



THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Développement d'un module d'aide à la décision multicritère dans un système d'aide au diagnostic remedial en médecine homéopathique

Leclef, Frédéric

Award date:
1995

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX
NAMUR
FACULTE DES SCIENCES

**DEVELOPPEMENT D'UN MODULE D'AIDE A LA
DECISION MULTICRITERE DANS UN SYSTEME
D'AIDE AU DIAGNOSTIC REMEDIAL
EN MEDECINE HOMEOPATHIQUE**

Mémoire présenté pour l'obtention du grade
de licencié et maître
en Informatique
par

LECLEF FREDERIC

Promoteur :
J. FICHEFET

Année académique : 1994-1995

Faculté Universitaire N-D De La Paix
Namur
Faculté Des Sciences

**DEVELOPPEMENT D'UN MODULE D'AIDE A LA DECISION
MULTICRITERE DANS UN SYSTEME D'AIDE AU DIAGNOSTIC
REMEDIAL EN MEDECINE HOMEOPATHIQUE**

Résumé. Ce mémoire a pour but d'ajouter une nouvelle méthode d'aide à la décision multicritère dans un logiciel informatique existant : R.A.D.A.R. (Rapid Aid to Drug Aimed Repertorization). Cette méthode est l'Analytic Hierarchy Process de Saaty (A.H.P.), elle permet de choisir le remède approprié au patient en fonction des remèdes susceptibles de guérir les symptômes retenus. L'A.H.P. est présentée ainsi que son implémentation et l'intégration à R.A.D.A.R..

Abstract. This master thesis aims at adding a new method allowing the multicriterion decision in a existing software: R.A.D.A.R. (Rapid Aid to Drug Aimed Repertorization). This method called Analytic Hierarchy Process by T. Saaty (A.H.P.), enables the physicians to choose the appropriate remedy to the patient among all the remedies that can cure the selected symptoms. In this master piece, we introduce the concept A.H.P. as well as its implementation and the integration the R.A.D.A.R..

Promoteur : FICHEFET J.

Mémoire présenté par LECLEF Frédéric

Année Académique 1994-1995

A cet endroit, je voudrais adresser un mot de remerciement à toutes les personnes qui ont contribué à la rédaction de ce mémoire.

Tout d'abord, je tiens à marquer ma gratitude envers mon promoteur, le Professeur J. Fichet pour la liberté qu'il m'a laissée dans la rédaction de ce mémoire.

Je tiens aussi à remercier la société Archibel et le Docteur Storet pour sa collaboration.

Table des matières

Chapitre 1 Introduction.....	8
1.1 Avant-Propos.....	8
1.2 Structure du mémoire.....	8
Chapitre 2 Initiation à l'homéopathie.....	11
2.1 Principes de l'homéopathie.....	11
2.2 Les outils du médecin homéopathe.....	12
2.2.1 Les Matières Médicales.....	12
2.2.2 Les Répertoires.....	13
2.2.3 La consultation du médecin homéopathe.....	13
Chapitre 3 Le Logiciel Radar for Windows 6.1	15
3.1 Introduction.....	15
3.1.1 L'individualisation du malade.....	15
3.1.2 L'individualisation du remède.....	16
3.2 Le système Radar.....	17
3.2.1 La base de connaissance de Radar.....	17
3.2.2 Le système cognitif de Radar.....	19
3.2.3 L'interface de Radar.....	20
3.3 Les méthodes d'aide à la décision.....	20
3.3.1 La méthode "Somme des symptômes".....	20
3.3.1.1 Le symptôme avec intensité.....	20
3.3.1.2 Le symptôme aux degrés limités.....	21
3.3.1.3 La méthode somme des symptômes.....	21
3.3.2 La méthode "Somme des degrés".....	22
3.3.3 La méthode "SPF".....	22
3.3.4 La méthode "VES".....	23
3.4 Conclusion.....	24
Chapitre 4 Développement de la méthode de Saaty.....	26
4.1 Introduction.....	26
4.1.1 Généralités.....	26
4.1.2 Intérêt de l'A.H.P.....	26
4.1.3 Type de problèmes traités.....	27
4.2 Conventions et définitions préliminaires.....	27
4.2.1 Généralités sur les matrices.....	27
4.2.2 La consistance d'une matrice.....	28
4.2.3 Les hiérarchies.....	30
4.2.4 La méthode des puissances.....	33
4.3 La méthode de rangement par paires.....	34
4.3.1 Introduction.....	34
4.3.2 Axiomatique de l'A.H.P.....	34
4.3.2.1 Définitions préliminaires.....	34

4.3.2.2 Les axiomes.....	35
4.3.3 La phase d'évaluation	37
4.3.4 Le choix d'une échelle	38
4.3.4.1 Si on ne connaît pas précisément l'échelle de rapports?	39
4.3.4.2 Choix d'une échelle.....	40
4.3.5 La méthode de Saaty	40
4.3.6 Exemple: estimation du poids de 5 objets.....	41
4.3.7 Utilisation de la méthode de Saaty dans les hiérarchies.....	43
4.3.7.1 La proposition de Saaty.....	43
Chapitre 5 Une application de l'A.H.P.	47
5.1 Introduction.....	47
5.1.1 La décomposition du problème en une hiérarchie	48
5.2.2 Les matrices de comparaisons par paires	49
5.2.3 La synthèse des priorités	53
5.3 Conclusion	54
Chapitre 6 Critiques de l'A.H.P.....	56
6.1 Le background théorique de l'A.H.P.	56
6.2 L'ambiguïté de la question.....	58
6.3 L'échelle à neuf points	59
6.4 La composition hiérarchique.....	60
6.4.1 Le "rank reversal"	60
6.4.1.1 La mesure relative et la mesure absolue.....	60
6.4.1.2 "Rank reversal " avec une mesure relative.....	61
6.4.2 La consistance	62
6.4.2.1 les sources de l'inconsistance.....	62
6.4.2.2 L'indice de consistance	63
6.4.2.3 Correction des jugements inconsistants	63
6.5 La méthode de globalisation des poids	63
6.5.1 La méthode des vecteurs propres (VP)	64
6.5.1.1 Rappel mathématique.....	64
6.5.1.2 Calcul du vecteur propre et de la valeur propre	65
6.5.2 La méthode des moindres carrés logarithmiques (MCL).....	65
6.5.3 La méthode des moindres carrés (MC)	66
6.5.4 La méthode de la programmation linéaire (PL)	66
6.5.5 Comparaison des quatre méthodes : VP,MC,MCL et PL.....	67
6.5.5.1 Au point de vue de l'unicité de la solution	68
6.5.5.2 Au point de vue de la mesure de la consistance.....	68
6.5.5.3 Au point de vue de la stabilité de la solution	69
6.5.5.4 Conclusion	70
Chapitre 7 Implémentation et structuration du problème.....	72
7.1 Introduction.....	72
7.2 Représentation des données	72
7.2.1 Au niveau de la base de données.....	72
7.2.2 Au niveau de la mémoire	73

7.2.2.1 Représentation des Clipboards.....	73
7.2.2.2 Représentation des symptômes sélectionnés.....	74
7.2.2.3 Représentation des remèdes associés à un symptôme.....	75
7.2.2.4 Représentation des résultats	77
7.3 Implémentation du problème (Algorithmes).....	78
Chapitre 8 Critiques par un homéopathe	89
8.1 Quelques règles de raisonnement suivies par le médecin homéopathe lors d'une consultation	89
8.1.1 Les catégories de symptômes.....	89
8.1.2 Le classement des symptômes par le médecin et le choix du remède adéquat	90
8.1.3 Le nombre de symptômes sélectionnés.....	91
8.2 Exemple d'un cas clinique et pondération	91
8.3 Comparaisons avec les autres méthodes d'analyse de Radar.....	93
8.4 Temps de réponses	94
8.5 Conclusions.....	94
Chapitre 9 Présentation des outils utilisés	96
9.1 Introduction.....	96
9.2 Qu'est-ce que XVT-Design ?.....	96
9.3 Les concepts.....	97
9.3.1 Les attributs des objets.....	97
9.3.1.1 La géométrie.....	97
9.3.1.2 Les titres.....	98
9.3.1.3 L'identifiant de la ressource.....	98
9.3.2 Le code de l'interface utilisateur.....	98
Annexes	100
1. Codes des différentes fonctions	100
Bibliographie	116

CHAPITRE 1

INTRODUCTION

Chapitre 1

Introduction

1.1 Avant-Propos

Cherchant à vivre en harmonie avec la nature, ou simplement curieux d'essayer d'autres façons de se soigner, les patients se précipitent depuis quelques années, aux portes des médecins homéopathes.

Voilà pourquoi, dans les années 1980-1981, un vaste projet de recherche sur l'homéopathie fut mis sur pied. Ce projet incluait évidemment des informaticiens. Ceux-ci proposaient la construction de systèmes d'aide permettant de faciliter le processus de prise de décision au sein d'une démarche homéopathique.

Actuellement, on en est arrivé à construire des systèmes informatiques qui permettent aux médecins de gérer les dossiers de leurs patients. Des systèmes aidant le médecin à poser un diagnostic médical se sont également développés tant pour le médecin généraliste que pour le médecin pratiquant l'homéopathie.

L'idée de ce mémoire est d'implémenter une nouvelle méthode qui devrait aider le médecin homéopathe à trouver et à sélectionner rapidement des symptômes dans des répertoires homéopathiques. Par la suite, en fonction des remèdes susceptibles de guérir les symptômes retenus et par l'intermédiaire de la méthode de Saaty le logiciel permettra de choisir le remède approprié au patient. On sait en effet que, même en médecine, il s'agit parfois de faire des choix sur la base de critères donnant des informations contradictoires. On sait qu'il est alors plus fiable de s'appuyer sur plusieurs méthodes d'aide à la décision pour éviter les erreurs.

Cette implémentation sera en fait un ajout à un logiciel informatique existant sous le nom de "Radar For Windows 6.1".

1.2 Structure du mémoire

Le sujet traité par la méthode étant l'homéopathie, nous en rappellerons les grands principes dans le chapitre 2. Une fois le système d'aide à la décision réalisé, il faudra l'intégrer au logiciel existant: "Radar For Windows 6.1" (Rapid Aid to Drug Aimed Repertorization), il est donc important que le chapitre 3 détaille le contenu et le fonctionnement de ce logiciel. Ensuite, la méthode de Saaty et les outils utilisés seront développés dans le chapitre 4. Une application de la méthode sur un cas directement lié au travail du médecin homéopathe sera développée dans le chapitre 5. Le chapitre 6 exposera, quant à lui, une critique de la méthode. On décrira l'implémentation et la structure du problème au chapitre 7. Le chapitre 8 commentera et interprétera les résultats obtenus à partir de cas réels. Les outils utilisés dans ce mémoire seront présentés dans le dernier chapitre.

CHAPITRE 2

INITIATION A L'HOMÉOPATHIE

Chapitre 2

Initiation à l'homéopathie

Nous allons, au fil de ce chapitre, rappeler les grands principes de l'homéopathie ainsi que les outils qui aident le médecin homéopathe dans sa relation au patient.

2.1 Principes de l'homéopathie

Pour découvrir les origines de l'homéopathie, il faut remonter à l'Antiquité. Il appartient à Hippocrate d'établir les principes immortels qui sont à la base de l'homéopathie. Cependant, ce n'est qu'à partir de la fin du XVIII^{ème} siècle que Samuel Hahnemann expérimente de manière systématique la loi de similitude qui régit cette science.

Voici quatre propositions sur lesquelles repose la base de l'homéopathie.

Proposition 1: Loi de Similitude

Le médicament qui, en agissant sur des hommes bien portants, a pu produire le plus de symptômes semblables à ceux de la maladie dont on se propose le traitement, possède réellement aussi, lorsqu'on l'emploie à des doses suffisamment atténuées, la faculté de détruire, d'une manière prompte, radicale et durable, l'universalité des symptômes de ce cas morbide, c'est-à-dire la maladie présente toute entière.

Proposition 2: Individualisation du malade

Toute maladie: "désaccord dynamique de la force qui anime virtuellement le corps de l'homme", est individuelle et se traduit par l'universalité, par l'ensemble des symptômes, qui, tous, ont leur importance, car l'image de la maladie, "c'est la totalité des symptômes qui caractérisent le cas présent."

Il faut donc individualiser, non seulement la maladie, ce qui est le diagnostic habituel, mais la forme qu'elle peut revêtir chez le malade considéré. Chaque forme clinique est en même temps thérapeutique.

Proposition 3: Individualisation du remède

Si la maladie exprime ses caractères par les symptômes qu'elle provoque chez l'homme malade, image de la maladie, le remède exprime ses caractères par les symptômes qu'il provoque chez l'homme sain, contre image du remède.

Proposition 4: Expérimentation sur l'homme sain

Il n'y a pas de moyen plus sûr et plus naturel pour trouver infailliblement les effets propres des médicaments sur l'homme, que de les essayer séparément les uns des autres, à des doses modérées, sur des personnes saines, et de noter les changements qui résultent de là, dans l'état physique et moral, c'est-à-dire les éléments de maladie que ces substances sont capables de produire, car toute la vertu curative des médicaments est fondée uniquement, sur le pouvoir qu'ils ont de modifier l'état de l'homme, et ressort de l'observation des effets qui résultent de l'exercice de cette faculté.

Notons aussi que c'est à Hahnemann que revient le mérite d'avoir montré l'immense avantage qu'on peut avoir à employer des dilutions de substances médicamenteuses et particulièrement, les alcoolatures. Il remarqua que les doses qu'il donnait, bien que petites, aggravaient les symptômes. Or la dose infinitésimale réduit au maximum les effets toxiques du remède administré tout en lui permettant d'agir efficacement à condition que le remède administré de la sorte soit le plus semblable possible.

2.2 Les outils du médecin homéopathe

2.2.1 Les Matières Médicales

Des expérimentations ont permis de récolter des listes de symptômes apparus chez un individu sain, à qui on a administré des substances ou des drogues. Ces listes de symptômes ou pathogénésies sont regroupées de manière alphabétique dans différents ouvrages et forment la Matière Médicale Homéopathique.

Chaque pathogénésie recense un maximum de symptômes induits par la substance ou la drogue sur une personne saine. Les symptômes sont classés alphabétiquement en 31 chapitres correspondant aux différentes parties du corps humain. La plus connue de ces Matières Médicales est celle de T.F.Allen. Cet ouvrage de 12 volumes recense plus de 1500 remèdes.

2.2.2 Les Répertoires

L'accès aux Matières Médicales sur base du remède étant fastidieux, certains auteurs ont eu l'idée de créer des répertoires dont l'accès se ferait sur base du symptôme.

Dans ces répertoires, chaque symptôme référence le ou les remèdes qui l'ont guéri ou qui le contiennent dans leur pathogénésie.

Le plus célèbre des répertoires est certainement celui de J.T.Kent. Il est subdivisé en 31 chapitres. Le nom d'un chapitre est un terme qui permet de préciser un peu le type de symptôme suivant les différentes parties du corps humain (abdomen, back, bladder, chest,...). Chacun de ces chapitres se compose de différentes rubriques correspondant à un symptôme général, classé par ordre alphabétique. Chaque rubrique contient:

- une liste générale des remèdes qui ont produit le symptôme
- une liste des remèdes selon le moment de la journée
- une liste des remèdes selon les circonstances d'aggravation de ce symptôme.

Chaque remède apparaissant dans une liste est affecté d'un degré. Ce degré est le coefficient d'efficacité (de 1 à 4) calculé en fonction de l'importance du remède constatée lors d'expérimentations ou sur des cas cliniques.

2.2.3 La consultation du médecin homéopathe

En fait, le travail de l'homéopathe, c'est l'individualisation du malade et du remède. Ce qui n'est pas simple car le médecin homéopathe doit essayer, en examinant le malade et en le questionnant, de déterminer précisément la totalité des symptômes que présente celui-ci. Cela pour tenter d'identifier, à partir des différents outils décrits ci-dessus (Matières Médicales et répertoires), le remède le plus adéquat à prescrire au malade. C'est pour aider le médecin homéopathe dans cette tâche que la société Archibel développe et commercialise depuis plusieurs années le logiciel "Radar for Windows 6.1".

CHAPITRE 3

LE LOGICIEL RADAR FOR WINDOWS 6.1

Chapitre 3

Le Logiciel Radar for Windows 6.1

3.1 Introduction

L'objectif de "Radar For Windows 6.1" (Rapid Aid to Drug Aimed Repertorization), comme son nom l'indique, est d'aider le médecin homéopathe dans sa recherche d'un remède à administrer à un patient présentant plusieurs symptômes. Pour cela le logiciel offre à l'utilisateur:

- un accès facile aux répertoires et aux Matières Médicales informatisés
- la possibilité de gérer les dossiers de ses patients
- plusieurs méthodes d'analyse des symptômes retenus
- etc...

Comme nous l'avons vu précédemment le travail du médecin homéopathe se réalise en trois étapes qui ont une importance égale dans le processus d'identification du remède à prescrire au malade.

Précisons pour chacune de ces étapes quelle est l'aide apportée par le logiciel Radar.

3.1.1 L'individualisation du malade

La collecte des symptômes auprès du malade ne saurait en aucune façon être assistée par ordinateur. Elle dépend essentiellement de l'expérience que possède le médecin homéopathe pour interroger le malade.

La recherche de ces symptômes dans les répertoires dépend également dans une large mesure de l'expérience de l'homéopathe. Il s'agit ici de sa connaissance du répertoire c'est-à-dire de sa faculté à traduire des symptômes énoncés dans le langage ordinaire du patient en termes techniques existant dans le répertoire.

Néanmoins, le logiciel Radar permet d'accélérer cette recherche dans les répertoires grâce à sa capacité de traitement rapide d'une énorme quantité de données. En proposant une recherche par mots-clé, Radar facilite ici une navigation *rapide dans tout le répertoire* (cfr Figure 1).

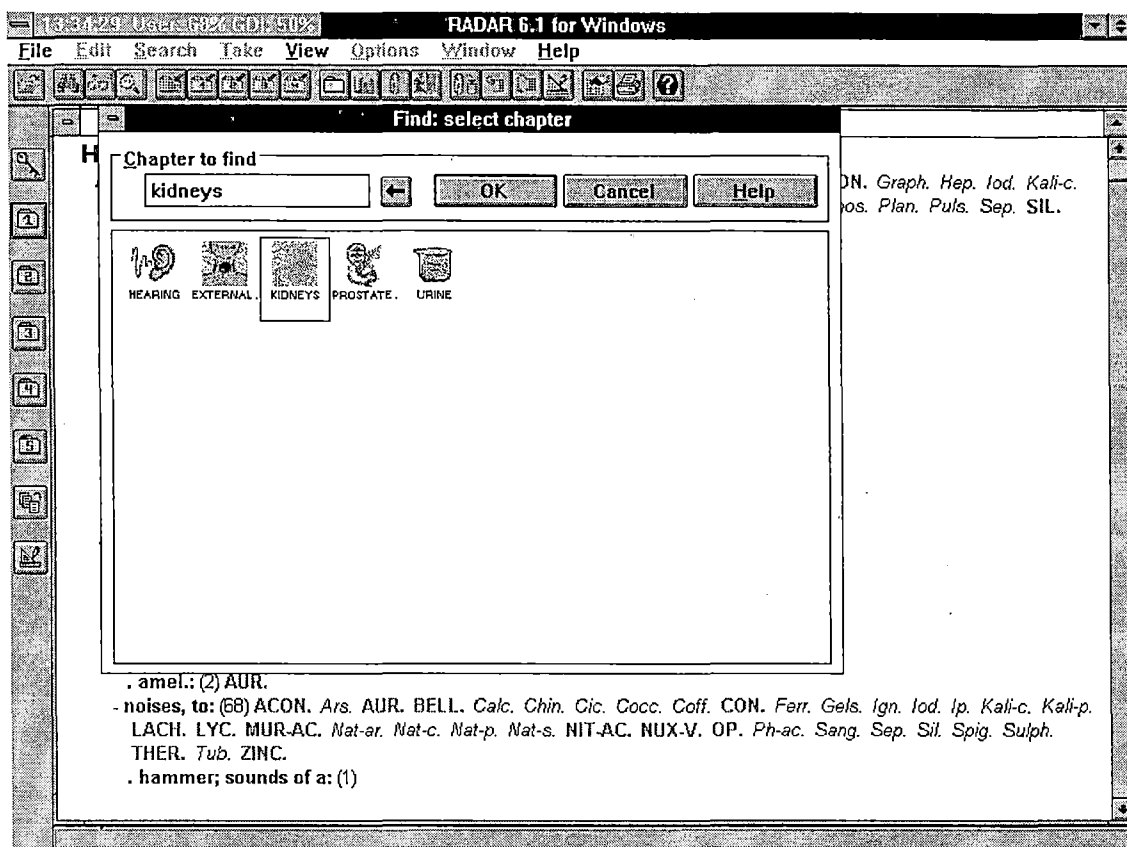


Figure 1 : Choix d'un chapitre du répertoire de Kent

3.1.2 L'individualisation du remède

“Radar For Windows 6.1” apporte une aide précieuse au médecin homéopathe dans sa recherche du remède adéquat à prescrire au patient.

En effet, comment trouver le bon remède quand on se trouve face à une liste considérable de remèdes correspondant aux symptômes collectés par le médecin?

Par sa possibilité de traiter de longues listes de remèdes et d'utiliser des méthodes d'aide à la décision, le logiciel Radar permet d'accélérer le processus de diagnostic remédial et d'ainsi suggérer des solutions au médecin.

3.2 Le système Radar

On peut considérer Radar comme un système expert puisqu'il est composé de trois modules: une interface, un système cognitif et une base de connaissance.

Nous allons décrire brièvement ces trois modules.

3.2.1 La base de connaissance de Radar

Cette base de connaissance repose sur des répertoires homéopathiques dont le plus connu est le Répertoire de Kent.

Elle représente l'ensemble des symptômes codifiés selon les répertoires homéopathiques. Elle est décomposée en chapitres (Mind, Vertigo, Head, ...) chaque symptôme étant éventuellement précisé par sa latéralité (côté gauche ou droit du corps) et par le moment de son apparition dans la journée (cfr Figure 2).

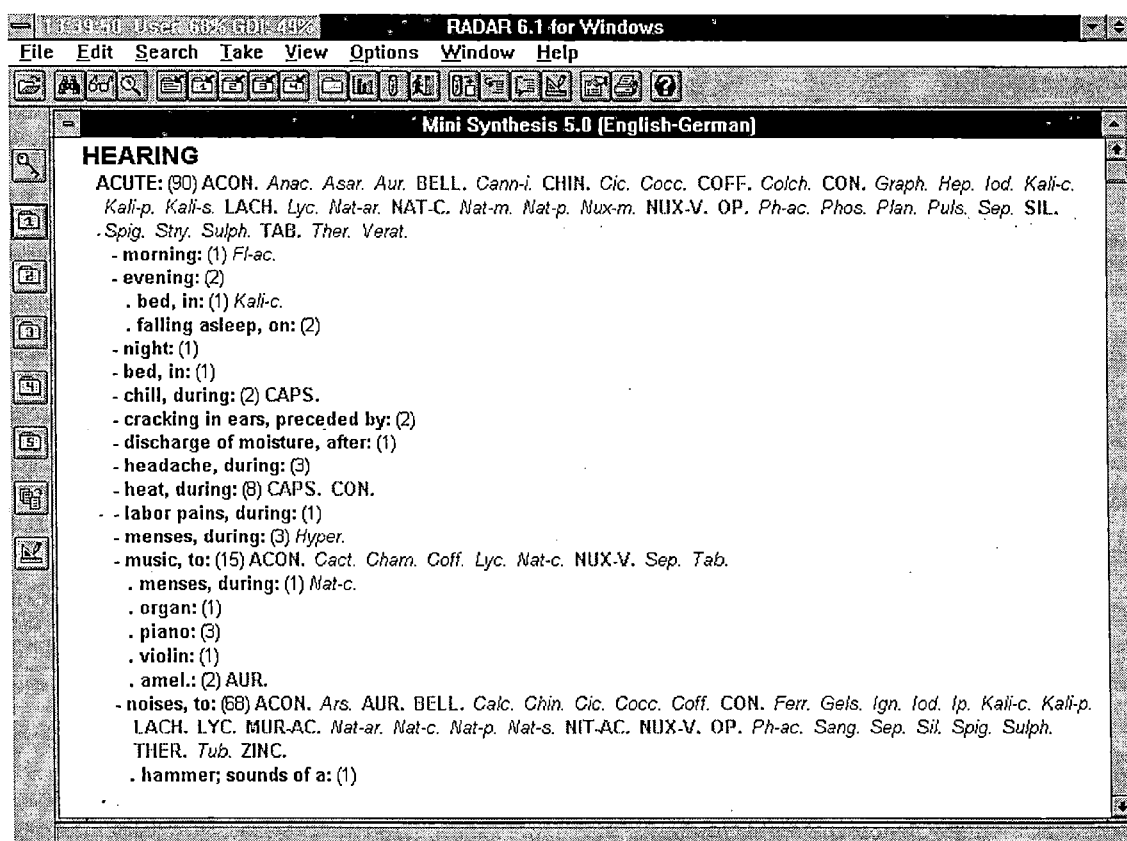


Figure 2 : Extrait du répertoire de Kent

Cette base de connaissances est organisée en une hiérarchie (cfr Figure 3).

En partant du chapitre "*vertigo*", on peut trouver la rubrique "*reading*" qui peut elle même être encore précisée, par exemple: *vertigo/reading/while/aloud* qui signifie globalement: "vertiges quand on lit à haute voix".

L'utilisateur peut évidemment s'arrêter en cours de route dans le cheminement de la hiérarchie à condition que le symptôme sélectionné possède des remèdes associés.

A chaque feuille ou élément de la hiérarchie est associé une liste de remèdes qui sont connus pour avoir guéri le symptôme correspondant dans des cas réels de maladie, ou pour avoir produit ce symptôme à l'occasion d'expérimentations sur l'individu sain. Les listes de remèdes deviennent évidemment de plus en plus courtes chaque fois que l'utilisateur descend d'un niveau dans la hiérarchie. Chaque remède apparaissant dans une liste est affecté d'un degré. Ce degré est le coefficient d'efficacité calculé en fonction de l'importance du remède constatée lors d'expérimentations ou sur des cas cliniques.

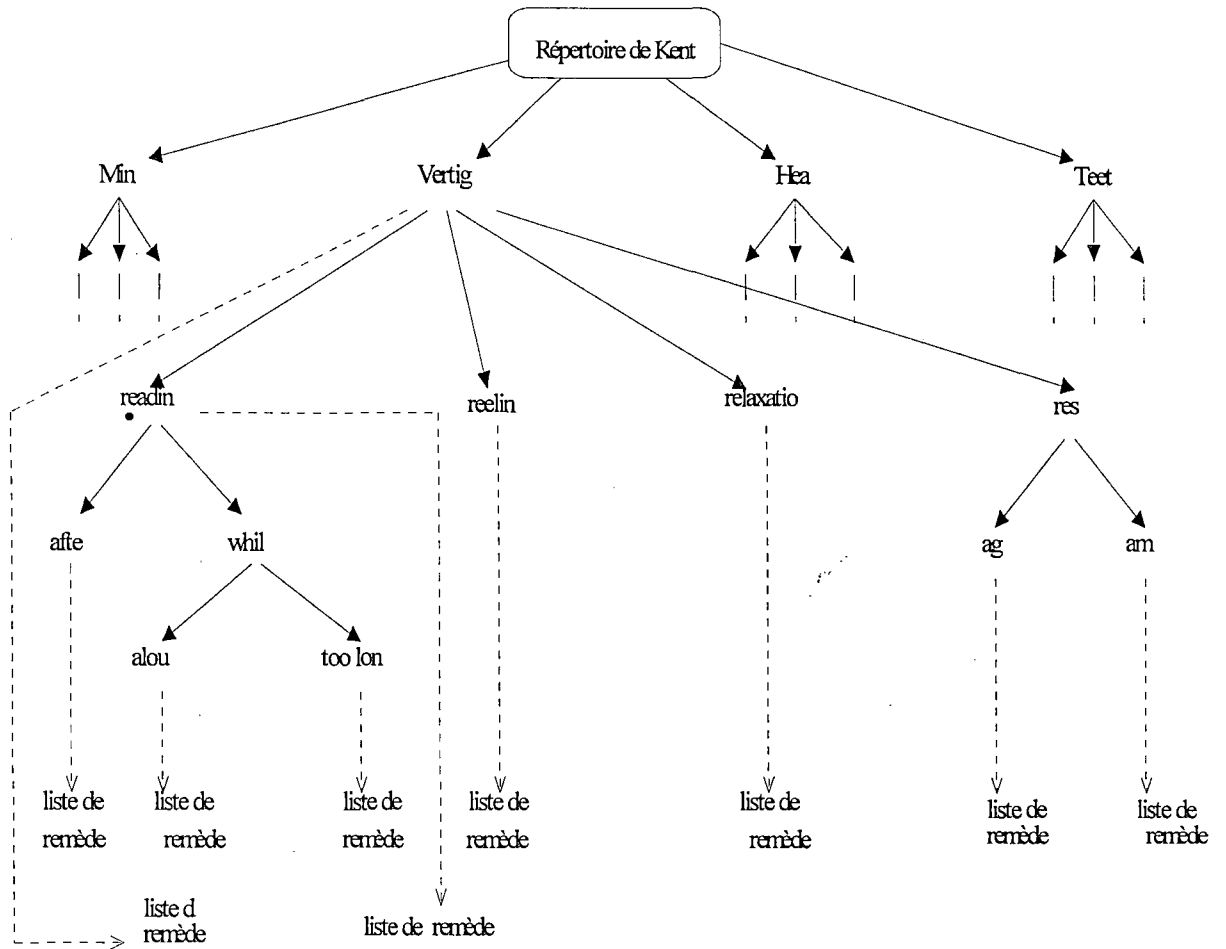


Figure 3 : Hiérarchie du répertoire de Kent

3.2.2 Le système cognitif de Radar

Le système cognitif contient le mécanisme de raisonnement qui permet de répondre aux questions ou de résoudre le problème posé par l'utilisateur.

L'information qui arrive au système cognitif par l'intermédiaire de la base de connaissances peut se résumer à un tableau à double entrée: *Symptômes observés chez le patient/Remèdes susceptibles de faire disparaître ces symptômes.*

Actuellement Radar présente plusieurs méthodes d'aide à la décision.

3.2.3 L'interface de Radar

L'interface d'un système expert est le module qui fournit la communication homme/machine et permet l'acquisition des données.

Dans Radar, la saisie des données revient en fait à demander à l'homéopathe d'introduire les symptômes qu'il a observés chez son patient. Quant à l'aspect communication homme/machine, disons simplement que l'utilisateur peut enregistrer des symptômes, modifier une liste de symptômes qui avaient été retenus, appeler une méthode d'aide à la décision, etc.

3.3 Les méthodes d'aide à la décision

Le logiciel Radar implémente plusieurs méthodes différentes pour traiter les listes de remèdes correspondant aux symptômes sélectionnés et suggérer des solutions au médecin. Les résultats de chacune d'elles sont présentés dans un tableau reprenant le nom des symptômes sélectionnés ainsi que la liste de tous les remèdes (cfr Figure 4).

Quelques unes des méthodes d'analyses sont détaillées ci-dessous.

3.3.1 La méthode "Somme des symptômes"

Radar permet d'ajouter une ou plusieurs qualifications aux symptômes sélectionnés. Nous allons définir deux qualifications retenues pour la méthode "Somme des symptômes".

3.3.1.1 Le symptôme avec intensité

Par défaut, tout symptôme a une intensité égale à un. Néanmoins on peut lui attribuer une intensité supérieure, tout au plus une intensité égale à dix.

Cela permet de marquer l'importance qu'on attache à tel ou tel symptôme. En effet, un symptôme avec une intensité égale à cinq, par exemple, sera considéré comme étant repris cinq fois. Cela aura notamment comme conséquence que les remèdes couvrant ce symptôme, au lieu d'être considérés comme couvrant un symptôme de plus, seront considérés comme en couvrant cinq de plus.

On peut également attribuer une intensité égale à zéro. Dans ce cas le symptôme ne sera pas considéré.

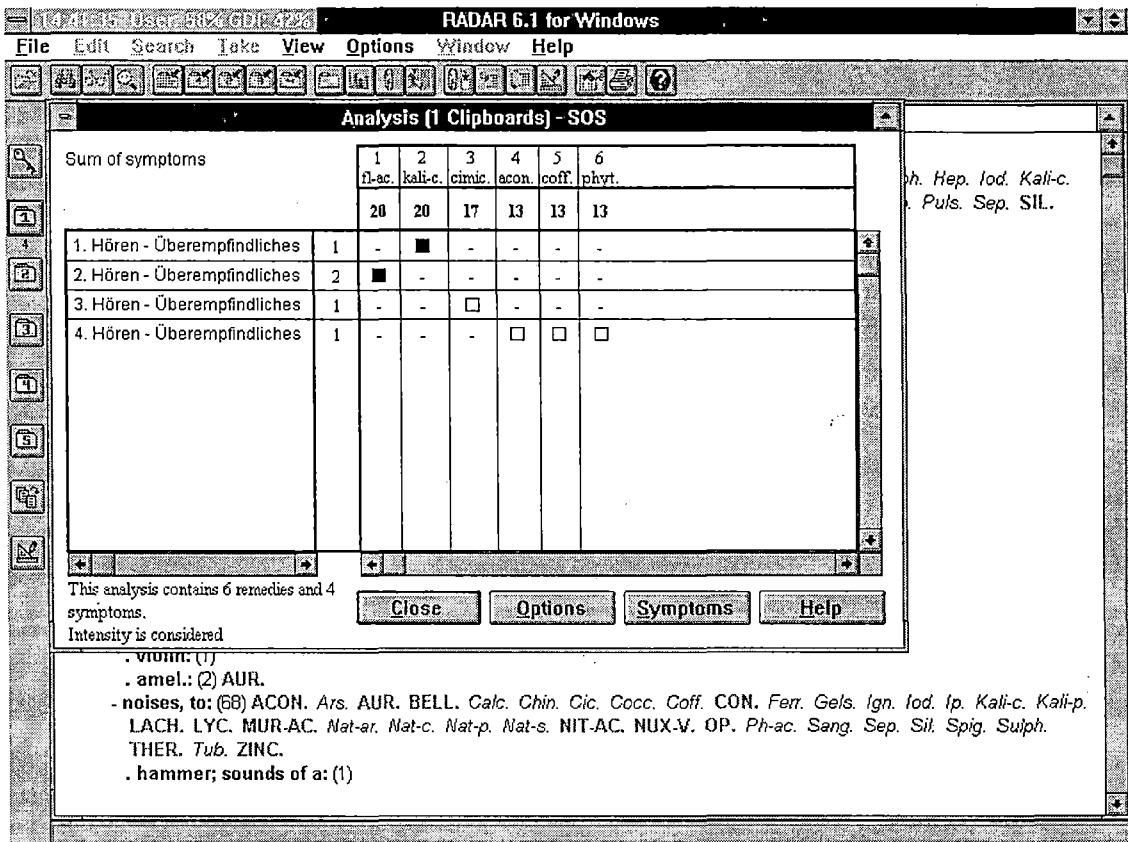


Figure 4 : Tableau des résultats des analyses

3.3.1.2 Le symptôme aux degrés limités

Lors des analyses, Radar considère normalement tous les remèdes se trouvant dans la liste de remèdes d'un symptôme, peu importe leur degré.

Cependant, puisque le degré d'un remède est lié à l'intensité avec laquelle se manifeste le symptôme provoqué par celui-ci, pour tenir compte de l'intensité avec laquelle un patient présente un symptôme, le logiciel Radar permet que ne soient considérés, dans l'analyse, que les remèdes ayant un des degrés donnés.

3.3.1.3 La méthode somme des symptômes

Dans cette méthode on calcule deux couples de nombres pour chaque remède apparaissant dans le classement fourni.

Le premier couple donne le nombre de symptômes couverts par le remède et la somme des degrés avec lesquels il couvre ceux-ci.

Le deuxième couple de nombres donne la même chose que le premier, tout en tenant compte des qualifications d'intensité et de limitation de degré.

Le classement qui se base sur ce deuxième couple de nombres sera effectué dans l'ordre décroissant du nombre de symptômes couverts. Quand il y a égalité, les remèdes égaux sont donnés par ordre alphabétique.

3.3.2 La méthode "Somme des degrés"

Cette méthode suit le même processus que la précédente. Le classement, quant à lui, sera effectué dans l'ordre décroissant du résultat de la somme des degrés des symptômes couverts.

En plus des deux couples de nombres, le classement fournit aussi, pour chaque remède, les symptômes couverts et le degré avec lequel ils le sont.

3.3.3 La méthode "SPF"

"SPF" sont les initiales pour "Samuel¹ Pondération Factor". Le but de cette méthode est de favoriser les remèdes se trouvant dans des listes comprenant peu de remèdes.

La raison d'être de cette méthode vient du fait qu'on tente d'appliquer le principe d'individualisation du malade qui veut que comme chaque patient réagit différemment face à une même maladie, l'on fasse spécialement attention aux symptômes caractéristiques.

Or, si l'on n'essaye pas de favoriser les petits symptômes, c'est-à-dire les symptômes pour lesquels la liste de remèdes en contient peu, les remèdes pour ceux-ci risquent d'être noyés dans la masse des autres remèdes alors qu'ils méritent d'être mis en évidence, vu qu'ils provoquent le symptôme caractéristique qui a été examiné chez le patient.

¹ Samuel est une association hollandaise créée en 1983 pour stimuler le développement et l'utilisation de l'informatique en homéopathie.

La méthode "SPF" utilise dès lors une formule sophistiquée qui, en tenant compte des tailles des listes de remèdes des symptômes analysés, calcule un score pour chaque remède en avantageant de façon pondérée ceux présents dans les listes de plus petite taille. Elle fournit alors un classement de remèdes par ordre décroissant du score obtenu.

Outre ce score, ce classement contient les mêmes informations que celles fournies avec les deux méthodes précédentes.

3.3.4 La méthode "VES"

"VES" sont les initiales pour "Vithoukas² Expert System". L'analyse des symptômes par cette méthode a pour but de fournir un remède, considéré comme étant le plus probable, sur base de la logique et de l'expérience de Vithoukas.

Cette méthode est un système de répertorisation qui possède les caractéristiques suivantes:

-tout comme n'importe quel système de répertorisation il se base exclusivement sur les symptômes proposés par le médecin

-la phase de répertorisation proprement dite est constituée d'un système mathématique qui est basé tant sur les répertoires et Matières Médicales que sur l'expérience clinique de Monsieur Vithoukas. C'est-à-dire que le système contient toute une technique de valorisation

- des différents symptômes d'après leur localisation
- des remèdes suivant qu'ils sont universels ou non.

Il est préférable que le médecin homéopathe utilise cette méthode en interaction avec le patient. C'est-à-dire qu'il ne doit pas attendre d'avoir recueilli tous les symptômes que présente celui-ci avant de lancer l'analyse. Car VES est conçu aussi pour aider à poser des questions au patient, pendant la consultation. En effet, cette méthode fournit comme résultat, un remède désigné comme étant le plus probable *pour l'instant*. Son score est un taux de confiance.

Outre ce remède, elle en fournit un ou plusieurs autres sur lesquels elle conseille d'enquêter (le premier au sujet duquel il faut enquêter, étant, bien entendu, celui donné comme étant le plus probable *pour l'instant*). Ces enquêtes doivent conduire à la sélection d'autres symptômes et à la reconsidération de ceux déjà mis en mémoire.

² Vithoukas est un médecin homéopathe grec, éminent au niveau mondial et possédant une très grande expérience clinique.

Finalement, à force d'investigations guidées par le système expert, on en arrive à ce que le remède, désigné comme le plus probable, ait un très bon score et un taux de confiance supérieur à 100 %. Bien souvent dans ce cas, il n'y a plus de remèdes proposés comme sujet d'investigation et le remède ayant le meilleur score est indiqué comme étant le plus probable et non plus, comme étant le plus probable *pour l'instant*. C'est cependant à l'homéopathe, que revient la décision finale de prescrire le remède proposé ou non.

3.4 Conclusion

Les quatre méthodes ci-dessus ne sont que quatre exemples parmi d'autres. Le chapitre suivant présente en détail une nouvelle méthode, l'**Analytic Hierarchy Process** dont l'implémentation est l'objet de ce mémoire.

CHAPITRE 4

DEVELOPPEMENT DE LA METHODE DE SAATY

Chapitre 4

Développement de la méthode de Saaty

4.1 Introduction

4.1.1 Généralités

Le problème principal dans un processus de prise de décision est de choisir la meilleure alternative en tenant compte de critères parfois contradictoires. En effet, la plupart d'entre nous éprouvent un certain nombre de difficultés pour examiner plusieurs idées en même temps. **L'Analytic Hierarchy Process (A.H.P.)** est un outil qui va nous permettre de résoudre de tels problèmes.

Cette méthode d'aide à la décision structure un problème complexe en une hiérarchie. Celle-ci est composée de plusieurs niveaux spécifiques: le niveau supérieur définit le but global, c'est-à-dire l'objet du problème, le niveau inférieur énumère l'ensemble des alternatives de décision possibles et les niveaux intermédiaires représentent des politiques, des acteurs, etc. Le nombre de paliers et leur dénomination dépendent de la complexité du problème et du niveau de détail souhaité.

4.1.2 Intérêt de l'A.H.P.

L'A.H.P. cherche par des calculs de priorités à quantifier les alternatives de décision pour faciliter la sélection de l'une d'elles. Ces calculs sont basés sur des jugements que fait le décideur à travers la hiérarchie.

Le résultat est un ordre de classement des priorités des alternatives de décision qui aide à déterminer la meilleure solution pour notre problème.

L'avantage principal de l'A.H.P. est donc la fiabilité puisque les décisions sont prises sur base d'un calcul mathématique logique et non plus sur base de l'intuition qui résulte d'une technique trop simple.

De plus, l'A.H.P. est capable de calculer aussi des priorités pour des critères abstraits ou intangibles, c'est-à-dire pour lesquels il n'existe pas de standards de mesure. C'est le cas notamment pour les situations nouvelles, celles dont on ne connaît pas le fonctionnement a priori. Exemple de critère intangible: la satisfaction.

4.1.3 Type de problèmes traités

L'Analytic Hierarchy Process est utilisé pour résoudre des problèmes complexes qui ne se limitent pas à une décision simple mais prennent en considération plusieurs critères à la fois; ce type de problème est dit multi-critères. Toutes les dimensions, tous les attributs, tous les buts de la situation doivent être pris en compte pour structurer le problème.

Cette démarche est très logique. En effet, quand il faut choisir la bonne solution, quoi de plus normal que de la comparer avec les autres, sous tous les critères jugés importants? C'est ce que fait l'A.H.P. et nous allons tenter de l'expliquer.

4.2 Conventions et définitions préliminaires

4.2.1 Généralités sur les matrices

Dans ce paragraphe, nous allons développer certaines parties de la théorie des matrices qui seront nécessaires pour la compréhension de la méthode de Saaty.

Dans la suite de ce chapitre, P désignera une matrice carrée $m \times m$ dont les éléments p_{ij} sont réels.

Définition 4.1

La matrice P est *positive* ($P > 0$) si $p_{ij} > 0$ pour tout $i, j = 1, \dots, m$.

Définition 4.2

Les vecteurs pour lesquels $Px = \lambda x$ ($\lambda \in \mathfrak{R}$, $x \neq 0$) sont appelés *vecteurs propres* et les nombres λ correspondant à ces valeurs propres sont appelés *valeurs propres* de la matrice P .

Définition 4.3

Soient $\lambda_1, \lambda_2, \dots, \lambda_m$ les valeurs propres d'une matrice P . La valeur propre λ_i est une *valeur propre dominante* de P si $|\lambda_i| \geq |\lambda_j|$ pour tout $j \neq i$.

Théorème 4.1 (de Perron)

Toute matrice positive possède une valeur propre λ_{\max} dominante, réelle, positive, de multiplicité 1 dont le module est strictement supérieur aux modules des autres valeurs propres de P . De plus, le vecteur propre w correspondant à λ_{\max} est à composantes positives $w_i > 0$ (pour tout $i = 1, \dots, m$).

Définition 4.4

Une matrice P est *réciproque* lorsque: $p_{ij} = 1/p_{ji}$ pour tout $i, j = 1, \dots, m$.

Remarque:

Les éléments diagonaux d'une matrice positive et réciproque sont tous égaux à 1.

Théorème 4.2

Si la matrice est positive et réciproque et si λ_{\max} est sa valeur propre dominante, alors $\lambda_{\max} \geq m$.

4.2.2 La consistance d'une matrice

Définition 4.5

Toute matrice P telle que: $p_{ij}p_{jk} = p_{ik}$ pour tout $i, j, k = 1, \dots, m$ est une matrice *cardinalement consistante*.

Exemple:

Voici une matrice $P = \begin{pmatrix} 1 & 5 & 6 & 7 \\ 1/5 & 1 & 6/5 & 7/5 \\ 1/6 & 5/6 & 1 & 7/6 \\ 1/7 & 5/7 & 6/7 & 1 \end{pmatrix}$ cardinalement consistante.

Remarques sur les matrices cardinalement consistantes:

- ✓ Les éléments diagonaux d'une matrice cardinalement consistante sont tous égaux à 1 puisque par définition une matrice cardinalement consistante est une matrice telle que: $p_{ij}p_{jk} = p_{ik}$ pour tout $i, j, k = 1, \dots, m$. Pour la diagonale on a donc pour $i = j$ $p_{ii}p_{ik} = p_{ik}$ ainsi $p_{ii} = 1$.
- ✓ Une matrice cardinalement consistante P est entièrement connue dès que l'on connaît l'une de ses lignes, soit $(p_{i1}, p_{i2}, \dots, p_{ij}, \dots, p_{im})$. En effet, pour $j \neq i$ on a $p_{ij}p_{j1} = p_{i1}$, $p_{ij}p_{j2} = p_{i2}$, etc (c'est la formule d'une matrice cardinalement consistante pour $k = 1, 2, \dots$). Par conséquent $p_{j1} = p_{i1}/p_{ij}$, $p_{j2} = p_{i2}/p_{ij}$, ..., $p_{jk} = p_{ik}/p_{ij}$, ..., $p_{jm} = p_{im}/p_{ij}$.
- ✓ Une matrice cardinalement consistante est réciproque car $p_{ij}p_{jk} = p_{ik}$, pour tout $i, j, k = 1, \dots, m$. Donc pour $k = i$ on a $p_{ij}p_{ji} = 1$ c'est-à-dire $p_{ij} = 1/p_{ji}$.

Théorème 4.3

Les propositions suivantes sont équivalentes:

- ✓ la matrice P réciproque et positive est cardinalement consistante;
- ✓ la valeur propre dominante de la matrice P réciproque et positive est telle que: $\lambda_{\max} = m$.

Théorème 4.4

Soient P une matrice positive et réciproque, λ_{\max} sa valeur propre dominante et w le vecteur propre correspondant. Alors pour tout i et $j \in \{1, \dots, m\}$ on a que si $p_{ik} > p_{jk}$ pour tout $k = 1, \dots, m$ alors $w_i \geq w_j$.

Définition 4.6

La valeur de l'indice de consistance se calcule grâce à la formule suivante:

$$I.C. = \frac{\lambda_{\max} - m}{m - 1}.$$

Remarque:

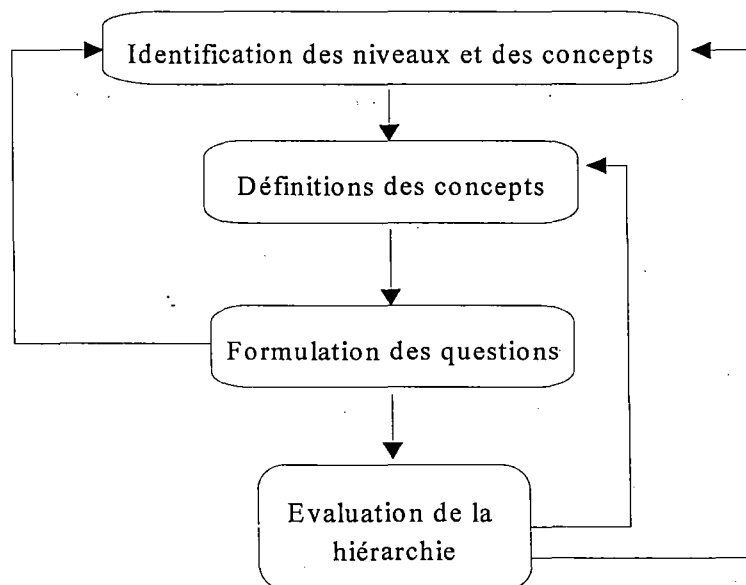
Plus la valeur de l'indice de consistance est proche de zéro plus on est proche de la consistance cardinale.

4.2.3 Les hiérarchies

Une hiérarchie est un type particulier de système basé sur le fait que les entités identifiées peuvent être groupées dans des ensembles disjoints. Les entités d'un groupe n'influencent que les entités d'un unique autre groupe et ne sont influencées que par un seul groupe. Les éléments dans chaque groupe (aussi appelé niveau de la hiérarchie) sont supposés être indépendants.

La conception d'une hiérarchie nécessite une certaine expérience et connaissance de l'ensemble du problème. Deux preneurs de décision obtiendraient normalement deux structurations hiérarchiques différentes d'un même problème. Ainsi une hiérarchie n'est pas unique.

La phase de conception d'une hiérarchie implique trois processus non séquentiels et reliés entre eux: l'identification des éléments et des niveaux, la définition des concepts et la formulation des questions. Le dessin ci dessous résume les relations entre ces trois composantes du design d'une hiérarchie.



Dans la première étape, les niveaux et les concepts sont identifiés. Ils sont ensuite définis et utilisés dans la phase de formulation des questions. Si le décideur éprouve des difficultés pour répondre à ces questions, alors les concepts

et les niveaux sont revus et modifiés. La conception d'une hiérarchie est un processus itératif au cours duquel les concepts, les questions et les réponses déterminent les éléments de chaque niveau de la hiérarchie.

Introduisons d'abord le concept d'ensemble ordonné que nous utiliserons pour définir une hiérarchie.

Définition 4.7

Un ensemble *partiellement ordonné* est un ensemble S muni d'une relation binaire \leq qui satisfait les conditions suivantes:

- (i) réflexivité: pour tout $x \in S: x \leq x$
- (ii) transitivité: pour tout $x, y, z \in S$ si $(x \leq y)$ et $(y \leq z)$ alors $x \leq z$
- (iii) antisymétrie: pour tout $x, y \in S$ si $(x \leq y)$ et $(y \leq x)$ alors $x = y$.

Les ensembles partiellement ordonnés comprenant un nombre fini d'éléments peuvent être représentés par un graphe orienté.

Définition 4.8

inversible, mais pas l'inverse

Soit (A, \leq) un ensemble partiellement ordonné.

- pour tout $x, y \in A: x \leq y \Leftrightarrow x > y \wedge \neg(x = y)$
- y couvre $x \Leftrightarrow (y > x) \wedge \neg(\exists z \in A \text{ tel que } x < z < y)$.

