Université de Nancy I
Faculté des Sciences

Centre de Recherche
en Informatique de Nancy

GESTION DES INFORMATIONS NUANCEES : UNE PROPOSITION DE MODELE ET DE METHODE POUR L'IDENTIFICATION NUANCEE D'UN PHENOMENE

THESE

Présentée et soutenue le **22 Mai 1989**
pour l'obtention du

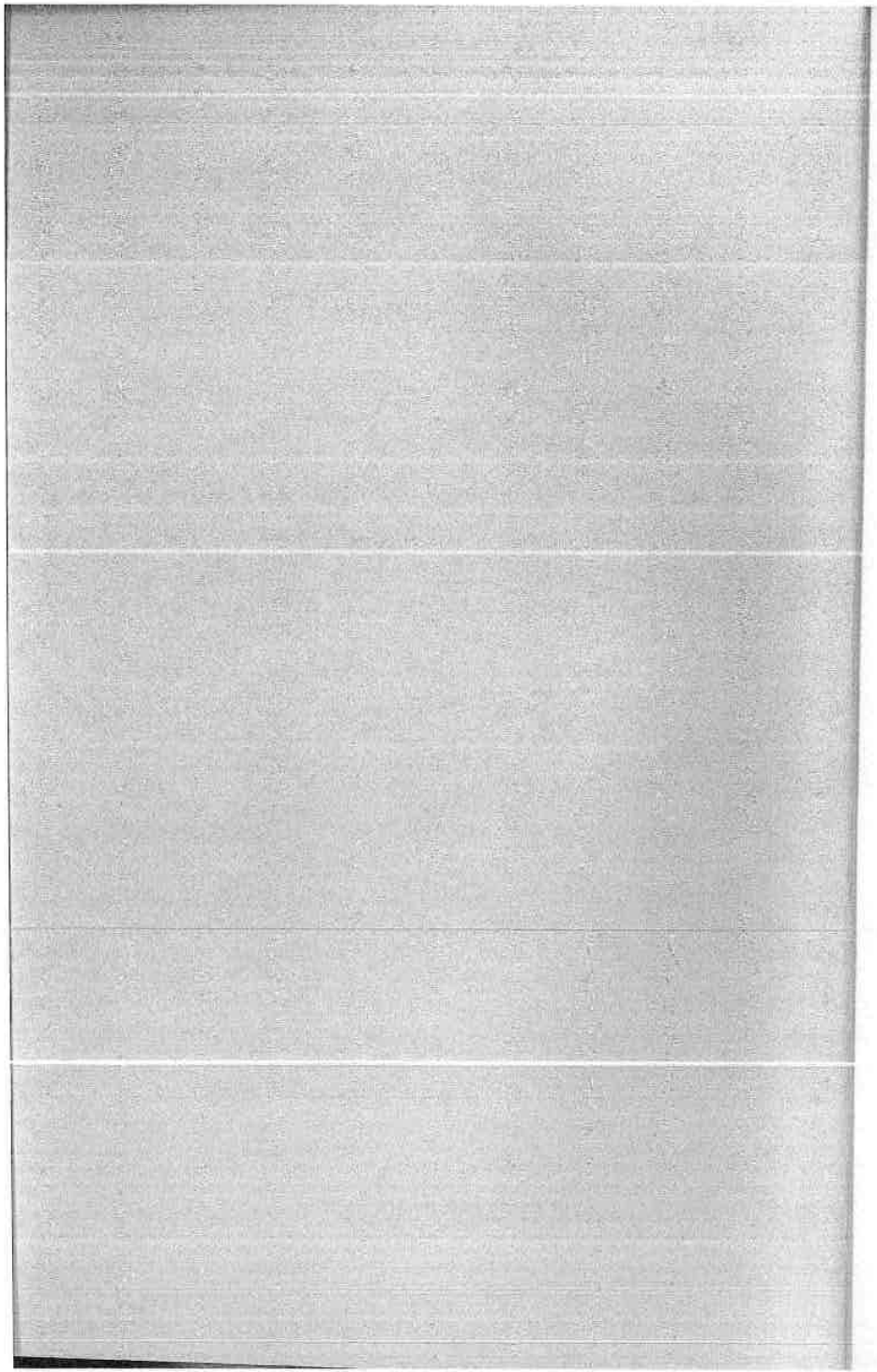
Doctorat de l'Université de Nancy 1
Spécialité INFORMATIQUE

par

Noureddine MOUADDIB

Composition du jury :

Président :	Marion CREHANGE	Professeur à l'Université de Nancy II
Rapporteurs :	Jean Paul HATON	Professeur à l'Université de Nancy I
	Claude CHRISMENT	Professeur à l'Université Paul Sabatier de Toulouse
Examinateurs :	Odile FOUCAUT	Professeur à l'Université de Nancy II
	Pierre MARCHAND	Professeur à l'Université de Nancy I
	G.T. NGUYEN	Chargé de recherche INRIA,LGI-IMAG, Grenoble
Invités :	François KOHLER	Professeur à la faculté de médecine de Nancy
	J. François FOUCAUT	Maitre de conférences à l'INP de Lorraine



Université de Nancy I
Faculté des Sciences

Centre de Recherche
en Informatique de Nancy

GESTION DES INFORMATIONS NUANCEES : UNE PROPOSITION DE MODELE ET DE METHODE POUR L'IDENTIFICATION NUANCEE D'UN PHENOMENE

THESE

Présentée et soutenue le **22 Mai 1989**
pour l'obtention du

Doctorat de l'Université de Nancy 1
Spécialité INFORMATIQUE

par

Noureddine MOUADDIB

Composition du jury :

Président :	Marion CREHANGE	Professeur à l'Université de Nancy II
Rapporteurs :	Jean Paul HATON	Professeur à l'Université de Nancy I
	Claude CHRISMENT	Professeur à l'Université Paul Sabatier de Toulouse
Examinateurs :	Odile FOUCAUT	Professeur à l'Université de Nancy II
	Pierre MARCHAND	Professeur à l'Université de Nancy I
	G.T. NGUYEN	Chargé de recherche INRIA,LGI-IMAG, Grenoble
Invités :	François KOHLER	Professeur à la faculté de médecine de Nancy
	J. François FOUCAUT	Maitre de conférences à l'INP de Lorraine

A mes parents,

Si je suis parvenu jusqu'à cette étape j'aimerais que vous sachiez que c'est en partie grâce à votre affection, à votre amour et votre profond sens de la vie.

Je vous dédie cette thèse avec toute ma tendresse en espérant qu'elle sera pour vous une "riche récolte".

A mes très chers frères et soeurs,

En témoignage de mon affection.

A Véronique,

Pour l'amour et la tendresse que tu m'apportes.

Pour ta patience et ton soutien moral tout au long de la rédaction de cette thèse.

Pour les bons moments que nous passons ensemble.

A toute ma famille.

REMERCIEMENTS

A Madame le Professeur M. CREHANGE de l'Université de Nancy 2,

Je suis très honoré par ta présence à la présidence de ce jury et je n'oublie pas que c'est à toi que je dois d'avoir entrepris mes études de thésard dans l'équipe EXPRIM ; tu m'as accueilli chaleureusement dans cette équipe il y a trois ans et as constamment manifesté de l'intérêt pour mon travail.

Je tiens à t'exprimer toute ma reconnaissance et mon profond respect.

A Messieurs les Professeurs J.P. HATON de l'Université de Nancy 1 et C. CHRISMENT de l'Université Paul Sabatier de Toulouse,

Je mesure pleinement l'honneur que vous m'avez fait en ayant accepté de rendre compte de mon travail.Vos remarques ont reflété d'autres points de vue que les miens et ont permis d'améliorer cette thèse.

Je tiens ici à vous exprimer toute ma respectueuse gratitude.

A Madame le Professeur O. FOUCAUT de l'Université de Nancy 2,

Merci tout d'abord de m'avoir fait découvrir le sujet de cette thèse d'en avoir accepté l'encadrement depuis le début, merci pour tes critiques toujours constructives, ta constante disponibilité, tes grandes qualités humaines et tes encouragements qui ont été pour moi des atouts déterminants dans la réalisation de ma thèse.

Je tiens à t'exprimer ici ma profonde gratitude.

A Monsieur le Professeur P. MARCHAND responsable du Département d'Informatique de l'Université de Nancy 1,

Je n'oublie pas que c'est sur tes conseils que j'ai entrepris ma carrière d'enseignant ; tu m'as permis ainsi de découvrir un univers passionnant que j'aurais ignoré sans toi.

J'espère être toujours digne de la confiance et de l'attention que tu m'as accordées.

Sois assuré de toute ma gratitude et de toute mon estime.

A Monsieur G.T. NGUYEN Attaché de Recherche INRIA au laboratoire LGI-IMAG de Grenoble,

Je te remercie pour ta présence dans ce jury et pour l'intérêt que tu as manifesté pour mon travail après les nombreuses discussions amicales et toujours enrichissantes que nous avons eues aux différents congrès de BD3 (Bases de Données 3ème génération) et aux différentes réunions du PRC (groupe Bases de Données).

Sois assuré de mon amitié la plus sincère.

A Monsieur J.F. FOUCAUT Maître de conférences à l'INP de Lorraine,

Tout au long de ce travail, votre aide fut déterminante pour réaliser l'application en mycologie. J'ai pu apprécier votre volonté, votre enthousiasme et vos conseils éclairés.

Je vous prie de croire à mon amicale reconnaissance.

A Monsieur le Professeur F. KOHLER de la Faculté de Médecine de Nancy,

Pour notre longue collaboration dans le milieu médical et pour l'intérêt et la confiance que tu m'as toujours accordés, je tiens à t'exprimer mon amitié la plus sincère.

A tous les membres de l'équipe EXPRIM,

Pour la bonne humeur, la bonne ambiance et le soutien qu'ils m'ont apporté pour réaliser ce travail.

RESUME

Cette thèse présente une solution globale au problème de l'identification d'un phénomène ou d'un objet mal défini dans un domaine d'application décrit par des connaissances nuancées.

Cette solution comprend trois éléments :

- un modèle de représentation des connaissances nuancées,
- une méthode de détermination des objets ressemblant au phénomène à identifier,
- un processus d'identification dans un système possédant une base de données multimedia.

Le modèle de représentation des connaissances présente les particularités suivantes :

- une ou plusieurs nuances, exprimées en langue naturelle, peuvent être associées à chacune des valeurs prise par un caractère d'un objet,
- à chaque domaine de définition discret de caractère peut être associé un micro-thésaurus dont les liens (généricité, synonymie, opposition) peuvent être munis de coefficients exprimant certaines "distances sémantiques" entre les termes,
- des poids d'importance ou de confiance peuvent être associés à chaque caractère aussi bien dans la description des objets de référence que dans la description du phénomène à identifier.

La méthode d'identification repose sur la théorie des possibilités dont nous avons assoupli l'application en diminuant le nombre de fonctions caractéristiques à fournir, par le spécialiste du domaine d'application, grâce à l'introduction d'heuristiques permettant soit de les générer à partir des micro-thésaurus soit de les calculer à partir d'autres déjà définies par composition ou par transformation

Le processus d'identification permet une identification interactive et progressive au cours de laquelle alternent des phases de filtrage, d'affichage de résultats, d'observation d'images et de consultation de textes. En cas d'échec, nous proposons une stratégie de "retour-arrière" qui s'appuie sur les poids des caractères.

Gestion des informations nuancées : une proposition de modèle et de méthode pour l'identification d'un phénomène

PLAN DE THESE

Chapitre II **GESTION DES INFORMATIONS NUANCEES DANS LES SYSTEMES D'INFORMATION. ETAT DE L'ART.** 15

INTRODUCTION

Une grande part du raisonnement humain s'effectue avec des informations nuancées, qu'elles soient incomplètes, imprécises, incertaines, floues ou vagues. La nuance sur une information est souvent inhérente au langage naturel, mais peut aussi provenir d'un appareil de mesure, d'un capteur ou autres.

Dans de nombreux domaines, la formulation précise d'un problème ou la description d'un phénomène n'est pas possible; par exemple : élaboration d'un diagnostic médical, description d'un objet, d'un individu ou d'une espèce naturelle (champignon, fleur, arbre ...). Il en résulte souvent des nuances qui ne sont pas de nature probabiliste.

La plupart des systèmes de gestion d'informations ne prennent pas en compte les informations et les données nuancées. En général, ils sont capables de :

- représenter des informations précises et certaines,
- fournir une réponse précise lors d'une interrogation,
- traiter des questions (requêtes) précises exprimées sous forme de conditions booléennes.

Le besoin de systèmes informatiques permettant la représentation et le traitement des informations nuancées a fait naître des recherches aussi bien dans le domaine des bases de données que dans le domaine de l'intelligence artificielle.

La réalisation de maquettes réelles de bases de données intégrant des textes et des images dans l'équipe EXPRIM (Système EXPert pour la Recherche d'IMages) du CRIN (Centre de Recherche en Informatique de Nancy) dans des domaines aussi variés qu'une photothèque du vieux Paris [EXP-87] et que la mycologie [FOU-87b] nous ont conduit à nous poser le problème de la représentation et de la gestion d'informations nuancées. En effet, qu'il s'agisse de décrire le contenu d'une photographie représentant une rue de Paris en 1900 ou qu'il s'agisse de décrire une espèce de champignon, il est très difficile de le faire avec des données précises telles qu'on les utilise actuellement dans les SGBDs (Systèmes de Gestion des Bases de Données). Réciproquement, exprimer clairement quel est le type de photo de rue que l'on recherche ou décrire précisément les différents aspects d'un champignon observé

est impossible en général. Dans ces deux exemples le problème à résoudre est le suivant : **chercher une réponse à une requête nuancée dans une base de données nuancées.**

Ce problème en inclut deux :
- celui de la représentation des informations nuancées ,
- celui du traitement de requêtes comportant des termes nuancés .

Dans cette thèse nous proposons une solution pour la gestion des informations nuancées et permettant de résoudre le **problème de l'identification d'un phénomène en présence d'informations nuancées .** Cette solution est applicable à de nombreux domaines, comme on le verra au chapitre I, dans lesquels informations textuelles nuancées et images sont complémentaires (sciences naturelles, médecine, architecture, ..) et pour lesquels des bases de données multimédia gérant les nuances seraient très utiles. Afin de rester concret et de bien montrer l'interêt de la solution proposée nous nous limitons volontairement ici à des domaines appartenant aux sciences naturelles, la transposition à d'autres champs d'application est facile à imaginer.

Notre solution présente les caractéristiques suivantes :

- **Utilisation de la logique floue et de la théorie des possibilités.**
- **Représentation d'une information nuancée par une liste de valeurs auxquelles il peut être associé une liste de nuances.**
- **Utilisation d'un thésaurus pour représenter les liens entre termes vagues et termes précis.**
- **Evaluation progressive et coopérative des réponses aux requêtes formulées.**

Le plan de la thèse est le suivant :

Nous introduisons au premier chapitre les concepts d'information nuancée et de nuance en nous appuyant sur de nombreux exemples concrets qui montrent l'intérêt de pouvoir gérer ces types d'information dans des systèmes automatisés.

Le second chapitre est consacré à une synthèse bibliographique des travaux déja menés dans les domaines des bases de données et de l'intelligence artificielle pour la représentation

et la gestion des informations nuancées. De cette étude, il apparaît que la logique floue et la théorie des possibilités ont été utilisées avec succès dans les deux domaines, ce qui justifie le choix que nous avons fait de cette théorie pour la gestion du flou en général.

Dans le troisième chapitre, nous présentons l'ensemble des connaissances sur lesquelles repose notre système de gestion d'information nuancée. Cette proposition est assez dépendante du domaine d'illustration retenu, les sciences naturelles, mais est cependant suffisamment générale pour pouvoir être adaptée à d'autres domaines.

Dans le quatrième chapitre, nous présentons notre méthode d'identification nuancée.Cette méthode intègre en plus des résultats de la théorie des possibilités un processus coopératif et progressif du type EXPRIM et la possibilité de pondérer les caractères (ou attributs), que ce soit dans les descriptions des espèces ou dans l'exemplaire à identifier.

Enfin, nous décrivons au chapitre cinq d'une part la maquette du système de gestion des informations nuancées que nous avons réalisé en LE_LISP et d'autre part l'application concrête sur laquelle nous avons testé l'ensemble de nos propositions. Il s'agit d'une base de connaissances en mycologie permettant une description nuancée des espèces de champignons, très proche de celle figurant dans les ouvrages mycologiques. Le processus d'identification permet à l'utilisateur de donner une description vague et incomplète de l'exemplaire qu'il veut identifier, le système lui fournissant en réponse une liste d'espèces "possibles" ordonnée suivant leur "ressemblance" avec l'exemplaire à identifier.

On trouvera en annexe des extraits de la base des connaissances et des exemples d'identification.

Chapitre I CONCEPTS D' INFORMATION NUANCEE ET DE NUANCE

Chapitre I CONCEPTS D' INFORMATION NUANCEE ET DE NUANCE

Ce chapitre introduit les concepts d'information nuancée et de nuance sur lesquels s'appuie l'ensemble de la thèse.

Nous commençons par donner quelques exemples de domaines réels pour lesquels il n'est pas possible d'utiliser les informations précises classiques pour les modéliser et dans lesquels les informations et les connaissances sont floues par essence.

Nous proposons de représenter les phénomènes réels par une extension du modèle Attributs, Objets, Valeurs (AOV) [FEL-69] dans lequel la valeur n'est plus une donnée précise mais une donnée nuancée appartenant à un nouveau type de donnée que nous introduisons : **le type information nuancée.**

Nous donnons une définition de ce type illustrée par des exemples concrets. Nous terminons par une typologie des nuances. Nous concluons en montrant les avantages et les limites de ce modèle de représentation des informations nuancées.

1. DOMAINES D'APPLICATION

Il existe à l'heure actuelle de nombreux univers réels ayant un grand besoin d'outils informatiques et dans lesquels les connaissances ne sont pas toujours exprimées d'une manière précise.

Parmi ces univers citons :

- la médecine,
- les sciences naturelles ,
- la reconnaissance des formes,
- l'archéologie,
- la cartographie,
- la documentation,
- le dépôt de brevets ou de marque (recherche de brevets similaires),
- l'identification d'individus,
- etc ...

Nous allons nous limiter à l'étude des trois premiers univers.

1.1. Médecine

Les applications de l'informatique les plus connues dans la pratique médicale sont des problèmes d'aide à la décision ou d'aide au diagnostic.

Le traitement du problème de l'aide à la décision en médecine pour élaborer un diagnostic sous-entend la prise en compte d'impressions subjectives, de la formation et de l'expérience humaine pour manipuler des informations souvent imprécises par nature ou rendues imprécises par le malade. Toutefois établir un diagnostic résulte d'une démarche logique qui consiste de manière schématique à utiliser, grâce à des règles de décision, une connaissance médicale compte tenu de l'état du patient.

L'impact de l'informatique, en particulier de l'intelligence artificielle, dans ce domaine a conduit à la réalisation de systèmes d'aide au diagnostic "intelligents", dont une majeure partie ne traite que des connaissances précises et certaines. Seuls, quelques systèmes comme MYCIN [BUC-84], SPHINX [FIE-84], DIABETO [BUI-85], SPII [CLO-85] peuvent exploiter , à des degrés très variés, des connaissances incertaines et parfois imprécises.

Voici un exemple, emprunté à DIABETO [BUI-87], de règles exprimant la connaissance médicale entachée d'imprécision et d'incertitude, telles qu'elles ont été énoncées par des diabétologues :

R1 : *si le malade a un parent diabétique non insulino-dépendant*
alors le malade est ***plus vraisemblablement*** *non insulino-dépendant*
R2 : *si le malade maigrit* ***fortement*** *et si le malade a* ***moins de 30 ans***
alors le malade est ***très vraisemblablement*** *insulino-dépendant*
R3 : *si le malade ne maigrit pas* ***fortement*** *et si le malade pèse* ***plus de 120% du poids idéal***
alors le malade est ***très vraisemblablement*** *non insulino-dépendant*

Comme les règles, les faits traduisant les observations, les résultats d'analyses sont souvent imprécis et incertains, par exemple :
le malade a ***beaucoup maigri*** *depuis* ***quelques mois.***

1.2. Sciences naturelles

Dans les ouvrages scientifiques consacrés à la description des phénomènes et des espèces de ces domaines, une large place est accordée à l'expression en langage naturel. En effet, les espèces sont décrites par des termes auxquels sont associées des informations complémentaires concernant soit leur fréquence d'apparition, soit leurs conditions d'observation, soit leur localisation.

Prenons un exemple simple en mycologie : la couleur du chapeau de "Russula Aurata" d'après l'ouvrage ROMAGNESI [ROM-61] :

Le chapeau ... est d'un beau rouge briqueté vif, rouge cuivré, rouge orangé, rarement *assombri de pourpe ou de violet,* ***souvent*** *avec des plaques* ***jaunes citron*** *mais* ***rarement*** *en entier de cette dernière couleur*

Parmi les travaux réalisés sur ces domaines, citons essentiellement quelques systèmes experts en geologie comme PROSPECTOR [DUD-81] et ELFIN [CLO-84] qui permettent des raisonnements approximatifs. D'autres systèmes experts ont été également développés en mycologie, principalement des systèmes d'identification [FOU-85] qui ne raisonnent que sur des données précises.

Dans la suite de cette thèse, les exemples seront souvent issus des sciences de l'observation et plus particulièrement de la mycologie car nous avons utilisé une nouvelle version de la base de connaissances du logiciel MYCOMATIC pour tester nos méthodes de résolution.

1.3. Reconnaissance des formes

C'est un autre domaine où les problèmes d'imprécision et d'incertitude sont omni-présents. En effet, on a souvent affaire à des problèmes de décision sur des données dont il est difficile d'apprécier les limites ou pour lesquelles la source est entachée d'incertitude. Pendant longtemps, ces probèmes ont été ignorés, par l'utilisation de logiques conventionnelles et par des méthodes de décision procédant par découpage de limites arbitraires entre classes d'objets à reconnaître.

Les deux principales variantes de la reconnaissance des formes, le traitement d'image(vision) et la reconnaissance de la parole, posent souvent des problèmes liés à ceux de la classification. Par exemple, pour reconnaître un objet, un système de vision se sert de diverses caractéristiques - taille , largeur, forme , ... - afin de déterminer si l'image interprétée par un robot correspond à un certain objet.

Parmi les travaux qui se sont intéressés au problème de l'imprécision et de l'incertitude dans ce domaine citons, essentiellement, ceux de HIRSCH [HIR-87] portant sur les problèmes des équations de relation floue et les mesures de l'incertain, et ceux de ROMARY [ROM-87] qui expose les problèmes liés à l'incertain .

2. DEFINITION DES CONCEPTS D'INFORMATION NUANCEE ET DE NUANCE

2.1. Définition sémantique du concept d'information nuancée

Une information nuancée est une information qui nécessite, pour son expression, ***une liste de valeurs dans laquelle chaque valeur est modulée, précisée, limitée ou altérée par une ou plusieurs informations complémentaires appelées nuances.***

Exemples d'informations nuancées
- la couleur du chapeau des espèces de champignon,
- le cri des espèces d'oiseaux,
- un type de situation météorologique,
- le résultat d'une série de mesures physico-chimiques,
- les symptômes d'une maladie,
- le contenu d'une photographie,
- l'analyse d'une scène par un robot.

2.2. Définition du type de données : information nuancée

Une donnée du type "information nuancée" est une liste de couples (v,l) où :

- v appartient à un ensemble D : domaine de la donnée.
- l est une liste, éventuellement vide, de termes appartenant à un ensemble ND : l'ensemble des nuances applicables aux valeurs de D.

Exemples :

- la couleur du chapeau de "Russula aurata", dont on a donné la description précédemment, peut être représentée par l'information nuancée suivante :

(rouge, (vif, souvent)) (orange, (vif, souvent)) (violet, (rarement))

(jaune, (en plaques, souvent)) (jaune, (totalement, rarement))

rouge, orange, violet, jaune sont des valeurs appartenant au domaine COULEUR contenant l'ensemble des couleurs utilisées pour décrire les chapeaux des champignons ;

vif, souvent, rarement, en plaques, totalement sont des nuances appartenant à l'ensemble NCOU des nuances applicables au domaine COULEUR.

- Le cri du "Rossignol Philomène", d'après sa description dans l'ouvrage de PETERSON, [PET-79] peut être décrit par l'information nuancée suivante :

(houit, (liquide)) (tac,(sonore)) (teux, (doux, très bref)) (harr, (rauque, en alarme))

houit, tac, teux, harr sont des valeurs appartenant au domaine CRI contenant l'ensemble des sons utilisés pour décrire les cris d'oiseaux ;

liquide, sonore, doux, très bref, rauque, en alarme sont des nuances appartenant à l'ensemble NCRI des nuances applicables au domaine CRI.

2.3. Interprétation d'une information nuancée

i = (v1, l1) (v2, l2) (v3, l3) ..... (vn, ln)

signifie que i a comme valeur :

- soit une des valeurs v1, v2 .... vn,
- soit une quelconque combinaison de ces n valeurs.

Ce sont les listes li qui apportent une information quant à l'exclusion de certaines valeurs, à leur concomitance, à leur chance d'apparition simultanée etc...

Il n'y a pas implicitement ni de OU ni de ET entre les valeurs v1, v2 .... vn, ce sont les informations contenues dans les listes l1, l2 ....ln, c'est à dire les nuances, qui apportent cette précision grâce à des informations sémantiques qui leur sont associées .

La longueur des listes li sera souvent limitée à un terme, elle atteindra au plus trois termes pour un objet donné.

Exemples :

Un poisson bleu et jaune sera décrit :

(bleu, toujours) (jaune, toujours)

Un poisson bleu et jaune ou plus rarement bleu et rouge :

(bleu,toujours) (jaune, généralement) (rouge, rarement)

Un poisson bleu à rayures jaunes :

(bleu, en fond) (jaune, en rayures)

Un poisson rouge vif parfois à queue blanche

(rouge, vif) (blanc, queue, parfois)

2.4. Définition du concept de nuance

Les exemples précédents montrent que nous donnons au concept de nuance un sens plus large que celui couramment admis.

En effet, pour nous, est nuance tout complément d'information apporté à une valeur, qu'il exprime une nuance au sens classique du terme (nuance de couleur par exemple comme vif), ou une fréquence (comme rarement), ou une localisation (comme "au bord") ou encore des circonstances (comme "en alarme").

Une nuance est un complément d'information sur une valeur

3. TYPOLOGIE DES NUANCES SELON LEUR SEMANTIQUE

L'étude d'exemples concrets d'informations nuancées nous a permis de répartir les nuances en 3 types, selon la nature de l'information complémentaire qu'elles apportent ; ces types sont :

- type incertitude,
- type imprécision,
- type limite.

Donnons une définition sémantique de chacun de ces types, illustrée par des exemples.

3.1. Type incertitude

Une nuance de type incertitude exprime soit la confiance que l'on accorde à une valeur soit sa fréquence d'apparition.

Exemples

Nuances exprimant la confiance : sûrement, certainement, possible, vraisemblablement. Ce type de nuance est utilisé par exemple dans les descriptions des individus isolés dont l'observation est difficile.

Nuances exprimant la fréquence : parfois, souvent, rarement, exceptionnellement. Ce type de nuance est utilisé pour la description des caractères variables des espèces ou des classes de phénomènes.

3.2.Type imprécision

Une nuance de type imprécision signifie que la valeur n'est pas stricte mais recouvre en fait un certain intervalle ou un certain ensemble de valeurs.

Exemples

Environ, voisin de, supérieur à, faible, forte, très, un peu.

Ce type de nuance peut se présenter à la fois dans la description des espèces ou des exemplaires à identifier.

Remarquons qu'il est parfois appliqué à une valeur inconnue; par exemple l'odeur d'un champignon peut être qualifiée simplement de forte alors que pour un autre on aura "odeur forte de farine".

Les nuances de type imprécision englobent donc à la fois ce que certains auteurs [BOS-85] appellent des valeurs vagues/floues et ce que d'autres [IME-84] appellent des valeurs incomplètes.

3.3. Type limite

Une nuance de type limite précise une condition d'observation ou d'apparition de la

valeur.

Exemples

Au bord, en plaques, en vieillissant, au frottement, en alarme, sur sol acide, par temps sec.

Ce type de nuance est nécessaire dans les domaines où les types de propriété utilisés sont trop généraux. Il est vrai que ces nuances pourraient être remplacées par l'introduction de nouveaux types de propriété plus précis, mais dans les univers réels, le nombre de types à introduire ainsi serait très élévé alors que la plupart d'entre eux n'auraient de sens, donc de valeurs, que pour un petit nombre d'individus. Par exemple l'information (noircissant, en vieillissant) donnée pour la couleur du pied d'une certaine espèce de champignon, pourrait être remplacée par l'introduction d'un nouveau type de propriété "changement de couleur au vieillissement", mais ce type de propriété ne serait valorisé que pour très peu d'espèces et donc compliquerait inutilement l'ensemble des descriptions par des valeurs nulles (ce sont des valeurs complètement inconnues ou des valeurs inapplicables (cf chapitre 2 §2.2.1.1.))

4. CONCLUSION

Le concept d'information nuancée est un type de données utile pour représenter ou interroger des informations incertaines, imprécises, incomplètes ou au contraire extrêmement précises. Sa complexité et la diversité des types de nuances qu'il inclut ont pour conséquence de soulever de nombreuses difficultés dans sa gestion par un SGBD ou tout autre logiciel.

Chapitre II GESTION DES INFORMATIONS NUANCEES DANS LES SYSTEMES D'INFORMATION. ETAT DE L'ART.

Chapitre II GESTION DES INFORMATIONS NUANCEES DANS LES SYSTEMES D'INFORMATION. ETAT DE L'ART.

Le présent chapitre vise à présenter un état de l'art des différents travaux de gestion des informations nuancées dans les systèmes d'information. Pour cela, nous distinguons les deux cadres de modélisation des systèmes d'informations suivants :

- le domaine des bases de données ,
- le domaine de l'intelligence artificielle.

Pour chacun de ces domaines nous montrons tout d'abord les possibilités et les limites des logiciels disponibles à l'heure actuelle pour la représentation et la gestion des informations nuancées. Puis nous présentons une synthèse bibliographique des travaux de recherche menés sur ce sujet.

Nous allons tout d'abord introduire certaines notions de la théorie des possibilités et de la théorie des sous-ensembles flous afin de pouvoir présenter certaines approches qui les utilisent.

1. PRELIMINAIRES : RAPPEL DES PRINCIPES DE LA THEORIE DES POSSIBILITES ET DES SOUS-ENSEMBLES FLOUS

Les principes fondamentaux de la théorie des sous-ensembles flous ont été introduits par Zadeh en 1965 **[ZAD-65]** . Ils ont été, ensuite, repris et généralisés par Zadeh **[ZAD-79]**, Dubois et Prade **[DUB-87]** qui les ont resitués dans le cadre formel de la théorie des possibilités.

La théorie des possibilités et la théorie des sous-ensembles flous conduisent à un type de logique échappant au schéma dichotomique des logiques classiques. En effet, les logiques

classiques n'offrent pas, selon les auteurs, suffisamment de nuances pour rendre pleinement compte de la réalité.

Nous allons présenter ici, tout d'abord le concept de sous-ensemble flou, puis les éléments fondamentaux de la théorie des possibilités en particulier le concept de distribution de possibilités et le concept de mesure de possibilité et de nécessité.

1.1. Concept de sous-ensemble flou

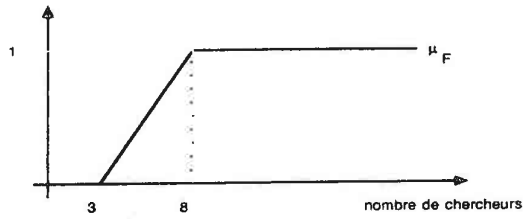
Vers 1965, Zadeh **[ZAD-65]** a proposé de prendre en compte au niveau ensembliste les concepts imprécis, vagues en introduisant la notion d'appartenance pondérée d'un élément à un ensemble. Jusque là, tout élément d'un univers se caractérisait soit par son appartenance soit par sa non appartenance à un sous-ensemble de cet univers. L'approche de Zadeh ne rejette pas cette forme d'appartenance mais l'étend en la modulant. Ainsi, il devient possible d'exprimer **la plus ou moins grande appartenance** d'un élément à un sous-ensemble que l'on qualifie alors de sous-ensemble flou.

Pour représenter un tel sous-ensemble, Zadeh utilise une **fonction caractéristique** (ou **fonction d'appartenance**) qui prend ses valeurs sur l'intervalle [0,1] plutôt que sur l'ensemble à deux éléments {0,1}. Un sous-ensemble flou F sur un univers U est défini par sa fonction caractéristique : $\mu_F : U \rightarrow [0,1]$

où $\mu_F(u)$ est le degré d'appartenance de u à F.

Les degrés 0 et 1 continuent à correspondre respectivement à la non-appartenance absolue et à l'appartenance totale.

Considérons par exemple le sous-ensemble flou F constitué des *grandes* équipes dans un laboratoire de recherche. Pour le définir, on associera à chaque équipe de recherche un coefficient d'appartenance, en fonction du nombre de chercheurs, de la manière suivante :



On appelle <u>support</u> de F l'ensemble : $\{u \in U / \mu_F(u) > 0\}$,

et <u>noyau</u> de F l'ensemble : $\{u \in U / \mu_F(u) = 1\}$.

. <u>Opérations ensemblistes sur les sous-ensembles flous</u>

Un sous-ensemble flou étant représenté par sa fonction caractéristique, les opérations entre sous-ensembles flous reviennent donc à des opérations entre leurs fonctions caractéristiques .

Soient F et G deux sous-ensembles flous de l'univers U et définis par les fonctions caractéristiques μ_F et μ_G.

. Inclusion

$$F \subset G \Leftrightarrow \forall u \in U, \mu_F(u) \leq \mu_G(u)$$

. Complémentation

La complémentation d'un sous-ensemble flou F est le sous-ensemble flou noté $\neg F$ défini par sa fonction caractéristique $\mu_{\neg F}$:

$$\forall u \in U, \mu_{\neg F}(u) = 1 - \mu_F(u)$$

. Intersection

$$\forall u \in U, \mu_{F \cap G}(u) = \text{Min}(\mu_F(u), \mu_G(u))$$

. Union

$\forall u \in U$, $\mu_{F \cup G}(u) = \text{Max}(\mu_F(u), \mu_G(u))$

. Produit cartésien

Soit F un sous-ensemble flou de l'univers U défini par sa fonction caratéristique μ_F et soit G un autre sous-ensemble flou de l'univers V défini par sa fonction caratéristique μ_G. Le produit cartésien FxG est un sous-ensemble flou défini par sa fonction caractéristique μ_{FxG} :

$\forall u,v \in UxV$, $\mu_{FxG}(u,v) = \text{Min}(\mu_F(u), \mu_G(v))$

Notes :

- Toutes ces opérations sur les sous-ensembles flous coïncident avec les opérations ensemblistes canoniques quand on se restreint à des ensembles à fonction caractéristique à valeurs dans {0,1}.
- Le couple (Min,Max) est une implantation possible de la théorie des sous-ensembles flous, il existe cependant, d'autres couples d'opérations que celui-ci pour définir l'intersection et l'union de deux sous-ensembles flous par exemple le couple d'opérateurs (Min(1,.+.),Max(0,.+.-1)). Pour une étude exaustive de l'ensemble des couples d'opérateurs possibles et de leurs propriétés, consulter l'ouvrage de Dubois et Prade [DUB-87].

1.2. Concept de distribution de possibilités

Zadeh a introduit le concept de distribution de possibilités [**ZAD-78**] qui repose sur la théorie des sous-ensembles flous. Les distributions de possibilités ont été introduites afin de pouvoir représenter et manipuler des propositions exprimées en langage naturel et considérées comme des propositions floues.

Par exemple, la proposition " *la température est assez élevée* " a un caractère flou en raison de la présence du terme " *assez élevée* " qui prête à des interprétations diverses.

Par définition [**ZAD-79**], une proposition floue est une proposition P avec la forme standard " **X est F**", où X est une variable prenant ses valeurs dans un univers U, qui peut être à plusieurs dimensions, et F un sous-ensemble flou de U qui traduit la plus ou moins

grande compatibilité entre les valeurs de U et le concept représenté par F.

Dans notre exemple, U est l'ensemble des valeurs de la température d'un domaine d'application donné, X est la variable température et F le sous-ensemble flou représentant l'information " *assez élevée* ".

Il est évident que la proposition " *la température est assez élevée* " n'apporte pas à elle seule la précision quant à la valeur exacte de la température. Néamoins, toute proposition floue du type " X est F " induit une distribution de possibilités, notée π_X, des valeurs susceptibles d'être prises par X. π_X est définie de l'univers U vers l'intervalle [0,1] :

$\pi_X : U \rightarrow [0,1]$

où $\pi_X(u)$ est le degré de possibilité que X prenne la valeur u.

π_X sera dite normalisée $\Leftrightarrow \exists u \in U$ telle que $\pi_X(u)=1$.

Plus précisément, la distribution de possibilités π_X est définie par la fonction caractéristique μ_F du sous-ensemble flou F qui représente les possibilités que la variable X prenne chacune des valeurs de U. La distribution de possibilités π_X, pour la proposition "X est F ", s'écrit :

$\pi_X(u) = \mu_F(u) \quad \forall u \in U.$

Ainsi la fonction caractéristique joue le rôle de la distribution de possibilités.

On vient d'introduire le concept de distribution de possibilités, dont il faut bien comprendre le sens. On considère souvent que la "possibilité" est la modalité tout ou rien, un événement étant possible ou ne l'étant pas. La notion de possibilité présentée ici est au contraire continue, de l'impossibilité complète à la possibilité totale, avec l'existence de degrés intermédiaires ordonnés. Elle correspond dans le langage courant aux expressions "très possible", "peu possible" qui qualifient un événement.

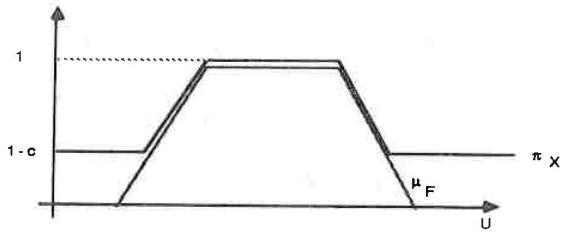
Les distributions de possibilités sont également utilisées pour représenter des propositions incertaines et/ou imprécises. Nous allons maintenant affiner la sémantique des distributions de possibilités en étudiant comment les déterminer pour plusieurs sortes de propositions (cette étude est inspirée des travaux de Buisson [BUI-87b]) :

. proposition " X est F (certitude c) " :

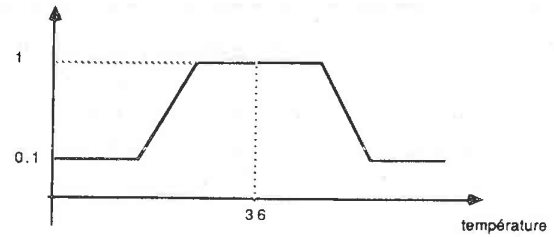
C'est la proposition la plus générale, qui peut être à la fois imprécise (ou floue) et incertaine. c est un nombre de [0,1], qu'on appellera le degré de certitude de la proposition : plus c est proche de 1, et plus on est certain que " X est F ".

Une telle proposition est représentée par la distribution de possibilités suivante :

$$\forall u \in U, \pi_X(u) = \text{Max}(\mu_F(u), 1-c) \qquad \text{où } c \in [0,1]$$



Une justification intuitive de cette formule peut être la suivante. Si on suppose la proposition vraie, la valeur de X peut être n'importe quel élément de U; par conséquent, toutes les valeurs ont une possibilité non nulle. Toutes les valeurs en dehors de F (de son support (cf §1.1.)) sont possibles au même degré 1-c. En ajoutant le niveau d'indétermination global 1-c, on arrive donc à exprimer l'incertitude de n'importe quelle proposition. Par exemple :



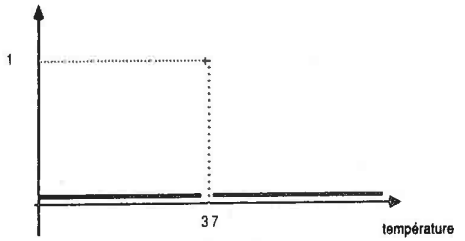
Cette proposition est floue et incertaine . Elle correspond à l'énoncé suivant : " *la température est très probablement d'environ 36 °C* " (le qualificatif *très probablement* a été traduit en un degré de certitude de 0.9).

Pour une justification théorique approfondie, on se reportera à **[DUB-88]**.

Nous allons montrer qu'avec cette forme générale de proposition, on peut représenter toute proposition certaine, qu'elle soit précise ou impécise (ou floue) :

+ Cas d'une proposition précise et certaine (c=1) :

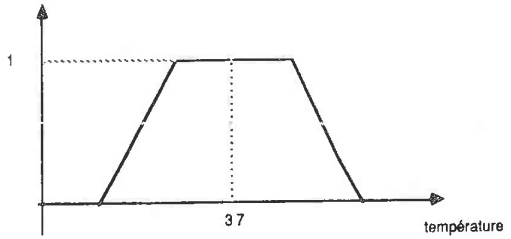
Par exemple " *la température du malade est 37* " :



+ Cas d'une proposition floue et certaine (c=1) :

Dans ce cas la distribution de possibilités est $\forall u \in U \; \pi_X(u) = \mu_F(u)$.

Par exemple " *la température du malade est d'environ 37 °C* " :



+ Cas où on ne sait rien quant à la valeur de X :

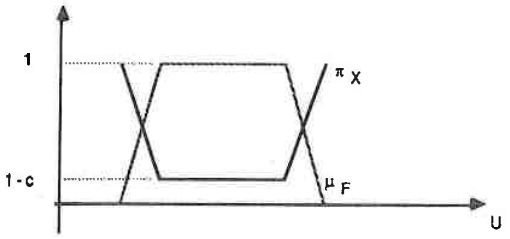
Dans ce cas toutes les valeurs de U sont tout à fait possibles, ce qui représente

l'indéterminisme complet. L'indéterminisme complet est représenté par $\forall u \in U\ \pi_X(u) = 1$.

. proposition " X n'est pas F (certitude c) " :

La distribution de possibilités associée à cette proposition est définie de la façcon suivante:

$$\forall u \in U, \pi_X(u) = \text{Max}(1-\mu_F(u), 1-c)$$



On peut faire la même étude de cas que précédement.

En conclusion, la distribution de possibilités induite par une proposition reflète la subjectivité de la personne qui la définit et le contexte dans lequel la proposition est énoncée.

La distribution de possibilités apparaît donc comme un formalisme simple capable de représenter de manière unifiée à la fois l'imprécision, le flou et l'incertitude des propositions.

1.3. Concept de mesure de possibilité et de nécessité

Considérons une proposition P (représentant une requête) et une donnée D (représentant l'état de connaissance d'un expert) faisant référence à la même variable X sur l'univers U et soient π_P et π_D leurs distributions de possibilités respectives. La compatibilité entre P et D est calculée, en théorie des possibilités, par deux mesures scalaires appelées :

- mesure de possibilité : $\Pi(P;D) = \text{Sup}_{u \in U}(\pi_P(u), \pi_D(u))$;
- mesure de nécessité : $N(P;D) = \text{Inf}_{u \in U}(\pi_P(u), 1-\pi_D(u))$.

Le concept de mesure de possibilité a été introduit par Zadeh **[ZAD-78]** , quant au concept de mesure de nécessité, il a été introduit par dualité à partir du premier concept (voir **[PRA-82c]**). Cette dualité entre possibilité et nécessité est exprimée par :

$N(P;D) = 1-\Pi(\neg P;D)$ où $\neg P$ est la négation de la proposition P.

La mesure de possibilité $\Pi(P;D)$ estime à quel degré il est possible que la valeur désigné par D soit compatible avec P; la mesure N(P;D) estime à quel degré cela est nécessaire (c'est à dire certain).

Ces deux mesures considérées ensemble donnent une estimation assez complète de la compatibilité de D avec P; on peut distinguer plusieurs classes de situations, par ordre de compatibilité décroissante :

- $\Pi(P;D)=1$ et $N(P;D)=1$: indique que $P \supset D$ (au sens des sous-ensembles flous); la compatibilité est totale.
- $\Pi(P;D)=1$ et $N(P;D)<1$: indique que les valeurs désignées par D sont tout à fait possiblement compatibles avec P, et que cela n'est pas totalement certain.
- $\Pi(P;D)<1$ et $N(P;D)=0$: indique qu'il est peu possible que les valeurs désignées par D soient compatibles avec P et que cela n'est pas du tout certain.
- $\Pi(P;D)=0$ et $N(P;D)=0$: indique que D n'est pas du tout compatible avec P.

On notera que la mesure $\Pi(P;D)$ est un degré d'intersection entre l'ensemble des valeurs possibles par D et celui des valeurs possibles par P, tandis que N(P;D) est un degré d'inclusion entre ces deux ensembles flous. Ainsi on a $N(P;D) \leq \Pi(P;D)$ ce qui exprime que N est une mesure de compatibilité plus 'exigeante' que Π.

2. EN BASE DE DONNEES

Nous limitons notre étude aux SGBD relationnels dans la mesure où toutes les extensions ou améliorations proposées dans les travaux de recherche concernent ce type de SGBD.

Nous montrons tout d'abord que les SGBD relationnels disponibles sur le marché actuellement et commençant à se répandre très largement dans les entreprises ne permettent pratiquement aucune représentation et gestion des informations nuancées.

Nous étudions ensuite les propositions effectuées en recherche pour la résolution de ces

problèmes.

2.1. Limites des SGBD actuels

Un SGBD classique est composé :

- d'un Langage de Définition des Données (LDD) qui permet de créer la base de données et de définir son schéma ,
- d'un Langage de Manipulation des Données (LMD) qui permet d'interroger et de modifier des données.

Ces deux langages sont complétés par un ensemble d'outils logiciels assurant les fonctions de mémorisation et de sécurité des informations.

2.1.1. Langage de Définition des Données

Tous les SGBDs relationnels actuels tels que INFORMIX, ORACLE, INGRES :

- imposent la 1ère Forme Normale (1FN), par conséquent dans chaque n-uplet les attributs sont monovalués,
- proposent des domaines très pauvres, voisins de ceux des langages de programmation classiques,
- offrent un seul type d'information nuancée : l'information incomplète par l'existence d'une valeur nulle adjointe à chaque domaine.

2.1.2. Langage de Manipulation des Données

Le langage d'expression d'une requête peut être basé :

- soit sur le calcul des prédicats : la requête est alors une formule logique,
- soit sur l'algèbre relationnelle : la requête est alors une suite d'opérations algébriques telles que la projection, la sélection, la jointure ...

Dans ce paragraphe, nous allons nous contenter d'étudier les limites des langages algébriques puisqu'il y a équivalence entre les deux modes de formulation [DEL-82].

Les opérateurs de base de l'algèbre relationnelle [COD-79] sont l'union, le produit cartésien, la différence, la projection et la sélection. Ce dernier opérateur est fondamental dans un langage d'interrogation; pour cela nous nous contentons d'étudier ses limites.

L'opérateur de sélection permet de sélectionner dans une relation les n-uplets qui satisfont une condition donnée. Cette condition est une formule booléenne utilisant des opérateurs de comparaison $<, >, \geq, \leq, \neq, =$ et les connecteurs ET, OU, NON .

L'emploi de la logique booléenne, dans la condition de sélection, est sujette à critique pour de nombreuses raisons :

- la difficulté de formulation des requêtes,
- la brutalité des comparaisons : le résultat d'une condition de sélection est soit vrai soit faux,
- le manque d'interactivité avec l'utilisateur.

2.1.2.1. Difficulté de formulation

Pour interroger une base de données, l'utilisateur formule d'abord sa requête en langage naturel puis la traduit dans le langage d'interrogation du SGBD.

La difficulté de formulation réside :

- d'une part dans la traduction des termes vagues et imprécis du langage naturel en termes précis que le système peut gérer. Par exemple, le terme "jeune" appliqué à l'âge d'une personne pourra être traduit par l'intervalle [14,36],
- d'autre part dans l'établissement des connecteurs logiques ET, OU entre les conditions élémentaires d'une requête. En effet, les connecteurs logiques exigent de l'utilisateur un apprentissage qui n'est pas toujours facile à acquérir.

2.1.2.2. Rigidité des comparaisons

Lorsqu'on applique une condition booléenne, le résultat de la comparaison est binaire : VRAI ou FAUX, mais jamais entre les deux ni les deux à la fois.

Par exemple, l'évaluation de la comparaison "âge $\geqslant$ 14 ET âge $\leq$ 36" , pour obtenir les personnes jeunes, donnerait le même résultat VRAI pour les valeurs 15 ans et 35 ans, et le même résultat FAUX pour les valeurs 13 ans 11 mois et 36 ans 1mois.

On remarque, ainsi, le manque de tolérance de la comparaison booléenne pour obtenir les données qui sont très proches de la condition.

Notons également le rôle joué par les connecteurs logiques ET, OU dans l'acceptation ou le rejet d'une donnée. En effet, en réponse à une requête contenant un ou plusieurs ET, une donnée vérifiant toutes les conditions sauf une est aussi mauvaise qu'une donnée ne vérifiant aucune condition.

2.1.2.3. Manque d'interactivité

Les systèmes d'interrogation proposés jusqu'à présent sont tous du type question-réponse; c'est à dire que si l'utilisateur pose une question (une requête relationnelle) le système retournera une réponse précise et rien d'autre. Dans la plupart des cas cette réponse ne satisfait pas l'utilisateur car :

- elle est incomplète, dans le sens où l'utilisateur est intéressé par d'autres informations qui ne sont pas explicitement demandées mais qui sont implicitement reliées à la requête. Par exemple, un utilisateur qui veut s'acheter une voiture pas très chère, pose la question suivante à une base de données de voitures d'occasion :

quelles sont les voitures qui coûtent entre 40000 et 50000 francs ?

La réponse fournie par un SGBD relationnel standard pourrait être par exemple :

VOITURE	PRIX
Peugeot 205	47000
Peugeot 205	42000
Renaut Super 5	43000

Cette réponse est incomplète aux yeux de l'utilisateur qui implicitement désirait connaitre l'année de mise en circulation de la voiture, le kilométrage, le nombre de chevaux fiscaux et l'état de la voiture, elle ne répond donc pas tout à fait à son besoin.

Ce cas est très fréquent lorsque l'utilisateur n'a pas une idée très précise de ce qu'il cherche comme c'est souvent le cas lorsqu'il interroge une base de données dans l'objectif de prendre une décision.

L'inefficacité des systèmes d'interrogation actuels provient du fait qu'ils ne permettent pas à l'utilisateur de donner son avis sur la réponse obtenue et ne lui fournissent aucune aide ni pour progresser dans son interrogation ni pour reformuler sa requête.

2.2. Etat de l'art

Dans ce paragraphe, nous présentons une étude bibliographique des travaux menés en bases de données pour la gestion des informations nuancées. Pour cela, nous classons les différents travaux de recherche suivant leur niveau d'introduction des informations nuancées en :

- bases de données nuancées : bases de données proposant un modèle pour la représentation de certaines informations nuancées,
- systèmes d'interrogation nuancée : SGBD permettant de traiter des requêtes comportant des informations nuancées.

2.2.1. Base de données nuancée

Une base de données est dite nuancée lorsque l'univers modélisé contient des informations nuancées et qu'elle offre un modèle pour leur représentation.

Parmi les travaux réalisés pour la modélisation d'une base de données nuancée, on distingue deux catégories suivant que les informations nuancées sont :

- incomplètes ou imprécises,
- incertaines.

2.2.1.1. Informations incomplètes ou imprécises

Dans la théorie des bases de données relationnelles, la modélisation des informations nuancées a souvent été réduite à la prise en compte des informations incomplètes en particulier **les valeurs nulles** [COD-79] . En général la valeur nulle correspond soit à une valeur inconnue soit au cas où la propriété correspondante est inapplicable.

Entre la connaissance complète d'une valeur et son absence totale, existent des situations intermédiaires où l'information est partielle [GRA-79] [JAE-79] [LIP-81] [IMI-81]; on est alors en présence d'informations imprécises.

Par exemple, la connaissance approximative de la taille d'une personne par "entre 1.70m et 1.80m" exprime l'imprécision concernant la valeur de la taille tout en sachant que celle-ci

appartient à un ensemble de valeurs possibles [1.70,1.80].

Parmi les approches traitant les informations partielles, citons celles de Lipski [LIP-79] [LIP-81] et Lipski Imielinski [IMI-81]. Dans la dernière approche les auteurs proposent un modèle dont le principe est d'associer à chaque élément inconnu un sous ensemble du domaine. Si on reprend l'exemple précédent, l'information "entre 1.70m et 1.80m" sera représentée par le sous ensemble [1.70,1.80] qui indique la validité de la taille de la personne.

D'autres approches, souvent rencontrées pour modéliser les informations incomplètes ou imprécises, consistent à utiliser des distributions de possibilités ou des concepts apparentés. Elles sont principalement basées sur la théorie des sous ensembles flous de Zadeh [ZAD-65] [ZAD-79] ou la théorie des possibilités également de Zadeh [ZAD-78] [PRA-82a] [PRA-82b]. Parmi ces approches, citons celles de Buckles et Petry [BUC-82a] [BUC-82b] [BUC-83], d'Umano [UMA-82] [UMA-83], de Prade et Testemale [PRA-84] [PRA-83b] [TES-84] :

- l'approche de Buckles et Petry consiste à représenter les valeurs possibles d'un attribut par des ensembles (non flous) d'éléments considérés comme interchangeables par rapport à une relation de similarité et à un seuil donné. C'est la relation de similarité qui permet de tenir compte de l'imprécision, car elle modélise à quel point deux éléments sont interchangeables,

- Umano, dans son approche, propose de représenter la valeur d'un attribut A (de domaine D) par un sous ensemble flou de D $\cup$ {NULL} où NULL représente la valeur complétement inconnue. Cette approche est fondée sur le concept de distribution de possibilités.

- l'approche proposée par Prade et Testemale présente un cadre plus général que les deux précédentes et mérite un plus long développement : le modèle de données utilisé consiste à représenter les informations dans des relations dites "étendues" et définies de la façon suivante : soient $D_1,...,D_n$ des domaines attachés à des attributs $A_1,...,A_n$; la valeur d'un attribut A_i est une distribution de possibilités (ou un ensemble flou, si l'on préfère) normalisée (cf §1.2.) sur $D_i \cup \{e\}$ si A_i est monovalué et sur $P(D_i)$ (ensemble des parties de D_i) si A_i est multivalué. {e} est un extra-élément permettant de prendre en compte les situations où il existe une possibilité non nulle que l'attribut A_i ne s'applique pas et une

possibilité non nulle qu'il s'applique.

Exemple (extrait de [TES-84]) : considérons une relation "étendue" définie par la note obtenue par un étudiant à un examen donné, par l'âge de cet étudiant et par les langues vivantes qu'il parle :

NOM	AGE	NOTE	LANGUE
Jean	19	*bonne*	*français-et-anglais*
Paul	*jeune*	*environ-12*	*au-moins-l'espagnol*

bonne , *jeune* , *environ-12* , *français-et-anglais* , *au-moins-l'espagnol* , sont des étiquettes de distributions de possibilités; par exemple, soit D={anglais,espagnol,français,italien, russe} l'univers de l'attribut LANGUE, la valeur *au-moins-l'espagnol* représente la distribution de possibilités π définie sur P(D) par :

$\pi(S) = 1 \quad \forall S \supset \{espagnol\}$
$\pi(S) = 0$ pour toute autre partie S de D.

Dans la pratique, chaque valeur présente dans une relation désigne une distribution de possibilités à définir, ainsi il y a autant de distributions de possibilités que de valeurs différentes, ce qui n'est pas envisageable dans une application réelle.

En conclusion, la généralité de ce modèle devrait permettre de traiter tous les types de nuances répertoriés au chapitre I. Cependant, nous émettons quelques réserves pour le type "limite" (cf chapitre 1 §3.3.), car il nous semble que des concepts utilisés en intelligence artificielle soient mieux adaptés pour rendre compte de l'extrême variété de certains domaines réels.

2.2.1.2. Informations incertaines

Une information est incertaine, s'il existe un doute quant à sa vérité. Ce doute peut être dû :

- à un manque de confiance dans la source d'information,

- à l'appartenance de l'information à un univers difficilement accessible à la vérification.

L'approche la plus souvent rencontrée pour représenter les informations incertaines

revient à attacher une valeur de vérité floue (nombre compris entre 0 et 1) à chaque n-uplet de la base de données; c'est le cas dans les approches de Haar [HAA-77], Philips, Beaumont, