nous avons bâti notre proposition sur cet ensemble en séparant dans la couche modélisant la connaissance, une sous-couche pour représenter la connaissance générale du domaine et une sous-couche pour représenter la connaissance spécifique à l'application. Cette connaissance est spécifique aux contraintes des utilisateurs. Cette deuxième partie permet de prendre en compte le savoir-faire du fournisseur de l'offre dans les propositions du système de recommandation. De plus, cette séparation rend le système plus portable, de sorte que la modélisation du contenu suivant la connaissance générale du domaine puisse être réutilisée pour une autre application.

Dans cette thèse, nous avons proposé de décomposer l'adaptation en deux sous-processus principaux, un processus dit de projection des individus du domaine sur les profils utilisateurs et un processus de recherche combinatoire. Le premier apporte une pondération, appelée note dans la plupart des systèmes de recommandation, donnant les intérêts probables des utilisateurs pour les différents items. Le processus de recherche combinatoire va rechercher parmi la multitude de combinaisons possibles, une solution convenable (optimale si possible) à proposer à l'utilisateur.

Cette architecture de système de recommandation combinatoire a été appliquée au domaine touristique pour l'agence de tourisme Côte-d'Or Tourisme, initiatrice du projet de recherche. Le but de cette application est de proposer à l'utilisateur un ensemble d'offres touristiques sous forme de séjour. Ce problème touristique est défini formellement comme un problème d'optimisation combinatoire. Ce problème est une variante d'un sous-problème (MMKP) du problème de sac à dos (KP). Pour résoudre ce problème, nous avons utilisé des métaheuristiques afin de tendre vers une bonne solution en un temps raisonnable. Les métaheuristiques les plus connues pour résoudre des problèmes de ce type ont été présentées. Parmi celles-ci, nous avons opté pour un algorithme basé sur le recuit simulé en vue de l'application mobile quasi temps réel à réaliser. En outre, un algorithme multi-objectif a aussi été développé pour d'éventuelles applications futures plus avancées.

L'instanciation de chaque couche de l'architecture pour le système touristique a été décrite en détail. De plus, le processus de création d'une ontologie de domaine en se basant sur une base de données et la connaissance d'experts a été détaillé. On remarque que l'ontologie de domaine touristique réalisée est très proche du modèle conceptuel de données de la base de données de Côte-d'Or tourisme. Cela se justifie par le fait que la base de données modélisait déjà la connaissance du domaine touristique.

Enfin, cette thèse a présenté les développements techniques qui ont été nécessaires pour ce projet, étant donné son contexte industriel. Ainsi, nous avons montré que notre architecture de système de recommandation combinatoire basé sur un pattern et prenant en compte le savoir-faire des fournisseurs de l'offre pouvait être instanciée afin d'engendrer un système de recommandation proposant des produits touristiques sous forme de séjours.

Le problème d'optimisation combinatoire que nous avons traité dans ce document est propre au domaine du tourisme. Il ne peut être simplement généralisé pour d'autres domaines d'application. Néanmoins, l'architecture que nous avons développée peut être réutilisée pour d'autres domaines, si le problème d'optimisation combinatoire de ces domaines est identifié.

La modélisation de la couche sémantique se fait en utilisant une ontologie. Ses intérêts principaux sont qu'elle permet d'exprimer, décrire pleinement la connaissance sans perte d'information. Elle permet en outre de conceptualiser un système distribué. De plus, sa standardisation à travers le langage OWL permet une portabilité accrue de cette couche du système.

En dehors de ces aspects, les ontologies peuvent être la base de futures améliorations de l'architecture. En effet, nous les sous-exploitions, elles ne sont pas le support d'inférences logiques qui nous permettraient par exemple de déduire de nouveaux buts d'un utilisateur à partir de ses buts connus et de la partie statique de son profil. Cette perspective accentue l'intérêt d'utiliser une ontologie dans l'architecture.

D'une manière générale, nous avons identifié plusieurs pistes pour poursuivre nos travaux de recherche.

Extension vers un système hybride. Notre proposition se classe parmi les systèmes de recommandation basés sur le contenu. Nous pouvons étendre notre proposition en ajoutant des méthodes collaboratives à notre architecture. Par exemple, il serait possible de lui ajouter une nouvelle couche appelée « modèle des groupes utilisateurs » contenant des clusters d'utilisateurs, de sorte de relier les utilisateurs entre eux selon leur profil et les notes données aux items ou combinaisons d'items. Il faudrait alors ajouter de nouvelles phases au sein de la couche intelligence pour prendre en compte ces groupements d'utilisateurs. Elles engendreraient une modification de la pondération des items selon ces groupements. Le système hybride résultant donnerait des résultats plus pertinents pour les utilisateurs.