Notations:

Si (A, \leq) est un ensemble partiellement ordonné,

pour tout $x \in A: x^- = \{ y \in A \text{ tel que } x \text{ couvre } y \}$

pour tout $x \in A: x^+ = \{ y \in A \text{ tel que } y \text{ couvre } x \}$.

plus haut de la hiérarchie

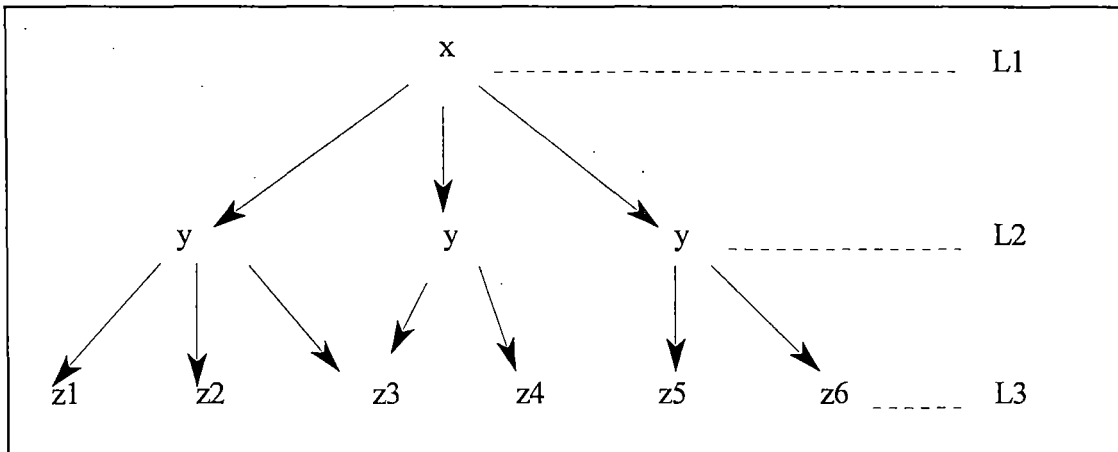
Définition 4.9

Soit H un ensemble fini partiellement ordonné dont le suprémum est l'élément $b \in H$. On dit que (H, \leq) est une *hiérarchie* s'il satisfait aux conditions:

- (i) Il existe une partition de H en ensembles $L_k, k = 1, \dots, h$ où $L_1 = \{b\}$;
- (ii) $x \in L_k \Rightarrow x^- \subset L_{k+1}, k = 1, \dots, h-1$;
- (iii) $x \in L_k \Rightarrow x^+ \subset L_{k-1}, k = 2, \dots, h$.

Les ensembles L_k sont appelés les niveaux de la hiérarchie.

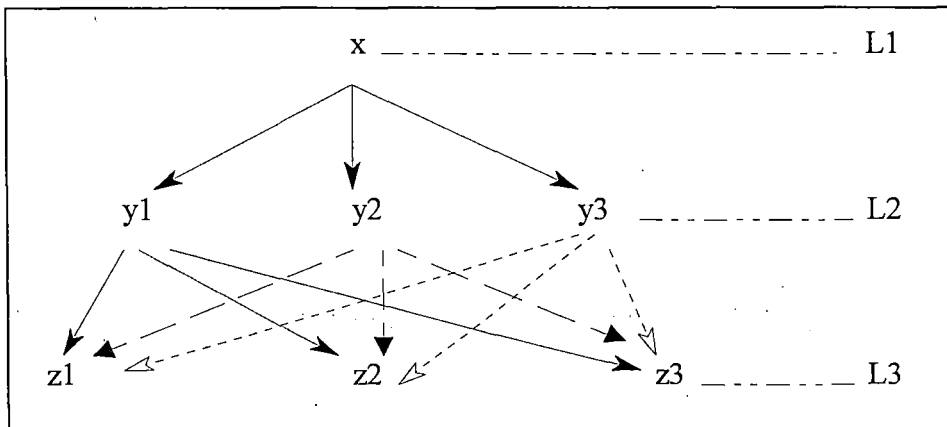
Exemple d'une hiérarchie:



Définition 4.10

Une hiérarchie est *complète* si: $\forall x \in L_k: x^+ = L_{k-1}, k=2, \dots, h$.

Exemple d'une hiérarchie complète



Définition 4.11

Soit H une hiérarchie. Une *fonction de priorité* des éléments couverts par $x \in H$ est une fonction $w_x: x^- \rightarrow [0,1]$ telle que: $\sum_{y \in x^-} w_x(y) = 1$.

w_x est la fonction de priorité d'un élément dans un niveau par rapport à l'objectif x. On observe que même si x^- est différent de L_{k+1} , w_x peut être défini en mettant tous les éléments de L_{k+1} qui ne sont pas dans x^- égal à zéro.

Avantages des hiérarchies:

1. Une représentation hiérarchique d'un système permet de décrire la manière dont les changements de priorité dans les niveaux supérieurs

affectent la priorité entre les éléments dans les niveaux inférieurs de la hiérarchie.

2. Elles sont stables et flexibles; stables dans le sens que de petits changements produisent de petits effets, et flexibles puisque des ajouts à une hiérarchie bien structurée n'influencent pas les performances.
3. Les systèmes assemblés hiérarchiquement évoluent plus efficacement que ceux assemblés comme un tout.
4. Elles donnent beaucoup d'informations sur la structure des niveaux inférieurs et fournissent une vue globale des acteurs et de leurs fonctions dans les niveaux supérieurs.

4.2.4 La méthode des puissances.

C'est une méthode simple pour trouver la valeur propre dominante d'une matrice P. Elle se base sur le théorème suivant.

Théorème 4.5

Soit P une matrice ayant m vecteurs propres linéairement indépendants et soit $u^{(0)}$ un vecteur normé arbitraire. Considérons l'itération suivante::

$$\begin{aligned} v^{(k)} &= P u^{(k-1)} \quad (k=1, 2, \dots) \\ u^{(k)} &= v^{(k)} / \|v^{(k)}\| \end{aligned}$$

Alors

$\lim_{k \rightarrow \infty} \|v^{(k)}\| = |\lambda_i|$ et $\lim_{k \rightarrow \infty} u^{(k)} = x^{(i)}$ où λ_i est une valeur propre dominante de P et $x^{(i)}$ est le vecteur propre correspondant à λ_i .

Preuve du théorème 4.5

Soit $u^{(0)} = \sum_{j=1}^m \gamma_j x^{(j)}$ (vecteur normé arbitraire).

Après k itérations, on obtient à un facteur de normalisation près:

$$u^{(k)} = \sum_{j=1}^m \gamma_j \lambda_j^k x^{(j)}$$

alors

$$u^{(k)} = \lambda_i^k \left\{ \gamma_i x^{(i)} + \sum_{\substack{j=1 \\ j \neq i}}^m \gamma_j \left(\frac{\lambda_j}{\lambda_i} \right)^k x^{(j)} \right\}$$

et donc

$$\lim_{k \rightarrow \infty} u^{(k)} = x^{(i)} \text{ car } \left(\frac{\lambda_j}{\lambda_i} \right)^k < 1$$
$$\lim_{k \rightarrow \infty} \|v^{(k)}\| = |\lambda_i| \text{ car } v^{(k)} = P u^{(k-1)}$$



4.3 La méthode de rangement par paires.

4.3.1 Introduction

Le problème de prise de décision étant souvent complexe, il est nécessaire de le structurer. Comme la plupart des chercheurs, Saaty utilise une hiérarchie afin d'organiser les différents objectifs et individus.

A partir de cette structuration du problème en une hiérarchie, nous devons trouver l'échelle des priorités ou poids des éléments pour tous les niveaux en terme de critères du niveau directement supérieur.

Concrètement, on voudrait ranger des actions d'un ensemble $A = \{A_1, A_2, \dots, A_n\}$ sur la base de comparaisons par paires s'appuyant sur des poids relatifs de ces actions, mesurés sur une échelle de rapports.

4.3.2 Axiomatique de l'A.H.P.

4.3.2.1 Définitions préliminaires

Définition 4.12

$A = \{A_1, A_2, \dots, A_n\}$ est l'ensemble fini des n alternatives.

Définition 4.13

C est l'ensemble des critères par rapport auxquels les éléments de A sont comparés.

Définition 4.14

Quand on compare deux objets selon un critère on réalise "des comparaisons par paires". La relation $>_C$ signifie "préféré à" selon le critère C et la relation \sim_C signifie "indifférent à" selon le critère C.

Définition 4.15 (Echelle fondamentale)

Soit

$$P: A \times A \rightarrow \mathbb{R}^+$$

$$f: C \rightarrow P$$

$P_C \in f(C)$ pour tout $C \in C$

Ainsi à chaque paire $(A_i, A_j) \in A \times A$ on peut faire correspondre un réel positif $P_C(A_i, A_j) = a_{ij}$ qui représente l'intensité relative avec laquelle un individu perçoit une propriété $C \in C$ d'une alternative $A_i \in A$ par rapport à une autre $A_j \in A$ c'est-à-dire:

$$A_i >_C A_j \leftrightarrow P_C(A_i, A_j) > 1$$

$$A_i \sim_C A_j \leftrightarrow P_C(A_i, A_j) = 1.$$

de ce fait pour comparer le degré d'importance

Définition 4.16

Etant donné un réel positif $\rho \geq 1$ un ensemble non vide $x \subseteq L_{k+1}$ est dit ρ -homogène par rapport à $x \in L_k$ si $1/\rho \leq P_C(y_1, y_2) \leq \rho$ pour tout $y_1, y_2 \in x$.

Définition 4.17

Un ensemble est dit *extérieurement dépendant* d'un ensemble C si une échelle fondamentale peut être définie sur A par rapport à chaque $C \in C$.

Définition 4.18

Soit A extérieurement dépendant de C. Les éléments de A sont dits *intérieurement dépendants* par rapport à $C \in C$ si pour certains $A \in A$, A est extérieurement dépendant de A.

Contre exemple?

4.3.2.2 Les axiomes

En utilisant les notions précédentes, Saaty a défini quatre axiomes sur lesquels se base l'A.H.P..

Axiome 1 (réciprocité)

Etant donné deux alternatives $(A_i, A_j) \in \mathbf{A} \times \mathbf{A}$, l'intensité de préférence de A_i sur A_j est inversement liée à l'intensité de préférence de A_j sur A_i c'est-à-dire

$$P_C(A_i, A_j) = 1/P_C(A_j, A_i) \text{ pour tout } (A_i, A_j) \in \mathbf{A} \times \mathbf{A} \text{ et pour tout } C \in \mathbf{C}.$$

Intuitivement on peut traduire ce premier axiome par l'idée que si A est 5 fois plus lourd que B, il s'en suit que B doit être 1/5 fois aussi lourd que A.

Axiome 2 (homogénéité)

Etant donné une hiérarchie H , $x \in H$ et $x \in L_k \subset H$ alors $x \in L_{k+1}$ est ρ -homogène pour tout $k = 1, 2, \dots, h-1$.

Cet axiome établit le fait que les individus sont capables d'exprimer les intensités de préférences significatives uniquement si les éléments sont comparables. Intuitivement, il est très difficile de comparer le poids du soleil et celui d'un atome car nos capacités cognitives ne nous permettent pas de telles comparaisons pour des objets si différents.

*OK ad
p fait*

Axiome 3 (dépendance des alternatives par rapport aux critères)

Soit H une hiérarchie avec L_1, \dots, L_h ses niveaux.

- (i) L_{k+1} est extérieurement dépendant de L_k
- (ii) L_{k+1} est intérieurement dépendant de L_k pour tout $x \in L_k$
- (iii) L_k est extérieurement dépendant de L_{k+1} .

?

Ce troisième axiome exprime la capacité de faire des comparaisons dans une structure hiérarchique. Cet axiome affirme que de telles comparaisons sont possibles quand un ensemble d'éléments L_{k+1} doit être comparé en terme d'un élément dans le prochain niveau supérieur.

Axiome 4 (axiome des attentes)

Tous les critères et toutes les alternatives sont représentés dans la hiérarchie c'est-à-dire $\mathbf{C} \subset H L_h$ et $\mathbf{A} = L_h$

Ainsi, toutes les attentes doivent être représentées en terme de critères et d'alternatives dans la structure et assignées aux priorités compatibles avec les attentes.

L'objet principal de cet axiome est de traiter les questions relatives à l'ajout ou au retrait d'alternatives à **A**.

Considérons le cas où une copie d'une des alternatives est ajoutée à **A**. L'axiome 4 nous dit qu'un nouveau critère doit être ajouté à **C**.

Cet axiome des attentes implique aussi que toute alternative A_j qui n'est pas une vraie alternative, dans le sens où une autre alternative $A_j \in \mathbf{A}$ lui est complètement équivalente pour tout critère, devrait être enlevée puisqu'elle n'ajoute rien de nouveau à l'ensemble des alternatives. Ainsi, seul les choix réels pourraient rester dans **A**.

Cependant, afin de prouver que deux alternatives sont des copies, on doit prouver qu'il n'existe aucun critère qui pourrait différencier les deux alternatives.

Ainsi, une copie par rapport à **C** ne devrait être ajoutée que si elle apporte une nouvelle alternative réelle à **A** en modifiant soit l'ensemble **C**, soit les priorités assignées aux critères et alternatives.

4.3.3 La phase d'évaluation

La phase d'évaluation est basée sur le concept de comparaison par paires. Les éléments d'un certain niveau de la hiérarchie sont comparés par rapport à un certain critère d'un niveau directement supérieur.

Lors de cette étape, les axiomes de la théorie sont transparents. En effet, si le preneur de décision ne peut produire une réponse, alors soit la question n'est pas significative soit les alternatives ne sont pas comparables (l'homogénéité n'est pas respectée).

Supposons que nous ayons affecté aux alternatives A_1, A_2, \dots, A_n les poids respectifs w_1, w_2, \dots, w_n mesurés sur une échelle de rapports et tels que $w_i > 0$ pour tout $i = 1, \dots, n$.

On suppose que les poids correspondant aux actions sont différents de zéro car une action ayant un poids nul est systématiquement écartée de **A**.

Ces poids déterminent automatiquement un rangement des actions et l'on peut se rendre compte de l'importance relative accordée aux diverses actions par l'intermédiaire de la matrice **P** des comparaisons par paires (cfr Figure 5).

Actions	a ₁	a ₂	...	a _m
a ₁	w ₁ /w ₁	w ₁ /w ₂	...	w ₁ /w _m
a ₂	w ₂ /w ₁	w ₂ /w ₂	...	w ₂ /w _m
.
.
.
a _m	w _m /w ₁	w _m /w ₂	...	w _m /w _m

Figure 5: Matrice de comparaisons par paires

Remarquons que cette matrice P est:

- positive
- réciproque
- consistante.

Par le théorème 4.3, cette matrice P possède donc une valeur propre dominante $\lambda_{\max} = m$.

Si on note $w = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{pmatrix}$ on a alors:

$$P \cdot w = \begin{pmatrix} w_1/w_1 & w_1/w_2 & \dots & w_1/w_m \\ w_2/w_1 & w_2/w_2 & \dots & w_2/w_m \\ \dots & \dots & \dots & \dots \\ w_m/w_1 & w_m/w_2 & \dots & w_m/w_m \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{pmatrix} = \begin{pmatrix} w_1 + w_1 + \dots + w_1 \\ w_2 + w_2 + \dots + w_2 \\ \dots \\ w_m + w_m + \dots + w_m \end{pmatrix} = \begin{pmatrix} m \cdot w_1 \\ m \cdot w_2 \\ \vdots \\ m \cdot w_m \end{pmatrix} = m \cdot \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{pmatrix}$$

On vient donc de voir que $P \cdot w = m \cdot w$, par conséquent, $(P - m \cdot I)w = 0$ avec I la matrice unité. Il résulte donc que le vecteur poids w est le vecteur propre de P correspondant à la valeur propre dominante $\lambda_{\max} = m$.

4.3.4 Le choix d'une échelle

Comme on vient de le voir, une matrice de comparaisons est un tableau à double entrée où le répondant doit juger de l'importance d'un élément par rapport aux autres éléments du même niveau en regard d'un certain critère. Pour remplir

cette matrice, il faut répondre à une double question: du point de vue du critère X, l'objet a est-il préféré ou plus important que l'objet b? et dans quelle mesure? Répondre à cette double question revient à établir une échelle de comparaison.

4.3.4.1 Si on ne connaît pas précisément l'échelle de rapports?

Le nouveau problème qui se pose est le suivant:

Ranger les alternatives de A en disposant des estimations \tilde{e}_{ij} des rapports w_i/w_j avec la convention que $\tilde{e}_{ij} = 1/\tilde{e}_{ji}$ pour tout $i, j = 1, \dots, m$. Ainsi la matrice sera réciproque mais pas nécessairement consistante. La question qui se pose est alors la suivante: comment pourrions-nous déduire de la matrice $\tilde{E} = (\tilde{e}_{ij})$ un rangement des alternatives de A via un système de pondération de ces alternatives?

Remarques:

↗ L'homme n'est pas toujours "cohérent" lorsqu'il émet des préférences.

↗ On constate fréquemment dans la pratique des comparaisons par paires, que le décideur préférerait ne pas trancher lorsqu'on lui demande s'il préfère telle alternative ou telle autre ou s'il est indifférent entre ces alternatives. Il arrive même que le graphe de préférence émis à cette occasion contienne des triades circulaires, révélant ainsi une non-transitivité des préférences. Pour cette raison, en pratique la matrice \tilde{E} est rarement consistante.

Elle est cependant toujours positive et réciproque et possède donc (par le théorème de Perron) une valeur propre λ_{\max} dominante, réelle, positive, de multiplicité 1 et dont le vecteur propre correspondant w est à composantes w_i positives pour tout $i = 1, \dots, m$. Ainsi, $\tilde{E}w = \lambda_{\max} w$.

↗ On dit que \tilde{E} est rarement consistant, mais lorsque \tilde{E} n'est pas trop inconsistant, il est normal de penser à ranger les alternatives A_1, A_2, \dots, A_m suivant l'ordre non croissant des composantes du vecteur propre normalisé w correspondant à la valeur propre dominante λ_{\max} de la matrice \tilde{E} . Pour mesurer cette inconsistance, on utilise l'indice de concordance (I.C.). Plus sa valeur est proche de zéro, plus on est proche de la consistance. Nous reviendrons sur ce problème de la consistance au chapitre 6.

4.3.4.2 Choix d'une échelle

L'échelle est utilisée pour obtenir des approximations des rapports w_i/w_j .

Saaty a tenu compte des spécifications suivantes:

✓ L'échelle doit permettre de refléter dans la mesure du possible toutes les nuances de jugements des individus ayant à fournir les rapports w_i/w_j .

✓ Choisir les échelons x_1, x_2, \dots, x_p de telle façon que: $x_{i+1} - x_i = 1$ pour tout $i = 1, \dots, p-1$.

Remarques:

- En moyenne, un individu ne peut pas comparer simultanément 7 objets sans faire de confusion.
- Saaty a choisi $p = 7+2$, d'où il résulte que les échelons varient de 1 à 9.
- $x_1=1$ représente une comparaison de deux alternatives aboutissant à l'attribution d'une importance égale à ces deux alternatives.

Pour réaliser les comparaisons par paires nous utiliserons donc l'échelle de Saaty qui est définie dans la Figure 6.

4.3.5 La méthode de Saaty

Voici la méthode de Saaty résumée en quatre points:

① Construire la liste $\mathbf{A} = \{A_1, A_2, \dots, A_m\}$ des alternatives à ranger.

② Construire une matrice $\mathbf{P} = \mathbf{P}_C(A_i, A_j)$ $i, j = 1, \dots, m$ de comparaisons par paires où $\mathbf{P}_C(A_i, A_j)$ est déterminé conformément à l'échelle de Saaty représentée dans la Figure 6 et où les deux propriétés suivantes sont respectées

1°) $\mathbf{P}_C(A_i, A_i) = 1$

2°) $\mathbf{P}_C(A_i, A_j) = 1 / \mathbf{P}_C(A_j, A_i)$ pour tout $(A_i, A_j) \in \mathbf{A} \times \mathbf{A}$
et pour tout $C \in \mathbf{C}$.

③ Déterminer la valeur propre dominante λ_{\max} de la matrice ainsi que le vecteur propre normalisé w correspondant.

④ Si la matrice P est suffisamment proche de la consistance, ranger les alternatives suivant l'ordre non croissant des valeurs des composantes de w . Ainsi si $w^* = \max w_i$ pour tout $i = 1, \dots, m$ alors l'alternative A^* associée à w^* a le rang 1 et "supplante" toutes les autres alternatives.

Valeur de $P_C(A_i, A_j)$	Définition	Signification
1	Dominance égale	A_i et A_j contribuent également à la réalisation de l'objectif
3	Dominance faible	L'expérience et le jugement favorisent légèrement A_i par rapport à A_j
5	Dominance moyenne	L'expérience et le jugement favorisent moyennement A_i par rapport à A_j
7	Dominance forte	A_i est fortement favorisée par rapport à A_j car la pratique montre que A_i domine nettement A_j
9	Dominance absolue	Il est tout à fait certain que A_i domine A_j
2,4,6,8	Valeurs intermédiaires entre deux dominances adjacentes	Compromis entre deux jugements successifs de la liste ci-dessous.

Figure 6: Echelle de Saaty

4.3.6 Exemple: estimation du poids de 5 objets

On dispose de cinq objets et on veut avoir une idée des poids relatifs de ces objets en les soupesant. Saaty a procédé en deux temps:

1. On soupèse d'abord successivement chaque objet pour avoir une idée de l'intervalle des poids;
2. On compare ensuite les poids des objets en les soupesant deux à deux (cfr Figure 7)

Objets	Radio	Machine à écrire	Grand attaché case	Projecteur	Petit attaché case
Radio	1	1/5	1/3	1/4	4
Machine à écrire	5	1	2	2	8
Grand attaché case	3	1/2	1	1/2	4
Projecteur	4	1/2	2	1	7
Petit attaché case	1/4	1/8	1/4	1/7	1

Figure 7: Matrice de comparaisons par paires

On calcule les valeurs propres. On obtient:

$$\begin{aligned}\lambda_1 &= 5.16 \\ \lambda_2 &= -0.056 + 0.433 i \\ \lambda_3 &= -0.056 - 0.433 i \\ \lambda_4 &= -0.025 + 0.798 i \\ \lambda_5 &= -0.025 - 0.798 i\end{aligned}$$

On obtient donc comme valeur propre dominante:

$$\lambda_{\max} = 5.16$$

L'indice de concordance vaut: $I.C. = \frac{\lambda_{\max} - m}{m-1} = \frac{5.16 - 5}{5 - 1} = 0.04$

Le vecteur propre correspondant à la valeur propre λ_{\max} et les poids relatifs réels se trouvent dans le tableau suivant (cfr Figure 8).

Objets	Vecteur propre	Poids relatifs réels
Radio	0.09	0.10
Machine à écrire	0.40	0.39
Grand attaché case	0.18	0.20
Projecteur	0.29	0.27
Petit attaché case	0.04	0.04

Figure 8: Vecteur propre

4.3.7 Utilisation de la méthode de Saaty dans les hiérarchies

4.3.7.1 La proposition de Saaty

Le problème est le suivant:

Soit H une hiérarchie de niveau L_1, L_2, \dots, L_k . Supposons qu'une fonction de priorité w_x des éléments de x soit définie pour tout $x \in H$. Soient $x \in L_\alpha$ et $S \subset L_\beta$ ($\alpha < \beta$). Comment définir une fonction $w_{x,S}: S \rightarrow [0,1]$ reflétant les propriétés des fonctions de priorité w_y des éléments y des niveaux $L_k, k=\alpha, \dots, \beta-1$?

Autrement dit: étant donné un système dont l'objectif principal est x et dont l'ensemble des activités de base est S , quelles sont les priorités des activités de S dans la réalisation de l'objectif x ?

Saaty propose de procéder comme pour les hiérarchies complètes. Plus précisément, considérons $x \in L_{k-1}$, $L_k = \{y_1, y_2, \dots, y_{m_k}\}$, $L_{k+1} = \{z_1, z_2, \dots, y_{m_{k+1}}\}$.

Alors la fonction de priorité des éléments de L_{k+1} par rapport à x est la fonction suivante:

$$w: L_{k+1} \rightarrow [0,1]$$
$$z_i \rightarrow w(z_i) = \sum_{j=1}^{m_k} w_{y_j}(z_i) w_x(y_j) \quad i = 1, \dots, m_{k+1}.$$

Cette fonction de priorité peut s'écrire sous forme matricielle:

$$w_i = \sum_{j=1}^{m_k} b_{ij} \bar{w}_j, \quad i = 1, \dots, m_{k+1}.$$

avec

$$\bar{W} = \begin{pmatrix} \bar{W}_1 \\ \cdot \\ \cdot \\ \bar{W}_{m_k} \end{pmatrix} = \begin{pmatrix} w_x(y_1) \\ \cdot \\ \cdot \\ w_x(y_{m_k}) \end{pmatrix},$$

pour $x \in L_{k-1}$, $L_k = \{y_1, \dots, y_{m_k}\}$ $L_{k+1} = \{z_1, \dots, z_{m_{k+1}}\}$

et

$B = (b_{ij})$, est appelé matrice de priorité de niveau L_{k+1}

On a posé:

$$b_{ij} = w_{y_j}(z_i).$$

Le vecteur $W = B\bar{W}$ est appelé **vecteur de priorité**:

$$W = \begin{pmatrix} W_1 \\ \cdot \\ \cdot \\ W_{m_{k+1}} \end{pmatrix} = \begin{pmatrix} w(z_1) \\ \cdot \\ \cdot \\ w(z_{m_{k+1}}) \end{pmatrix}.$$

Définition 4.19

Supposons que, pour chaque sous-objectif ou activité C_j du niveau L_k , il existe une échelle ordinale O_j sur les activités \bar{C}_α ($\alpha=1, \dots, m_{k+1}$) du niveau L_{k+1} . L'ordre partiel du niveau L_{k+1} est l'ordre \geq défini par:

$$\forall \bar{C}_\alpha, \bar{C}_\beta \in L_{k+1}: \bar{C}_\alpha \geq \bar{C}_\beta \Leftrightarrow O_{\alpha j} \geq O_{\beta j}, j = 1, \dots, m_k.$$

Théorème 4.6

Soient $L_k = \{C_1, \dots, C_m\}$ et $L_{k+1} = \{\bar{C}_1, \dots, \bar{C}_{m_{k+1}}\}$ deux niveaux successifs de la hiérarchie complète H . Supposons que $\{w_{C_j}(\bar{C}_1), \dots, w_{C_j}(\bar{C}_{m_k})\}$ soit le vecteur propre normalisé correspondant à la valeur propre dominante de la matrice des comparaisons par paires des alternatives de L_{k+1} par rapport à C_j , pour tout $j = 1, \dots, m_k$.

