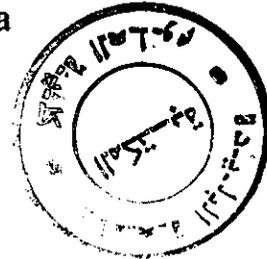


République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab, Blida
USDB.

Faculté des sciences.
Département informatique.



**Mémoire pour l'obtention
du diplôme d'ingénieur d'état en informatique.**
Option : Système d'information

Sujet :

**Introduction de la logique floue dans
le raisonnement à base de cas
Application dans le domaine médical**

**Présenté par : BENDAOUH Houria
REKIK Hassiba**

Promoteur : M^{me} ABED

Organisme d'accueil : Université de Blida.

Soutenue le: 02/07/2007, devant le jury composé de :

M^r HADJ SADOUK

Président

M^{me} BOUMAHDJ

Examineur

M^{me} BENSTITI

Examineur

RÉSUMÉ

L'aide au diagnostic médical par logiciels informatiques connaît un développement important, la médecine est une science fondée sur l'observation de l'organisme humain. Les maladies sont des processus dynamiques dont les syndromes cliniques ne sont que des clichés correspondant à des stades d'évolution. Certaines maladies progressent par étape d'une manière linéaire.

Le système d'aide à la décision intègre différents modèles décisionnels et les utilise pour traiter les étapes nécessaires du processus clinique : diagnostic, pronostic et thérapie. Le processus de décision comporte les données relatives au patient, les connaissances des experts, les données épidémiologiques et statistiques ainsi que les expériences précédentes.

Le domaine médical est caractérisé par un vocabulaire extrêmement riche mais également difficile à manipuler. Les termes utilisés sont souvent flous et imprécis. Ils font rarement l'objet de définitions rigoureuses acceptables par l'ensemble de la communauté médicale. Par raisonnement, nous entendons le processus général qui consiste à utiliser des connaissances sur un système, pour en bâtir d'autres relatives à ce même système. Reasonner en présence de connaissances imprécises a été l'une des préoccupations premières énoncées par L.A.Zadeh, [4a] [4b] qui souhaitait approcher le raisonnement humain. S'il est vrai que les être humains sont capables de formuler des conclusions, précises ou imprécises selon le cas, à partir de connaissances exprimées de façon approximative ou linguistiquement. Automatiser ce processus n'a été possible que par l'intermédiaire de la théorie des ensembles flous, encore a-t-il fallu l'introduction de la théorie des possibilités pour que des incertitudes puissent être manipulées en même temps que des imprécisions, condition sans laquelle il n'est pas possible de raisonner.

Il apparaît que la technique de Raisonnement à Base de Cas flou est la plus séduisante car elle propose un recours à l'expérience passée pour résoudre de nouveaux problèmes. De plus, cette approche est indiquée dans les domaines où l'activité repose de façon importante sur les connaissances comme le domaine médical. Dans notre système, nous proposons donc d'utiliser la technique de Raisonnement à Base de Cas flou.

Mots clés : gestion des connaissances, logique floue, Raisonnement à base de cas, Diagnostic médical, Intelligence artificielle.

ABSTRACT

Helping the medical diagnostic with software of a computer knows an important development; medicine is a science which is based on the observation of the human being organism. The diseases are a dynamic process where the clinical syndromes are not typographies. Which correspond to the development stages. Some diseases progress through a stage of a linear manner.

The helping system of decision integrates models of decision and uses them to treat the necessary steps of the clinical process: diagnostique, prognostic and therapy. The process of decision includes the relative data of a patient, knowledge's of experts, epidemic and statistic data and the previous experiences.

The medical domain is characterized by a very rich vocabulary but also difficult to manipulate. The using terms are often blurred and imprecised. They are rarely the object of acceptable rigorous definitions from the whole medical community. Reasoning, we hear the general process which consists of using the knowledge's in a system, to build other relatives to the system. Arguing with the presence of imprecised knowledge was one of the first preoccupations pronounced by L.A.Zadeh,[4a] [4b] that hoped to approach human being reasoning. If it is true that human beings are able to formulate conclusions, whether precised or imprecised depending on the case, from the knowledge's that are expressed in an approximate or linguistic way, to automate this process was not possible only through the theory of the whole blurred unities. And was the introduction of the theory of the necessary possibilities for the uncertainty can be manipulated at the same time as the imprecision's, that is the condition of reasoning.

The technique of reasoning at base of blurred case seems to be the most attractive because it suggests having recourse to the previous experience to solve the new problems.

In addition, this approach is specified in the domains where the activities rested in an important way on the knowledge's as the medical domain.

In our system we suggest to use the technic of reasoning at base of blurred case.

Keywords: Knowledge Management, Fuzzy Logic, Case-based Reasoning (CBR), medical Diagnostique, Intelligence artificial.

REMERCIEMENTS

Nous remercions avant tous Dieu tous puissant qui nous a donné la force et le courage pour réalisé ce modeste travail.

Nous tenons a remercier vivement notre promotrice Mme Abed d'avoir accepté de nous encadré et proposer ce sujet, nous exprimant toutes notre gratitude et nos s'incère remerciement pour tout leur effort, coopération, aide, patience, encouragement et vaillance pour la réalisation ce travail, prient dieu de leur donné toute la force pour continué leur noble mission.

Nous remercions vivement les membres de jury de nous avoir fait l'honneur d'être rapporteurs et examinateurs tous en apportant leurs remarques et leurs contributions à l'enrichissement de ce mémoire.

Aussi, nous remercions tous les enseignants du département d'informatique, pour nous avoir formé tout au long de notre cursus, et nous espérons dans le futur proche.

Merci aussi à Abdelhadi, qui m'a inspiré et m'a donné la volonté d'aller jusqu'au bout.

Nous remercions a tous ceux que d'une manière ou autre, nous ont apporté de l'aide durant tous notre cursus.

DÉDICACE

Je dédie ce modeste travail à ma très chère mère à mon père, à mes sœurs et mes frères et je les remercie pour leur soutien tout au long de mes études

A mes neveux et nièce

A mon binôme et amie Hassiba et toute sa famille

A mes chers amis de promotion surtout Rima, Samia, Safia, Fouzia, Samir, Mehdi, Allal, Bilal, Assia...

A mes proches : Bendaoud Abdelhadi, Meriem, Siham, Rabia, Fatiha, Nahla, Basma...

A tous ceux que je n'est pas cité par oublie, qu'il me pardonne.

Bendaoud Houria

DÉDICACE

Je dédie ce modeste travail à :

Ma très chère mère et mon père

Mes chères sœurs : Baya, Karima, Hadjera, Yamina

Mes frères

A mes neveux et nièce

Mon binôme Houria et toutes sa famille

Mes plus chères amies Akouche safia et Amina

Tous mes amis de promotion 2007 Rima, Samia, Safia, Fouzia, Samir,
Mehdi, Allal, Bilal, Assia...

A tous ceux que je n'est pas cité par oublie, qu'il me pardonne.

Rekik Hassiba

TABLES DES MATIERES

RÉSUMÉ.....	i
ABSTRACT.....	ii
REMERCIEMENTS.....	iii
DEDICACES.....	iv
TABLES DES MATIERES	vi
INTRODUCTION GÉNÉRALE	xiii
LA PROBLÉMATIQUE	xvi

Chapitre 1 : La gestion des connaissances

Introduction	1
I. Concepts de base de la Connaissance	2
II. Les Types de Connaissances	4
III. Les modes de conversion de la connaissance	6
IV. Gestion des connaissances	7
IV.1. Définitions	7
IV.2. Cycle de vie de la connaissance	8
V. L'imperfection des connaissances	8
Conclusion	11

Chapitre 2 : la logique floue

Introduction	12
I. Principe de la logique floue	13
I.1. Conditions d'utilisation	14
I.2. Domaines d'application	14
II. Théorie des ensembles flous	15
II.1. Définition d'un sous-ensemble flou	15
II.2. Caractéristiques d'un sous-ensemble flou	17
II.3. Opérations sur les sous-ensembles flous	19
II.4. Relations et Quantités floues.....	22

Tables des matières

III. Eléments de base de la logique floue	23
III.1. Variables linguistiques	23
III.2. Opérateurs linguistiques	24
III.3. Propositions floues	24
III.4. Implications floues	24
III.5. Règles floues	25
IV. Théorie des possibilités	26
IV.1. Mesure de possibilité.....	26
IV.2. Distribution de possibilité	27
IV.3. Mesure de nécessité	27
IV.4. Relations entre mesure de possibilité et de nécessité	28
IV.5 Mesures floues	29
Conclusion	30

Chapitre 3 : Le raisonnement à base de cas RBC

Introduction	31
I. Le principe du Raisonnement à Base de Cas (RBC)	32
II. Définitions	32
III. Les composants d'un système RBC	33
III.1. Processus	34
III. 2. Connaissances	34
IV. Structure d'un cas	35
V. Les tâches et les domaines d'application	37
V.1. La tâche de résolution de problèmes	38
V.2. La tâche d'interprétation	38
VI. Modèles RBC	39
VI.1. Modèle structurel	39
VI.2. Modèle conversationnel	39
VI. 3. Modèle textuel	40
VII. Le cycle du RBC	40

Tables des matières

VII.1. Remémoration.....	41
VII.2. Réutilisation	43
VII.3. Révision	44
VII. 4. Mémorisation	46
VIII. Notion de similarité	47
Conclusion	50

Chapitre4 : La logique floue et le raisonnement à base de cas pour le diagnostic médical

Introduction	51
I. Traitement de données imprécises et incertaines.....	52
II. La logique floue et le raisonnement à base de cas dans le cadre du diagnostic médical	53
II.1. Représentation du Cas	53
II.1.1. Description de problème	53
II.1.2. Description de la solution	54
II.2. Le cycle du RBCF dans le cadre du diagnostic médical	56
1. La phase de remémoration	57
2. La phase de choix de la solution	60
3. La phase de révision	61
4. La phase de mémoration	61
Conclusion	62

Chapitre5 : Démarche de développement

Introduction	63
I. La démarche de développement	64
II. L'analyse	65
II.1. Les cas d'utilisation	65
II.2. Diagrammes de séquences	70
II.3. Diagramme de collaboration	78
II.4. Diagramme de classes	79

Tables des matières

III. Conception	80
A. La conception du système	80
B. La conception des objets	83
IV. Implémentation	85
IV.1. Choix du langage de programmation.....	85
IV.2. Mise en œuvre de l'interface utilisateur.....	85
V. Test du logiciel.....	97
Conclusion	99

Conclusions et perspectives	100
-----------------------------------	-----

Annexes

Annexes A.....	102
Annexes B.....	109
Références bibliographiques.....	111

Listes des figures

Figure 1.1 : La connaissance vue par Ermine.....	3
Figure 1.2 : Les quatre modes de conversion de la connaissance (D'après Ikujiro Nonaka et Hiroataka Takeuchi).....	6
Figure 2.1 : Niveaux d'appartenance	13
Figure 2.2 : Représentation des ensembles flous	16
Figure 2.3 : Exemples de fonctions d'appartenance monotones décroissantes.....	17
Figure 2.4 : Exemples de fonctions d'appartenance monotones croissantes.....	17
Figure 2.5 : Exemples de fonctions d'appartenance.....	17
Figure 2.6 : Caractéristiques d'un ensemble flou.....	18
Figure 2.7 : Fonctions d'appartenance avec trois ensembles pour la variable «taille»...	20
Figure 2.8 : Opérateur OU	20
Figure 2.9 : Opérateur ET.....	21
Figure 2.10 : Opérateur NON.....	21
Figure 2.11 : Exemple d'intervalle flou et de nombre flou	23
Figure 3.1 : Modèle générique d'un système RBC.....	33
Figure 3.2 : Cycle d'un RBC	41
Figure 3.3 : La décomposition de la tâche « Remémorer »	42
Figure 3.4 : La décomposition de la tâche « Réutiliser »	44
Figure 3.5 : La décomposition de la tâche « Révision »	45
Figure 3.6 : La décomposition de la tâche « Mémoriser »	46
Figure 3.7 : Méthode de calcul du plus proche voisin.....	48
Figure 4.1 : Ensemble flou représentant la valeur «forte» pour l'attribut «fièvre»	52
Figure 4.2 : Le cycle de raisonnement à base de cas flou dans le cadre de diagnostic médical.....	57
Figure 4.3 : Degré de similarité pour des attributs nominaux.....	60
Figure 5.1 : Le cycle de vie en cascade.....	64
Figure 5.2 : Représentation des catégories d'utilisateurs.....	65

Tables des matières

Figure 5.3 : Diagramme générale de cas d'utilisation de notre système.....	66
Figure 5.4 : Diagramme de cas d'utilisation «consultation de la base de cas ».....	67
Figure 5.5 : Diagramme de cas d'utilisation « mise à jour de la base de cas ».....	67
Figure 5.6 : Diagramme de cas d'utilisation « Effectuer un diagnostic médical ».....	68
Figure 5.7 : Diagramme générale de cas d'utilisation de notre système.....	69
Figure 5.8 : Diagramme de séquence pour effectuer un diagnostic par un utilisateur externe.....	71
Figure 5.9 : Diagramme de séquence pour effectuer un diagnostic par un médecin.....	72
Figure 5.10 : Diagramme de séquence pour Consultation de la base de cas.....	73
Figure 5.11 : Diagramme de séquence pour l'ajout d'un cas.....	74
Figure 5.12 : Diagramme de séquence pour la suppression d'un cas	76
Figure 5.13 : Diagramme de séquence pour la modification d'un cas	77
Figure 5.14 : Diagramme de collaboration pour effectu un diagnostic par le médecin....	78
Figure 5.15 : Diagramme de classes de la phase d'analyse.....	79
Figure 5.16 : Architecture générale du système.....	80
Figure 5.17 : Diagramme d'activités pour le cas d'utilisation « Effectuer un diagnostic médical »	82
Figure 5.18 : Diagramme de classes de la phase de conception	84
Figure 5.19 : Représentation de l'interface principal.....	86
Figure 5.20 : L'interface de la session médecin.....	87
Figure 5.21 : L'interface de la session utilisateur externe.....	88
Figure 5.22 : Consultation de la base de cas.....	89
Figure 5.23 : Saisie de la description d'un cas.....	90
Figure 5.24 : Recherche du cas similaire.....	91
Figure 5.25 : Saisie de la description d'un cas.....	92
Figure 5.26 : Recherche du cas similaire.....	93
Figure 5.27 : Saisie du diagnostic.....	94
Figure 5.28 : Demande de la suppression d'un cas.....	95
Figure 5.29 : Demande de la modification d'un cas.....	96
Figure 5.30 : Mise à jour des poids des attributs.....	97
Figure 5.31 : Test du logiciel.....	98

Tables des matières

Listes des tableaux

Tableau 3.1 : Tableau de notations des graphes	42
Tableau 4.1 : Présentation de la table des poids.....	59
Tableau 5.1 : Description des attributs.....	83

INTRODUCTION GÉNÉRALE

Les bases de données relationnelles floues ont fait l'objet de nombreuses recherches afin de construire des systèmes d'information permettant la représentation et l'interrogation de données floues et imprécises. Dans la plupart des cas, la théorie des sous-ensembles flous et la théorie des possibilités ont été largement utilisées pour la modélisation de valeurs d'attributs connues, mal connues et nulles. En général dans ces modèles, les informations sur les données peuvent être complètes, incomplètes ou nulles.

Les situations où intervient un facteur humain sont privilégiées dans l'utilisation de la logique floue. C'est le cas du domaine médical. La représentation de l'imprécision et de l'incertitude est au centre de très nombreux problèmes de décision médicale. Les aspects les plus courants concernent la fréquence des maladies, des associations entre présence d'un signe et présence d'une maladie, l'indisponibilité d'une information utile au moment de la décision. La logique floue est le formalisme le plus souvent employé pour représenter ces connaissances, les combiner et les propager dans des raisonnements où les règles de la logique classique ne sont plus applicables.

Dans le domaine médical, l'utilisation de l'approche du Raisonnement à Base de Cas Flou (RBCF) semble particulièrement intéressante dans le sens où le type de raisonnement utilisé dans la démarche diagnostique repose essentiellement sur l'expérience liée à la résolution de cas rencontrés dans la pratique. La présentation des cas du passé à l'utilisateur permet en effet à celui-ci de s'orienter rapidement vers une solution. Le RBCF est ainsi une alternative possible à la problématique de la gestion des connaissances dans le domaine médical.

Nous avons choisi le raisonnement à base de cas. Le RBC est un l'un des paradigmes proposés par l'IA pour la résolution de problème, il est considéré comme une technique intéressante pour la gestion des connaissances, le RBC est en effet adopté pour résoudre des problèmes difficiles à modéliser et même à l'expertise est difficile à rendre explicite. Il a pour particularité de faire intervenir des cas déjà résolus pour résoudre un nouveau problème. Il intègre donc en lui-même un processus d'apprentissage. Il utilise comme base de connaissance une base d'expériences passées ou base de cas.

Introduction générale

Son principe est de chercher dans la base de cas le cas qui soit le plus proche possible du problème courant, d'adapter la solution du cas trouvé pour résoudre le problème cible, et mémoriser ce nouveau problème avec la solution produite dans la base de cas comme un nouveau cas.

Le RBC présente des avantages parmi lesquels: elle permet d'éviter les problèmes d'acquisition de connaissance qui rendent difficile la conception de bases de connaissances de taille importante, la facilité de maintenance, l'amélioration de la performance par association des solutions existantes, l'utilisation des solutions existantes, la pro-activité (amélioration avec le temps), l'élimination des connaissances superflues, la structuration de la connaissance(cas), la flexibilité et la souplesse. Le RBC est particulièrement bien adapté aux applications dont la tâche est accomplie par des humains expérimentés.

Organisation du mémoire :

Le mémoire est subdivisé cinq chapitres.

Dans le **premier chapitre**, nous présentons les différentes définitions de la connaissance explorons les notions de gestion de connaissances

Le **deuxième chapitre** sera consacré à la présentation des concepts de base de la théorie des sous-ensembles flous et la théorie des possibilités, ...etc.

Dans le **troisième chapitre**, nous présentons le raisonnement à base de cas, nous mettons en évidence ses étapes, ses composants et son mécanisme de fonctionnement.

Dans le **quatrième chapitre**, nous présenterons d'abord le traitement de données imprécises et incertaines. Ensuite, nous présentons les différentes étapes de raisonnement à base de cas flou dans le diagnostic médical.

Dans le **cinquième chapitre**, il est réservé à la présentation de la démarche de développement, en commençant par la partie analyse et spécification des besoins, conception, implémentation et enfin la partie test du logiciel. Dans ce chapitre l'accent, est mis sur le langage de modélisation UML que nous avons choisi pour modéliser notre système. En dernier, on finalise notre travail en offrant des conclusions et des recommandations futures qui seraient intéressantes d'apporter au logiciel développé.

Introduction générale

Pour conclure, nous présentons la bibliographie et les annexes.

L'Annexe A donne une brève description du langage de modélisation UML.

L'Annexe B apporte un aperçu sur le cycle de vie en cascade.

LA PROBLÉMATIQUE

En médecine, les observations sur lesquelles se basent les décisions médicales sont très imparfaites. Elles peuvent être floues, ambiguës, incertaines. Le processus décisionnel, qu'il soit diagnostique ou thérapeutique est donc un processus sous incertitude.

Plusieurs éléments participent à cette situation.

Par exemple l'analyse d'une maladie, donne le type de réponses suivantes :

- _ Etes vous malade ?
- _ Oui (quelle question !)
- _ Toussez vous ?
- _ Oui beaucoup.
- _ Toussez vous OUI ou NON ?
- _ Oui (pff).
- _ Avez vous une toux sèche ou grasse ?
- _ Un peu sèche.
- _ Avez vous une toux SECHE ou GRASSE ?
- _ un peu sèche !

On comprendra que des réponses logiques, limitées à un OUI ou un NON ne correspondent pas toujours aux réponses données dans la réalité. En effet, pour rendre fine et précise la compréhension de ce que nous exprimons, nous ajoutons des adjectifs qualificatifs (grand, modérée..) ou des adverbes (beaucoup, un peu) qui permettent d'apporter des nuances beaucoup plus subtiles et plus concrètes qu'une simple affirmation ou négation : C'est toute l'échelle des gris qui s'oppose au simple choix du blanc ou du noir.

Effectuer un diagnostic médical nécessite une véritable expertise humaine. Pour cela, nous avons pensé à la constitution d'une base de connaissance qui permettra de capitaliser le savoir faire des médecins et sera utilisée pour effectuer un diagnostic médical.

Cette base sera représentée en utilisant le raisonnement à base de cas, et la logique floue a été proposée pour représenter les concepts médicaux.

Notre travail se concentre donc sur l'introduction de la logique floue dans le raisonnement à base de cas, ce travail décrit la démarche que nous proposons sur aborder le problème de la gestion des connaissances imparfaites à l'aide de la technique de Raisonnement à Base de Cas Flou.

OBJECTIFS :

Les données imprécises existent dans tous les domaines de la vie. Leur représentation et leur structuration dans une base de données ne sont pas encore bien maîtrisées.

Le but de ce travail est de proposer une représentation de ces données dites imprécises ou floues.

Pour éclaircies cet objectif davantage on a pris comme exemple d'étude un cas appartenant au domaine médical.

Chapitre 1 : La gestion des connaissances

Introduction :

Diverses disciplines sont concernées par le concept de *connaissance*, en tant qu'objet à construire, à exprimer, à transmettre, à acquérir ou à exploiter.

Les connaissances constituent une donnée majeure de l'Intelligence Artificielle qui se préoccupe de leur acquisition, leur structuration, de leur représentation pour construire des bases de connaissances en vue de leur utilisation.

Gérer les connaissances consiste donc à gérer les informations en prenant en compte leur contexte et leur sémantique. Plusieurs voies sont possibles pour une telle gestion, elles dépendent essentiellement du type de connaissances que l'on souhaite traiter.

La représentation des connaissances désigne un ensemble d'outils et de technologies destinés d'une part à représenter et d'autre part à organiser le savoir humain pour l'utiliser et le partager.

L'objectif de ce chapitre est de donner un aperçu de ce qu'est la connaissance et la gestion des connaissances.



Chapitre 1 : La gestion des connaissances

I. Concepts de base de la Connaissance :

Définitions :

Qu'est-ce que la connaissance ?

Selon Ermine [14] « La connaissance est perçue comme de l'information qui a une certaine signification par rapport à un contexte donnée ».

Selon Franck [26] « La connaissance est le résultat d'une expérience et d'une réflexion basée sur un ensemble de croyances et résidant dans des objets fictifs dans le cerveau des personnes ».

D'après ces définitions nous pouvons distinguer plusieurs notions, celle de donnée, d'information et de connaissance.

Donnée

Une donnée est un élément brut qui n'a pas été traité ou qui n'a pas été mis en contexte. Elle a peu de valeur en elle même, mais se prête facilement au traitement et au stockage sur ordinateur.

Exemple : Soit la donnée « 39°C ». Il s'agit bien d'une mesure de température, mais n'a aucune valeur en elle même. S'agit-il de la température de l'air ambiant ? de l'eau d'une casserole ? Ou celle du corps humain ? Ou d'autre objet ?

Information :

Une information est, par définition, une donnée interprétée ou mise en contexte. C'est cette mise en contexte qui donne de la valeur ajoutée à la donnée pour constituer l'information. Cette information a, aussi, peu de valeur tant qu'elle ne peut servir à quelque chose ; agir ou prendre une décision.

Exemple : La mise en contexte de la donnée précédente peut être « la température du patient X est de 39° C ». C'est un fait, une information.

Connaissance :

La connaissance est une information comprise, c'est à dire assimilée et utilisée, qui permet d'aboutir à une action ou de prendre une décision. Ainsi, la compréhension et la faculté d'apprentissage et d'action constituent des conditions du passage de l'information à la connaissance.

Chapitre 1 : La gestion des connaissances

Exemple : L'information précédente « la température du patient X est de 39° C » peut être considérée comme une connaissance si elle permet de prendre une décision telle que ; prescrire un traitement médical.

On peut schématiser ces définitions comme suit :

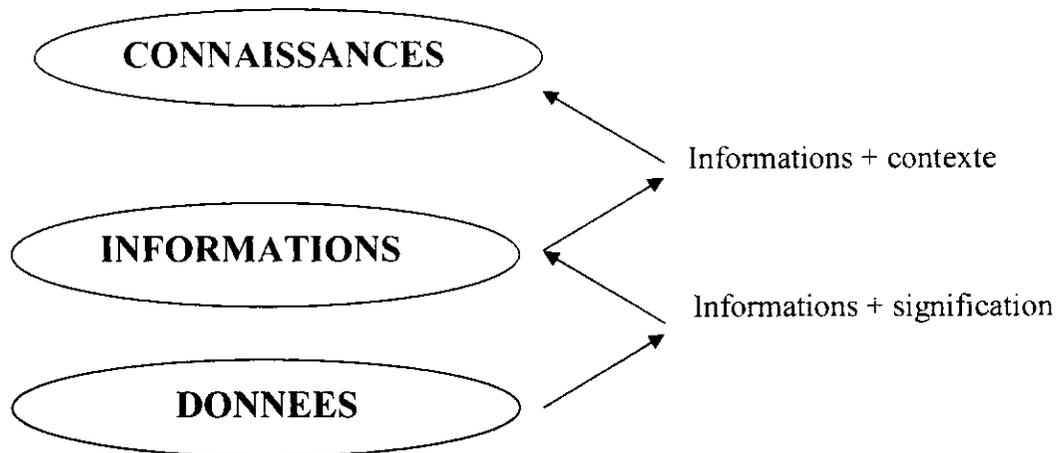


Figure 1.1 : La connaissance vue par Ermine.

En résumé, la connaissance est donc un dérivé d'informations qui est elle même un dérivé de données. Alors que l'on trouve les données dans des transactions ou dans des enregistrements, l'information se trouve dans des messages, la connaissance est obtenue à travers des individus ou des groupes de personnes possédant cette dernière, ou parfois encore dans des routines organisationnelles. Elle est livrée au moyen de supports structurés comme des livres ou des documents divers.

D'autres concepts peuvent être liés au concept « connaissance » :

Savoir :

C'est l'ensemble des connaissances acquises par un individu par l'apprentissage ou l'expérience.

Savoir-faire :

Le savoir faire est un ensemble des connaissances nécessaire pour réaliser une tâche précise.

Chapitre 1 : La gestion des connaissances

Compétence :

La compétence est la capacité d'un individu à agir en apportant ou créant de la valeur ajoutée. Elle a trait à l'utilisation du savoir, et donc des connaissances, d'un individu.

II. Les Types de Connaissances :

Plusieurs typologies des connaissances peuvent être citées, selon le contexte [21] [17] :

Les connaissances explicites :

Ces connaissances concernent un savoir facilement codifié, pour lesquelles il est donc relativement aisé de rédiger des manuels et procédures.

Ce groupe comprend également les informations pour lesquelles une distinction supplémentaire peut être faite entre :

- Les informations brutes primaires : les documents originaux (par document il faut comprendre aussi bien les textes internes qu'externes, les annonces de la concurrence, les brevets, ... sous forme de textes, images, sons, ...).
- Les informations brutes secondaires : les résumés, les noms d'auteurs, les index.
- Les informations élaborées : les notes de synthèse, études de marché, prévisions sur la concurrence.

Toutes ces connaissances peuvent être transmises à l'aide d'un service informatique.

Les connaissances tacites :

Ces dernières connaissances, ou savoir-faire, sont les plus difficilement transférables car elles ne peuvent qu'être difficilement imitées. Elles acquièrent un caractère stratégique : une entreprise qui sait fabriquer un produit de meilleure qualité peut se permettre de le vendre en dégageant des marges plus importantes tout en augmentant ses parts de marché. L'entreprise doit se préoccuper de la gestion de ce savoir qui contribue fortement à son avance technologique sur la concurrence.

Parmi ces connaissances une distinction peut également être faite entre :

- Les connaissances qui ne peuvent être transmises que par apprentissage dans l'action : "c'est en forgeant que l'on devient forgeron". Des expressions de ce type montrent que l'informatique, ni aucun autre support, n'est suffisant pour l'acquisition de ce type de connaissances : il est illusoire d'apprendre à tenir sur une bicyclette, conduire une automobile, ... sans apprentissage dans l'action ; l'entreprise n'a d'autres possibilités que de proposer des monitorats, mobilités, rotations des salariés pour favoriser ces transferts et acquisitions.

Chapitre 1 : La gestion des connaissances

- Les connaissances (difficilement formalisables) issues d'expériences. Ces connaissances (ou raisonnements d'experts) peuvent être élucidées grâce aux techniques d'interview : l'interview centrée, l'interview structurée, l'énonciation, l'introspection ; elles font l'objet du paragraphe se rapportant au raisonnement à base de cas.

Les connaissances individuelles :

Chaque individu possède un ensemble de connaissances qu'il gère de manière intuitive et subjective.

La connaissance individuelle permet à l'individu d'extraire de l'information pour lui donner du sens. En retour, ce processus met à jour la connaissance de l'individu.

Les connaissances collectives :

Les connaissances sont distribuées ou partagées entre différents individus. Au niveau de la connaissance collective, l'information rapportée par l'individu est collectivisée (avec la formalisation par exemple) pour produire un sens collectif désigné par l'expression « connaissances des entreprises ».

Les connaissances déclaratives :

Les connaissances déclaratives appelées aussi, connaissances statiques, donnent des informations sur les objets (réels ou hypothétiques) du monde.

Exemple : « le diabète est une maladie chronique ».

Les connaissances procédurales :

Les connaissances procédurales appelées aussi, connaissances dynamiques, donnent des indications sur les procédures et les conditions d'utilisation de ces procédures. Exemple : « si hypoglycémie, prendre un morceau de sucre ».

III. Les modes de conversion de la connaissance :

Nonaka et Takeuchi se sont intéressé à la création et à la transformation des connaissances, et ont décrit quatre modes de conversion des connaissances au sein d'une entreprise [21] :

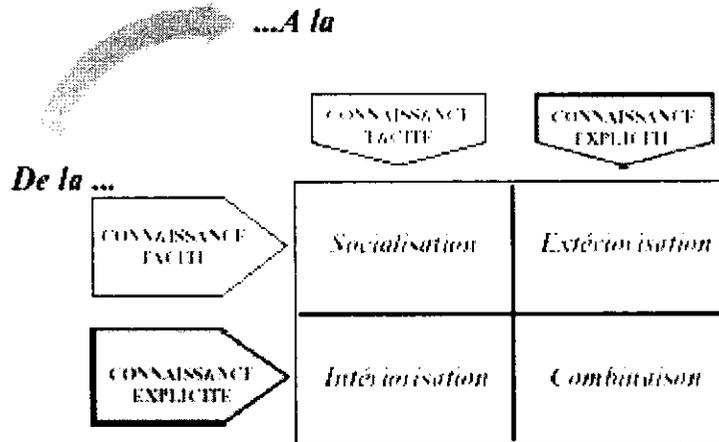


Figure 1.2 : Les quatre modes de conversion de la connaissance
(D'après Ikujiro Nonaka et Hirotaka Takeuchi)

Du tacite au tacite, c'est la *socialisation* où les connaissances tacites des uns (notamment celui du maître) sont transmises directement aux autres (notamment à l'apprenti) sous forme de connaissances tacites, par l'observation, l'imitation et la pratique. Au cours de ce processus aucun des protagonistes n'explique son art pour le rendre directement accessible à tous. Ces connaissances ne pourront donc pas être exploitées au niveau collectif de l'entreprise.

Du tacite à l'explicite, c'est l'*extériorisation* où l'individu essaie d'expliquer son art et de convertir son expérience en connaissances explicites.

De l'explicite à l'explicite, c'est la *combinaison* où l'individu combine divers éléments de connaissances explicites pour constituer de nouvelles connaissances, explicites elles aussi.

De l'explicite au tacite, c'est l'*intériorisation* où, peu à peu, les connaissances explicites diffusées dans l'organisation sont assimilées par le personnel. Ces nouvelles connaissances viennent compléter la somme des connaissances dont dispose l'individu. Elles sont intériorisées et deviennent partie intégrante de chacun. Les connaissances explicites deviennent tacites.

Chapitre 1 : La gestion des connaissances

IV. Gestion des connaissances :

IV.1. Définitions :

Il est nécessaire de définir ce qu'est la gestion des connaissances (*Knowledge management* en anglais). Nous allons donc proposer deux définitions, la première de J.-L. Ermine (1996) et la seconde rapportée par M. Stefanelli (2002) [14] :

- La gestion des connaissances (déf. 1) : vise à rassembler le savoir et le savoir-faire sur des supports facilement accessibles, faciliter leur transmission en temps réel à l'intérieur de l'établissement et en différé à nos successeurs et garder la trace de certaines activités ou actions sur lesquelles on peut devoir rendre des comptes à l'avenir. Chaque direction (...) est chargée de définir, dans son domaine de responsabilité, ce qui doit être écrit et conservé et d'organiser cette conservation.
- La gestion des connaissances (déf. 2) : vise à améliorer les performances de l'organisation en permettant aux individus de capturer, partager et appliquer des connaissances collectives pour faire prendre des décisions optimales... en temps réel.

La gestion des connaissances ou ingénierie des connaissances est l'ensemble des méthodes et des techniques permettant de percevoir, d'identifier, d'analyser, d'organiser, de mémoriser, et de partager des connaissances entre les membres des organisations, en particulier les savoirs créés par l'entreprise elle-même (ex : marketing, recherche et développement) ou acquis de l'extérieur (ex : intelligence économique).

Objectifs :

Les objectifs de la gestion des connaissances sont donc de :

- formaliser et échanger des savoirs spécifiques à l'organisation (entreprise, association, administration, ...), savoirs souvent non formalisés tels les savoir-faire et procédures complexes résultant de l'expérience,
- fournir les informations utiles, et seulement elles, au moment opportun sans que l'on ait besoin de lancer des recherches longues et fastidieuses ;
- permettre de capitaliser les informations de manière organisée, afin de les pérenniser.

IV.2. Cycle de vie de la connaissance :

Les cinq principales activités autour des connaissances sont :

1. Identification et sélection des connaissances stratégiques à partir de l'objectif fixé
2. Création de (nouvelles) connaissances aux niveaux individuel et organisationnel
3. Enregistrement des connaissances pour constituer le capital intellectuel
4. Partage des connaissances par création de flux (travail collaboratif)
5. Utilisation des connaissances.

Ordre Général

1-identification 2-création 3-stockage 4-partage 5-utilisation

V. L'imperfection des connaissances :

D'une façon générale, on peut dire que les connaissances sur un système réel sont souvent imparfaites. D'après Bouchon-Meunier [4a], les raisons donnant lieu à ces imperfections sont de deux types :

- Le premier type de raison concerne l'obtention des connaissances à partir du réel s'effectue en deux étapes : l'observation et la représentation. La première se produit à travers des intermédiaires instrumentaux ou humains qui sont généralement soumis à des erreurs, des imprécisions et des incertitudes. La seconde étape, celle de la représentation, prend forme dans le langage naturel, les nombres fixés à une certaine précision ou encore une formulation mathématique. Autant l'observation que la représentation entraînent une perte d'information d'autant plus grande que le système est complexe.

- Les raisons du seconde type sont liées à une absence de rigueur ou une flexibilité inhérente au système lui-même et à son fonctionnement, c'est le cas pour toutes les caractéristiques de phénomènes naturels tels que la durée de maturation d'un fruit, la taille d'un animal adulte, le passage progressif et non strict du jour à la nuit ; c'est aussi le cas de certains systèmes artificiels, tels que la charge maximale d'un ascenseur, indiquée en kilogrammes dans un souci de simplicité mais à laquelle on peut ajouter quelques grammes sans problème majeur ou le nombre maximal de voyageurs que peut contenir un wagon de métro, dépendant du degré de compression accepté par les passagers.

Chapitre 1 : La gestion des connaissances

Les imperfections dans les connaissances sont de natures différentes, principalement au nombre de trois :

- **Les incertitudes** concernant un doute sur la validité d'une connaissance. Celui-ci peut provenir d'une fiabilité relative de l'intermédiaire d'observation, peu sûr de lui ou susceptible de commettre des erreurs (« je crois que la voiture était blanche ») ou de donner intentionnellement des informations erronées, ou encore d'une difficulté dans l'obtention ou la vérification de la connaissance (incertitude sur la situation ennemie dans un cadre militaire par exemple, affirmation d'une douleur forte par un patient). Des incertitudes sont également présentes dans le cas de prévisions (en météorologie par exemple).
- **Les imprécisions** correspondent à une difficulté dans l'énoncé de la connaissance, soit parce que des connaissances numériques sont mal connues, soit parce que des termes du langage naturel sont utilisés pour qualifier une caractéristique du système de façon vague. Le premier cas est la conséquence d'une insuffisance des instruments d'observation (2000 à 3000 manifestants), d'erreurs de mesure (poids à 1% près) ou encore de connaissances flexibles (la taille d'un adulte est environ entre 1.50 et 2 mètres). Le second provient de l'expression spontanée de connaissances (température douce, grand appartement, proche de la plage) ou de l'utilisation de catégories aux limites mal définies (enfant, adulte, vieillard).
- **Les incomplétudes** sont des absences de connaissances ou des connaissances partielles sur certaines caractéristiques du système. Elles peuvent être dues à l'impossibilité d'obtenir certains renseignements (fichiers de malades dans lesquels certaines rubriques ne sont parfois pas remplies) ou à un problème au moment de la captation de la connaissance (image avec une partie cachée). Elles peuvent aussi être associées à l'existence de connaissances générales sur l'état d'un système, habituellement vraies, soumises à des exceptions que l'on ne peut pas énumérer ou prévoir, selon les cas, (« généralement, Mohamed est à son bureau tous les jours », sauf s'il est malade ou si un événement grave survient dans sa famille). Elles sont généralement liées à l'existence de connaissances implicites, par exemple dans une recherche d'information auprès d'experts.

Ces types de connaissances ne sont pas indépendants les uns des autres.

Chapitre 1 : La gestion des connaissances

Traitement des connaissances imparfaites :

Manipulation des imprécisions

Tous les spécialistes s'accordent à dire que le cadre le plus adapté à la représentation et la manipulation de connaissances imprécises est celui des ensembles flous. Les concepts d'ensembles flous et de logique floue ont été introduits par Zadeh [4a]. L'idée de Zadeh était de pouvoir manipuler des informations exprimées en langage naturel. Cet objectif nécessitait d'étendre la théorie des ensembles et la logique propositionnelle classiques.

Manipulation des incertitudes

La théorie des possibilités est l'un des outils mathématiques pour représenter, gérer et évaluer des incertitudes à l'aide des ensembles flous définis sur un domaine ordonné. Elle a été introduite et formalisée aussi par Zadeh au travers de la *mesure de possibilité* et de la *mesure de nécessité*, puis développée par Prade et Dubois [4a] [4b].

Chapitre 1 : La gestion des connaissances

Conclusion

On a présenté au cours de ce chapitre les différentes définitions de la connaissance ainsi que les concepts de base liés à celle-ci.

La gestion des connaissances est une démarche stratégique pluridisciplinaire visant à atteindre l'objectif fixé grâce à une exploitation optimale des savoirs de l'entreprise.

Dans un contexte de gestion des connaissances, les connaissances se répartissent en plusieurs types : les connaissances explicites et tacites (ou implicites), les connaissances statiques et dynamiques et les connaissances individuelles et collectives.

Le problème qui nous intéresse ici est celui de la gestion de l'imprécision et de l'incertitude sur des connaissances qui peuvent être numériques ou symboliques.

La logique floue est l'axe de travail que nous avons choisi pour atteindre cet objectif, qui sera présenté dans le chapitre suivant.

Chapitre 2 : La logique floue

Introduction :

Depuis longtemps l'homme recherche à maîtriser les incertitudes et les imperfections inhérentes à sa nature. La première réelle manifestation de la volonté de formaliser la prise en compte des connaissances incertaines fut le développement de la théorie des probabilités à partir du XVII^e siècle. Mais les probabilités ne peuvent maîtriser les incertitudes psychologiques et linguistiques. On a donc assisté aux développements des théories de probabilité subjective (dans les années 50) puis de l'évidence (dans les années 60).

Puis la Logique Floue est apparue en 1965 à Berkeley dans le laboratoire de Lotfi Zadeh avec la théorie des sous-ensembles flous puis en 1978 avec la théorie des possibilités. Ces deux théories constituent aujourd'hui ce que l'on appelle la Logique Floue.

La Logique Floue permet la formalisation des imprécisions dues à une connaissance globale d'un système très complexe et l'expression du comportement d'un système par des mots.

Elle permet donc la standardisation de la description d'un système et du traitement de données aussi bien numériques qu'exprimées symboliquement par des qualifications linguistiques.

Chapitre 2 : La logique floue

I. Principe de la logique floue :

Le concept de logique floue vient de la constatation que la variable booléenne, qui ne peut prendre que deux valeurs (vrai ou faux) est mal adaptée à la représentation de la plupart des phénomènes courants. Cela permet de répondre à de nombreuses situations. Mais dans certains cas une transition « abrupte » est gênante. En revanche, la logique floue distingue une infinité de valeurs de vérité (entre 0 et 1).

Prenons un exemple simple

Un patient atteint d'hépatite présente généralement les symptômes suivants :

- Le patient a une forte fièvre,
- sa peau présente une coloration jaune,
- il a des nausées.

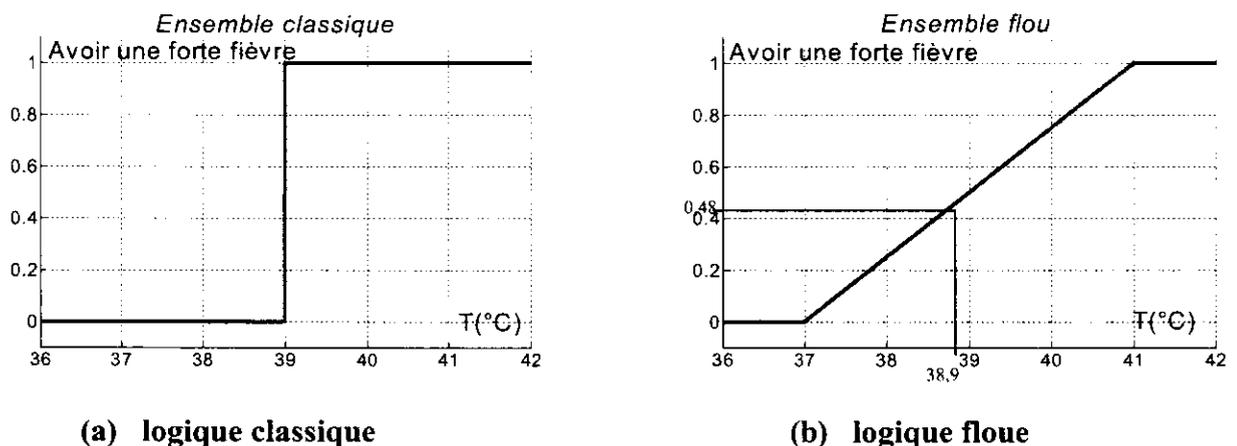


Figure 2.1 : Niveaux d'appartenance

En logique classique, le patient soit à une forte fièvre, soit n'a pas de forte fièvre. Stupide non ?

Si le patient à 39°C de température \Rightarrow Le patient a une forte fièvre

Si le patient à 38,9°C de température \Rightarrow Le patient n'a pas de forte fièvre donc n'a pas d'hépatite, cette discontinuité est totalement absurde.

En logique floue

Si le patient à 38,9°C de température \Rightarrow Le patient a une forte fièvre à 48% (facteur d'appartenance=0.48) donc Le patient a une hépatite à x %.

Chapitre 2 : La logique floue

I.1. Conditions d'utilisation :

Les raisons les plus générales qui peuvent conduire à utiliser la logique floue sont :

-Les connaissances concernant le système à réaliser sont imparfaites, c'est-à-dire que certaines d'entre elles sont soumises à des imprécisions, éventuellement aussi à des incertitudes. Par exemple, des données sont fournies par des instruments d'observation dont la précision n'est pas absolue.

-Certaines de ces connaissances sont exprimées avec des termes du langage naturel, par un expert ou un observateur humain, et d'autres sont fournies numériquement par des instruments d'observation. Le traitement simultané de connaissances numérique ou symbolique n'est réalisable de façon simple que dans le cadre de la logique floue.

-Il existe des incertitudes de nature non numérique sur certaines connaissances, sur certaines connaissances, reflétant la fiabilité de celui qui les fournit ou la variabilité des caractéristiques décrites, ou encore des difficultés d'observation.

-Il existe des connaissances graduelles sur le système.

-Il existe des classes aux limites mal définies, dont les frontières ne peuvent pas être indiquées précisément, ou des catégories mal séparées qui se chevauchent partiellement.

-Certaines des caractéristiques des attributs décrivant le système ne peuvent pas être décrites avec précision et sont nécessairement vagues en raison d'une trop grande variabilité de ces caractéristiques.

-Il n'existe pas de modèle rigoureux du problème que l'on doit traiter ou il existe un modèle d'une très grande complexité.

Si l'une au moins de ces caractéristiques est présente dans le système à traiter, il est intéressant de faire appel à la logique floue pour effectuer la réalisation visée.

I.2. Domaine d'application :

Les principaux domaines de recherche et d'application de la logique floue sont les suivants :

Les domaines techniques :

- Systèmes experts
- Bases de données (objets flous et/ou requêtes floues)
- Commande de processus
- Langage de programmation
- Traitement de l'information telles que les données, recherche de l'information, modélisation des systèmes.....etc.

Chapitre 2 : La logique floue

Les domaines applicatifs :

- médecine et biologie (cancérologie, cardiologie, génétique, psychiatrie, aide au diagnostic médical...)
- génie industriel (génie civil, génie électrique, énergie nucléaire, conception industriel....)
- sciences humaines (linguistique, psychologie, géographie, sociologie.....)
- Recherche scientifique (mécanique quantique, chimie, statistiques.....)....etc.

II. Théorie des ensembles flous :

II.1. Définition d'un sous-ensemble flou :

Soient X : L'univers du discours.

A : un sous-ensemble de X .

Un *sous-ensemble flou* A de X est défini par une *fonction d'appartenance* qui associe à chaque élément x de X , le *degré* $f_A(x)$, compris entre 0 et 1, avec lequel x appartient à A :

$$f_A: X \longrightarrow [0,1]$$

Dans le cas particulier où f_A ne prend que des valeurs égales à 0 ou 1, le sous- ensemble flou A est un sous-ensemble classique de X .

$$\text{Notation : } \quad A = \sum_{x \in X} f_A(x) / x, \quad \text{si } X \text{ est dénombrable,}$$
$$A = \int_X f_A(x) / x, \quad \text{si } X \text{ est non dénombrable.}$$

où $f_A(x)$ est le degré d'appartenance de l'élément x à A et le signe / un séparateur.

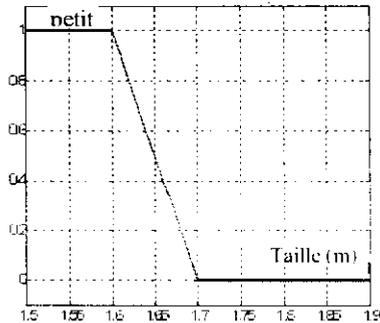
Exemple :

On peut considérer une variable linguistique « taille » caractérisée selon trois classes petites, moyennes et grandes.

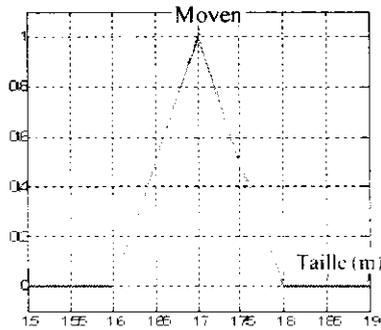
La représentation de ses classes se fait au moyen de trois fonctions qui calculent l'appartenance de chaque personne à l'ensemble flou petit, moyenne et/ou grand.

Chapitre 2 : La logique floue

Ensemble flou personne de petit taille



Ensemble flou personne de taille moyenne



Ensemble flou personne de grand taille

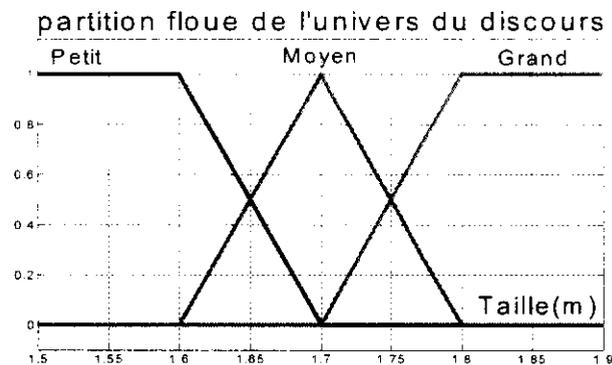
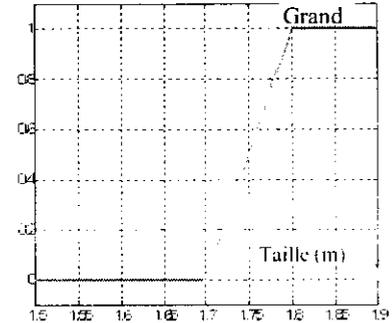


Figure 2.2 : Représentation des ensembles flous

Ici, Amine mesure 1m625 dans cet exemple, se traduit en logique floue par Amine est considéré comme « petit » avec un degré de 0.75 (75%) comme « moyen » avec un degré de 0.25 (25%) et comme « grand » avec un degré de 0 (0%).

Différentes formes de fonctions d'appartenance :

Tout ensemble flou peut être représenté par une fonction d'appartenance. Elles peuvent avoir plusieurs formes :

- monotones (croissantes ou décroissantes) (figures 2.3 et 2.4)
- triangulaires (fig 2.5 a)
- trapézoïdales (fig 2.5 b)
- en forme de cloche (Gaussiennes) (fig. 2.5 c)

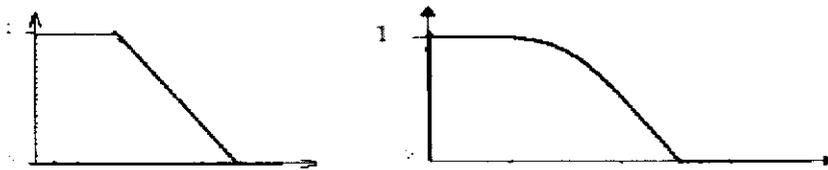


Figure 2.3 : Exemples de fonctions d'appartenance monotones décroissantes

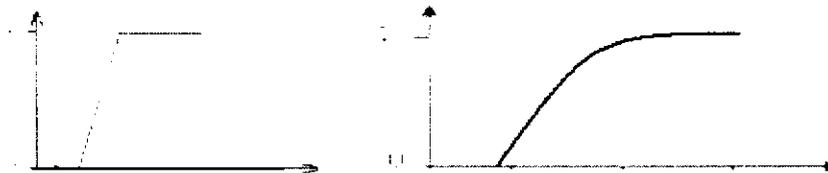


Figure 2.4 : Exemples de fonctions d'appartenance monotones croissantes

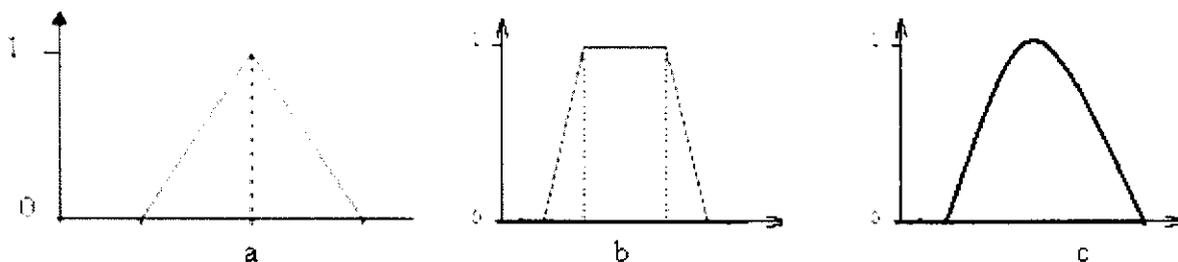


Figure 2.5 : Exemples de fonctions d'appartenance

II.2. Caractéristique d'un sous-ensemble flou :

Un sous-ensemble flou est complètement défini par la donnée de sa fonction d'appartenance. A partir d'une telle fonction, un certain nombre de caractéristiques du sous-ensemble flou peuvent être étudiées.

II.2.1. Support :

Le *support* de A, noté $\text{supp}(A)$, est la partie de X sur laquelle la fonction d'appartenance de A n'est pas nulle :

$$\text{supp}(A) = \{x \in X / f_A(x) \neq 0\}$$

II.2.2. Hauteur :

La *hauteur*, notée $h(A)$, du sous-ensemble flou A de X est la plus grande valeur prise par sa fonction d'appartenance :

$$h(A) = \sup_{x \in X} f_A(x).$$

Le sous-ensemble flou A de X est *normalisé* si sa hauteur $h(A)=1$.

II.2.3. Noyau :

Le *noyau* de A, noté $\text{noy}(A)$, est l'ensemble des éléments de X pour lesquels la fonction d'appartenance de A vaut 1 :

$$\text{noy}(A) = \{x \in X / f_A(x) = 1\}.$$

II.2.4. Cardinalité :

La *cardinalité* du sous-ensemble flou A de X est définie par :

$$|A| = \sum_{x \in X} f_A(x).$$

II.2.5. a-coupe :

Une a-coupe d'un sous-ensemble flou est

$$B_\alpha = \{x, f_B(x) \geq \alpha\}, 0 \leq \alpha \leq 1$$

C'est le sous-ensemble de tous les éléments ayant un degré d'appartenance supérieur ou égal à α .

Les a-coupes vérifient :

- $(A \cup B)_\alpha = A_\alpha \cup B_\alpha$
- $(A \cap B)_\alpha = A_\alpha \cap B_\alpha$
- $A \subset B \Rightarrow A_\alpha \subset B_\alpha$
- $A_1 = \text{noy}(A)$
- $A_0 = X$

Une a-coupe stricte est :

$$B^\alpha = \{x, f_B(x) > \alpha\}, 0 \leq \alpha < 1$$

Pour une fonction d'appartenance $f_A(x)$ de forme trapézoïdale, le support, la hauteur et le noyau sont définis comme suit :

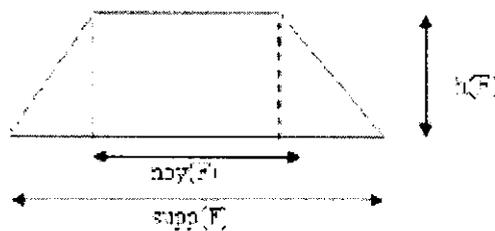


Figure 2.6 : Caractéristiques d'un ensemble flou

Chapitre 2 : La logique floue

Exemple :

Soit X un ensemble fini de pays par exemple :

$X = \{\text{France, Belgique, Suisse, Canada, Espagne, Allemagne}\}$, l'ensemble des pays susceptibles d'être un lieu de résidence pour un individu donné, notés A, B, E, G, I. On peut définir les sous-ensembles flous suivants, correspondant à la description des souhaits de l'individu :

$$A_1 = 0.6 / A + 0.7 / B + 0.4 / E + 0.3 / F + 0.8 / G + 0.5 / I,$$

$$\text{supp}(A_1) = X, h(A_1) = 0.8, \text{noy}(A_1) = \emptyset, |A_1| = 3.3$$

$$A_2 = 0 / A + 0 / B + 1 / E + 0.8 / F + 0 / G + 1 / I.$$

$$\text{supp}(A_2) = \{E, F, I\}, h(A_2) = 1 (\text{sous-ensemble flou normalisé}), \text{noy}(A_2) = \{E, I\} \\ |A_2| = 2.8$$

II.3. Opérations sur les sous-ensembles flous :

Comme pour la théorie classique des ensembles. On définit la réunion, l'intersection, le complément...d'ensembles flous

La logique booléenne standard est un cas particulier de la logique floue donc tous les résultats obtenus en logique classique doivent être retrouvés par la logique floue.

II.3.1. Egalité et inclusion de sous-ensembles flous :

Deux sous-ensembles flous A et B d'un même ensemble de référence X sont *égaux* si leurs fonctions d'appartenance prennent la même valeur pour tout élément de X :

$$\forall x \in X \quad f_A(x) = f_B(x)$$

Etant donné deux sous-ensembles flous A et B de X, on dit que A est *inclus* dans B et on note $A \subseteq B$, si leurs fonctions d'appartenance sont telles que :

$$\forall x \in X \quad f_A(x) \leq f_B(x)$$

II.3.2. Intersection et union de sous-ensembles flous :

L'*intersection* de deux sous-ensembles flous A et B de X est le sous-ensemble flou C, que l'on note $A \cap B$, tel que :

$$\forall x \in X \quad f_C(x) = \min(f_A(x), f_B(x)),$$

min désignant l'opérateur de minimisation .

Chapitre 2 : La logique floue

Le degré avec lequel chaque élément x de X appartient à $A \cap B$ est le plus petit des degrés avec lesquels il appartient à A et à B .

L'union de deux sous-ensembles flous A et B de X est le sous-ensemble flou D , que l'on note $A \cup B$, tel que :

$$\forall x \in X \quad f_D(x) = \max(f_A(x), f_B(x)),$$

max désignant l'opérateur de maximisation.

Le degré avec lequel chaque élément x de X appartient à $A \cup B$ est le plus grand des degrés avec lesquels il appartient à A et à B .

Exemple :

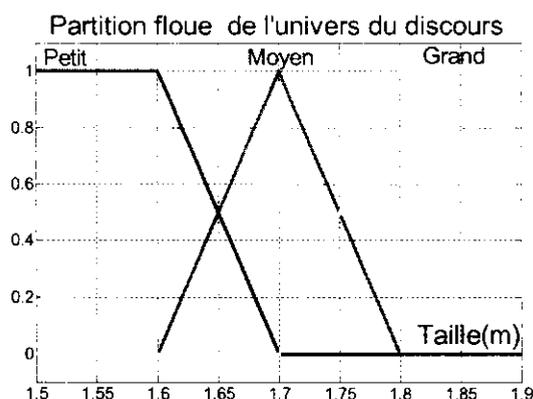


Figure 2.7 : Fonctions d'appartenance avec trois ensembles pour la variable «taille»

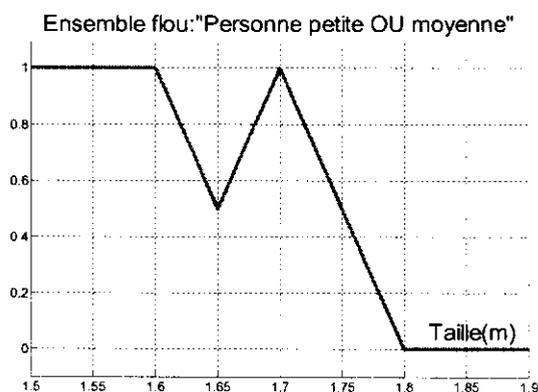


Figure 2.8 : Opérateur OU

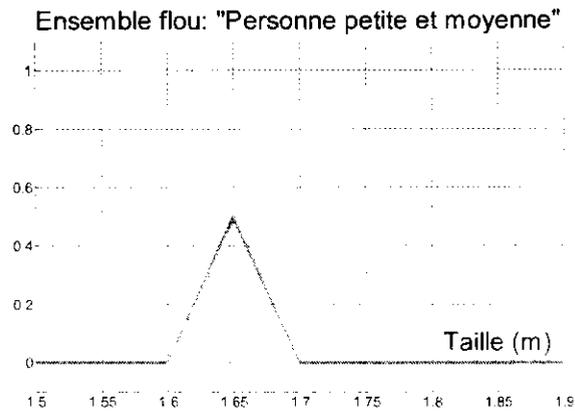


Figure 2.9 : Opérateur ET

II.3.3. Complément d'un sous-ensemble flou :

Le complément A^c d'un sous-ensemble flou A de X est défini comme le sous-ensemble flou de X de fonction d'appartenance :

$$\forall x \in X \quad f_{A^c}(x) = 1 - f_A(x).$$

Exemple :

L'ensemble des personnes NON petites est un ensemble flou de fonction d'appartenance :

$$f_{A^c}(x) = 1 - f_A(x) \quad \forall x \in X$$

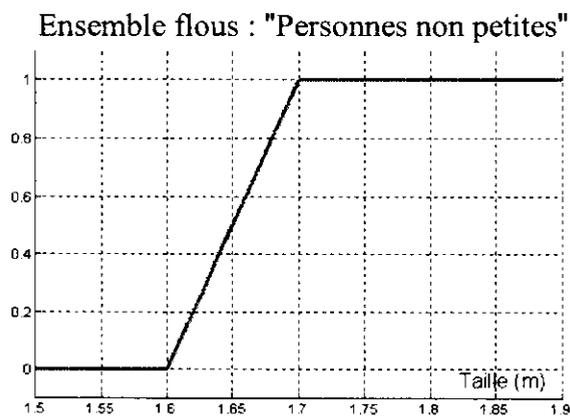


Figure 2.10: Opérateur NON

II.4. Relations et Quantités floues :

II.4.1. Relations floues :

Une relation floue R entre r ensembles de référence X_1, X_2, \dots, X_r , est un sous-ensemble flou de $X_1 \times X_2 \times \dots \times X_r$, de fonction d'appartenance f_R .

Cas particulier : Si $X=Y$, une relation floue R définie sur les deux univers X et Y est une *relation binaire floue* définie sur X .

L'*inverse* de la relation floue R entre X et Y est une relation floue R^{-1} entre Y et X définie par :

$$\forall y \in Y, \forall x \in X, f_{R^{-1}}(y, x) = f_R(x, y).$$

Composition de relations floues :

La *composition* de deux relations floues R_1 sur $X \times Y$ et R_2 sur $Y \times Z$ définit une relation floue $R = R_1 \circ R_2$ sur $X \times Z$ de fonction d'appartenance définie par :

$$\forall (x, z) \in X \times Z, f_R(x, z) = \sup_{y \in Y} \min(f_{R_1}(x, y), f_{R_2}(y, z)).$$

Relations de similarité :

Une *relation de similarité* est une relation floue symétrique, réflexive et transitive max-min.

II.4.2. Quantités floues :

Les quantités floues sont des sous-ensembles flous normalisés (de hauteur 1), c'est-à-dire des sous-ensembles flous dont un élément au moins (la valeur modale) à une appartenance totale. On distingue, parmi les quantités floues, les intervalles flous (intervalles aux limites imprécises, généralement représentés par des trapèzes notés (a, b, c, d) et les nombres flous (intervalles flous ayant une valeur modale unique, généralement représentés par des triangles notés (a, b, c)).

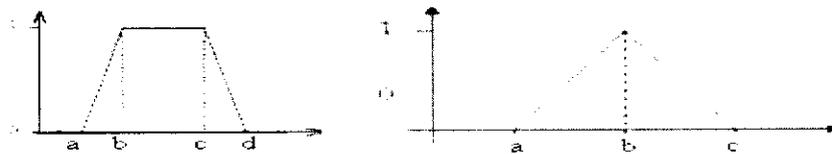


Figure 2.11: Exemple d'intervalle flou et de nombre flou

III. Eléments de base de la logique floue :

III.1. Variables linguistiques :

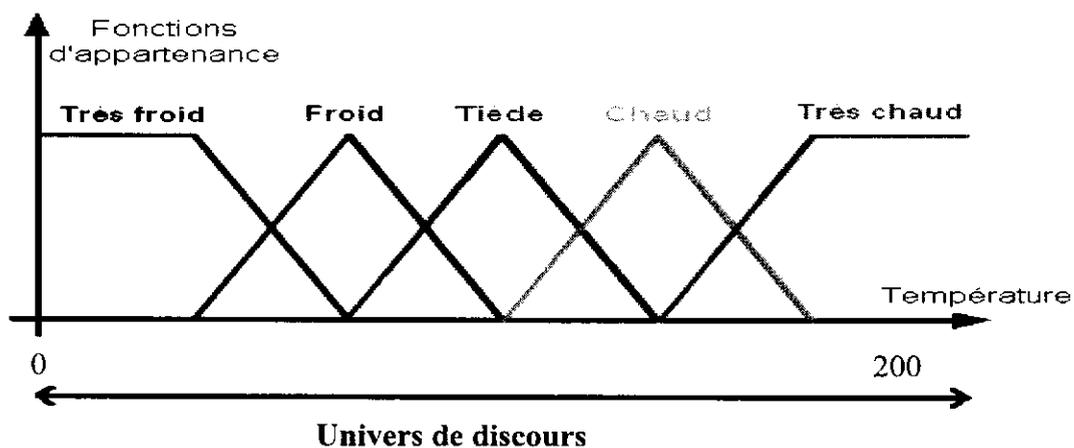
La logique floue est basée sur des variables floues dites *variables linguistiques* à valeurs linguistiques dans l'univers du discours X.

Chaque *valeur linguistique* constitue alors un ensemble flou de l'univers du discours.

Chaque variable linguistique est représentée par un triplet (V, X, T_v) où

- V est le nom de la variable,
- T_v est l'ensemble des valeurs linguistiques que peut prendre V,
- X est l'univers de discours associé avec la valeur de base.

Exemple :



Univers du discours : Gamme de température de 0°C à 200°C.

Variable linguistique : La température.

Valeurs linguistiques : « Très froid » « Froid » « Tempéré » « Chaud » « Très Chaud »

III.2. Opérateurs linguistiques :

Dans le langage humain, on utilise très souvent les opérateurs linguistiques comme : *très, près de, environ, presque, plus ou moins*, etc. Ces opérateurs s'appellent les modificateurs des ensembles flous. Ces modificateurs permettent de moduler une proposition en l'atténuant ou en la renforçant.

Exemple :

La fonction d'appartenance du sous-ensemble flou "très petit" peut être définie par :

$$f_{\text{très_petit}}(x) = (f_{\text{petit}}(x))^2$$

III.3. Propositions floues :

Étant donné un ensemble L de variables linguistiques et un ensemble M de modificateurs, une *proposition floue élémentaire* est définie à partir d'une variable linguistique (V, X, T_v) de L par la qualification 'V est A'

où :

V : c'est un variable prenant une valeur x dans X

A : c'est un sous-ensemble flou défini en X, qui représente un prédicat flou

Une *proposition floue générale* est obtenue par conjonction, disjonction, négation et implication de propositions floues quelconques.

Exemple :

La taille est moyenne

La vitesse est plutôt rapide

La surface est grande et le prix est élevé

III.4. Implications floues :

Dans la logique classique, on raisonne en utilisant le *modus tollens* et le *modus ponens*. L'implication floue est basée sur les mêmes principes, mais les propositions basées sur les valeurs de vérité floues sont permises.

On va utiliser le *modus tollens généralisé* :

- Prémisse : y est B'
- Implication : Si x est A, alors y est B
- Conséquence : x est A'

Chapitre 2 : La logique floue

On va utiliser le *modus ponens généralisé* :

- Prémisse : x est A'
- Implication : Si x est A , alors y est B
- Conséquence : y est B'

où x et y sont des variables linguistiques : A, A', B, B' sont les valeurs linguistiques définies dans les univers de discours de X et Y .

Pour une règle de la forme Si x est A , alors y est B , une conclusion doit être obtenue relativement à y lorsque la donnée disponible est de la forme x est A' , pour A' plus ou moins différent de A , deux propositions floues « x est A » et « x est A' » sont considérées comme proches dans la mesure où les sous-ensembles flous A et A' ont des fonctions d'appartenance peu différentes.

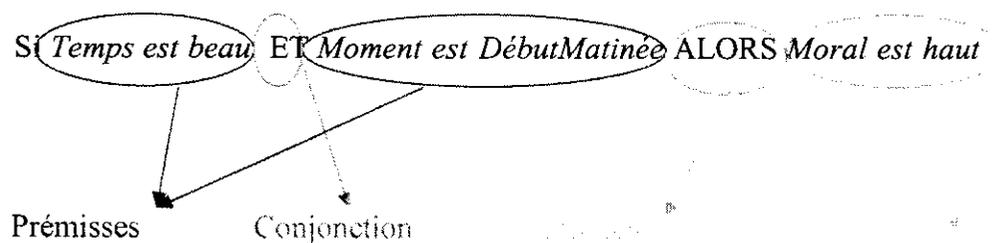
Les conditions d'utilisation du *modus ponens généralisé* sont les suivantes :

Règle floue :	Si x est A , alors y est B	
Fonctions d'appartenance :	f_A	f_B
Fait observé :	x est A'	
Fonctions d'appartenance :	$f_{A'}$	
Conclusion	y est B'	
Fonctions d'appartenance :	$f_{B'}$	

III. 5. Règles floues :

Une *règle floue* est une proposition floue de la forme " Si p alors q " utilisant une implication entre deux propositions floues quelconques p et q .

Exemple :



IV. Théorie des possibilités :

La théorie des possibilités a été introduite en 1978, également par Lotfi Zadeh, pour permettre de manipuler des incertitudes de nature non probabiliste, donc auxquelles les moyens classiques de la théorie des probabilités n'apportent pas de solution, et pour constituer un cadre dans lequel connaissances imprécises et connaissances incertaines peuvent coexister et être traitées conjointement. [4a]

Cette théorie est le deuxième volet de la Logique Floue. Elle est basée sur les fonctions de possibilités comme l'autre l'est sur les fonctions d'appartenance.

IV.1. Mesure de possibilité :

Une *mesure de possibilité* Π est une fonction définie sur l'ensemble $P(X)$, prenant ses valeurs dans $[0,1]$, telle que :

$$i) \Pi(\emptyset) = 0, \quad \Pi(X) = 1,$$

$$ii) \forall \text{ famille d'ensembles } A_1 \in P(X), A_2 \in P(X), \dots$$

$$\Pi(\cup_{i=1,2,\dots} A_i) = \sup_{i=1,2,\dots} \Pi(A_i)$$

X : un ensemble de référence.

$P(X)$: un sous-ensemble ordinaire de X .

Cas particulier :

La propriété (ii) se réduit, dans le cas de deux sous-ensembles de X , à :

$$\forall (A, B) \in P(X)^2 \quad \Pi(A \cup B) = \max(\Pi(A), \Pi(B))$$

- Si $\Pi(A) = 0$ alors l'événement A est complètement impossible,
- si $\Pi(A) = 1$ alors l'événement A est complètement possible,
- et si $0 < \Pi(A) < 1$ alors l'événement A est plus ou moins possible.

IV.2. Distribution de possibilité :

Une *distribution de possibilité* π est une fonction définie sur X , prenant ses valeurs dans $[0,1]$, satisfaisant la condition de normalisation suivante :

$$\sup_{x \in X} \pi(x) = 1.$$

Propriétés :

- La fonction Π définie à partir de la distribution de possibilité π par :

$$\forall A \in P(X) \quad \Pi(A) = \sup_{x \in A} \pi(x)$$

est une mesure de possibilité.

- La fonction π définie à partir de la mesure de possibilité Π par :

$$\forall x \in X \quad \pi(x) = \Pi(\{x\})$$

est une distribution de possibilité, si X est dénombrable.

Cas particulier :

La fonction caractéristique d'un sous-ensemble S de X peut être regardée comme une distribution de possibilité π définie sur X . S'il est tout à fait possible qu'un élément x de X appartienne à S et tout à fait impossible qu'il n'appartienne pas, alors

$\pi(x) = 1$ pour tout x appartenant à S et $\pi(x) = 0$ pour tout x appartenant à son complément S^c .

A partir de la connaissance du sous-ensemble S auquel appartient l'élément x de X , on peut alors déterminer avec quel degré il est possible que x appartienne à un sous-ensemble quelconque A de X , et la fonction définie sur $P(X)$ par :

$$\Pi(A) = 1 \text{ si } A \cap S \neq \emptyset,$$

$$\Pi(A) = 0 \text{ sinon}$$

est une mesure de possibilité.

IV.3. Mesure de nécessité :

Une *mesure de nécessité* est une fonction N définie sur l'ensemble $P(X)$ des parties de X , à valeurs dans $[0,1]$, telle que :

$$\text{i) } N(\emptyset) = 0, \quad N(X) = 1,$$

$$\text{ii) } \forall A_1 \in P(X), A_2 \in P(X), \dots, N(\bigcap_{i=1, 2, \dots} A_i) = \inf_{i=1, 2, \dots} N(A_i),$$

Chapitre 2 : La logique floue

Cas particulier :

La propriété (ii) se réduit, dans le cas de deux sous-ensembles de X, à :

$$\forall (A, B) \in P(X)^2 \quad N(A \cap B) = \min(N(A), N(B))$$

Propriétés :

- Soit A, B deux sous-ensembles de X, sont tels que $B \subseteq A$, alors $N(A) \geq N(B)$. En particulier, $N(A) \geq N(A \cap B)$
- $\forall (A, B) \in P(X)^2 \quad N(A \cup B) \geq \max(N(A), N(B))$
- Tout sous-ensemble A de X vérifie :

$$\begin{aligned} \min(N(A), N(A^c)) &= 0 \\ N(A) + N(A^c) &\leq 1 \end{aligned}$$

- Soit S un sous-ensembles de X, auquel appartient l'élément x de X, on peut alors déterminer avec quel degré il est certain que x appartienne à un sous-ensemble quelconque A de X, et la fonction définie sur P(X) par :

$$\begin{aligned} N(A) &= 1 \quad \text{si } S \subseteq A \\ N(A) &= 0 \quad \text{sinon} \end{aligned}$$

est une mesure de nécessité.

IV.4. Relations entre mesure de possibilité et de nécessité

Etant donné un ensemble de référence X, soit A^c le complémentaire de toute partie A de X. Une mesure de nécessité N peut être obtenue à partir de la donnée d'une mesure de possibilité Π , par :

$$\forall A \in P(X) \quad N(A) = 1 - \Pi(A^c)$$

Plus l'événement A est affecté d'une grande nécessité, moins l'événement complémentaire A^c est possible, donc plus on est certain de la réalisation de A.

Si la mesure de possibilité Π est définie à partir d'une distribution de possibilité π , on peut définir la mesure de nécessité N duale de Π par :

$$\forall A \in P(X) \quad N(A) = \inf_{x \notin A} (1 - \pi(x))$$

Chapitre 2 : La logique floue

Quelle que soit la partie A de X, mesure de possibilité et mesure de nécessité duales vérifient :

- $\Pi(A) \geq N(A)$,
- $\max(\Pi(A), 1 - N(A)) = 1$,
- Si $N(A) \neq 0$, alors $\Pi(A) = 1$,
- Si $\Pi(A) \neq 1$, alors $N(A) = 0$.

IV.5. Mesures floues :

Une *mesure floue* est une fonction g , définie sur l'ensemble $P(X)$ des parties de X, à valeurs dans $[0,1]$, telle que :

- i) $g(\emptyset) = 0, g(X) = 1$,
- ii) si $B \subseteq A$, alors $g(A) \geq g(B)$

La propriété (ii) de monotonie de g implique que :

$$g(A \cup B) \geq \max(g(A), g(B))$$

$$g(A \cap B) \leq \min(g(A), g(B))$$

Remarque:

Les mesures de possibilité, de nécessité sont des mesures floues.

Chapitre 2 : La logique floue

Conclusion :

La logique floue est actuellement le seul cadre dans lequel puissent être traitées des imprécisions et des incertitudes qui autorise également le traitement de certaines incomplétudes, et le seul cadre dans lequel puissent être traitées des connaissances numériques et des connaissances exprimées symboliquement.

Le concept de sous-ensemble flou a été introduit pour gérer des imprécisions, et la théorie des possibilités ayant été introduit pour gérer des incertitudes.

Une alternative permet donc de raisonner sur des connaissances imparfaites :

Ou bien les connaissances disponibles sont imprécises et éventuellement aussi incertaines, et on utilise la logique floue

Ou bien elles sont seulement incertaines, et on utilise la logique possibiliste.

Dans notre travail, nous proposons d'utiliser une base de cas où les attributs ont des valeurs floues ou imprécises. Ce système sera présenté dans le chapitre suivant.



Chapitre 3 : Le raisonnement à base de cas RBC

Introduction :

Le raisonnement à base de cas (RBC) diffère à bien des égards des autres approches de l'intelligence artificielle. Au lieu d'exploiter la connaissance générale d'un domaine, le RBC vise à utiliser la connaissance spécifique et pragmatique des problèmes précédemment expérimentés (les cas). Un nouveau problème est résolu en trouvant un cas passé semblable et en le réutilisant dans la nouvelle situation. Le RBC est, de plus, une approche progressive, évolutive puisqu'une nouvelle expérience est conservée chaque fois qu'un problème a été résolu, la rendant immédiatement disponible pour des problèmes futurs.

Le raisonnement à partir de cas est un processus qui semble très naturel et il est bien évidemment souvent employé pour résoudre des problèmes de la vie courante. On peut ici faire deux observations qui, bien qu'évidentes, légitiment l'utilisation du RBC :

- Les problèmes similaires ont des solutions similaires
- On se retrouve souvent face à un problème que l'on a déjà rencontré.

L'étude de RBC est conduite par deux motivations principales : la première, venant des sciences cognitives, trouve sa source dans le désir de copier le raisonnement humain et la deuxième, venant de l'intelligence artificielle, vise à développer des techniques d'IA et de raisonnement plus efficaces.

Le raisonnement basé sur les cas Possède un vaste domaine d'application, il est souvent utilisé dans les domaines où l'on sait décrire les cas, comme en droit, en médecine, en réglementation. Un cas est constitué d'un problème source et de sa solution source associée. Le RBC peut être modélisé par un cycle constitué de quatre étapes : recherche, réutilisation, révision et apprentissage, gravitant autour d'une base de connaissances du domaine d'application. Chacune des étapes du cycle mobilise ces connaissances pour supporter la recherche de la solution du problème cible.

Dans ce chapitre, nous définirons la technique de raisonnement à base de cas, puis nous en présenterons les composants : les connaissances et les étapes du cycle de raisonnement à base de cas.

Chapitre 3 : Le raisonnement à base de cas RBC

I. Le principe du Raisonnement à Base de Cas (RBC) :

Le raisonnement à base de cas est une approche de résolution de problèmes qui utilise des expériences passées pour résoudre de nouveaux problèmes. L'ensemble des expériences forme une base de cas. Typiquement un cas contient au moins deux parties : une description de situation représentant un problème et une solution utilisée pour remédier à cette situation. Parfois, le cas décrit également les conséquences résultant de l'application de la solution (ex : succès ou échec). Les techniques RBC permettent de produire de nouvelles solutions en extrapolant sur les situations similaires au problème à résoudre. Cette approche est adéquate pour les domaines où la similarité entre les descriptions de problèmes nous donne une indication de l'utilité des solutions antécédentes.

Comme nous l'avons dit le RBC est un mode de résolution de problèmes qui est relativement différent des autres approches d'IA. Au lieu de se reposer seulement sur des connaissances générales d'un domaine, ou de faire des associations générales entre descriptions du problème et conclusions, le RBC est capable d'utiliser les connaissances spécifiques d'expériences précédentes, les cas.

Illustrons ces principes généraux par quelques exemples :

- Un médecin après avoir examiné un patient se souvient d'un autre patient qu'il a traité deux semaines plus tôt. Ce souvenir est causé par une similarité importante entre les symptômes des deux patients, le médecin utilise le diagnostic et le traitement du patient précédent pour déterminer le cas du nouveau patient.
- Un chimiste, qui a été confronté à deux explosions dramatiques, se rappelle soudain de l'une de ces situations lorsque les conditions d'une expérience sont semblables à celles qui avaient provoqué les explosions. Il utilise alors son expérience d'un cas passé pour éviter de provoquer un nouvel incident.
- etc.

II. Définitions :

Le RBC se définit par ses buts mais aussi par les étapes qui le constituent.

Plusieurs définitions de raisonnement à partir de cas existent :

D'après Schank [1]: « Le raisonnement à base cas résout de nouveaux problèmes par adaptation des solutions qui ont été utilisées pour résoudre des anciens problèmes ».

D'après Riesbeck [1] : le raisonnement à partir des cas peut être défini comme :

« Retrouver un cas antérieur dans la mémoire, tenter de déterminer sa pertinence et décider ce qu'il y a à faire en fonction du cas extrait ».

Chapitre 3 : Le raisonnement à base de cas RBC

D'après Carbonell [1] le raisonnement à partir des cas est une forme de raisonnement analogique et « la résolution de problème analogique consiste à transférer des connaissances à partir d'épisodes passés de résolution de problèmes aux nouveaux problèmes qui partagent des aspects significatifs avec l'expérience passée correspondante et à utiliser la connaissance transférée pour construire des solutions aux nouveaux problèmes ».

D'après Kolonder [1], Le raisonnement à base cas peut être défini comme «...Adapter les anciens solutions aux nouvelles demandes, utiliser les anciens cas pour expliquer les nouvelles situations...».

L'idée principale que cache le concept RBC est que « les mêmes problèmes ont les mêmes solutions » ou mieux encore « la solution d'un problème est un bon point de départ pour la résolution du problème courant ».

III. Les composants d'un système RBC :

Un système RBC est une combinaison de processus et de connaissances qui permettent de préserver et d'exploiter les expériences passées.

Pour simplifier notre présentation, nous nous appuyons sur le modèle générique proposé par Lamontagne & Lapalme [18] présenté dans la **Figure 1** :

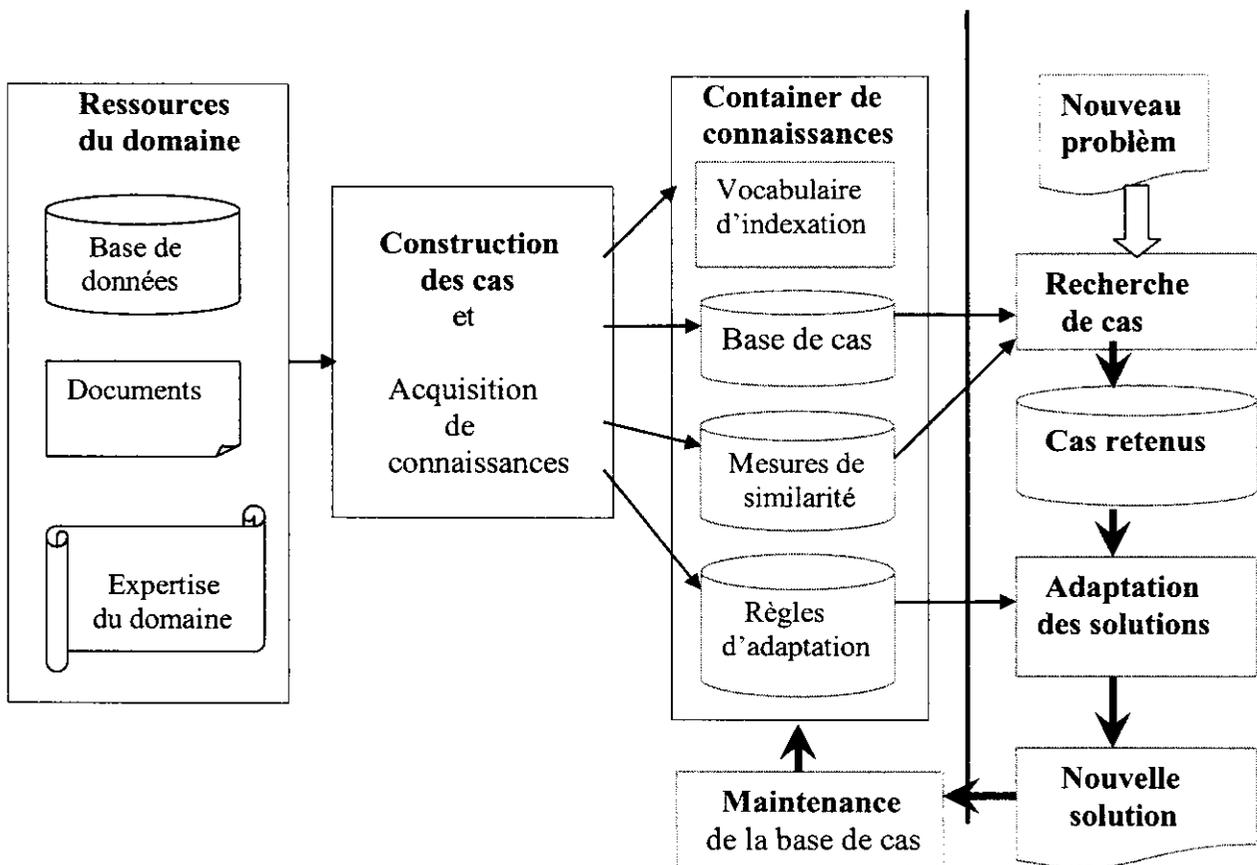


Figure 3.1 : Modèle générique d'un système RBC

Chapitre 3 : Le raisonnement à base de cas RBC

On y note comme principaux processus la recherche, l'adaptation, la maintenance et la construction et comme structures de connaissances le vocabulaire d'indexation, la base de cas, les métriques de similarité et les connaissances d'adaptation.

III.1. Processus :

- **La recherche** : cette phase permet de déterminer les cas de la base qui sont les plus similaires au problème à résoudre.
- **L'adaptation** : suite à la sélection de cas lors de la phase de recherche, le système RBC aide l'utilisateur à modifier et à réutiliser les solutions de ces cas pour résoudre son problème courant.
- **Maintenance** : permet de mémoriser le nouveau problème, soit par la création d'un nouveau cas, soit par la modification des parties du cas déjà dans la base.
- **Construction** : ce processus, en amont des activités de résolution de problèmes du système RBC, guide la structuration initiale de la base de cas et des autres connaissances du système à partir de différentes ressources tels des documents, bases de données ou transcriptions d'interviews avec des praticiens du domaine. Ce processus, souvent effectué manuellement par le concepteur du système, se prête moins bien à l'automatisation car il nécessite une connaissance du cadre applicatif pour guider, entre autre, la sélection du vocabulaire d'indexation et la définition des métriques de similarités.

III. 2. Connaissances :

Les différentes connaissances utilisées par un système RBC sont regroupées en quatre catégories ("container de connaissances") :

- **Vocabulaire d'indexation** : un ensemble d'attributs qui caractérisent la description de problèmes et de solutions du domaine. Ces attributs sont utilisés pour construire la base de cas et jouent un rôle important lors de la phase de recherche.
- **Base de cas** : l'ensemble des expériences structurées qui seront exploitées par les phases de recherche, d'adaptation et de maintenance.
- **Mesures de similarité** : des fonctions pour évaluer la similarité entre deux ou plusieurs cas. Ces mesures sont définies en fonction des traits et sont utilisées pour la recherche dans la base de cas.

Chapitre 3 : Le raisonnement à base de cas RBC

- *Connaissances d'adaptation* : ce sont des heuristiques du domaine, habituellement sous forme de règles, permettant de modifier les solutions et d'évaluer leur applicabilité à de nouvelles situations.

IV. Structure d'un cas :

La représentation des cas prend une place importante dans la réalisation d'un système RBC. cette représentation va déterminer l'efficacité et la rapidité de la recherche des cas dans la base. Il est donc nécessaire de choisir les informations à stocker dans chaque cas et de trouver sous quelle forme.

C'est quoi un cas ?

Un cas est un morceau contextuel de connaissance représentant une expérience. Il contient une leçon passée qui est le contenu du cas et le contexte dans lequel la leçon peut être employée. Un cas comprend *le problème* qui décrit l'état des choses quand le cas est arrivé, *la solution* qu'expose la solution tirée de ce problème, et/ou *le résultat* qui décrit l'état des choses après le cas.

En démonstration automatique par exemple un problème est l'énoncé d'un problème et la solution en est la preuve. Dans le cas de diagnostic médical un problème est une description de l'état d'un patient et la solution est le diagnostic associé à cet état.

Que doit-il contenir ?

Les parties qui constituent un cas sont :

- La description du problème ou de la situation, ceci comprend l'état du monde quand le cas a existé.
- La solution.
- Le résultat d'application de la solution (succès ou échec).

Quelle doit être sa taille ?

La troisième question évoquée à laquelle il est relativement aisé d'apporter une réponse concerne la taille que doit avoir un cas.

Un cas doit représenter un sous problème auquel l'utilisateur est typiquement confronté. Par exemple dans le domaine de la conception de bâtiment, si le cas représente tout le bâtiment, sa réutilisabilité et son intérêt seront beaucoup plus faibles que si le bâtiment est décomposé en l'ensemble de ses parties et de ses fonctions.

Comment Collecter un Cas ?

➤ Qu'est-ce qu'un bon cas ?

Il faut commencer par déterminer ce que peut être un "*bon cas*" avant de chercher comment *le collecter*. Un *bon cas* est un cas qui satisfait aux deux conditions suivantes:

- être retrouvé par le système RBC quand il correspond au problème posé par l'utilisateur, au moment voulu.
- proposer la meilleure solution possible à ce problème.

Comment le Collecter ?

Un cas peut être obtenu par extraction et structuration de données déjà enregistrées sous une autre forme, et/ou le plus souvent par dialogue avec l'expert.

Il peut être intéressant d'interviewer plusieurs experts, en même temps ou successivement, mais on risque de se retrouver face à des désaccords entre experts, désaccords qu'il faudra résoudre.

Comment indexer les cas ?

Généralement un cas est un cas est indexé pour permettre de le retrouver suivant certaines caractéristiques pertinentes et discriminantes.

L'indexation est particulièrement utile lorsque le nombre des cas est important, car elle contribue à améliorer la qualité et la rapidité des solutions proposées, en limitant la recherche à des contextes particuliers au lieu de porter sur la totalité de la base.

Organisation des cas en mémoire :

Il faut construire un modèle d'organisation et d'indexation pour relier les cas entre eux. Ce modèle doit posséder certaines qualités. Il est nécessaire que l'ajout assure l'accessibilité aux anciens cas. La recherche de cas similaires doit conserver une complexité constante au fur et à mesure que la base de cas se remplit. Un système de RBC n'étant intéressant qu'avec une base importante de cas, il faut évidemment envisager une solution permettant de retrouver rapidement les cas similaires. Généralement on utilise l'indexation pour cette raison.

Il existe de nombreuses façons d'ordonner les cas, nous nous attarderons sur deux modèles en particulier :

Chapitre 3 : Le raisonnement à base de cas RBC

- **La mémoire plate :**

Dans une mémoire plate, les cas sont rangés séquentiellement dans une simple liste, un tableau ou un fichier. Ils sont retrouvés en appliquant une fonction d'appariement, séquentiellement à tous les cas. Il n'y a aucune organisation particulière des cas. La recherche est très simple.

Les avantages de cette approche sont : La totalité de la base est scrutée; la précision de la recherche est fonction de la qualité de l'appariement. De plus, l'ajout d'un nouveau cas n'est pas coûteux. Cependant, cette structure est coûteuse lorsque la base est trop volumineuse. Pour remédier à cet inconvénient, on utilise des variantes telles que : Indexation superficielle à un niveau pour réduire l'ensemble des candidats. Ou encore, partitionnement de la base en tranches que le système peut distinguer.

- **La mémoire hiérarchique :**

Quand la mémoire des cas est large, il y a nécessité d'organiser les cas hiérarchique. Ceci permet de simplifier la remémoration. Les cas sont organisés en réseaux, dans le sens où les cas qui partagent des caractéristiques communes sont groupés ensemble. Chaque nœud d'un tel graphe contient les caractéristiques partagées par les cas qui sont au niveau inférieur. Les feuilles contiennent les cas eux-mêmes. La méthode générale de la remémoration à partir d'une telle structure garantit une remémoration efficace en terme de temps de calcul par rapport à la mémoire plate. Ceci est dû à l'organisation hiérarchique des cas qui permet de gagner du temps pendant la remémoration. Néanmoins, elle présente un certain nombre d'inconvénients [21] :

- L'ajout des nouveaux cas est une opération compliquée car elle nécessite une mise à jour de l'arbre.
- Il n'est pas aisé de garder l'arbre optimal.
- La nécessité d'un grand espace.
- la remémoration parfaite n'est pas garantie car les cas ne sont pas tous visités.

V. Les tâches et les domaines d'application :

Les tâches réalisées par les systèmes de raisonnement basé sur les cas sont assez variées et peuvent se classer en deux catégories [1].

En effet, un système RBC peut être vu comme un « résolveur » de problème ou bien comme un « interpréteur » de situations.

Chapitre 3 : Le raisonnement à base de cas RBC

V.1. La tâche de résolution de problèmes :

D'une manière générale, dans ce type de tâche, un utilisateur doit résoudre un problème qu'il spécifie au système résolveur. Ce dernier peut soit donner une solution directe au problème posé, soit dialoguer avec l'utilisateur pour l'aider à préciser le problème et arriver ainsi, petit à petit, à une solution acceptable. Dans cette voie on distingue trois types de résolution de problèmes : la planification, la conception et le diagnostic.

Le processus de planification consiste à construire une séquence d'étapes permettant d'arriver à une situation précise à partir d'une situation donnée. Nous pouvons citer ici CHEF qui permet la création de recettes de cuisine.

Dans le domaine de la conception, le problème est défini comme un ensemble de contraintes et le système doit fournir une solution qui satisfait ces contraintes. Par exemple, le système JULIA, permet la construction de menus culinaires en fonction des invités, de leurs goûts respectifs et des ingrédients (ou catégories d'ingrédient) que l'on souhaite utiliser.

Enfin, dans le cadre du diagnostic, le problème est de trouver la cause d'un ensemble de symptômes ou de faits inhabituels. En tant que domaine d'application, nous citerons le diagnostic médical avec PROTOS dans le domaine de l'audiologie et le diagnostic.

V. 2. La tâche d'interprétation :

La tâche d'interprétation consiste, pour le système, à déduire des connaissances à partir de celles qui lui sont présentées. Les cas fournissent un contexte pour la compréhension ou l'estimation de solutions. C'est par exemple le cas de HYPO considéré comme le premier système RBC d'interprétation. HYPO a été développé dans le domaine juridique. Il a pour fonction de traiter une situation de conflit juridique donnée en entrée et de fournir un ensemble de cas pertinents pour l'argumentation du procès, aussi bien pour la défense que pour l'accusation.

Une autre approche des systèmes de RBC d'interprétation concerne le problème de la classification. Cet aspect du RBC consiste globalement en l'association d'instances et de classes particulières. Deux étapes constituent ce processus. La première consiste à générer des catégories, la seconde à classer de nouveaux cas dans ces catégories.

ROTOS propose un système de classification dont les catégories sont représentées par des exemples typiques. Pour classer un nouveau cas, le système évalue la similarité entre les exemples typiques de chaque catégorie et ce nouveau cas.

Ce système a été développé dans le domaine du diagnostic médical en audiologie et validé auprès d'un panel important de cliniciens du domaine.

VI. Modèles RBC :

Il existe plusieurs modèles pour le raisonnement à base de cas. Ces modèles sont regroupés en trois grandes familles : structurelle, conversationnelle et textuelle [19a].

VI.1. Modèle structurel :

Le modèle structurel a émergé des premières vagues applicatives de systèmes RBC. Dans ce modèle, toutes les caractéristiques importantes pour décrire un cas sont déterminées à l'avance par le concepteur du système. Ainsi, le concepteur élabore un modèle de données du domaine applicatif.

Les cas sont complètement structurés et sont représentés par des paires <attribut, valeur> (similaire à un frame ou à un objet). D'un point de vue applicatif, un attribut représente une caractéristique importante du domaine d'application. Les échelles de valeurs les plus fréquemment utilisées pour structurer les attributs sont les entiers/réels, les booléens et les symboles.

La représentation des cas peut être sur un seul niveau ou sur plusieurs niveaux (hiérarchie d'attributs).

VI.2. Modèle conversationnel :

Lorsque les caractéristiques importantes pour décrire un cas sont difficiles à déterminer à l'avance par le concepteur de système, comme le cas dans le service à la clientèle, le modèle conversationnel reste le plus adéquat pour surmonter ces difficultés.

Ce modèle conversationnel mise sur l'interaction entre l'utilisateur et le système ('conversation') pour définir progressivement le problème et sélectionner les solutions les plus appropriées.

Un cas conversationnel consiste en trois parties :

- un problème P : une brève description textuelle, de quelques lignes, de la nature du problème exprimée.
- une série de questions et de réponses : des index, exprimés sous forme de questions, permettant d'obtenir plus d'information sur la description du problème.
- une action A : une description textuelle de la solution à mettre en oeuvre pour ce problème. Cette description n'est pas structurée.

Chapitre 3 : Le raisonnement à base de cas RBC

VI. 3. Modèle textuel :

Les travaux sur le raisonnement à base de cas textuels portent sur la résolution de problème à partir d'expériences dont la description est contenue dans des documents textuels.

Les cas textuels sont de deux types :

- Non structurés : la description est complètement en "free-text".
- Semi-structurés : le texte est découpé en plusieurs portions étiquetées par des descripteurs tels que "problème", "solution".

Un cas textuel non structuré est un cas dont le seul attribut est textuel.

Un cas textuel semi-structuré est un cas dont un sous-ensemble de ses attributs est textuel.

Cette représentation peut être une finalité: par exemple, obtenir le texte d'un jugement légal servant de jurisprudence à une nouvelle cause ou du diagnostic médical.

VII. Le cycle du RBC :

Le but fondamental des systèmes RBC est d'éviter de reproduire les erreurs passées, et de faciliter l'acquisition des connaissances. Lorsqu'un nouveau problème est à résoudre, il est intégré dans un cas cible où la partie solution est inconnue et doit être apportée par le raisonnement. Les cas sources représentent des expériences passées qui sont stockées dans une mémoire. Les cas sources et les cas cibles partagent le même formalisme de représentation. L'objectif du raisonnement est de transférer des enseignements pertinents des cas sources pour élaborer la solution du cas cible. Ce raisonnement se décompose habituellement en quatre phases principales qui peuvent être totalement ou partiellement automatisées [2]:

1. phase de **recherche** dont le but est de sélectionner un ou plusieurs cas utiles pour la suite du raisonnement, tout en prenant en compte les indices du cas cible déterminés par une étape d'élaboration ;
2. phase de **réutilisation** permettant de construire une solution au problème courant en se basant sur les cas identifiés dans la phase précédente ;
3. phase de **révision** de la solution qui permet d'affiner la solution grâce à son évaluation
4. phase d'**apprentissage** chargée de mettre à jour les éléments du raisonnement en prenant en compte l'expérience qui vient d'être réalisée, et qui pourra ainsi être utilisée dans les raisonnements futurs.

Chapitre 3 : Le raisonnement à base de cas RBC

Le schéma ci dessous résume clairement le cycle d'un RBC :

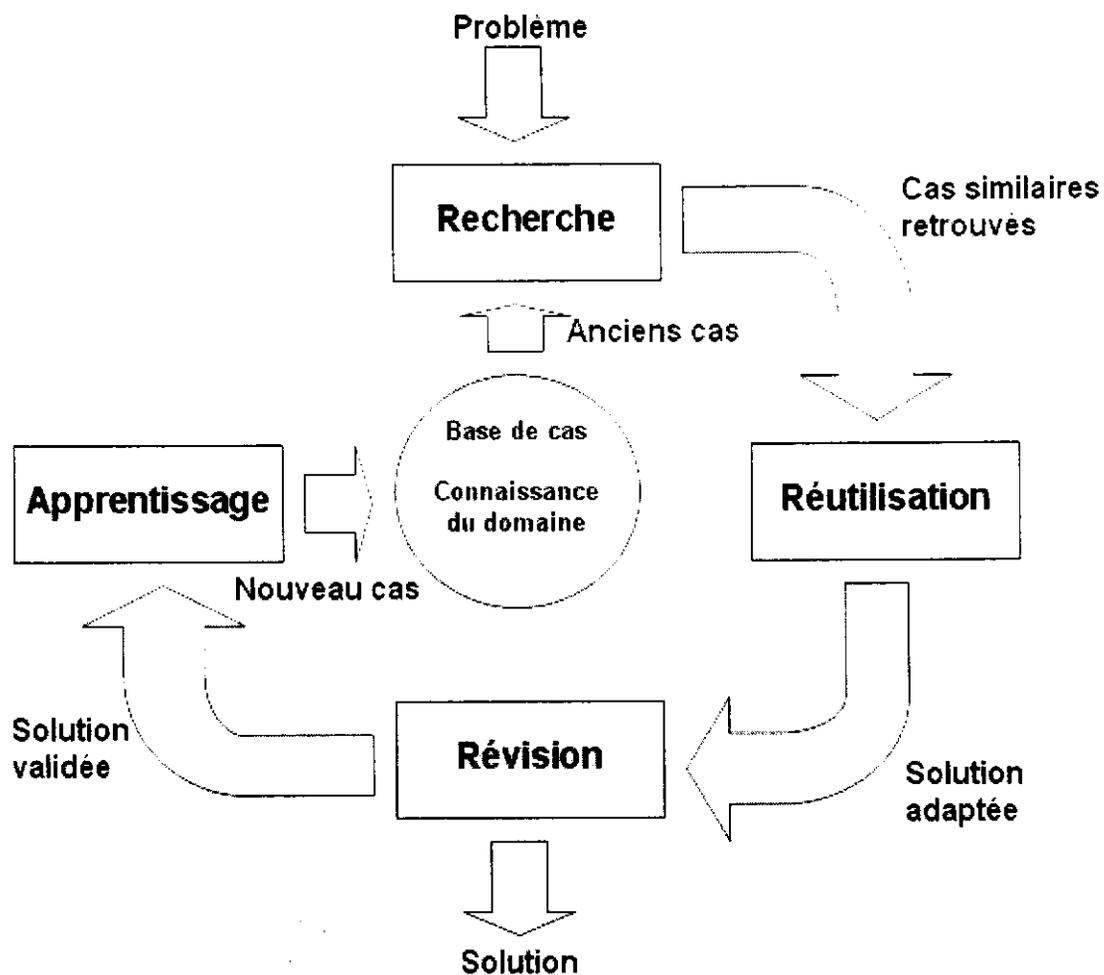


Figure 3.2: Cycle d'un RBC

VII.1. Remémoration :

L'étape de remémoration consiste à chercher dans la base de cas un ou plusieurs cas passés résolus jugés similaires au cas cible. La sélection des cas similaires s'appuie sur une mesure de similarité. Certains systèmes conservent plusieurs cas remémorés pour les combiner par la suite, mais la plupart du temps, un seul cas est conservé pour la suite du processus. On l'appelle le cas source. La sélection du cas source parmi les cas remémorés peut être la conséquence d'une sélection par le système ou bien résulter d'un choix effectué par l'utilisateur. La remémoration est composée de deux sous tâches qui sont représentées dans la Figure 3.3 :

Chapitre 3 : Le raisonnement à base de cas RBC

- La tâche de recherche initiale qui consiste à trouver un ensemble de cas qui sont suffisamment similaires au nouveau cas.
- La sélection : parmi l'ensemble des cas similaires trouvés pendant la tâche précédente, le meilleur est choisi.
-

Remarque :

Pour permettre de bien comprendre les figures suivantes, il y a lieu de se référer à la définition des notions suivantes :

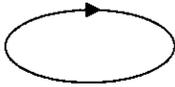
	Indique que la tâche A doit être réalisée avant la tâche B.
	Indique que la tâche principale peut être réalisée par l'une ou l'autre des sous tâches fils.
	Indique que la tâche est répétitive.
L'absence annotation	Indique un ordre non défini (en séquence ou en parallèle).

Tableau 3.1 : Tableau de notations des graphes

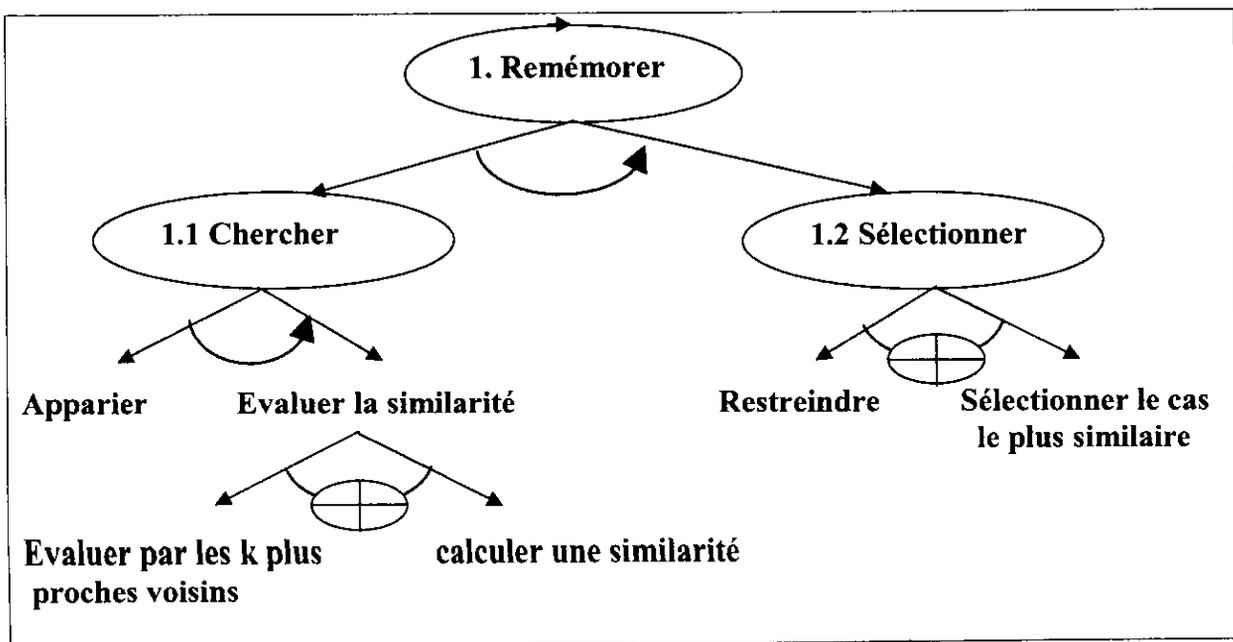


Figure 3.3 : La décomposition de la tâche « Remémorer »

VII.2. Réutilisation :

La réutilisation ou l'adaptation d'un cas est le processus permettant l'application de la solution trouvée dans l'étape de remémoration pour correspondre au problème courant.

La réutilisation est composée de deux sous tâches qui sont représentées dans la **Figure 3.4** :

- **Copier** : la réutilisation commence par copier soit la solution, soit la méthode d'obtention de la solution du cas remémoré, puis procède à l'adaptation.
- **Adapter** : Les systèmes distinguent généralement l'adaptation par transformation et l'adaptation par dérivation.
 - **L'adaptation par transformation** : dans ce cas, c'est la solution du cas elle même qui est réutilisée. La solution trouvée n'est pas une solution directe pour le nouveau problème, mais elle peut être transformée pour être appliquée à la résolution de ce problème.
 - **L'adaptation par dérivation** : dans ce cas, c'est la trace du raisonnement ayant produit la solution du cas qui est réutilisée. L'adaptation par dérivation re-crée complètement la solution pour le nouveau problème en ré appliquant, sur les données du nouveau problème, le raisonnement qui avait produit la solution du cas passée.

Les sous tâches de la tâche d'adaptation sont les suivantes :

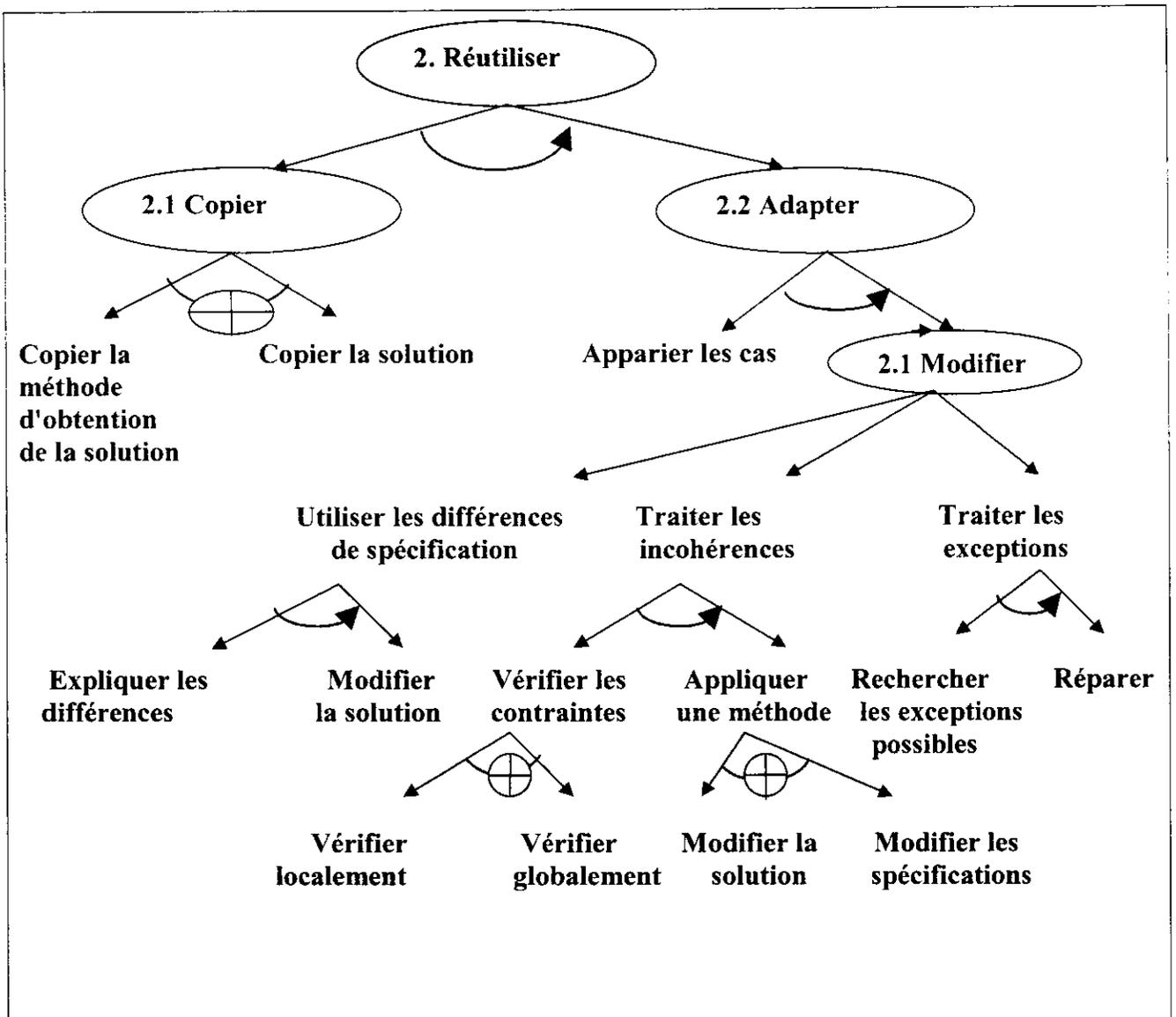


Figure 3.4 : La décomposition de la tâche « Réutiliser »

VII.3. Révision :

La révision est souvent considérée comme la continuation de l'adaptation. Elle a pour but l'évaluation de la solution produite par l'étape précédente. Si cette évaluation est réussie, on passe à l'étape suivante qui l'étape d'apprentissage pour maintenir la nouvelle expérience, sinon réparer la solution. L'étape de révision permet d'identifier les causes éventuelles des échecs et de proposer des adaptations supplémentaires pour conduire à une solution satisfaisante : il s'agit du cas révisé.

Chapitre 3 : Le raisonnement à base de cas RBC

D'une manière générale l'étape révision comporte trois sous tâches qui sont représentées dans la Figure 3.5 :

- La première sous tâche est la mise en oeuvre de la solution.
- La deuxième sous tâche est la réparation de la solution. La solution peut être réparée par un utilisateur ou par un processus extérieur.
- La troisième sous tâche est l'explication des différences entre la solution avant et après sa correction. Les explications élaborées servent de base à la tâche d'apprentissage. Les explications peuvent être élaborées par un raisonnement en s'appuyant sur les connaissances disponibles ou sur un modèle causal spécifique. Lorsque les connaissances sont insuffisantes pour générer les explications, l'utilisateur peut alors être sollicité pour fournir les éléments nécessaires.

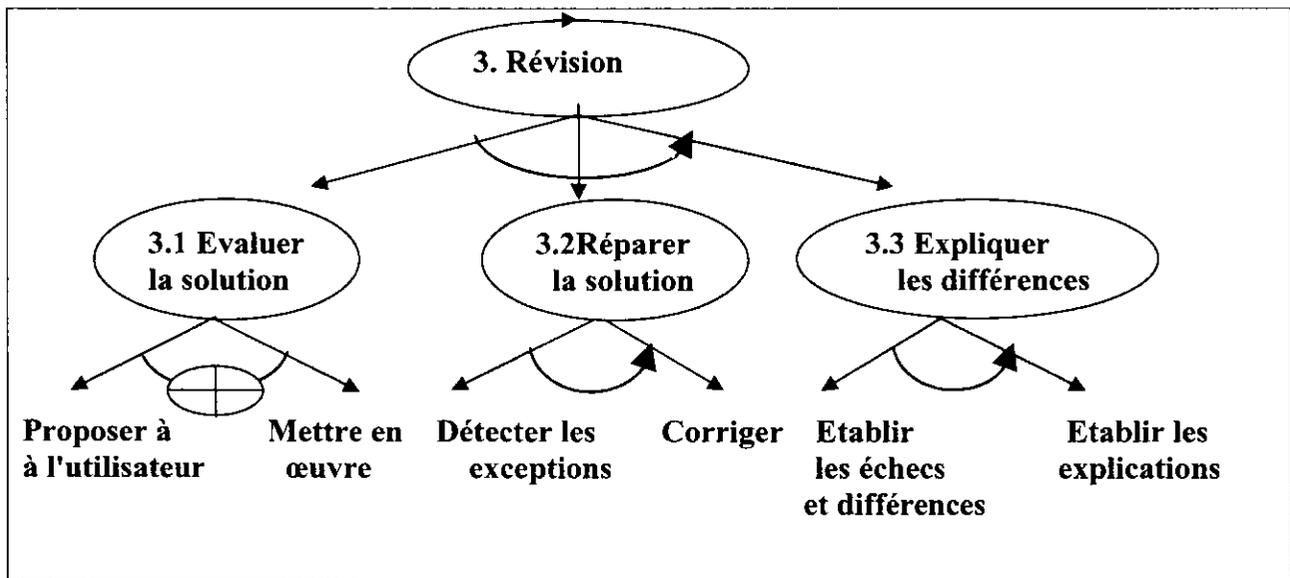


Figure 3.5 : La décomposition de la tâche « Révision »

VII. 4. Mémorisation :

On considère l'étape de mémorisation comme l'étape durant laquelle la base de cas est enrichie par le cas cible révisé. Cette mémorisation implique une mise à jour des index permettant de retrouver les cas et parfois un processus de maintenance pour réorganiser la base de cas. L'étape de mémorisation est également le siège de l'apprentissage d'autres types de connaissances, c'est durant cette étape que l'on peut concrétiser les efforts effectués durant les autres étapes pour apprendre d'autres connaissances.

La mémorisation est composée de trois sous tâches qui sont représentées dans la **Figure 3.6** :

- La première sous tâche de la mémorisation est la tâche d'apprentissage.
- La deuxième sous tâche évalue l'utilité du cas afin de déterminer s'il doit être conservé ou non dans la base des cas :
- Si le cas est jugé utile, on stocke le nouveau cas dans la base des cas avec toutes les données représentant ce problème pour les réutiliser dans la résolution d'un nouveau problème.
- Si le cas est jugé inutile c'est-à-dire une expérience qui à échoué, on stocke également le problème sous forme de cas avec les raisons de l'échec, pour éviter de refaire la même erreur dans le future.

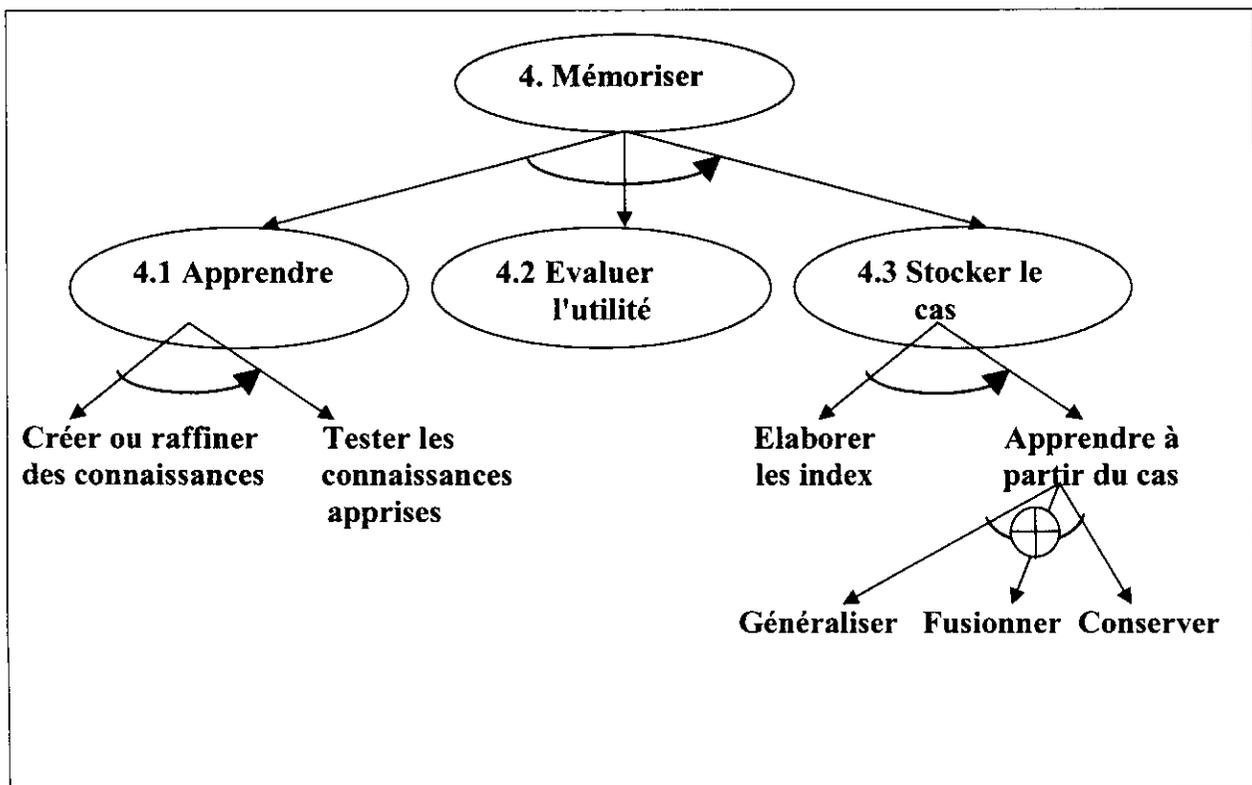


Figure 3.6 : La décomposition de la tâche « Mémoriser »

VIII. Notion de similarité :

La phase de recherche de cas permet de déterminer les cas de la base qui sont les plus similaires au problème à résoudre. Ce paragraphe explicite le principe d'une des méthodes de calcul de similarité : La méthode du plus proche voisin (ou des k plus proches voisins)

La mesure de similarité par la méthode du plus proche voisin :

La méthode des plus proches voisins consiste à rechercher les plus proches voisins du nouveau cas avec une fonction de similarité.

Le choix de la fonction de similarité est crucial. Il aura une influence directe sur le temps de réponse du RBC. Pour définir la fonction similarité, on doit définir d'abord une similarité locale sur chacun des champs des enregistrements, puis on combine ces similarités pour définir la similarité globale entre enregistrements.

L'algorithme permet de sélectionner un cas qui a la plus grande mesure de similitude. A titre d'exemple, la **figure 3.1** présente un cas à analyser ayant deux attributs dont les valeurs sont x_1 et x_2 . Le cas **A** et le cas **B** sont des cas contenus dans la base. Les valeurs de leurs attributs sont y_1 , y_2 et z_1 , z_2 . Pour le cas à analyser et le cas **A**, l'algorithme permet de calculer une valeur de comparaison entre les attributs x_1 et y_1 d'abord ; puis entre x_2 et y_2 .

Il fait la somme de toutes les valeurs de comparaison pour trouver une mesure de similitude. Le même calcul est refait pour le cas **B**. Si la mesure de similarité trouvée entre le cas **A** et le cas à analyser est supérieur à la mesure trouvée pour le cas **B**, alors l'algorithme sélectionne le cas **A** comme étant le plus similaire.

L'utilisation de cet algorithme sans affecter des poids aux différents attributs laisse supposer que ces attributs ont la même importance, or ceci n'est pas toujours vrai dans la réalité et en particulier dans notre cas. Il est donc nécessaire d'attribuer des poids aux attributs selon leurs importances.

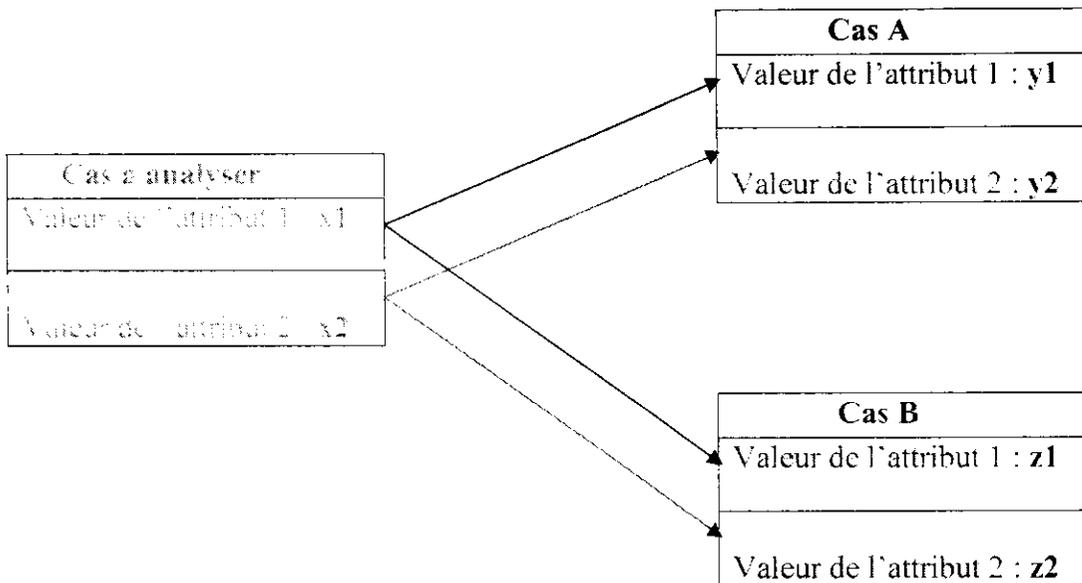


Figure 3.7 : Méthode de calcul du plus proche voisin

La mesure de similarité peut être donc calculée selon les formules suivantes :

a) Mesures de similarités locales :

▪ Similarité symbolique :

Pour les attributs symboliques, les mesures peuvent être calculées en utilisant l'équation (1) pour les valeurs nominales et l'équation (2) pour les valeurs ordinales ; valeurs ordonnées selon un échelle.

$$\text{sim}(x_i, y_i) = \begin{cases} 1 & \text{si } x_i = y_i \\ 0 & \text{si } x_i \neq y_i \end{cases} \quad (1)$$

$$\text{sim}(x_i, y_i) = \begin{cases} 1 & \text{si } x_i = y_i \\ 1 - (d / N) & \text{si } x_i \neq y_i \end{cases} \quad (2)$$

d représente le nombre de valeurs symbolique utilisées pour cet attribut particulier.
N représente le nombre de valeurs séparant les valeurs **x** et **y**.

Chapitre 3 : Le raisonnement à base de cas RBC

Exemple :

- Soit un ensemble de valeurs $V = \{\text{Rouge, vert, Bleu, Noir}\}$ et soient x et y ; deux valeurs spécifiques, avec $x_i = \text{vert}$ et $y_i = \text{Noir}$.

La mesure de similarité $\text{sim}(x, y) = 0$ selon l'équation (1).

- Soit un ensemble de valeurs $S = \{\text{Très faible, faible, Haut, Très haut}\}$ (c'est-à-dire $N = 4$) et soient x et y ; deux valeurs spécifiques, avec $x = \text{faible}$ et $y = \text{Haut}$.

La mesure de similarité $\text{sim}(x, y) = 1 - (1/N) = 1 - 0.25 = 0.75$ selon l'équation (2).

▪ Similarité numérique :

Pour les attributs numériques, la similitude entre deux attributs peut être calculée en utilisant la métrique de Canberra.

$$\text{sim}(x_i, y_i) = 1 - [(|x_i - y_i|) / (|x_i| + |y_i|)] \quad (3)$$

Le poids w_i reflète l'importance relative de chaque attribut.

b) Mesures de similarités globales :

Dans le RBC le calcul des similarités globales est fait par à partir des similarités locales.

$$\text{sim}(x, y) = \left[\sum_{i=1}^n w_i \text{sim}(x_i, y_i) \right] / \sum_{i=1}^n w_i \quad (4)$$

Le poids w_i reflète l'importance relative de chaque attribut.

Chapitre 3 : Le raisonnement à base de cas RBC

Conclusion :

De manière générale, le raisonnement à base de cas est une technique de résolution de problèmes basée sur la réutilisation par analogie d'expériences passées appelées *cas*.

Dans ce chapitre nous avons défini le raisonnement à base de cas, donné son cycle et expliqué en quoi consiste chacune de ces étapes.

Nous avons pu démontrer clairement que la mesure de similitude est une étape cruciale dans le RBC. Parmi les mesures qui existent et que nous avons détaillées, on note : La méthode du plus proche voisin (ou des k plus proches voisins) [25]

Il apparaît clairement que l'usage de raisonnement à base de cas est approprié dans les domaines où les données et solutions des problèmes précédents existent, les cas précédents sont considérés comme des acquis à préserver et le souvenir des expériences précédentes est utile.

Introduction :

En médecine, peut-être plus que dans d'autres disciplines, les observations sur lesquelles se basent les décisions médicales sont très imparfaites. Elles peuvent être floues, ambiguës, incomplètes, et incertaines. Le processus décisionnel, qu'il soit diagnostique, thérapeutique ou pronostique est donc un processus sous incertitude.

Le problème qui nous intéresse ici est celui de la gestion de l'imprécision et de l'incertitude sur des informations qui peuvent être numériques ou symboliques. Nous avons choisi les travaux de la logique floue et de la théorie des possibilités qui permettent de représenter ces informations, les manipuler, les combiner et les propager dans des raisonnements où les règles de la logique classique ne sont plus applicables.

Nous avons basé notre réflexion sur la notion d'expérience, pour cela, Après une étude des principales méthodes de diagnostic médical, notre choix s'est tourné vers le Raisonnement à Base de Cas Flou (RBCF) pour résoudre notre problématique. Le RBCF est un concept puissant qui fournit un mode de raisonnement analogique pour résoudre des problèmes. Cette capacité permet au RBCF d'exprimer la connaissance liée à l'expérience médicale et ainsi de l'utiliser pour améliorer le diagnostic, le pronostic et la thérapeutique. Le RBCF est fondé sur la comparaison des nouveaux cas avec des cas cliniques indexés et précédemment stockés. Il s'agit ensuite de rechercher ceux qui sont similaires et d'appliquer les décisions et les actions correspondantes au patient en espérant que « ce qui sera efficace une fois le sera plusieurs fois».

La recherche d'un cas similaire est réalisée à l'aide de l'algorithme de plus proche voisin flou (noté kppv floue) [2] [4b] qui permet de sélectionner parmi les cas passés les k plus proches cas du cas courant.

Nous présenterons d'abord le raisonnement en logique floue. Ensuite, nous expliquerons l'utilisation de la technique du raisonnement basé sur les cas (RBCF) dans le cadre de notre système de diagnostic médical.

I. Traitement de données imprécises et incertaines :

La théorie des sous-ensembles flous a été introduite pour répondre aux besoins de la représentation des connaissances exprimées symboliquement, en langage naturel, soumises à des imprécisions ou présentant un caractère vague.

Les informations imprécises permettant de décrire un cas sont également prises en compte. Pour cela, des distributions de possibilité sont utilisées. Lorsque le domaine de définition de l'attribut est symbolique, la distribution de possibilité est représentée par un ensemble de couples (symbole, degré). Le degré, compris dans l'intervalle $[0,1]$, est utilisé pour décrire l'imprécision de la caractéristique (à chaque symbole est associé un degré). Par exemple, une valeur imprécise de la caractéristique «**Vomissement**» peut être représentée à l'aide de l'ensemble suivant : {(fréquent 1) (rare 0.5) (inexistante 0)}.

Concernant les attributs numériques, les distributions de possibilité sont représentées à l'aide d'un ensemble trapézoïdal. Celui-ci permet de définir un ensemble de valeurs complètement possibles (degré de possibilité à 1) et, « autour » de ces valeurs complètement possibles, des valeurs plus ou moins possibles (degrés de possibilité décroissant de 1 à 0).

Comme exemple pour les fonctions d'appartenance, on présente l'exemple de la température du patient. Dans le cas le plus simple, l'attribut "température" peut avoir les étiquettes linguistiques «**forte fièvre**», «**modérée**», et «**inexistante**», définies sur un ensemble entre 36°C et 42°C. Elles forment trois ensembles flous. Nous utiliserons la représentation trapézoïdale comme celle représentée dans la **Figure 4.1**.

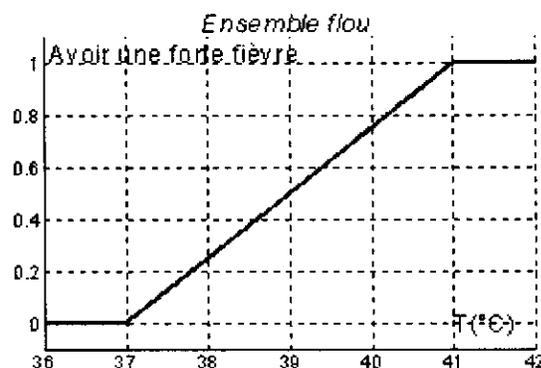


Figure 4.1 : Ensemble flou représentant la valeur « forte » pour la attribut « fièvre »

II. La logique floue et le raisonnement à base de cas dans le cadre du diagnostic médical :

Le type de raisonnement que nous avons choisi est inspiré du raisonnement à base de cas flou (RBCF) est un type de RBC qui emploie des méthodes floues pour représenter et comparer des cas, et pour former des solutions.

La logique floue est particulièrement utile pour le RBC parce que : le RBC est fondamentalement un raisonnement analogique, le raisonnement analogique peut fonctionner avec des expressions linguistiques, et la logique floue est conçue pour fonctionner avec des expressions linguistiques tandis que la logique classique fonctionne avec des variables booléennes.

Notre problématique concerne l'aide au diagnostic médical. Nous avons choisi de le valider pour le cas de la grippe, vu que celle-ci est connue de tout le monde. L'approche par raisonnement à partir de cas a été choisie pour les avantages qu'elle présentait.

II.1. Représentation du Cas :

Par définition, un cas est une connaissance particulière représentant un épisode de résolution de problème composé de deux parties : le problème et sa solution. Dans notre système, un cas est constitué de la description de l'examen associé à une consultation pour un patient (le problème) et de la description de la prise en charge (la solution).

II.1.1. Description de problème :

La démarche que nous avons suivi pour définir le problème dans un cas est la suivante. Nous avons demandé aux médecins de choisir parmi l'ensemble des paramètres d'une consultation celles qu'ils utilisent pour définir la prise en charge d'un patient grippé. Nous avons obtenu vingt variables jugées utile à la définition de la prise en charge de la grippe. Comme indiqué précédemment, la nature de ces données est hétérogène : certaines variables sont numériques et d'autres symboliques.

Fièvre, asthénie (fatigue), frissons, douleurs musculaires, douleurs articulaires, douleurs digestives, courbatures, céphalées (maux de tête), maux de gorge, toux, nez bouche, éternuements, écoulements nasaux, douleur de la poitrine, conjonctivite, des sueurs nocturnes, somnolence, bronchite, diarrhée, vomissements.

Ces attributs sont définis par des étiquettes linguistiques, L'étiquette linguistique permet de symboliser une valeur imprécise d'un attribut flou par une distribution de possibilités sur un intervalle réel $[0,1]$ (comme présente le **Tableau 5.1**).

Chapitre4 : La logique floue et le raisonnement à base de cas pour le diagnostic médical

On rajoute les attributs suivants :

Age du malade, *Poids du malade* et *Sexe du malade* puisque les médicaments sont donnés en fonction de l'âge et surtout du poids du malade, le *Nom du médecin* qui a créé le cas et *Date de consultation*.

Ces vingt-cinq variables représentent la partie « problème » de notre cas.

II.1.2. Description de la solution :

La prise en charge comprend la proposition du diagnostic, et le choix d'un traitement adapté.

Le diagnostic pour la pathologie de la grippe est défini par la maladie et le traitement.

Ces deux variables (maladie et le traitement) représentent la « solution » pour notre cas. elles sont de nature symbolique.

Dans notre problématique, un cas est par conséquent un vecteur composé de vingt-huit attribut- valeurs : vingt-six attributs sont dédiés à la description du problème et deux dédié à la solution.

Chapitre4 : La logique floue et le raisonnement à base de cas pour le diagnostic médical

Description du Cas :

Attribut	Nature
Numéro du cas	Numérique
Age du malade	Numérique
Sexe du malade	Alphabétique
Poids du malade	Numérique
Nom du médecin	Alphabétique
Date de consultation	Date
Fièvre	Numérique
Asthénie (fatigue)	Symbolique
Frissons	Symbolique
Douleurs musculaires	Symbolique
Douleurs articulaires	Symbolique
Douleurs digestives	Symbolique
Courbatures	Symbolique
Céphalées (maux de tête)	Symbolique
Maux de gorge	Symbolique
Toux	Symbolique
Nez bouche	Symbolique
Éternuements	Symbolique
Écoulements nasaux	Symbolique
Conjonctivite	Symbolique
Des sueurs nocturnes	Symbolique
Douleur de la poitrine	Symbolique
Somnolence	Symbolique
Bronchite	Symbolique
Diarrhée	Symbolique
Vomissements	Symbolique

Attribut	Nature
Maladie	Symbolique
Traitement médicament	Symbolique

Problème

Solution

II.2. Le cycle du RBCF dans le cadre de diagnostic médical :

Le raisonnement à base de cas flou dans le cadre du diagnostic médical est établi en quatre étapes.

Premièrement, l'étape de la remémoration (la phase de la recherche) qui a pour but de retrouver dans la base de cas le ou les cas les plus similaires. Elle est généralement basée sur une mesure de similarité floue. Deuxièmement, l'étape du choix de solution (adaptation) : qui comme son nom l'indique adapte la solution retrouvée au problème courant. Troisièmement, l'étape de révision : permet d'évaluer/corriger la solution au problème courant. Dans notre contexte la phase de révision est effectuée par les médecins. Enfin, l'étape de mémoration : il s'agit d'ajouter le nouveau cas à la mémoire. Dans notre contexte, la base de cas n'est pas enrichie automatiquement, mais par les experts. Ces différentes phases sont détaillées dans les paragraphes suivants.

La figure suivante illustre le cycle de RBCF dans le cadre de diagnostic médical :

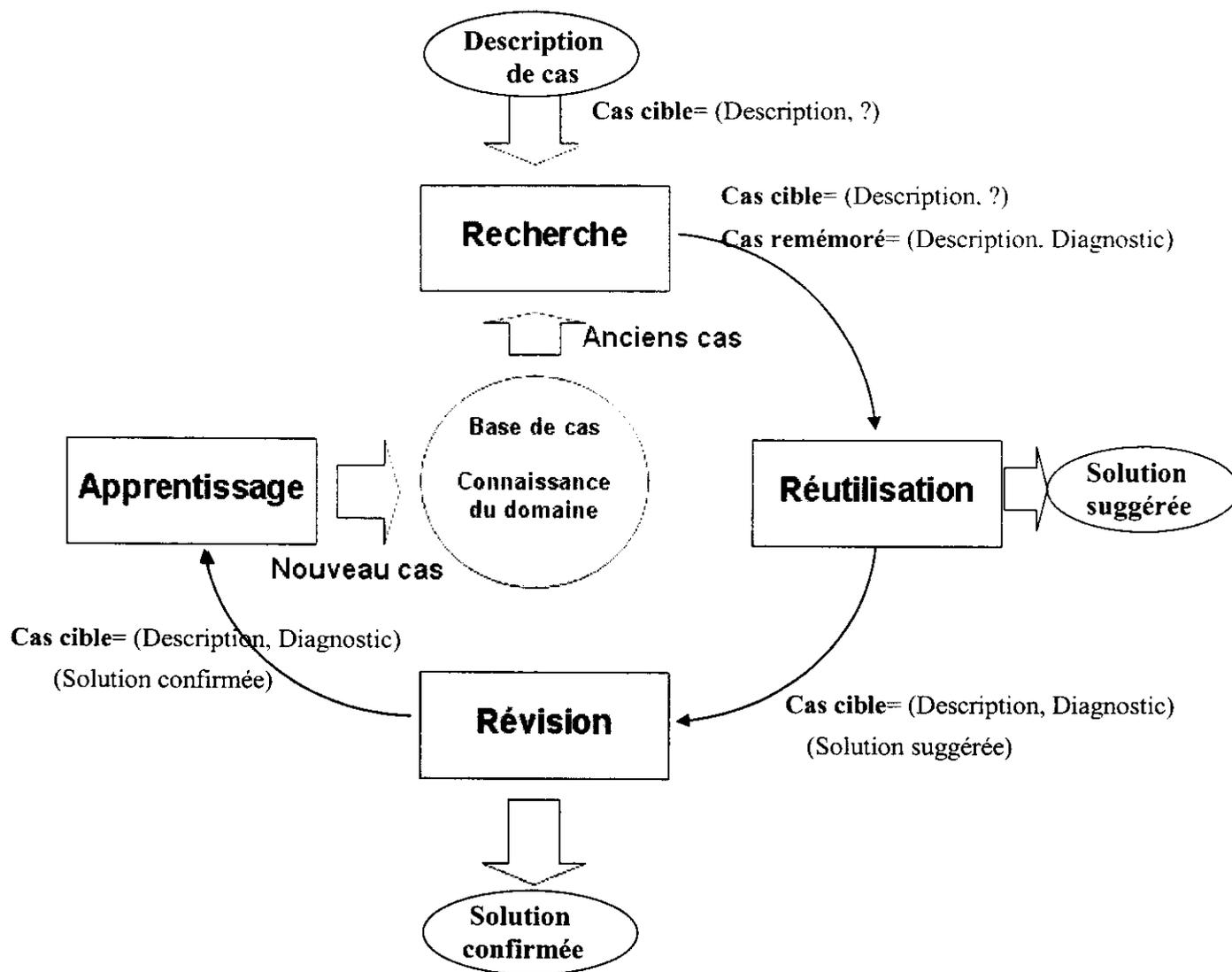


Figure 4.2 : Le cycle de raisonnement à base de cas flou dans le cadre de diagnostic médical

Nous détaillons dans le paragraphe suivant chacune des étapes du cycle du raisonnement à base de cas flou citées précédemment.

1. La phase de remémoration :

L'étape de remémoration permet d'identifier les cas sources les plus similaires au cas cible. Pour un cas cible donné la remémoration permet de trouver un ensemble de cas source similaires au cas cible.

Chapitre4 : La logique floue et le raisonnement à base de cas pour le diagnostic médical

Nous avons focalisé dans cette étape sur la façon de prendre en compte l'imprécision pour décrire la similarité entre valeurs. Des fonctions de similarité définies par les experts sont caractérisées à l'aide de notion graduelles de similarité (valeurs *très* proches, valeurs *presque* proches, etc.). Ces fonctions de similarité sont exploitées lors de la phase de recherche de cas passés. Nous détaillons ici cette étape.

Mesure de similarité :

Les limites des modèles de similarité présentés précédemment (chapitre 3) sont nombreuses. Si l'on recherche les cas similaires avec le modèle de similarité relative par exemple, nous allons devoir fixer un seuil à partir duquel les cas vont être considérés comme intéressants. Si l'on prend 0.8 comme exemple, un cas avec un calcul de similarité de 0.81 sera considéré comme intéressant et un cas de 0.79 comme ne l'étant pas, et ce alors que le calcul n'est qu'une estimation pouvant difficilement approcher ce que l'on considère comme similaire dans tous les cas. On se rend compte alors aisément que l'application de ce modèle ne correspond plus du tout à l'interprétation humaine de la similarité entre cas.

Pour pallier à ce problème, il nous faut introduire la logique floue dans le calcul de la similarité entre deux cas. Ainsi il sera possible de définir des états comme « assez similaire » « peu similaire » ou « très similaire » sans qu'un seuil binaire soit fixé entre ce qui est considéré comme faisant partie d'un état et ce qui n'en fait pas partie, ce qui permet de se rapprocher de la conception d'appartenances telle qu'elle est considérée par l'être humain.

Notre recherche de cas se fait à l'aide de la méthode des k plus proches voisins floue (kppv floue) [4b] [6]. Une technique de K plus proches voisins floue est simplement une technique de K plus proches voisins dans laquelle la technique de base de mesure est floue.

Le principe implicite de la méthode kppv floue est exprimé par : « plus les attributs de description de problème sont plus semblables, plus les attributs de résultats sont plus semblables »

L'utilisation de techniques classiques de k plus proches voisins a pour but, lorsqu'un nouveau cas est présenté, de trouver à quel prototype de cas il appartient, ces prototypes étant définis par avance. L'utilisation d'un algorithme flou de kppv permet de définir des degrés d'appartenance d'un nouveau cas à chacun des prototypes au lieu d'avoir une appartenance ou une non-appartenance d'un cas à un prototype. L'élément principal de la technique des k plus proches voisins est la mesure de similarité ou distance de similarité. La mesure de similarité mise en place pour la recherche des plus proches voisins prend en entrée deux cas et donne en sortie une valeur de similarité comprise entre 0 et 1.

Chapitre4 : La logique floue et le raisonnement à base de cas pour le diagnostic médical

Généralement, la méthode k plus proches voisins floue attribue le même poids pour tous les attributs (c'est-à-dire $w_i=1$). L'utilisation de cette méthode sans affecter des poids aux différents attributs laisse supposer que ces attributs ont la même importance.

Dans notre cas, l'attribution d'un même coefficient pour tous les attributs fausserait les résultats, vu que certains attributs sont plus importants que d'autres.

Ce travail de pondération n'est pas facile à réaliser, puisqu'il nécessite une grande expérience dans le domaine. Nous avons demandé à des médecins de nous aider à faire le choix des coefficients de pondération. Ces coefficients sont choisis selon le degré d'importance des attributs. Ils sont présents dans le **tableau 4.1**.

Attribut	Coefficient de pondération
Fièvre	5
Asthénie (fatigue)	5
Frissons	2
Douleurs musculaires	2
Douleurs articulaires	2
Courbatures	3
Céphalées (maux de tête)	3
Toux	3
Maux de gorge	4
Nez bouche	1
Éternuements	1
Ecoulements nasaux	3
Douleur de la poitrine	1
Conjonctivite	2
Des sueurs nocturnes	1
Somnolence	1
Une bronchite	1
diarrhée	1
Vomissements	1
Poids	2
Age	2

Tableau 4.1 : Présentation de la table des poids

Notre système va effectuer dans un premier temps un calcul de similarité locale c'est-à-dire, comparer chaque attribut i du cas cible avec l'attribut i de chaque cas de la mémoire de cas. Puis dans un deuxième temps il mesurera la ressemblance globale grâce à la fonction mathématique (celle de $kppv$) que nous avons cité précédemment.

Dans notre cas, nous avons constaté quelle contient plusieurs types d'attributs qu'il est important de différencier, car ceci influe sur le calcul de similitude entre les cas.

Chapitre4 : La logique floue et le raisonnement à base de cas pour le diagnostic médical

- Des attributs de type Numérique tel que âge, poids du patient,...

Nous rappelons que pour ce type d'attributs le calcul de similarité s'effectue en utilisant l'équation mathématique suivante :

$$\text{Sim}(x_i, y_i) = 1 - (|x_i - y_i|) / (|x_i| + |y_i|)$$

Où x_i et y_i sont les attributs qui caractérisent deux cas A et B.

- Pour les attributs flous (des attributs définis sur des domaines discrets par des expressions linguistiques) tel que Courbatures, Céphalées, Toux, Vomissements,

La mesure de similarité est décrite à l'aide d'ensembles flous, pour comparer ces attributs, des degrés de similarités peuvent être spécifiés à l'aide de tables (Figure 4.3) [6].

Comme exemple on prend l'attribut «Vomissements ».

	fréquent	rare	inexistante
fréquent	1	0.50	0.05
rare	0.50	1	0.10
inexistante	0.05	0.10	1

Figure 4.3 : Degré de similarité pour des attributs nominaux.

2. La phase de choix de la solution :

Les différents cas seront donc classés selon leur indice de similarité, de plus chaque cas sera affiché avec une rubrique qui indiquera la valeur de cette mesure de similarité.

L'utilisateur aura toutes les données pour une prise de décision efficace.

- Soit le cas étudié coïncide avec un cas de la base de cas, dans ce cas la solution à adopter sera appliquée intégralement.
- Soit le cas similaire n'existe pas dans la base, l'expert peut ainsi s'inspirer des solutions des cas qui se rapprochent du cas étudié. Il peut donc proposer une solution en se basant sur les cas les plus proches.

3. La phase de révision :

La révision consiste à continuer éventuellement l'élaboration de la solution cible en cas de besoin. La révision comporte deux phases : l'évaluation et la réparation de la solution.

L'évaluation détermine si la solution est correcte. Cela peut être fait par le médecin. Si l'évaluation est réussie, on passe à la phase d'apprentissage. Si ce n'est pas le cas, il faut réparer la solution, c'est-à-dire détecter les erreurs, les expliquer puis modifier la solution de façon à ne plus commettre ces erreurs.

Une étape de révision automatique ou semi-automatique n'est pas toujours présente dans les systèmes de RBC. C'est le cas de notre système où la phase de révision est effectuée par le médecin.

4. La phase de mémoration :

Cette phase est peut être la phase la plus sensible du cycle de raisonnement à base de cas. En fait, la cohérence de la base de cas ne peut être maintenue que si cette phase est menée correctement. Les experts en domaine médical, qui sont les seuls concernés par la mémoration des cas, doivent donc avoir conscience de la délicatesse des opérations de mise à jour de la base.

Un nouveau cas sera ajouté à la base que si après calcul de similarité entre ce cas et ceux de la base, le système détermine qu'il n'existe pas de cas identique à celui-ci.



Chapitre4 : La logique floue et le raisonnement à base de cas pour le diagnostic médical

Conclusion

L'utilisation des travaux de la logique floue et des ensembles flous permet de caractériser les connaissances imprécises et incertaines contenues dans les cas. Au-delà de cette caractérisation, ces travaux permettent d'exprimer des degrés de similarité entre valeurs. L'imprécision ne concerne plus alors directement les valeurs des cas mais la façon de les manipuler.

Nous avons présenté dans ce chapitre nos travaux concernant la mise au point d'un système de diagnostic médicale pour la prise en charge de la grippe. Le Raisonnement à Base de Cas Floue est à priori l'approche la plus séduisante pour notre problématique. Il a été montré que la représentation des cas est une tâche particulièrement difficile dans le domaine médicale notamment à cause de la qualité des données et de la complexité de la tâche de prise de décision médicale. Nos données médicales sont de mauvaise qualité à cause des valeurs imprécises, manquantes, etc. Une mesure de similarité floue basée sur la méthode kppv s'avère la mieux adaptée pour la sélection du meilleur cas.

Le chapitre suivant sera réservé à la présentation des étapes les plus importantes dans la conception et la réalisation de notre système de diagnostic médical en utilisant le langage de modélisation UML.

Chapitre5 : Démarche de développement

Introduction :

Afin de mener à bien un projet informatique, une bonne assise doit être fondée pour garantir une longévité en terme de qualité et de rentabilité. Cette assise n'est au fait rien d'autre que la conception, car une erreur de conception peut avoir des répercussions néfastes voir fatales sur tout le projet qui risquera d'être ruiné en conséquence.

Il existe plusieurs approches de conception, parmi eux figure la conception orientée objet que nous avons adoptée. Le choix de cette approche trouve origine dans le fait qu'elle est caractérisée par la stabilité de la modélisation par rapport au monde réel.

Pour présenter et modéliser notre système qui se base sur la représentation des données dite imprécises ou floues et le raisonnement à base de cas floue, depuis la phase de l'analyse jusqu'à l'implémentation, on a choisi d'utiliser le langage de modélisation UML (Unified Modeling Language).

Chapitre 5 : Démarche de développement

I. La démarche de développement :

Pour développer un logiciel, il est nécessaire de passer par plusieurs étapes. Pour cela des méthodes de développement ont été définies. Ces méthodes permettent de mieux organiser, de réduire la complexité des applications, et permettent une plus grande facilité dans l'interprétation des concepts logiciels, pour modélisation de notre travail, nous utilisons la notion UML.

La démarche de développement que nous utilisons est le modèle en « cascade ». Le modèle en cascade permet de couvrir le cycle de vie d'un logiciel depuis l'analyse jusqu'au test et maintenance (voir la figure 3.1).

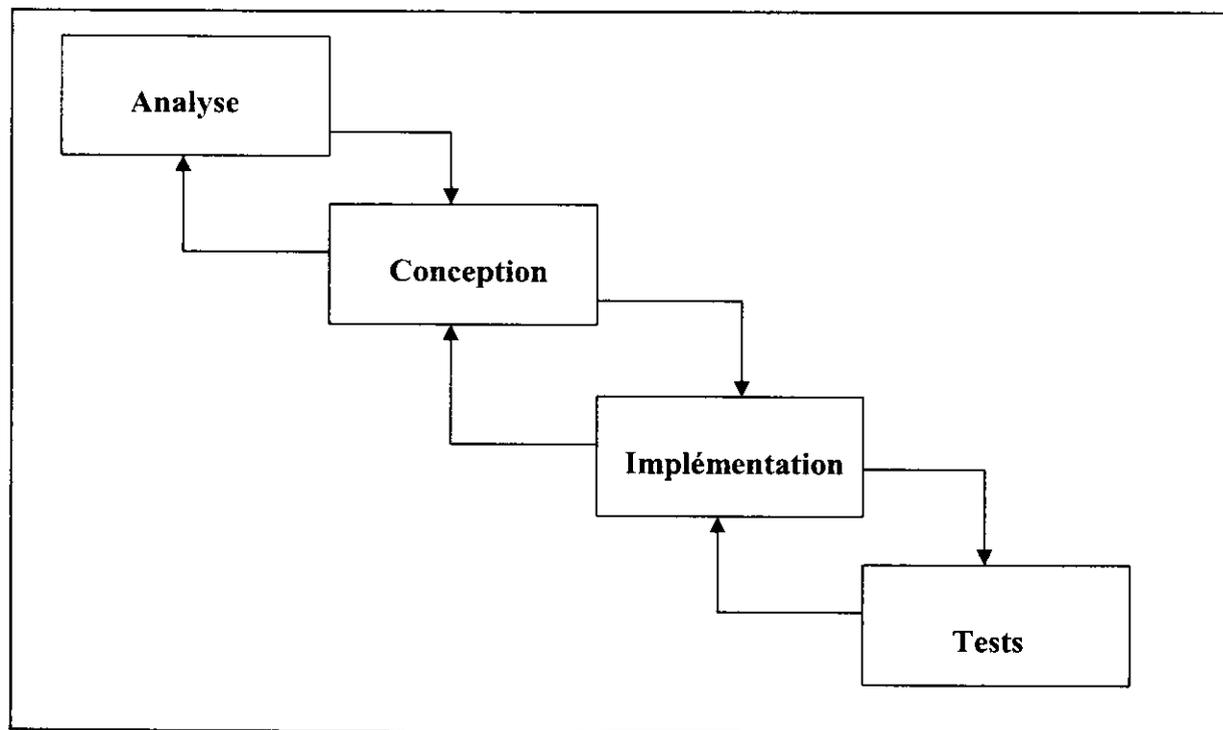


Figure 5.1 : Le cycle de vie en cascade

Chapitre 5 : Démarche de développement

II. L'analyse :

Le but de l'analyse orienté objet est de modéliser le système du monde réel afin qu'il soit compréhensible. Pour cela, il faut examiner les besoins et analyser leurs implications. Un bon modèle d'analyse doit définir ce qui se fait sans se préoccuper de la manière dont cela doit être fait, et donc omettre les détails d'implémentation

II.1. Les cas d'utilisation :

Le diagramme de cas d'utilisation décrit la succession des opérations réalisées par un acteur (personne qui assure l'exécution d'une activité). C'est le diagramme principal du modèle UML, celui où s'assure la relation entre l'utilisateur et les objets que le système met en oeuvre.

L'étude des cas d'utilisation commence par la détermination des acteurs (utilisateurs) du système.

a) Détermination des acteurs :

Un acteur représente un rôle joué par une personne, un groupe de personnes ou par une chose qui interagit avec le système. Par définition les acteurs sont à l'extérieur du système.

Dans notre système qui se base sur la représentation des données dite imprécises ou floues et le raisonnement à base de cas floue les acteurs se répartissent dans les catégories suivantes :

L'utilisateur externe : Il peut être un patient ou toute personne intéressée par le diagnostic médical et ayant un minimum de connaissance et de savoir faire dans le domaine médical, ce type d'acteur est autorisé à faire un diagnostic médical (en recherchant un cas similaire au cas à traiter). Pour cela il doit pouvoir remplir les différents champs de cas à traiter, il peut également consulter la base de cas.

Médecin : Est un acteur qui enrichit et maintient la base de cas (modifier, supprimer, ajouter), et qui peut rechercher un cas similaire à un cas traité.

Voici leur représentation graphique :

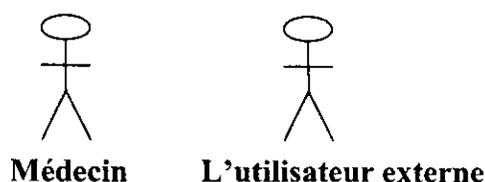


Figure 5.2 : représentation des catégories d'utilisateurs.

Chapitre 5 : Démarche de développement

b) Description textuelle des cas d'utilisation :

Les cas d'utilisation de notre système :

- Effectuer un diagnostic médical : C'est-à-dire le remplissage des différents champs du cas à traiter, puis la recherche d'un cas similaire à ce dernier dans la base de cas.
- Consultation de la base de cas : La consultation consiste en la visualisation des différents cas enregistrés dans la base de cas, elle peut se faire par n'importe quel utilisateur du système.
- Mise à jour des cas : C'est-à-dire l'enregistrement, la modification, et la suppression, des cas dans la base :
 - Seuls les médecins sont autorisés à modifier, à ajouter ou à suppression des cas. La saisie d'un nouveau cas ne peut se faire donc que par les médecins, et ceci après vérification de la non existence d'un cas identique dans la base, et après authentification de l'utilisateur.
- Authentification.
- Mise à jour de la table des poids : Seuls les médecins sont autorisés à modifier les poids, après authentification de l'utilisateur.
- Mise à jour de la table des médecins : Seul l'administrateur peut effectuer la mise à jour, en ajoutant, en supprimant un enregistrement de la table (code de médecin).

c) Diagramme de cas d'utilisation :

Les diagrammes de cas d'utilisation représentent les cas d'utilisation identifiés et l'acteur associé à chacun. Chaque utilisation possible du système se traduit par une ellipse étiquetée avec le nom de cas d'utilisation, à laquelle est connecté le ou les acteurs concernés par cette interaction avec le système.

Dans le cas de notre étude, le diagramme des cas d'utilisations s'organise en deux modules :

A. Mise à jour et consultation de la base de cas.

B. Effectuer un diagnostic médical.

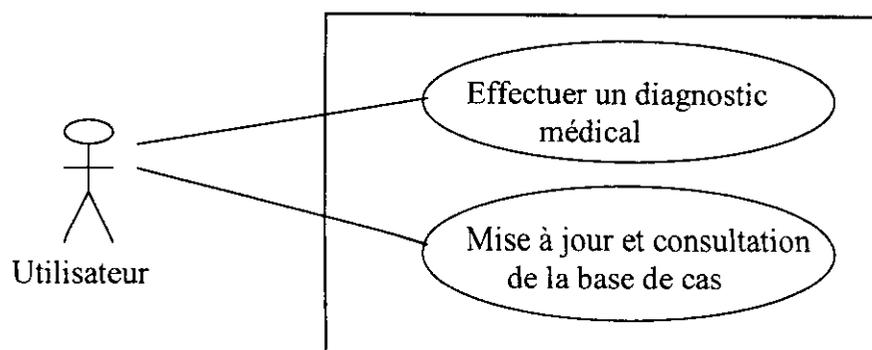


Figure 5.3 : Diagramme générale de cas d'utilisation de notre système

Chapitre5 : Démarche de développement

Dans la paragraphe suivant, nous expliquerons chaque cas d'utilisation en donnant les diagrammes de sous cas d'utilisation qui lui correspondent, afin de pouvoir comprendre le diagramme générale.

A. Mise à jour et consultation de la base de cas :

A.1. Cas d'utilisation « consultation de la base de cas » :

La consultation de la base de cas peut se faire par n'importe quel type d'utilisateurs.

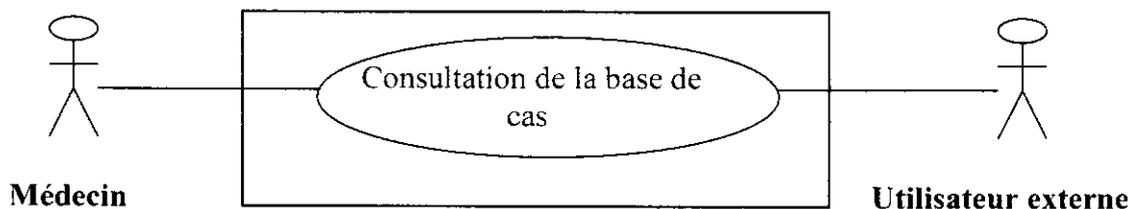


Figure 5.4 : Diagramme de cas d'utilisation «consultation de la base de cas »

A.2. Cas d'utilisation « mise à jour de la base de cas » :

Seuls les médecins sont autorisés à modifier, à ajouter ou à suppression des cas. La saisie d'un nouveau cas ne peut se faire donc que par les médecins, et ceci après vérification de la non existence d'un cas identique dans la base, et après authentification de l'utilisateur.

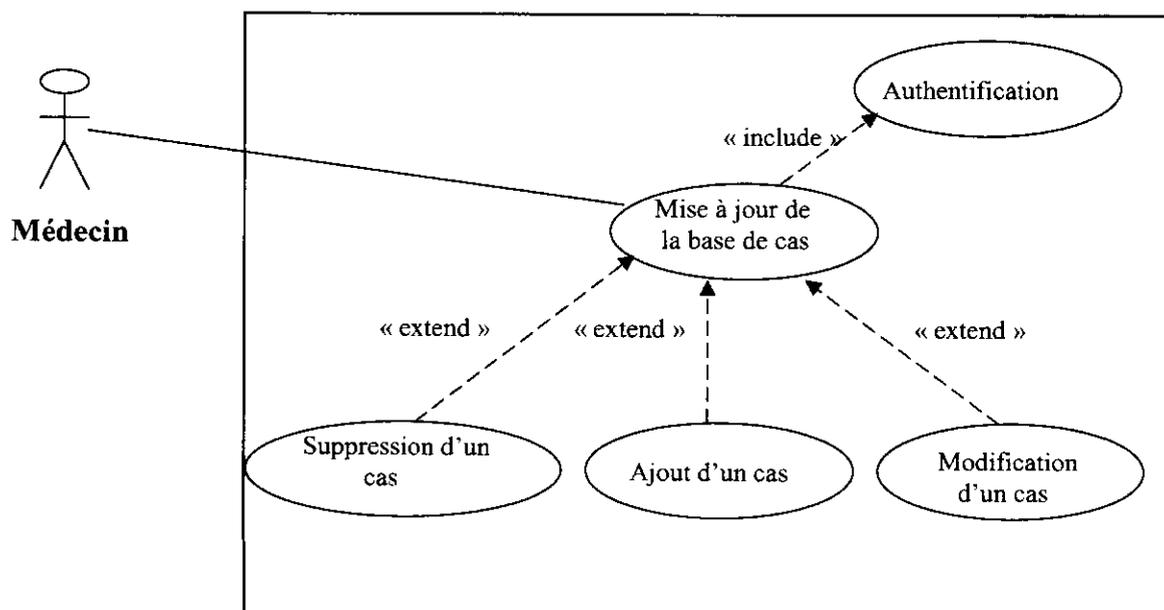


Figure 5.5 : Diagramme de cas d'utilisation « mise à jour de la base de cas »

B. Effectuer un diagnostic médical :

C'est-à-dire le remplissage des différents champs du cas à traiter par le médecin ou par un utilisateur externe, et le médecin peut modifier la table des poids après authentification si nécessaire.

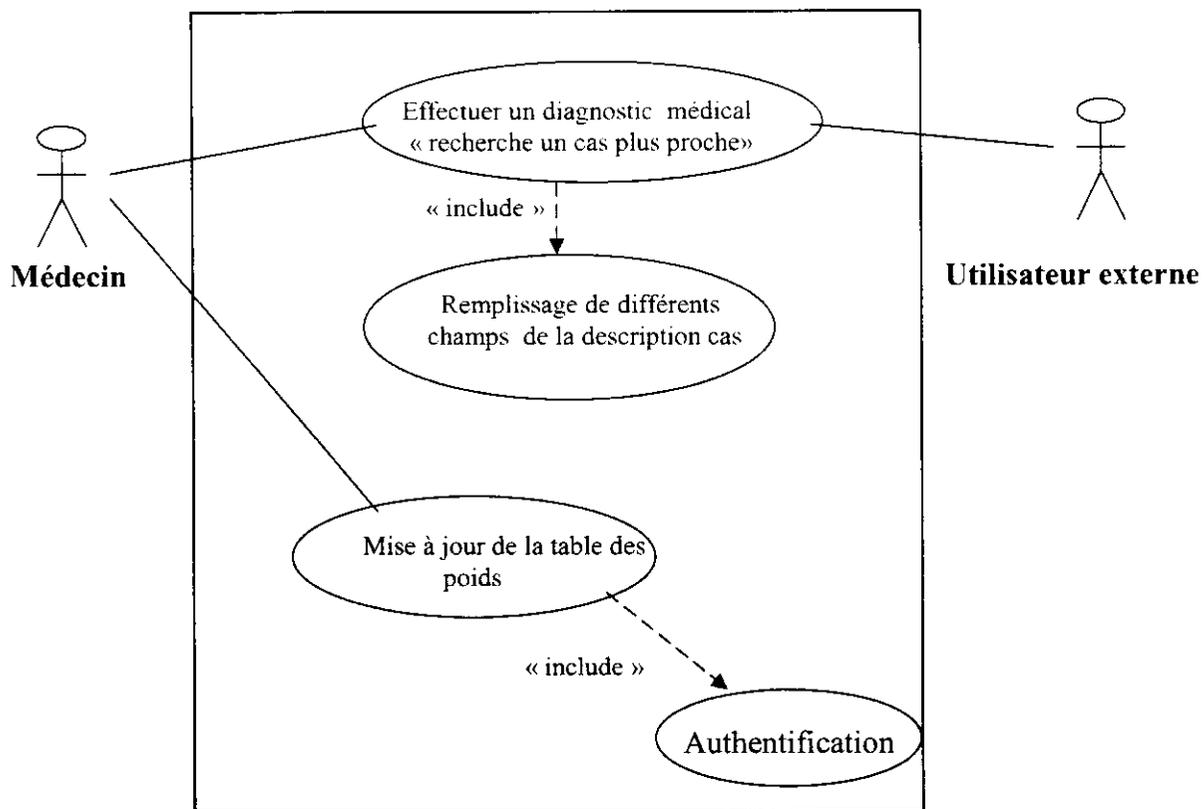


Figure 5.6 : Diagramme de cas d'utilisation « Effectuer un diagnostic médical »

Diagramme de cas d'utilisation général :

Les différentes fonctionnalités offertes par notre système forment ainsi un ensemble de cas d'utilisation (“Use Case”), afin de les formaliser, UML propose à travers des diagrammes des cas d'utilisation de répertorier et de structurer les fonctionnalités que le futur système offrira à ses utilisateurs.

La figure suivante illustre ce diagramme.

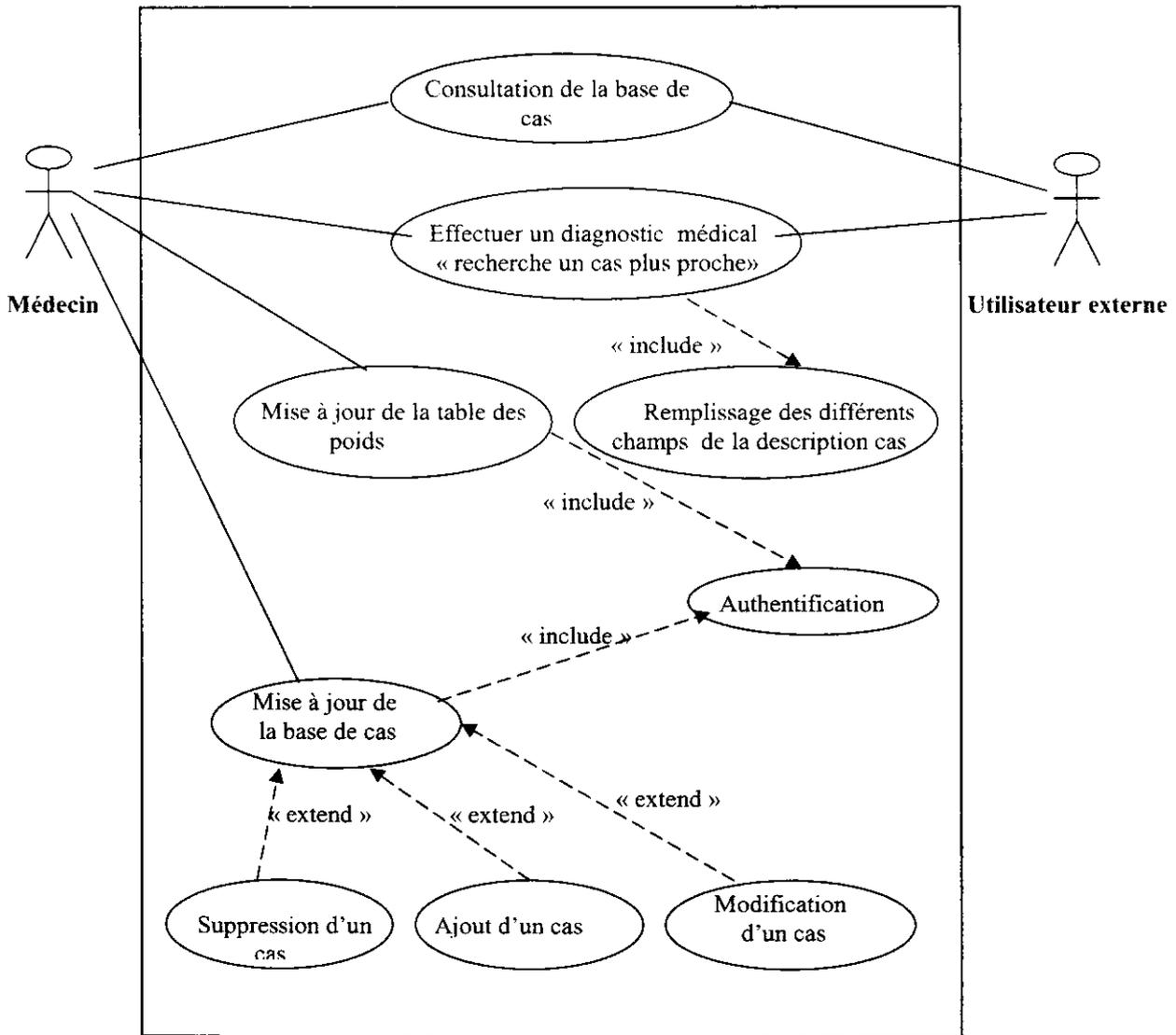


Figure 5.7 : Diagramme générale de cas d'utilisation de notre système

II.2. Diagrammes de séquences :

Les diagrammes de séquences permettent de représenter des collaborations entre objets selon un point de vue temporel, on y met l'accent sur la chronologie des envois de messages. L'ordre d'envoi d'un message est déterminé par sa position sur l'axe vertical du diagramme ; le temps s'écoule "de haut en bas" de cet axe.

Les diagrammes de séquences peuvent servir à illustrer un cas d'utilisation.

Scénario 1 :

a) Effectuer un diagnostic médical par un utilisateur externe :

Le module effectuer un diagnostic médical constitue la partie la plus important de notre système dans le cas de représentation des données imprécises ou floue et le calcul de similarité par des méthodes floues.

- Quand l'utilisateur externe veut effectuer un diagnostic médical, le système lui affiche une fiche à remplir, cette fiche contient les champs de la description du cas à traiter.
- L'utilisateur externe remplit la fiche.
- L'utilisateur externe lance la recherche des cas similaires au cas saisi (cas à traiter) dans la base de cas.
- Le système fait le calcul de similitude, ensuite il affiche les différents cas trouvés.
- D'après ces résultats, l'utilisateur externe prend la décision :
 - Soit il trouve un cas qui est similaire au cas à traiter, dans ce cas, le diagnostic du cas trouvé est copié au cas recherché.
 - Soit le casb identique au cas à traiter ne se trouve pas dans la base de cas, donc l'utilisateur peut diagnostiquer le cas en s'inspirant du cas le plus proche de son cas.

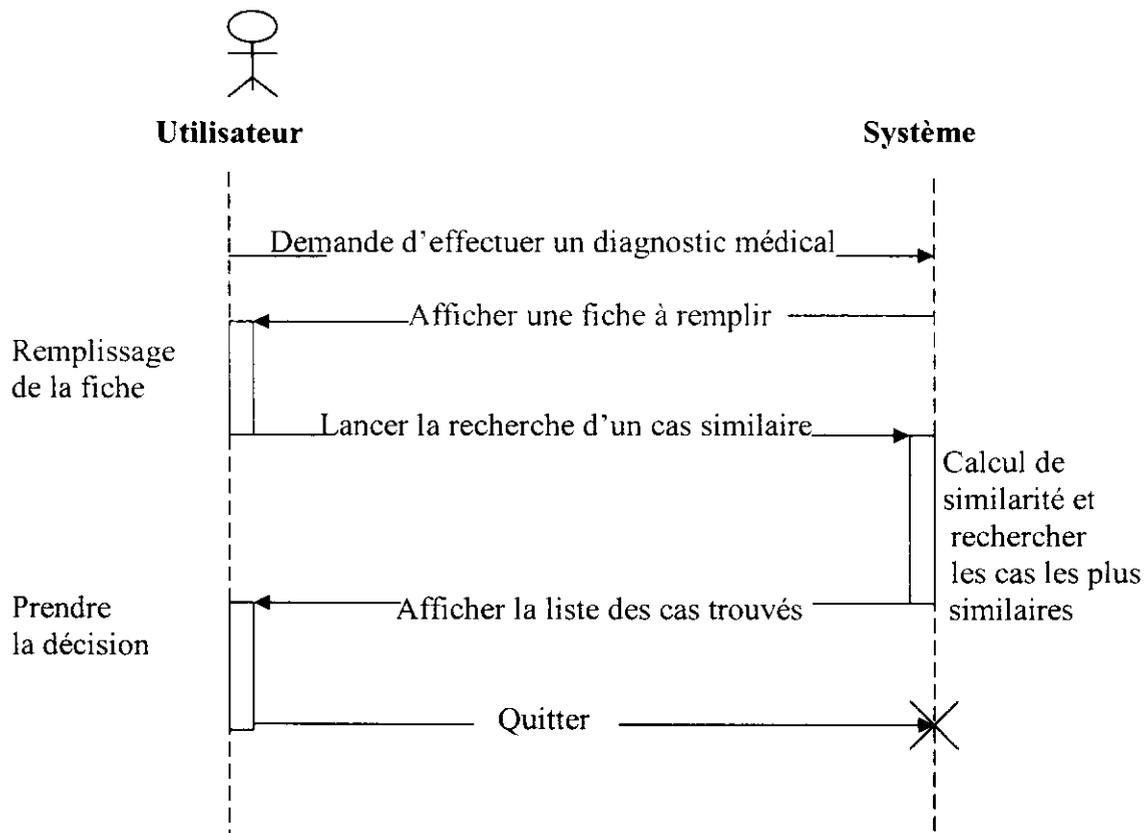


Figure 5.8 : diagramme de séquence pour effectuer un diagnostic par un utilisateur externe

b) Effectue un diagnostic médical par un médecin:

- Quand le médecin veut effectuer un diagnostic médical, le système lui affiche une fiche à remplir, cette fiche contient les champs de la description du cas à traiter.
- Le médecin remplit la fiche.
- Le médecin lance la recherche des cas similaires au cas saisi (cas à traiter) dans la base de cas.
- Le système fait le calcul de similitude, ensuite il affiche les différents cas trouvés.

Chapitre 5 : Démarche de développement

- D'après ces résultats, le médecin prend la décision :
 - Soit il trouve un cas qui est similaire au cas à traiter, dans ce cas, le diagnostic du cas trouvé est copié au cas recherché.
 - Soit le cas identique au cas à traiter ne se trouve pas dans la base de cas, donc l'utilisateur peut diagnostiquer le cas en s'inspirant du cas le plus proche de son cas.
 - S'il ne trouve pas un cas assez similaire, il peut faire le diagnostic lui-même.

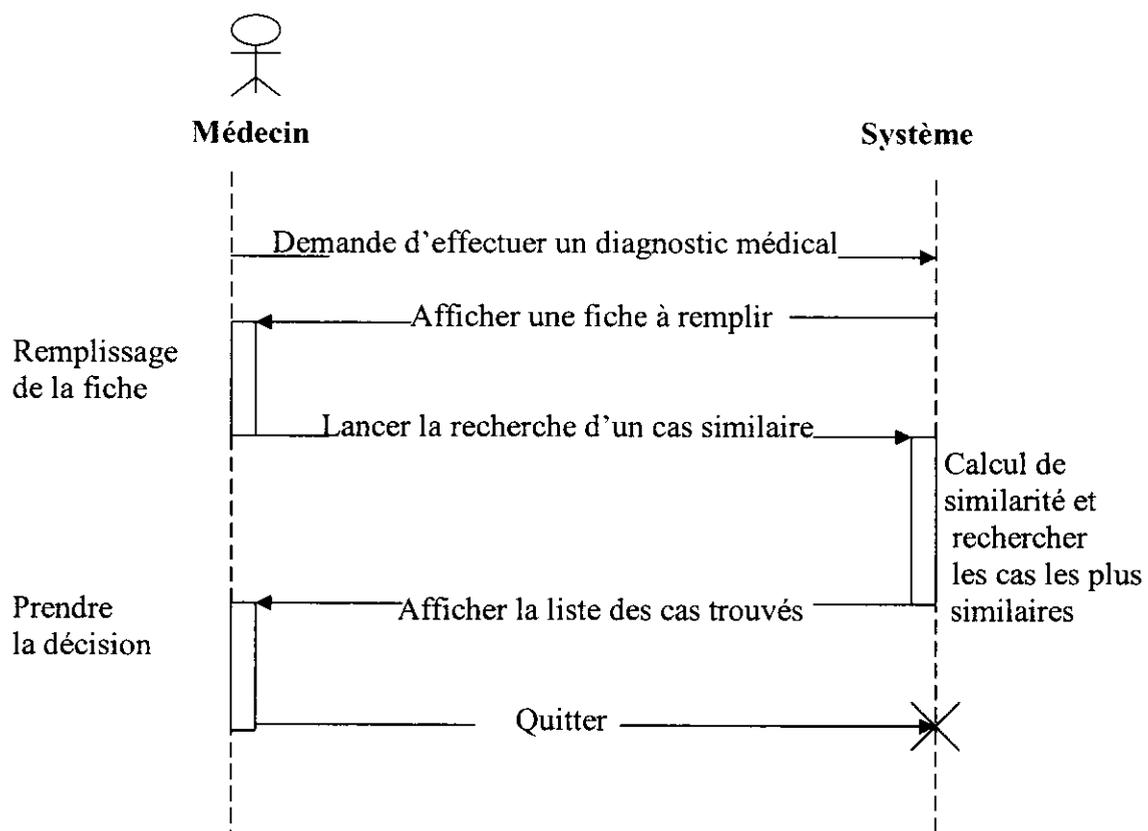


Figure 5.9 : diagramme de séquence pour effectuer un diagnostic par un médecin

Scénario 2 :

Consultation de la base de cas :

- L'utilisateur demande la consultation de la base de cas.
- Le système affiche tous les cas de la base de cas.
- L'utilisateur consulte la liste des cas.

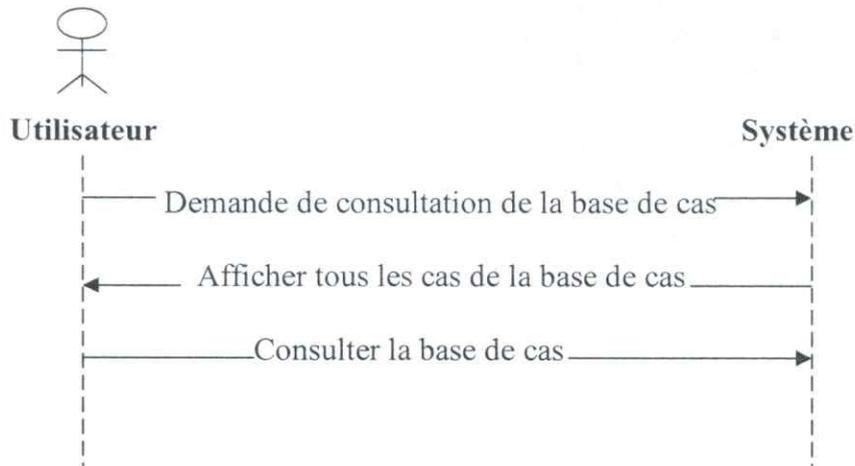


Figure 5. 10 : diagramme de séquence pour Consultation de la base de cas

Scénario 3 :

L'ajout d'un nouveau cas par le médecin :

- Le médecin demande l'ajout d'un nouveau cas.
- Le système lui demande d'authentification (saisi son code).
- Après contrôle du code, une fiche de saisie de nouveau cas s'affiche.
- Le médecin remplit la fiche.
- Le médecin demande l'enregistrement du cas.
- Le système effectue un calcul automatique de similitude floue pour s'assurer qu'aucun cas identique n'existe dans la base de cas.
 - Si aucun cas n'existe, le système lui affiche un message de confirmation d'enregistrement.
 - ✓ Le médecin accepte l'enregistrement de nouveau cas.
 - ✓ Le système enregistre le nouveau cas.
 - Sinon le système affiche un message informant qu'un cas identique existe déjà dans la base de cas.

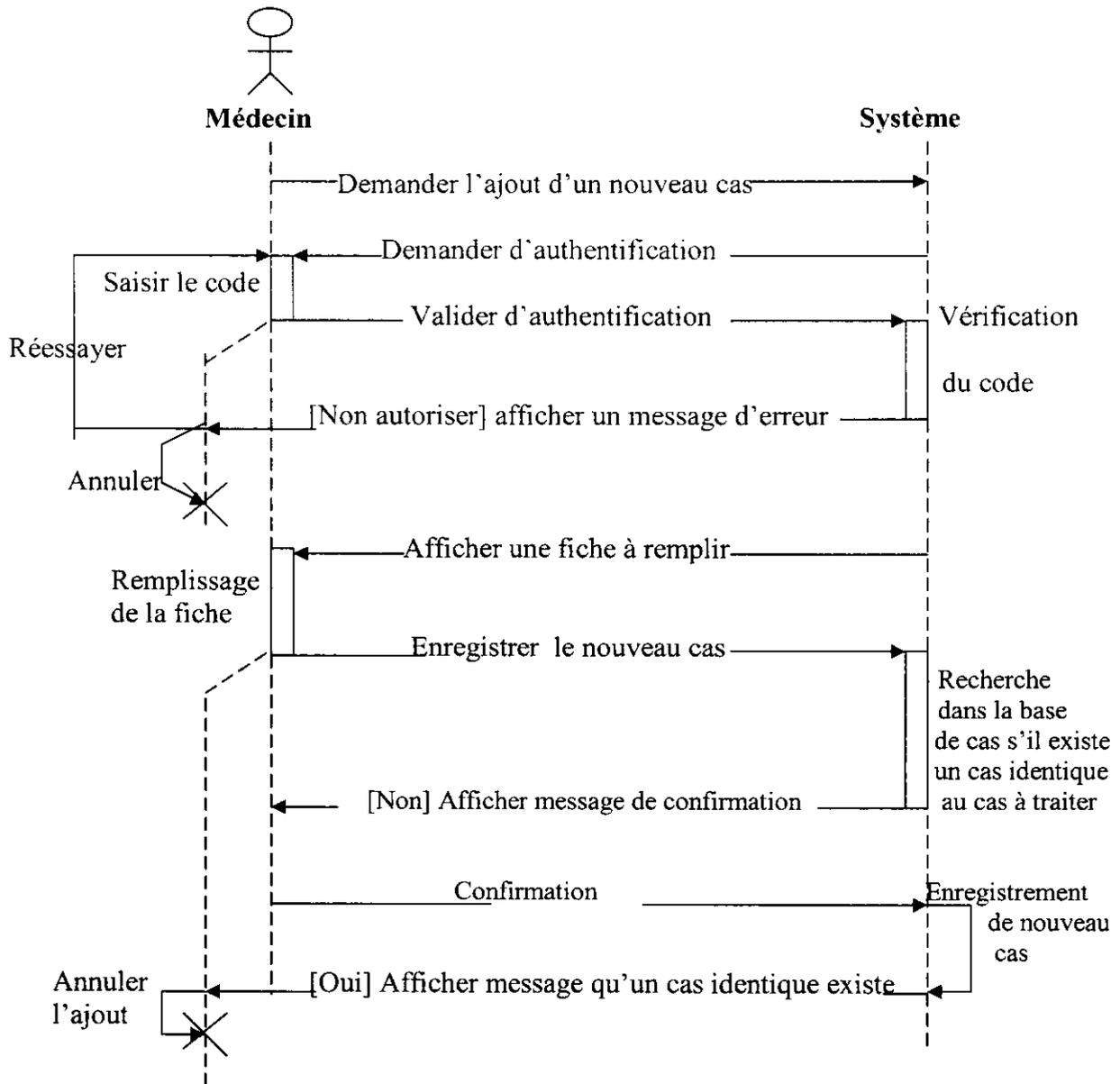


Figure 5.11 : Diagramme de séquence pour l'ajout d'un cas

Chapitre 5 : Démarche de développement

Scénario 4 :

La suppression d'un cas dans la base de cas par le médecin :

Le médecin a la possibilité de supprimer son cas lorsque le traitement donné n'est pas efficace.

- Le médecin demande la suppression d'un cas.
- Le système lui demande d'authentification (saisi son code).
- Après contrôle du code. le système lui demande d'écrire le numéro du cas à supprimer.
- Le médecin écrit le numéro du cas à supprimer.
- Le système fait la recherche de ce cas :
 - S'il trouve le cas dans la base de cas. alors le système demande la confirmation de la suppression.
 - ✓ Le médecin confirme la suppression de ce cas.
 - ✓ Le système supprime le cas.
 - Sinon le cas ne se trouve pas dans la base de cas, donc la suppression est annulée.

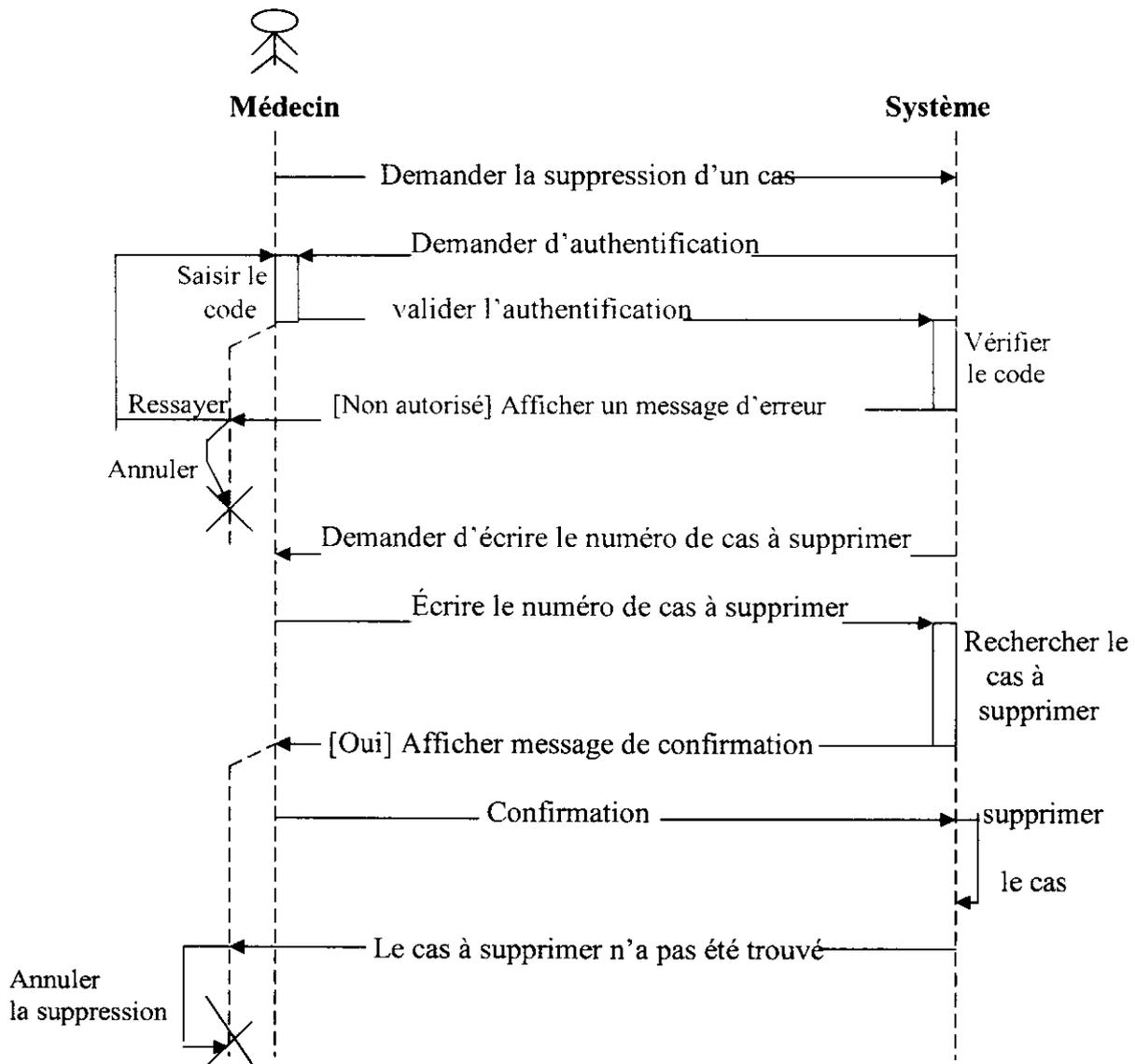


Figure 5.12 : Diagramme de séquence pour la suppression d'un cas

Scénario 5 :

La modification d'un cas dans la base de cas par le médecin :

- Le médecin demande la modification d'un cas.
- Le système lui demande d'authentification (saisi son code).
- Après contrôle du code, le système lui demande d'écrire le numéro du cas à modifie.
- Le médecin écrit le numéro du cas à modifie.
- Le système fait la recherche de ce cas :

Chapitre 5 : Démarche de développement

- S'il trouve le cas dans la base de cas, alors le système affiche la fiche.
 - ✓ Le médecin modifie ce qu'il veut modifier, et demande d'enregistrer la modification.
 - ✓ Le système demande la confirmation de la modification.
 - ✓ Le médecin accepte la modification.
- Sinon le cas ne se trouve pas dans la base de cas, donc la modification est annulée.

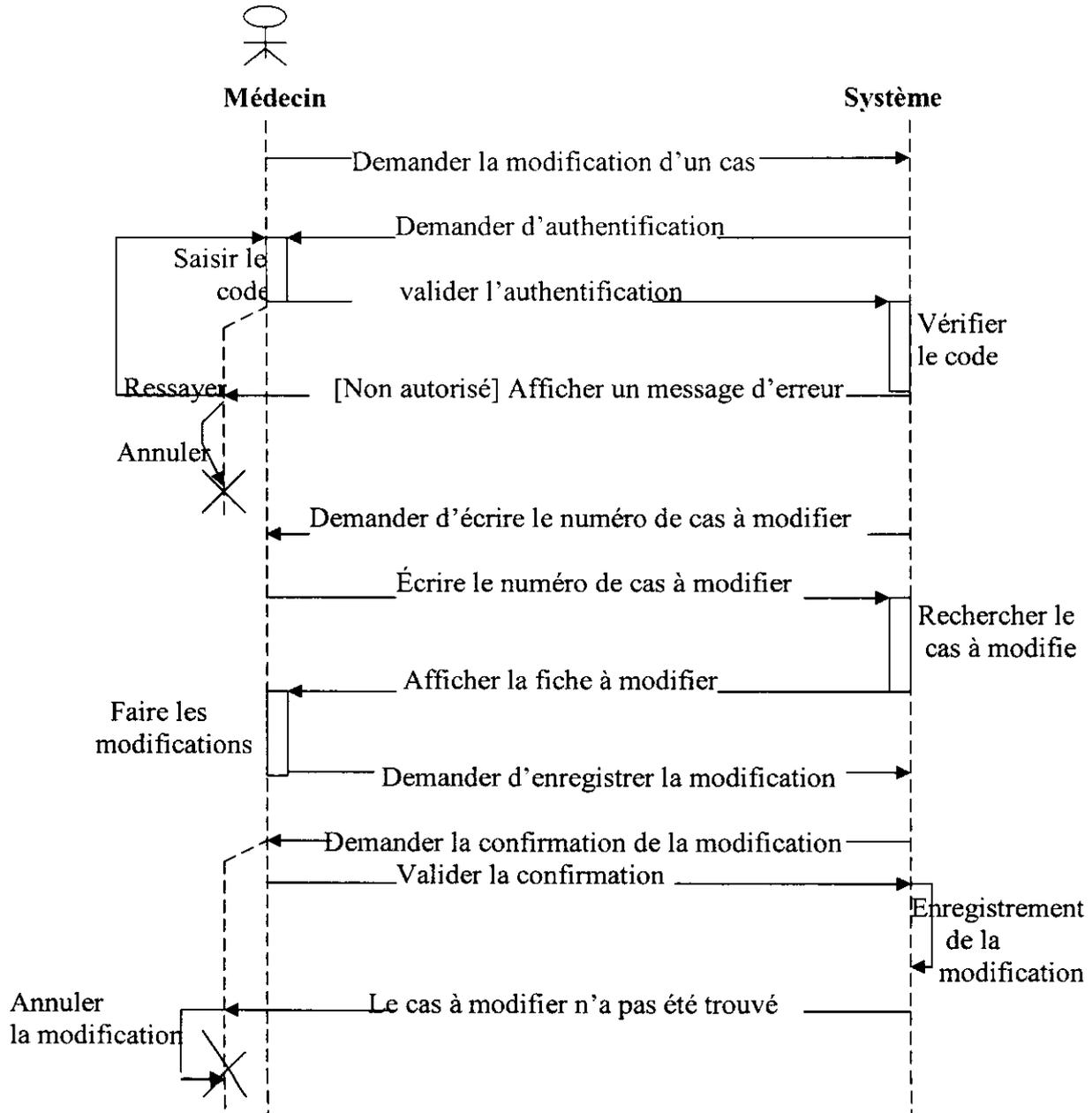


Figure 5.13 : Diagramme de séquence pour la modification d'un cas

II.3. Diagramme de collaboration :

Les diagrammes de collaboration montrent des interactions entre objets (instances de classes) et acteurs. Ils permettent de représenter le contexte d'une interaction, car on peut y préciser les états des objets qui interagissent.

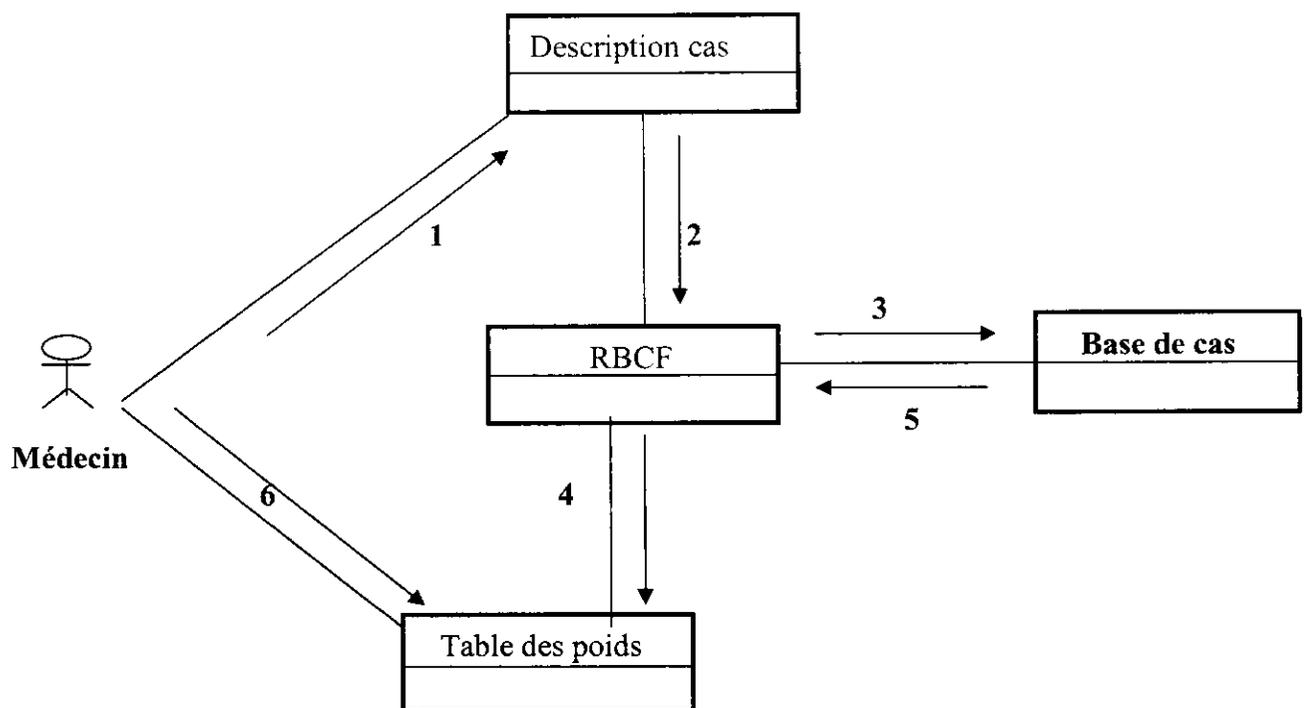


Figure 5.14 : Diagramme de collaboration pour effectuer un diagnostic par le médecin

- 1- Décrire la description de cas
- 2- Le système effectue le raisonnement à base de cas flou par la méthode k-nn **floue**.
- 3- Le raisonnement à base de cas flou permet de chercher dans la base de cas.
- 4- Le raisonnement à base de cas flou utilise la table des poids pour calculer la similarité.
- 5- Les cas similaires sont trouvés.
- 6- Le médecin peut mettre à jour la table de poids.

II.4. Diagramme de classes :

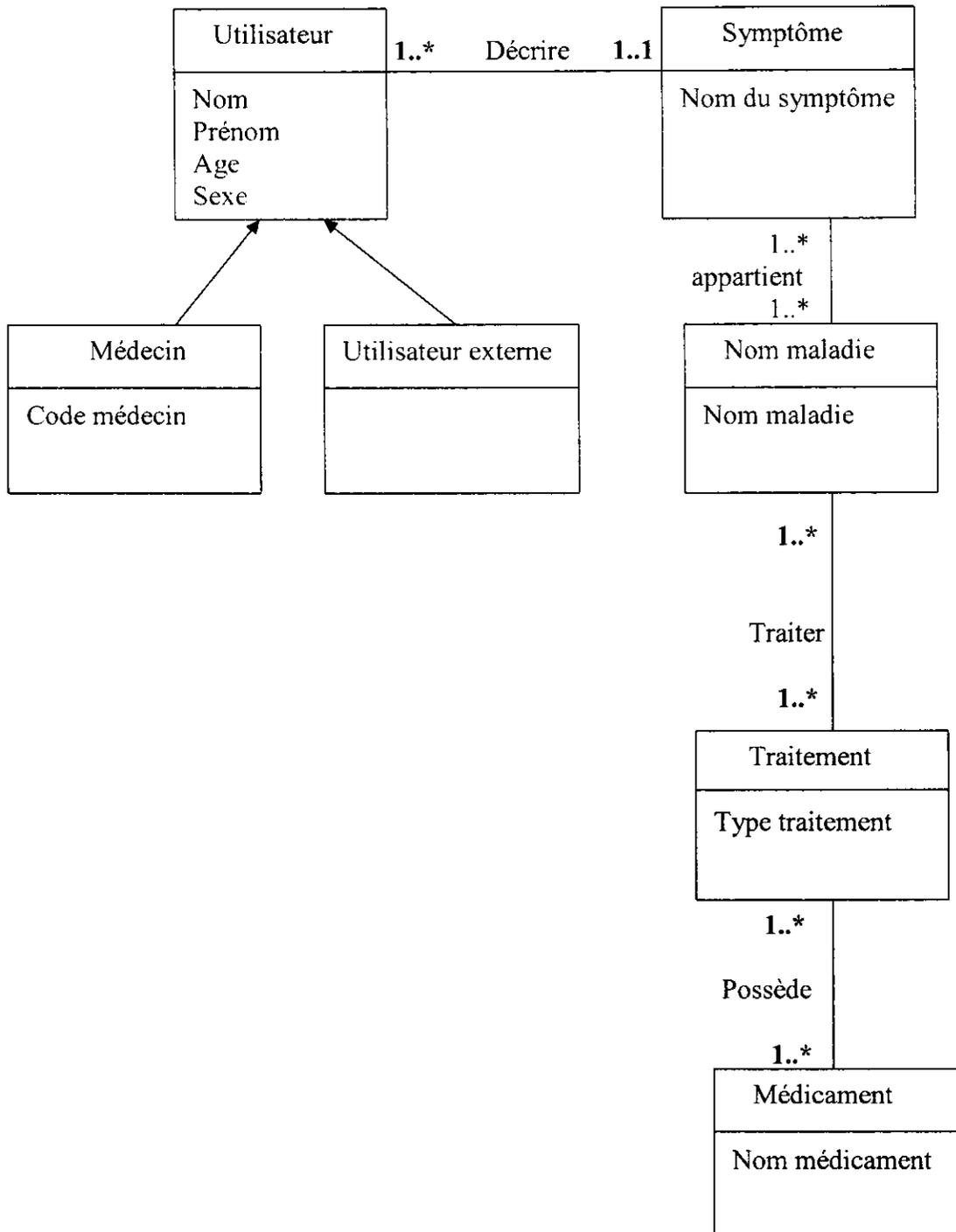


Figure 5.15 : Diagramme de classes de la phase d'analyse

III. Conception :

A. La conception du système :

C'est une phase très stratégique dans le processus de résolution d'un problème et de construction d'une solution. La conception du système implique des décisions sur l'organisation du système en sous système. L'organisation complète du système est appelée *architecture du système*.

A.1. Architecture du système :

Nous avons choisi de présenter notre système par une base de cas qui contient des données floues ou imprécises. l'architecture générale du notre système est schématisée dans la **figure** suivant.

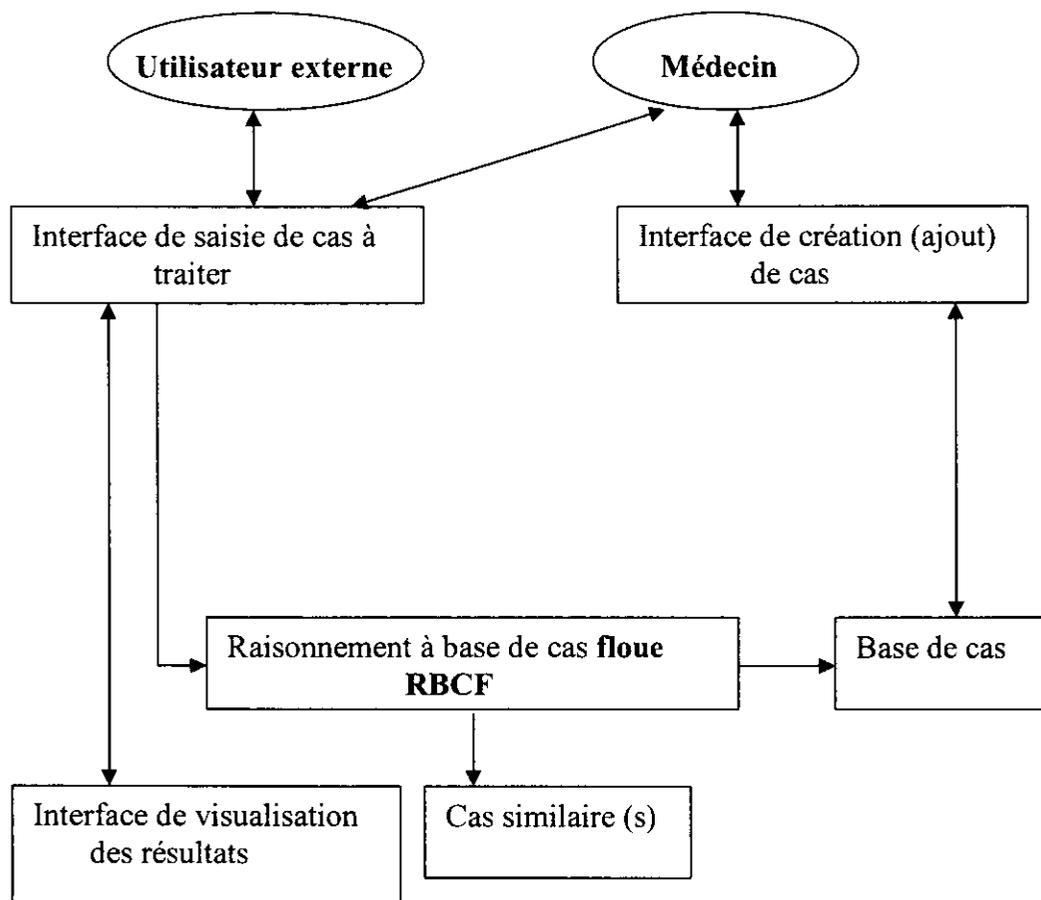


Figure 5.16 : Architecture générale du système.

Chapitre5 : Démarche de développement

a. Module de raisonnement à base de cas floue :

Ce module contient la mise en œuvre des différentes étapes du raisonnement: la remémoration la réutilisation, la révision et la mémorisation.

b. Module interface utilisateur :

- ❖ **Interface de création d'un nouveau cas** : Elle est chargée de créer un nouveau cas dans la base de cas.
- ❖ **Interface de saisie du cas à traiter** : Elle correspond à une interface de saisie d'un cas, afin de faire un diagnostic médical.
- ❖ **Interface de visualisation des résultats** : Elle permet à l'utilisateur de visualiser les différents cas similaires trouvés dans la base de cas.

A. 2. Diagramme d'activités :

Le diagramme d'activités permet de représenter la dynamique du système d'information. C'est un graphe orienté qui décrit un enchaînement de traitement. Le déroulement ainsi présenté est appelé flot de contrôle, l'enchaînement des activités peut être soumis à des branchements ou à des synchronisations. La visualisation de couloirs d'activités permet de représenter la répartition de la responsabilité des activités entre différents acteurs.

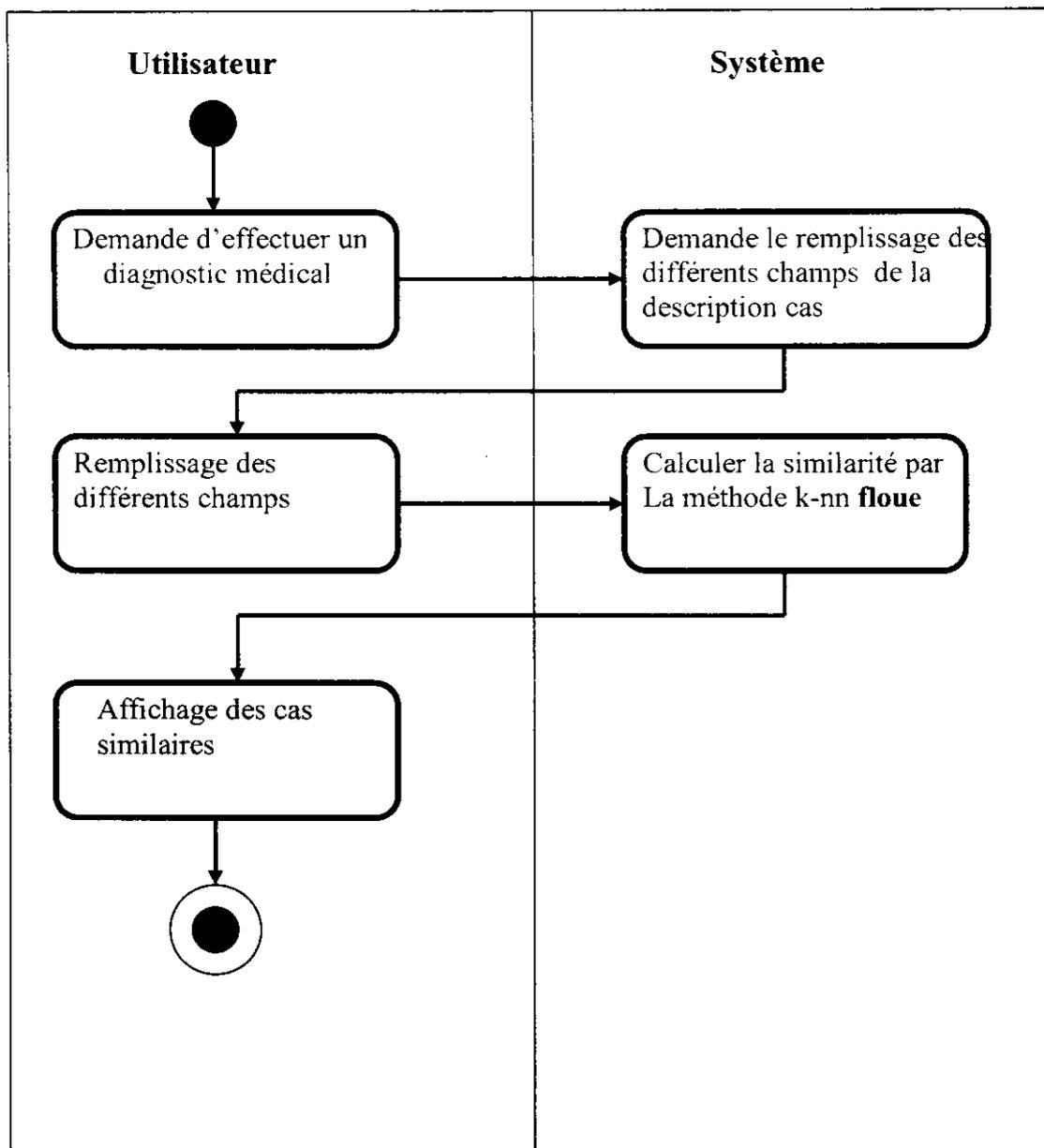


Figure 5.17 : Diagramme d'activités pour le cas d'utilisation « Effectuer un diagnostic médical »

Chapitre 5 : Démarche de développement

B. La conception des objets :

Elle définit de façon complète les classes et les associations utilisées dans l'implémentation, ainsi que les interfaces et les algorithmes des méthodes employées pour implémenter les opérations.

B.1. Diagramme de classes :

Description des attributs :

Classe	Attribut	Code attribut	Type	Valeur	
Cas	Numéro cas	Num_cas	N	Variable	
Description cas	Nom du médecin	Nom_med	A	Variable	
	Age	age	A	Variable	
	Sexe	sexe	N	Variable	
	Poids	Poids	A	Variable	
	Adresse	Adresse	N	Variable	
	Date de consultation	Date_cons	A	Variable	
	Fièvre	Fièvre	D	forte, modérée, inexistante	
	Asthénie (fatigue)	Asthénie	VF ¹	légère, très légère, intense, inexistante	
	Frissons	Frissons	VF	présent, absent, rare	
	Douleurs musculaires	Doul_musc	VF	légère, modérée, important	
	Douleurs articulaires	Doul_artic	VF	légère, modérée, important	
	Courbatures	Courbatures	VF	présente, absent, légère	
	Céphalées (maux de tête)	Céphalées	VF	modérée, intense, inexistante	
	Toux	Toux	VF	sèche, grasse, un peu sèche, un peu grasse	
				VF	
		Maux de gorge	Maux_gorge		légère, inexistante, intense
		Nez bouche	Nez_bouche	VF	présent, absent
		Éternuements	Éternuements	VF	présent, absent
		Ecoulements nasaux	Ecoule_nasaux	VF	présente, inexistant, rare
		Douleur de la poitrine	Doul_poit	VF	Fréquent, rare, inexistante
		Conjonctivite	Conjonctivite	VF	présente, absent
		Des sueurs nocturnes	Nocturnes	VF	présente, inexistant, rare
		Somnolence	Somnolence	VF	présente, inexistante, rare
		Une bronchite	bronchite	VF	présente, absent
		Une diarrhée	diarrhée	VF	rare, inexistante
		Vomissements	Vomiss	VF	Fréquent, rare, inexistante
				VF	
Diagnostic	Nom maladie	Nom_maladie	A	Variable	
Médicament	Nom médicament	Nom_medicam	A	Variable	
Table des poids	Poids	Poids	N	Variable	

Tableau 5.1 : Description des attributs

¹VF : Valeur Floue

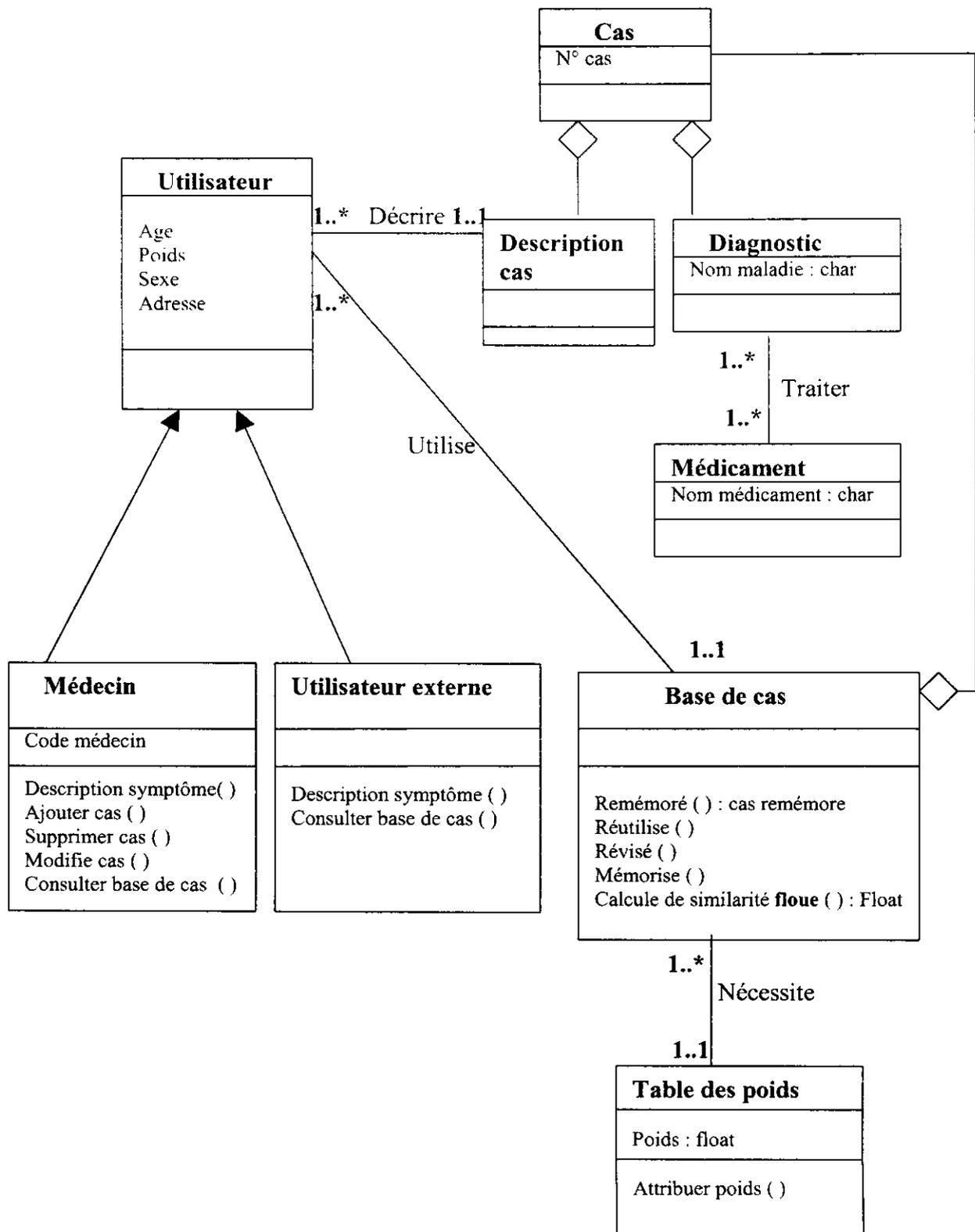


Figure 5.18 : Diagramme de classes de la phase de conception

Chapitre5 : Démarche de développement

IV. Implémentation :

L'étape d'implémentation consiste à produire le logiciel correspondant à la conception décrite dans la phase de conception. Autrement dit, l'implémentation est une phase au cours de laquelle les algorithmes (dans notre cas les diagrammes de séquences) définies dans la partie conception sont traduits dans un langage de programmation.

IV.1. Choix de langage de programmation :

Le système que nous avons conçu est implémenté sous le système d'exploitation Windows. Cette implémentation est effectuée sur un PC (Personal Computer) en utilisant le langage de programmation Builder C++, ce choix est motivé par les critères suivants :

Pour quoi C++ Builder ?

- Le langage C++ Builder est souple et puissant ;
- Programmation à base Orienté Objet ;
- Permet d'implémenter des interfaces rapides d'application sous Windows ;
- Permet l'exécution rapide surtout dans le cas des calculs compliqués et itératives le temps de réponse est réduit dans le cas de recherche dans les grandes base de données.

Dans le paragraphe qui suit nous allons présenter un ensemble de résultats obtenus pendant le développement de notre système de diagnostic médical.

IV.2. Mise en œuvre de l'interface utilisateur :

La figure suivante illustre la page principale de notre système, qui présente les différents types d'utilisateurs du système. Chaque type d'utilisateur possède une session qui contient des fonctionnalités qui lui sont propre.

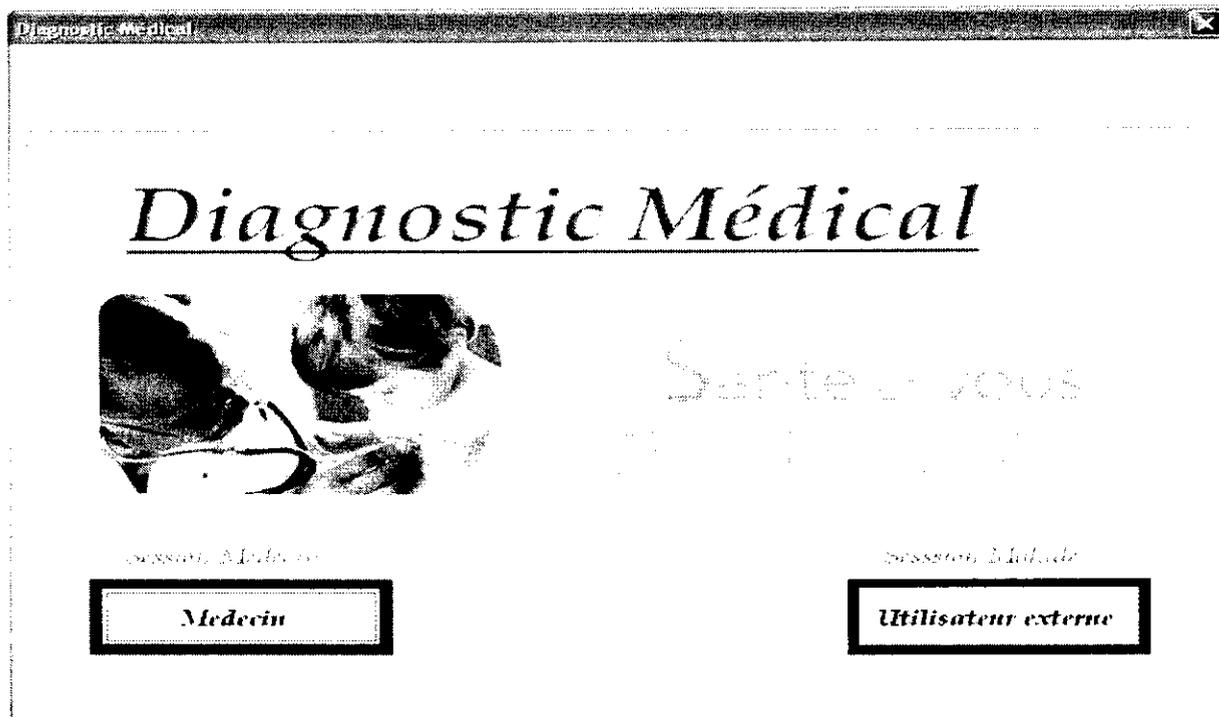


Figure 5.19 : Représentation de l'interface principal

- **Session Médecin :**

Une authentification de l'utilisateur est exigée. Pour cela, le système lui demande de saisir son mot passe. Après contrôle du mot de passe l'interface de session médecin s'affiche.

Le type d'opérations que nous présentons dans cette session correspond à l'insertion d'un nouveau cas, la suppression, la modification d'un cas et la modification de la table des poids. Comme précisé lors de l'étape de conception, ce type d'opérations ne peut être effectué que par les médecins. Ce type d'utilisateur peut également effectuer un diagnostic médical ainsi qu'une consultation de la base de cas. (Figure 5.20)

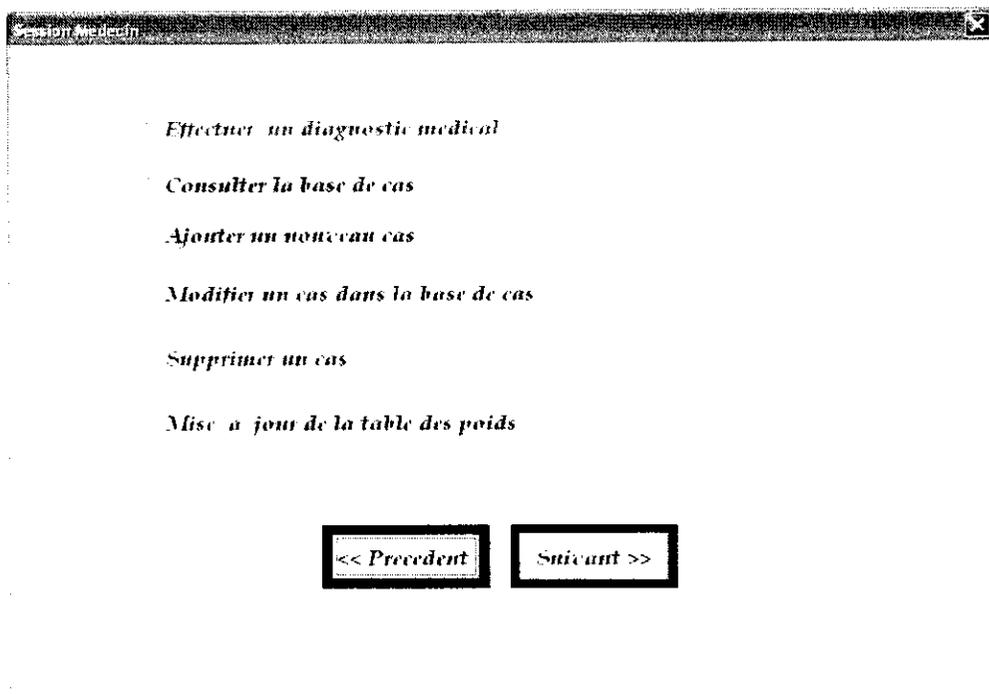


Figure 5.20 : L'interface de la session médecin

- **Session Utilisateur externe :**

Ce type d'utilisateur n'ayant pas une grande expérience du domaine, il n'a accès qu'à la base de cas, et à effectuer un diagnostic médical. (Figure 5.21)

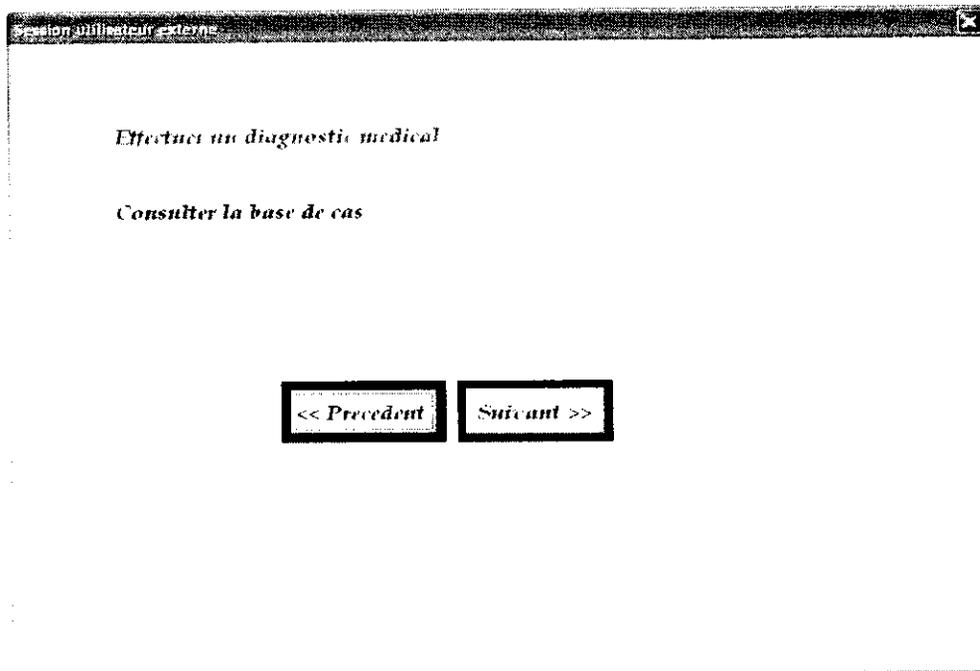


Figure 5.21 : L'interface de la session utilisateur externe

- **Consultation de la base de cas :**

La consultation de la base de cas peut se faire par n'importe quel type d'utilisateur. La **figure 5.22** illustre un exemple de consultation de la base de cas. Le bouton « Quitter » permet de fermer la fenêtre de consultation.

Chapitre5 : Démarche de développement

Consultation de la base de cas

Num. Cas	Age_Malade	Sexe_Malade	Poids_Malade	Adresse_Malade	Nom_medicin	Date_Consultation	Fievre	Asthénie	Frissons
1	55	Féminin	50	Bhida	kebfar	15-06-2007	faible	non	fréquent
2	56	Masculin	90	Tapava	Chelal	15-06-2007	modéré	fréquent	fréquent
3	57	Masculin	90	Bhida	fonati	15-06-2007	faible	insistant	fréquent
4	58	Masculin	75	Chelal	Bou laoued	16-05-2007	faible	non	non
5	45	Masculin	96	Tapava	kekak	16-05-2007	faible	fréquent	fréquent
6	55	Masculin	75	chechell	salika	01-01-2007	non pon	fréquent	fréquent
7	54	Féminin	74	Hachout	kebfar	04-04-2007	faible	fréquent	fréquent
8	19	Féminin	65	chechell		03-03-2007	faible	fréquent	fréquent
9	50	Féminin	75	Méda	chechall	22-01-2007	faible	insistant	fréquent
10	24	Féminin	60	Oua	salika	14-06-2007	faible	fréquent	fréquent

Précédent Quitter

Figure 5.22 : Consultation de la base de cas

- **Effectuer un diagnostic médical (Recherche du cas similaire)**

Le module « effectuer un diagnostic médical » constitue la partie la plus importante de notre système de raisonnement à base de cas fluu.

Quand l'utilisateur de notre système (Médecin ou Utilisateur externe) veut effectuer un diagnostic médical une fiche de saisie de cas lui est affichée. La **figure 5.23** présente le formulaire qu'il doit remplir.

Chapitre 5 : Démarche de développement

Label	Value	Label	Value
Age du malade :	22	Fièvre :	modere
Poids du malade :	75	Asthénie :	fréquent
Sexe du malade :	Feminin	Frissons :	inexistant
Adresse :	Alger	Toux :	
Date de consultation :	12/06/2007	Courbatures :	
Maux de tête (céphalées) :	fréquent	Nez bouche :	
gorge (pharyngite) :	fréquent instantané rare	Eternement :	
Douleurs Musculaires :		Ecoulements nasaux :	
Articulaires :		Conjonctivite :	
Poitrines :		Sueurs nocturnes :	
Digestives :		Somnolence :	
		Diarrhée :	
		Vomissements :	
		Bronchite :	

<< Precedent
Suivant >>

Figure 5.23 : Saisie de la description d'un cas

Après avoir rempli les différents champs de la fiche, l'utilisateur clique sur le bouton «Suivant » pour passer à la page suivante. Une autre fenêtre s'affiche, elle est structurée en deux parties : la partie supérieure de la page contient le cas à traiter, la partie inférieure contient les cas similaires trouvés. Un bouton «Recherche » permet de lancer la recherche des cas similaires. La **figure 5.24** illustre un cas de calcul de similitude, le cas à traiter est rempli automatiquement par les différentes valeurs saisies par l'utilisateur dans la fiche précédente **Figure 5.23**.



Chapitre5 : Démarche de développement

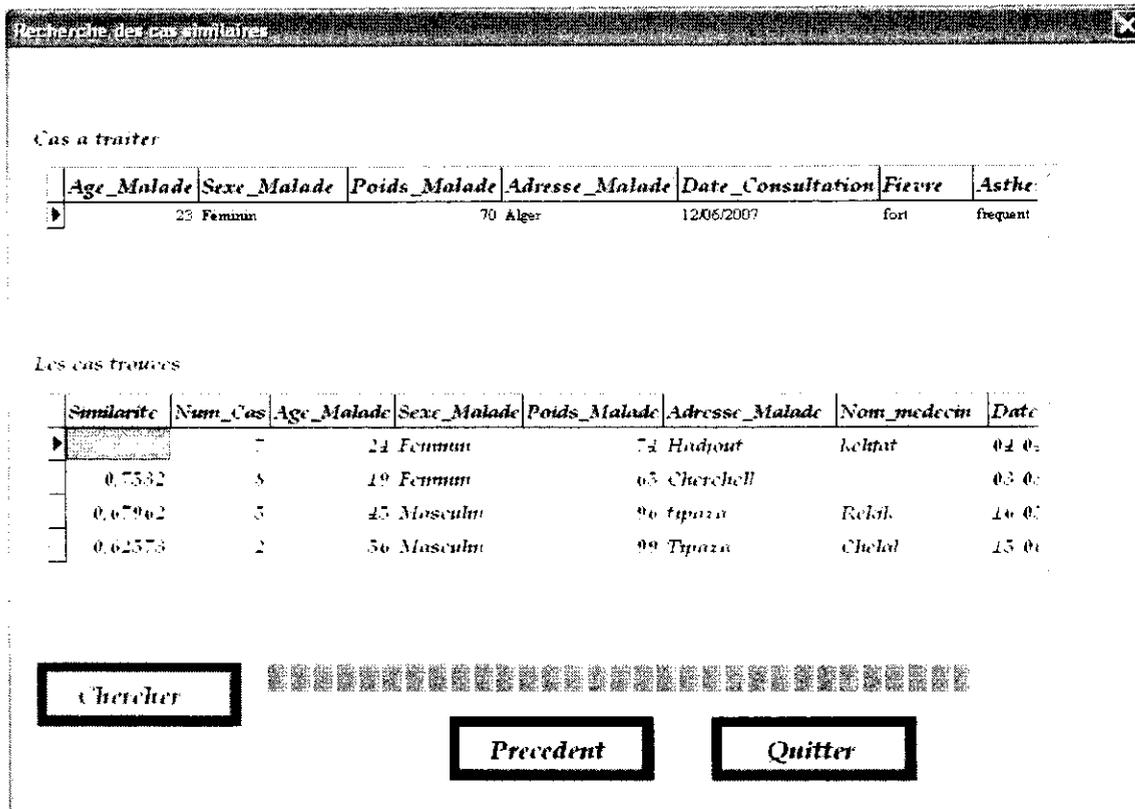


Figure 5.24 : Recherche du cas similaire

- L'ajout d'un nouveau cas dans la base de cas

L'ajout d'un nouveau cas dans la base de cas, étant une tâche propre aux médecins. Pour cela la fiche «Description du cas» s'affiche, le médecin remplit les différents champs de la fiche. (Figure 5.25)

Chapitre 5 : Démarche de développement

The screenshot shows a window titled "Description du cas" with a close button (X) in the top right corner. The form is organized into two columns of fields, each with a label and a value or a dropdown menu. The fields are as follows:

Numéro du cas :	12	Fièvre :	fort
Age du malade :	24	Asthénie :	fréquent
Poids du malade :	65	Frissons :	fréquent
Sexe du malade :	Féminin	Toux :	fréquent
Adresse :	ALGER	Courbatures :	fréquent
Date de consultation :	12/06/2007	Nez bouche :	inexistant
Nom du médecin :	Yousf	Eternement :	fréquent
Maux de :		Écoulements nasaux :	rare
tête (céphalées) :	fréquent	Conjonctivite :	inexistant
gorge (pharyngite) :	fréquent	Sueurs nocturnes :	inexistant
Douleurs :		Sommolence :	inexistant
Musculaires :	fréquent	Diarrhée :	rare
Articulaires :	inexistant	Vomissements :	inexistant
Poitrines :	rare	Bronchite :	inexistant
Digestives :	rare		

At the bottom right of the window, there are two buttons: "Suivant" and "<< Précédent".

Figure 5.25 : Saisie de la description du cas

Après avoir rempli les différents champs de la fiche, le médecin clique sur le bouton «Suivant » pour passer à la page suivante. Une autre fenêtre s'affiche, elle est structurée en deux parties : la partie supérieure de la page contient le cas à ajouter, la partie inférieure contient les cas similaires trouvés. Un bouton «Recherche » permet de lancer la recherche des cas similaires. Dans cette fenêtre le médecin vérifie si le cas saisi existe déjà dans la base de cas ou non. (Figure 5.26)

Recherche de cas similaires

Cas traiter

Age_Malade	Sexe_Malade	Poids_Malade	Adresse_Malade	Date_Consultation	Fievre	Asthénie
24	Féminin	65	ALGER	12/06/2007	fort	fréquent

Les cas trouvés

Similarite	Num_Cas	Age_Malade	Sexe_Malade	Poids_Malade	Adresse_Malade
0,77329	7	24	Féminin	74	Hadjout
0,70663	8	19	Féminin	65	Cherchell
0,67451	5	45	Masculin	96	tipaza
0,61443	9	20	Féminin	78	Media
0,61416	6	88	Masculin	78	cherchell
0,60474	4	20	Masculin	75	Chelf

Recherche

Progress bar

Valider l'enregistrement

Annuler

Figure 5.26 : Recherche du cas similaire

Si un cas similaire trouver donc le médecin annule l'ajout du nouveau cas, si non, le médecin clique sur le bouton « Valider l'enregistrement » pour saisir le diagnostic. (Figure 5.27)

The screenshot shows a window titled "Diagnostic" with a table for data entry and three buttons on the right. The table has two columns: "Nom du malade" and "Groupe". The "Groupe" column contains the text "Grippe". Below the table, there are six rows labeled "Medicament 1" through "Medicament 6". The "Medicament 1" row contains the text "Doliprane 1000mg" and the "Medicament 2" row contains "BUTAFEN 200mg". The other rows are empty. To the right of the table are three buttons: "Enregistrer", "Précédant", and "Quitter".

Nom du malade	Groupe
Medicament 1	Doliprane 1000mg
Medicament 2	BUTAFEN 200mg
Medicament 3	
Medicament 4	
Medicament 5	
Medicament 6	

Buttons: Enregistrer, Précédant, Quitter

Figure 5.27 : Saisie du diagnostic

Après avoir remplir le cas correctement, le médecin peut ajouter son cas en appuyant sur le bouton « Enregistrer », la base de cas est mise à jour automatiquement, le médecin peut lui-même vérifier la prise en compte du nouveau cas en lançant la consultation de la base cas.

- **Suppression d'un cas dans la base de cas**

Lorsque le médecin veut faire une suppression d'un cas, il clique sur le bouton «Supprimer un cas», puis sur le bouton «Suivant», (**Figure 5.20**)

Une boîte de recherche s'affiche pour saisir le numéro du cas à supprimer. Si le cas existe, un message de confirmation de suppression de cas s'affiche à l'utilisateur, si l'utilisateur accepte la suppression, le cas est supprimé, sinon la suppression est annulée.

Dans la situation où le cas n'existe pas dans la base, un message d'information s'affiche signalant que celui-ci n'existe pas. (**Figure 5.28**)

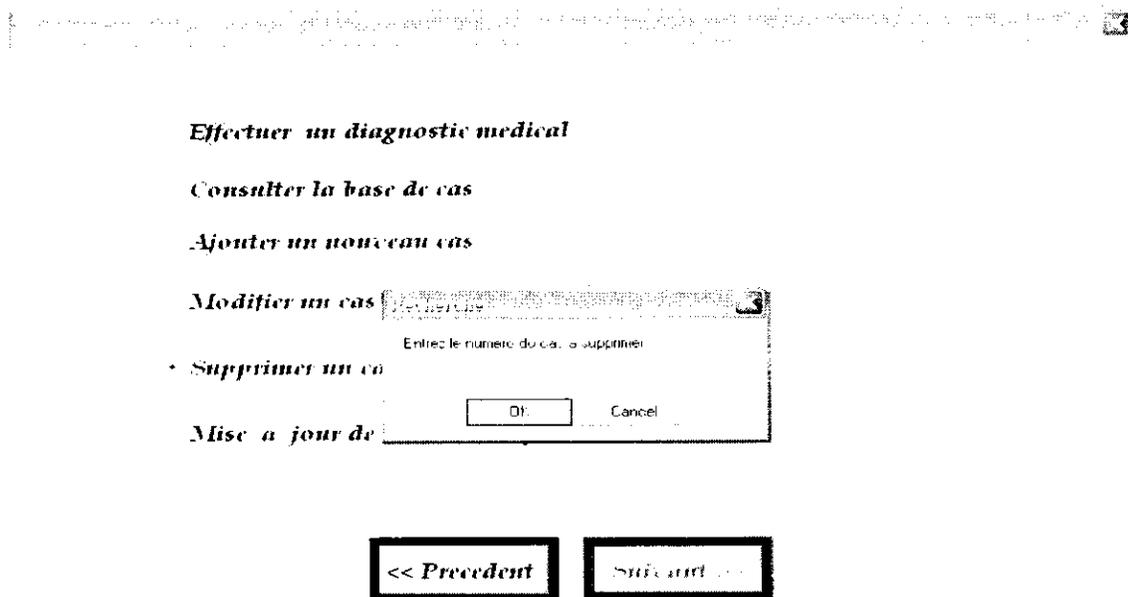


Figure 5.28 : Demande de la suppression d'un cas

- **Modification d'un cas dans la base de cas**

Lorsque le médecin veut faire une modification d'un cas, il clique sur le bouton «Modifier un cas dans la base de cas», puis sur le bouton «Suivant», (**Figure 5.20**)

Une boite de recherche s'affiche pour saisir le numéro du cas à modifier, si le cas existe la fiche de description du cas s'affiche au médecin pour faire les modifications nécessaires et clique sur le bouton « Enregistrer » pour enregistrer les modifications. Dans la situation ou le cas n'existe pas dans la base, un message d'information s'affiche signalant que celui-ci n'existe pas. (**Figure 5.29**)

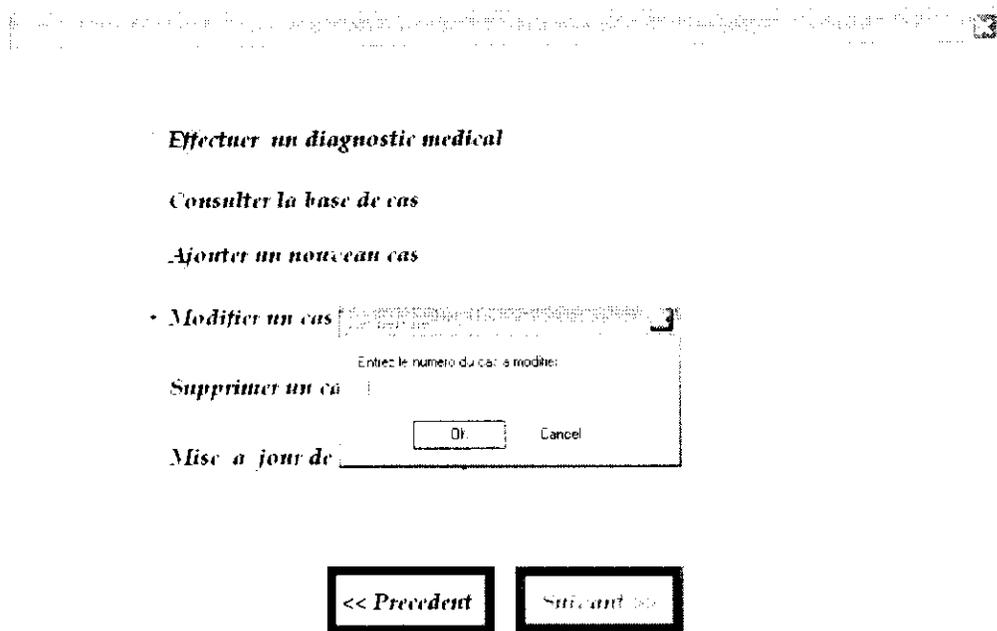


Figure 5.29 : Demande de la modification d'un cas

- **Mise à jour de la table des poids**

Le médecin a la possibilité de mettre à jour la table des attributs intervenant dans le calcul de similitude. Il peut donc modifier les coefficients de pondération des attributs. (Figure 5.30)

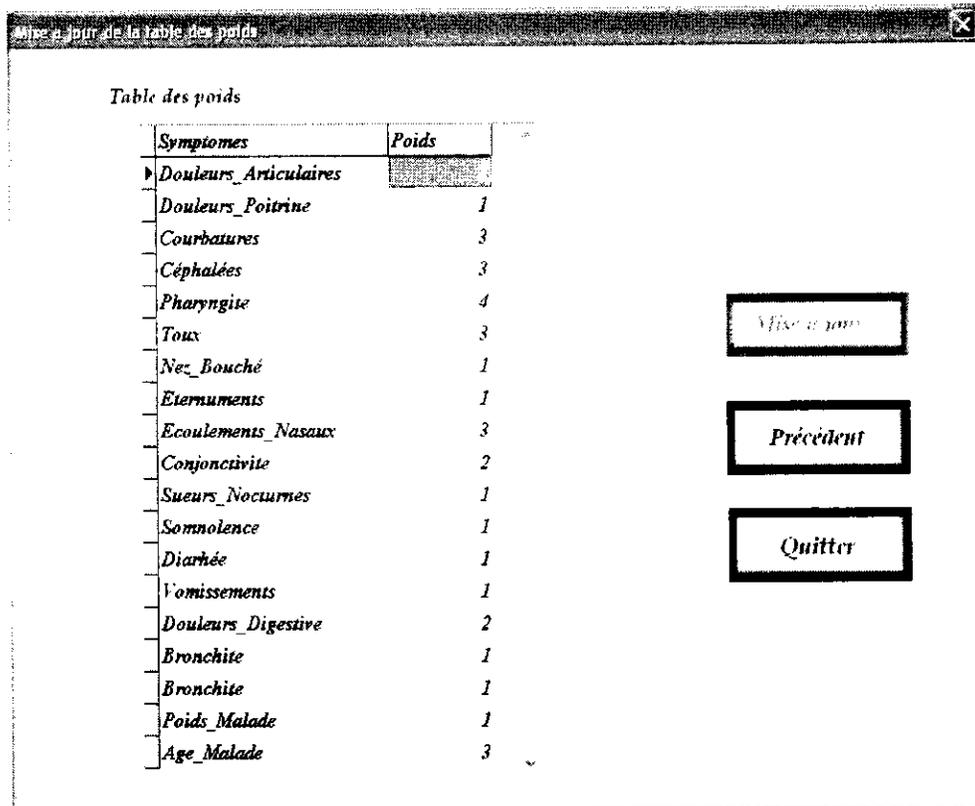


Figure 5.30 : Mise à jour des poids des attributs

V. Test du logiciel

Pour pouvoir confirmer le bon fonctionnement de notre système du diagnostic médical, nous avons élaboré un test, afin de pouvoir vérifier que celui-ci fonctionne correctement, et ne produit pas de faux résultats.

Parmi les cas de la base voici les cas les plus similaires du cas que nous traitons, du plus proche au moins proche.

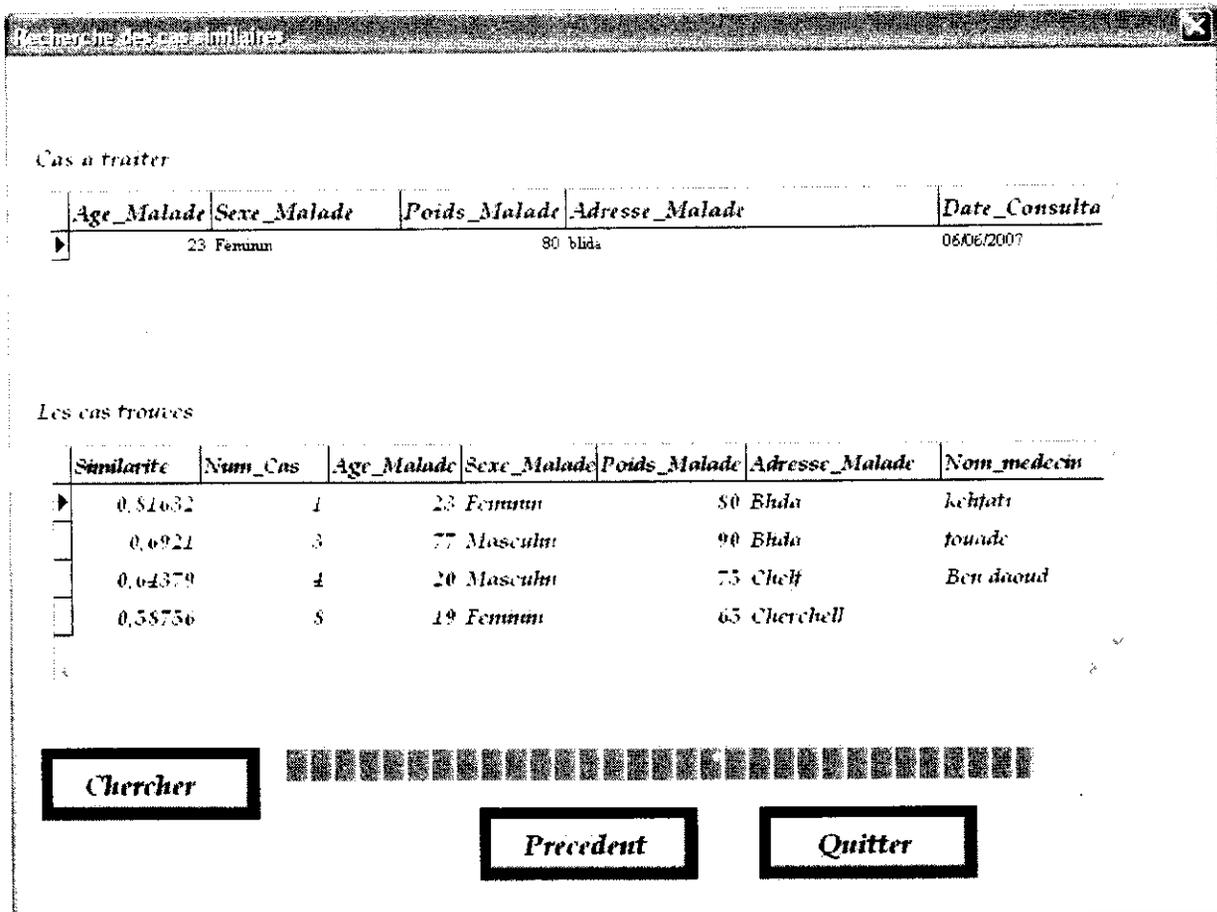


Figure 5.31: Test du logiciel

Résultats et interprétation du test

Suite aux résultats établis par le test, nous avons constaté que le résultat est assez satisfaisant. Ce qui prouve la fiabilité du système.

Chapitre5 : Démarche de développement

Conclusion

Nous avons présenté, au cours de ce chapitre, la mise en œuvre d'un système de raisonnement basé sur les cas flou pour effectuer un diagnostic médical, pour cela on a choisi le formalisme de modélisation UML.

Nous avons commencé la démarche de développement par l'identification des différents besoins de l'utilisateur face au problème de diagnostic médical, puis à la conception du système.

Aussi nous avons élaboré la partie implémentation de notre projet, en donnant un exemple sur la modification, la suppression et l'ajout d'un cas dans la base. Enfin la partie test du logiciel nous a permis de donner un pourcentage quant à la fiabilité du système.

Conclusions et perspectives

Le travail présenté s'inscrit dans le cadre de la modélisation des connaissances appliquée à l'aide au diagnostic des maladies. La méthodologie de résolutions de problèmes choisie s'appuie sur le paradigme RBCF. La structure de ce modèle est un ensemble de connaissances utiles pour le diagnostic. Ces connaissances sont imprécises et incertaines. Le but de notre système est d'obtenir un diagnostic le plus précis possible et avec la plus grande certitude. La structure de la base de cas facilite la recherche des cas similaires au cas d'entrée en sélectionnant le cas le plus probable.

La conception d'un système de raisonnement à base de cas flou est une tâche très délicate et difficile, car la performance d'un tel système dépend de la représentation des connaissances du domaine d'application, des méthodes utilisées pour l'exécution des quatre étapes du cycle du systèmes et surtout que de tels systèmes doivent avoir une grande durée de vie, puisqu'ils sont conçu pour apprendre des expériences du passé, donc de tels systèmes doivent être capables d'identifier des nouveaux cas, d'identifier les cas utiles, d'identifier des nouveaux questions et surtout mettre à jour en permanence sa base de connaissance.

Comme on a déjà dit, ces connaissances sont, souvent, imparfaites (imprécises, incertaines, incomplètes), ce qui pose le problème de leur prise en charge et leur traitement. Plusieurs approches ont été développées, mais celle qui sied le mieux, est sans conteste, la logique floue.

En effet, la logique floue, développée par Zadeh (1965) [4a] [4b], permet de capture l'incertitude et l'imprécision de la pensée humaine et l'exprimer avec des outils mathématiques appropriés ; les ensembles flous pour l'imprécision et la théorie des possibilités pour l'incertitude.

Le concept de logique floue vient de la constatation que la variable booléenne, qui ne peut prendre que deux valeurs (vrai ou faux) est mal adaptée à la représentation de la plupart des phénomènes courants. Alors que la logique classique considère qu'une proposition est soit vraie soit fausse, la logique floue distingue une infinité de valeurs de vérité (entre 0 et 1). Il s'agit donc d'une généralisation de la logique binaire à une logique multivaluée.

Conclusions et perspectives

Une spécification UML (Unified Modeling Language) a été faite pour faciliter le développement d'une interface utilisateur mais aussi pour bien identifier les besoins des utilisateurs. Nous avons choisi UML car ce modèle n'impose pas de méthode de développement, c'est un standard de communication.

De plus, il utilise une approche objet adapté à la programmation objet et au développement itératif qui nous permettra d'ajouter de nouvelles fonctionnalités par la suite. Ce modèle va donc nous aider à avoir une compréhension commune du résultat afin de désambiguïser la conception. Il nous permettra également d'éviter les oublis et bien répondre à nos spécifications.

De nombreuses améliorations sont bien sûr possibles.

Notre système concerne l'aide au diagnostic médical. Nous avons choisi de le valider pour le cas de la grippe, donc la première amélioration qui peut être faite est d'effectuer un diagnostic médical pour d'autres maladies comme diabète, cancer du sein, paludisme, etc.

La deuxième perspective consiste à implémenter l'accès via Internet. Chaque personne voulant effectuer un diagnostic, peut accéder au système. Les bases de cas peuvent être consultées, modifiées via le Web pour les différents acteurs en fonction de leurs droits d'accès sur le système.

Une autre perspective consiste la possibilité de représenter graphiquement les valeurs imprécises et de les rendre modifiables par l'expert du domaine en agissant directement sur la représentation graphique est aussi envisagée.

Introduction:

UML (Unified Modeling Language), ou bien langage de modélisation unifié est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion des plusieurs méthodes existantes auparavant, et est devenu désormais la référence en terme de modélisation objet.

A partir de 1997, UML est devenue une norme de l'OMT (Object Management Group), ce qui lui a permis s'imposer en tant que langage de modélisation pour les méthodes objet et être reconnue et utilisée par de nombreuses entreprises.

I. Concepts et notations :

Nous décrivons dans cette partie les concepts de base de langage UML. Ce dernier est l'acronyme de « Unified Modeling Language », il représente l'état de l'art des langages de modélisation objets.

UML s'appuie sur des concepts, des relations et des diagrammes :

- Des concepts (structurels, comportementaux, annotationnels, groupements).
- Des relations (associations, généralisations, agrégations, compositions...etc.).
- Des diagrammes (statiques et dynamiques)

I.1. Les concepts :

UML supporte quatre types de concepts :

- Les concepts structurels : représentés par les classes, les interfaces, les collaborations...
- Les concepts comportementaux : représentés par les interactions et les états des objets.
- Les concepts annotationnels : représentés par les notes. Une note est un commentaire attaché à un ou plusieurs éléments de modélisation [23].

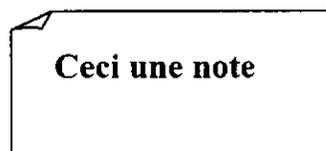
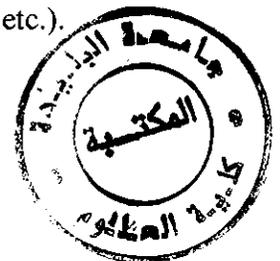


Figure A. 1 : Représentation d'un note

- Les concepts de groupements : représentés par les sous systèmes et les paquetages.

Les paquetages : un paquetage regrouper un ensemble d'objets logiques solidaires du point de vue du développement.



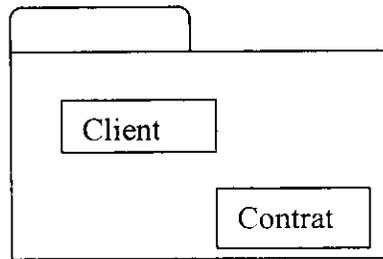


Figure A.2 : représentation d'un paquetage.

1.2. Les relations :

Elles permettent de relier les concepts entre eux. On distingue quatre types de relations, les associations, les généralisations, les compositions...etc.

1.2.1. L'association :

Relation structurelle précisant que les objets d'un élément sont reliés aux objets d'un autre élément.

1.2.2. La généralisation :

La généralisation est une relation entre une classe principale superclasse et les autres classes les sous classe, la notation utilise est le triangle

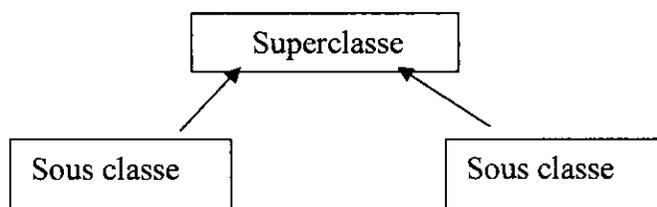


Figure A.3 : représentation d'une généralisation.

1.2.3. L'agrégation :

Représente une association non symétrique dans laquelle une extrémité joue un rôle prédominant par rapport à l'autre extrémité.

1.2.4. La composition :

Relation d'agrégation mettant en avant une notion de propriété forte et de coïncidence des cycles de vie. Les parties sont créées et détruites en même temps que le tout. Elle est distinguée par un losange plein.

I.3. Les diagrammes d'UML :

Le métamodèle UML fournit une panoplie d'outils permettant de représenter l'ensemble des éléments du monde objet (classes, objets, ...) ainsi que les liens qui les relie.

Toutefois, étant donné qu'une seule représentation est trop subjective, UML fournit un moyen astucieux permettant de représenter diverses projections d'une même représentation grâce aux vues.

Une vue est constitué d'un ou plusieurs diagrammes. On distingue deux types de vues:

- **Les vues statiques**, c'est-à-dire représentant le système physiquement
 - diagrammes d'objets
 - diagrammes de classes
 - diagrammes de cas d'utilisation
 - diagrammes de composants
 - diagrammes de déploiement
- **Les vues dynamiques**, montrant le fonctionnement du système
 - diagrammes de séquence
 - diagrammes de collaboration
 - diagrammes d'états-transitions
 - diagrammes d'activités

I.3.1. Diagrammes d'objets :

Ce type de diagramme UML montre des objets (instances de classes dans un état particulier) et des liens entre ces objets.

Le diagramme d'objets s'utilisent principalement pour montre un contexte, par exemple avant ou après une interaction, mais également pour faciliter la compréhension des structures de données complexes, comme les structures récursives.

I.3.2. Diagrammes de classes :

Les diagrammes de classes montrent la structure du système et les éléments des classes tels que : les classes, les relations d'héritages entre classes, les associations, dont les agrégations, les attributs, les opérations et la spécification d'opérations et contraintes au niveau des entités.

I.3.3. Diagrammes de cas d'utilisation :

Un diagramme de cas d'utilisation permet de décrire les interactions entre les acteurs de l'organisation et le système dans chacun des cas d'utilisation envisagés. Il décrit le comportement d'un système au point de vue de l'utilisateur et fixe les limites de système et les relations entre le système et l'environnement [23].

Les divers cas d'utilisation du système vont être présentés dans les diagrammes de cas d'utilisation. Les cas d'utilisation peuvent être liés les uns aux autres par trois types de relations.

- La relation d'utilisation : lorsqu'une ou plusieurs tâches sont utilisées régulièrement, on peut les factoriser dans un même use case et faire de telle sorte que d'autres use cases l'utilisent en le pointant par une flèche [16].

Cet use case est en fait une sous-partie de chaque use case qui l'utilise. Ce qui permet de décomposer un use case complexe en plusieurs use cases.

- La relation d'inclusion : le cas d'utilisation source comprend également le comportement de son cas d'utilisation destination. Cette relation a un caractère obligatoire (à la différence de la généralisation) et permet ainsi de décomposer des comportements partageables entre plusieurs cas d'utilisations différents [16].
- La relation d'extension : le cas d'utilisation source ajoute son comportement au cas d'utilisation destination. L'extension peut être soumise à condition. Cette relation permet de modaliser des variantes de comportement d'un cas d'utilisation [16].

I.3.4. diagrammes de composants :

Composant : élément physique qui représente une partie implémentée d'un système. Il peut être du code, un script, un fichier de commandes, etc. les composants présentent un ensemble d'interfaces [24].

Les diagrammes de composants permettent de décrire l'architecture physique et statique d'une application en terme de modules ; fichier source, librairie, exécutables etc. [13]. Ils décrivent les éléments physiques et leurs relations dans l'environnement de relation, ils montrent les choix de relations [23].

Annexe A. UML

Les composants du système : le sous système, le module, le programme et le sous programme, le processus et la tâche.

- **Module** : un système peut être décomposé en module, chaque module correspond à un ensemble d'éléments physique (fichiers, bibliothèques, sous ensemble de logiciel.) [16].

La figure suivante donne le formalisme d'un module.

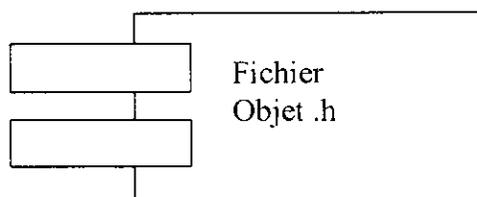


Figure A.4 : Représentation d'un composant

I.3.5. Diagrammes de déploiement :

Les diagrammes de déploiement montrent la disposition physique des matériels qui composent le système et la répartition des composants sur ces matériels. Ils peuvent montrer des classes de nœuds ou des instances de nœuds [23].

Un diagramme de déploiement permet également de représenter les relations entre différents nœuds.

I.3.6. Diagrammes de séquence :

Les diagrammes de séquences permettent de représenter les interactions entre objets en précisant la chronologie des échanges de messages. Ils peuvent être utilisés pour représenter les scénarios d'un cas d'utilisation donné [16].

- **Interactions** : modélisent un comportement dynamique entre objets [23]. Elles se traduisent par l'envoi de messages entre objets. Un diagramme de séquence représente une interaction entre objet, en insistant sur la chronologie des envois de messages [24].
- **Les messages** : les messages échangés sont représentés au moyen de flèches horizontales partant de l'émetteur vers le récepteur. L'ordre de l'envoi est donné par la position sur l'axe vertical [27].

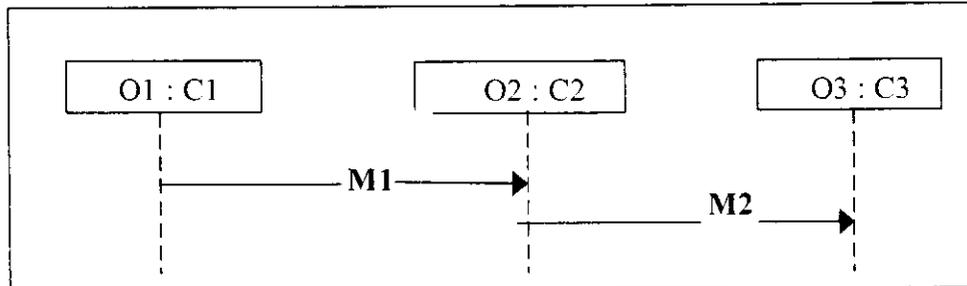


Figure A. 5 : Agencement de messages

Le diagramme de séquence distingue 4 types de messages prédéfinis qui sont présentés dans le tableau suivant :

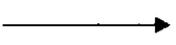
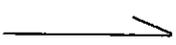
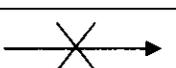
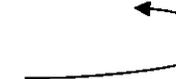
Type de message	La signification
	Message simple, à une passation de contrôle en mono tâche.
	Message synchrone, pas de réponse attendue par l'émetteur.
	Message synchrone, réponse nécessaire du destinataire.
	Message déroband, le destinataire doit être à l'écoute.

Table A.1 : les différents types de messages.

- *Période d'activation* : correspond au temps pendant lequel un objet effectue une action, soit directement, soit par l'intermédiaire d'un autre objet.

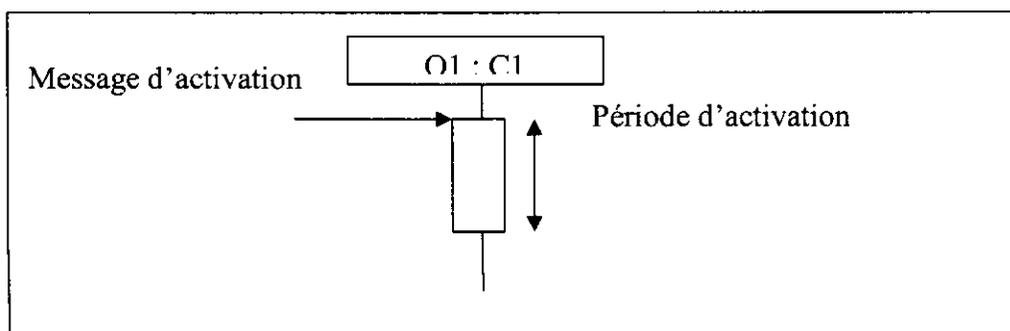


Figure A.6 : activation d'un objet de manière simple.

1.3.7. Diagrammes de collaboration :

Les diagrammes de collaboration montrent des interactions entre les objets et les acteurs. Ils permettent de représenter le contexte d'une interaction, car on peut y préciser les états des objets qui interagissent et peuvent être utilisés pour identifier les principaux objets [8]. La figure suivante présente le formalisme de base d'un diagramme de collaboration : un échange de message entre deux objets.

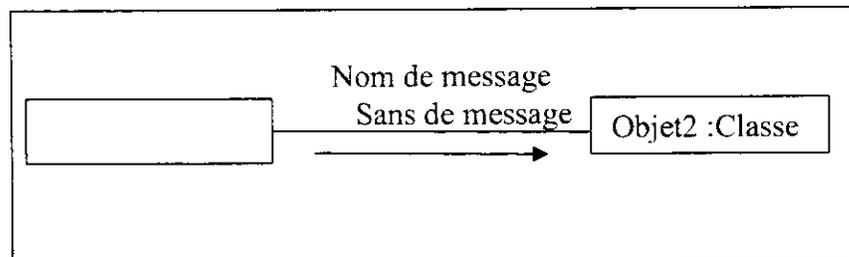


Figure A.7 : Formalisme de base du diagramme de collaboration

1.3.8. Diagramme d'état-transition :

Un diagramme d'état-transition est un graphe constitué de nœuds représentant des états ainsi que des flèches représentant des transitions, portant des paramètres et des noms d'événements [27]. Les diagrammes d'état transition permettent de définir le comportement d'un objet particulier vis-à-vis des sollicitations internes ou externes auxquelles il peut être soumis [16].

1.3.9. Diagrammes d'activités :

Ce diagramme permet de décrire le déroulement d'un cas d'utilisation particulier. Il est possible de décrire les acteurs responsables de chaque activité par l'utilisation des « couloirs d'activités » qui permettent de répartir graphiquement les différentes activités entre les acteurs opérationnels [24]. Chaque activité est placée dans le « couloir » correspondant à l'acteur qui assume cette activité.

Annexe B. Le modèle en cascade

Introduction :

Ce modèle est décrit par Royce en 1970, qui a été largement employé depuis la description générale des activités liées aux logiciels [23].

Le modèle «en cascade» présente un cycle de vie d'un logiciel par une suite de phases ou d'étape (analyse, conception, implémentation, test et validation) (Figure B.1) qui s'enchaînent dans un déroulement linéaire depuis l'analyse des besoins jusqu'à la maintenance [23]. Les résultats de chaque étape sont testés, et on ne passe à l'étape suivante que s'ils sont validés.

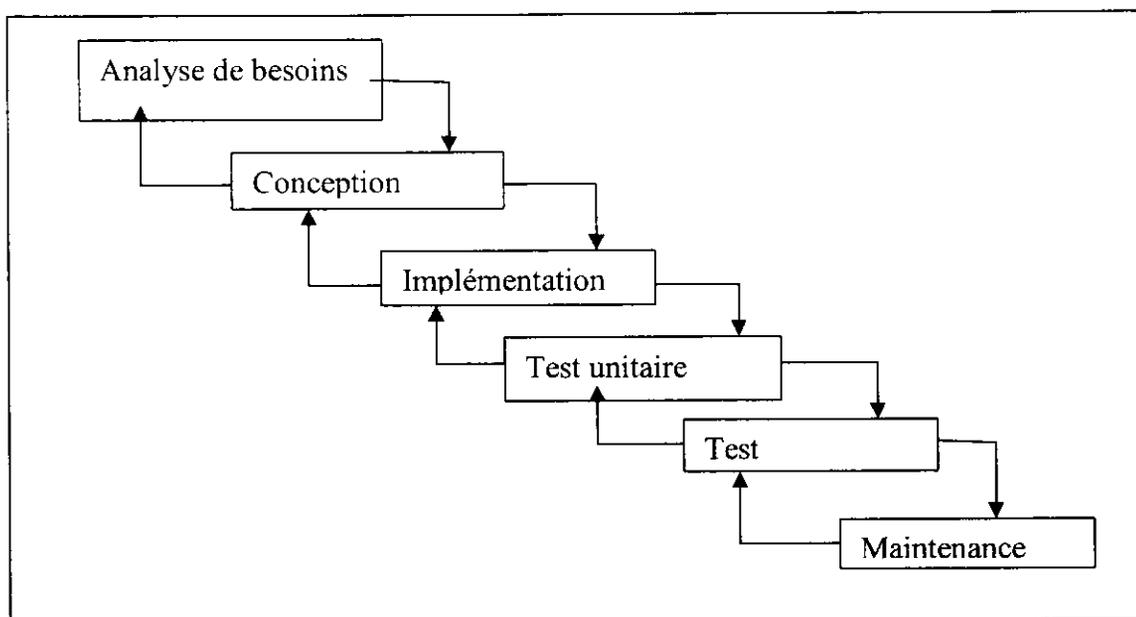


Figure B.1 : le modèle de développement en cascade plus détaillé.

a. Analyse et définition des besoins :

Les fonctionnalités du système et les contraintes sont établies en consultant les usagers (utilisateurs), elles doivent être définies d'une façon compréhensible ce dernier et par l'équipe du développement.

b. spécification et conception :

En partant de l'analyse des besoins, on représente les diverses fonctions du système d'une manière permettant d'obtenir un ou plusieurs programmes réalisant ces fonctions.

Annexe B. Le modèle en cascade

c. réalisation et test unitaire :

On réalise un ensemble d'unités de programmes en langage exécutable. Les tests unitaires permettent de vérifier que ces unités répondent à leurs spécifications.

d. Test de système :

On intègre les unités du programme et on réalise des tests globaux pour être sûr que les besoins ont été satisfaits, le système est alors livré au client.

e. exploitation et maintenance :

C'est la plus longue étape, elle consiste à :

- Corriger les erreurs qui n'ont pas été découvertes lors des étapes précédant.
- Améliorer la réalisation des unités du système, et augmenter ses fonctionnalités au fur et à mesure que de nouveaux besoins apparaissent.

Références

RÉFÉRENCES

- [1] AFOUBA NGAYIHI, KERBRAT, LABARANG « Le raisonnement à partir des cas : Définitions et principes de fonctionnement », Novembre 2004.
- [2] Béatrice Fuchs, « Représentation des connaissances pour le raisonnement à partir de cas », Thèse pour obtenir le titre de Docteur de l'Université Jean Monnet de Saint-Etienne Spécialité informatique, octobre 1997.
- [3] Benjamin DEVEZE, Matthieu FOUQUIN, «CASE-BASED REASONING», Juin 2004
- [4a] Bernadette *Bouchon*-Meunier, « La logique floue et ses applications », Collection Vie Artificielle. Addison-Weiley, 1995.
- [4b] Bernadette *Bouchon*-Meunier, Christophe Marsala « Logique floue, principes, aide à la décision », lavoisier, 2003.
- [5] Bernadi, méthode d'analyse orienter objet UML, DUNOD, 2002.
- [6] Bjarne Kristian Hansen,« Weather Prediction Using Case-Based Reasoning and Fuzzy Set Theory », 2000.
- [7] Bruno TIXIER, « La problématique de la gestion des connaissances », rapport de recherche No 01.9, IRIN, Université de Nantes, Septembre 2001.
- [8] Christine Riou, Pierre Le Beux, Pierre Lenoir, «La représentation des connaissances dans le système d'aide à la décision médicale ADM » Laboratoire d'informatique Médicale, Faculté de Médecine Université de Rennes I, Volume 5 – 1992.
- [9] Suzanne Pinson, « La représentation des connaissances en Intelligence Artificielle », Université Paris-Dauphine, Avril 2001.
- [10] Guillermo CORTES ROBLES, « Management de l'innovation technologique et des connaissances : synergie entre la théorie TRIZ et le Raisonnement à Partir de Cas. Application en génie des procédés et systèmes industriels », Thèse pour obtenir le titre de docteur de l'institut national polytechnique de Toulouse, juillet 2006.
- [11] HAJ SAID Anis, « Distances sémantiques pour la comparaison des connaissances objets dans le cadre du raisonnement à partir de cas », Mémoire de DEA Informatique Théorique et Applications, Université du Havre.
- [12] Haouchine, Brigitte Chebel-Morello et Zerhouni, « Méthode de Suppression de Cas pour une Maintenance de Base de Cas », 14e Atelier de Raisonnement à Partir de cas Besançon (France), mars 2006.
- [13] Icham Sefion, Marc Gailhardou, Abdel Ennaji, « Le Raisonnement à Partir de Cas pour l'aide à la prise en charge de l'asthme », Laboratoire PSI ,Université de Rouen,2002.

Références

- [14] Jean-Louis Ermine, « Enjeux, démarches et processus de la gestion des connaissances »
- [15] Joël Colloc, Laïd Bouzidi, « Utilisation du raisonnement par cas pour élaborer un système d'aide à la décision en médecine », No. 1, Février 2001.
- [16] Joseph Gabay, Merise vers OMT et UML.
- [17] LAURENT Jean-Marc, « KM : KNOWLEDGE MANAGEMENT, Les trois dimensions d'un projet », mars 2005. Disponible sur : <http://ocsima.neuf.fr>.
- [18] Luc Lamontagne – Guy Lapalme. « Raisonnement à base de cas textuels –état de l'art et perspectives ». Université de Montréal, Département d'informatique et de recherche opérationnelle.
- [19a] Luc Lamontagne, « RAISONNEMENT À BASE DE CAS TEXTUEL POUR LA RÉPONSE AUTOMATIQUE AU COURRIER ÉLECTRONIQUE », Proposition de projet doctoral, 16 novembre 2001.
- [19b] Luc Lamontagne, « Une approche CBR textuel de réponse au courrier électronique », Thèse présentée à la faculté des études supérieures en vue de l'obtention du grade de Philosophiae Doctor (Ph.D.) en informatique, Université de Montréal, 2004.
- [20] Magali RUET, « Capitalisation et réutilisation d'expériences dans un contexte multiacteur », Thèse pour obtenir le titre de docteur de l'institut national polytechnique de Toulouse, 2002.
- [21] Malek M, « Un modèle hybride de mémoire pour le raisonnement à partir de cas », Thèse de l'université Joseph Fourier, Grenoble, 1996.
- [22] Michel GRUNDSTEIN, « DE LA CAPITALISATION DES CONNAISSANCES AU MANAGEMENT DES CONNAISSANCES DANS L'ENTREPRISE, LES FONDAMENTAUX DU KNOWLEDGE MANAGEMENT », Université Paris Dauphine, 2003.
- [23] Muller, « modélisation objet avec UML », EYROLLES, 1997.
- [24] Muller, « modélisation objet avec UML », EYROLLES, 2001.
- [25] Muller, « modélisation objet avec UML », EYROLLES, 2002.
- [26] Nicolas Turenne, « Méthode des KNN (k nearest neighbours, ou *k plus proches voisins*) » INRA, 2006.
- [27] Rémy Fannader, Hervé Lerroux, « UML principe de modélisation », DUNOD, 2000.
- [28] Roxane LOPEZ « D'un système de management de la qualité basé sur l'amélioration à un système de management de la qualité basé sur les connaissances », thèse pour obtenir le grade de docteur de L'INPG, 2006.