Extension vers un système à base d'apprentissage. Dans notre système de recommandation, aucun avis n'est demandé à l'utilisateur sur la combinaison qui lui est proposée. On peut néanmoins supposer que tout ou certains items ne lui plaisent pas lorsqu'il demande une modification de la combinaison. Dans tous les cas, ce retour utilisateur n'est pas pris en compte pour d'éventuelles futures propositions. Il n'y a pas d'apprentissage suivant l'avis ou les modifications des utilisateurs sur les propositions qui lui sont faites. Nous pourrions envisager d'ajouter une phase d'apprentissage dans la couche intelligence, qui modifierait la pondération, voire le pattern de combinaison, suivant l'avis/les modifications des utilisateurs sur les combinaisons qui lui sont fournies. L'ajout de cette partie d'apprentissage est nécessaire pour une évolution de la qualité des recommandations sans l'intervention d'administrateurs du système et experts.

Limiter l'espace de recherche. Actuellement, dans l'application touristique, nous considérons que la tolérance de l'utilisateur sur la dispersion des items d'une combinaison n'est pas stricte. Si la dispersion d'items d'une combinaison dépasse la tolérance, cela engendre une diminution de la pertinence de la combinaison, bien que celle-ci reste proposable. Avec cette possibilité, le système est toujours capable de proposer une solution quelques soient les critères demandés, même si aucune n'est possible dans la zone de tolérance de l'utilisateur. Néanmoins, ce caractère ouvert du système engendre un espace de recherche combinatoire important. Pour répondre à ce problème, une optimisation pourrait être de limiter la recherche combinatoire à un espace plus restreint de solutions selon la position géographique de l'utilisateur et sa tolérance de dispersion. La recherche ne se ferait alors que sur les items se trouvant dans la zone correspondante. Lorsqu'aucune solution n'est trouvée dans la zone spécifiée et pour garder la capacité du système à proposer des solutions quelques soient les critères, une possibilité serait d'effectuer une nouvelle recherche avec une tolérance plus grande. Néanmoins, cette modification ne donnerait pas les mêmes résultats que la méthode de recherche initiale. En effet, par exemple, la meilleure combinaison trouvée dans la zone limitée peut être moins bonne qu'une

combinaison contenant un item (avec un grand poids) hors de la zone et avec tous les autres dedans. Ainsi, il faudrait étudier l'impact de cette modification sur la qualité des résultats et sur les temps de calcul.

Métaheuristique pour optimiser les paramètres de l'algorithme du recuit simulé. Dans l'application touristique, un algorithme basé sur le recuit simulé est utilisé pour réaliser la recherche combinatoire. Pour avoir de bonnes solutions en un temps raisonnable, plusieurs paramètres doivent être définis par tâtonnement. Ces paramètres incluent la tolérance initiale d'acceptation, le coefficient de décroissance de la température, le nombre d'itérations par palier. Ils influent sur le temps pour trouver la solution finale, et sur la pertinence de cette dernière. Pour éviter de définir ces paramètres de manière arbitraire, nous pourrions utiliser une métaheuristique qui nécessite moins de paramétrages. Pour ce faire, le temps d'exécution n'a pas besoin d'être temps réel, ces paramètres ne devant pas être calculés à chaque itération de la recherche combinatoire. Deux objectifs sont à optimiser : maximiser la pertinence de la solution finale et minimiser le temps d'exécution pour la trouver. On peut alors envisager un algorithme multi-objectif basé sur une population tel que défini dans le chapitre 5. La population initiale serait un ensemble P d'individus i_k aléatoires de la forme:

$i_k = \{\text{tolerance, coef, nbiter}\}$, tel que $0 < \text{tolerance} < 1$, $0 < \text{coef} < 1$ et $0 < \text{nbiter} < 100000$.

tolerance est la tolérance d'acceptation initiale des mauvaises solutions, coef le coefficient de décroissance de la température, et nbiter le nombre d'itérations des modifications par palier. Du fait du caractère stochastique de l'algorithme du recuit simulé, des paramètres initiaux identiques peuvent donner des résultats différents dans les deux objectifs que nous souhaitons optimiser. Ainsi, pour un individu donné de la population, l'algorithme du recuit simulé avec ses paramètres est exécuté plusieurs fois, nous obtiendrons généralement des valeurs différentes dans les objectifs d'une itération à une autre. De ce fait, pour calculer les valeurs des objectifs d'un individu de la population, nous pourrions réaliser la moyenne des valeurs obtenue dans chaque objectif sur plusieurs itérations du recuit simulé (100 itérations par exemple). Ces calculs doivent être faits sur tous les individus à chaque nouvelle population (sauf sur les individus qui restent de la population précédente). On imagine alors très bien le temps important d'exécution que nécessiterait un tel algorithme multi-objectif pour optimiser le recuit simulé. Néanmoins, son intérêt serait réel, car on obtiendrait au final une population de bons individus. Le choix de l'individu optimal se ferait en prenant celui qui génère en moyenne les solutions les plus pertinentes avec un temps d'exécution inférieur au temps maximal que l'on s'autorise. On pourrait appeler cette optimisation, une méta-optimisation.

Vers le tout adaptatif. Dans un système hypermédia adaptatif, il s'agit d'adapter tout un ensemble de composants qui sont le contenu, la présentation et les liens de navigation dans une page hypermédia. De façon similaire à notre vision d'une recommandation comme une combinaison d'items à suggérer à l'utilisateur selon un pattern, on peut voir la page hypermédia d'un système hypermédia adaptatif comme une combinaison de sous-combinaisons d'items à proposer à l'utilisateur dont le pattern est aussi adapté dynamiquement. Ce problème peut donc aussi être vu comme un problème d'optimisation combinatoire où il faut chercher une combinaison de sous combinaisons d'items optimisant une fonction de pertinence composée de sous-fonctions de pertinence. Il convient aussi de chercher un pattern tel qu'il maximise une autre fonction de pertinence de pattern. La définition de ces fonctions semble être l'une des principales difficultés de ces problèmes d'optimisation. Cette approche me semble intéressante pour de futures recherches en systèmes hypermédia adaptatifs.

Ontologie support d'inférence. Dans notre architecture, l'ontologie utilisée par la couche sémantique n'est pas le support d'inférence logique. Une amélioration serait de modéliser toutes les informations utilisateur au sein de l'ontologie pour réaliser de l'inférence logique déduisant de nouveaux buts de l'utilisateur à partir de ses buts connus et de la partie statique de son profil.

Amélioration de la pondération des items par propagation à travers l'ontologie. Actuellement, les items sont pondérés suivant leur affiliation aux concepts buts (par des règles) choisis par l'utilisateur. On peut néanmoins imaginer une méthode de pondération des items plus complexe par propagation des poids à travers l'ontologie de domaine suivant les relations entre concepts et individus. Ainsi, on peut envisager différentes manières de propager les poids suivant les types des relations. Par exemple, une relation de type opposition entre deux concepts pourrait engendrer la propagation d'un poids négatif. Cette méthode par propagation peut clairement améliorer la méthode de pondération utilisée, car elle peut entraîner l'augmentation ou la diminution des poids d'items qui ne l'auraient pas été forcément avec la seule méthode initiale. De plus, on peut imaginer une capacité d'apprentissage qui modifierait la manière de propager les poids à travers l'ontologie suivant les relations.

Bibliographie

Last.fm: Music recommendation service . (2009). Retrieved from <http://www.last.fm>

Aciar, S., Zhang, D., Simoff, S., & Debenham, J. (2007). Informed Recommender: Basing Recommendations on Consumer Product Reviews. *IEEE Intelligent Systems* 22, 39-47.

Adomavicius, G., & T. A. (2005). Toward the next generation of recommender systems: A survey. *IEEE Transactions on Knowledge and Data*, 734-749.

Ahn, J., Brusilovsky, P., Grady, J., He, D., & Syn, S. (2007). Open User Profiles for Adaptive News Systems: Help or Harm? *16th International Conference on World Wide Web* (pp. 11-20). ACM.

Ali, K., & van Stam, W. (2004). TiVo: Making Show Recommendations Using a Distributed Collaborative Filtering Architecture. *10th ACM SIGKDD Int. Conference on Knowledge Discovery and Data Mining*, (pp. 394-401).

Anand, S., & Mobasher, B. (2005). Intelligent techniques for web personalization. *Intelligent Techniques for Web Personalization*, 1-36.

Armstrong, J. S. (2001). Principles of Forecasting. *A Handbook for Researchers and Practitioners*.

Asnicar, F., & Tasso, C. (1997). ifWeb: a Prototype of User Model-based Intelligent Agent for Documentation Filtering and Navigation in the World Wide Web. *First International Workshop on Adaptive Systems and User Modeling on the World Wide Web*, (pp. 3-12). Chia Laguna, Sardinia, Italy.

Avella, P., Dauria, B., & Salerno, S. (2006). A Ip-based heuristic for a time-constrained routing problem. *European Journal of Operational Research*.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). Modern Information Retrieval. *Addison-Wesley*.

Balabanovic, M. S. (1997). Fab: Content-based, Collaborative Recommendation. *Communications of the ACM*, 66-72.

Basile, P., de Gemmis, M., Gentile, A., Iaquinta, L., & Lops, P. (2007). The JUMP project: Domain Ontologies and Linguistic Knowledge @ Work. *4th Italian Semantic Web Applications and Perspectives - SWAP 2007*.

Basile, P., de Gemmis, M., Gentile, A., Iaquinta, L., Lops, P., & Semeraro, G. (2007). An Electronic Performance Support System Based on a Hybrid Content-Collaborative Recommender System. *Neural Network World: International Journal on Neural and Mass-Parallel Computing and Information Systems* 17, 529-541.