Supposons aussi que ce vecteur propre préserve l'ordre des \bar{C}_α sur l'échelle O_j , correspondant à C_j .

Alors

Si $(w_1, \dots, w_{m_{k+1}})$ est le vecteur de priorité des activités de L_{k+1} par rapport à un objectif z du niveau L_{k-1} **on a**:

$$\bar{C}_\alpha \geq \bar{C}_\beta \Leftrightarrow W_\alpha \geq W_\beta$$

Théorème 4.7

Soit H une hiérarchie complète dont le suprémum est b et les niveau L_1, \dots, L_h .

Si B_k est la matrice de priorité du niveau L_k , $k=2, \dots, h$ et si \bar{W} est le vecteur de priorité du niveau L_p , par rapport à $z \in L_{p-1}$, alors le vecteur de priorité W du niveau L_q ($p>q$) par rapport à z peut s'écrire: $W = B_q B_{q-1} \dots B_{p+1} \bar{W}$

En particulier, le vecteur de priorité L_h par rapport à $L_1 = \{b\}$ peut s'écrire:

$$W = B_h B_{h-1} \dots B_3 \bar{W}.$$

CHAPITRE 5

UNE APPLICATION DE L'A.H.P.

Chapitre 5

Une application de l'A.H.P.

5.1 Introduction

Un médecin homéopathe doit choisir entre plusieurs médicaments susceptibles de guérir le patient qui vient se présenter à son cabinet. Celui-ci présente différents symptômes.

L'homéopathe va pouvoir naviguer dans Radar afin d'établir les listes de remèdes correspondant aux symptômes sélectionnés ainsi que le degré attribué à chacun des remèdes. Voici les listes de ceux-ci pour chaque symptôme correspondant, le degré sera indiqué entre parenthèses ainsi que l'abréviation utilisée par Radar.

1. Vertigo morning compelled to lie down

Pulsatilla pratensis (*puls*; 3)

Nitricum acidum (*nit-ac*; 2)

2. Eyes cataract viridis

Pulsatilla pratensis (*puls*; 1)

Colchicum autumnal(*colch*; 1)

Phosphorus (*phos*; 2)

3. Bleeding from eyes on blowing nose

Nitricum acidum (*nit-ac*; 1)

4. Head sensitive to cold air in the night

Phosphorus (*phos*; 2)

La résolution de cet exemple se déroulera en trois étapes:

1. la décomposition du problème en une hiérarchie;
2. l'établissement des différentes matrices de comparaisons par paires;
3. la synthèse des priorités.

5.1.1 La décomposition du problème en une hiérarchie

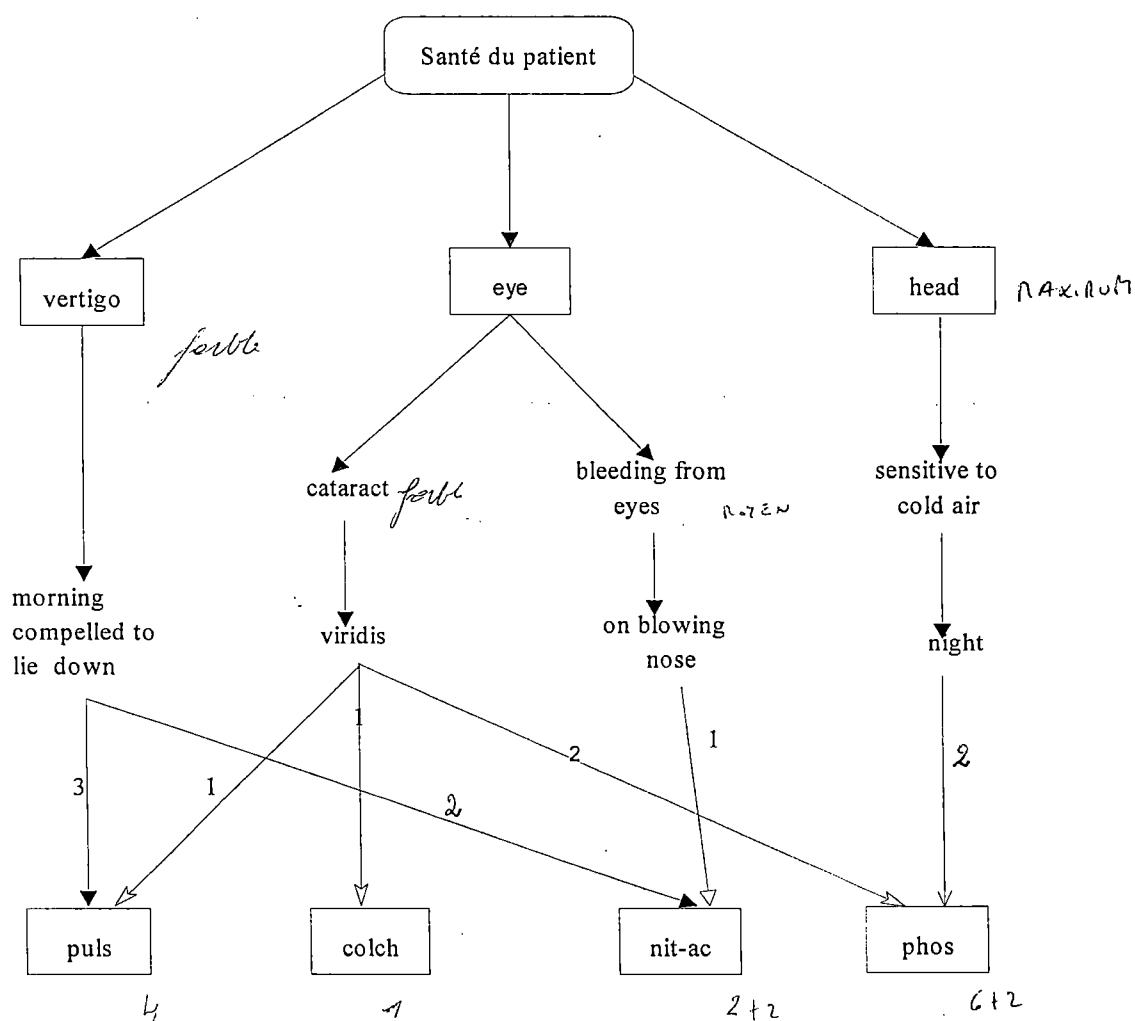
On décomposera la hiérarchie en trois niveaux:

- niveau 1 : santé du patient
- niveau 2 : l'ensemble des différents symptômes
- niveau 3 : l'ensemble des différents remèdes

Remarque:

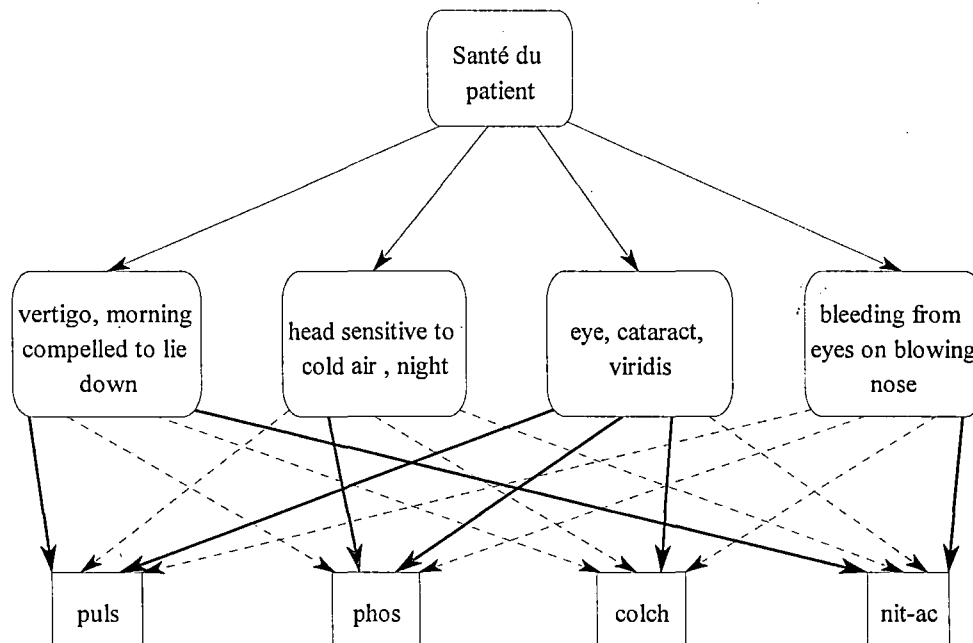
En suivant exactement la structuration du répertoire de Kent on obtiendrait plus de trois niveaux. Cependant pour résoudre notre problème, on a regroupé tous les niveaux intermédiaires.

La figure ci-dessous représente la hiérarchie en considérant tous les niveaux du Répertoire de Kent. Nous avons également précisé les degrés attribués à chaque remède.



Pour les besoins de la méthode nous allons tout d'abord redéfinir la hiérarchie ci-dessus en une hiérarchie à trois niveaux et ensuite nous allons rendre cette hiérarchie complète.

Les flèches en pointillé représentent les flèches rajoutées pour rendre la hiérarchie complète.



5.2.2 Les matrices de comparaisons par paires

Puisque la hiérarchie comprend quatre symptômes différents, l'analyse va commencer par évaluer leur force et importance relative. Cette évaluation est réalisée en posant la question suivante au médecin homéopathe: "par rapport à la santé du patient, lequel des deux symptômes suivants est le plus influent?"

Les réponses à ces questions sont résumées dans la Figure 9.

SYMPTOMES	vertigo morning compelled to lie down	Head sensitive to cold air, night	eye cataract viridis	bleeding from eyes on blowing nose
vertigo morning compelled to lie down	1	1/7	3	1/3
Head sensitive to cold air, night	7	1	9	3
eye cataract viridis	1/3	1/9	1	1/5
bleeding from eyes on blowing nose	3	1/3	5	1

Figure 9: Matrice de comparaison des symptômes

Ici nous avons une matrice de rang 4, la valeur propre dominante de cette matrice est :

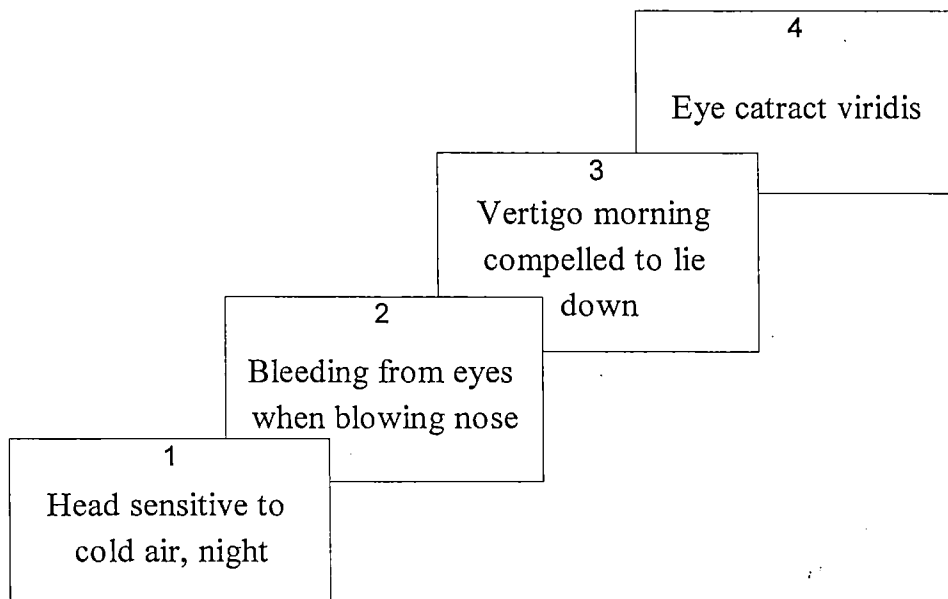
$$\lambda_{\max} = 4.08$$

Le vecteur propre normalisé correspondant à la valeur propre dominante vaut:
(0.101, 0.607, 0.048, 0.242).

Le vecteur de priorité indique que le symptôme "*head sensitive to cold air, night*" est dominant et influence la santé du patient avec une priorité de 0.607. Le symptôme "*bleeding from eyes on blowing nose*" possède une priorité de 0.242, celui "*vertigo morning compelled to lie down*" a une priorité de 0.101. Le symptôme "*eye cataract viridis*" est relativement sans importance (avec une priorité de 0.055) par rapport à la santé du malade.

Ce résultat reflète le fait que pour le médecin homéopathe le symptôme "*head sensitive to cold air night*" est le plus important par rapport aux autres.

La figure suivante résume l'ordre d'importance relative de chaque symptôme:



Comme on l'a déjà fait remarquer, un des aspects important de l'A.H.P. est la consistance, voilà pourquoi on vérifiera chaque fois la consistance des matrices de comparaisons par paires par l'intermédiaire du calcul de l'I.C..

L'indice de consistance vaut 0.027 pour la matrice de comparaison des différents symptômes cela signifie qu'elle est très proche de la consistance.

A présent on va dériver les priorités des différents remèdes par rapport au symptôme "*vertigo morning compelled to lie down*".

N'ayant pas d'échelle standard prédéfinie nous devons donc établir une échelle de rapport.

Celle-ci se base sur l'échelle à neuf points de Saaty et sur les différences de degré entre deux remèdes.

On calcule tout d'abord la différence de degré entre deux remèdes. Par exemple la différence entre *puls* et *nit-ac* pour le symptôme "*vertigo morning compelled to lie down*" vaut $3-2 = 1$. A cette différence on attribue une valeur de trois sur l'échelle de Saaty. Cette valeur correspond à une dominance faible; c'est-à-dire que l'expérience et le jugement favorisent légèrement *puls* par rapport à *nit-ac* pour le symptôme choisi.

On procède de façon similaire pour tous les remèdes. L'échelle est reprise dans la Figure 10.

Différence de degré	Echelle de Saaty
0	1
1	3
2	5
3	6
4	7
5	9

Figure 10: Echelle de rapport

- Vertigo, morning, compelled to lie down

REMEDES	<i>puls</i>	<i>colch</i>	<i>nit-ac</i>	<i>phos</i>	Vecteur propre
<i>puls</i>	1	6	3	6	0.56
<i>colch</i>	1/6	1	1/5	1	0.07
<i>nit-ac</i>	1/3	5	1	5	0.29
<i>phos</i>	1/6	1	1/5	1	0.07

rapport 1/3
rapport 1/2

3
-
2
-

La valeur propre dominante vaut $\lambda_{\max} = 4.08$ et l'indice de consistance est égal à I.C. = 0.027. Cela veut dire que la matrice est très proche de la consistance puisque I.C. tend vers 0 et λ_{\max} vers 4. Le vecteur de priorité indique que par rapport au symptôme "vertigo morning compelled to lie down", le remède *puls* est dominant avec une priorité de 0.56. Les remèdes *colch* et *phos* possèdent une priorité de 0.07 et *nit-ac* a une priorité de 0.29.

On procède de la même façon pour les trois autres symptômes.

- Eye, cataract, viridis

REMEDES	<i>puls</i>	<i>colch</i>	<i>nit-ac</i>	<i>phos</i>	Vecteur propre
<i>puls</i>	1	1	3	1/3	0.2
<i>colch</i>	1	1	3	1/3	0.2
<i>nit-ac</i>	1/3	1/3	1	1/5	0.08
<i>phos</i>	3	3	5	1	0.52

La valeur propre dominante est égale à $\lambda_{\max} = 4.04$ et l'I.C. est égal à 0.014. Cette matrice est également très proche de la consistance. Le vecteur propre indique que *phos* est le remède le plus adéquat pour ce symptôme avec une priorité de 0.52.

- **Bleeding from eyes on blowing nose**

REMEDES	<i>puls</i>	<i>colch</i>	<i>nit-ac</i>	<i>phos</i>	Vecteur propre
<i>puls</i>	1	1	1/3	1	0.166
<i>colch</i>	1	1	1/3	1	0.166
<i>nit-ac</i>	3	3	1	3	0.5
<i>phos</i>	1	1	1/3	1	0.166

On obtient ici une matrice parfaitement consistante puisque $\lambda_{\max} = 4$ et que l'indice de consistance vaut 0. Le vecteur de priorité indique que par rapport au symptôme "Bleeding from eyes on blowing nose", le remède *nit-ac* est dominant avec une priorité de 0.5. Les remèdes *colch*, *phos* et *nit-ac* possèdent une priorité de 0,166.

- **Head sensitive to cold air , night**

REMEDES	<i>puls</i>	<i>colch</i>	<i>nit-ac</i>	<i>phos</i>	Vecteur propre
<i>puls</i>	1	1/5	1	1	0.125
<i>colch</i>	5	1	5	5	0.625
<i>nit-ac</i>	1	1/5	1	1	0.125
<i>phos</i>	1	1/5	1	1	0.125

incisive

Cette matrice est également parfaitement consistante puisque I.C. est égal à zéro et $\lambda_{\max} = 4$. Ici *colch* possède une priorité de 0.625 tandis que les trois autres remèdes semblent avoir une priorité nettement inférieure à *colch*.

5.2.3 La synthèse des priorités

Les priorités globales sont dérivées en mettant en balance le vecteur de priorité de chaque symptôme par la force respective de ces symptômes, cela donne la matrice des priorités suivante :

$$\begin{pmatrix} 0.56 & 0.2 & 0.166 & 0.125 \\ 0.05 & 0.2 & 0.166 & 0.625 \\ 0.29 & 0.08 & 0.5 & 0.125 \\ 0.07 & 0.52 & 0.166 & 0.125 \end{pmatrix} \begin{pmatrix} 0.101 \\ 0.607 \\ 0.048 \\ 0.242 \end{pmatrix} = \begin{pmatrix} 0.218 \\ 0.285 \\ 0.132 \\ 0.36 \end{pmatrix} \begin{matrix} \textit{puls} \\ \textit{colch} \\ \textit{nit-ac} \\ \textit{phos} \end{matrix}$$

Le vecteur de priorité vaut (0.218, 0.285, 0.132, 0.36). Ce résultat indique que le remède *phos* est dominant et influence la santé du patient avec une priorité de 0.36. Suivent dans l'ordre *colch* (0.285), *puls* (0.218) et *nit-ac* (0.132).

L'ordre de classement des remèdes est repris dans la Figure 11.

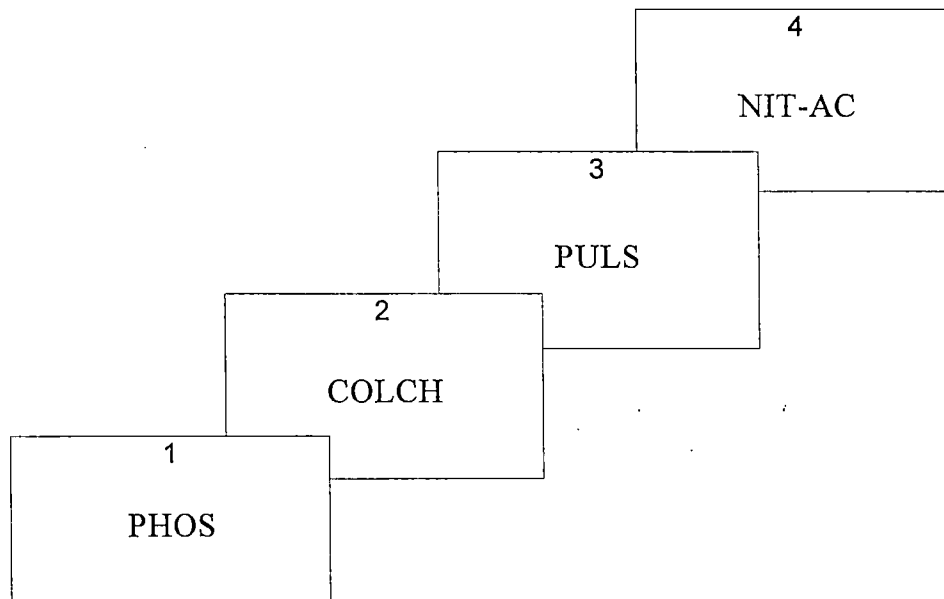


Figure 11: Ordre de classement des remèdes

Le choix final reste évidemment entre les mains du médecin homéopathe, mais celui-ci aura une meilleure idée de la façon dont les différents remèdes influencent la santé de son patient.

5.3 Conclusion

L'A.H.P. est une méthode bien adaptée à ce type de problème. Cependant, il faut savoir que lorsqu'on compare beaucoup plus de remèdes les résultats tendent à s'uniformiser, car plus le nombre de remèdes est grand plus leurs poids seront petits.

CHAPITRE 6

CRITIQUES DE L'A.H.P.

Chapitre 6

Critiques de l'A.H.P.

Les polémiques autour de l'A.H.P. ont assurément fait couler beaucoup d'encre, voilà pourquoi j'ai décidé de réserver un chapitre afin d'explorer les principales critiques faites sur l'A.H.P. et d'essayer de les renverser à son propre avantage.

6.1 Le background théorique de l'A.H.P.

Le premier reproche fait à Saaty est son manque de rigueur dans l'énoncé de la théorie axiomatique de la méthode. Or Harker et Vargas soulignent le fait qu'il a bien défini les concepts, les critères, l'échelle à neuf points, et énoncé des axiomes.

Rappelons brièvement les quatre axiomes les plus importants.

Condition de réciprocité: lorsqu'un élément A est 5 fois plus préféré à un élément B, cette condition impose que l'élément B soit 1/5 fois préféré à A.

Cette propriété se traduit par l'obligation pour la matrice de comparaisons d'être réciproque, ce qui nous permet de ne remplir qu'une moitié de la matrice, l'autre moitié étant constituée des valeurs inverses.

Condition d'homogénéité: les éléments d'un même niveau doivent être comparables pour que les individus puissent exprimer pleinement leurs préférences et l'intensité de celles-ci.

C'est une condition nécessaire pour que les poids aient un sens dans le contexte de l'échelle à neuf points.

Condition de dépendances des alternatives par rapport aux critères: exprime le fait que l'on puisse effectuer des comparaisons entre les objets d'un même niveau en terme d'un objet du niveau supérieur.

L'axiome des attentes: implique que la structure hiérarchique soit complète.

C'est à partir de cet axiome que surgit le problème du "rank reversal", c'est-à-dire le changement dans l'ordre des préférences. Nous reviendrons plus tard sur ce point.

La méthode de Saaty est historiquement différente et indépendante des théories issues de "l'Utility Theory" (théorie à un seul critère). Ces dernières se basent sur trois axiomes que nous allons vous présenter afin de pouvoir comparer avec l'A.H.P.

Soit \succ une relation binaire sur un ensemble non vide de lotteries X, Y, Z interprétée comme une préférence stricte.

La relation d'indifférence peut être définie à partir de \succ comme suit:

$$\begin{aligned} X \approx Y & \text{ si } Y \succ X \text{ et } X \succ Y \\ X \succ \approx Y & \text{ si } X \succ Y \text{ ou } X \approx Y \end{aligned}$$

Axiome 1: (Ordre complet)

Pour toutes lotteries X et Y on a soit $X \succ \approx Y$ ou $Y \succ \approx X$
Pour toutes lotteries X, Y, Z si $X \succ \approx Y$ et $Y \succ \approx Z$ alors $X \succ \approx Z$.

Axiome 2: (Continuité)

Etant donné les lotteries X, Y, Z tel que $X \succ Y \succ Z$ il existe une probabilité $p \in]0,1[$ tel que $X \approx pX + (1-p)Z$

Axiome 3: (Indépendance)

Si $X \succ Y$ alors pour tout $p \in]0,1[$ et pour tout Z on a $pX + (1-p)Z \succ pY + (1-p)Z$

Pour Dyer, ces derniers axiomes sont clairs et évidents et reflètent bien un comportement de choix. De plus, ces axiomes utilisent uniquement la relation binaire \succ et l'existence d'une échelle peut facilement être dérivée de ces axiomes.

Il reproche donc à Saaty de ne pas avoir basé son système axiomatique sur celui de l'Utility Theory; en outre, il aurait préféré que l'ensemble des axiomes de l'A.H.P. soit plus fondamental et permette de prouver l'existence de l'échelle de rapport plutôt que de supposer l'existence préalable de celle-ci.

Ces dernières critiques n'en sont pas. D'abord, si on veut généraliser l'Utility Theory, il faut également faire des suppositions irréelles; par exemple, à

propos de la transitivité et de la consistance des préférences. En outre, comme le font remarquer les utilisateurs, la difficulté d'utiliser des lotteries (avec tous les paradoxes que cela entraîne) diminue la validité et la pertinence de l'Utility Theory.

Par contre, l'axiomatique de l'A.H.P. étant assez intuitive, il est facile d'exprimer quelque chose de général d'une façon précise, et les utilisateurs parviennent aisément à exprimer leurs préférences à travers les échelles de comparaisons.

6.2 L'ambiguïté de la question

Une deuxième critique porte sur l'ambiguïté des questions auxquelles le décideur doit répondre en procédant au remplissage de la matrice de comparaison.

Ce problème est totalement dépendant du contexte et de l'environnement cognitif dans lequel on travaille. Les questions du style " est-ce plus important? " ou " est-ce beaucoup plus important? " sont fonction du contexte. Les réponses apportées vont varier de jour en jour et d'un individu à l'autre puisque l'environnement évolue dans l'espace et dans le temps.

Il est vrai aussi qu'une série de questions bien ficelée augmente considérablement l'efficacité de la méthode.

Ce problème n'est donc pas un dysfonctionnement de la méthode, mais il provient du fait que les analystes ou les preneurs de décisions ne parviennent pas à formuler correctement leurs questions.

Ainsi, on peut reconnaître que seuls le contexte et le mode de questionnement pourront en général affecter les résultats et la seule façon de corriger ces effets est de tenter d'éclaircir les questions autant que possible.

Cette ambiguïté est par conséquent un phénomène présent dans toutes les méthodes d'aide à la décision et la critique est non fondée.

