

Université de Montréal

Le raisonnement à base de cas
dans la planification financière

par

Kamel Boudina

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

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

Avril 1998

© Kamel Boudina, 1998



QA
76
U54
1998
v.018



Université de Montréal

Bibliothèque



Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé :

Le raisonnement à base de cas
dans la planification financière

Présenté par :

Kamel Boudina

a été évalué par un jury composé des personnes suivantes :

Président du jury : Claude Frasson
Directrice de recherche : Esma Aïmeur
Membre du jury : Paul Bratley

Mémoire accepté le : / 0 . 0 8 . 1 9 9 8

Remerciements

Je tiens en tout premier lieu à remercier vivement ma Directrice de recherche, Madame Esma Aïmeur, pour sa disponibilité et ses précieux conseils tout au long de cette maîtrise. Qu'elle trouve ici, l'expression de ma profonde gratitude.

Je remercie Monsieur Claude Frasson pour l'intérêt qu'il a porté à ce travail.

Je remercie Monsieur Benakezou, gestionnaire de portefeuille au sein du groupe *La Mutuelle*, pour les idées fructueuses qui m'ont aidé à mener à bien ce projet.

Je remercie Monsieur Gagné, gestionnaire de portefeuilles pour le compte de l'institution *Interinvest corporation* et Monsieur Steele, responsable des obligations au sein de l'institution *Interinvest corporation*, pour l'attention qu'ils m'ont accordée.

Enfin, je remercie très chaleureusement toute ma famille pour son soutien permanent.

À ma grand-mère.

À mes parents.

Sommaire

L'analyse financière se base sur des concepts et des règles complexes afin de proposer des solutions adaptées aux problèmes à résoudre. L'évaluation d'une situation financière particulière doit tenir compte de facteurs humains, comme la tolérance aux risques de la part des épargnants et le comportement des consommateurs. Elle doit également tenir compte de facteurs politiques, comme les variations des taux d'intérêts et la politique monétaire. Le planificateur financier procède à l'analyse de la situation financière d'un client afin d'aboutir à l'élaboration d'un portefeuille financier adapté à ses besoins. Il s'agit d'une tâche complexe qui exige de la part de l'analyste financier; une maîtrise de son domaine et un niveau d'expertise considérable. En raison de la diversité et de la nature des paramètres dressant le profil financier d'un client, il est nécessaire de disposer d'outils qui mémorisent ces informations afin de les réutiliser dans d'autres situations. Pour répondre, même partiellement à ces problèmes, nous proposons un système qui utilise le raisonnement à base de cas. Dans ce système, l'approche raisonnement à base de cas est utilisée, évidemment, dans la phase de recherche de cas similaires, mais aussi, de façon réflexive, dans la phase d'adaptation. Ce système permet la saisie d'informations décrivant un client, comme les objectifs financiers, le risque acceptable, le revenu, etc., afin d'arriver à proposer un portefeuille financier équilibré. Le système se base sur des cas déjà traités et gardés en mémoire pour adapter la solution du cas le plus similaire au cas courant.

Mots clés :

Système de raisonnement à base de cas, situation financière, portefeuille financier, approche réflexive.

Table des matières

SOMMAIRE.....	I
TABLE DES MATIÈRES.....	IV
LISTE DES TABLEAUX.....	VI
LISTE DES FIGURES.....	VII
1 INTRODUCTION	1
2 RAISONNEMENT À BASE DE CAS.....	5
2.1. PRINCIPES DE BASE.....	5
2.2. SENSIBILISATION DE L'INDUSTRIE AU CBR.....	6
2.3. HISTORIQUE	6
2.4. VOCABULAIRE ET INDEXATION	8
2.5. ORGANISATION DE LA MÉMOIRE	8
2.5.1. <i>Mémoire plate avec recherche séquentielle</i>	9
2.5.2. <i>Mémoire hiérarchique</i>	9
2.6. MESURES DE SIMILARITÉ	10
2.7. ALGORITHMES DE CLASSIFICATION	11
2.7.1. <i>Système d'évaluation de crédit</i>	11
2.7.2. <i>Système INRECA</i>	12
2.7.3. <i>Algorithme k-NNcost</i>	12
2.7.4. <i>Représentation de la précision et du coût d'une classification</i>	13
2.7.5. <i>Algorithme du gradient conjugué pour l'apprentissage incrémental des poids d'attributs</i> .	15
2.8. ALGORITHME DES K-PLUS PROCHES VOISINS	17
2.9. ADAPTATION	20
2.9.1. <i>Le problème d'adaptation de solutions</i>	21
2.9.2. <i>L'adaptation à base de règles</i>	21
2.9.3. <i>L'adaptation à base de cas</i>	22
2.9.4. <i>Le processus d'adaptation mixte</i>	23
3 CBR ET AUTRES TECHNIQUES.....	25
3.1. LE CBR PAR RAPPORT AUX AUTRES TECHNIQUES :	25
3.1.1. <i>Le CBR et le "dépietage d'information" ("Information Retrieval")</i>	25
3.1.2. <i>Le CBR et l'apprentissage automatique ("machine learning")</i>	26
3.1.3. <i>Le CBR et les systèmes experts à base de règles</i>	27
3.1.4. <i>Le CBR et les réseaux neuronaux</i>	28
3.2. DOMAINES D'APPLICATION	28
3.2.1. <i>Le diagnostic</i>	29
3.2.2. <i>La planification</i>	30
3.2.3. <i>Le raisonnement à base de cas dans le domaine juridique</i>	31
3.2.4. <i>Le CBR pour l'aide à la conception</i>	32
3.2.5. <i>Le raisonnement par analogie</i>	34
3.2.6. <i>Le CBR dans la résolution de conflits</i>	34

3.2.7. <i>Les systèmes CBR mettant l'accent sur l'adaptation</i>	35
3.2.8. <i>Les systèmes tutoriels à base de cas</i>	36
4 CBR ET PLANIFICATION FINANCIÈRE.....	37
4.1. GESTION DE PORTEFEUILLES ET ÉTUDES DE CAS	37
4.2. EXEMPLES DE CAS.....	37
4.2.1. <i>Exemple1</i>	37
4.2.2. <i>Exemple2</i>	41
5 RÉALISATION D'UN PROTOTYPE.....	45
5.1. INTRODUCTION.....	45
5.2. PRÉSENTATION DU SYSTÈME GIPF.....	45
5.2.1. <i>Architecture du système</i>	47
5.3. APPROCHE FORMELLE POUR GIPF.....	48
5.3.1. <i>Domaine de valeurs</i>	48
5.3.2. <i>Pondération</i>	49
5.3.3. <i>Mesure de distance</i>	51
5.3.4. <i>Fonction de similarité</i>	52
5.3.5. <i>Relation entre distance et similarité</i>	53
5.3.6. <i>Voisinage</i>	53
5.3.7. <i>Classification</i>	54
5.4. EXEMPLE.....	54
5.5. REPRÉSENTATION	58
5.5.1. <i>Structure d'un cas</i>	58
5.5.2. <i>Description de la structure d'une solution associée à un cas</i>	64
5.5.2.1. <i>Structure d'une solution</i>	68
5.5.3. <i>Modèle de Mémoire pour la représentation de la base de cas</i>	69
5.5.4. <i>Mémoire cache</i>	71
5.6. INDEXATION DANS GIPF	72
5.6.1. <i>Algorithme d'indexation de cas</i>	73
5.7. RECHERCHE DE CAS	73
5.7.1 <i>Algorithme de recherche de cas</i>	74
5.8. ADAPTATION DE CAS.....	75
5.8.1. <i>Processus automatique d'adaptation de cas</i>	75
5.8.2. <i>Adaptation dans le système GIPF</i>	76
5.8.3. <i>Base de cas d'adaptation</i>	77
5.8.4. <i>Indexation des cas d'adaptation</i>	81
5.8.5. <i>Distance utilisée dans la base de cas d'adaptation</i>	81
5.8.6. <i>Règles d'adaptation de cas pour le système GIPF</i>	81
5.8.7. <i>Processus d'adaptation dans le système GIPF</i>	83
5.8.8. <i>Algorithme d'adaptation</i>	84
5.8.9. <i>Exemple d'adaptation</i> :	85
6 CONCLUSION.....	87
6.1. APPROCHE RETENUE	87
6.2. LIMITES DU SYSTÈME	88
6.3. AMÉLIORATIONS POSSIBLES.....	89
6.4. CONCLUSION	90
RÉFÉRENCES BIBLIOGRAPHIQUES.....	92
RÉFÉRENCES COMPLÉMENTAIRES	98

Liste des tableaux

TABLEAU 1 : MATRICE DE CLASSIFICATION.....	13
TABLEAU 2 : MATRICE DES COÛTS	14
TABLEAU 3 : EXEMPLE D'AVOIRS.....	38
TABLEAU 4 : COMBINAISON DE TITRES FINANCIERS.....	40
TABLEAU 5 : PREMIER EXEMPLE DE PORTEFEUILLE.....	41
TABLEAU 6 : DEUXIÈME EXEMPLE DE PORTEFEUILLE	43
TABLEAU 7 : RISQUE SELON LES CATÉGORIES D'AVOIR.....	67

Liste des figures

FIGURE 1 : PROCESSUS CBR	6
FIGURE 2 : STRUCTURE DU SYSTÈME GIPF	47
FIGURE 3 : LE SYSTÈME GIPF	47
FIGURE 4 : STRUCTURE D'UN CAS	58
FIGURE 5 : DESCRIPTION D'UN CAS	59
FIGURE 6 : TOLÉRANCE AU RISQUE D'UN CLIENT.....	59
FIGURE 7 : BILAN D'UN CLIENT.....	60
FIGURE 8 : AVOIRS FACILEMENT NÉGOCIABLES.....	60
FIGURE 9 : AVOIRS FINANCIERS NON LIQUIDES	61
FIGURE 10 : AUTRES AVOIRS.....	61
FIGURE 11 : DETTE PERSONNELLE.....	62
FIGURE 12 : DETTE COMMERCIALE.....	62
FIGURE 13 : DETTE ÉVENTUELLE.....	63
FIGURE 14 : AVOIR NET	63
FIGURE 15 : STRUCTURE D'UNE SOLUTION	68
FIGURE 16 : MODÈLE DE MÉMOIRE POUR LA BASE DE CAS	69
FIGURE 17 : BASE DE CAS	70
FIGURE 18 : MÉMOIRE CACHE	71
FIGURE 19 : PONDÉRATION ET VOISINAGE	72
FIGURE 20 : BASE DE CAS D'ADAPTATION.....	78
FIGURE 21 : EXEMPLE DE PORTEFEUILLE FINANCIER	79
FIGURE 22 : TENDANCE À LA BAISSÉ DES TAUX D'INTÉRÊT.....	79
FIGURE 23 : PORTEFEUILLE FINANCIER ADAPTÉ.....	80

1 Introduction

Depuis plusieurs années, la majorité des systèmes informatiques qui ont essayé de simuler le raisonnement humain se sont basés sur les deux outils complémentaires que sont le moteur d'inférences et la base de règles.

Ces règles de production composées de prémisses et de conclusions sont censées traduire les fondements théoriques du domaine d'application.

Quand il y avait un problème à résoudre, le système, par le biais de son moteur d'inférences essayait de sélectionner la règle dont les prémisses s'accordaient ('matching') avec la description du problème afin de l'appliquer et de générer des sous-buts qui serviraient, à leur tour de façon récursive, de critères de recherche dans la base de règles. Les systèmes experts de ce type sont plus adaptés à des domaines à fort fondement théorique. Cependant, les connaissances concernant beaucoup de domaines ne sont pas toujours évidentes à traduire en un ensemble de règles. Par ailleurs, l'ensemble de règles d'un système expert n'est pas évolutif. En d'autres termes, ce genre de système n'a pas la capacité d'enrichir ses connaissances concernant le domaine d'application dans lequel il opère.

L'approche classique des systèmes experts est une approche qui privilégie l'aspect théorique alors que l'approche des systèmes de raisonnement à base de cas (Case-Based Reasoning ou CBR) tente plutôt d'adopter une approche pragmatique, en se basant sur l'expérience développée au fur et à mesure que les problèmes sont résolus, exactement comme un expert humain développe et raffine une expertise.

Une caractéristique de l'intelligence humaine est notre capacité de repérer les analogies entre des situations différentes. Nous utilisons souvent cette capacité pour prendre des décisions. Le raisonnement à base de cas est un cas particulier du raisonnement par analogie, à travers les huit dernières années, a évolué d'une aire de recherche isolée et assez spécifique à un champ d'intérêt étendu. Les activités ont rapidement évolué avec le nombre élevé de documents de recherche et la validité de

produits commerciaux. Le CBR est une approche de résolution de problèmes dont de nombreux aspects sont fondamentalement différents des autres approches. Au lieu de faire confiance à des connaissances générales pour résoudre un problème particulier, le CBR est capable d'utiliser les connaissances d'expériences passées et les mettre en pratique dans la nouvelle situation. Une autre différence qu'il importe de souligner concerne l'aspect, approche d'apprentissage, présenté par le CBR. En effet, à chaque fois qu'un problème est résolu, le système est enrichi d'une nouvelle expérience.

Différentes études ont mis en évidence le rôle dominant des situations passées et déjà expérimentées dans la résolution de problèmes humains. Schank [Schank, 82] a développé une théorie sur le savoir et sur les bases de rappel d'expériences dans une structure de mémoire évoluée. Anderson [Anderson, 83] a montré que les gens utilisaient les cas passés comme modèles dans l'apprentissage de résolution de problèmes. D'autres résultats indiquent que l'utilisation de cas passés est une méthode de résolution dominante, chez les experts dans divers champs d'activités.

En terminologie CBR, un cas décrit habituellement une situation de problème. Une situation passée, qui a été capturée et apprise, peut être mise en application dans une future résolution de cas et sert de référence. Par exemple, un médecin qui a vu à différentes reprises, les symptômes d'une maladie particulière, la décèlerait rapidement chez un nouveau patient. Un avocat va se référer à des cas légaux déjà vus dans le passé afin de défendre un nouveau client. Un mécanicien sait reconnaître un bruit anormal causé par un moteur défectueux, car il a déjà rencontré cette anomalie par le passé.

Dans le domaine de l'analyse financière, la nature des problèmes rencontrés ne se prête pas à l'application pure et simple d'un ensemble de règles générales. En effet, les situations financières des clients sont tellement différentes et spécifiques à chacun que l'application aveugle de règles pour les traiter serait, sans aucun doute, une démarche inappropriée. En effet, l'expert financier procède à une étude de cas pour chaque nouveau client et essaie d'apporter les ajustements nécessaires au portefeuille financier, afin que ce dernier réponde aux besoins de son propriétaire de façon adéquate.

Pour répondre, même partiellement à cette problématique, nous proposons un système qui utilise le raisonnement à base de cas. L'objectif de ce travail serait d'apporter des réponses aux questions suivantes :

- . Comment représenter un cas ? En d'autres termes, quels seront les éléments descriptifs d'un cas ?
- . Quelles seraient les relations de similarité entre les cas ?
- . Comment regrouper les cas par rapport aux relations de similarité ? Comment se ferait leur classification ?
- . Comment seront stockés les cas en mémoire ?
- . De quelle façon procédera-t-on afin de retrouver des cas similaires préalablement archivés dans une base ?
- . Comment modifier une ancienne solution afin de prendre en ligne de compte les différences présentées par le cas courant ?

Présentation du mémoire

Le chapitre 2 présente le raisonnement à base de cas. Nous introduisons quelques notions relatives à un système CBR, comme la définition du vocabulaire utilisé dans la description et l'indexation des cas. Nous traitons, ensuite, des mesures de similarité où nous parlons de pondération des attributs décrivant un cas. Nous passons également en revue quelques algorithmes de classification. Nous présentons enfin quelques systèmes connus dans différents domaines d'application.

Avant de clore ce chapitre, nous traitons des problèmes que présente le processus d'adaptation de solutions dans les systèmes de raisonnement à base de cas.

Le chapitre 3 fait une étude comparative du raisonnement à base de cas par rapport à d'autres approches que nous présenterons. Il passe également en revue les différents domaines d'application possibles pour les systèmes CBR.

Dans le chapitre 4, nous abordons des exemples de cas dans le domaine de l'analyse de situations financières.

À travers le chapitre 5, nous introduisons les différents types de titres financiers qui seront les principales composantes d'un portefeuille financier. Ensuite, nous passerons à la description de l'implantation de notre prototype baptisé système "G.I.P.F" pour "Gestion Intelligente de Portefeuille Financier". Nous y expliquons également les structures définies pour l'organisation de la mémoire de cas, la mesure de similarité ainsi que l'indexation. Nous exposons par la suite, l'approche retenue par notre système pour le processus d'adaptation de solutions.

Le 6eme et dernier chapitre présente, les conclusions de ce travail ainsi que les perspectives futures et les améliorations possibles de notre prototype, le système GIPF.

2 Raisonnement à base de cas

2.1. Principes de base

Un raisonnement à base de cas est un processus cyclique de résolution de problèmes dont le fonctionnement est basé sur la réutilisation de solutions associées à des problèmes passés, pour la résolution de nouveaux problèmes similaires.

En terminologie CBR, un cas dénote une situation de problème. Le terme résolution de problème est utilisé dans son sens le plus large. Cela peut consister à confirmer ou infirmer une solution proposée par l'utilisateur, à interpréter une situation de problème ou à générer un ensemble de solutions possibles.

Le processus de raisonnement dans un système de raisonnement à base de cas est composé des étapes suivantes :

- Extraction d'anciens cas similaires au nouveau cas à traiter.
- Sélection du meilleur candidat qui va servir de base au raisonnement.
- Proposition d'une solution après adaptation.
- Évaluation des résultats.

La figure suivante illustre le déroulement du raisonnement à base de cas.

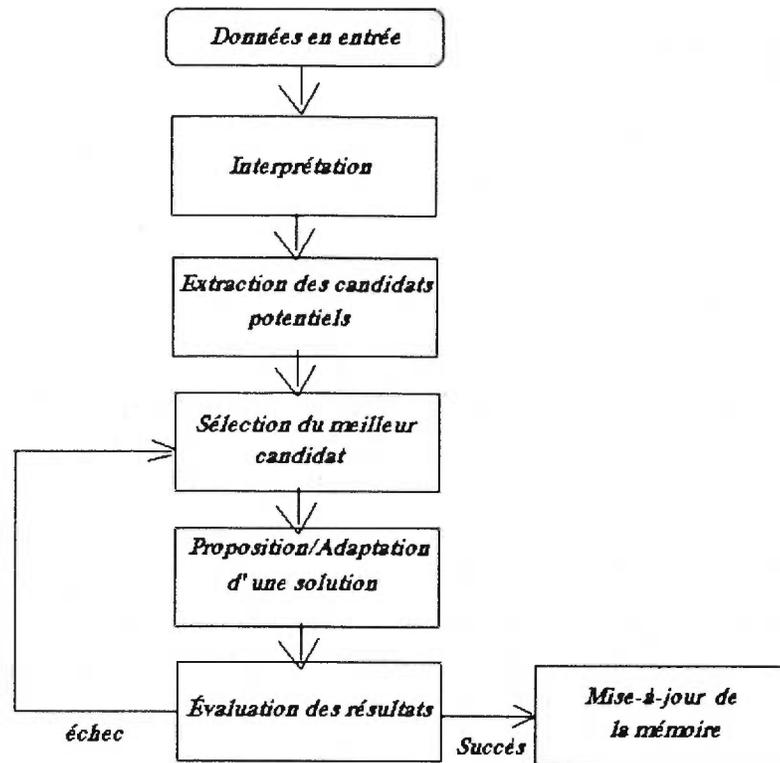


Figure 1: Processus CBR

2.2. Sensibilisation de l'industrie au CBR

Tout au long de l'existence d'une compagnie, des quantités considérables d'informations sont mémorisées dans les bases de données des systèmes informatiques. Cependant, lors de l'élaboration d'un diagnostic par un professionnel ou lors de la résolution d'un problème par un expert, l'existence de cette mémoire n'est pas mise à contribution. D'autre part, cette mémoire est conservée sous une forme qui rend son utilisation difficile, voir même impossible, pour résoudre de nouveaux problèmes.

2.3. Historique

Le raisonnement à base de cas est issu de la recherche des sciences cognitives. Le travail de Schank et Abelson en 1977 [Schank et Abelson, 77] est largement perçu comme

le fondement du CBR. La notion de script y est introduite pour la première fois. Le cerveau enregistre les informations décrivant une situation sous forme de script. Les scripts sont proposés comme structure pour la mémoire conceptuelle. Ils décrivent les informations relatives à des événements comme rendre visite au médecin ou aller au restaurant. Cependant l'expérimentation des scripts a montré qu'ils ne sont pas une théorie complète de représentation pour la mémoire. En effet, les gens confondent souvent les événements dont les scripts sont similaires. Par exemple, on peut faire la confusion entre des étapes qui sont dans le script d'une visite chez le médecin avec celles du script d'une visite chez le dentiste.

Schank a continué à explorer le rôle joué par la mémorisation de situations passées ou les MOPs [Schank, 82] (Memory Organization Packets) dans la résolution de problèmes et l'apprentissage.

Alors que les fondements philosophiques du raisonnement à base de cas peuvent être attribués à beaucoup de monde, il ne fait aucun doute que c'est le travail de pionnier de l'équipe de Schank au début des années 80 qui a donné naissance au modèle cognitif du CBR ainsi qu'aux toutes premières applications basées sur ce modèle. En 1983, Janet Kolodner [Kolodner, 83] a développé CYRUS qui fut le premier système CBR. CYRUS était une implémentation du modèle de mémoire dynamique de Schank. Son modèle de mémoire de cas servit par la suite de base à de nombreux autres systèmes comme MEDIATOR [Simpson, 85], CHEF [Hammond, 86], PERSUADER [Sycara, 87], CASEY [Koton, 89] et JULIA [Hinrichs, 89].

Une approche alternative fut développée par Bruce Porter [Porter et Bareiss, 86], à l'université du Texas, où l'on traite aussi bien de l'apprentissage ("machine learning") [Laird et *al.*, 86], que de l'utilisation d'heuristiques pour la classification. Ce travail a donné naissance au système PROTOS. Ce dernier présente un modèle de mémoire de cas dans lequel on trouve une combinaison des connaissances liées au domaine général avec des connaissances spécifiques à des situations particulières.

Vers la fin des années 80, le programme DARPA, aux États Unis a sponsorisé une série d'ateliers ("workshop") sur le CBR et le développement d'outils CBR. Ces outils ont donné naissance au système cognitif ReMind et ont marqué la transition de cette discipline

du domaine de recherche purement académique en sciences cognitives et intelligence artificielle vers le domaine commercial [Watson, 97].

2.4. Vocabulaire et Indexation

Afin de formaliser toutes les descriptions possibles des différents objets et concepts intervenant dans le domaine d'application, ainsi que leurs interrelations, nous procédons à la construction d'un vocabulaire d'indexation. Un index doit nous permettre de différencier les cas, tout en saisissant au maximum leurs similarités. Les index représentent donc les différentes facettes décrivant un cas.

Un vocabulaire d'indexation doit être pertinent, c'est-à-dire capable de décrire les concepts intervenant dans le domaine d'application. Il doit également être assez général, pour regrouper les cas en classes. D'autre part, le vocabulaire doit être assez spécifique afin de présenter une bonne discrimination des cas.

2.5. Organisation de la mémoire

Dans cette section, nous présentons différentes organisations possibles de la mémoire des cas avec les avantages et les inconvénients de chacune. En général, l'efficacité du processus de recherche des cas dépend de la structure de la mémoire ainsi que de la complexité de la similarité entre les cas. Le but poursuivi par les stratégies de recherche est la sélection d'un nombre réduit de cas qui se rapprochent du problème courant.

La structure organisationnelle d'une mémoire et les algorithmes de recherche qui en découlent dépendent fortement du nombre de cas stockés dans la base et de la complexité des index [Kolodner, 93].

Quelque soit la méthode utilisée, l'opération qui retrouve les cas est une combinaison du processus de recherche avec le processus de traitement de la similarité. La recherche dans la structure de mémoire retourne un ensemble de cas potentiellement

proches du cas problème. Le processus de traitement de la similarité se charge par la suite de les traiter un par un afin de retenir celui qui se rapproche le plus du cas à traiter. En général, certains attributs décrivant un cas ont plus d'importance que d'autres. Donc un bon algorithme de recherche de cas doit tenir compte de ce critère qu'est l'importance accordée à chaque caractéristique et doit s'en servir afin de faire correspondre les cas.

2.5.1. Mémoire plate avec recherche séquentielle

Dans une mémoire plate, les cas sont stockés séquentiellement dans une simple liste, un tableau ou un fichier. La recherche de cas se fait en parcourant séquentiellement la liste et en évaluant la similarité entre chaque élément et le problème courant. Finalement, le cas le plus proche sera retenu. Dans ce type de mémoire, l'organisation est réduite à sa plus simple expression et l'algorithme de recherche est élémentaire. En fait, le processus qui traite la similarité se charge de tout le travail. L'avantage de ce type de mémoire c'est l'assurance que l'on a concernant le parcours de tous les cas. L'efficacité de la recherche dépend donc, uniquement de la précision de la fonction de similarité. Par conséquent, si la fonction de similarité est bonne, alors le meilleur cas sera toujours retrouvé. Un autre avantage de cette organisation concerne la facilité avec laquelle nous pouvons rajouter de nouveaux cas.

D'autre part, ce type de structure présente également des inconvénients, ce système est évidemment coûteux en terme de temps. Effectivement, le temps de recherche en mémoire croît dramatiquement avec l'augmentation du nombre de cas. Cette organisation de mémoire s'adapte donc beaucoup plus à une petite base qu'à une base renfermant un nombre important de cas.

2.5.2. Mémoire hiérarchique

Une mémoire comportant un nombre important de cas est organisée en structure hiérarchique afin d'optimiser le processus de recherche. On dénote deux types de structures hiérarchiques. Le premier est une mémoire organisée en réseau à caractéristique

partagée. Les cas similaires y sont regroupés en sous-ensembles. Devant une nouvelle situation de problème, on essaie de retrouver le sous-ensemble de cas le plus similaire. L'analyse se fait par la suite dans ce sous-ensemble afin de trouver le cas le plus proche du problème courant. Ce découpage de la mémoire réduit donc considérablement le nombre de cas à analyser. La hiérarchie se construit au fur et à mesure que les sous-ensembles éclatent en fonction de leur similarité vis à vis de nouvelles caractéristiques. Un même cas peut figurer au sein de plusieurs sous-ensembles différents, on peut donc avoir une redondance dans la mémorisation.

Le second type d'organisation hiérarchique est l'organisation en réseau de discrimination. Dans ce genre de structure, les cas sont regroupés en se basant sur une discrimination qui tient compte de la pondération accordée à chaque attribut décrivant un cas. Les cas sont donc regroupés suivant des caractéristiques à fort pouvoir de discrimination.

2.6. Mesures de similarité

Un système CBR a besoin d'évaluer la similarité existant entre deux cas. Cela lui permet de décider si l'erreur faite en choisissant un cas, au lieu d'un autre, est acceptable. La difficulté lors de l'implantation d'un système CBR pour un domaine d'application particulier, comprend la définition par un expert du type de chaque champ ainsi que la définition d'une mesure de similarité pour les instances de chacun de ces champs.

Dans la détermination de la mesure de similarité, on procède, entre autres, à la détermination du poids de chaque champ intervenant dans la description d'un cas. Ceci peut se faire de deux façons différentes :

- Soit, c'est l'expert qui fournit les pondérations,
- Soit, c'est le système qui utilise un algorithme d'apprentissage. Ceci se fera dans le cas où l'expert ne peut affecter de poids aux champs de façon précise. Donc, quand les poids donnés par l'expert ont besoin d'être validés.

Une classe particulière d'algorithmes regroupe ceux qui ajustent la pondération des champs, en se basant sur le retour (feed-back) recueilli lors des pondérations testées

auparavant. Le système EACH ajuste la pondération en incrémentant le poids des champs qui ont permis une classification correcte et en décrémentant le poids des champs à l'origine d'une mauvaise classification [Wilke et Bergmann, 96].

2.7. Algorithmes de classification

Différentes approches sont utilisées dans les systèmes VDM [Stanfill et Waltz, 86], IB4 [Aha,91], PATDEX/INRECA [Wess et Ritcher, 93]. Ces systèmes présentent des différences dans la manière de modifier la pondération des champs, tout en partageant le même but, qui consiste à ajuster et corriger la classification. Ces algorithmes mettent donc l'accent sur l'amélioration de la précision du processus de classification.

Cependant, dans beaucoup de domaines d'application, l'accroissement de la précision de classification ne représente pas un but en soit. C'est plus, le coût d'une décision qui joue un rôle majeur. Le but serait donc, d'évaluer le coût d'une bonne classification par rapport à celui d'une classification inappropriée.

Dans les sections suivantes, nous allons présenter quelques exemples de systèmes afin de souligner l'importance des conséquences qui peuvent résulter d'une mauvaise classification. Ensuite, nous présentons des algorithmes qui ajustent automatiquement cette pondération afin de réduire le risque d'erreur de classification. Dans la section qui suit, le travail de Wilke et Bergmann [Wilke et Bergmann, 96] a servi de référence et ne sera plus cité par la suite.

2.7.1. Système d'évaluation de crédit

Afin d'illustrer cela, prenant l'exemple d'un système qui évalue la crédibilité des compagnies afin de permettre à la banque de leurs accorder un crédit. Pour ce faire, le système va ranger dans la classe A, les compagnies qui présentent une bonne santé

financière et dont la demande de prêt est acceptée. La classe B va renfermer les compagnies dont les données financières ne sont pas bonnes et donc dont le prêt va être rejeté par la banque. Si jamais le système, par erreur, classe une compagnie dans la classe B au lieu de la mettre dans la classe A, le coût de cette mauvaise classification se traduirait par la perte des intérêts pour la banque, du moment que le prêt serait refusé. Alors que si le système classait comme classe A, une compagnie qui est, en réalité dans la classe B, le coût pour la banque serait beaucoup plus important car elle perdrait le montant total du prêt. Ici, le coût d'une mauvaise décision dépend fortement de la pertinence de la classification d'un cas.

2.7.2. Système INRECA

Le système INRECA [Althoff et *al*, 95], est un système CBR de diagnostic des cas d'empoisonnement provoqués par la prise de psychotropes. Le système propose huit sortes de thérapies possibles pour traiter l'empoisonnement. Les thérapies ont différents effets sur la santé des patients. Dans le pire des cas, une thérapie inadéquate peut être mortelle pour le patient. Cependant, pour une catégorie particulière de diagnostics, les thérapies sont identiques ou, à la limite, très similaires. Donc, proposer un mauvais diagnostic qui débouche sur la même thérapie (ou presque) que celle associée au bon diagnostic, ne pose pas de problèmes. Le coût de décision dans cette application est nettement différent de celui de l'application qui évalue les crédits.

Les exemples vus, précédemment, mettent en évidence l'importance du rôle du coût de décision dans l'optimisation des systèmes CBR.

2.7.3. Algorithme $k\text{-NN}_{cost}$

L'algorithme d'apprentissage $k\text{-NN}_{cost}$, optimise les poids des différents attributs d'un cas en fonction du coût et du profit occasionné par les décisions prises par un système de classification CBR. En d'autres termes, pour chaque classification effectuée

par le système, il y a une évaluation du coût occasionné par une mauvaise classification ainsi que l'évaluation du bénéfice occasionné par une bonne classification. Des exemples sont présentés par la suite afin d'illustrer cela.

2.7.4. Représentation de la précision et du coût d'une classification

Une représentation courante de la précision de la classification est la "confusion matrix". Pour un ensemble de cas donné, l'entrée d'une telle matrice compte le nombre de classifications faites par le système, par rapport à la classification réelle de ces cas.

L'approche retenue ici, introduit une modification des entrées de la "confusion matrix", qui tient compte de la probabilité $P_{q,t}$ de classer le cas q comme membre de la classe t . La matrice contient donc, la probabilité pour chaque décision possible du système, pour un ensemble de cas donné.

Afin d'évaluer la précision de classification à partir d'un ensemble test de cas L , on peut essayer de classer chaque cas $q \in L$, en utilisant l'ensemble $L - \{q\}$. Pour chaque cas q , on calcule les probabilités $P_{q,t}$ pour les k voisins les plus proches et ceci pour chaque classe $t \in T$. Les probabilités obtenues sont combinées aux entrées de la "confusion matrix".

La matrice obtenue représente les probabilités pour les différents résultats de classification pour l'ensemble d'apprentissage L . Cette matrice est une approximation de la "confusion matrix", car elle contient les probabilités de décisions (lors de la classification) au lieu du nombre d'occurrences de chaque décision.

La table suivante montre un exemple de matrice pour l'application qui accorde des crédits à des compagnies en les classant dans une des deux catégories A ou B.

$P_{i,j}$	Classe proposée par le système	
	A	B
A	0.55	0.45
B	0.25	0.75
Classe correcte		

Tableau 1 : Matrice de classification

Dans la diagonale de la matrice, on trouve les probabilités pour qu'une classe proposée par le système soit correcte. Dans notre exemple, ces probabilités sont de 55% pour la classe A et de 75% pour la classe B. Les autres entrées de la matrice représentent les probabilités d'erreurs produites lorsque le système fait une mauvaise classification.

Pour représenter les coûts, on utilise une matrice de coûts similaire à celle vue précédemment [Wilke et Bergmann, 96]. Dans cette matrice, les entrées non diagonales représentent les coûts pour une mauvaise classification alors que les entrées dans la diagonale représentent les profits tirés d'une classification réussie. Les profits et les coûts seront représentés par des valeurs positives ou négatives. Cette matrice représente donc, les coûts et les profits liés à chaque décision possible prise par le système.

La table suivante représente un exemple de matrice de coûts dans l'application des crédits accordés aux compagnies.

$C_{i,j}$	Classe proposée par le système	
	A	B
A	-1	1
B	10	-10
Classe		

Tableau 2 : Matrice des coûts

Les valeurs de la matrice représentent la relation entre les deux types d'erreurs possibles. Ici, on peut voir que le coût d'une mauvaise classification d'un mauvais client est 10 fois plus élevé que celui d'une mauvaise classification pour un bon client. Dans le premier cas, la banque perdrait le montant du crédit accordé au client, alors que dans le deuxième cas, elle perdrait juste les intérêts. Donc le profit réalisé par la détection d'un mauvais client est dix fois plus grand que celui réalisé lors d'une bonne classification.

A partir de la "confusion matrix" et de la matrice des coûts, on peut définir le coût d'une décision pour un système de classification comme suit :

$$\sum \sum P_{i,j} * C_{i,j} \quad \text{avec } i \in T \text{ et } j \in T$$

Les $P_{i,j}$ sont les probabilités de la "confusion matrix" et les $C_{i,j}$ sont les entrées correspondantes dans la matrice des coûts.

Dans cette section, nous avons présenté une approche possible pour le processus de classification afin de souligner l'importance de cette tâche dans un système de raisonnement à partir de cas. Nous présenterons dans le chapitre 5, l'approche retenue par le système que nous avons implementé.

2.7.5. Algorithme du gradient conjugué pour l'apprentissage incrémental des poids d'attributs

L'algorithme du gradient conjugué tente d'optimiser un système en ajustant les poids $w = \{ w_1, w_2, \dots, w_n \}$, tout en tenant compte d'une fonction d'erreur E . C'est un processus itératif qui recherche un minimum local pour la fonction d'erreur E . Une étape d'apprentissage λ est utilisée afin de fixer la taille de chaque intervalle dans le processus d'apprentissage en direction du gradient conjugué. Avant de présenter l'algorithme de base, nous donnons ci-dessous, la définition des différents symboles qui y figurent :

- w est le vecteur des poids affectés aux attributs qui décrivent un cas.
- w_a est le poids d'un attribut a en particulier.
- $E(w)$ correspond à l'erreur que l'on commet si on affecte aux attributs, le vecteur de poids w
- δE représente la variation de l'erreur $E(w)$.
- δw_a correspond à la variation du poids de l'attribut a .
- $E(\varpi)$ est le nouveau vecteur de poids obtenu après modification des poids de tous les attributs
- λ représente l'étape d'apprentissage.

L'algorithme de base est présenté ci-après :

1. Initialiser le vecteur des poids w et étape d'apprentissage λ .
2. Calculer l'erreur $E(w)$ du système initial.
3. Tant que non(critère d'arrêt) faire
 - Étape d'apprentissage : $\forall a \ \varpi := w_a - \delta E / \delta w_a * \lambda$
 - Calculer $E(\varpi)$
 - si $E(\varpi) < E(w)$ alors $w := \varpi$ sinon $\lambda := \lambda/2$ finsi
4. Afficher w .

Les poids initiaux sont initialisés à des valeurs aléatoires ou alors, prennent la valeur $1/n$. Ils peuvent également être donnés par un expert. Après avoir calculé la valeur initiale de E , l'algorithme exécute plusieurs étapes d'apprentissage, en fonction du critère d'arrêt. Si une étape d'apprentissage a contribué à réduire l'erreur $E(\varpi) < E(w)$, les poids sont modifiés pour faire tendre la fonction E vers un minimum local. Si on ne réussit pas, l'étape d'apprentissage λ est réduite. A la fin, l'algorithme renvoi le vecteur des poids finaux.

Le choix de l'étape d'apprentissage λ n'est pas évident, car elle dépend de plusieurs facteurs liés au domaine d'application, qui ne sont pas forcément connus. Si λ est trop petit, il faudrait plusieurs étapes d'apprentissage pour trouver le prochain minimum local dans le voisinage du point de départ. L'algorithme le trouvera en respectant le point de départ initial (le vecteur de poids initiaux).

Dans l'autre cas, si λ est trop grand, l'algorithme peut sauter des étapes et sera obligé de décrémenter la valeur de λ , avant de réduire l'erreur E . Il peut également arriver qu'un λ trop grand aboutisse à une amélioration de l'erreur E , mais cette amélioration serait basée sur minimum local qui ne tient pas compte du vecteur de poids initiaux. Ce

minimum local pourrait donc être plus grand qu'un autre minimum local qui serait beaucoup plus proche du vecteur de poids initiaux et que l'algorithme aurait raté en raison de l'étape d'apprentissage qui serait trop grande (les poids initiaux ayant, dans ce cas là, subi une modification trop grande).

L'algorithme s'arrête en général, une fois que la réduction de l'erreur devient de plus en plus petite entre deux étapes d'apprentissage consécutives. Donc même si on exécute encore une étape d'apprentissage supplémentaire, l'erreur n'en serait pas sensiblement améliorée. Le choix du critère d'arrêt peut être un (ou une combinaison) des points suivants :

- Un nombre fixe d'étapes d'apprentissage.
- Une variation minimale de la fonction d'erreur : $|E(\varpi) - E(w)| \leq \epsilon$
- Une variation minimale des poids : $\sum_{a=1,n} |\delta E / \delta w_a| \leq \epsilon$
- Une étape d'apprentissage assez petite : $\lambda \leq \epsilon$

À titre d'exemple, on pourrait choisir, comme critère d'arrêt, une variation minimale de la fonction d'erreur et fixer une valeur $\epsilon = (5\%, 4\%, 7\%, 3\%)$. Si le vecteur d'erreur $E(w)$ a la valeur $(10\%, 20\%, 12\%, 9\%)$ et qu'une itération dans la boucle retourne comme nouvelle valeur $E(\varpi) = (9\%, 17\%, 8\%, 7\%)$, l'algorithme s'arrête.

Nous constatons, effectivement que le critère d'arrêt est vérifié avec la condition $|E(\varpi) - E(w)| = (1\%, 3\%, 4\%, 2\%) < \epsilon$.

2.8. Algorithme des k-plus proches voisins

Pour résoudre un nouveau problème, l'algorithme des k-plus proches voisins tente d'extraire les candidats potentiels à partir de la base de cas. Plus précisément, cet algorithme cherche à déterminer la classe qui présente la plus haute probabilité pour être celle à laquelle appartient le cas courant, c'est-à-dire celle du problème à résoudre.

Les étapes de la recherche du plus proche voisin sont les suivantes :

- Associer une pondération à chaque attribut décrivant un cas.
- Chaque attribut du cas à traiter est comparé à l'attribut correspondant dans le cas retrouvé dans la base de cas.
- Établir le degré de similarité entre les attributs , en tenant compte du poids affecté à chacun d'eux.
- Calculer la somme des degrés de similarité pondérés pour chaque attribut, afin d'obtenir un indicateur combiné de similarité.

Plusieurs systèmes CBR implémentent l'algorithme du voisin le plus proche (MEDIATOR [Simpson, 85], PERSUADER [Sycara, 87], REMIND [Watson, 97], etc.). Dans le cas du système REMIND, le degré de similarité est compris entre 0 et 1.

De façon plus formelle , un cas c est défini comme un vecteur d'attributs :

$c = (f_1, \dots, f_n, t_c)$, avec n descriptions d'attributs et une classe t_c à laquelle le cas appartient.

Soit T l'ensemble des classes possibles :

$$T = \{ t_1, \dots, t_n \}$$

La base de cas CB est définie comme l'ensemble des cas les plus représentatifs pour un domaine d'application donné.

Soit un nouveau cas $q = \{q_1, \dots, q_n\}$ et une base de cas CB , les k cas les plus similaires au nouveau cas q vont déterminer sa classe t_q .

La similarité $sim(q, c)$ entre un cas q et un cas c de la base cas, est définie comme suit :

$$sim(q, c) = \sum_{a=1, n} w_a * sim_a(q_a, f_a)$$

avec w_a : poids de l'attribut f_a

et sim_a : La mesure de similarité locale pour l'attribut a .

On suppose donc que :

$$\sum_{a=1, n} w_a = 1 \text{ et } sim(q_a, f_a) \in [0, 1]$$

On a donc : $sim(q,c) \in [0,1]$

Le système va trouver la classe du nouveau cas q en retrouvant ses k voisins les plus proches et leur appliquant la méthode du vote majoritaire.

Soit $K = \{r_1, \dots, r_k\}$ l'ensemble de ces cas voisins.

Soit $P_{q,t}$, la probabilité pour que le cas q soit membre de la classe $t \in T$:

$$P_{q,t} = \sum_{r \in K} \delta_{r,t} * sim(q,r) / \sum_{r \in K} sim(q,r)$$

$$\text{avec: } \delta_{r,t} = \begin{cases} 1 & \text{Si } t_r = t; \\ 0 & \text{Si } t_r \neq t \end{cases}$$

$t_r \in T$: Classe du cas r .

Le système CBR propose donc, la classe présentant la plus haute probabilité calculée à partir de l'ensemble des k plus proches voisins.

Il existe d'autres algorithmes de vote pour la classification basée sur les k voisins les plus proches. Par exemple, le vote à majorité simple qui tient compte uniquement des fréquences d'apparition des différentes classes dans l'ensemble des k voisins les plus proches. Cependant, le vote à majorité pondérée a l'avantage de prendre en considération la distance séparant les cas voisins du cas courant.

L'exemple suivant illustre le fonctionnement de l'algorithme des k -plus proches voisins. Si, pour un nouveau cas q , l'algorithme trouve deux cas proches s_1 et s_2 avec les degrés de similarité suivants :

$$sim(s_1, q) = 50\%$$

$$sim(s_2, q) = 40\%$$

Si $classe(s_1) = t_1$ et $classe(s_2) = t_2$, on a :

$$\delta_{s_1, t_1} = 1$$

$$\delta_{s_1, t_2} = 0$$

$$\delta_{s_2, t_1} = 0$$

$$\delta_{s2,t2} = 1$$

Pour déterminer la classe du cas q , on procède au calcul de probabilités suivant :

$$P_{q,t1} = (1 * 50\% + 0 * 40\%) / (50\% + 40\%)$$

$$P_{q,t1} = 0.55$$

La probabilité pour que la classe du cas q soit $t2$:

$$P_{q,t2} = (0 * 50\% + 1 * 40\%) / (50\% + 40\%)$$

$$P_{q,t2} = 0.44$$

En conclusion, la probabilité pour que $t1$ soit la classe du nouveau cas q est la plus élevée.

On obtient donc :

$$classe(q) = classe(s1) = t1$$

2.9. Adaptation

Comme nous l'avons vu précédemment, les systèmes CBR résolvent les problèmes, en réutilisant les solutions correspondant à des problèmes passés similaires. Étant donné que le cas retrouvé est similaire au problème courant et pas exactement identique, il faut adapter les solutions retrouvées afin qu'elles correspondent à la variation présentée par le nouveau problème à résoudre.

Par exemple, pour deux individus qui sont dans la même tranche d'âge, on pourrait ajuster leurs tolérances aux risques dans le domaine de l'investissement financier, pour refléter le fait que la personne qui est plus âgée accepte, en général, une plus faible tolérance aux risques.

2.9.1. Le problème d'adaptation de solutions

Malgré les progrès considérables réalisés dans les techniques utilisées lors de la recherche de cas, la partie adaptation de solution reste un problème ouvert [Allemang, 93]. Le problème d'adaptation de solution est si pointu que, jusqu'à présent, la majorité des applications CBR se sont bornées à contourner entièrement ce problème, en développant des systèmes agissant comme conseillers tout en laissant à l'utilisateur, la charge d'adapter lui-même les solutions proposées [Barletta, 94].

Malgré les avantages pratiques de ce genre de systèmes, une utilisation efficace exige une grande expertise liée au domaine d'application, de la part de l'utilisateur. L'adaptation de cas est importante aussi bien pour permettre aux CBR d'être autonomes que pour aider les utilisateurs néophytes.

2.9.2. L'adaptation à base de règles

Formaliser des règles d'adaptation nécessite une connaissance approfondie des tâches que le système CBR doit effectuer, son domaine d'application et le contenu de sa mémoire. Malheureusement, cette connaissance n'est pas toujours disponible. Ainsi, dans la définition des règles d'adaptation de cas, les développeurs font face au même problème que celui lié à l'acquisition de connaissances dans des domaines à faible base théorique. Ces domaines ne s'adaptent donc pas au développement de systèmes à base de règles. Par exemple, dans le raisonnement légal, les lois étant ambiguës et imprécises, la résolution de problème est fondée sur la comparaison avec des cas appelés des précédents.

L'adaptation à base de règles peut se faire suivant plusieurs stratégies :

- **Adaptation par substitution** : Elle consiste à remplacer une valeur d'attribut jugée inadéquate dans la nouvelle situation de problème.

- **Adaptation par transformation :** Elle consiste à ajouter, supprimer ou remplacer un attribut.

- **Adaptation par dérivation :** C'est une séquence d'inférences qui est appliquée au nouveau cas afin de dériver une nouvelle valeur pour un attribut.

Dans plusieurs de ces champs d'application, le problème de l'acquisition de connaissances a été sensiblement amélioré par l'utilisation du raisonnement à base de cas. Il serait donc naturel d'appliquer le CBR au processus d'adaptation de cas lui-même, en remplaçant des règles d'adaptation prédéfinies par des adaptations de cas qui reflètent les expériences d'adaptation passées [Berger, 95], [Leake, 94]. Cela reviendrait donc à faire du raisonnement à base de cas récursif.

2.9.3. L'adaptation à base de cas

Nous allons présenter une approche qui aborde l'adaptation de cas en la décomposant en deux parties. La première est un petit ensemble de transformations et la seconde représente des stratégies de recherche en mémoire afin de retrouver l'information nécessaire à l'application de ces transformations.

Ce procédé présente la base d'une approche dans laquelle les nouvelles adaptations de problèmes sont résolues en commençant par sélectionner une transformation qui a été mémorisée et pour laquelle on a créé un index construit sur le type de problème. Ensuite on définit une stratégie de recherche en mémoire de l'information nécessaire à l'application de cette transformation [Leake, 95].

Cette approche présente une base d'apprentissage pour améliorer les aptitudes d'adaptation au fil des expériences. En effet, une trace de ce processus peut être stockée

comme une adaptation de cas et utilisée ultérieurement dans un CBR relatif au processus d'adaptation lui-même.

Par exemple, une transformation qui modifierait la structure d'un portefeuille financier pourrait se faire en se basant sur la différence qui existe au niveau des objectifs financiers des clients. Cette transformation pourrait donc, être stockée en mémoire, en l'indexant sur la différence entre les objectifs financiers des clients.

2.9.4. Le processus d'adaptation mixte

Ce modèle combine une approche à base de cas avec un raisonnement à base de règles afin d'élaborer une expertise dans l'adaptation de cas [Leake et *al*, 96]. L'objectif de cette approche est de rendre les systèmes CBR capables d'assurer la transition de l'adaptation guidée par des règles générales vers l'adaptation guidée par un raisonnement à base de cas.

Le processus d'adaptation va suivre les étapes suivantes:

1- **Adaptation à base de cas:** Le système va essayer de retrouver un cas d'adaptation qui a bien fonctionné dans le passé pour un problème similaire. Si le cas existe, il sera appliqué de nouveau et le contrôle passe à l'étape 3.

2- **Adaptation à base de règles:** Si aucun cas passé n'est retrouvé, le système sélectionne une règle de transformation associée au type de problème à traiter. Il va, par la suite, essayer de remplir les conditions requises afin que la règle de transformation (voir 2.9.2) soit applicable.

3- **Évaluation :** La solution adaptée va être évaluée par un module qui vérifie la compatibilité entre ses contraintes et celles de la solution de départ.

4- **Mémorisation** : Quand l'adaptation a réussi, la solution ainsi que le cas d'adaptation sont mémorisés pour une utilisation future.

Dans le chapitre suivant, nous présentons une étude comparative du raisonnement à base de cas par rapport aux autres techniques. Nous passons également en revue plusieurs systèmes CBR évoluant dans différents domaines d'application.

3 CBR et autres techniques

3.1. Le CBR par rapport aux autres techniques :

L'objectif de cette section est de souligner les faiblesses et les forces du raisonnement à base de cas par rapport à d'autres techniques comme les systèmes à base de règles, le "dépistage d'information" ou IR ("Information Retrieval") [Rau, 87], l'apprentissage automatique ("Machine Learning") [Laird et *al.*, 86] et les réseaux neuronaux.

3.1.1. Le CBR et le "dépistage d'information" ("Information Retrieval")

Le raisonnement à base de cas et "dépistage d'information" ont plusieurs points en commun. En effet l'IR peut utiliser des techniques plus sophistiquées que les requêtes de base de données classiques quand il s'agit de retrouver des données à partir de sources d'informations de grandes tailles, comme les disques compacts (CD) et le réseau Internet. Une technique, pour retrouver de l'information de type texte en particulier, consiste à traiter les mots clés présents dans une requête de recherche afin de retrouver des mots similaires dans la base d'informations, ceci dans le but d'élargir le champ de recherche de cette requête. Par exemple, une requête comme "remède pour rhume" devrait inclure des mots comme traitement et grippe, qui sont synonymes avec les mots remède et rhume.

Vus sous cet angle, le CBR et l'IR offrent tous deux, la possibilité de traiter des requêtes un petit peu floues et retournent des résultats potentiellement corrects avec une marge d'erreur. Cependant, ces deux techniques présentent des différences :

. Alors que les systèmes d'IR tendent à se spécialiser dans le traitement de documents texte dans des bases de données de grandes tailles, le CBR peut utiliser une gamme plus large de types de données, comme les valeurs numériques, les symboles et les booléens.

. Les systèmes d'IR n'exploitent pas les connaissances liées aux informations qui sont traitées, contrairement aux systèmes CBR qui utilisent ces informations pour, par exemple, attribuer une pondération aux dimensions décrivant un cas.

Cette différence tend à s'estomper car le CBR est de plus en plus utilisé pour retrouver de l'information à partir de bases de données textes de grandes tailles. À titre d'exemple, Microsoft utilise un moteur de recherche d'information de type CBR, au niveau du système d'aide dans le produit Microsoft Office pour Windows 95 [Watson, 97].

3.1.2. Le CBR et l'apprentissage automatique ("machine learning")

Il peut sembler bizarre de vouloir comparer ces deux approches quand on voit que les algorithmes d'induction utilisés par le CBR sont dérivés de la recherche du domaine de l'apprentissage ("Machine Learning"). Cependant, il y a une différence dans la façon d'appliquer ces techniques. Un système d'apprentissage analyse les cas passés afin de dériver les règles qu'il appliquerait afin de résoudre de nouveaux problèmes. Ici, on sépare de façon claire, le processus d'apprentissage de règles, du processus de résolution de problèmes. En revanche, le CBR utilise les algorithmes d'induction afin de procéder à la classification des cas. Le résultat de ce processus est un arbre d'indexation qui servirait à faire la correspondance entre les nouveaux cas et les cas préalablement stockés. Ici, la différence entre le processus d'apprentissage et le processus de résolution de problèmes est beaucoup moins marquée. Une autre différence à souligner se trouve au niveau de la justification de la solution proposée par le système. Un système d'apprentissage automatique va justifier une décision en exhibant les règles induites d'un exemple d'apprentissage "training example" précédent. Mais le système a perdu toute trace d'un

exemple d'apprentissage en particulier alors que le CBR va utiliser l'ancien cas comme référence pour appuyer sa décision. L'importance de cette différence réside dans le fait que les gens, en général, sont plus facilement convaincus par une justification qui se réfère à un exemple concret plutôt qu'à une règle abstraite [Watson, 97].

3.1.3. Le CBR et les systèmes experts à base de règles

Un système à base de règles découpe la résolution d'un problème en un ensemble de règles individuelles qui résolvent chacune une partie de ce dit problème. La création de ces règles est une tâche ardue et complexe car elle nécessite la maîtrise du savoir faire quant à la résolution des problèmes. C'est là que les systèmes CBR diffèrent fondamentalement des systèmes à base de règles, car ils n'ont pas à savoir comment résoudre ces problèmes mais juste à déterminer s'ils ont résolu un problème similaire dans le passé. Bien que les systèmes à base de règles et les systèmes CBR soient souvent utilisés pour résoudre des problèmes similaires, comme le diagnostic, le CBR présente les avantages suivants :

- . Économie de l'effort d'établissement des règles qui nécessite l'intervention d'un expert. En effet, pour construire un système CBR, il suffit d'obtenir des cas initiaux avec leurs solutions.

- . La capacité du système CBR d'enrichir sa connaissance en faisant l'acquisition de nouveaux cas. On n'aura donc pas à modifier de règles existantes ou à en rajouter de nouvelles.

- . La capacité du système CBR de justifier sa solution en exhibant un cas qui servira de référence au lieu de donner une trace des règles qui ont conduit à la solution.

En résumé, le but n'étant pas d'affirmer que le CBR est la meilleure approche dans l'absolu, on dira plutôt qu'elle est plus adaptée à des domaines à faible fondement

théorique ou dont la connaissance n'est pas facilement exprimable en termes de règles et qu'elle présente l'avantage certain d'enrichir la connaissance du système sans avoir recours à l'ajout manuel de règles [Watson, 97].

3.1.4. Le CBR et les réseaux neuronaux

Les réseaux neuronaux et le CBR présentent des similarités superficielles. Effectivement, les deux approches se basent sur des cas rencontrés dans le passé pour résoudre les problèmes. Cependant, la similarité s'arrête ici car les réseaux neuronaux sont plutôt adaptés aux domaines où les données n'ont pas de représentation symbolique comme la reconnaissance vocale ou le traitement de signaux provenant d'instruments scientifiques. Par contre, le CBR est plus adapté au traitement de données symboliques structurées et complexes comparativement aux données purement numériques.

L'inconvénient majeur des réseaux neuronaux par rapport au CBR c'est le fait qu'ils fonctionnent comme des boîtes noires. La solution proposée par un réseau neuronal est une fonction des vecteurs pondérés de ses neurones. Aucune autre explication n'est fournie pour justifier la solution proposée. Cette situation est pire que celle des systèmes à base de règles ou les systèmes d'apprentissage ("machine learning") qui eux, fournissent au moins une trace des règles qui ont conduit à la solution [Watson, 97].

3.2. Domaines d'application

Cette section passe en revue les systèmes CBR les plus connus, classés suivant leurs domaines d'application. Le but est d'avoir une idée de la diversité des types de problèmes pour lesquels le CBR peut être utilisé.

3.2.1. Le diagnostic

Le diagnostic médical était l'un des domaines les plus populaires en intelligence artificielle depuis l'apparition du système MYCIN qui était un système expert classique à base de règles et qui diagnostiquait les infections bactériennes. Mais le diagnostic ne concerne pas juste le domaine médical. Par exemple, un mécanicien automobile procède de la même façon qu'un médecin au moment de diagnostiquer une défaillance mécanique. D'autre part, il a été prouvé qu'un médecin chevronné faisait appel à sa mémoire pour utiliser d'anciens cas de patients pour établir un diagnostic.

Nous passons en revue ici, une liste de systèmes CBR qui ont été développés pour le domaine du diagnostic :

- Le système PROTOS [Porter et Bareiss, 86] a été développé pour apprendre à classifier les problèmes auditifs présentés par les patients à partir de symptômes et de résultats de tests. L'apprentissage du système a été fait avec 200 cas classés en 24 catégories.

- Le système CASEY [Koton, 89] est un système qui fait le diagnostic des défaillances cardiaques. Il dispose en entrée de symptômes du patient et essaie d'établir une liste d'états de santé internes possibles qui auraient pu présenter les symptômes observés. Pour traiter un nouveau patient, le système CASEY tente de trouver des cas avec des symptômes similaires et pas nécessairement identiques. Ensuite, il procède à l'adaptation du diagnostic retrouvé en tenant compte des différences entre les symptômes de l'ancien et du nouveau cas.

- Le système CASCADE [Simoudis, 92] fait le diagnostic des causes de "crashes" du système d'exploitation VMS dans le but de suggérer des solutions. Cependant c'est un système qui ne fait pas d'adaptation et qui se borne donc à assister le recouvrement d'erreurs après un "crash" du système.

. Le système PAKAR [Watson et Abdullah, 94] procède à l'identification des causes possibles de défaillance dans la construction (édifices, etc.) et suggère des actions afin d'éviter ces problèmes. Le système est capable de combiner des informations de type texte avec des dessins de plans d'architecture et donne des conseils afin d'éviter les défaillances qui risquent de se présenter [Watson, 97].

3.2.2. La planification

La planification est également un domaine où on se réfère beaucoup à nos expériences passées. Par exemple, si vous aviez à faire un voyage vers un endroit pour une raison professionnelle, vous devez planifier votre séjour en commençant par le billet d'avion, la chambre d'hôtel et éventuellement une visite de musée ou une soirée spectacle. Si éventuellement, le mois prochain, vous avez à repartir au même endroit et si votre ancien séjour vous a satisfait, l'avion était à l'heure et l'hôtel était confortable, vous allez probablement reprendre votre ancienne planification en utilisant la même compagnie aérienne et le même hôtel. Cependant, vous allez probablement adapter votre ancien plan en programmant une visite à un autre musée ou une soirée pour un spectacle différent.

Les systèmes qui sont présentés dans cette section, opèrent dans le domaine de la planification dans différents champs d'application:

. Le système BATTLE [Goodman, 89] fait une projection des résultats d'application d'un plan de bataille dans le domaine de la guerre terrestre (batailles avec des soldats d'infanterie). Le système a été construit à partir d'une base de données existante comportant 600 cas. L'utilisateur décrit une situation de bataille et choisit un plan de bataille particulier. Le système retrouve d'anciens cas composés de parties de batailles et d'évaluation d'experts afin de prédire l'issue du plan de bataille choisi par l'utilisateur.

. Le système BOLERO [Lopez et Plaza, 93] établit un plan pour faire le diagnostic en utilisant les informations connues concernant le patient. Ce système combine le raisonnement à base de cas avec une approche basée sur des règles. La partie du système à base de règles renferme les connaissances concernant des maladies spécifiques alors que la partie CBR s'occupe des informations liées aux traitements des patients.

. Le système TOTLEC [Costas et Kashyap, 93] a été développé pour résoudre les problèmes complexes de planification dans le domaine de l'industrie manufacturière. Cela va de la détection d'erreurs lors de la phase de conception aux recommandations liées à la possibilité de manufacturer de certains produits [Watson, 97].

3.2.3. Le raisonnement à base de cas dans le domaine juridique

Dans beaucoup de pays, le système légal est basé sur la notion de précédent en terme de jugements rendus par le passé. Il n'est donc pas étonnant de voir que le CBR soit largement utilisé comme support au raisonnement dans le domaine légal. Les systèmes qui sont présentés dans cette section, opèrent dans le domaine juridique avec différentes spécialisations:

. Le système JUDGE [Bain, 86] est un modèle à base de cas pour rendre des sentences dans le domaine criminel. Au départ, le système travaille avec un ensemble simple de stratégies pour l'établissement d'une sentence ensuite, il commence par retrouver d'anciens cas (précédents juridiques) afin de rendre de nouvelles sentences. Le système JUDGE utilise sa base de cas afin de maintenir une certaine cohérence dans la manière de rendre des sentences.

. Le système HYPO [Ashley, 88] effectue un raisonnement juridique dans le domaine des lois sur les brevets et inventions. HYPO se base sur des cas de violation de droits liés aux brevets comme la divulgation de secrets industriels (espionnage industriel),

pour établir une argumentation plausible qui servirait aussi bien à la poursuite qu'à la défense.

. Le système KICS [Yang et Robertson, 94] opère dans le domaine de la réglementation du secteur de la construction. Le système accumule des cas d'interprétation du règlement qui vont servir à établir des précédents. Ces précédents sont par la suite utilisés au moment de la révision de la réglementation en vigueur et viendront enrichir les nouvelles versions de cette réglementation. Ces cas vont également servir de source d'information précieuse pour les experts qui auront à interpréter et prendre des décisions concernant des cas en appel (en cas de contestation de décision) [Watson, 97].

3.2.4. Le CBR pour l'aide à la conception

Il a été prouvé que les concepteurs aussi bien les architectes que les ingénieurs logiciel, réutilisent d'anciens schémas de conception, à toutes les étapes de leur carrière. Par exemple, un architecte ne réutilisera certainement pas un ancien plan dans sa totalité mais plutôt une partie de ce plan. Ceci est vrai pour beaucoup d'autres domaines et en particulier pour le développement de logiciels. Par conséquent, malgré le fait que le processus de conception soit plutôt une tâche de synthèse, il n'est pas étonnant de voir l'utilisation du CBR dans le domaine de la conception croître de plus en plus.

Les systèmes qui suivent ont été développés pour travailler dans le domaine de la conception ("design") :

. Le système CYCLOPS [Navinchandra, 88] a pour but de résoudre les problèmes dans le domaine de l'aménagement paysager. Il se concentre sur la recherche de problèmes potentiels lors de la conception d'un nouvel aménagement.

. Le système JULIA [Hinrichs, 89] opère dans le domaine de la conception de menus (recettes de cuisine). Il exploite un répertoire de méthodes d'adaptation afin de

transformer une ancienne recette tout en respectant certaines contraintes, propres au problème à traiter (comme l'indisponibilité de certains ingrédients).

. Le système CADET [Sycara, 92] est un système de conception à base de cas qui fonctionne comme assistant pour le concepteur dans le domaine de la mécanique. Il retrouve d'anciens cas de conception qui ont bien fonctionné en évitant l'utilisation de matériaux inadéquats ou trop coûteux. CADET transforme la description conceptuelle du fonctionnement d'une pièce en une description exploitable pour la recherche d'anciens cas similaires et génère une variété d'alternatives de conceptions équivalentes pour un ensemble de spécifications de conception.

. Le système ARCHIE [Pearce et al, 92] a été développé en utilisant l'outil ReMind, pour aider les architectes dans la phase de conception. Il leur donne accès à des plans de conception élaborés par d'autres architectes et souligne les facteurs à prendre en considération dans la résolution de problèmes. Dans le système ARCHIE, chaque cas est constitué de plusieurs types d'informations comme les buts à atteindre, les plans de conception, les indications afin de faire une bonne conception et satisfaire les buts projetés et les contraintes fixées et enfin les leçons à retenir.

. Le système NIRMANI [Perera et Watson, 95] a été implémenté avec l'outil ART**Entreprise* afin d'assister les développeurs d'unités de stockages industrielles. Le système dispose d'une bibliothèque de conception d'entrepôts et d'espaces de stockage avec leurs coûts de construction. Les cas sont stockés dans une structure hiérarchisée. On trouve dans les informations décrivant un cas, des dessins en deux et trois dimensions, des spécifications, des photographies et même des enregistrements vidéo donnant une description complète des édifices. À partir des besoins d'un client, le système retrouve la description d'édifice qui lui convient le mieux avec les coûts inhérents. D'autre part, avec l'utilisation d'anciens cas et l'aide du multimédia, le client raffine ses besoins et améliore la solution qui lui est proposée [Watson, 97].

3.2.5. Le raisonnement par analogie

Une caractéristique de l'intelligence humaine est notre capacité de repérer les analogies entre des situations différentes. Nous utilisons souvent cette capacité pour prendre des décisions. Le système SWALE [Kass, 89] est basé sur le raisonnement analogique. Il dispose d'une base de cas qui décrivent des histoires au sujet de gens et d'animaux morts dans des circonstances bizarres. Par exemple, si le système doit expliquer une situation anormale comme la mort d'un jeune cheval de course en pleine santé, il va chercher des cas de décès dans des circonstances et un contexte analogues, comme le décès d'un joueur de basket-ball à la suite d'une overdose de drogue. Le système va peut être suggérer que la mort du cheval de course a été causée par une administration illégale de drogue.

3.2.6. Le CBR dans la résolution de conflits

L'arbitrage est une tâche qui nécessite de la compétence et qui fait souvent appel à d'anciens cas qui servent de références. Les systèmes présentés dans cette section se spécialisent dans la résolution de conflits et de la médiation :

- Le système MEDIATOR [Simpson, 85] se spécialise dans la résolution de conflits. Lorsque plusieurs parties sont en désaccord, le système essaye de trouver un compromis. Si la solution proposée n'arrive pas à satisfaire tout le monde, MEDIATOR fait une nouvelle proposition en mémorisant l'échec qu'il vient de connaître pour éviter de le reproduire dans le futur.

- Le système PERSUADER [Sycara, 87] se propose de résoudre les conflits entre les travailleurs et le patronat. A partir d'une description de désaccord au niveau des salaires, le système propose un compromis sous forme de contrat en se référant et en adaptant des contrats utilisés par des compagnies similaires. D'autre part, PERSUADER dispose également d'un planificateur qui lui permet, contrairement aux autres systèmes

CBR, de générer de nouveaux contrats quand aucun ancien cas n'est trouvé ou adapté [Watson, 97].

3.2.7. Les systèmes CBR mettant l'accent sur l'adaptation

Un aspect important pour un système CBR est sa capacité d'adapter d'anciennes solutions afin de les réutiliser. Cependant de nombreux systèmes CBR ne font pas d'adaptation du tout alors que les systèmes présentés ci-dessous mettent spécialement l'accent sur l'adaptation :

- . Le système CHEF [Hammond, 86] crée de nouvelles recettes à partir d'anciennes. Le système commence par trouver la recette qui correspond autant que possible aux objectifs de la nouvelle recette. Ensuite, il utilise un ensemble de règles de transformations afin de modifier le cas qu'il a trouvé pour atteindre les buts de la recette courante. CHEF fournit également des explications quand il rencontre un échec afin d'expliquer comment cela s'est produit et maintient un lien entre cet échec et la recette associée afin d'éviter de faire la même erreur dans le futur.

- . Le système PLEXUS [Alterman, 86] est un planificateur qui adapte d'anciens plans à de nouvelles situations. Le système est conçu pour comparer deux plans, par exemple, en adaptant un plan pour se déplacer dans le métro d'une ville pour avoir un plan qui servirait à se déplacer dans le métro d'une autre ville. Le système dispose d'un mécanisme pour déceler les similarités lors du processus d'adaptation.

- . Le système COACH [Collins, 87] opère dans le domaine du football. Le système génère un nouveau plan de match en améliorant un ancien. COACH dispose d'un nombre réduit de cas dans sa base mais il a beaucoup de stratégies d'adaptation pour générer de nouveaux plans de match [Watson, 97].

3.2.8. Les systèmes tutoriels à base de cas

Nous apprenons souvent après avoir vu comment résoudre un problème pour ensuite essayer d'en résoudre un nouveau qui soit similaire. Il est peut être surprenant de voir qu'il n'y a pas beaucoup de systèmes tutoriels à base de CBR. Une des raisons, serait liée au fait que les éducateurs ont le sentiment que le CBR ne fournit pas d'aide pour la compréhension car il sait comment résoudre un problème mais n'essaie pas de comprendre pourquoi la solution fonctionne. Les systèmes qui suivent sont des systèmes tutoriels à base de cas :

- Le système DECIDER [Farrel, 87] aide les étudiants à comprendre et à résoudre un problème pédagogique en sélectionnant dans sa base les cas qui permettent à l'étudiant d'atteindre ses objectifs.

- Le système HYPO [Aleven et Ashley, 92] est un système tutoriel à base de cas pour les étudiants en droit. Un tuteur introduit un thème à enseigner et le système présente un ensemble de cas pédagogiques qui sera utilisé par le tuteur afin d'enseigner ce thème particulier aux étudiants [Watson, 97].

Dans le chapitre suivant, nous présentons, sous forme d'étude de cas, une introduction au domaine d'application de notre système.

4 CBR et planification financière

4.1. Gestion de portefeuilles et études de cas

Dans cette section, nous présentons des exemples d'études de cas afin de souligner les différentes étapes par lesquelles passe un planificateur financier, pour arriver à proposer un portefeuille qui tienne compte de la situation financière du client ainsi que de ses objectifs financiers. Ceci a pour but de nous faciliter aussi bien la lecture que la compréhension quand nous aborderons dans le chapitre 5, le système que nous avons implementé.

Un ouvrage complet sur le commerce des valeurs mobilières [CCVM, 95] est fourni par l'institut canadien des valeurs mobilières comme support pour le cours de formation de courtiers. Il a servi de référence à cette section et ne sera plus cité par la suite.

4.2. Exemples de cas

Dans ce qui suit, nous présentons deux exemples de situations avec une analyse de toutes les facettes importantes intervenant dans le développement d'une argumentation qui nous conduira à une solution financière.

4.2.1. Exemple 1

Monsieur X, 45 ans et son épouse Y, 44 ans, se sont mariés en 1971. Ils ont un fils unique, Z, qui est militaire de carrière depuis 1990.

Au cours des neuf dernières années, monsieur X a eu trois promotions dans la compagnie. Il aime son travail et veut le garder. En plus de son salaire annuel de \$52 000, il bénéficie de plusieurs avantages sociaux. Madame Y a un salaire de \$27 000, mais n'a pas d'avantages sociaux.

Le couple peut vivre confortablement avec \$3000 par mois et pourrait épargner environ \$20 000 par année.

Le couple possède une maison évaluée à \$160 000 ainsi que deux voitures neuves évaluées à \$14 000 chacune. Monsieur X a un REER de \$26 000 placé dans un dépôt à terme et vient juste de recevoir un legs d'environ \$200 000. Madame Y possède des bijoux d'une valeur de \$7 000.

D'autre part, chacun a \$5000 dans des comptes d'épargne. Leurs dettes consistent en une première hypothèque de \$38 000 à 13%, des prêts autos d'une valeur totale de \$7000 et le solde de leurs cartes de crédit s'élève à \$1 800.

a- Évaluation de la situation personnelle

Le couple X & Y reçoit deux salaires pour un total de \$79 000. Ils ont des emplois stables et leur fils est financièrement indépendant. Leur avoir net et leur budget familial se présentent comme suit :

Avoirs	
Espèces (comptes en banque)	\$210 000
REER	\$ 26 000
Maison (valeur du marché)	\$160 000
Bijoux, œuvres d'art, etc.	\$ 7 000
Voitures	\$ 28 000
Total	\$431 000

Tableau 3 : Exemple d'avoirs

Le taux d'imposition marginal de monsieur X est de 39,78%. Comme il est dans une tranche d'imposition supérieure à la moyenne, l'incidence fiscale des différents placements doit être prise en considération dans le choix des titres (produits financiers).

a.1. Évaluation du risque

D'autre part, le couple va prendre sa retraite dans une quinzaine d'années, mais il ne possède qu'un petit REER et les pensions du gouvernement comme revenu de retraite.

Monsieur X et Madame Y doivent donc beaucoup compter sur leurs placements pour avoir un revenu à la retraite et ne peuvent donc pas assumer un risque (indépendant du marché) trop élevé.

a.2. Contraintes temporelles

A part le fait que le couple part en retraite dans une quinzaine d'années, il n'a pas de contraintes particulières quant à la période pendant laquelle il peut placer son capital.

b. Objectifs de placement

La situation de Monsieur X exige un portefeuille équilibré ayant la plus-value comme objectif de placement premier et la réduction de l'impôt ainsi que la sécurité du capital comme objectifs secondaires. La plus-value vient en premier lieu parce que le couple doit s'assurer des capitaux pour la retraite et parce qu'il dispose d'un revenu excédentaire. Le taux d'imposition marginal de Monsieur X indique qu'il faudrait favoriser les placements qui offrent un taux de rendement élevé après impôts. Comme leur revenu de retraite dépend essentiellement des résultats du portefeuille, il faudrait éviter les titres à risque élevé.

c. Planification fiscale

Comme la planification fiscale doit faire partie intégrante de la gestion des placements, Monsieur X devrait examiner les points suivants :

. Il doit utiliser toutes les déductions possibles, telles que les frais financiers relatifs au REER et les frais de conseils en placements.

. Il peut reporter son revenu en l'étalant jusqu'à ce qu'il se trouve à un palier d'imposition moins élevé.

. Il peut prendre des mesures pour transférer une partie de son revenu futur à sa femme pour fins d'impôts.

. Il peut choisir des placements qui offrent un meilleur rendement après impôts.

d. Combinaison des avoirs

Un portefeuille bien structuré doit comprendre plusieurs catégories de titres financiers. Dans un souci d'équilibre entre les risques et les profits présentés par un portefeuille, le planificateur financier va donc opter pour une diversification des produits composant la solution qu'il propose. Les différentes catégories de titres financiers se comportent de façon diamétralement opposée en réponse aux fluctuations des marchés et à la conjoncture économique. En d'autres termes, lorsque le rendement d'une catégorie de titres est à la hausse, celui d'une catégorie différente est à la baisse. Par conséquent, le fait qu'il y ait compensation entre les différents types de titres financiers permet au planificateur de maintenir un portefeuille avec un rendement optimal. Le mot clé, ici, est donc la diversification. Dans cette approche, le principe n'est rien d'autre qu'une application directe du vieil adage « Il ne faut jamais mettre tous ses œufs dans le même panier ». Le tableau suivant présente une combinaison possible des différentes catégories d'avoirs et peut être structurée comme indiqué dans le tableau suivant :

Type de titres financiers	Plage de valeurs
Espèces ou Quasi-Espèces	15% à 20%
Valeurs à revenu fixe (obligations)	40% à 60%
Actions	20% à 40%

Tableau 4 : Combinaison de titres financiers

Les plages de valeurs pour chacune des classes de titres financiers qu'on voit dans le tableau 4, sont des recommandations données par les experts financiers afin de s'assurer du bon équilibre d'un portefeuille financier.

e. Portefeuille recommandé

Le premier objectif de placement de Monsieur X étant la plus-value, il est conseillé d'utiliser les montants maximums pour les actions, le montant minimum pour les espèces et le reste pour les valeurs à revenu fixe. L'expert financier va lui proposer la pondération suivante :

Espèces ou Quasi-Espèces (15%)		
Émission des gouvernements	15%	30000\$
Émission des sociétés	0%	
Valeurs à revenu fixe (45%)		
Court Terme	25%	50000\$
Moyen Terme	15%	30000\$
Long Terme	5%	10000\$
Actions (40%)		
Valeurs Sûres	20%	40000\$
Valeurs d'avenir	15%	30000\$
Capital de Risque	5%	10000\$
Valeurs Spéculatives	0%	
Valeur du portefeuille		
Total	100%	200000\$

Tableau 5 : Premier exemple de portefeuille

4.2.2. Exemple2

Monsieur X est célibataire et vient de prendre sa retraite à 65 ans. Durant sa carrière, il s'est occupé de la comptabilité de plusieurs petites entreprises et a connu des

périodes de chômage assez longues. Il a lancé sa propre entreprise de comptabilité qui n'a pas fonctionné et lui a occasionné des dettes qu'il a dû rembourser pendant trois ans.

Monsieur X n'a jamais cotisé à un régime de retraite et n'a pas de REER.

Sa situation financière est rehaussée par un compte en banque de \$30000 et une grande maison qu'il a achetée en 1954. D'autre part, il n'a aucune dette.

Monsieur X songe à vendre sa maison dont la valeur du marché est de \$125 000. Il songe à placer l'argent de la vente ainsi que ses économies afin de s'assurer un revenu. Comme il n'a pas versé les cotisations maximales au Régime de pensions du Canada, il ne recevra qu'environ 700\$ par mois de prestations de la sécurité de la vieillesse et du RPC.

Ce montant sera toutefois indexé en fonction de l'inflation. Monsieur X s'est trouvé un appartement et estime qu'il aura besoin de 2000\$ par mois pour vivre à l'aise.

a. Évaluation personnelle

Monsieur X, à la retraite, gardera le même style de vie modeste et n'aura aucune personne à sa charge. Cependant, sa situation est compliquée du fait qu'il n'a qu'un faible revenu garanti, que ses possibilités d'emploi sont limitées et qu'il n'est assuré que d'un montant fixe de revenu de placement. Comme son revenu annuel suffira à peine à satisfaire ses besoins, la réduction du risque devient un facteur très important dans son portefeuille et la réduction des impôts n'est plus qu'un facteur secondaire.

b. Objectifs de placement

Comme le portefeuille de Monsieur X produira environ les deux tiers de son revenu, son objectif de placement premier est le revenu et son objectif secondaire est la sécurité. Pour atteindre ces objectifs, le portefeuille de Monsieur X doit surtout être constitué de valeurs à revenu fixe.

c. Portefeuille recommandé

Le fait que Monsieur X dépende du revenu que lui rapportera son portefeuille exige du gestionnaire de portefeuille qu'il alloue des pourcentages de revenu fixe supérieurs à ceux suggérés selon la méthode de prévisions relatives au rendement (40% à 60%). L'échéance moyenne pour cette catégorie devrait être plus courte que celle que dictent les prévisions relatives aux taux d'intérêt. D'autre part, les émissions gouvernementales devraient être favorisées afin de réduire les risques. Toutefois, comme Monsieur X n'est âgé que de 65 ans, une petite partie du portefeuille pourrait être consacrée à des valeurs sûres (actions). Le portefeuille proposé aurait la pondération suivante :

Espèces ou Quasi-Espèces (10%)		
Émission des gouvernements	10%	15000\$
Émission des sociétés	0%	
Valeurs à revenu fixe (80%)		
Court Terme	50%	75000\$
Moyen Terme	20%	30000\$
Long Terme	10%	15000\$
Actions (10%)		
Valeurs Sûres	10%	15000\$
Valeurs d'avenir	0%	
Capital de Risque	0%	
Valeurs Spéculatives	0%	
Valeur du portefeuille		
Total	100%	150000\$

Tableau 6 : Deuxième exemple de portefeuille

Les deux exemples vus précédemment nous donne une idée de la démarche à suivre lorsqu'on a à traiter les informations décrivant un client afin lui proposer une

solution sous forme de portefeuille financier. Dans le chapitre qui suit, nous présentons les différentes composantes du prototype (le système GIPF) que nous avons implanté. Nous verrons également, les différents types de titres financiers plus en détails en soulignant les avantages et les inconvénients de chacun d'eux.

5 Réalisation d'un prototype

5.1. Introduction

Le but de ce projet est de développer un système qui analyse les informations décrivant le profil d'un client et propose un portefeuille financier adapté à ce dernier. Le système utilise une base de cas qu'il structure de manière à effectuer une classification efficace de ses données. Il bâtit, également une base de cas d'adaptation au fur et à mesure qu'il adapte d'anciennes solutions. Outre le fait que le système intègre dans le processus d'indexation la notion de similarité, l'originalité de ce travail réside dans l'utilisation de l'approche CBR de manière récursive. Effectivement le CBR est utilisé aussi bien pour retrouver des cas similaires que pour retrouver une ancienne adaptation de solutions.

Dans ce chapitre, nous commençons par une présentation globale du système, ensuite, nous présentons une description formelle illustrée par un exemple. Pour finir, nous détaillons chacun des modules du système ainsi que les objets manipulés.

5.2. Présentation du système GIPF

Le système GIPF est composé de quatre principaux modules. Ces derniers sont décrits ci-après :

- Indexation :
Le but de ce module est d'indexer les cas sur les différents attributs afin de les stocker dans la base.
- Recherche :
L'objectif de ce module est de retrouver, dans la base, les cas similaires au cas courant.
- Adaptation :
Ce module s'occupe de la modification d'anciennes solutions et de la gestion de la base d'adaptation.
- Saisie de données :
Ce module permet à l'utilisateur d'introduire les données. Il s'occupe également de représenter ces informations sous une forme interne exploitable par les autres modules du système.

5.2.1. Architecture du système

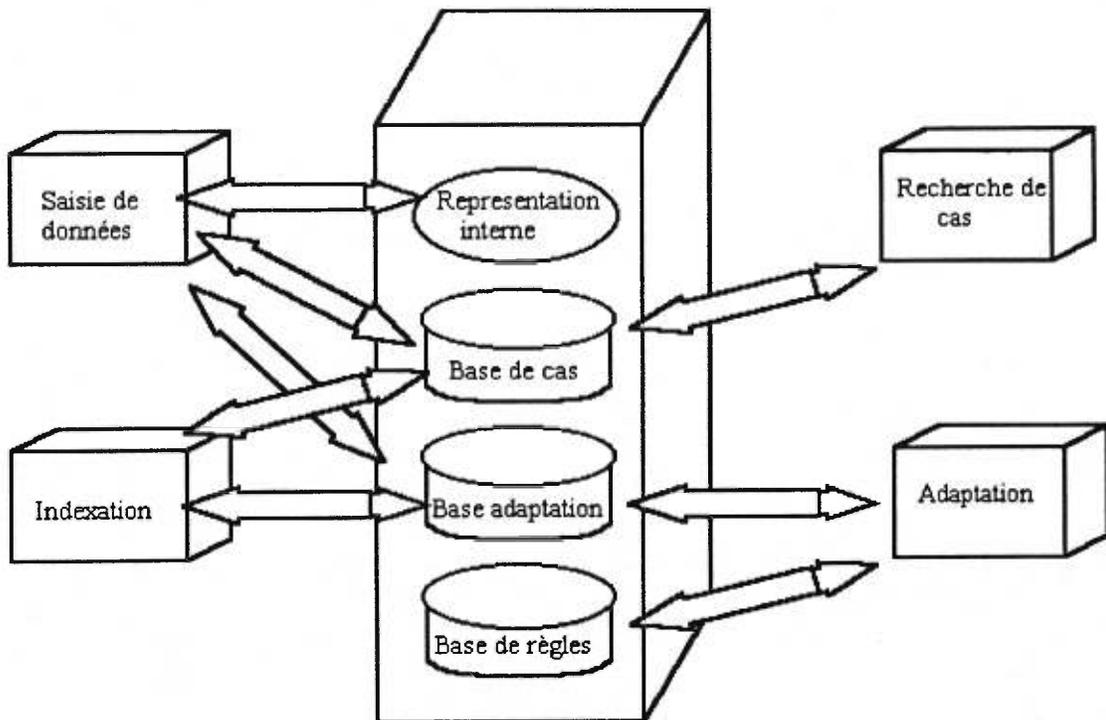


Figure 2 : Structure du système GIPF

La figure ci-dessous présente le menu principal du système.

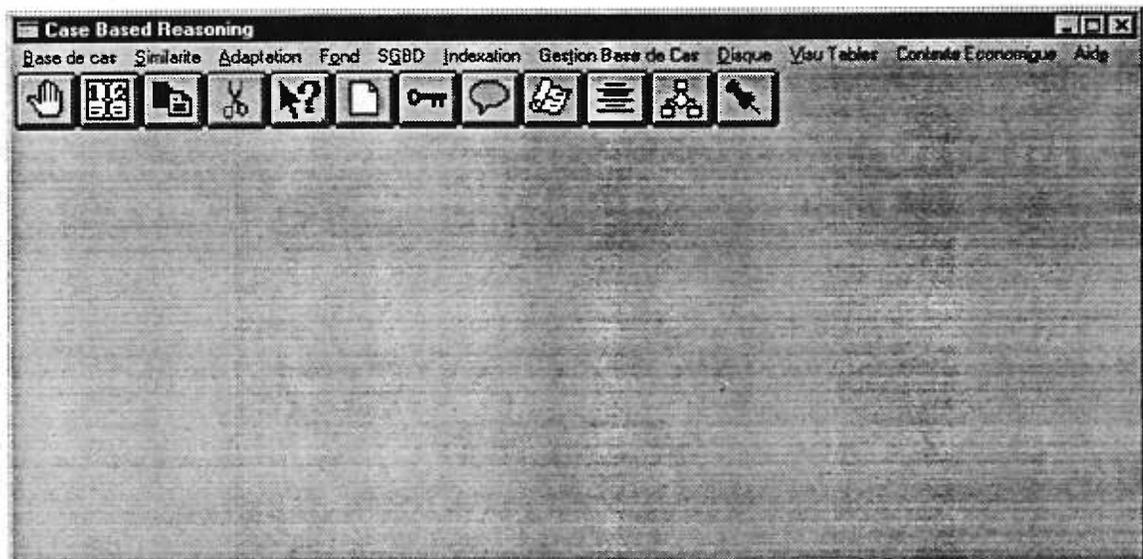


Figure 3 : Le système GIPF

5.3. Approche formelle pour GIPF

Dans cette section, nous présentons une description formelle du système GIPF. L'approche, présentée ici, s'inspire des travaux de Wess et Ritcher [Wess et Ritcher, 93] ainsi que ceux de Wess et Globig [Wess et Globig, 94]. Le formalisme est ensuite illustré par un exemple, dans la section 5.4.

5.3.1. Domaine de valeurs

Soit U l'ensemble des objets manipulés par un système CBR et qui constitue la base de cas. Chaque cas est représenté par un vecteur u à n dimensions. Le nombre de dimensions étant le nombre d'attributs décrivant un cas. D'autre part chaque attribut x_i prend ses valeurs dans un domaine D_i .

Cette description est présentée de façon plus formelle comme suit :

$u \in U =$ univers d'objets.

$u = (x_1, x_2, \dots, x_n)$

$x_i \in D_i : D_i =$ Domaine de l'attribut x_i

$u \in D_1 \times D_2 \times \dots \times D_n$

Par exemple, U pourrait être l'ensemble des personnes. Parmi les caractéristiques qui décrivent une personne, nous retrouvons l'attribut salaire et l'attribut âge. L'attribut salaire pourrait prendre ses valeurs dans le domaine $D_1 = [10000, 500000]$, alors que l'attribut âge pourrait prendre ses valeurs dans le domaine $D_2 = [0, 100]$.

5.3.2. Pondération

Les différents attributs décrivant un cas sont regroupés en catégories. À chaque catégorie est associé un poids.

Soit le cas $u \in U$:

$$u = (x_{11}, x_{12}, x_{21}, x_{22}, x_{23}, \dots, x_{nk})$$

x_{ij} représente l'attribut $j \in$ Catégorie i

w_{ij} est le poids de l'attribut $j \in$ Catégorie i

$$w_{ij} \in [0,1]$$

w_j représente le poids associé aux attributs appartenant à la catégorie j .

Pour toute catégorie j , constituée de n attributs, on a :

$$\sum_{i=1}^n w_{ij} = 1$$

D'autre part, pour $m =$ nombre de catégories d'attributs, on a:

$$\sum_{j=1}^m w_j = 1$$

On a également, pour les n attributs des m catégories :

$$\sum_{j=1}^m \sum_{i=1}^n w_j w_{ij} = 1$$

Note : $w(x_i) = 0$ Si la valeur de l'attribut x_i n'est pas connue.

Par exemple, si pour décrire une personne, nous avons les attributs âge, salaire, objectifs financiers et tolérance au risque. Nous pouvons avoir d'une part, une première catégorie d'attributs *personnels* qui regroupe l'âge et le salaire, et d'autre part une deuxième catégorie d'attributs *investissement* qui comprend les objectifs financiers et la tolérance au risque. Pour les deux catégories que nous venons de voir, la pondération est la suivante $W = \{ 40\%, 60\% \}$.

Dans la catégorie d'attributs personnels (catégorie 1), nous avons :

w_1 représente le poids associé aux attributs appartenant à la catégorie 1.

$$w_1 = 40\%$$

w_{11} est le poids de l'attribut 1 (*âge*) \in Catégorie 1

$$w_{11} = 50\%$$

w_{12} est le poids de l'attribut 2 (*salaires*) \in Catégorie 1

$$w_{12} = 50\%$$

$$w_{11} * w_1 + w_{12} * w_1 = 50\% * 40\% + 50\% * 40\% = 1$$

Quant à la catégorie investissement (catégorie 2), la répartition est comme suit:

w_2 représente le poids associé aux attributs appartenant à la catégorie 2.

$$w_2 = 60\%$$

w_{21} est le poids de l'attribut 1 (*risque*) \in Catégorie 2

$$w_{21} = 40\%$$

w_{22} est le poids de l'attribut 2 (*objectifs financiers*) \in Catégorie 2

$$w_{22} = 60\%$$

$$w_{21} * w_2 + w_{22} * w_2 = 40\% * 60\% + 60\% * 60\% = 1$$

Notons que la somme des poids est égale à 1. Ici nous avons deux catégories et deux attributs par catégorie.

Pour la première catégorie *attributs personnels*, nous avons:

$$\sum_{i=1}^n w_{ij} = \sum_{i=1}^2 w_{i1} = 50\% + 50\% = 1$$

Pour la deuxième catégorie *investissement*, nous avons :

$$\sum_{i=1}^n w_{ij} = \sum_{i=1}^2 w_{i2} = 40\% + 60\% = 1$$

En fait, nous avons quatre attributs en tout :

$$\sum_{j=1}^4 w_j = 20\% + 20\% + 24\% + 36\% = 1$$

D'autre part, le total des poids de toutes les catégories est :

$$\sum_{j=1}^m \sum_{i=1}^n w_j w_{ij} = (50\% + 50\%) * 40\% + (40\% + 60\%) * 60\% = 1$$

5.3.3. Mesure de distance

Les cas étant représentés par des vecteurs à n dimensions, nous définissons une distance séparant deux cas, comme la somme pondérée des distances locales séparant les attributs.

Soit d_i , la fonction de distance locale séparant deux valeurs différentes de l'attribut x_i . Cet attribut x_i prend ses valeurs dans le domaine D_i .

Soient v_1, v_2 deux valeurs de x_i , on a donc :

$$v_1, v_2 \in D_i$$

$$d(v_1, v_2) : D_i \times D_i \rightarrow R$$

La distance séparant un cas q d'un cas r est définie comme suit :

$$\boxed{dis(r, q) = \sum_{a=1,4} w_a * d_a(q_a, r_a)}$$

avec w_a : poids de l'attribut a .

q_a : l'attribut a du cas q .

r_a : l'attribut a du cas r .

et d_a : La distance locale pour l'attribut a .

Par exemple, parmi les caractéristiques qui décrivent une personne, nous retrouvons l'attribut âge. La distance d_i qui sépare deux valeurs de l'attribut âge, peut être la valeur absolue de la différence arithmétique.

Pour deux valeurs $v_1 = 30$ et $v_2 = 40$, la distance les séparant est :

$$|30-40| = 10.$$

5.3.4. Fonction de similarité

Le degré de similarité entre deux cas est une valeur entre 0 et 1 qui reflète l'ampleur de la différence les séparant. Plus la valeur de la mesure de similarité est proche de la valeur 1, plus les cas sont proches. Inversement, plus elle est proche de la valeur 0, plus les cas sont éloignés les uns des autres. La mesure de similarité est définie comme suit :

$$sim(q,c) : D_i \times D_i \rightarrow [0, 1]$$

Plus précisément, la similarité $sim(q,c)$ entre un cas q et un cas c de la base cas, est définie comme suit :

$$sim(q,c) = \sum_{a=1,n} w_a * sim_a(q_a, c_a)$$

q_a : attribut a du cas q

c_a : attribut a du cas c

w_a : poids de l'attribut c_a

sim : La mesure de similarité entre le cas q et le cas a .

sim_a : La mesure de similarité locale pour l'attribut a .

Si les cas sont décrits par n attributs, nous avons:

$$\sum_{a=1,n} w_a = 1 \text{ et } sim(q_a, c_a) \in [0,1]$$

$$\text{On obtient donc : } sim(q,c) \in [0,1]$$

La similarité entre les cas se traduit par un degré de similarité composé qui tient compte de la similarité entre leurs attributs ainsi que la pondération associée à ces derniers.

Par exemple, nous dirons que deux personnes présentent une similarité de plus en plus grande, exprimée en termes de pourcentage, donc entre 0 et 1 (100%), si la distance séparant leurs âges est de plus en plus petite.

5.3.5. Relation entre distance et similarité

Comme nous l'avons vu précédemment, les notions de distance et de similarité sont inversement proportionnelles. Autrement dit, plus deux cas sont proches donc présentent un degré de similarité élevé, plus la distance qui les sépare est petite.

Soient les cas u_1, u_2, u_3, u_4 :

$$d(u_1, u_2) \leq d(u_3, u_4) \Leftrightarrow \text{sim}(u_1, u_2) \geq \text{sim}(u_3, u_4)$$

d : distance entre les cas

sim : mesure de similarité entre les cas

La similarité est définie en fonction de la distance comme suit :

$$\text{sim}(q, r) = 1 / (1 + d(q, r))$$

5.3.6. Voisinage

Soient $v_j(x_i)$, les valeurs possibles de l'attribut x_i dans le domaine D_i .
Le voisinage d'une valeur particulière $v_k(x_i)$ sera l'ensemble :

$$V_{\delta_i}(v_k(x_i)) = \{ v_m(x_i) / d(v_k(x_i), v_m(x_i)) \leq \delta_i \}$$

$v_k(x_i)$: valeur k de l'attribut x_i .

$v_m(x_i)$: valeur m de l'attribut x_i .

L'intervalle δ_i est déterminé relativement au domaine D_i de l'attribut x_i .

Par exemple, si nous regroupons des personnes par tranches d'âge. Nous pouvons dans ce cas, avoir un voisinage de $\pm 16\%$ pour chaque valeur de l'attribut âge. Les individus qui sont dans le voisinage de 30 ans ont donc un âge compris entre 25 et 35 ans.

5.3.7. Classification

La classe d'un cas est l'ensemble des objets faisant partie de son voisinage. Soit le cas $u = (x_1, x_2, \dots, x_n)$.

La classe du cas u sera donc définie comme suit :

$$\text{Classe}(u) = \{ t = (v_{k1}(t_1), v_{k2}(t_2), \dots, v_{kn}(t_n)) / v_{ki}(t_i) \in V_{\delta_i}(v_{ki}(x_i)) \ i = 1, n \}$$

$v_{ki}(t_i)$: valeur k de l'attribut t_i

5.4. Exemple

L'exemple suivant illustre une utilisation du formalisme présenté précédemment. Supposons que les cas soient décrits par 4 attributs numériques. Un cas est donc représenté de la façon suivante $q = \{ x_1, x_2, x_3, x_4 \}$. Le vecteur de pondération des attributs sera $W = \{ w_1, w_2, w_3, w_4 \}$.

Soit un nouveau cas $q = \{ 12, 30, 40, 60 \}$

Les attributs ont le vecteur de pondération suivant :

$$W = \{ 20\%, 15\%, 40\%, 25\% \}$$

On a bien, $\sum_{a=1,4} w_a = 20\% + 15\% + 40\% + 25\% = 1$

Supposons que le processus de recherche trouve trois cas proches du cas courant q .

Soit l'ensemble des voisins trouvés $K = \{ r_1, r_2, r_3 \}$ avec :

$$r_1 = \{ 11, 33, 44, 57 \}$$

$$r_2 = \{ 10, 28, 38, 62 \}$$

$$r_3 = \{ 13, 32, 42, 61 \}$$

Reprenons la relation entre la fonction de similarité locale et la distance, présentée précédemment, en 5.3.5 :

$$sim_a(q_a, r_{ia}) = 1 / (1 + dis_a(q_a, r_{ia})) \quad \text{pour } a = 1 \dots 4$$

avec r_{ia} = l'attribut a du cas de r_i

Calculons les voisinages des attributs du cas q . Soient les valeurs des différents δ_i :

$$\delta_1 = 2$$

$$\delta_2 = 3$$

$$\delta_3 = 4$$

$$\delta_4 = 3$$

Pour les attributs x_1, x_2, x_3 et x_4 du cas q , qui ont respectivement les valeurs 12, 30, 40, 60, on a les voisinages suivants :

$$V_{\delta_1}(12) = \{ x_1 / d(12, x_1) \leq 2 \}$$

$$V_{\delta_2}(30) = \{ x_2 / d(30, x_2) \leq 3 \}$$

$$V_{\delta_3}(40) = \{ x_3 / d(40, x_3) \leq 4 \}$$

$$V_{\delta_4}(60) = \{ x_4 / d(60, x_4) \leq 3 \}$$

Pour le cas $r_1 = \{ 11, 33, 44, 57 \}$ on a :

$$11 \in V_{\delta_1}(12)$$

$$33 \in V_{\delta_2}(30)$$

$$44 \in V_{\delta_3}(40)$$

$$57 \in V_{\delta_4}(60)$$

Pour le cas $r_2 = \{ 10, 28, 38, 62 \}$ on a :

$$10 \in V_{\delta,1}(12)$$

$$28 \in V_{\delta,2}(30)$$

$$38 \in V_{\delta,3}(40)$$

$$62 \in V_{\delta,4}(60)$$

Et enfin pour le cas $r_3 = \{ 13, 32, 42, 61 \}$ on a :

$$13 \in V_{\delta,1}(12)$$

$$32 \in V_{\delta,2}(30)$$

$$42 \in V_{\delta,3}(40)$$

$$61 \in V_{\delta,4}(60)$$

Si on procède à la classification des cas r_1, r_2 et r_3 :

$$\text{Classe}(q) = \{ t = (t_1, t_2, \dots, t_n) / t_i \in V_{\delta,i}(x_i) \ i=1,4 \}$$

Ces trois cas appartiennent à la classe de q .

D'autre part, si on calcule le degré de similarité entre les attributs de r_1 et ceux du cas q , on obtient :

$$\text{sim}_1(12,11) = 1/(1 + \text{dis}_1(12,11)) = 1/(1 + |12-11|) = 1/2 = 50\%$$

$$\text{sim}_2(30,33) = 1/(1 + \text{dis}_2(30,33)) = 1/(1 + |30-33|) = 1/4 = 25\%$$

$$\text{sim}_3(40,44) = 1/(1 + \text{dis}_3(40,44)) = 1/(1 + |40-44|) = 1/5 = 20\%$$

$$\text{sim}_4(60,57) = 1/(1 + \text{dis}_4(60,57)) = 1/(1 + |60-57|) = 1/4 = 25\%$$

Le calcul de la similarité entre le cas r_1 et le cas q donne :

$$\text{sim}(r_1, q) = \sum_{a=1,4} w_a * \text{sim}_a(r_{1a}, q_a)$$

$$\text{sim}(r_1, q) = 20\% * 50\% + 15\% * 25\% + 40\% * 20\% + 25\% * 25\%$$

$$\boxed{\text{sim}(r_1, q) = 28\%}$$

De la même façon, si on calcule le degré de similarité entre les attributs de r_2 et ceux du cas q , on obtient :

$$\text{sim}_1(12,10) = 1/(1 + \text{dis}_1(12,10)) = 1/(1 + |12-10|) = 1/3 = 33,33\%$$

$$sim_2(30,28) = 1/(1 + dis_2(30,28)) = 1/(1 + |30-28|) = 1/3 = 33,33 \%$$

$$sim_3(40,38) = 1/(1 + dis_3(40,38)) = 1/(1 + |40-38|) = 1/3 = 33,33 \%$$

$$sim_4(60,62) = 1/(1 + dis_4(60,62)) = 1/(1 + |60-62|) = 1/3 = 33,33 \%$$

De même pour le calcul de la similarité entre le cas r_2 et le cas q :

$$sim(r_2, q) = 20\% * 33,33\% + 15\% * 33,33\% + 40\% * 33,33\% + 25\% * 33,33\%$$

$$sim(r_2, q) = 33,33 \%$$

Pour la similarité entre les attributs de r_3 et ceux du cas q , on obtient :

$$sim_1(12,13) = 1/(1 + dis_1(12,13)) = 1/(1 + |12-13|) = 1/2 = 50 \%$$

$$sim_2(30,32) = 1/(1 + dis_2(30,32)) = 1/(1 + |30-32|) = 1/3 = 33,33 \%$$

$$sim_3(40,42) = 1/(1 + dis_3(40,42)) = 1/(1 + |40-42|) = 1/3 = 33,33 \%$$

$$sim_4(60,61) = 1/(1 + dis_4(60,61)) = 1/(1 + |60-61|) = 1/2 = 50 \%$$

La similarité entre les cas r_3 et q donne :

$$sim(r_3, q) = 20\% * 50\% + 15\% * 33,33\% + 40\% * 33,33\% + 25\% * 50\%$$

$$sim(r_3, q) = 40,83 \%$$

Reprenons la formule qui calcule la distance séparant un cas r_i d'un cas q , présentée précédemment en 5.3.3 :

$$dis(r_i, q) = \sum_{a=1,4} w_a * dis_a(q_a, r_{ia})$$

avec w_a : poids de l'attribut a .

et dis_a : La distance locale pour l'attribut a .

Le calcul de la distance composée qui tient compte de la pondération associée à chaque attribut et qui sépare le cas r_1 du cas q donne :

$$dis(r_1, q) = w_1 * dis(11, 12) + w_2 * dis(33, 30) + w_3 * dis(44, 40) + w_4 * dis(57, 60)$$

$$dis(r_1, q) = 20\% * 1 + 15\% * 3 + 40\% * 4 + 25\% * 3$$

$$dis(r_1, q) = 3$$

La distance séparant le cas r_2 du cas q donne :

$$dis(r_2, q) = w_1 * dis(10, 12) + w_2 * dis(28, 30) + w_3 * dis(38, 40) + w_4 * dis(62, 60)$$

$$dis(r_2, q) = 20\% * 2 + 15\% * 2 + 40\% * 2 + 25\% * 2$$

$$dis(r_2, q) = 2$$

Enfin, la distance séparant le cas r_3 du cas q donne :

$$dis(r_3, q) = w_1 * dis(13, 12) + w_2 * dis(32, 30) + w_3 * dis(42, 40) + w_4 * dis(61, 60)$$

$$dis(r_3, q) = 20\% * 1 + 15\% * 2 + 40\% * 2 + 25\% * 1$$

$$Dis(r_3, q) = 1,55$$

Le système va retenir r_3 comme le cas le plus proche du nouveau cas q , car nous avons :

$$dis(r_3, q) = \min \{ dis(r_1, q), dis(r_2, q), dis(r_3, q) \}$$

$$sim(r_3, q) = \max \{ sim(r_1, q), sim(r_2, q), sim(r_3, q) \}$$

5.5. Représentation

Dans cette section, nous présentons la structure des cas ainsi que celle des solutions. Nous y voyons également la structure fonctionnelle du système GIPF ainsi que l'organisation de mémoire retenue pour représenter la base de cas.

5.5.1. Structure d'un cas

Dans la figure 4, nous avons la structure d'un cas ainsi que sa représentation dans le système GIPF. À part les informations décrivant un client comme l'état civil, l'âge et les objectifs de placement, nous avons sa tolérance au risque et son bilan financier.

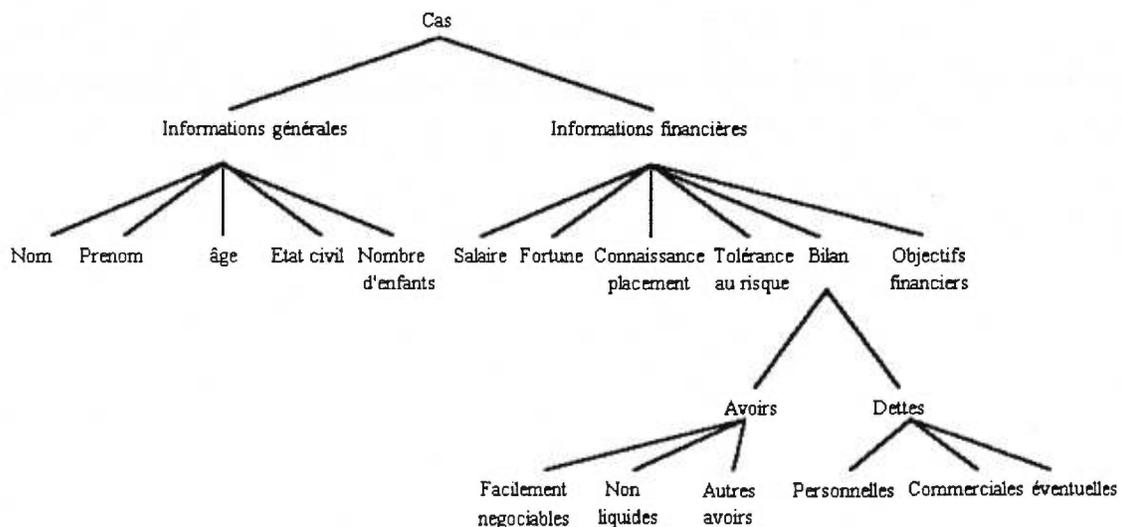


Figure 4 : Structure d'un cas

Informations Client

Nom: Barry
 Prénom: Jean
 NAS: 250938291
 Téléphone: 5144373982
 Etat civil: C
 Enfants: 0
 Province: Qc
 Ville: St-Hilaire
 Code postal: J3H 5E5

Fortune nette

10M\$
 20M\$
 50M\$
 100M\$
 Plus de 100M

Connaissance de placement

Excellent
 Bonne
 Passable
 Aucune

Age

Mineur Majeur-30 31-40
 41-60 61-65 Plus de 65

Objectifs du client

Revenu
 Sécurité
 Plus-value

Revenu annuel: 140000

Portefeuille Bilan Risque

OK Cancel Help

Figure 5 : Description d'un cas

Une caractéristique importante et propre à chaque client est sa capacité à gérer la part de risque encouru lors de la réalisation d'un placement. L'expert financier évalue quantitativement cette capacité en affectant à chaque client, une valeur numérique entre 1 et 10 ou une valeur sous forme de pourcentage. La figure 6 présente donc un exemple de tolérance au risque d'un client dans le système GIPF.

Risque

Tolérance au risque:

66 %

OK Cancel Help

Figure 6 : Tolérance au risque d'un client

La figure 7 illustre le bilan de client avec les différentes catégories d'avoirs et de dettes:

Figure 7 : Bilan d'un client

Dans les sections suivantes, nous allons passer en revue, les avoirs et les dettes qui apparaissent dans le bilan d'un client.

Category	Value
Especes (comptes en banque, obligations d'epargne du Canada, etc.) :	16114
Depot a terme et certificats de placement garantis :	26724
Obligations, a la valeur du marche :	7055
Actions, a la valeur du marche :	26116
Fonds communs de placement, a la valeur de rachat :	26065
Valeur de rachat d'un contrat d'assurance sur la vie :	31259
Hypotheques a leur valeur en capital	22918
Autres :	9192

Figure 8 : Avoirs facilement négociables

La figure 8 nous décrit les différents types d'avoires dits facilement négociables. Ce genre d'avoires peut rapidement être converti en argent liquide. La figure 9 présente les avoires non liquides comme les régimes de retraite, les régimes de rente et les abris fiscaux. Ceux sont des avoires qui ne sont pas convertibles en argent liquide très rapidement.

Avoirs financiers non liquides	
AVOIRS	
Avoirs financiers non liquides	
Regimes de retraite, a la valeur acquise	8551
REER :	23145
Abris fiscaux, au prix coutant ou a la valeur estimative :	1559
Rentes :	26568
Autres	6213

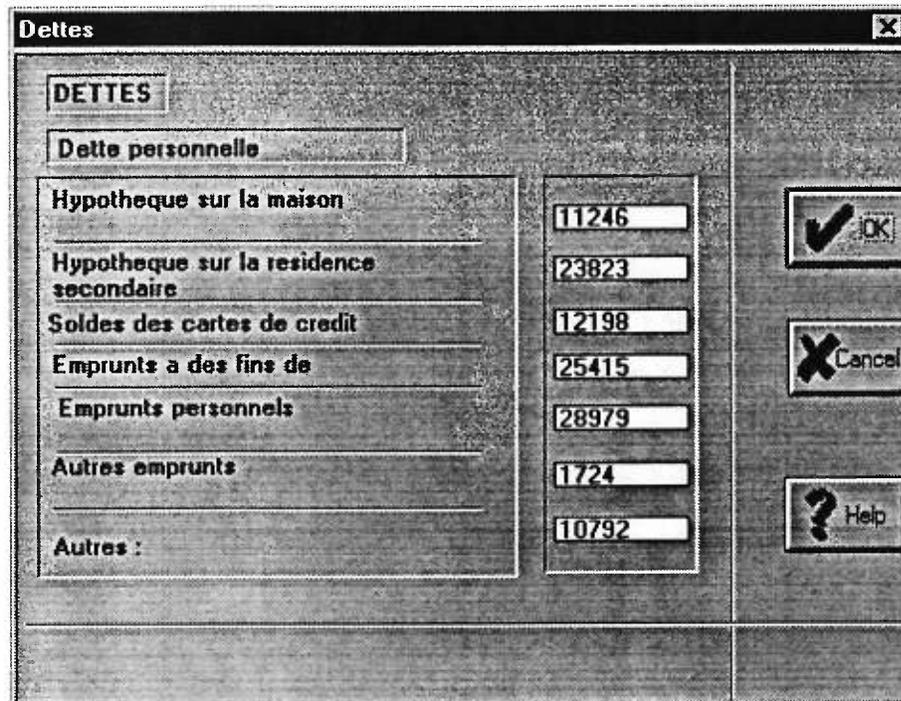
Figure 9 : Avoirs financiers non liquides

Le reste des avoires, comme les biens immobiliers, les œuvres d'art, les bijoux etc., sont regroupés dans la catégorie *Autres avoires*, comme le montre la figure 10.

Autres avoires	
AVOIRS	
Autres avoires	
Maison, a la valeur du marche	30573
Residence secondaire, a la valeur du marche	11939
Participation dans des entreprises, a la valeur du marche	15406
Antiquites, oeuvres d'art, bijoux, or et argent, objets de collection, etc.	937
Automobiles, bateaux, etc.	1658
Autres biens immobiliers	16437
Autres :	12531

Figure 10 : Autres avoires

La figure 11 décrit la dette personnelle qui est composée d'hypothèques sur les biens immobiliers ainsi que de prêts personnels et de soldes de cartes de crédit.



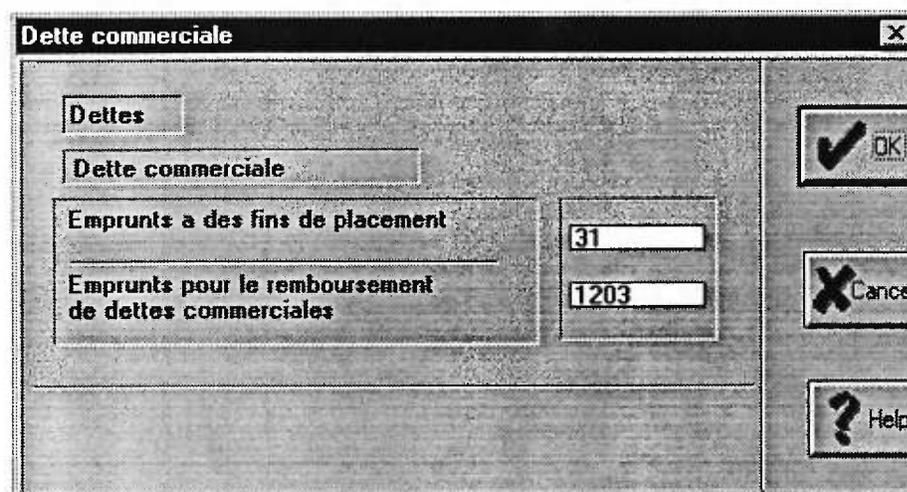
The screenshot shows a dialog box titled "Dettes" with a close button (X) in the top right corner. Inside, there is a section labeled "DETTE PERSONNELLE" with a sub-section "Dette personnelle". Below this, there are several rows of labels and input fields:

Category	Amount
Hypothèque sur la maison	11246
Hypothèque sur la résidence secondaire	23823
Soldes des cartes de crédit	12198
Emprunts à des fins de	25415
Emprunts personnels	28979
Autres emprunts	1724
Autres :	10792

On the right side of the dialog, there are three buttons: "OK" (with a checkmark), "Cancel" (with an X), and "Help" (with a question mark).

Figure 11 : Dette personnelle

Parmi les dettes qu'un client peut avoir, nous avons les dettes commerciales comme décrit dans la figure ci-dessous.



The screenshot shows a dialog box titled "Dette commerciale" with a close button (X) in the top right corner. Inside, there is a section labeled "Dettes" with a sub-section "Dette commerciale". Below this, there are two rows of labels and input fields:

Category	Amount
Emprunts à des fins de placement	31
Emprunts pour le remboursement de dettes commerciales	1203

On the right side of the dialog, there are three buttons: "OK" (with a checkmark), "Cancel" (with an X), and "Help" (with a question mark).

Figure 12 : Dette commerciale

Comme dette, nous retrouvons également, regroupés dans la catégorie *Dettes éventuelles*, les prêts pour lesquels le client a pris un engagement de cautionnement. La figure 13 nous donne un exemple de dette éventuelle.

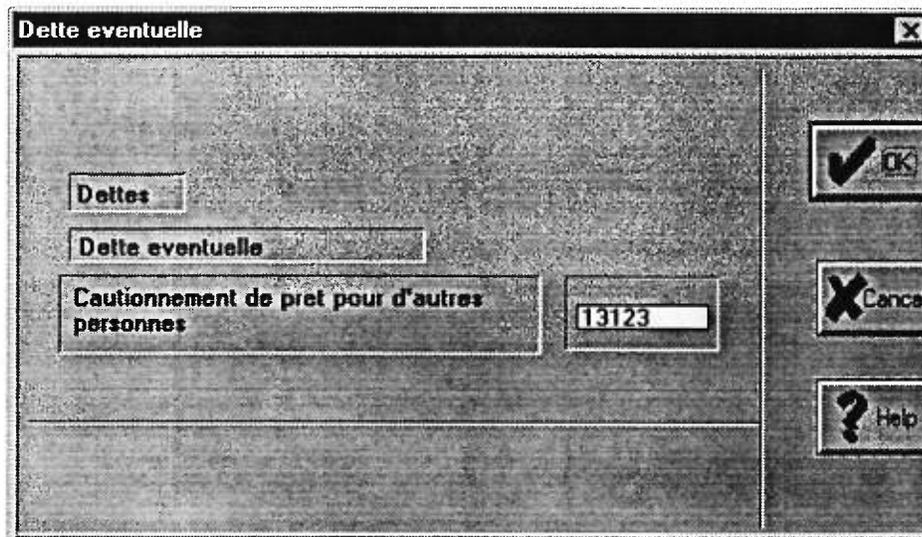


Figure 13 : Dette éventuelle

Enfin, nous avons l'avoir net du client obtenu en faisant la différence entre le total des avoirs et le total des dettes.

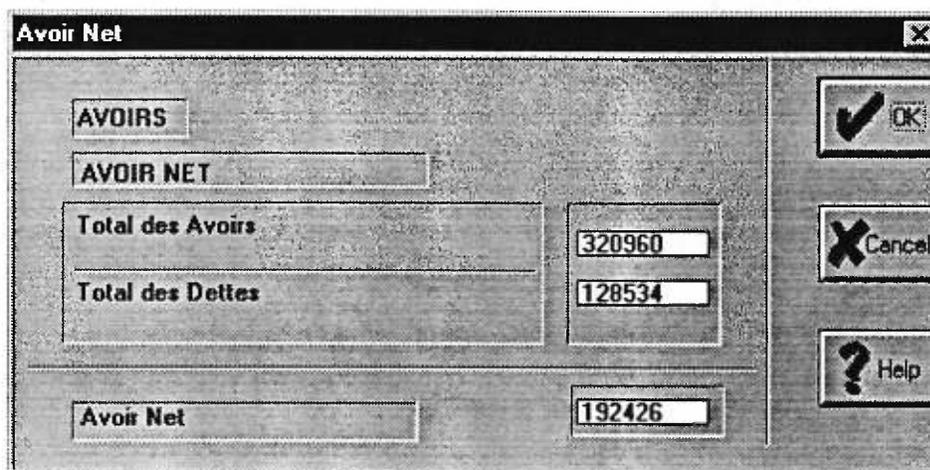


Figure 14 : Avoir net

5.5.2. Description de la structure d'une solution associée à un cas

Le système propose ses solutions sous forme de portefeuilles financiers. Nous allons donc passer en revue les différentes composantes d'une solution. La combinaison des avoirs dans un portefeuille financier doit tenir compte des objectifs du client ainsi que de la situation économique courante. Le système doit tenir compte d'un contexte économique constitué d'un ensemble de facteurs. Ces facteurs sont de diverses natures et influencent de façon différente les prévisions relatives au rendement d'un portefeuille financiers.

Comme les prévisions relatives au rendement peuvent être imprécises, il faut faire une série d'estimations. Il est difficile de quantifier les facteurs inhérents à chaque catégorie de titres financiers. On utilise alors des évaluations qui prennent leurs valeurs qualitatives dans l'ensemble {positive, négative, neutre}.

Lorsque chaque facteur a été évalué, on peut établir une perspective d'ensemble pour chaque catégorie d'avoir. Cette prévision va influencer le pourcentage que l'on va accorder à une catégorie de titres dans la constitution d'un portefeuille financier et la façon dont on va adapter un cas, en appliquant des règles de transformation, afin d'obtenir une solution adéquate au cas courant.

Avant d'introduire les règles qui vont servir à ajuster une solution pour un cas particulier, nous allons introduire certaines notions liées à l'analyse financière qui est donc le domaine d'application de notre système.

Les différentes composantes d'un portefeuille peuvent être découpées en trois grandes catégories. Ceux sont les actions, les valeurs à revenu fixe (les obligations) ainsi que les espèces ou quasi-espèces (bons du trésor). Pour chaque catégorie, nous allons passer en revue les facteurs pouvant l'influencer.

- **Actions**

Pour évaluer les cours futurs des actions, et donc de leurs rendements, on se sert de quatre grandes catégories d'indicateurs. Les indicateurs fondamentaux qui sont représentés par les bénéfices des sociétés, les indicateurs techniques qui sont les courbes des indices boursiers (Dow Jones, TSE300,..), les indicateurs économiques qui sont divisés à leur tour, en trois sous-ensembles: les indicateurs précurseurs (mises en chantier, nouvelles commandes aux fabricants,...) , les indicateurs simultanés (PNB, revenu des particuliers, ventes au détail,...) , les indicateurs retardataires (taux de chômage, dépenses en biens de production,...).

Nous avons, finalement les indicateurs de rendement (Ratios cours-bénéfice, rendements boursiers composés,...).

- **Valeurs à revenu fixe**

Les valeurs à revenu fixes se découpent en trois catégories en fonction de leurs échéances. Nous avons donc des valeurs à court, moyen et long terme.

Afin de déterminer le rendement futur des titres à revenu fixe, il faut prévoir la tendance future des taux d'intérêt. Contrairement au cours des actions, les taux d'intérêt sont en général fixés et contrôlés par les gouvernements, qui désirent atteindre des objectifs économiques précis.

Les facteurs pouvant influencer la tendance des taux d'intérêt peuvent être découpés en quatre principaux groupes : La politique monétaire, les indicateurs économiques, l'inflation et la valeur du dollar canadien.

- **Espèces ou quasi-espèces**

Le rendement des espèces et des quasi-espèces se calcule en fonction du taux d'intérêt prévu lors de l'analyse des valeurs à revenu fixe.

Cette catégorie est généralement perçue comme celle qui garantit au gestionnaire de portefeuille un taux de rendement précis sans risque pour le capital.

Récapitulatif du risque selon la catégorie d'avoirs [CCVM, 95]

Le tableau suivant présente, sous forme d'étude comparative, les caractéristiques de chaque type de titres financiers.

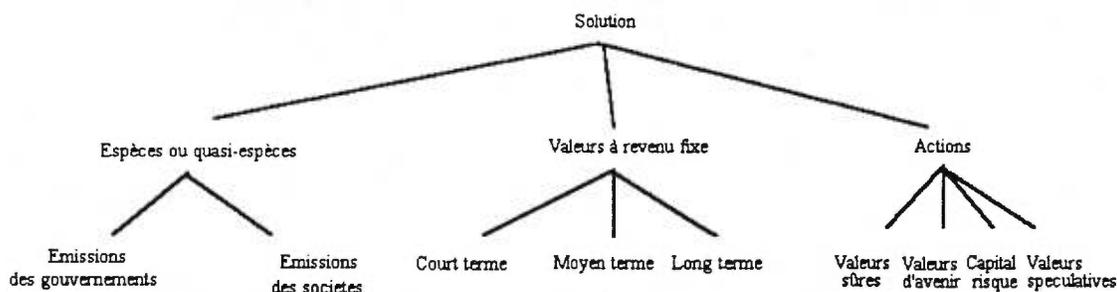
Chaque catégorie d'avoirs financiers est évaluée en fonction de critères comme le risque, le bénéfice, le cours, le terme et le rendement. Il est évident qu'il n'y a pas de titre parfait, chacun ayant ses avantages et ses inconvénients relativement aux besoins du client. Le choix va donc se faire en fonction des objectifs financiers de chacun et la solution idéale va être une bonne combinaison de plusieurs produits financiers différents.

<u>Espèces ou quasi-espèces</u>	
1. Émissions des gouvernements	- Risque le plus faible, meilleure qualité
2. Émission de sociétés	- Risque le plus élevé, qualité la plus faible
<u>Valeurs à revenu fixe</u>	
1. Court terme (jusqu'à trois ans)	- Risque faible, cours stable
2. Moyen terme (trois à dix ans)	- Risque moyen, cours modérément stable
3. Long terme (plus de dix ans)	- Risque élevé, cours instable
<u>Actions</u>	
1. Valeurs sûres	-Risque faible, capital-actions important, bénéfices prévisibles, rendement élevé, fort pourcentage du bénéfice versé en dividendes faible ratio cours-bénéfice, cours peu volatils.
2. Valeurs d'avenir	- Risque moyen, capital-actions moyen, probabilité d'une croissance des bénéfices supérieure à la moyenne, direction dynamique, pourcentage moins élevé du bénéfice versé en dividendes, ratio cours-bénéfice plus élevé, cours plus instables.
3. Capital de risque	- Risque élevé, capital-actions peu important, peu de bénéfices réalisés, aucun dividende versé, activités récentes, ratio cours- bénéfice sans importance et cours instables.
4. Valeurs spéculatives	- Risque maximum, terme plus court, plus grande instabilité des cours, aucun bénéfice déclaré, aucun dividende versé, ratio cours-bénéfice sans importance.

Tableau 7 : Risque selon les catégories d'avoir

5.5.2.1. Structure d'une solution

Le système GIPF présente les solutions aux problèmes sous forme de portefeuilles financiers. La figure suivante décrit la structure d'une solution.



Portefeuille		
Pourcentages des diverses catégories de titres		
	% du portefeuille	Montant
Espèces, quasi-espèces (15% - 20%)		
Emissions des gouvernements	4	
Emissions des sociétés	11	
Total	15	
Valeurs à revenu fixe (40% - 60%)		
Court terme	13	
Moyen terme	17	
Long Terme	21	
Total	51	
Actions (20% - 40%)		
Valeurs sûres	9	
Valeur d'avenir	8	
Capital de risque	8	
Valeurs spéculatives	9	
Total	34	
Total du portefeuille	100	

Figure 15 : Structure d'une solution

5.5.3. Modèle de Mémoire pour la représentation de la base de cas

La base de cas est représentée par une arborescence. Les feuilles de l'arbre étant les pages constituant la base de cas.

Chaque nœud de l'arborescence pointe soit sur un autre nœud, soit sur une page dans la base de cas. A chaque cas correspond une adresse virtuelle qui permet de lui affecter un numéro de page où il sera stocké.

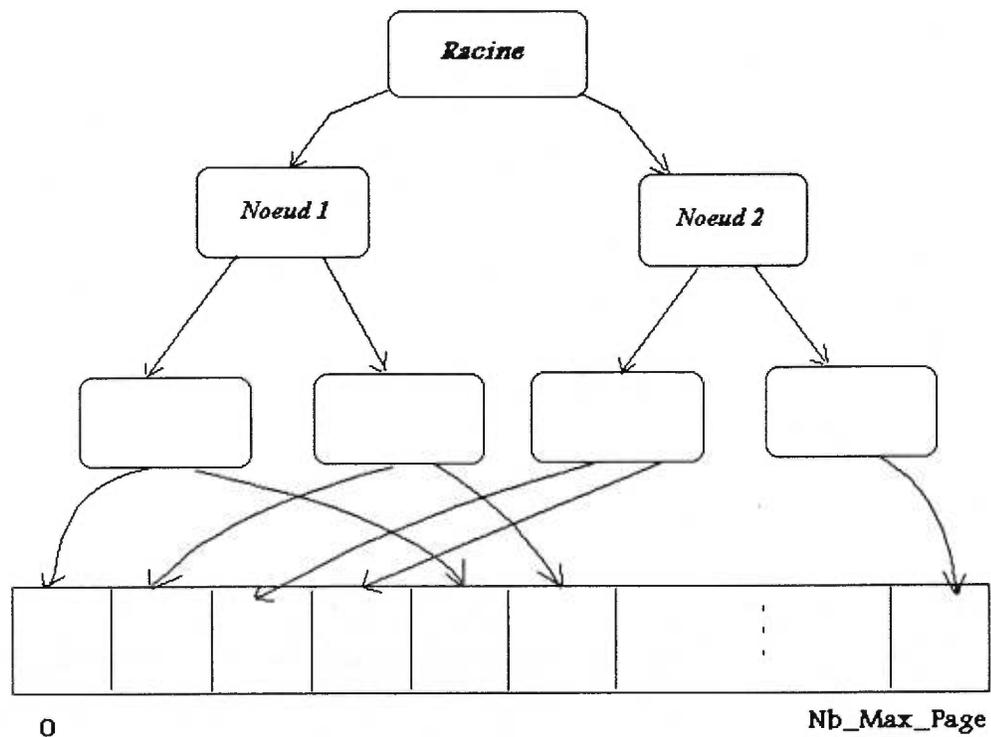


Figure 16 : Modèle de mémoire pour la base de cas

A chaque fois qu'un nouveau cas se présente, le système calcule son adresse virtuelle. Ensuite la fonction de hashage va nous renvoyer son numéro de page. S'il y a assez d'espace libre dans la page pour insérer le cas, ce dernier y sera stocké. Dans le cas contraire, la page est éclatée en deux pages distinctes, et nous rajoutons un nœud dans l'arbre pour descendre d'un niveau.

La fonction de hashage recalcule le nouveau numéro de page de chaque cas en utilisant la nouvelle adresse virtuelle. Une partie des cas sera stockée dans une première page et une autre partie se retrouvera dans une deuxième page. En dernière étape, le cas qui a provoqué l'éclatement de la page sera inséré dans la première ou la deuxième page, en fonction de son adresse virtuelle.

La figure suivante présente un exemple de contenu de la base de cas du système.

Nom	Prenom	Age	E.Civil	Enfants	Revenu	Risque
Aubin	Richard	32	C	0	34000	26
Beaulieu	Denise	56	M	0	45000	34
Bessette	Nathalie	30	C	0	45000	47
Boisvert	Pierre	46	C	0	175000	22
Bonneau	Richard	45	M	0	150000	34
Bourdy	Christian	46	M	0	100000	29

Nombre de clients trouves : 91

Figure 17 : Base de Cas

Après avoir soumis une requête au système, le fait de cliquer sur le bouton résultat va afficher tous les cas de la base qui sont les plus proches du cas courant. Pour avoir plus d'informations concernant un cas en particulier dans la liste, on a juste à double-cliquer sur le champs « nom ». Le bouton « Plus-Proche » sert à avoir à l'écran le cas de la liste qui est le plus proche du cas courant.

5.5.4. Mémoire cache

Afin d'optimiser le temps d'accès aux cas, le système dispose de tampons qui serviront de mémoire cache. La taille d'un tampon est la même que celle d'une page disque de la base de cas. Lors de la recherche d'un cas, le système commence par calculer son numéro de page pour, ensuite parcourir la mémoire cache afin de vérifier si cette page n'a pas été chargée au préalable.

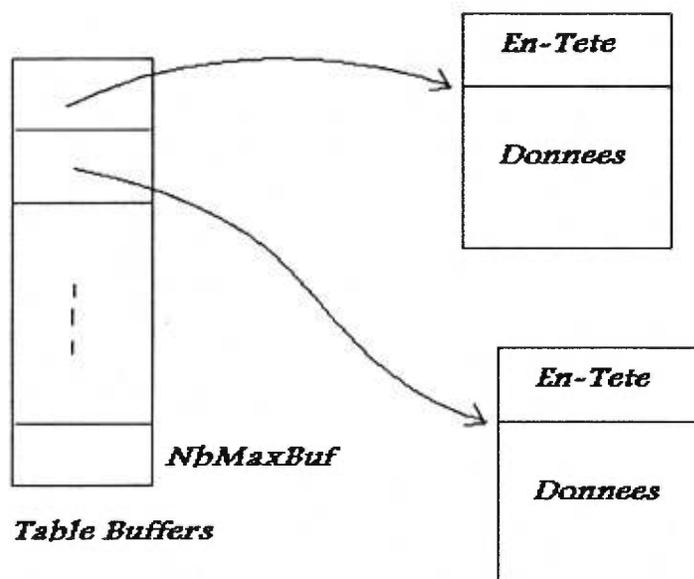


Figure 18 : Mémoire Cache

Un tampon cache est constitué de deux parties. Une partie en-tête qui contient le numéro de la page chargée dans ce tampon, un indicateur d'occupation ainsi qu'un compteur temporel d'accès à la page qui y est logée.

L'autre partie du tampon contient les données qui représentent le contenu de la page disque elle-même. Lorsque le système veut accéder à un cas, il calcule son numéro de page à partir de son adresse virtuelle, ensuite il parcourt la mémoire cache afin de

chercher le tampon contenant la page disque concernée. Si cette page a déjà été chargée en mémoire, le système retournera le numéro de tampon la contenant. Dans le cas contraire, il cherche un tampon libre afin de la charger à partir du disque. Si le système ne trouve pas de tampon libre, il choisit un tampon, en appliquant la politique du moins récemment utilisé (LRU). Ce tampon est libéré et sera utilisé pour contenir la nouvelle page à charger à partir du disque.

5.6. Indexation dans GIPF

Dans le système GIPF, chaque champ figurant dans un article a un poids qui représente l'importance de cette dimension relativement aux autres. L'adressage virtuel des cas, a comme conséquence la création d'une hiérarchie d'index. L'index de plus haut niveau sera celui relatif au champ de plus haut poids. Nous obtenons une hiérarchie d'index allant de pair avec la hiérarchie créée par les poids des champs.

La figure suivante montre à l'aide d'un exemple, un poids de 20% et un voisinage de 10%.

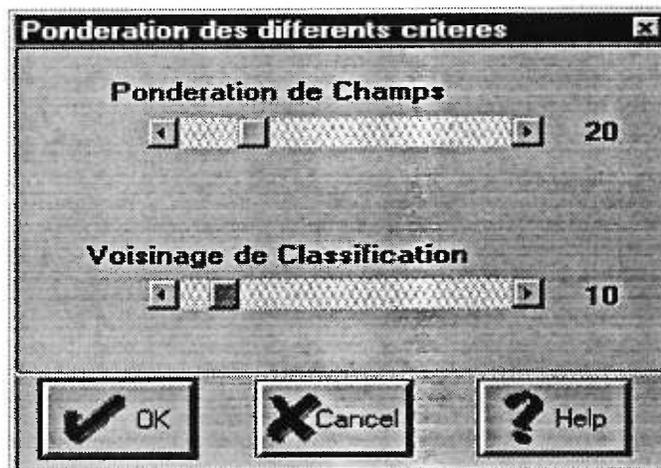


Figure 19 : Pondération et voisinage

5.6.1. Algorithme d'indexation de cas

L'algorithme d'indexation est composé de plusieurs étapes :

- Lire le cas q en entrée
- Choisir dans chaque champ (attribut), un nombre de bits proportionnel au poids de cet attribut pour obtenir une adresse virtuelle
- Déterminer le nœud N de l'arbre qui contient des cas similaires au cas q
- Récupérer le numéro de page physique p contenu dans ce nœud N
- Chercher cette page dans la mémoire cache
- **Si** p n'existe pas dans la mémoire cache
 Alors Charger p à partir du disque
- Finsi**
- **Si** *espace libre dans la page p insuffisant pour insérer le cas q*
 Alors *Éclater la page p en deux*
 Reindexer le contenu des deux demi-pages
 Insérer le cas q dans la demi-page correspondante
- Finsi**

5.7. Recherche de cas

Dans le système GIPF, chaque champ apparaissant dans un cas présente un voisinage exprimé en pourcentage. En d'autres termes, pour une valeur v_i d'un champ particulier, toutes les valeurs v_j dont la distance par rapport à v_i est inférieure à un certain seuil δ_i , sont dans le voisinage de v_i . La distance séparant les valeurs v_i et v_j est la valeur absolue de la différence.

Deux cas dont les valeurs de champs sont dans le même voisinage, deux à deux, font partie de la même classe de cas. Cependant, il se peut que deux cas n'aient pas tous leurs champs dans le même voisinage. Il faudra alors, tenir compte de la pondération des champs. La distance séparant un cas c d'un cas q est calculée comme suit :

$$Dis(c,q) = \sum_{a=1,n} w_a * dis_a(q_a, c_a)$$

avec w_a : poids de l'attribut a .

et dis_a : La distance locale pour l'attribut a .

On dira que le cas q_k est similaire au cas c si on a $dis(c,q_k) < \delta$. Lorsqu'un nouveau cas c se présente, le système cherche dans la base les n cas q_k qui vérifient la condition $dis(c,q_k) < \delta$.

Ensuite, il retient le cas q_k qui sera le plus proche du nouveau cas c , donc celui dont la distance par rapport au cas c est égale à $\min \{dis(c,q_k)\}_{k=1,n}$.

5.7.1 Algorithme de recherche de cas

L'algorithme de recherche se compose des étapes suivantes :

- Lire le cas q en entrée
- Choisir dans chaque champ (attribut), un nombre de bits proportionnel au poids de cet attribut pour obtenir une adresse virtuelle
- Déterminer le nœud N de l'arbre qui contient des cas similaires au cas q
- Récupérer le numéro de page physique p contenu dans ce nœud N
- Chercher cette page dans la mémoire cache
- Si p n'existe pas dans la mémoire cache
 Alors Charger p à partir du disque
 Finsi
- Lire le premier cas c de la page p
- Initialiser cas_retenu à c
- Initialiser $similarite_max$ à 0
- Tant que Non fin de page
 Calculer le degré de similarité entre c et q : $similarite(c,q)$
 Si $similarite(c,q) > similarite_max$
 alors $similarite_max = similarite(c,q)$
 $cas_retenu = c$
 Finsi
- FinTantque
- Récupérer la solution s associée au cas cas_retenu
- Adapter la solution s . (voir l'algorithme d'adaptation en 5.8.8)

5.8. Adaptation de cas

Dans cette section, nous présentons le processus d'adaptation à base de règles ainsi que le processus d'adaptation utilisant un raisonnement à base de cas. Nous y abordons également l'approche retenue pour adapter les solutions dans le système GIPF.

5.8.1. Processus automatique d'adaptation de cas

L'adaptation de cas automatique permet à un système CBR de fonctionner de façon autonome et le rend utilisable aussi bien par un utilisateur expert du domaine d'application que par un utilisateur novice. La majorité des systèmes CBR ont un processus d'adaptation à base de règles.

Dans notre approche, le processus d'adaptation est décomposé en deux parties. La première étant l'ensemble des transformations appliquées à un cas et la seconde étant le processus de recherche des informations en mémoire.

L'apprentissage de nouvelles stratégies d'adaptation présente une conséquence intéressante pour le traitement de la similarité. Dans les systèmes CBR, la similarité est généralement basée sur des critères fixes.

Comme le système CBR apprend comment adapter les cas pour traiter de nouveaux types de problèmes, la fonction de similarité devrait être ajustée afin de refléter les changements de l'état des connaissances dans le processus d'adaptation.

L'approche CBR peut être appliquée de façon **réflexive** au processus d'adaptation dans un système CBR. Le système peut garder en mémoire une trace du processus d'adaptation pour un cas particulier afin de le réutiliser dans le futur [Leake et al, 96].

Un cas d'adaptation serait donc constitué du type de transformations à appliquer (Comment transformer le cas d'adaptation le plus proche ?) et du type de recherche en mémoire (Comment retrouver le cas d'adaptation le plus similaire?).

Le codage de règles d'adaptation nécessite une connaissance approfondie du domaine d'application. Cette connaissance n'est pas toujours évidente à traduire en terme de règles. Le même type de problème est rencontré lors du développement de systèmes experts (systèmes à base de règles de production).

Pour un système CBR, il est donc naturel de penser à utiliser l'approche CBR pour remplacer les règles d'adaptation prédéfinies par un processus d'adaptation à base de cas.

Une des difficultés de cette approche réside dans le fait de définir un noyau initial. Au départ, on a besoin d'un certain nombre de règles, indépendantes du domaine d'application, qui permettraient de faire les premières adaptations. Ces dernières seront mémorisées comme des cas d'adaptation et seront utilisées pour les adaptations futures. Ces règles d'adaptation indépendantes du domaine d'application présentent l'inconvénient d'être difficilement applicables, car trop générales. De part leur généralité, un ensemble réduit de règles d'adaptation permet de couvrir un grand nombre de cas d'adaptation. D'autre part, les règles plus spécifiques, donc liées au domaine d'application, sont faciles à appliquer, mais présentent une généralité limitée et sont difficiles à définir car elles font appel à la connaissance du domaine d'application. Cependant, malgré la difficulté de définition des règles spécifiques, nous avons retenu cette approche pour tirer profit de leur précision dans le processus d'adaptation.

5.8.2. Adaptation dans le système GIPF

Le processus d'adaptation pour le système GIPF combine une approche à base de règles avec un raisonnement à base de cas afin d'élaborer une expertise dans l'adaptation de cas. Le but de cette approche est de permettre au système d'assurer la transition de

l'adaptation à l'aide de règles vers une adaptation guidée par un raisonnement à base de cas. Autrement dit, au lieu d'avoir un ensemble de règles que l'on réappliquerait à chaque fois que l'on doit adapter une solution, on mémorise les adaptations de solutions dans une base de cas pour une réutilisation ultérieure.

5.8.3. Base de cas d'adaptation

La base de cas du système GIPF est un ensemble de couples $\langle C_i, S_i \rangle$, C_i étant les cas et S_i les solutions associées à ces cas.

Chaque fois qu'un nouveau cas se présente, le système récupère le cas le plus similaire C_i se trouvant dans sa base et adapte la solution S_i en utilisant des règles que nous détaillerons plus loin.

La solution S_i' est le résultat de la transformation de S_i et est proposée par le système comme solution pour le nouveau cas à traiter. En dernière étape, le système stocke le couple $\langle S_i, S_i' \rangle$ dans la base d'adaptation.

L'indexation dans la base d'adaptation se fait sur :

- La distance qui sépare le cas associé à la solution S_i et le nouveau cas à traiter.
- Les attributs de la solution S_i .

La figure 20 présente un exemple de portefeuilles financiers adaptés. Chaque ligne de la liste représente un portefeuille. La première colonne représente les titres d'émissions du gouvernement, la seconde correspond aux émissions de compagnies. La colonne « C.T » représente les obligations à court terme. Le sigle « M.T. » correspond aux obligations à moyen terme alors que la colonne « L.T. » renferme les obligations à long terme. Le mnémonique « V.Sr. » correspond aux valeurs sûres alors que « V.A » est pour les valeurs d'avenir. Les colonnes « V.R » et « V.Sp. » correspondent respectivement aux valeurs à risque et aux valeurs spéculatives. La dernière colonne renferme la différence

séparant le cas pour lequel la solution a été adaptée du cas le plus proche que le système a trouvé.

Base d'Adaptation

Base d'Adaptations

Infos Portefeuilles

Emg	Ems	C.T.	M.T.	L.T.	V.St.	V.A.	V.R.	V.Sp.	Diff.
4	14	13	14	20	9	9	9	8	5000
4	14	13	14	20	9	9	9	8	7000
4	14	13	14	20	9	9	9	8	1000
4	14	13	14	20	9	9	9	8	2000
4	14	13	14	20	9	9	9	8	3000
2	12	11	19	21	8	9	9	9	5015

Nombre de cas trouves : 8

OK Resultat PortAdapte Plus-Proche

Figure 20 : Base de cas d'adaptation

Dans les sections suivantes, nous présentons un exemple d'adaptation de portefeuille financier. Pour un client âgé de 35 ans célibataire et qui a un revenu de 140000\$ avec une tolérance au risque de 66%, on dispose d'un portefeuille financier dont la structure est détaillée dans la figure 21.

Portefeuille

Pourcentages des diverses categories de titres

	% du portefeuille	Montant
Especes, quasi-especes (15% - 20%)		
Emissions des gouvernements	4	
Emissions des societes	14	
Total	18	
Valeurs a revenu fixe (40% - 60%)		
Court terme	13	
Moyen terme	14	
Long Terme	20	
Total	47	
Actions (20% - 40%)		
Valeurs surs	9	
Valor d'avenir	9	
Capital de risque	9	
Valeurs speculatives	8	
Total	35	
Total du portefeuille	100	

OK
Cancel
Help

Figure 21 : Exemple de portefeuille financier

Supposons que la tendance des taux d'intérêt soit prévue à la baisse, comme l'indique la figure suivante.

Prevision des taux d'interets

Taux d'interets hauts

Taux d'interets bas

OK
Cancel
Help

Figure 22 : Tendance à la baisse des taux d'intérêt

Si le système doit traiter un nouveau client âgé de 32 ans, qui soit célibataire et qui ait un revenu de 135000\$ avec une tolérance au risque de 70%, il adapterait le portefeuille présenté précédemment pour obtenir un portefeuille adapté au nouveau client.

Le système modifie le pourcentage de titres à revenu fixe à court terme ainsi que celui des titres à revenu fixe à moyen et long terme. Les valeurs à revenu fixe à court terme passent de 13% à 11%. Celles qui sont à moyen terme passent de 14% à 12%. Enfin les titres à long terme passent de 20% à 24%. Le fait que la tendance des taux d'intérêt soit prévue à la baisse intervient dans la modification des échéances des titres à revenu fixe. Le portefeuille résultant du processus d'adaptation est décrit dans la figure 23.

Portefeuille		
Pourcentages des diverses categories de titres		
	% du portefeuille	Montant
Especes, quasi-especes (15% - 20%)		
Emissions des gouvernements	4	
Emissions des societes	14	
Total	18	
Valeurs a revenu fixe (40% - 60%)		
Court terme	11	
Moyen terme	12	
Long Terme	24	
Total	47	
Actions (20% - 40%)		
Valeurs sures	9	
Valeur d'avenir	9	
Capital de risque	9	
Valeurs speculatives	8	
Total	35	
Total du portefeuille	100	

Figure 23 : Portefeuille financier adapté

5.8.4. Indexation des cas d'adaptation

Dans la base d'adaptation, les couples $\langle S_i, S_i' \rangle$ sont indexés sur les différents champs de la solution S_i .

Il y a autant de champs dans la solution S_i que de types de titres financiers. Cependant, il y a un champ supplémentaire qui correspond à la distance séparant le cas C_i du nouveau cas à traiter. De la même façon que pour la base de cas, les champs des éléments de la base d'adaptation ont des pondérations différentes.

On a donc une hiérarchie d'index, comme pour la base de cas, correspondant à la hiérarchie établie par les poids des champs de la solution S_i .

5.8.5. Distance utilisée dans la base de cas d'adaptation

Comme pour la base de cas, chaque attribut d'une solution dans la base d'adaptation a un voisinage dans lequel les valeurs sont similaires.

De la même façon que pour la base de cas, la distance séparant les différentes valeurs d'un champ composant un cas de la base d'adaptation, est la valeur absolue de la différence.

La distance séparant deux cas de la base d'adaptation est également calculée en tenant compte de la pondération des différents champs.

5.8.6. Règles d'adaptation de cas pour le système GIPF

La constitution d'un portefeuille financier prend en considération deux volets. Le premier aspect représente la situation et les objectifs financiers du client, le deuxième aspect la conjoncture économique.

Lorsque le système propose une solution pour un cas, il se base sur la solution associée au cas le plus proche du cas courant, qu'il trouve dans sa base de cas. D'autre part, l'adaptation de solution se fait en deux volets. Le premier concerne les informations décrivant le cas tandis que le second traite des données relatives au contexte économique

courant. L'adaptation de cette solution va se faire d'une part, en utilisant des règles relatives au contexte économique et d'autre part, en utilisant des règles relatives à la distance séparant le cas courant du cas retrouvé dans la base de cas.

Règles relatives au contexte économique

R1 :

*Si les taux d'intérêt sont prévus à la hausse **alors** réduire l'échéance des valeurs à revenu fixe **finsi***

R2 :

*Si les taux d'intérêt sont prévus à la baisse **alors** prolonger l'échéance des valeurs à revenu fixe **finsi***

L'adaptation d'une solution doit tenir compte du contexte économique et en particulier des taux d'intérêt. La modification d'un portefeuille financier se fait en fonction de paramètres qui ne sont pas liés à la situation d'un client en particulier. Ces paramètres reflètent la situation économique courante. Les taux d'intérêt représentent une composante importante de la situation économique. Cependant ce n'est pas le seul facteur important à prendre en considération. Les prévisions de variation des taux d'intérêt sont introduites dans le système par le biais de la boîte de dialogue présentée dans la figure 22.

Règles relatives à la situation du client

R3:

*Si les objectifs du client sont le revenu et la sécurité **alors** réduire le pourcentage d'actions et augmenter les titres à revenu fixe **finsi**.*

R4 :

Si les objectifs du client sont la plus-value (augmentation de la valeur de ses avoirs) et le client a une meilleur tolérance au risque alors augmenter le pourcentage d'actions finsi.

R5 :

Si le client est plus jeune alors augmenter le pourcentage d'actions (tolérance au risque plus grande, meilleur rendement) et réduire le pourcentage de titres à revenu fixe finsi.

R6 :

Si le revenu du client est plus important alors réduire le pourcentage de titres à revenu fixe (le but est d'alléger le fardeau fiscal du client en réduisant son taux d'imposition) et réduire le pourcentage d'actions finsi.

Dans un premier temps, les règles énumérées ci-dessus sont utilisées par le système GIPF afin de réaliser les premières adaptations de solutions. Les transformations appliquées lors de ces premières adaptations seront stockées dans la base de cas d'adaptation. Cette dernière sera enrichie au fur et à mesure que de nouvelles solutions à adapter se présentent.

5.8.7. Processus d'adaptation dans le système GIPF

Le processus d'adaptation dans le système GIPF va commencer avec une ensemble de règles *ER* constitué d'un noyau de règles $\{R_i\}$ liées au domaine d'application.

Lorsqu'un nouveau cas *nouvC* se présente, le système lui trouve le cas le plus similaire *Ci*. Ce dernier est stocké dans la base de cas avec sa solution *Si* qui doit être

adaptée afin de tenir compte de la différence qui existe entre le cas courant $nouvC$ et le cas Ci .

En deuxième étape, le système va chercher dans l'ensemble de règles ER , la règle qui correspond au type d'adaptation de la solution Si .

Par exemple si le cas courant $nouvC$ et l'ancien cas retrouvé Ci présentent une différence au niveau de l'attribut qui représente l'âge du client, le système peut utiliser la règle Rj suivante :

***Si** client est plus jeune **alors** augmenter le pourcentage d'actions (tolérance au risque plus grande) et réduire le pourcentage de titres à revenu fixe **finsi**.*

Une fois que la règle d'adaptation sélectionnée dans l'ensemble de règles ER est appliquée à la solution Si , on obtient la solution adaptée Si' . Cette dernière sera stockée dans la Base d'adaptation BA sous forme de couple (Si, Si') .

5.8.8. Algorithme d'adaptation

L'algorithme d'adaptation est constitué des étapes suivantes :

- Lire le cas $nouvC$ en entrée
- Trouver le cas le plus proche Ck (voir algorithme de recherche de cas)
- Récupérer la solution Sk , associée au cas Ck
- Chercher une règle d'adaptation dans l'ensemble de règles ER .
- **Si** règle correspondant au type de différence qui existe entre $nouvC$ et Ck
- Alors**
- Appliquer règle à Sk pour obtenir Sk'
- Insérer le couple (Sk, Sk') dans la base d'adaptation (BA)
- Retourner la solution adaptée Sk'
- Sinon**
- Trouver la solution Sk' la plus proche de la solution Sk dans la base d'adaptation
- Récupérer la solution Sk'' trouvée dans le couple (Sk', Sk'')
- Retourner la solution Sk''
- Finsi**

La solution adaptée S_k'' sera proposée par le système comme le résultat de l'adaptation de la solution S_k .

L'utilité d'une approche CBR dans le processus d'adaptation est évidente quand on sait que le noyau de règles d'adaptation est incomplet. D'autre part, un système d'adaptation à base de règles est impuissant devant une situation non prévue par ses règles.

5.8.9. Exemple d'adaptation :

Dans cette section, nous présentons un exemple d'adaptation afin d'illustrer ce qui a été présenté précédemment.

Soit un cas c décrit par les attributs suivants :

$$c = \{ \text{Age, État civil, Salaire, Tolérance au risque, Objectifs financiers} \}$$

Soit la description de solution suivante :

$$S = \{ \text{Espèces, Court terme, moyen terme, long terme, valeurs sûres, valeurs d'avenir, capital de risque, valeurs spéculatives} \}$$

Soit un premier cas $Cas1$:

$$Cas1 = \{ 38, \text{célibataire}, 120000, 60\%, \text{sécurité} \}$$

Soit $S1$ la solution associée au cas $Cas1$:

$$S1 = \{ 14\%, 11\%, 19\%, 21\%, 8\%, 9\%, 9\%, 9\% \}$$

Avec un deuxième cas à traiter $Cas2$:

$$Cas2 = \{ 36, \text{célibataire}, 115000, 55\%, \text{sécurité} \}$$

Le système trouve *Cas1* comme le cas le plus proche de *Cas2* et va adapter la solution *S1* en appliquant les règles *R2* et *R5* (voir 5.8.6) pour obtenir *S2*:

$$S2 = \{ 14\%, 8\%, 19\%, 20\%, 9\%, 10\%, 10\%, 10\% \}$$

S2 est donc la solution proposée pour *Cas2*.

Le système stocke par la suite, le couple (*S1*, *S2*) dans la base d'adaptation en l'indexant sur les attributs de *S1* et sur la distance entre *Cas2* et *Cas1*. Notons que la pondération de l'attribut qui représente la distance entre *Cas1* et *Cas2* est beaucoup plus importante que celle des autres attributs.

Pour traiter un troisième *Cas3* = { 35, célibataire, 117000, 63%, sécurité } , le système va retrouver *Cas1* = { 38, célibataire, 120000, 60%, sécurité } , comme le cas le plus proche et va mettre à contribution l'adaptation précédente de la solution *S1*.

La distance séparant le *cas3* du *cas1* va servir à faire la recherche dans la base d'adaptation. Si le système retourne le couple (*S1*, *S2*), c'est que la distance entre *Cas2* et *Cas1* est proche de la distance séparant *Cas3* du *Cas1*. Par conséquent, *Cas3* et *Cas2* sont assez similaires pour pouvoir utiliser *S2* comme solution adaptée pour *Cas3*.

Le chapitre suivant présente les conclusions de ce travail ainsi que les perspectives futures et améliorations possibles.

6 Conclusion

L'approche CBR, bien qu'assez jeune, a prouvé son efficacité par le biais des différents systèmes fonctionnant dans l'industrie depuis ces dernières années. La chose la plus importante à souligner concernant ces systèmes, c'est que leur façon de résoudre les problèmes est fondamentalement différente de celle des systèmes experts classiques qui fonctionnent avec une base de règles et un moteur d'inférence.

Les partisans des deux approches, systèmes experts d'un côté et CBR de l'autre, ont certainement des arguments en faveur de l'un ou l'autre type de systèmes. Cependant l'approche CBR présente un avantage certain qui est celui de rajouter à la résolution de problèmes, un processus d'apprentissage en retenant une nouvelle expérience à chaque fois qu'un problème est résolu et en la rendant immédiatement disponible pour résoudre des cas semblables dans le futur.

6.1. Approche retenue

L'approche retenue pour la description des cas dans notre système fut d'abord l'établissement de pondération liée à l'importance relative de chaque caractéristique décrivant un cas. Ceci a permis, par la suite, l'organisation de la mémoire des cas en structure hiérarchique découlant d'un mécanisme de discrimination opérant sur les rubriques à fort pouvoir de discrimination. Cette structure de mémoire évolue au fur et à mesure que de nouveaux cas arrivent. Son organisation est donc dynamique et la hiérarchie se construit à chaque fois qu'un sous-ensemble de cas explose en fonction de la discrimination introduite par l'analyse d'une nouvelle caractéristique.

Pour traiter un nouveau problème, la fonction de recherche commence par déterminer le sous-ensemble de cas susceptible de contenir des éléments proches du problème courant. Ensuite, une analyse de ce sous-ensemble est faite par la fonction de similarité qui se base sur un degré de similarité composé. C'est-à-dire que pour chaque élément de cet ensemble, le degré de similarité est quantifié en tenant compte du poids de chaque champ et de la distance le séparant du cas courant. Une fois le cas présentant le plus haut degré de similarité retenu, la fonction d'adaptation modifie la solution trouvée afin de refléter les différences qui existent entre le problème et le cas retrouvé. Le processus d'adaptation utilise également une organisation de mémoire hiérarchique pour stocker les adaptations réalisées au fur et à mesure que le système évolue. Les adaptations sont mémorisées dans une base de cas en étant indexées sur les différentes composantes de la solution ainsi que sur le degré de similarité composé calculé lors de la recherche de cas.

6.2. Limites du système

La distance séparant un problème du cas retrouvé se traduit par le calcul du degré de similarité qui est la combinaison des degrés de similarité existant entre les attributs décrivant les deux cas, pris deux à deux. Cette distance composée est établie en tenant compte de critères comme le voisinage et la pondération associés à chaque attribut apparaissant dans un cas. Or deux valeurs très proches pour cette valeur composée (voir 5.4) pourraient être associées à deux types de différences complètement distinctes.

D'autre part, lorsqu'un analyste financier propose un portefeuille à un client, il prend en considération les données qui sont propres à ce dernier ainsi que les informations liées au contexte économique. L'expert évalue les indicateurs économiques comme la tendance des taux d'intérêt, le taux de chômage, etc. Par conséquent, dans deux contextes économiques différents, deux solutions tout à fait distinctes pourraient être proposées à un **même** client. La situation devient également assez complexe quand l'expert se met à interpréter des indicateurs politiques liés à l'économie afin de faire des prévisions. Dans

notre système, nous avons intégré la notion de tendance des taux d'intérêt. Ce qui n'est évidemment pas suffisant.

6.3. Améliorations possibles

Une extension intéressante serait de rajouter un module qui garde une trace des échecs potentiels du système. Par exemple, à chaque fois que le système tente d'adapter une solution, il pourrait attendre la validation d'un expert qui lui expliquerait pourquoi l'adaptation n'a pas fonctionné. De cette manière, le système ne reproduirait pas les mêmes erreurs dans le futur et apprendrait de ses échecs.

On pourrait également rajouter un module qui servirait d'outil de définition de nouvelles règles d'adaptation. Ce serait donc un éditeur qui permettrait de modifier et d'enrichir l'ensemble de règles qui servent au processus d'adaptation dans ce système.

Par ailleurs, une amélioration importante pourrait se faire au niveau du choix du représentant d'une classe. Quand le système ne trouve pas un cas proche du cas à traiter, il garde ce nouveau cas comme représentant d'une nouvelle classe, cette situation est donc relativement simple. Cependant, si le système trouve un cas proche dans sa base, comment va-t-il décider s'il faut garder l'ancien ou le nouveau cas comme représentant de cette classe ?

Une solution à explorer serait de chercher les deux cas les plus proches du cas retrouvé qui encadrent ce dernier, l'un à gauche et l'autre à droite. Dans une deuxième étape, on calcule la distance séparant le cas gauche du cas droit. Enfin, entre le nouveau cas et le cas retrouvé, le système retient, comme représentant de la classe, celui qui se rapproche le plus du milieu de l'intervalle [cas gauche, cas droit].

6.4. Conclusion

Le système GIPF a été testé sur une base d'une centaine de cas et a atteint les objectifs fixés au début de ce travail. Les choix qui ont été faits pour les données reliées à la planification financière auraient été pratiquement impossibles à établir sans l'aide précieuse d'experts dans le domaine, Monsieur Benakezou, gestionnaire de portefeuille au sein du groupe *La Mutuelle*, Monsieur Gagné, CFA et gestionnaire de portefeuille pour l'institution *Interinvest corporation* et enfin Monsieur Steele, responsable des obligations pour *Interinvest corporation*. Cela couvre aussi bien la pondération des caractéristiques décrivant les cas que les types de solutions à proposer pour ces cas.

Le système que nous avons implanté est une mise en œuvre de toutes les facettes d'un système de raisonnement à base de cas, avec un module de recherche de cas, une partie d'indexation, l'implémentation d'une fonction de similarité ainsi qu'un module d'adaptation de solutions.

L'originalité de ce travail réside dans l'utilisation de l'approche CBR de manière récursive. Effectivement le CBR est utilisé de façon récursive, aussi bien pour retrouver des cas similaires que pour retrouver une ancienne adaptation de solutions. En d'autres termes, à chaque fois qu'une solution est adaptée, une trace de cette opération est conservée de façon à être utilisée pour une adaptation future. En fait, le processus d'adaptation présente une approche mixte qui utilise aussi bien le raisonnement à base de cas que celui à base de règles.

Par ailleurs, notre système implémente une hiérarchie d'index et une structure particulière de la mémoire des cas afin d'intégrer la notion de similarité dans le processus d'indexation.

Le système dispose également d'un utilitaire d'importation de données à partir d'une base de données dBase. Cela permet d'injecter dans le système CBR des informations provenant d'une base de données au lieu d'y insérer les cas un par un.

Nous terminerons en soulignant que les processus d'indexation, de recherche de cas et d'adaptation dans le système GIPF sont conçus de façon à être facilement réutilisables si nous voulons changer de domaine d'application. Cela confère à notre système un certain degré de généralité.

Références Bibliographiques

- [Aha, 91] Aha, D. W., "Case-Based Learning Algorithms.", *Proceedings of the DARPA Case-Based Reasoning Workshop, 147-158*. San-Fransisco. Edit. By E.R. Bareiss. S.F.: Morgan Kaufmann Publishers. 1991.
- [Althoff et al, 95] Althoff, K. D., Auriol, E., Bergmann, R., Breen, S., Dittrich, S., Johnston, R., Manago, M., Traphoner, R., Wess, S., "Case-Based Reasoning for Decision Support and Diagnostic Problem Solving: The INRECA Approach", University of Kaiserslautern, 1995.
- [Aleven et Ashley, 92] Aleven, V., et Ashley, K. D., "Automated Generation of examples for a Tutorial in Case-Based Argumentation", *Proceedings of The Second International Conference on Intelligent Tutoring Systems (ITS 92)*, 575-584. Edit. By C. Frasson; G. Gauthier; G.L. McCallan. Berlin: Springer-Verlag. Montréal, 1992.
- [Allemang, 93] Allemang, D., "Case-Based Reasoning Newsletter 2(3)", *Review of the First European Workshop on CBR*, Kaiserslautern, Germany, 1993.
- [Alterman, 86] Alterman, R., "An Adaptive Planner.", *Proceedings of AAAI-86*, 65-69. Cambridge, MA: AAAI Press/ MIT Press, 1986.
- [Anderson, 83] Anderson, J.R., "The architecture of cognition", Harvard University Press, Cambridge, 1983.

- [Ashley, 88] Ashley, K.D., "Arguing by Analogy in Law: A Case-Based Model", *Analogical Reasoning: Perspectives of Artificial Intelligence, Cognitive Science, and Philosophy*. Norwell, MA: D. Reidel. 1988.
- [Bain, 86] Bain, W.M., "Case-Based Reasoning: A computer-Model of Subjective Assessment", *Ph. D. Thesis*. Yale University. 1986.
- [Barletta, 94] Barletta, R., "A Hybrid Indexing and Retrieval Strategy for Advisory CBR Systems Built with REMIND". *Proceedings of the Second European Workshop on Case-Based Reasoning*, 49-58. Chantilly, France. 1994.
- [Berger, 95] Berger, J., "Roentgen : A Case-Based Radiation Therapy Planner", *Ph.D. diss.*, Computer Science Dept., University of Chicago. 1995.
- [CCVM, 95] "Cours sur le Commerce des Valeurs Mobilières au Canada", *Institut Canadien des Valeurs Mobilières*, 1995.
- [Collins, 87] Collins, G., "Plan Creation: Using Strategies as Blueprints", *Ph.D. thesis*. Departement of Computer Science, Yale University. 1987.
- [Costas et Kashyap, 93] Costas, T., Kashyap, P., "Case-Based Reasoning and Learning in Manufacturing with TOTLEC Planner." *IEEE Transactions on Systems, Man, and Cybernetics*. July/August 1993.
- [Farrel, 87] Farrel, R., "Intelligent Case Selection and Presentation", *Proceeding of the Tenth International Joint Conference on Artificial Intelligence*. IJCAI-87,1,74-76. Milan, Italie 1987.
- [Goodman, 89] Goodman, M. , "CBR in Battle Planinng", *Proceeding of the DARPA Case-Based Reasoning Workshop*, 264-269. Edit K.J. Hammond. San F.: Morgan Kaufmann Publishers. 1989.

- [Hammond, 86] Hammond, K. J. ,“CHEF: A Model of Case-Based Planning”, *Proceedings of the AAAI, August 86, 267-271*. Menlo Park, Calif. Cambridge, MA: AAAI Press/ MIT Press. 1986.
- [Hinrichs, 89] Hinrichs, T. ,“Strategies for Adaptation and Recovery in a Design Problem Solver”, *Proceedings of the DARPA Case-Based Reasoning Workshop, 115-118*. Pensacola Beach, Florida. Edit. By K.J. Hammond. San Francisco: Morgan Kaufmann Publishers. 1989.
- [Kass, 89] Kass, A. M., “Strategies for Adapting Explanations”, *Proceedings of the DARPA Case-Based Reasoning Workshop, 119-123*. Edit. By K.J. Hammond. San Francisco:Morgan Kaufmann Publishers. 1989.
- [Kolodner, 83] Kolodner, J. L., “Maintaining Organization in a Dynamic Long-Term Memory”, *Cognitive Science*,7, 243-280. 1983.
- [Kolodner, 93] Kolodner, J. L., “Case-Based Reasoning”. Morgan Kaufmann Publishers. 1993.
- [Koton, 89] Koton, P., “Using Experience in Learning and Problem Solving”, *Ph.D. Thesis, Laboratory of Computer Science, Massachusetts Institute of Technology*. P. MIT/LCS/TR-441. 1989.
- [Laird et al., 86] Laird, J. E., Rosenbloom, P. S., Newell, A., “Chunking in SOAR : The anatomy of a general learning mechanism”, *Machine learning 1:11-46. Reading in Machine Learning*. Eds. Shavlik. J. W., et Dietterich, 555-572. San fransisco, Calif. : Morgan Kaufmann.
- [Leake, 94] Leake, D. B. , “Toward a Computer Model of Memory Search Stategy Learning.” *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society, 549-554*. Hillsdale, N.J. 1994.

- [Leake, 95] Leake, D. B., "Combining Rules and Cases to Learn Case Adaptation" Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society, 84-89. Ann Arbor, Michigan. 1995.
- [Leake et al, 96] Leake, D. B., Kinley, A., Wilson, D., "Learning to Improve Case Adaptation by Introspective Reasoning and CBR". Case-Based Reasoning. Experiences, Lessons and Future Directions. 185-197. Ed. Leake, D. B., AAAI Press/ MIT Press. 1996.
- [Lopez et Plaza, 93] Lopez, B., Plaza, E., "Case-Based Planning for Medical Diagnosis." Methodologies for Intelligent Systems, 7th International Symposium, ISMIS-93. Lecture Notes in Artificial Intelligence 689. Berlin : Springer-Verlag. 1993.
- [Navinchandra, 88] Navinchandra, D., "Case-Based Reasoning in CYCLOPS, a Design Problem Solver", *Proceedings of the DARPA Case-Based Reasoning Workshop*, 286-301. Edit. By J.L. Kolodner. San Francisco: Morgan Kaufmann Publishers. 1988.
- [Pearce et al, 92] Pearce, M., Ashok, K. G., Kolodner, J. L., Zimring, C., Billigton, R., "Case-Based Support. A Case Study in Architectural Design", 14-20. *IEEE Expert*. October 1992.
- [Perera et Watson, 95] Perera, R. S., Watson, I., "A Case-Based Design Approach for the Integration of Design and Estimating", 3-16. *Progress in Case-Based Reasoning*. Edit. By I. Watson. Lecture Notes in Artificial Intelligence 1020. Berlin: Springer-Verlag. 1995.

- [Porter et Bareiss, 86] Porter, B. W., Bareiss, E. R., "PROTOS : An Experiment in Knowledge Acquisition for Heuristic Classification Tasks", *Proceeding of the First International Meeting on Advances in Learning (IMAL)*, 159-174. Les Arcs, France. 1986.
- [Rau, 87] Rau, L. F., "Knowledge organization and access in a conceptual information system", *Information processing and management*, 23(4) :269-283. 1987.
- [Simpson, 85] Simpson, R. L., "A Computer Model of Case-Based Reasoning in Problem Solving: An Investigation in the Domain of the Dispute Mediation", *Georgia Institute of Technology. Technical Report No.GIT-ICS-85/18*. 1985.
- [Schank et Abelson, 77] Schank, R., Abelson, R., "Scripts, Plans, Goals and Understanding", Hillsdale, NJ: Lawrence Erlbaum Associates. 1977.
- [Schank, 82] Schank, R., "Dynamic Memory: a theory of reminding and learning in computers and people", Cambridge University Press. 1982.
- [Simoudis, 92] Simoudis, E., "Using Case-Based Retrieval for Customer Technical Support", *IEEE Expert*, 7(5),7-13. 1992.
- [Stanfill et Waltz, 86] Stanfill, C., Waltz, D., "Toward Memory-Based Reasoning", *Communications of the ACM* 29(12), 1213-1228. 1986.
- [Sycara, 87] Sycara, K., "Resolving Adversarial Conflictss : An approach integrating case-based and analytic methods", Ph.D. thesis. Georgia Institute of technology. 1987.

- [Sycara, 92] Sycara, K., "CADET: A case-based Synthesis tool for engineering design", *International Journal for Expert Systems*. 4(2),157-188. 1992.
- [Watson, 97] Watson, I., "Applying Case-Based Reasoning: Techniques for Enterprise Systems", Morgan Kaufmann Publishers, San Francisco, California. University of Salford. United kingdom. 1997.
- [Watson et Abdullah, 94] Watson, I., Abdullah, S., "Developing Case-Based Reasoning Systems : A Case Study in Diagnosing Building Defects", *Proceedings of the IEEE Colloquium on Case-Based Reasoning : Prospects for Applications*, Digest No:1994/057, 1/1-1/3. 1994.
- [Wess et Globig, 94] Wess, S., Globig, C., "Case-Based and Symbolic Classification Algorithms", University of Kaiserslautern, Germany.1994.
- [Wess et Ritcher, 93] Wess, S., Ritcher, M., "Similarity, Uncertainty and Case-Based Reasoning in PADTEX". University of Kaiserslautern. 1993.
- [Wilke et Bergmann, 96] Wilke, W., Bergmann, R., "Considering Decision Cost During Learning of Feature Weights" *Third European Workshop on CBR*. Lausanne. 1996.
- [Yang et Robertson, 94] Yang, S., Robertson, D., "A Case-Based Reasoning System for Regulatory Information." *Proceedings of the IEEE Colloquium on Case-Based Reasoning: Prospects for Applications*. Digest No: 1994/057, 3/1-3/3. 1994.

Références Complémentaires

- [Althoff et al, 94] Althoff, K. D., Manago, M., Bergmann, R., Maurer, F., Wess, S., Auriol, E., Conruyt, N., Traphoner, R., Brauer, M., Dittrich, S., "Induction and Case-Based Reasoning for Classification Tasks", University of Kaiserslautern, 1994.
- [Bichindaritz, 94] Bichindaritz, I., "Apprentissage de Concepts dans une Memoire Dynamique: Raisonnement à partir de Cas Adaptable à la Tache Cognitive". Universite René Descartes Paris V. 1994.
- [Bhanu, 95] Bhanu, P., "Adaptation of Knowledge for Reuse", *1995 AAAI Fall Symposium*. P.V.S.R. 1995.
- [Berger et Hammond, 91] Berger, J., Hammond, K. J., "Roentgen: A Memory-Based Approach to Radiation Therapy Treatment" *Proceeding of the DARPA Case-Based Reasoning Workshop ed. R. Bareiss, 203-214*. San Francisco, Calif. : Morgan Kaufmann. 1991.
- [Carbonell et al, 83] Carbonell, J. , Michalski, R., Mitchell, T., "Learning by Analogy Formulating a Generalizing Plans from Past Experience." *Machine Learning: An Artificial Intelligence Approach, 137-162*. San Franscisco, Calif. 1983.

- [Chang et Harrison, 95] Chang, L., Harrison, P., "A Testbed for Experiments in Adaptive Memory Retrieval and Indexing", technical report FS-95-04. Ed. Aha D. W. et Ram A. AAAI.1995.
- [Hammond, 89] Hammond, K. J. , "Case-Based Planning: Viewing Planning as a Memory Task" San-Diego, Calif. 1989.
- [Hanney et al, 95] Hanney, K. , Keane, M. T. , Cunningham, P. , Smyth, B. "What Kind of Adaptation do CBR Systems need ? A Review of Current Practice" 1995.
- [Kass et Leake, 88] Kass, A. M. , Leake, D. B. , "Case-Based Reasoning Applied to Construction Explanation", *Proceedings: Workshop on Case-Based Reasoning (DARPA)*, ed. J. Kolodner, 190-208. San Francisco, Calif. : Morgan Kaufmann. 1988.
- [Kolodner et Wills, 96] Kolodner, J. L., Wills, L. M., "Towards More Creative Case-Based Design Systems".Case-Based Reasoning, Experiences, Lessons and Future Directions. 81-91. Leake, D. B., MIT Press 1996.
- [Leake, 96] Leake, D. B. ,"Case-Based Reasoning: Experiences, Lessons & Future Directions", *AAAI Press. MIT Press. Cambridge, Massachusetts*. 1996.
- [Mark et al, 96] Mark, W. , Simoudis, E., Hinkle, D., "Case-Based Reasoning: Expectations and Results". Case-Based Reasoning, Experiences, Lessons and Future Directions. 269-294.Leake, D.B., *MIT Press*. 1996.
- [Ritcher, 93] Ritcher, M. M., "Classification and Learning of Similarity Measures".University of Kaiserslautern. 1993.
- [Schank, 88] Schank, R., "Reminding and Memory, from Dynamic Memory", *Proceeding of the DARPA Case-Based Reasoning Workshop*. J.L.

Kolodner. San Fransisco: Morgan Kaufmann Publishers. 1988.

- [Schank, 96] Schank, R., "Goal-Based Scenarios: Case-Based Reasoning Meets Learning by Doing", *Case Based-Reasoning : Experiences, Lessons and Future Directions*, 295-347, D.B. Leake. Cambridge, MA: AAAI Press/MIT Press, 1996.
- [Segre, 87] Segre, A., "On the Operationality-Generality Tradeoff in Explanation-Based Learning", *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*. 242-247. Menlo Park, Calif. 1987.
- [Smaïl, 94] Smaïl, M., "Raisonnement à base de cas pour une recherche évolutive d'information; Prototype Cabri-n. Vers la définition d'un cadre d'acquisition de connaissances". Thèse de doctorat. Université Henri Poincaré. Centre de Recherche en Informatique de Nancy. 1994.
- [Smyth et Keane, 96] Smyth, B., Keane, M. T., "Design à la Déjà Vu. Reducing the Adaptation Overhead". *Case-Based Reasoning, Experiences, Lessons and Future Directions*. 151-166. Leake, D. B., *MIT Press*. 1996.
- [Talbi, 96] Talbi, H., "Interface Iconique pour un Système d'aide au diagnostic médical à base de cas". Mémoire de maîtrise. Université de Montréal. 1996.
- [Thoré, 95] Thoré, I., "Raisonnement par cas et notion d'inconduite en droit de l'assurance-chômage" Mémoire de maîtrise. Université de Montréal. 1995.

- [Wess et al, 94] Wess, S., Althoff, K. D., Derwand, G., "Using k-d Trees to Improve the Retrieval Step in Case-Based Reasoning", University of Kaiserslautern. 1994.