Basile, P., Degemmis, M., Gentile, A., Lops, P., & Semeraro, G. (2007). UNIBA: JIGSAW algorithm for Word Sense Disambiguation. *4th ACL 2007 International Workshop on Semantic Evaluations (SemEval-2007)*, (pp. 398-401). Prague, Czech Republic.

Bell, R., & Koren, Y. (2007). Scalable collaborative filtering with jointly derived neighborhood interpolation weights. *2007 Seventh IEEE Int. Conf. on Data Mining* (pp. 43-52). Washington, DC, USA: IEEE Computer Society.

Bell, R., Koren, Y., & Volinsky, C. (2007). Modeling relationships at multiple scales to improve accuracy of large recommender systems. *13th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining* (pp. 95-104). New York, NY, USA: ACM.

- Berners-Lee, T. (1998). *What the Semantic Web can represent ?* Retrieved from <http://www.w3.org/DesignIssues/RDFnot.html>
- Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American*.
- Billsus, D., & Pazzani, M. (1999). A Hybrid User Model for News Story Classification. *Seventh International Conference on User Modeling*. Banff, Canada.
- Billsus, D., & Pazzani, M. (2000). User Modeling for Adaptive News Access. *User Modeling and User-Adapted Interaction*, 147-180.
- Blanco-Fernandez, Y., J., P.-A. J., G.S.A., Ramos-Cabrer, M., & Lopez-Nores, M. (2008). Providing Entertainment by Content-based Filtering and Semantic Reasoning in Intelligent Recommender Systems. *IEEE Transactions on Consumer Electronics* 54, 727-735.
- Blei, D., Ng, A., & Jordan, M. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 993-1022.
- Bollacker, K., & Giles, C. (1998). CiteSeer: An Autonomous Web Agent for Automatic Retrieval and Identification of Interesting Publications. *Second International Conference on Autonomous Agents* (pp. 116-123). ACM Press .
- Breese, J., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. *14th Annual Conf. on Uncertainty in Artificial Intelligence*, (pp. 43-52).
- Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Models and User Adapted Interaction* 6.
- Burke, R. (2007). Hybrid web recommender systems. *The Adaptive Web*, 377-408.
- Cantador, I., & Castells, P. (2006). A multilayered ontology-based user profiles and semantic social networks for recommender systems. *2nd International Workshop on Web Personalisation Recommender Systems and Intelligent User Interfaces in Conjunction with 7th International Conference in Adaptive Hypermedia*.
- Cantador, I., Bellogin, A., & Castells, P. (2008). News@hand: A Semantic Web Approach to Recommending News. *Adaptive Hypermedia and Adaptive Web-Based Systems*, 279-283.
- Cauchy, A. (1847). *Méthode générale pour la résolution des systèmes d'équations simultanées*. Comptes Rendus de l'Académie des Sciences de Paris.
- Celma, O., & Serra, X. (2008). FOAFing the Music: Bridging the Semantic Gap in Music Recommendation. *Web Semantics*, 250-256.
- Celma, O., Ramirez, M., & Herrera, P. (2005). Foafing the Music: A Music Recommendation System based on RSS Feeds and User Preferences. *6th International Conference on Music Information Retrieval (ISMIR)*, (pp. 464-467). London, UK .
- Chen, L., & Sycara, K. (1998). WebMate: A Personal Agent for Browsing and Searching. *2nd International Conference on Autonomous Agents* (pp. 9-13). New York : ACM Press.
- Chen, L., Khan, S., Li, K., & Manning, E. (1999). Building an adaptive multimedia system using the utility model. *Lecture Notes in Computer Science*.
- Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., & Sartin, M. (1999). Combining Content-Based and Collaborative Filters in an Online Newspaper. *ACM SIGIR Workshop on Recommender Systems*.

- Collins, A., & Loftus, E. (1975). A Spreading Activation Theory of Semantic Processing. *Psychological Review*, 407-428.
- Colorni, A., Dorigo, M., & Maniezzo, V. (1991). Distributed optimization by ant colonies. *Proceedings of the First European Conference on Artificial Life*, (pp. 134-142). Paris, France.
- Cristea, A., & de Mooij, A. (2003). LAOS: Layered WWW AHS Authoring Model and their corresponding Algebraic Operators. *WWW 2003 Conference*. ACM.
- De Bra, P., & Calvi, L. (1998). AHA! an open adaptive hypermedia architecture. *Hypermedia*, 115-139.
- De Bra, P., Aerts, A., Smits, D., & Stash, N. (2002). AHA! Version 2.0, More Adaptation Flexibility for Authors. *World Conference on e-Learning in Corporate, Government, Healthcare and Higher Education*. Montreal, Canada.
- De Bra, P., Houben, G.-J., & Wu, H. (1999). AHAM: A dexter-based reference model for adaptive hypermedia. *ACM Conference on Hypertext and Hypermedia: Returning to our Diverse Roots*, 147-156.
- De Bra, P., Wu, H., & Houben, G.-J. (2000). Supporting user adaptation in adaptive hypermedia applications. *InfWet'00 Conference*.
- Degemmis, M., Lops, P., & Semeraro, G. (2007). A Content-collaborative Recommender that Exploits WordNet-based User Profiles for Neighborhood Formation. *User Modeling and User-Adapted Interaction: The Journal of Personalization Research (UMUAI)*, 217-255.
- Deshpande, M., & Karypis, G. (2004). Item-based top-N recommendation algorithms. *ACM Transaction on Information Systems*, 143-177.
- DMOZ open directory project. (n.d.). Retrieved from <http://www.dmoz.org/>
- Dueck, G., & Scheuer, T. (1990). Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing. *Journal of Computational Physics*, 161-175.
- Eirinaki, M., Vazirgiannis, M., & Varlamis, I. (2003). SEWeP: Using Site Semantics and a Taxonomy to enhance the Web Personalization Process. *Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (pp. 99-108).
- Geman, S., & Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (pp. 721-741).
- Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 533-549.
- Goldberg, D., Nichols, D., Oki, B., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 61-70.
- Gondran, M., & Minoux, M. (1985). *Graphes et algorithmes*. Editions Eyrolles.
- Grcar, M., Fortuna, B., Mladenic, D., & Grobelnik, M. (2006). k-NN versus SVM in the collaborative filtering framework. *Data Science and Classification*, 251-260.
- Gruber, T. R. (1993). Toward principles for the design of ontologies used for knowledge sharing. *International Workshop on Formal Ontology*. Padova, Italy.
- Halasz, F., & Schwartz, M. (1990). The dexter reference model. *NIST Hypertext Standardization Workshop*.

- Hayes, P. (2004). *RDF Semantics*. W3C. <http://www.w3.org/TR/2004/REC-rdf-nt-20040210/>.
- Henze, N. (2000). *Adaptive hyperbooks: Adaptation for project-based learning resources*. University of Hannover.
- Hill, W., Stead, L., Rosenstein, M., & Furnas, G. (1995). Recommending and Evaluating Choices in a Virtual Community of Use. *Human Factors in Computing Systems*.
- Hofmann, T. (2003). Collaborative filtering via Gaussian probabilistic latent semantic analysis. *26th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval* (pp. 259-266). New York, NY, USA: ACM.
- Holland, J. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press.
- Ikeda, M. (1998). Ontology as Theoretical Foundations of Knowledge Engineering. *J. Jpn. Soc. for Artificial intelligence*, 13, 11-13.
- Jacobs, I., & Walsh, N. (2004). *Architecture of the World Wide Web*. W3C. <http://www.w3.org/TR/webarch/>.
- Jannach, D. (2006). Finding preferred query relaxations in content-based recommenders. *3rd International IEEE Conference on Intelligent Systems*, (pp. 355-360).
- Joachims, T., Freitag, D., & Mitchell, T. (1997). Web Watcher: A Tour Guide for the World Wide Web. *15th International Joint Conference on Artificial Intelligence*, (pp. 770-777).
- Kamba, T., Sakagami, H., & Koseki, Y. (2007). Antagonomy: A Personalized Newspaper on the World Wide Web. *Int'l J. Human-Computer Studies*.
- Karp., R. M. (1972). Reducibility among combinatorial. *Complexity of Computer Computations*, 85-103.
- Koch, N., & Wirsing, M. (2002). The Munich Reference Model for Adaptive Hypermedia Applications. *2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, (pp. 213-222).
- Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., & Riedl, J. (1997). GroupLens: applying collaborative filtering to usenet news. *Communications of the ACM*, 77-87.
- Koren, Y. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. *14th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining* (pp. 426-434). New York, NY, USA: ACM.
- Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *IEEE Computer*, (pp. 30-37).
- Lagoudakis, M. (1996). *The 0-1 knapsack problem: An introductory survey*. University of Southwestern Louisiana: The Center for Advanced Computer Studies.
- Lieberman, H. (1995). Letizia: An Agent that Assists Web Browsing. *International Joint Conference on Artificial Intelligence (IJCAI-95)* (pp. 924-929). Montreal, Canada: Morgan Kaufmann publishers Inc.
- Lilien, G., Kotler, P., & Moorthy, K. (1992). *Marketing Models*. Prentice Hall.
- Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 76-80.
- Magnini, B., & Strapparava, C. (2000). Experiments in Word Domain Disambiguation for Parallel Texts. *SIGLEX Workshop on Word Senses and Multi-linguality*. Hong-Kong.

- Magnini, B., & Strapparava, C. (2001). Improving User Modelling with Content-based Techniques. *8th International Conference of User Modeling*, (pp. 74-83).
- Mahmood, T., & Ricci, F. (2009). Improving recommender systems with adaptive conversational strategies. *Hypertext*, 73-82.
- Mak, H., Koprinska, I., & Poon, J. (2003). INTIMATE: A Web-Based Movie Recommender Using Text Categorization. *IEEE/WIC International Conference on Web Intelligence* (pp. 602-605). IEEE Computer Society .
- Martello, S., & Toth, P. (1990). *Knapsack problems: algorithms and computer implementations*. New York, NY, USA: John Wiley & Sons.
- McSherry, F., & Mironov, I. (2009). Differentially private recommender systems: building privacy into the net. *15th ACM SIGKDD international conference on Knowledge discovery and data mining*, (pp. 627-636). New York, USA.
- Melville, P., Mooney, R., & Nagarajan, R. (2002). Content-Boosted Collaborative Filtering for Improved Recommendations. *Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI-02)* (pp. 187-192). Menlo Parc, CA, USA: AAAI Press.
- Metropolis, N., Rosenbluth, M., Rosenbluth, A., Teller, A., & Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, p. 1087-1091.
- Middleton, S., Alani, H., Shadbolt, N., & De Roure, D. (2002). Exploiting synergy between ontologies and recommender systems. *WWW international workshop on the semantic web*. Maui, HW, USA.
- Middleton, S., Shadbolt, N., & De Roure, D. (2004). Ontological User Profiling in Recommender Systems. *ACM Transactions on Information Systems*, 54-88.
- Miller, B. N., Albert, I., Lam, S., Konstan, J., & Riedl, J. (2003). MovieLens Unplugged: Experiences with an Occasionally Connected Recommender System. *Intelligent User Interfaces*.
- Minio, M., & Tasso, C. (1996). User Modeling for Information Filtering on INTERNET Services: Exploiting an Extended Version of the UMT Shell. *UM96 Workshop on User Modeling for Information Filtering on the WWW*. Kailua-Kona, Hawaii.
- Mladenic, D. (1999). Machine learning used by PersonalWebWatcher. *ACAI-99 Workshop on Machine Learning and Intelligent Agents* .
- Mladenic, D. (1999). Text-learning and Related Intelligent Agents: A Survey. *IEEE Intelligent Systems* , 44-54.
- Mooney, R., & Roy, L. (2000). Content-Based Book Recommending Using Learning for Text Categorization. *5th ACM Conference on Digital Libraries* (pp. 195-204). New York, US, San Antonio: ACM Press.
- Moukas, A. (1997). Amalthea Information Discovery and Filtering Using a Multiagent Evolving Ecosystem. *Applied Artificial Intelligence* , 437-457.
- Mukherjee, R., Jonsdottir, G., Sen, S., & Sarathi, P. (n.d.). MOVIES2GO: an Online Voting based Movie Recommender System. *Fifth International Conference on Autonomous Agents* (pp. 114-115). ACM Press.
- Murthi, B., & Sarkar, S. (2003). The Role of the Management Sciences in Research on Personalization. *Management Science*, 1344-1362.
- Netflix*. (n.d.). Retrieved from www.netflix.com/
- Nicholson, T. (1971). Optimization in industry. *Optimization Techniques*.

- Pandora Internet Radio*. (n.d.). Retrieved from <http://www.pandora.com>
- Paterek, A. (2007). Improving regularized singular value decomposition for collaborative filtering. *KDD Cup and Workshop*.
- Pazzani, M. B. (2007). Content-Based Recommendation Systems. *The Adaptive Web*, 325-341.
- Pazzani, M., & Billsus, D. (1997). Learning and Revising User Profiles: The Identification of Interesting Web Sites. *Machine Learning*, 313-331.
- Pazzani, M., Muramatsu, J., & Billsus, D. (1996). Syskill and Webert: Identifying Interesting Web Sites. *Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference* (pp. 54-61). Menlo Park: AAAI Press / MIT Press.
- Powell, M. J. (1981). *Approximation Theory and Methods*. Cambridge Univ. Press.
- Quillian, R. (1968). Semantic memory. *Semantic information processing*, 227-270.
- Razmerita, L. V. (2003). *Modèle Utilisateur et Modélisation Utilisateur dans les Systèmes de Gestion des Connaissances: une Approche fondée sur les Ontologies*.
- Rechenberg, I. (1973). *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien*. Frommann-Holzboog, Stuttgart.
- Resnick, P., & Varian, H. (1997). Recommender systems. *Communications of the ACM* 40, 56-58.
- Resnick, P., Iakovou, N., Sushak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. *Computer Supported Cooperative Work Conf.*
- Rich, E. (1979). User Modeling via Stereotypes. *Cognitive Science*, 329-354.
- Salakhutdinov, R., Mnih, A., & Hinton, G. (2007). Restricted Boltzmann machines for collaborative filtering. *24th international conference on Machine learning* (pp. 791-798). New York, NY, USA: ACM.
- Salter, J., & Antonopoulos, N. (2006). CinemaScreen Recommender Agent: Combining collaborative and content-based filtering. *IEEE Intelligent Systems*, 35-41.
- Salton, G. (1989). *Automatic Text Processing*. Addison-Wesley.
- Schafer, J., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. *The Adaptive Web*, 291-324.
- Schwartz, B. (2004). *The Paradox of Choice*. ECCO. New York.
- Schwefel, H. (1995). *Evolution and Optimum Seeking*. Wiley, New York.
- Semeraro, G., Basile, P., de Gemmis, M., & Lops, P. (2009). User Profiles for Personalizing Digital Libraries. *Handbook of Research on Digital Libraries: Design, Development and Impact*, 149-158.
- Semeraro, G., Degemmis, M., Lops, P., & Basile, P. (2007). Combining Learning and Word Sense Disambiguation for Intelligent User Profiling. *20th International Joint Conference on Artificial Intelligence* (pp. 2856-2861). ISBN.
- Shardanand, U., & Maes, P. (1995). Social Information Filtering: Algorithms for Automating "Word of Mouth". *Human Factors in Computing Systems*.
- Sheth, B., & Maes, P. (1993). Evolving Agents for Personalized Information Filtering. *Ninth Conference on Artificial Intelligence for Applications* (pp. 345-352). IEEE Computer Society Press.

- Sieg, A., Mobasher, B., & Burke, R. (2007). Learning Ontology-Based User Profiles: A Semantic Approach to Personalized Web Search. *IEEE Intelligent Informatics Bulletin*.
- Smith, B., & Cotter, P. (2000). A Personalized TV Listings Service for the Digital TV Age. *Knowledge-Based Systems*, 53-59.
- Sorensen, H., & McElligott, M. (1995). PSUN: A Profiling System for Usenet News. *CIKM '95 Intelligent Information Agents Workshop*.
- Sorensen, H., O'Riordan, A., & O'Riordan, C. (1997). Profiling with the INFormer Text Filtering Agent. *Journal of Universal Computer Science*, 988-1006.
- Stash, N. (2007). *Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System*. PhD Thesis. Eindhoven University of Technology, Netherlands.
- Stefani, A., & Strapparava, C. (1998). Personalizing Access to Web Sites: The SiteIF Project. *Second Workshop on Adaptive Hypertext and Hypermedia*. Pittsburgh.
- Straffin, P. (1980). Topics in the Theory of Voting. *The UMAP expository monograph series*.
- Symeonidis, P. (2008). Content-based Dimensionality Reduction for Recommender Systems. *Data Analysis, Machine Learning and Applications, Studies in Classification, Data Analysis, and Knowledge Organization*, 619-626.
- The Internet Movie Database*. (n.d.). Retrieved from <http://www.imdb.com/>
- Wasfi, A. (1999). Collecting User Access Patterns for Building User Profiles and Collaborative Filtering. *International Conference on Intelligent User Interfaces*, (pp. 57-64).
- Welty, C., Smith, M., & McGuinness, D. (2004). *OWL Web Ontology Language Guide*. W3C. <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>.
- Zitnick, C., & Kanade, T. (2004). Maximum entropy for collaborative filtering. *20th Conf. on Uncertainty in Artificial Intelligence* (pp. 636-643). Arlington, Virginia, United States : AUAI Press.

Rapports techniques

Picot Clément R., *Hypermédia adaptatif*, Le2i - Université de Bourgogne, Dijon, France, Décembre 2008.

Picot Clément R., *Modèle de domaine*, Le2i - Université de Bourgogne, Dijon, France, Décembre 2008.

Picot Clément R., *Modèle utilisateur*, Le2i - Université de Bourgogne, Dijon, France, Décembre 2008.

Picot Clément R., *Adaptabilité*, Le2i - Université de Bourgogne, Dijon, France, Décembre 2008.

Picot Clément R., *Résumé thèse Cédric Jacquot : Modèle générique pour la création de systèmes hypermédias adaptatifs*, Le2i - Université de Bourgogne, Dijon, France, Janvier 2009.

Picot Clément R., *Comparaison entre Ajax et Flash*, Le2i - Université de Bourgogne, Dijon, France, Janvier 2009.

Picot Clément R., *Radial tree*, Le2i - Université de Bourgogne, Dijon, France, Janvier 2009.

Picot Clément R., *Introduction aux ontologies*, Le2i - Université de Bourgogne, Dijon, France, Février 2009.

Picot Clément R., *Construction, initialisation et mise à jour du modèle utilisateur*, Le2i - Université de Bourgogne, Dijon, France, Février 2009.

Picot Clément R., *Etat de la recherche en eTourisme*, Le2i - Université de Bourgogne, Dijon, France, Février 2009.

Picot Clément R., *Systèmes de recommandation basés sur les ontologies*, Le2i - Université de Bourgogne, Dijon, France, Mars 2009.

Picot Clément R., *Mécanisme d'adaptation pour la navigation adaptative*, Le2i - Université de Bourgogne, Dijon, France, Avril 2009.

Picot Clément R., *Définition d'un ensemble de règles pour la construction de l'ontologie et prospective technique de Jena et JDBC*, Le2i - Université de Bourgogne, Dijon, France, Mai 2009.

Picot Clément R., *Définition d'une ontologie modélisant le domaine et une ontologie modélisant l'utilisateur issue de capitalisation de la connaissance depuis un corpus de documents et de l'interview des acteurs du domaine*, Le2i - Université de Bourgogne, Dijon, France, Juin 2009.

Picot Clément R., *Modélisation de l'utilisateur et moteur d'adaptation*, Le2i - Université de Bourgogne, Dijon, France, Juin 2009.

Picot Clément R., *Système hypermédia adaptatif et processus de construction d'une ontologie de domaine basée sur une base de données*, Le2i - Université de Bourgogne, Dijon, France, Juillet 2009.

Picot Clément R., *Etude « GAF: Generic Adaptation Framework, 2009 »*, Le2i - Université de Bourgogne, Dijon, France, Juillet 2009.

Picot Clément R., *Construction ontologie à partir de vues sur la base de données*, Le2i - Université de Bourgogne, Dijon, France, Août 2009.

Picot Clément R., *Etude de « AHA! An open Adaptive Hypermedia Architecture, 1998, Paul De Bra et al. »*, Le2i - Université de Bourgogne, Dijon, France, Septembre 2009.

Picot Clément R., *Etude de « Creating Adaptive Web-Based Applications, 2005, Paul De Bra et al. »*, Le2i - Université de Bourgogne, Dijon, France, Septembre 2009.

Picot-Clément R., *Définition d'une architecture générique des systèmes hypermédia adaptatifs*, Le2i - Université de Bourgogne, Dijon, France, Octobre 2009.

Picot-Clément R., *L'algorithme du recuit simulé appliqué à l'adaptation du contenu d'un système hypermédia adaptatif*, Le2i - Université de Bourgogne, Dijon, France, Décembre 2009.

Picot-Clément R., *Fonctionnement de l'application hypermédia adaptative touristique - du profilage utilisateur à la recommandation*, Le2i - Université de Bourgogne, Dijon, France, Janvier 2010.

Picot-Clément R., *Avancement du développement du système hypermédia adaptatif*, Le2i - Université de Bourgogne, Dijon, France, Mars 2010.

Picot-Clément R., *Logiques de représentation de la connaissance, application à notre ontologie*, Le2i - Université de Bourgogne, Dijon, France, Juin 2010.

Picot-Clément R., *Perspectives*, Le2i - Université de Bourgogne, Dijon, France, Juin 2010.

Picot-Clément R., *Recommandation à des groupes*, Le2i - Université de Bourgogne, Dijon, France, Juillet 2010.

Picot-Clément R., *Formalisation du problème*, Le2i - Université de Bourgogne, Dijon, France, Juillet 2010.

Picot-Clément R., *Etat de l'art et comparaison entre systèmes de recommandation*, Le2i - Université de Bourgogne, Dijon, France, Juillet 2010.

Picot-Clément R., *Problèmes d'optimisation*, Le2i - Université de Bourgogne, Dijon, France, Novembre 2010.

Picot-Clément R., *Principales métaheuristiques*, Le2i - Université de Bourgogne, Dijon, France, Novembre 2010.

Picot-Clément R., *Proposition d'amélioration du principe de pondération*, Le2i - Université de Bourgogne, Dijon, France, Décembre 2010.

Publications

Picot-Clémente R., Mendes F., Cruz C. & Nicolle C., *TOURISM-KM, A variant of MMKP applied to the tourism domain*. 1st International Conference on Operations Research and Enterprise Systems (ICORES 2012), Portugal, February 2012.

Picot-Clémente R., Cruz C. & Nicolle C., *An ontology-based approach to provide personalized recommendations using a stochastic algorithm*. WEBIST Special Session on Semantic Web Applications and Tools (SWAT 2011), Noordwijkerhout, Netherlands, May 2011.

Picot-Clémente R., Cruz C. & Nicolle C., *A Semantic-based Recommender System Using A Simulated Annealing Algorithm*. SEMAPRO 2010, The Fourth International Conference on Advances in Semantic Processing, Florence, Italy, October 2010.

Picot-Clémente R., Cruz C. & Nicolle C., *TourismKM: a new Web Semantic based approach for E-Tourism*. International Journal of Trade, Economics, and Finance (IJTEF) **1**, 1, June 2010.

Picot-Clémente R., Cruz C. & Nicolle C., *Contribution of Semantic Web on Customer Relationship Management in E-Tourism*. CBETM 2010, International Conference on Business, Economics and Tourism Management, Singapore, February 2010.