Des définitions claires et précises des critères et des alternatives sont essentielles et devraient être une des principales préoccupations des utilisateurs de l'A.H.P.. L'élimination de l'ambiguïté est donc entre les mains des analystes.

6.3 L'échelle à neuf points

Une des parties les plus controversées de l'A.H.P. concerne l'échelle utilisée pour mesurer l'intensité de préférence entre deux alternatives.

Cette échelle est une échelle de rapports et on suppose que l'intensité de préférence entre deux alternatives peut être exprimée par cette échelle. L'utilisation d'une échelle de comparaisons dans l'A.H.P. constitue une grande différence par rapport aux méthodes traditionnelles d'aide à la décision qui emploient habituellement des échelles d'intervalles. Un certain nombre de questions peuvent être posées à propos de ce choix.

Une des raisons pour lesquelles des échelles d'intervalles ne peuvent être utilisées dans l'A.H.P. est que le produit de deux nombres de la même ou de deux échelles d'intervalles différentes n'appartient pas à une échelle d'intervalles. Tandis que les produits de deux échelles de rapports est une échelle de rapports et la somme de nombres d'une même échelle de rapports (comme c'est le cas dans une des propriétés de Saaty) est un nombre qui appartient à cette échelle de rapports.

La seconde raison pour laquelle on n'utilise pas d'échelles d'intervalles dans l'A.H.P., est que les critères sont parfois des concepts abstraits et leur poids est plus facilement déterminé par comparaisons ou par rapports.

Malgré tout, la limitation de l'échelle de 1 à 9 impose des restrictions non naturelles. En effet, si A domine moyennement B (5 sur notre échelle) et B domine moyennement C, alors qu'en est-il de la dominance de A par rapport à C?, la valeur sur l'échelle devrait être 25.

La justification de l'emploi d'une échelle à neuf points est purement pratique. Saaty a en effet comparé 27 transformations verbales différentes en échelles de rapports. Il s'est avéré que dans la plupart des cas c'est cette échelle qui fut la meilleure. Elle reproduit donc assez bien les préférences des répondants et est plus flexible et mieux adaptée aux besoins des individus que d'autres échelles, comme par exemple l'échelle d'intervalles.

6.4 La composition hiérarchique

Une autre critique a été énoncée à l'encontre du principe même de composition hiérarchique. Certains auteurs disent qu'elle est trop simple et qu'elle serait à l'origine de l'aspect le plus controversé de l'A.H.P.: le "rank reversal". Or, comme nous allons le voir, le problème du rank reversal apparaît uniquement lors d'une mauvaise utilisation de la méthode. De plus, il existe une forme plus complète de composition: le système avec "feedback" ou rétroaction (la dépendance se fait aussi dans le sens critères - alternatives). Nous allons également réserver un paragraphe pour faire quelques remarques au sujet de la consistance (ou cohérence).

6.4.1 Le "rank reversal"

Le "rank reversal" n'est rien d'autre qu'un phénomène qui bouleverse l'ordre des préférences.

Dans la vie de tous les jours il se passe fréquemment: qu'un individu qui a connaissance d'une nouvelle alternative, puisse changer l'ordre de ces priorités. Rappelons le célèbre exemple de Luce et Raiffa (1957), montrant que le phénomène du "rank reversal" est présent dans la vie courante.

Dans un nouveau restaurant, un gentleman choisit du saumon plutôt qu'un steak jusqu'au moment où il entendit qu'on servait également des escargots et des cuisses de grenouilles. A cet instant, il décida de prendre un steak.

On peut se poser une double question: est-ce que la procédure de décision donne le bon ordre et les bonnes valeurs pour les alternatives et est-ce que cet ordre peut être modifié si on ajoute ou si on supprime une alternative?

6.4.1.1 La mesure relative et la mesure absolue

Dans un de ses nombreux articles, Saaty distingue deux types de mesure: la mesure relative et la mesure absolue.

Leur point commun est d'avoir toutes les deux recours aux comparaisons par paires d'éléments pour dériver les priorités. Mais elles diffèrent en ceci: avec une mesure relative, les comparaisons sont effectuées à travers toute la hiérarchie et elles donnent des valeurs relatives. En utilisant la mesure absolue, on compare également les éléments à travers toute la hiérarchie, exception faite des

alternatives elles-mêmes; celles-ci ne sont pas confrontées deux à deux, mais elles sont comparées à des standards connus d'avance. Ces standards permettent de juger l'acceptabilité d'une alternative. Citons l'exemple de Saaty illustrant la mesure absolue: si on veut savoir quels étudiants peuvent être admis dans une école, on confronte les scores de chaque étudiant -calculés par l'A.H.P.- avec les normes d'admission de cette école.

Pourquoi cette distinction entre les deux façons de mesurer nous intéresse-t-elle? Parce que chacune a une attitude différente face au rank reversal.

En cas d'ajout ou de suppression d'une nouvelle alternative, le rang reste inchangé si on mesure dans l'absolu; tandis qu'avec la mesure relative, il peut être bouleversé à cause de la dépendance structurelle des critères sur les alternatives. Donc si on recourt à un procédé absolu, on peut éviter ce rank reversal à condition de posséder des standards. Or ceux-ci ne sont pas toujours disponibles pour tous les types de problèmes, mais seulement pour ceux dont le fonctionnement est bien connu. Pour les autres qui sont nouveaux ou qui présentent des éléments intangibles, il est difficile de trouver des normes. Dans ce cas, on utilise la mesure relative. Et c'est justement le procédé qui nous occupe.

6.4.1.2 "Rank reversal " avec une mesure relative

Qu'arrive-t-il lorsqu'on ajoute une alternative qui soit la copie d'une autre déjà existante dans l'ensemble des choix finaux?

Si elle ne se distingue absolument pas des autres, cela veut dire qu'on ne rajoute aucune information, on ne fait qu'alourdir la structure et fausser le résultat. Si par contre, on introduit en même temps que la nouvelle alternative un critère permettant de la distinguer des autres, on élargit nos possibilités de choix et on enrichit la hiérarchie. Dans ce cas, il n'y a aucun danger de changement dans l'ordre des priorités. L'exemple suivant nous éclairera directement. Si le seul critère de différenciation est le type de véhicule, on peut distinguer un bus d'une auto, mais pas un bus rouge d'un bus bleu. Pour cela, il faudrait introduire la couleur comme facteur discriminant dans le choix d'un véhicule.

L'explication habituellement donnée pour l'exemple de Luce et Raiffa est la suivante: ayant entendu qu'on proposait également des escargots et des cuisses de grenouilles, le gentleman a conclu que cela devait être un bon restaurant et donc que le steak devait être de bonne qualité. C'est-à-dire qu'un nouveau critère de qualité du restaurant a été ajouté à l'ensemble des critères.

Notons que ce phénomène n'a rien à voir avec la consistance des jugements; si l'introduction de la nouvelle alternative conserve la consistance de la matrice, cela n'empêchera pas qu'il puisse avoir rank reversal.

On peut aussi éviter le rank reversal en groupant les alternatives ou les critères identiques ou semblables en "classes d'équivalence" de sorte qu'ils respectent une propriété commune.

Il existe aussi des situations où certaines données critiques sont modifiées. Même légères, ces modifications perturbent inévitablement le résultat; c'est ce que certains auteurs appellent une "catastrophe". Elle peut être évitée en introduisant dans le problème un critère qui prend en compte l'inconnu. On calcule donc les priorités entre les alternatives en considérant ce critère supplémentaire.

Plusieurs critiques ont été faites à l'encontre de l'A.H.P. sur le fait que rien ne garantissait la stabilité et la fiabilité de l'ordre donné. Cependant Saaty a démontré la bonne performance de la méthode des vecteurs propres; il a même élaboré un théorème et des conditions pour éviter ce bouleversement dans le rang.

Ce qu'il est surtout important de retenir de tout cela, c'est que ce phénomène de rank reversal peut se produire avec l'A.H.P., mais qu'il est acceptable.

6.4.2 La consistance

6.4.2.1 les sources de l'inconsistance

Le manque de consistance se mesure à deux niveaux : à chaque palier de la hiérarchie et pour la hiérarchie toute entière.

Les inconsistances dans les paliers proviennent avant tout du fait que les répondants sont des êtres humains et qu'ils ne sont jamais infallibles; on peut dire que A est préféré à B et que B est préféré à C et, plus loin, que C est préféré à A. Il arrive que l'on oublie ou que l'on se trompe tout simplement. D'où l'importance de limiter le nombre d'éléments à comparer (pas plus de sept plus ou moins deux).

Une deuxième source d'erreur résulte souvent d'une mauvaise compréhension de la question ou du sujet traité. Le répondant doit pouvoir émettre ses jugements en connaissance de cause sinon ses réponses n'ont pas de sens.

Au niveau de la hiérarchie toute entière, les inconsistances découlent soit des erreurs faites dans les paliers, soit d'un éventuel mauvais calcul. Celui-ci étant fait le plus souvent par ordinateur, ce risque est faible.

6.4.2.2 L'indice de consistance

La vérification de la consistance dans la matrice de comparaisons est un point très important dans la procédure hiérarchique. De ce contrôle dépend la fiabilité des résultats obtenus.

Rappelons que la matrice P satisfait la propriété de consistance si et seulement si $p_{ik} p_{kj} = p_{ij}$. Cela implique qu'à partir de n'importe quelle colonne de P, on peut déterminer le reste des entrées grâce à cette relation.

Il existe un indice de consistance $I.C. = (\lambda_{\max} - m) / (m-1)$ s'il vaut 0, nous avons la consistance parfaite puisque (par le théorème 4.3) $\lambda_{\max} = m$. S'il est supérieur à 0, nous avons inconsistance.

La rapport d'inconsistance suivant nous dit dans quelle mesure nous avons inconsistance : $R.C. = (I.C. / I.C.H.) 100$

Où I.C.H. est l'indice de consistance moyen pour des poids générés au hasard. La valeur limite supérieure pour ce pourcentage a été fixée à 10 % pour un seuil d'inconsistance raisonnable; au delà de ce seuil, les jugements sont supposés inadéquats pour la procédure de prise de décision.

6.4.2.3 Correction des jugements inconsistants

L'Analytic Hierarchy Process n'exige pas la consistance parfaite et tolère un certain niveau d'inconsistance lorsque cela n'a pas grand effet sur les poids finaux. Cependant, si celui-ci est trop élevé, une révision des jugements s'avère nécessaire.

Les inconsistances proviennent souvent de la mauvaise information du répondant sur le problème posé. Dès lors, la meilleure et la plus simple solution consiste à instruire davantage nos répondants et à éliminer toute ambiguïté sur le sujet de l'étude.

6.5 La méthode de globalisation des poids

La technique utilisée pour calculer les poids globaux des alternatives est la synthèse par un modèle additif: on pondère les priorités relatives de chaque

alternative sous chaque critère par le poids de ces critères. Il existe plusieurs méthodes pour dériver les priorités relatives: les vecteurs propres (VP), choisis par Saaty, les moindres carrés logarithmiques (MCL), les moindres carrés (MC) et la programmation linéaire (PL).

Dans un premier temps, nous allons décrire ces différentes méthodes qui permettent d'évaluer l'échelle pour l'A.H.P.. Ensuite, nous essayerons de montrer la supériorité des vecteurs propres par rapport aux autres techniques sur base d'un critère déterminé: la préservation du rang.

Remarquons qu'il existe d'autres méthodes que celles énoncées ici: la méthode arithmétique, la méthode géométrique, le moyen dit harmonique, etc... Trop simples et généralement fausses, elles ne nous intéressent pas.

6.5.1 La méthode des vecteurs propres (VP)

Saaty préconise l'utilisation des vecteurs propres si l'utilisateur veut s'assurer du bon fonctionnement de la procédure. Il justifie son choix en disant que cette technique fournit un ordre de priorité entre les alternatives qui respecte le mieux possible les jugements des répondants. En d'autres termes, son résultat est fiable, le vecteur propre normalisé est supposé donner le bon ordre entre les choix de décision. De plus, la valeur propre dominante λ_{\max} associée au vecteur propre donne une mesure de la consistance de la matrice.

6.5.1.1 Rappel mathématique

Par définition, un vecteur propre est un vecteur w non nul tel que: $Pw = \lambda w$. Où les λ sont les valeurs propres associées. Donc, w est un vecteur propre s'il est une solution non nulle de $(P-\lambda I)w = 0$, avec I matrice d'identité d'ordre m .

On détermine les valeurs propres λ en calculant les racines du polynôme de degré m en λ :

$$\lambda^m - p_1 \lambda^{m-1} + \dots + (-1)^m \det(P) = 0$$

De plus on sait que les racines de ce polynôme satisfont les relations suivantes:

$$1^\circ) \sum_{i=1}^m \lambda_i = p_1 = \text{trace}(P)$$

$$2^\circ) \prod_{i=1}^m \lambda_i = |P|$$

De là, il découle que m est le nombre maximum de valeurs propres et que m est la valeur propre dominante si P est une matrice consistante.

6.5.1.2 Calcul du vecteur propre et de la valeur propre

Nous voyons vite que les calculs sont fastidieux et ne rendent pas l'A.H.P. attractive à ce point de vue.

Cependant grâce à la méthode des puissances, il nous est possible facilement de calculer la valeur propre dominante et le vecteur propre normalisé qui lui est associé. Ce système automatisé permet l'utilisation de l'A.H.P. sur un très grand nombre de problèmes. Il calcule le degré d'inconsistance de la matrice et permet donc une révision des jugements inconsistants en les représentant au décideur pour une éventuelle révision.

En pratique, pour résoudre l'expression $Pw = \lambda w$ on distinguera deux possibilités suivant que la matrice P est consistante ou non.

En résumé, la méthode des vecteurs propres est relativement claire et précise. Si les calculs ne sont pas toujours simples, ils ont l'avantage de pouvoir être calculés par un système automatisé.

6.5.2 La méthode des moindres carrés logarithmiques (MCL)

Cette technique optimise les différences au carré des écarts entre les logarithmes des jugements et de leur poids relatif.

Il s'agit donc de résoudre l'équation:

$$\min \sum_{i,j=1}^m (\log p_{ij} - \log \frac{u_i}{u_j})^2$$

La solution de cette minimisation provient du produit normalisé des éléments de chaque ligne:

$$u_i = \left(\prod_{j=1}^m p_{ij} \right)^{\frac{1}{m}} \left(\sum_{j=1}^m w_j \right)^{\frac{1}{m}}$$

Concrètement, cela veut dire que nous faisons l'hypothèse que le logarithme de p_{ij} représente bien p_{ij} . Donc le résultat des MCL est un estimateur non biaisé sous des hypothèses de normalité. Cette méthode est une démarche statistique. Ses résultats devront donc être interprétés.

On peut se demander ce que deviennent les solutions des MCL quand on ne raisonne plus sous des hypothèses de normalité. Cette interrogation met en lumière le manque de stabilité et de fiabilité de cette démarche.

6.5.3 La méthode des moindres carrés (MC)

Il s'agit de trouver le vecteur des priorités $v = (v_1, \dots, v_n)^T$ qui optimise l'équation:

$$\min \sum_{i,j=1}^n \left(p_{ij} - \frac{v_i}{v_j} \right)^2$$

Autrement dit, on cherche à minimiser les carrés des différences entre les jugements (p_{ij}) et leur poids relatif (v_i / v_j)

Pratiquement, Saaty propose d'utiliser les moindres carrés pour approximer une matrice de données par une autre matrice de rang inférieur. En effet, le vecteur colonne $v = (v_i / v_j)$ est en fait une matrice de rang égal à 1. Saaty dit que ce vecteur est une approximation de $P = (p_{ij})$. Il le prouve en recourant aux moindres carrés.

A partir de cela un problème se pose. Comment obtenir une échelle de priorité au départ de cette approximation?

A nouveau, il faut distinguer une matrice consistante d'une matrice inconsistante. Dans le cas d'une matrice consistante, on peut choisir n'importe laquelle de ces colonnes et la normaliser, cela donne la même solution que le calcul du vecteur propre. Reste à savoir si ce vecteur est aussi bon en terme de qualité de l'approximation de la réalité que le vecteur propre calculé sur P

Nous pouvons aussi ajouter que, contrairement aux vecteurs propres, cette technique ne possède pas de concept de décision sous-jacent comme la dominance; on ne fait que minimiser et en termes de décision cela ne veut rien dire.

6.5.4 La méthode de la programmation linéaire (PL)

Cette technique permet une estimation simultanée des poids des priorités de tous les éléments dans la hiérarchie dans le cas d'une hiérarchie incomplète.

Le calcul du vecteur des priorités relatives des alternatives de décision par la programmation linéaire se fait comme suit:

$$\min B = \sum_{i,j} z_{ij}^+ + \sum_{i,j} z_{ij}^-$$

Cette formule représente la fonction objectif du problème. Les variables z_{ij}^+ et z_{ij}^- sont des variables dites artificielles. Elles sont toutes les deux positives et servent à compenser les écarts de la solution inconsistante par rapport à la solution consistante.

La première contrainte à respecter est la suivante:

$$p_{ij}w_j - w_i - z_{ij}^+ + z_{ij}^- = 0$$

Sans les variables artificielles, cette expression représente donc le problème en cas de consistance parfaite.

La solution de ce problème fournit les poids relatifs des éléments de décision de la hiérarchie.

Pour obtenir les résultats globaux à partir des poids relatifs, il suffit de procéder de la même façon que pour la méthode des vecteurs propres.

Tous ces calculs sont surtout pertinents si la hiérarchie est incomplète. Quand elle est complète on peut aussi faire une estimation simultanée des priorités mais les relations multiplicatives entre les différents niveaux de la hiérarchie rendent le travail plus complexe.

Cette méthode s'avère bonne puisqu'elle garantit une solution globale optimale et qu'elle fournit une mesure de la consistance à travers B.

Mais on ne sait pas encore si elle est meilleure que celle des vecteurs propres en cas d'inconsistance. De même, on ignore encore quels sont exactement les avantages et les désavantages de cette nouvelle méthode. Ce sont les expériences et la pratique qui nous le diront.

6.5.5 Comparaison des quatre méthodes : VP,MC,MCL et PL

Toutes ces méthodes fournissent le même résultat lorsque la matrice des comparaisons est consistante. Ce n'est plus le cas lorsqu'elle est inconsistante et on ne sait alors pas dire a priori quelle est la meilleure méthode. C'est cela que nous allons essayer de déterminer.

6.5.5.1 Au point de vue de l'unicité de la solution

Saaty et d'autres auteurs s'accordent à dire que la méthode des vecteurs propres est supérieure aux autres méthodes.

D'abord parce qu'elle fournit une solution unique. En effet, les priorités entre les alternatives respectent un et un seul ordre: celui donné par une colonne normalisée de la matrice lorsque celle-ci est consistante ou celui donné par le calcul d'une limite lorsque ce n'est pas le cas.

D'après Saaty, la technique des moindres carrés et celle des moindres carrés logarithmiques ne fournissent pas de solution unique. Il est alors difficile et hasardeux de se fier à une solution plutôt qu'à une autre.

Cependant, si en théorie, une matrice inconsistante A traitée par les moindres carrés engendre des solutions multiples, en pratique, cela n'arrive pas souvent.

En ce qui concerne la programmation linéaire, nous pouvons dire qu'elle fournit aussi une solution unique quand on élimine les réponses triviales, w_i et w_j nuls.

On peut dire que, à ce point de vue, à part celle des moindres carrés logarithmiques, toutes les méthodes semblent fiables.

6.5.5.2 Au point de vue de la mesure de la consistance

Nous venons de voir que les vecteurs propres tiennent compte de la consistance: ils distinguent une matrice consistante d'une matrice inconsistante et, dans ce dernier cas, ils tentent de déterminer les priorités telles qu'elles seraient sans inconsistance dans les jugements. Cette situation étant fréquente - dans la pratique on pourrait même dire qu'elle est la règle - cette propriété des VP s'avère d'une très grande importance lors de la prise de décision.

Précisons cependant que ces inconsistances ne sont tolérées que jusqu'à un certain degré. Au delà d'un rapport d'inconsistance égal à dix pourcents, les données sont considérées comme inutilisables et le choix final aurait peu de chance d'être le bon.

Si cela s'avère être le cas, il reste une solution: le programme permettant de calculer le vecteur propre normalisé fournit non seulement une mesure de la consistance, mais permet également de la rectifier. C'est un gain de temps et de

fiabilité important, qui n'est pas permis avec les moindres carrés logarithmiques et avec les moindres carrés.

La programmation linéaire a elle aussi un indicateur de consistance: à partir de la valeur objectif B , on calcule C compris entre 0 et 1.

$$C = B / (B+1)$$

6.5.5.3 Au point de vue de la stabilité de la solution

Qu'en est-il de la stabilité de la solution donnée par les différentes méthodes? Rappelons qu'une bonne technique d'estimation est celle qui, lors d'une légère modification des données, n'enregistre pas ou peu de perturbations dans le résultat (l'ordre de priorité entre les alternatives).

C'est ici que se manifeste la supériorité des vecteurs propres sur les autres méthodes: les VP ont cette propriété de ne pas bouleverser l'ordre même lorsque les jugements sont inconsistants; ils permettent d'éviter le rank reversal. Ceci n'est pas systématiquement vrai pour les autres méthodes.

Ceci est l'argument principal en faveur des vecteurs propres: tandis que cette méthode fait la différence entre une matrice consistante et une matrice inconsistante, les autres techniques l'ignorent.

Le défaut des MC et des MCL est que l'on part d'un processus de pondération des lignes de la matrice et que donc, si on introduit une modification dans ces lignes, cela se répercute sur le classement des alternatives. D'où la mesure de stabilité et donc le manque de fiabilité qui handicapent ces méthodes.

De nombreux auteurs (Vargas par exemple) utilisent dans leur article la théorie des graphes pour démontrer les aspects essentiels suivants:

- le vecteur propre normalisé est la meilleure technique pour déterminer la dominance entre les alternatives;
- ce même vecteur propre a la capacité de préserver le rang (ou de ne pas le bouleverser) en cas d'inconsistance dans les jugements.

Nous ne développerons pas cette théorie des graphes ici, mais il est tout de même important de souligner qu'elle utilise le concept de dominance comme principe sous-jacent pour dériver l'échelle des priorités. C'est ce principe de dominance qui distingue les vecteurs propres des autres méthodes. Les MC et MCL ainsi que la programmation linéaire ne sont pas ainsi fondées sur un concept de décision; elles ne font que minimiser des écarts, ce qui ne veut rien dire en tant que tel car il faut encore interpréter leur résultat.

6.5.5.4 Conclusion

Après ces comparaisons entre les quatre méthodes, il est difficile de conclure à la supériorité de l'une d'elles en termes de validité prédictive.

Cependant, nous pouvons dire que le vecteur propre normalisé est la seule et unique façon de percevoir la dominance dans la matrice en cas d'inconsistance. De plus, elle reste la seule méthode (avec la programmation linéaire) à fournir une mesure de cette inconsistance grâce à la valeur propre associée au vecteur propre, λ_{\max} . Et la facilité de calcul que procure un système automatisé des calculs est un atout supplémentaire.

Pour terminer, nous reprenons les sept principales raisons énoncées par Saaty en faveur des vecteurs propres:

1. Ils nous donnent une solution unique.
2. Il ne faut pas se laisser charmer par des méthodes trop simples et pas toujours bonnes.
3. Le renversement de rang n'est pas toujours aussi légitime avec d'autres méthodes.
4. Le concept de la dominance est préféré à la minimisation des erreurs.
5. La procédure des vecteurs propres est descriptive, toutes les autres sont techniquement prescriptives et impliquent la minimisation.
6. Elle est directement analytique et non statistique car il existe une relation entre la solution et la mesure de consistance qui ne dépend pas d'une distribution de probabilités.
7. La pondération successive des critères d'après l'importance mène au vecteur propre.

Tous ces points semblent faire pencher la balance en faveur de la solution fournie par les vecteurs propres: c'est une solution stable et fiable. Celle fournie par les moindres carrés et par la programmation linéaire me semble moins bonne, si ce n'est en cas d'inconsistance.

CHAPITRE 7

IMPLEMENTATION ET STRUCTURATION DU PROBLEME

Chapitre 7

Implémentation et structuration du problème

7.1 Introduction

Ce chapitre sera composé de deux parties. La première expliquera comment les données sont représentées au niveau de la mémoire afin de pouvoir les manipuler facilement. La seconde quant à elle nous expliquera le fonctionnement interne de la partie programmation. Pour des détails d'implémentation, on pourra toujours faire référence au code des différentes procédures et fonctions dans les annexes.

7.2 Représentation des données

La représentation des données est toujours un problème épineux dans tout système informatique car c'est de celle-ci que peut dépendre la complexité du problème. En effet si cette représentation est inadéquate, le programmeur peut se retrouver devant des problèmes insurmontables.

On doit diviser la représentation des données en deux parties. La première est la représentation au niveau de la base de données tandis que la seconde se situe au niveau de la mémoire.

7.2.1 *Au niveau de la base de données*

La hiérarchie des symptômes ainsi que les remèdes associés à ces différents symptômes doivent être enregistrés au niveau de la base de données. Ce point n'a pas été développé car la société Archibel a mis au point une couche englobant la base de données. Donc on ne fait jamais d'accès direct à la base de données mais on utilise ces fonctions.

7.2.2 Au niveau de la mémoire

On doit représenter au niveau de la mémoire, d'une part des symptômes sélectionnés pour les différents clipboards, d'autre part, pour chaque symptôme, on doit avoir la liste des remèdes associés et leur degré respectif. De plus, on doit avoir une structure qui contiendra la liste des remèdes triés selon la méthode de l'A.H.P.. Un clipboard est un endroit où l'on peut collectionner les symptômes et les analyser. Radar dispose de dix clipboards utilisables par le médecin. Celui-ci pourra se servir de ces clipboards pour affiner son diagnostic

7.2.2.1 Représentation des Clipboards

Pour représenter les clipboards en mémoire, on utilise un tableau de dix éléments. Chaque cellule du tableau sera composée

- du nombre de symptômes sélectionnés;
- du nom du clipboard courant;
- d'un pointeur vers le premier symptôme sélectionné (premier symptôme de la liste chaînée);
- d'un pointeur vers le dernier symptôme sélectionné.

Grphe conceptuel des SYMPTOM_CLIPBOARD (cfr Figure 12)

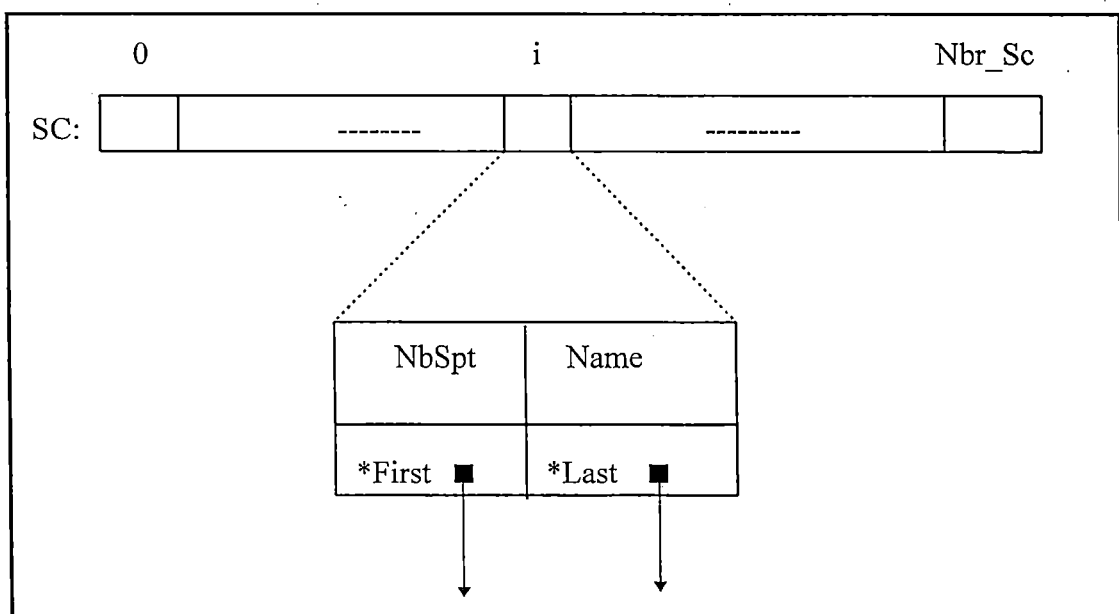


Figure 12: Graphe conceptuel des SYMPTOM_CLIPBOARD

Définition en C de la structure

```
typedef struct {  
    char                InMemory;  
    unsigned short      NbSpt;  
    char                *Name;  
    SYMPTOM_IN_MEMORY  *First;  
    SYMPTOM_IN_MEMORY  *Last;  
    #ifdef DEBUG  
    unsigned char       Check;  
    #endif  
} SYMPTOM_CLIPBOARD;
```

7.2.2.2 Représentation des symptômes sélectionnés

Pour représenter les symptômes sélectionnés en mémoire, on utilisera une liste chaînée. Chaque cellule sera composée

- du nom du symptôme dans la langue sélectionnée;
- du nom du symptôme dans la deuxième langue;
- d'un pointeur vers un tableau de remèdes associés;
- du nombre de remèdes associés au symptôme;
- d'un pointeur vers le symptôme suivant de la liste.

Grappe conceptuel des SYMPTOM_IN_MEMORY(cfr Figure 13)

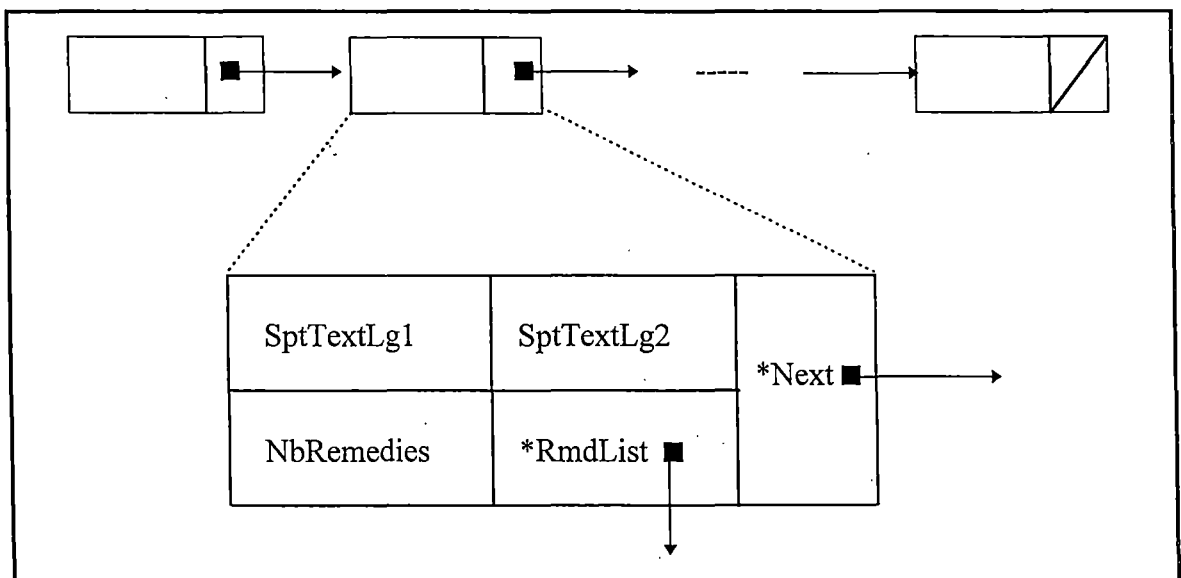


Figure 13: Grappe conceptuel des SYMPTOME_IN_MEMORY

Définition en C de la structure

```
typedef struct {  
  
    unsigned char    *SptTextLg1;  
    unsigned char    *SptTextLg2;  
    char             Intensity;  
    unsigned char    Group;  
    REMED_LIST       *RmdList; /* Pointer to a table of Remedy List */  
    unsigned char    Degrees;  
    unsigned char    Qualif;  
    unsigned short   NbRemedies;  
    char             Langue[2];  
    unsigned char    Origin;  
    unsigned short   View; /*6006*/  
    void             *Next; /* Pointer to the next Spt in memory */  
#ifdef DEBUG  
    unsigned char    Check;  
#endif  
  
} SYMPTOM_IN_MEMORY;
```

7.2.2.3 Représentation des remèdes associés à un symptôme

Pour représenter les remèdes associés en mémoire, on utilisera un tableau. Chaque cellule sera composée

- de l'identifiant du remède
- du degré du remède
- de l'auteur du remède

Grappe conceptuel des REMED_LIST(cfr Figure 14)

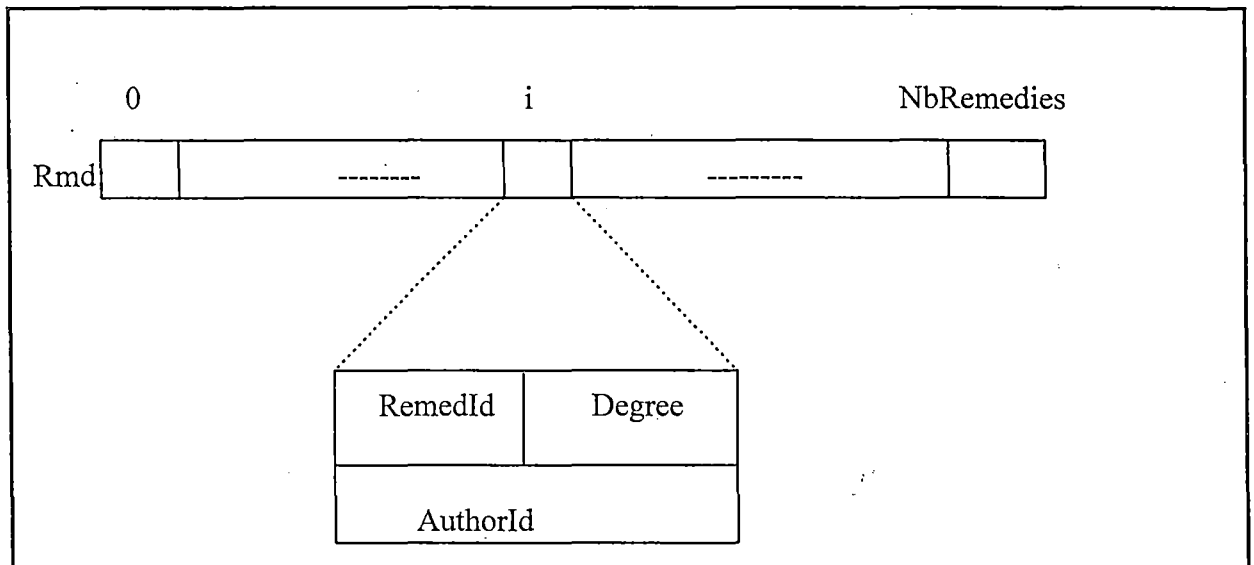
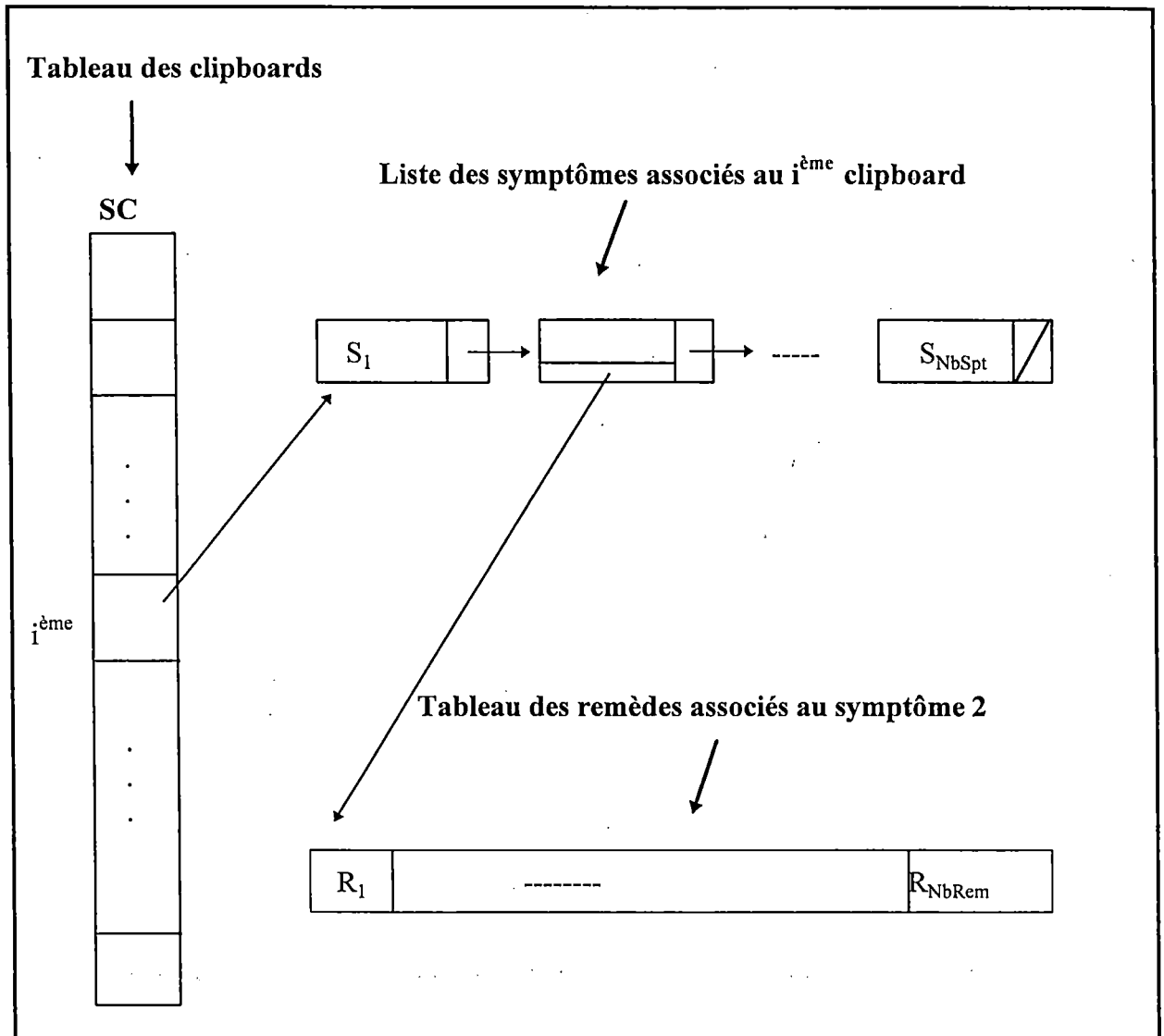


Figure 14: Grappe conceptuel des REMED_LIST

Définition en C de la structure

```
typedef struct {  
  
    unsigned short    RemedId;  
    unsigned char     Degree;  
    unsigned short    AuthorId;  
    #ifdef DEBUG  
    unsigned char     Check;  
    #endif  
  
} REMED_LIST;
```

Grphe général



7.2.2.4 Représentation des résultats

Pour représenter les résultats, on utilisera un tableau (RemedyScore) comprenant la totalité des remèdes. Chaque cellule sera composée

- de l'identifiant du remède
- du score du remède
- de l'auteur du remède

Grphe conceptuel des REMEDY_SCORE(cfr Figure 15)

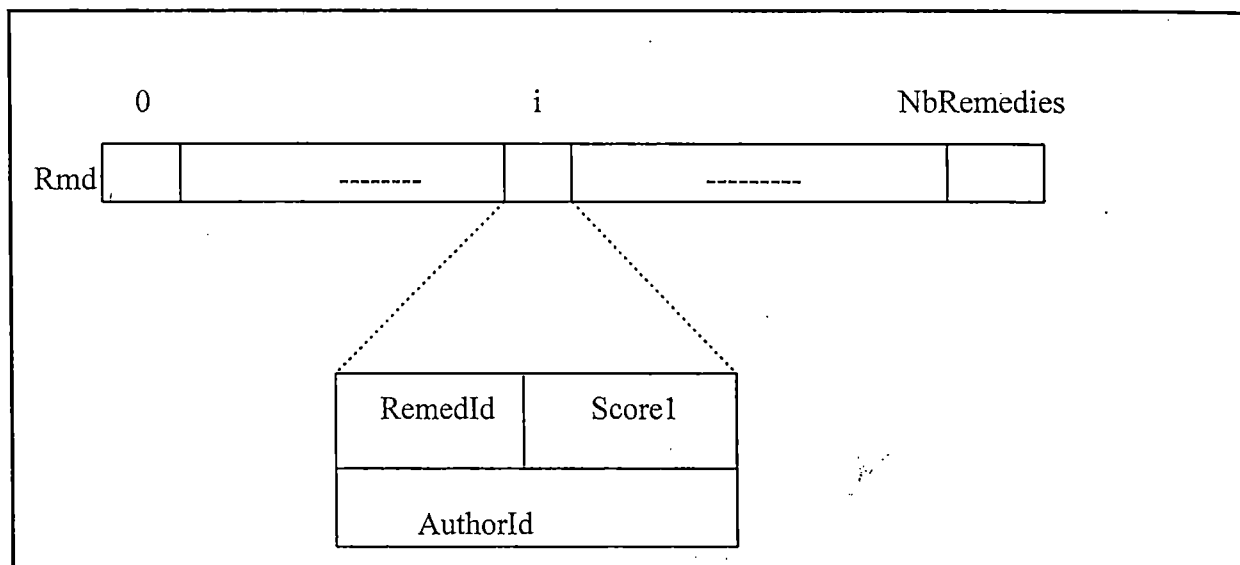


Figure 15: Grphe conceptuel des REMEDY_SCORE

Définition en C de la structure

```
typedef struct {  
    unsigned short    RemedId;  
    float             Score1;  
    float             Score2;  
    int               Flag;  
    #ifdef DEBUG  
    unsigned char     Check;  
    #endif  
} REMEDY_SCORE;
```

7.3 Implémentation du problème (Algorithmes)

Voici la liste des principales variables et constantes qui vont être utilisées pour implémenter le problème.

Soit

nbre_max_symp = 10.
nbre_max_rem = 50.

SYMPTOM_CLIPBOARD	SC[10] tableau des clipboards.
float	m_symptômes[nbre_max_symp][nbre_max_symp] matrice des comparaisons par paires des symptômes.
float	v_propre_symptômes[nbre_max_symp] vecteur propre correspondant à la valeur propre dominante.
float	val_propre valeur propre dominante pour les matrices de comparaisons par paires.
float	_ic indice de consistance pour la matrice de comparaison par paires des symptômes.
float	v_propre_remedes[nbre_max_rem] vecteur propre correspondant à la valeur propre dominante.
float	m_remedes[nbre_max_rem] matrice des comparaisons par paires des remèdes.
int	echelle_deg[5] poids pour la comparaison par paires des remèdes
REMEDY_SCORE	RemedyScore[Nbre_Remedes] tableau reprenant les scores des remèdes après analyse.

Le développement de la méthode de Saaty selon le problème traité se décompose en traitements successifs que voici:

Algorithme

Step 1) Sélection des symptômes pour les différents clipboards
goto 2.

Step 2) Comparaison par paires des symptômes
goto 3.

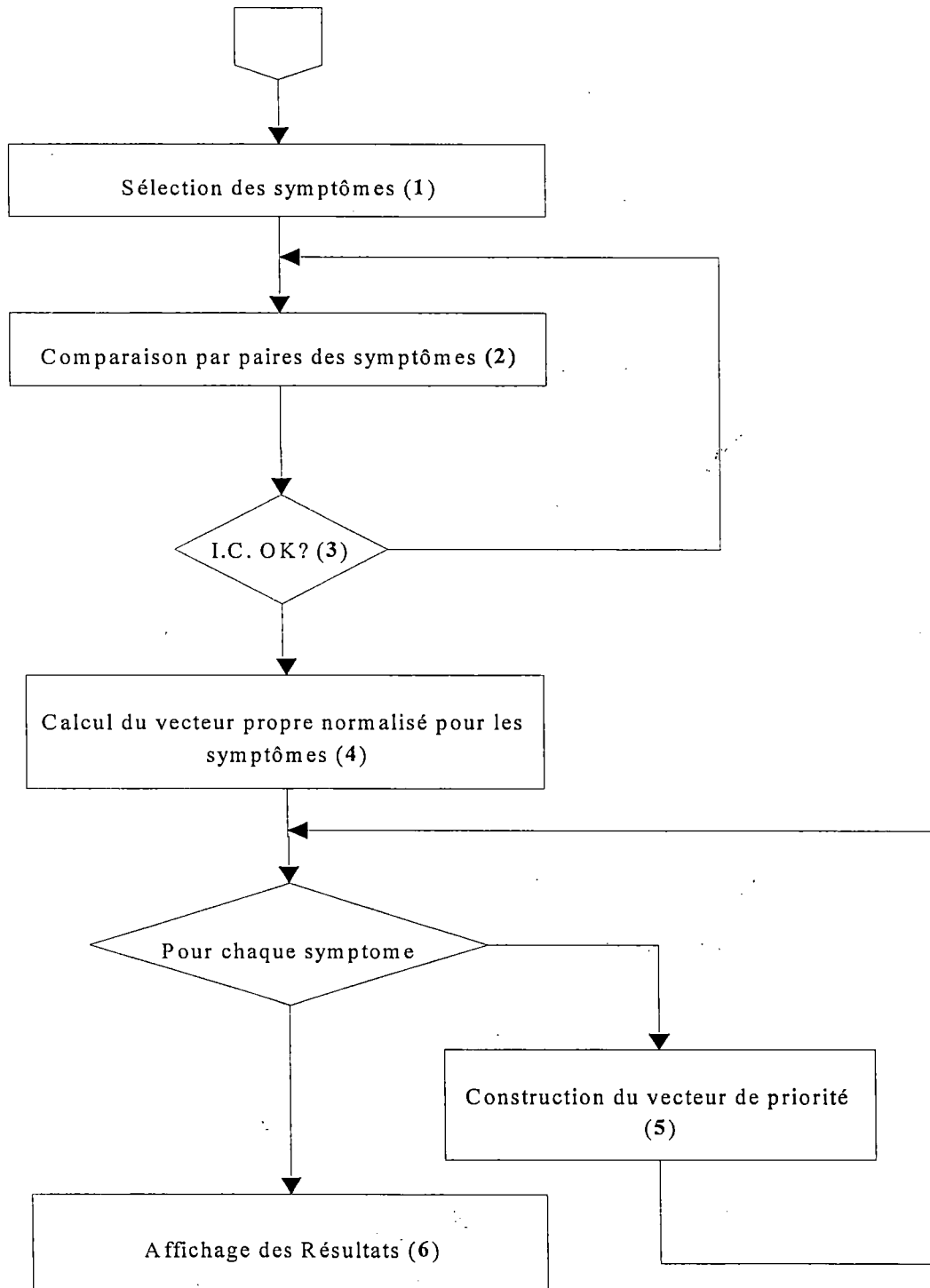
Step 3) Vérification de l'indice de concordance
Si l'indice est Ok alors goto 4
sinon goto 2.

Step 4) Construction du vecteur propre normalisé
goto 5.

Step 5) Pour chaque symptôme construction du vecteur propre des
remèdes et du vecteur de priorité
goto 6.

Step 6) Affichage des résultats.

Algorithme principal



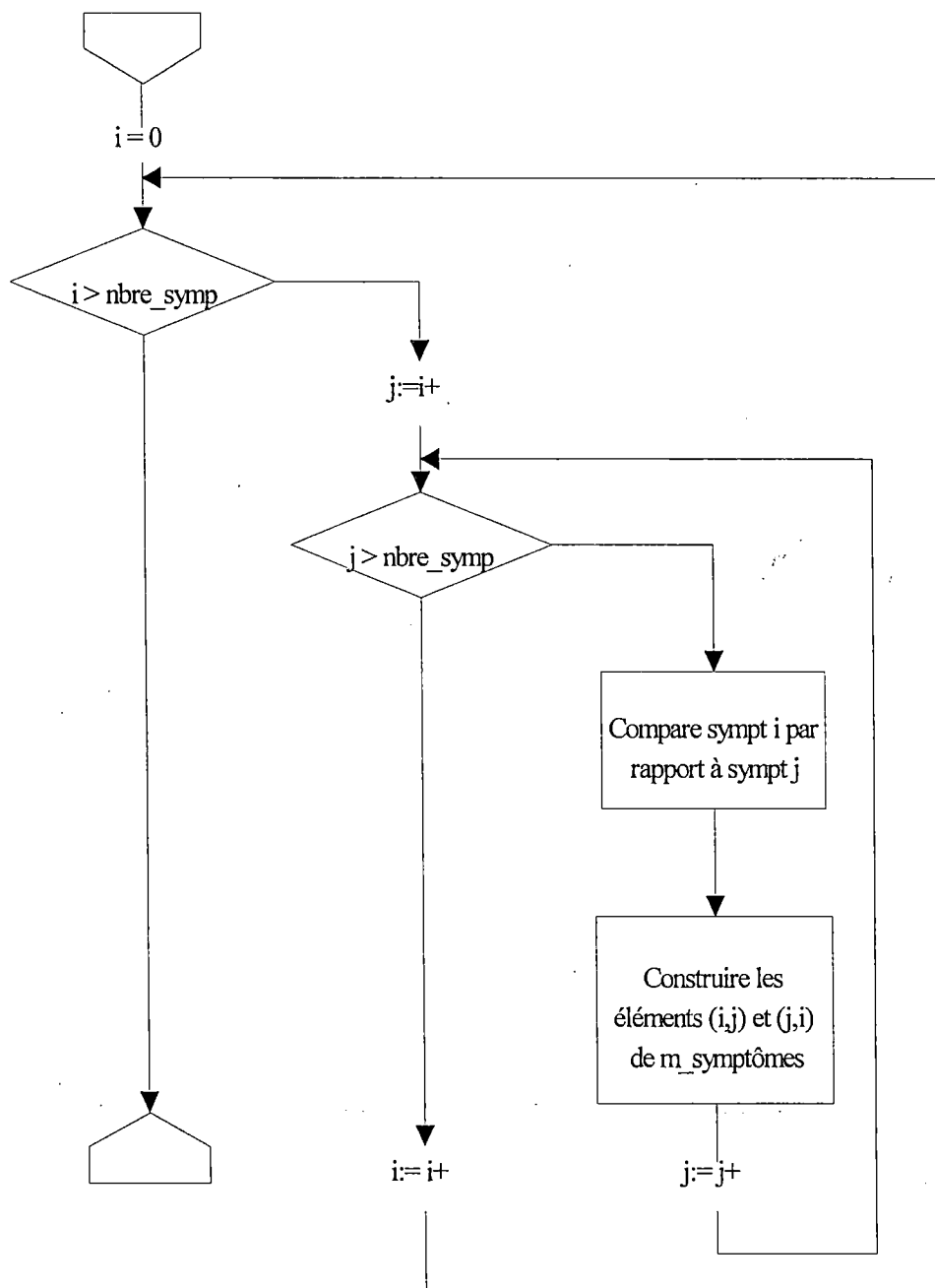
Step 1) Sélection des symptômes

La sélection des symptômes a pour but de remplir le tableau SC[10]. L'utilisateur en sélectionnant les symptômes doit préciser dans quel clipboard il veut le placer. Après ce traitement la structure expliquée dans la partie représentation des données sera initialisée selon le souhait de l'utilisateur.

Step 2) Comparaison par paires des symptômes

L'utilisateur doit comparer les symptômes deux à deux et donner un poids de préférence pour chaque paire. Cette étape effectuée par le médecin doit être réalisée en ayant connaissance de l'échelle de Saaty. Si on doit comparer le symptôme s1 avec le symptôme s2 et que le médecin estime que le symptôme s1 est plus important que le symptôme s2, il donnera un poids entier positif supérieur à 1. Mais si le symptôme s2 est préféré au symptôme s1 alors il donnera un poids entier négatif inférieur à -1. On procède de la sorte pour que le médecin ait plus de facilité à comparer les symptômes. De cette manière on comparera le premier symptôme avec tous les autres, puis le second avec tous les autres moins le premier et ainsi de suite.

Cette étape initialisera la matrice m_symptômes. Le schéma suivant illustre cette étape.



Step 3) Vérification de l'indice de cohérence

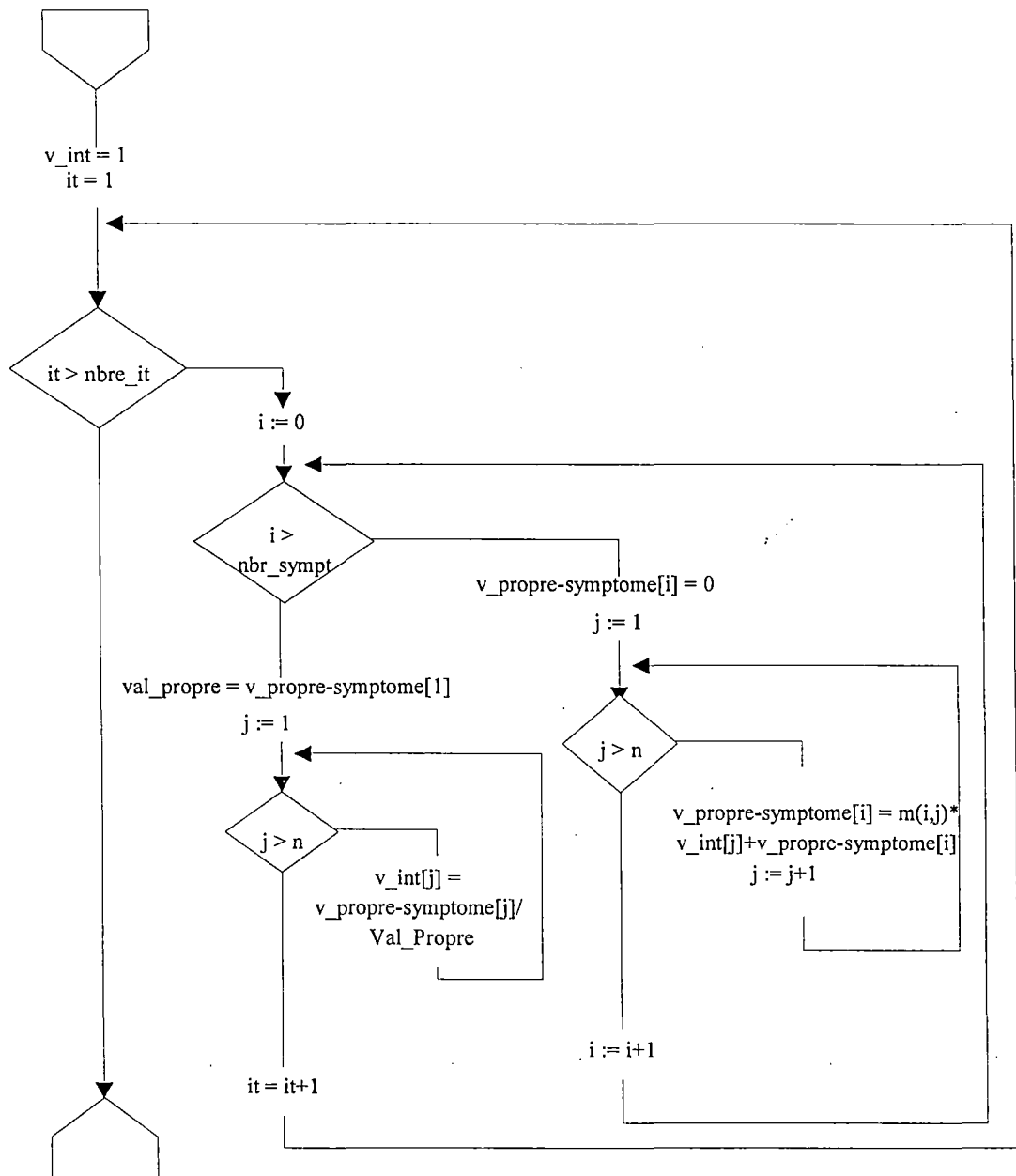
Cette étape est constituée d'une part du calcul de la valeur propre dominante grâce à la méthode des puissances et d'autre part du calcul de l'indice de cohérence. Le calcul de la valeur propre est nécessaire pour le calcul de l'indice de cohérence.

$$IC = (\text{valeur_propre} - \text{nbr_symp}) / (\text{nbr_symp} - 1)$$

Si cette valeur est supérieure à un taux de cohérence émis par l'utilisateur, on doit répéter le point 2. Et pour calculer la valeur propre dominante on doit calculer le vecteur propre associé à cette valeur. Le schéma suivant illustre ce calcul.

Soit

- nbr_sympt = dimension de la matrice carrée des comparaisons par paires (n dans le schéma)
- nbre_it = paramètre de la méthode des puissances indiquant le nombre d'itérations choisies par l'utilisateur.
- v_int = vecteur intermédiaire
- $\text{v_propre_symptômes}$ = vecteur nécessaire pour le calcul du vecteur propre
- m_symptômes = matrice de comparaison par paires des symptômes (m dans le schéma)

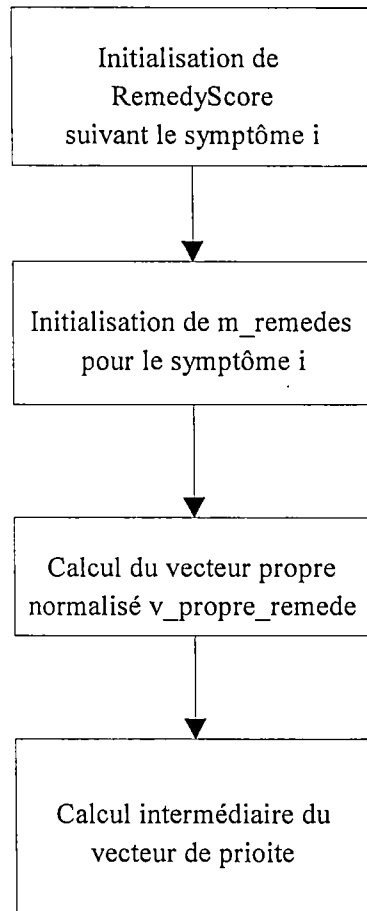


Step 4) Calcul du vecteur propre normalisé pour les symptômes

On calcule la somme des éléments du vecteur propre des symptômes et on divise chaque élément par cette somme. Il ne faut pas être plus explicite pour cette étape.

Step 5) Construction du vecteur de priorité

Pour chaque symptôme sélectionné, on construira le vecteur de priorité. Pour réaliser cette opération, on suivra le schéma suivant



La première étape consiste à initialiser le champs *Score2* de la table *RemedyScore* à la valeur des degrés correspondant au symptôme courant. Si un remède n'est pas repris pour un symptôme, son degré est zéro. Donc, on aura la table *RemedyScore* qui contiendra pour le symptôme courant le degré approprié.

La seconde étape consiste à initialiser la matrice de comparaisons par paires des remèdes. Pour cette initialisation, on utilise le tableau *echelle_deg* qui correspond au poids attribué par l'utilisateur pour la comparaison entre deux remèdes. Si on doit comparer R_i de degré d_i avec R_j de degré d_j , on effectue la différence des degrés (*diff*). Si *diff* est positif, on attribue le poids *echell_deg[diff]* à l'élément *m_remedes[i][j]* et $1/echell_deg[diff]$ à l'élément *m_remedes[j][i]* sinon *echell_deg[-diff]* à l'élément *m_remedes[j][i]* et $1/echell_deg[diff]$ à l'élément *m_remedes[i][j]*.

La troisième étape consiste à calculer le vecteur propre des remèdes. Celle-ci est réalisée de la même façon que le calcul du vecteur propre des symptômes. Donc on ne va pas s'étendre sur le sujet.

La dernière étape consiste à construire progressivement le vecteur de priorité. Le vecteur de priorité est en fait compris dans la table *RemedyScore* et plus précisément dans le champs *Score1*. On calcule le vecteur de priorité selon le vecteur propre des remèdes et le symptôme *i*. On construit alors le vecteur de priorité de la façon suivante:

$$RemedyScore[j].Score1 = RemedyScore[j].Score1 + v_propre_remede[j] * v_propre_symptome[i]$$

Remarque:

Si le nombre de remèdes sélectionnés est supérieur à un certain seuil défini pour des raisons de performances, le calcul du vecteur propre des remèdes s'effectue selon une méthode d'approximation. En effet, comme le problème qui nous préoccupe est assez précis, il y a moyen de mettre sur pied une méthode qui utilise une échelle absolue qui peut être modifiée par l'utilisateur. Pour un degré donné, on donne un poids entre 0 et 2. On divise ce poids par le nombre de remèdes sélectionnés. Et ensuite on normalise le vecteur. Ce vecteur approximera de manière plus que convenable le vecteur propre évidemment si les poids sont correctement évalués.

Step 6) Tri décroissant et affichage des résultats

La table *RemedyScore* contient pour chaque remède le poids accordé par la méthode de Saaty dans le champs *Score1*. Pour afficher les résultats à l'écran, on utilise les fonctions existantes dans Radar. Pour de plus amples informations, les annexes pourront vous éclairer.

CHAPITRE 8

CRITIQUES PAR UN MEDECIN HOMEOPATHE

Chapitre 8

Critiques par un homéopathe

Après avoir implémenté et intégré l'Analytic Hierarchy Process dans le logiciel Radar, j'ai pensé qu'il serait très intéressant de présenter cette méthode d'aide à la décision à un médecin homéopathe. Voilà pourquoi j'ai demandé à Monsieur Storet de Mussy-la-Ville de critiquer cette nouvelle méthode d'aide à la décision à travers l'utilisation du logiciel Radar.

Nous allons tout d'abord essayer de comprendre comment ce médecin homéopathe raisonne lorsqu'il doit choisir un remède susceptible de guérir un patient. Il est important de signaler que Monsieur Storet n'utilise aucun outil informatique que ce soit pour gérer les dossiers de ses patients ou pour établir un diagnostic.

8.1 Quelques règles de raisonnement suivies par le médecin homéopathe lors d'une consultation

8.1.1 Les catégories de symptômes

Monsieur Storet nous a fait remarquer que les symptômes étaient classés dans plusieurs catégories. Chaque catégorie est plus ou moins importante par rapports aux autres catégories. Voici les cinq catégories classées selon leur importance.

1. Les symptômes étiologiques
2. Les symptômes psychologiques
3. Les symptômes généraux
4. Les symptômes bizarres
5. Les symptômes directement liés à la maladie

Comme on peut le constater, les médecins homéopathes accordent beaucoup plus d'importance aux symptômes étiologiques, c'est-à-dire aux causes de la maladie, qu'aux symptômes directement liés à la maladie. Voici un cas clinique illustrant l'ordre d'importance des différentes catégories de symptômes.

Prenons l'exemple d'un enfant souffrant d'une otite. En discutant avec lui lors de la consultation, le médecin homéopathe découvre que l'enfant a subi un choc émotionnel en se disputant violemment avec ses parents et qu'il éprouve depuis lors de grosses difficultés à communiquer avec eux. Pour guérir l'otite, le médecin homéopathe traitera d'abord la cause de la maladie, c'est-à-dire le choc causé par la dispute avec ses parents, ensuite il devra tenir compte du symptôme psychologique ou plus précisément du problème de communication entre son patient et les parents de celui-ci. Il tiendra compte du fait que l'enfant souffre d'une oreille uniquement après avoir traité les symptômes étiologiques et psychologiques.

Critiques:

Pour traiter un cas clinique avec le logiciel Radar nous n'avons que des symptômes directement liés à la maladie (dernière catégorie suivant l'ordre de préférence) à notre disposition. Les exemples de cas cliniques que nous rencontrerons par la suite ne reflèteront donc pas vraiment la réalité.

8.1.2 Le classement des symptômes par le médecin et le choix du remède adéquat

Monsieur Storet va classer les symptômes rencontrés durant la consultation d'un patient suivant l'ordre d'importance établi dans le paragraphe précédent.

Ensuite en consultant le répertoire de Kent, il établit la liste des remèdes correspondant aux symptômes sélectionnés ainsi que leur degré respectif. Finalement, il choisit un remède à prescrire au patient en tenant compte de l'ordre de préférence des symptômes et du degré d'efficacité attaché à chaque remède.

Critiques:

La méthode de Saaty permet de faire la même chose d'une manière différente. En effet, en comparant les symptômes deux par deux, le médecin homéopathe peut accorder plus ou moins d'importance à un symptôme. Concrètement si il juge que le symptôme A est beaucoup plus important que le symptôme B, il lui suffit de regarder sur l'échelle de Saaty la valeur qui correspond à son jugement de comparaison.

Remplir la matrice de comparaisons revient donc à classer les symptômes par ordre d'importance. Le vecteur propre de cette matrice de comparaisons correspond au classement des symptômes.

Le classement obtenu par la méthode de Saaty est en réalité plus "fin" que celui fait à la main. En effet, il est plus facile de comparer des symptômes deux

par deux plutôt que de classer toute une série de symptômes par ordre de préférence (surtout si les symptômes sont nombreux).

8.1.3 Le nombre de symptômes sélectionnés

En général, Monsieur Storet sélectionne très peu de symptômes (trois ou quatre maximum).

Critiques:

La méthode de Saaty fonctionne très bien ici puisqu'il y a peu de symptômes sélectionnés. En effet, moins il y a de symptômes, moins il y a de comparaisons à réaliser et plus la dimension de la matrice de comparaisons est petite. Ceci entraîne une très grande efficacité de la méthode.

La difficulté de gérer manuellement de longues listes de remèdes contraint peut-être Monsieur Storet à travailler avec trois ou quatre remèdes tout au plus. En effet, plus le nombre de symptômes sélectionnés est grand, plus la liste de remèdes s'allonge et ceci ne facilite en rien le choix du remède adéquat.

8.2 Exemple d'un cas clinique et pondération

N'ayant pas à notre disposition l'entièreté du répertoire de Kent nous avons été obligé d'adapter un cas clinique aux cinq chapitres suivants:

1. Hearing
2. External Throat
3. Kidneys
4. Prostate
5. Urine

Voici trois symptômes sélectionnés par Monsieur Storet:

1. Kidneys-Pain-Right
2. External Throat-Torticollis
3. Urine-Sediment

Pour remplir la matrice de comparaisons par paires nous avons posé les questions suivantes à Monsieur Storet:

1. Par rapport à la santé du patient, le symptôme "Kidneys-Pain-Right" est-il plus influent que le symptôme "External Throat-Torticollis" et avec quel intensité?
2. Par rapport à la santé du patient, le symptôme "Kidneys-Pain-Right" est-il plus influent que le symptôme "Urine-Sediment" et avec quel intensité?
3. Par rapport à la santé du patient, le symptôme "External Throat-Torticollis" est-il plus influent que le symptôme "Urine-Sediment" et avec quel intensité?

A l'aide de l'échelle de Saaty, Monsieur Storet est parvenu à chiffrer ses réponses. Celles-ci sont résumées dans la Figure 16.

Symptômes	Kidneys-Pain-Right	External Throat-Torticollis	Urine-Sediment
Kidneys-Pain-Right	1	1/2	1
External Throat-Torticollis	2	1	2
Urine-Sediment	1	1/2	1

Figure 16: Matrice des comparaisons par paires des symptômes

Après avoir lancé la méthode d'analyse on obtient le classement suivant:

The screenshot shows the RADAR 6.1 for Windows interface. The main window is titled "Analysis (1 Clipboard) - untitled". It contains a table with the following data:

A.H.P.	1	2	3	4	5	6	7	8	9	10	11
	lve.	phos.	lachn.	calc.	caust.	cupr.	graph.	rhus-t.	ars.	mez.	sulph.
20	18	17	14	14	14	14	14	14	13	13	13
1. Nieren - Schmerz - rechts	1	-	-	-	-	-	-	-	-	-	-
2. Äußerer Hals - Torticollis	1	■	■	■	■	■	■	■	□	□	□
3. Urin - Sediment	1	■	■	□	□	□	□	□	■	■	■

Below the table, there are buttons for "Close", "Options", "Symptoms", and "Help". A status bar at the bottom indicates "This analysis contains 95 remedies and 3 symptoms. Intensity is considered" and lists various remedy characteristics:

- chocolate-colored: (1)
- circles: (3)
- clay-colored: (15) Am-c. Am-m. Anac. BERB. Sars. Sep. Sulph. ZINC.
- adhesive: (1) Sep.
- menses, difficult to wash off; during: (1) Sep.
- clots, yellow-red: (1)

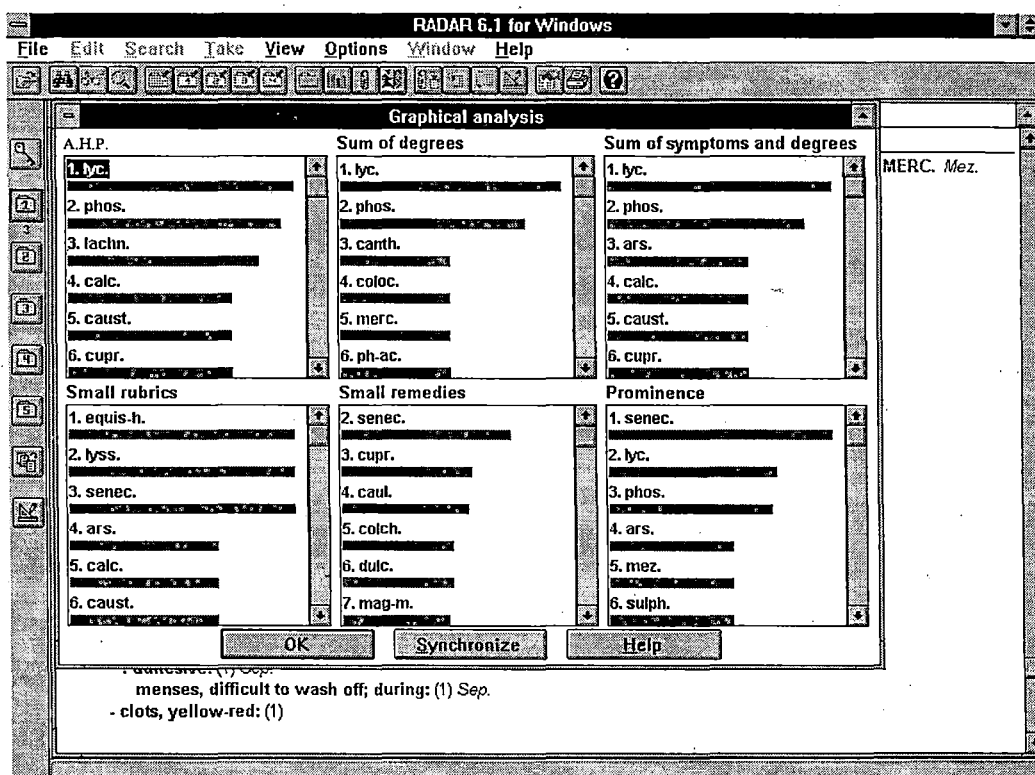
Critiques:

Monsieur Storet a uniquement utilisé les degrés 1 et 2 de l'échelle de Saaty pour comparer les symptômes. Peut-être est-ce dû au fait que tous les symptômes sélectionnés appartiennent à la même catégorie: celle des symptômes directement liés à la maladie.

Monsieur Storet avait pronostiqué comme remède le "Phosphorus" pour son diagnostic. Mais à la lecture des résultats, il a reconnu que le "Lycopodium Clavatum" était tout aussi valable. Ce qui d'ailleurs le fit changer d'avis après réflexion.

8.3 Comparaisons avec les autres méthodes d'analyse de Radar

Voici les résultats obtenus avec les autres méthodes d'analyse pour le cas clinique du paragraphe précédents.



On peut remarquer que la plupart des méthodes donnent un résultat similaire à l'A.H.P.. Mais malheureusement on n'a pas pu investiguer davantage dans les comparaisons avec les autres méthodes car le manque de chapitre mis à notre disposition s'est révélé cruel.

8.4 Temps de réponses

Les tests de performances ont été réalisés sur la configuration suivante: On disposait d'un PC Olivetti M4 66 Modulo muni d'une carte mère 80486 DX2-66 Mhz et de 20 MégaBytes de mémoire. Le disque dur était d'une capacité de 500 MégaBytes et d'un temps d'accès de 12 ms.

Les temps de réponses sont performants. Il ne nous a pas semblé opportun de les calculer précisément car ils étaient de l'ordre du *temps de réponse humain*. Le temps pour encoder les comparaisons par paires est énormément plus long que le temps de calcul.

8.5 Conclusions

L' Analytic Hierarchy Process correspond à certaines attentes du médecin homéopathe. En effet, les résultats sont cohérents avec les résultats qu'aurait obtenus Monsieur Storet de manière manuelle et lourde.

L'utilisation d'une méthode d'aide à la décision et en particulier l'utilisation de l'A.H.P. permet de susciter la réflexion chez le médecin homéopathe. En effet, en obtenant des résultats complets, il peut remettre ses propres jugements en question. Le travail d'analyse demandé à un médecin homéopathe est toujours très lourd.

Les temps de réponses sont plus que satisfaisants et permettent à l'homéopathe de rectifier rapidement ses jugements.

Le nombre de méthodes d'aide au diagnostic n'est pas pénalisant car les temps de réponses sont très bons et permettent à l'homéopathe une plus grande suggestion de remèdes. Mais l'excès nuit en tout.

Malgré tout, Monsieur Storet a manifesté une certaine appréhension en ce qui concerne l'utilisation de l'échelle de Saaty. Celle-ci est d'ailleurs bien soulevée par le fait qu'il n'a choisi que deux degrés particuliers de l'échelle. Cependant après un certain temps d'adaptation à l'échelle de Saaty, l'A.H.P. peut devenir très performante.

En conclusion, l'A.H.P. est une méthode d'aide à la décision, le choix final du remède est toujours du ressort de l'homéopathe. Il ne faut pas croire que cette méthode prescrit le remède adéquat sans aucune réflexion de l'utilisateur.

CHAPITRE 9

**PRESENTATION
DES
OUTILS UTILISES**

Chapitre 9

Présentation des outils utilisés

9.1 Introduction

Pour implémenter la méthode de Saaty nous avons utilisé plusieurs outils. Ceux-ci ont été choisis par la société Archibel qui commercialise et développe le logiciel Radar.

Le problème du stockage des symptômes et des remèdes a été résolu par l'utilisation du gestionnaire de bases de données Ctree. Celui-ci est bien adapté au problème qui nous concerne puisque c'est un gestionnaire de bases de données hiérarchiques.

La partie programmation a été écrite en C. Nous ne vous ferons pas l'affront de décrire ce langage.

Pour réaliser l'interface homme/machine, XVT-Design a été choisi pour sa portabilité sur différents systèmes d'exploitation et sa facilité d'intégration avec le C. On expliquera dans la suite de ce chapitre les grandes lignes de ce dernier outil.

9.2 Qu'est-ce que XVT-Design ?

XVT-Design est un outil de design interactif et graphique. XVT-Design peut également servir de générateur d'application.

Il simplifie le design et l'implémentation des interfaces graphiques utilisateurs de trois façons :

1. XVT-Design permet de créer les objets de l'interface utilisateur de manière graphique et interactive plutôt que de la programmer manuellement.
2. XVT-Design fournit un TestMode permettant d'avoir un aperçu de l'interface sans séparer les étapes de compilation et de linking. On peut

utiliser XVT-Design pour construire rapidement ou affiner des prototypes d'applications sans écrire aucune ligne de code.

3. Au lieu de réécrire à la main le code générique de l'interface utilisateur pour chaque nouvelle application, on peut utiliser XVT-Design pour créer automatiquement ce code.

Sans XVT-Design il est nécessaire d'écrire le code source à la main, afin de définir, de façon portable, les ressources de l'interface utilisateur d'une application. Ce processus itératif est très pesant puisqu'on ne sait pas visualiser les ressources avant de faire tourner l'application.

Avec XVT-Design on crée un fichier qui contient les ressources de l'interface. Pour chaque projet on peut créer un nombre quelconque de boîtes de dialogues et de fenêtres. Ensuite on dessine les boutons de commande dont on a besoin sur l'écran, en les plaçant à l'endroit voulu dans une fenêtre ou une boîte de dialogue .

9.3 Les concepts

Ce paragraphe introduit quelques concepts utilisés avec XVT-Design.

9.3.1 Les attributs des objets

Tous les objets de l'interface disponible avec XVT-Design (fenêtres, boîtes de dialogues,...) possèdent certains attributs qui décrivent leur apparence et leur comportement. XVT-Design présente des boîtes de dialogues avec des commandes pour ces attributs. Ceci permet d'établir interactivement les valeurs de ces attributs plutôt que de devoir les spécifier dans un programme.

9.3.1.1 La géométrie

La taille et la position des fenêtres, des boîtes de dialogues et des boutons de commandes sont spécifiées par des rectangles. La taille est indiquée par la longueur et la largeur du rectangle. Quand à la position elle est précisée par les coordonnées du coin supérieur gauche du rectangle. XVT-Design permet donc de créer et de modifier graphiquement les objets comme si on travaillait avec une application graphique.

9.3.1.2 Les titres

Chaque objet possède un string indiquant le nom de l'objet d'un point de vue utilisateur de l'application. Un objet peut avoir un titre visible ou invisible dépendant de son type et des conventions initiales du système de fenêtrage.

9.3.1.3 L'identifiant de la ressource

Les objets ont un deuxième string utilisé par le développeur de l'application. Il permet de donner un nom symbolique à l'objet à la place de son nom ID de ressource.

9.3.2 Le code de l'interface utilisateur

En plus de créer graphiquement les ressources, XVT-Design apporte une certaine aide dans la création du code pour les objets de l'interface utilisateur de l'application.

Il définit lui-même les endroits où l'on peut écrire le code selon tel ou tel événement. De plus lors de la génération du code, il créera les entêtes et le corps du scripte sera placé dans la bonne procédure automatiquement.

ANNEXES

Annexes

1. Codes des différentes fonctions

```
/*----- TYPEDEF SECTION -----*/
```

```
typedef struct {  
    unsigned short  RemedId;  
    unsigned char   Degree;  
    unsigned short  AuthorId;  
    /* unsigned short Views; /*6005*/  
#ifdef DEBUG  
    unsigned char   Check;  
#endif  
} REMED_LIST;
```

```
typedef struct {  
    unsigned char   *SptTextLg1;  
    unsigned char   *SptTextLg2;  
    char           Intensity;  
    unsigned char   Group;  
    REMED_LIST     *RmdList; /* Pointer to a table of Remedy List */  
    unsigned char   Degrees;  
    unsigned char   Qualif;  
    unsigned short  NbRemedies;  
    char           Langue[2];  
    unsigned char   Origin;  
    unsigned short  View; /*6006*/
```

```
void           *Next; /* Pointer to the next Spt in memory */
```

```
#ifdef DEBUG  
    unsigned char   Check;  
#endif  
} SYMPTOM_IN_MEMORY;
```

```
typedef struct {  
    unsigned char   *s1;  
    unsigned char   *s2;  
    float           comp;  
} COMP_ECH_SAATY;
```

```

typedef struct {
    char          InMemory;
    unsigned short NbSpt;
    char          *Name;
    SYMPTOM_IN_MEMORY *First;
    SYMPTOM_IN_MEMORY *Last;
#ifdef DEBUG
    unsigned char  Check;
#endif
} SYMPTOM_CLIPBOARD;

```

```

typedef struct {
    char          Intensity;
    unsigned char Group;
    unsigned char Degrees;
    unsigned char Qualif;
    unsigned short SCNumber[MAX_NB_SC+2]; /***/
#ifdef DEBUG
    unsigned char  Check;
#endif
} CURRENT_TAKE_SETTINGS;

```

```

/*****
*****

```

NAME

InitializeAnalyse - Initialize the Analyse structures

SYNTAX

```

void
InitializeAnalyse ( Calculate )

```

PARAMETERS

```

int    Calculate  I Tell if it is a real Analyse or just an
                    I evaluation for the POLYCHRESTS and the SMALL
                    I REMEDIES ( 1 = Analyse; 0 = Evaluation )

```

INCLUDE

```

xvt.h
radar.h

```

DESCRIPTION

This function makes all initializations to analyse symptoms

RETURN

EXAMPLE

SEE ALSO

```
*****  
*****/
```

```
void  
InitializeAnalyse ( Calculate )  
    int Calculate;  
{  
int      _i, _j, _k, _nb_tot_remedies;  
int      _x, _y;  
long     _max_remedies;  
SYMPTOM_IN_MEMORY * _spt;  
unsigned short _old_remed;  
  
long     _pwd1,  
         _pwd2,  
         _rad,  
         _code;  
  
    if ( Calculate == 2 ) {  
        Calculate = 1;  
    }  
  
        /* Read protection parameters */  
  
    _pwd1 = ProtectDecodeLong ( &ProtectArray [ PROTECT_HASP + 8 ], 2, 0,  
3, 1 );  
    _pwd1 = PROTECT_GET_XOR ( _pwd1 );  
  
    _pwd2 = ProtectDecodeLong ( &ProtectArray [ PROTECT_HASP + 12 ], 3, 1,  
0, 2 );  
    _pwd2 = PROTECT_GET_XOR ( _pwd2 );  
  
    _rad = ProtectDecodeLong ( &ProtectArray [ PROTECT_HASP + 20 ], 0, 2, 1,  
3 );  
    _rad = PROTECT_GET_XOR ( _rad );
```

```

if ( FirstAnalyse == 1 ) {
    DefaultAnalyseSettings ( );
    FirstAnalyse = 0;
}
else {

    if ( RemedyScore ) {
        Free ( (unsigned char *) RemedyScore );
    }
}

/* Read protection code */

if ( ( ProtectCounter % 3 ) == 0 )
    _code = ProtectHaspGetCode ( 47253L, 16523L, ProtectCallHasp [
ProtectCounter % 3 ]
);
else
    _code = ProtectHaspGetCode ( _pwd1, _pwd2, ProtectCallHasp [
ProtectCounter % 3 ] );
    _code = PROTECT_GET_HASP_CODE ( _code );

set_cursor ( TASK_WIN, CURSOR_WAIT );

for ( _i=0; _i<52; _i++) Groups [ _i ] = 0;

/*
! 12023 On doit avoir au moins 1 SC selectionne pour l'analyse
*/

for ( _i=0; _i<MAX_NB_SC && AnalyseSettings.SelectedSC[ _i ] == 0; _i++ );
if ( _i >= MAX_NB_SC ) {
    for ( _i=0; _i<MAX_NB_SC && SC[ _i ].NbSpt == 0; _i++ );
    if ( SC[ _i ].NbSpt > 0 ) AnalyseSettings.SelectedSC[ _i ] = 1;
}

/* Really test protection */

if ( PROTECT_IS_NOT_GOOD ( _code, _rad ) )
    xvt_fatal ( GetMessage ( FileMessage, 836,
Option.General.Lang, &RecMessage ),
PROTECT_ERR_CODE );
PROTECT_NEXT_COUNTER;

```



```

/*****
/* Determines the maximum number of remedies */
/*****
_nb_tot_remedies = 0;
for (_i=0; _i < MAX_NB_SC; _i++) {
    if ( SC[_i].NbSpt > (unsigned short) 0 &&
        AnalyseSettings.SelectedSC [_i] == 1 ) {
        _nb_tot_remedies += GetTotRemediesInSC (_i);
    }
}
_max_remedies = DbCountEntries ( FileRemed );
if ( _nb_tot_remedies > (int)_max_remedies )
    _nb_tot_remedies = (int) _max_remedies;

/*****
/* Allocating a the table for the remedy scores */
/*****
if ( !(RemedyScore =
        (REMEDY_SCORE *) Malloc (sizeof
(REMEDY_SCORE)*_nb_tot_remedies)) ) {
    xvt_msg ( GetMessage ( FileMessage, 194, Option.General.Lang,
&RecMessage) );
    return;
}

/*****
/* Maintenant on remplit la table : FLE VOIR ICI */
/*****
NbRemedyInScoreTable = 0;

for (_i=0; _i < MAX_NB_SC; _i++) {
    if ( SC[_i].NbSpt > (unsigned short) 0 &&
        (AnalyseSettings.SelectedSC [_i] == 1) ) {
        _spt = SC [_i].First;

        for (_j=0; _j < (int)SC[_i].NbSpt; _j++) {

            /* On remplit le vecteur Groups */
            if ( _spt->Group != '\0' ) {
                if ( _spt->Group >= 'A' && _spt->Group <= 'Z' ) {
                    Groups [ _spt->Group-65 ] ++;
                }
                if ( _spt->Group >= 'a' && _spt->Group <= 'z' ) {
                    Groups [ _spt->Group-71 ] ++;
                }
            }
        }
    }
}

```

```

    }
    _old_remed = 0;
    for (_k=0; _k<(int)_spt->NbRemedies; _k++) {

        if ( _spt->RmdList[_k].RemedId != _old_remed ) {
            if ( IsItAValidRemedy ( _spt, _k ) ) {
                _old_remed = _spt->RmdList[_k].RemedId;
                if ( IsRemedyAlreadyInScoreTable ( _spt-
>RmdList[_k].RemedId,
NbRemedyInScoreTable) < 0 ) {

                    /*****/
                    /* Tri par inserton !!! */
                    /*****/
                    _x = NbRemedyInScoreTable - 1;
                    _y = 0;

                    while ( _x >= 0 && !_y ) {
                        if ( _spt->RmdList[_k].RemedId <
RemedyScore[_x].RemedId ) {
                            RemedyScore[_x+1].RemedId =
RemedyScore[_x].RemedId;
                            RemedyScore[_x+1].Score1 =
RemedyScore[_x].Score1;
                            RemedyScore[_x+1].Score2 =
RemedyScore[_x].Score2;
                            RemedyScore[_x+1].Flag = RemedyScore[_x].Flag;
                            _x --;
                        }
                        else _y = 1;
                    }

                    RemedyScore[ _x+1 ].RemedId = _spt-
>RmdList[_k].RemedId;
                    RemedyScore[ _x+1 ].Flag = 0;
                    RemedyScore[ _x+1 ].Score1 = (float)0.0;
                    RemedyScore[ _x+1 ].Score2 = (float)0.0;
                    NbRemedyInScoreTable ++;
                }
            }
        }
        _spt = _spt->Next;
    }
}

/*****/

```

```

/* On doit regarder si il y a des symptômes éliminatoires */
/*****/

ResetRmdFlagSptInMemory ();

if ( TestSymptomsGroupValidity () == FAIL ) {
    TerminateAnalyse ();
    return;
}

if ( AnalyseSettings.Method != VES ) HandleEliminatifSymptoms ();

if ( Calculate == 1 ) {

    _i = FlagValidRemediesToHandle ();

    switch ( AnalyseSettings.Method ) {

        case COMBINED_ANALYSES:
            if ( ProtectIsDocumentAvailable ( (unsigned char *) NULL,
                PROTECT_MODULES -
RADAR_MODULE_COMBINED_ANALYSIS ) )
                _i = CalculateScoreCombi ();
            break;

        case SAATY_METHOD:
            _i = CalculateScoreSaaty ( 1 );
            break;

        case SUM_OF_DEGREES:
            _i = CalculateScoreFrq ( 1 );
            break;

        case SUM_OF_SYMPTOMS_DEGREES:
            _i = CalculateScoreSptPlusFrq ();
            break;

        case SMALL_RUBRICS:
            if ( ProtectIsDocumentAvailable ( (unsigned char *) NULL,
                PROTECT_MODULES - RADAR_MODULE_SPF
            ) )
                _i = CalculateScoreSPF ();
            break;

        case SMALL_REMEDIES:
            _i = CalculateScoreSmallRemedies ();
            break;
    }
}

```

```

case PROMINENCE:
    _i = CalculateScoreProminence ( );
    break;

case VES:
    if ( ProtectIsDocumentAvailable ( (unsigned char *) NULL,
                                     PROTECT_MODULES - RADAR_MODULE_VES
    ))
        Ves ( );
    break;

case SUM_OF_SPT_DEG:
    _i=CalculateScoreSpt ( 1 );
    _i=CalculateScoreFrq ( 2 );
    break;

case SUM_OF_DEG_SPT:
    _i=CalculateScoreFrq ( 1 );
    _i=CalculateScoreSpt ( 2 );
    break;
}
set_cursor ( TASK_WIN, CURSOR_ARROW );

DisplayAnalyse ( AnalyseSettings.Method );
}
else {
/*
    22/08/94: DP

    We don't have to free Remedy score here, it will be done the next
    time we call InitializeAnalyse or at the end of Radar

    TerminateAnalyse ( );
*/

    set_cursor ( TASK_WIN, CURSOR_ARROW );

}
}

```

```

/*****
*****

```

NAME

Analyse de l' AHP- Calcule les scores pour les méthodes Saaty

SYNTAX

void

CalculateScoreSaaty (_i)

PARAMETERS

int _i; I i=1 si on doit mettre le score dans Score 1 et =2 si Score2

INCLUDE

xvt.h
radar.h

DESCRIPTION

Calculates the score for method of Saaty.

RETURN

EXAMPLE

SEE ALSO

*****/

```
int
CalculateScoreSaaty ( ScoreToUse )
int ScoreToUse;

{
SYMPTOM_IN_MEMORY * _spt,* _spt1,* _spt2;
int      _i, _j, _jj, _k, _jjj;
unsigned short  _old_remede;
float      m_symptomes[20][20];
float      _int,_ic,somme,val_propre;
float      v_int[40];
int        _it, _nbr_it;
float      v_propre_symptomes[20];
float      echelle_deg[5];
float      m_remedes[40][40];
float      v_propre_remedes[40];
int        indice_rem_in_table;
int        diff;
int        _nbr_rem_p_degre[5];

if ( !_symp =
(COMP_ECH_SAATY *) Malloc (sizeof (COMP_ECH_SAATY)) ) ) {
xvt_msg ("error allocating memory");
return;
```

```

    }
    /* if (!(m_remedes =
       (float) Malloc (50*50*sizeof (float))) ) {
       xvt_msg ("error allocating memory");
       return;
       } */
    /* m_remedes =(float*) malloc(50*50*sizeof (float));
       */

    for (_i = 0; _i < MAX_NB_SC; _i++) {
        if ((SC [_i].NbSpt > 0) && (AnalyseSettings.SelectedSC [_i] == 1)) {
            _spt = SC[_i].First;

            /* Comparer les symptomes jusqu'a coherence */

            _ic = (float)11;
            _nbr_it = 10;
            while (_ic > 10){

                /* Comparaison par paire des symptomes pour le SC _i */

                /*xvt_dbg (" Avant le for sur le nombre de spt" );*/
                for (_j = 0; _j < (int)SC[_i].NbSpt; _j++) m_symptomes[_j][_j] =
                (float) 1;
                _spt1 = SC[_i].First;
                for (_j = 0; _j < (int)SC[_i].NbSpt; _j++) {
                    _spt2 = _spt1->Next;
                    for(_jj = _j+1; _jj < (int)SC[_i].NbSpt; _jj++) {

                        /* Ouverture normalement d'une fenetre XVT modale pour la comparaison
                           paire par paire. Pour l'instant Harcode.
                           */

                            _symp->s1 = _spt1->SptTextLg2;
                            _symp->s2 = _spt2->SptTextLg2;
                            _symp->comp = (float) 1;

                                create_res_dialog(WD_MODAL,DLG_220,EM_ALL,dlg_220_eh,(lon
                                g)NULL);

                                    _int = _symp->comp;
                                    if (_int > 0){
                                        m_symptomes[_j][_jj] = _int;
                                        m_symptomes[_jj][_j] = 1/_int;}
                                    else {
                                        m_symptomes[_jj][_j] = (float) 0 - _int;

```

```

        m_symptomes[_j][_jj] = (float) 0 - (1/_int);}

        _spt2 = _spt2->Next;
    }
    _spt1 = _spt1->Next;
}

for (_j = 0; _j < (int)SC[_i].NbSpt; _j++) v_int[_j] = 1;

for (_it = 0; _it < _nbr_it ; _it++){
    for (_j = 0; _j < (int)SC[_i].NbSpt; _j++){
        v_propre_symptomes[_j] = 0;
        for (_jj = 0; _jj < (int)SC[_i].NbSpt; _jj++){
            v_propre_symptomes[_j] = m_symptomes[_j][_jj] *
v_int[_jj] +
v_propre_symptomes[_j];
        }
        val_propre = v_propre_symptomes[1];
        for (_j = 0; _j < (int)SC[_i].NbSpt; _j++) v_int[_j] =
v_propre_symptomes[1]/val_propre;
    }
    _ic = (val_propre - SC[_i].NbSpt)/(SC[_i].NbSpt - 1);
}

somme = 0;
for (_j = 0; _j < (int)SC[_i].NbSpt; _j++){
    v_propre_symptomes[_j] = v_propre_symptomes[_j]/val_propre;
    somme = somme + v_propre_symptomes[_j];
}

for (_j = 0; _j < (int)SC[_i].NbSpt; _j++) v_propre_symptomes[_j] =
v_propre_symptomes[_j]/somme;

/* Le vecteur propre des symptomes est calcule */

/* Initialisation du tableau des differences de degres */

    echelle_deg[0] = 1;
    echelle_deg[1] = 4;
    echelle_deg[2] = 6;
    echelle_deg[3] = 8;
    echelle_deg[4] = 9;

/* Construction du vecteur de priorite */

    for (_jjj = 0; _jjj < (int)SC[_i].NbSpt; _jjj++){

```

```

        for (_jj = 0; _jj < NbRemedyInScoreTable; _jj++)
RemedyScore[_jj].Score2 = 128;
        for (_jj = 0; _jj < 5; _jj++) _nbr_rem_p_degre[_jj] = 0;
        _old_remede = 0;
        for (_k=0; _k<(int)_spt->NbRemedies; _k++) {

            if (_spt->RmdList[_k].RemedId != _old_remede ) {
                if ( IsItAValidRemedy ( _spt, _k ) ) {
                    _old_remede = _spt->RmdList[_k].RemedId;
                    indice_rem_in_table = IsRemedyAlreadyInScoreTable ( _spt-
>RmdList[_k].RemedId, NbRemedyInScoreTable);
                    if ( indice_rem_in_table >= 0 ) {
                        RemedyScore[indice_rem_in_table].Score2 = _spt-
>RmdList[_k].Degree;

                            if (RemedyScore[indice_rem_in_table].Score2 ==
129) {
                                _nbr_rem_p_degre[1] = _nbr_rem_p_degre[1] + 1;
                                }
                            else {
                                if (RemedyScore[indice_rem_in_table].Score2 ==
130) {
                                    _nbr_rem_p_degre[2] =
_nbr_rem_p_degre[2] + 1;
                                    }
                                    else {
                                        if
(RemedyScore[indice_rem_in_table].Score2 == 131) {
                                            _nbr_rem_p_degre[3] =
_nbr_rem_p_degre[3] + 1;
                                            }
                                            else {
                                                _nbr_rem_p_degre[4] =
_nbr_rem_p_degre[4] + 1;
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
        _old_remede = _spt->RmdList[_k].RemedId;
    }
    if (NbRemedyInScoreTable <= 40){

```



```

for (_jj = 0; _jj < NbRemedyInScoreTable; _jj++){
    for (_k = _jj; _k < NbRemedyInScoreTable; _k++){
        diff = (int)RemedyScore[_jj].Score2 -
(int)RemedyScore[_k].Score2;
        if ( diff >= 0 ){
            m_remedes[_jj][_k] = echelle_deg[diff];
            m_remedes[_k][_jj] = 1/echelle_deg[diff];
        }
        else {
            m_remedes[_jj][_k] = 1/echelle_deg[-diff];
            m_remedes[_k][_jj] = echelle_deg[-diff];
        }
    }
}
/*****New Version*****/
for (_j = 0; _j < (int)NbRemedyInScoreTable; _j++) v_int[_j] = 1;

for (_it = 0; _it < _nbr_it ; _it++){
    for (_j = 0; _j < (int)NbRemedyInScoreTable; _j++){
        v_propre_remedes[_j] = 0;
        for (_jj = 0; _jj < (int)NbRemedyInScoreTable;
_jj++){
            v_propre_remedes[_j] = m_remedes[_j][_jj] *
v_int[_jj] + v_propre_remedes[_j];
        }
        val_propre = v_propre_remedes[1];
        for (_j = 0; _j < (int)NbRemedyInScoreTable; _j++) v_int[_j] =
v_propre_remedes[1]/val_propre;
    }
}

/*****/

somme = 0;
for (_j = 0; _j < NbRemedyInScoreTable; _j++){
    v_propre_remedes[_j] = v_propre_remedes[_j]/val_propre;
    somme = somme + v_propre_remedes[_j];
}
}
else {
    echelle_deg[0] = (float).4;
    echelle_deg[1] = (float).8;
    echelle_deg[2] = (float)1;
    echelle_deg[3] = (float)1.4;
    echelle_deg[4] = (float)1.8;

    somme = 0;

```

```

        for ( _j = 0; _j < (int)NbRemedyInScoreTable; _j++){
                diff = (int)RemedyScore[_j].Score2 - 128;
                v_propre_remedes[_j]
echelle_deg[diff]/NbRemedyInScoreTable;
                somme = somme + v_propre_remedes[_j];
        }
}

        for ( _j = 0; _j < (int)NbRemedyInScoreTable; _j++) v_propre_remedes[_j]
=
v_propre_remedes[_j]/somme;
        for ( _j = 0; _j < (int)NbRemedyInScoreTable; _j++){
                RemedyScore[_j].Score1 = RemedyScore[_j].Score1 +
v_propre_symptomes[_jj]
* v_propre_remedes[_j];
        }
        _spt = _spt->Next;
}

}
if ( NbRemedyInScoreTable < 20 ) {
        for ( _j = 0; _j < (int)NbRemedyInScoreTable; _j++){
                RemedyScore[_j].Score1 = RemedyScore[_j].Score1 * 100;
        }
}
else {
        for ( _j = 0; _j < (int)NbRemedyInScoreTable; _j++){
                RemedyScore[_j].Score1 = RemedyScore[_j].Score1 * 1000;
        }
}
return ( RADAR_OK );
}
}
/*----- TYPEDEF SECTION -----*/

typedef struct {
        unsigned short  RemedId;
        unsigned char   Degree;
        unsigned short  AuthorId;
        /* unsigned short  Views;*/ /*6005*/
#ifdef DEBUG
        unsigned char   Check;
#endif
} REMED_LIST;

```

```

typedef struct {
    unsigned char *SptTextLg1;
    unsigned char *SptTextLg2;
    char Intensity;
    unsigned char Group;
    REMED_LIST *RmdList; /* Pointer to a table of Remedy List */
    unsigned char Degrees;
    unsigned char Qualif;
    unsigned short NbRemedies;
    char Langue[2];
    unsigned char Origin;
    unsigned short View; /*6006*/

    void *Next; /* Pointer to the next Spt in memory */

#ifdef DEBUG
    unsigned char Check;
#endif
} SYMPTOM_IN_MEMORY;

typedef struct {
    unsigned char *s1;
    unsigned char *s2;
    float comp;
} COMP_ECH_SAATY;

typedef struct {
    char InMemory;
    unsigned short NbSpt;
    char *Name;
    SYMPTOM_IN_MEMORY *First;
    SYMPTOM_IN_MEMORY *Last;
#ifdef DEBUG
    unsigned char Check;
#endif
} SYMPTOM_CLIPBOARD;

typedef struct {
    char Intensity;
    unsigned char Group;
    unsigned char Degrees;
    unsigned char Qualif;
    unsigned short SCNumber[MAX_NB_SC+2]; /****/

```

```
#ifdef DEBUG
    unsigned char  Check;
#endif
```

```
} CURRENT_TAKE_SETTINGS;
```

Bibliographie

- [1] Allen T.F.A. (1977). *The Encyclopedia of Pure Materia Medica: a record of the positive effects of drugs upon the Healthy Human Organism*. B.Jain Publishers.
- [2] Arbel A. and Orgler Y.E. (1990). An application of the A.H.P. to bank strategic planning: the mergers and acquisitions process. *European Journal of Operational Research*, 48, pp 27-37.
- [3] Archibel S.A. (1991). *User's Manual for Radar*.
- [4] Aupetit B and Genest C. (1993). On some useful properties of the Perron eigenvalue of a positiv reciprocal Matrix in the context of the A.H.P..*European Journal of Operational Research*, 70, pp 263-268.
- [5] Basak I.(1990). Testing for the rank ordering of priorities of the alternatives in Saaty's Ratio-Scale Method. *European Journal of Operational Research*, 48 pp 148-152
- [6] Belton V. and Gear T. (1985). Feedback-The legitimacy of rank reversal: a comment. *Omega*, 13, pp 143-144.
- [7] Dyer J.S. (1990). Remarks on the Analytic Hierarchy Process. *Management Science*, 36 pp 249-258.
- [8] Fichet J. (1992). About some traps set by "crude" repertorization methods. *British Homeopathic Journal*, 81, pp 2-12.
- [9] Fichet J., Jacques A., Gardin P. et Paris J. R.A.D.A.R. : un système expert à base de connaissances pour l'homéopathie.
- [10] Harker P.T. and Vargas L.G. (1987). The theory of ratio scale estimation: Saaty's Analytic Hierarchy Process. *Management Science*, 33 pp 1383-1402.
- [11] Harker P.T. and Vargas L.G. (1990). Reply to "remarks on the A.H.P." by J.S. Dyer. *Management Science*, 36 pp 269-273.
- [12] Jacques A. .Les principes de base de l'homéopathie. *Universitas Internationalis Hominum Novarum*.
- [13] Millet I. and Harker P.T. (1990). Globally effective questioning in the Analytic Hierarchy Process. *European Journal of Operational Research*, 48, pp 88-98.
- [14] Murphy C.K. (1993). Limits on the Analytic Hierarchy Process from its consistency index. *European Journal of Operational Research*, 65 pp 138-139.
- [15] Saaty T.L. and Vargas L.G. (1987). Uncertainty and rank order in the Analytic Hierarchy Process. *European Journal of Operational Research*, 32 pp 107-117.

- [16] Saaty T.L. (1990). How to make a decision : The Analytic Hierarchy Process. European Journal of Operational Research, 48 pp 9-26.
- [17] Saaty T.L. and Vargas L.G. (1984). The legitimacy of Rank -Reversal. Omega, 12, pp 513-516.
- [18] Saaty T.L. (1990). Eigenvector and logarithmic least squares. European Journal of Operational Research, 48, pp 156-160.
- [19] Saaty T.L., Rogers P.C. and Pell R..(1980) Portfolio selection through hierarchies. The Journal of Portfolio Management. pp 16-20.
- [20] Saaty T.L. (1987). Concepts, theory and techniques: rank reversal, preservation and reversal in the Analytic Hierarchy Decision Process. Decision Science, 18, pp 157-177.
- [21] Saaty T.L. and Vargas L.G. (1984). Inconsistency and Rank Reversal. Journal of Mathematical Psychology, 28, pp 205-214.
- [22] Saaty T.L. (1990). An exposition of the Analytic Hierarchy Process in reply to the paper "remarks on the A.H.P.". Management Science, 36 pp 259-268.
- [23] Saaty T.L. (1980). The Analytic Hierarchy Process. Mc Graw-Hill International Book Company.
- [24] Saaty T.L. (1986). Axiomatic foundation of the Analytic Hierarchy Process. Management Science, 32 pp 841-856.
- [25] Vannier P. Que sais-je ? : l'homéopathie. Presses Universitaires de France.
- [26] Vargas L.G. (1990). An overview of the Analytic Hierarchy Process and its applications. European Journal of Operational Research, 48 pp 2-8.
- [27] Winkler R.L. (1990). Decision modeling and rational choice : Analytic Hierarchy Process and Utility Theory. Management Science, 36 pp 247-248.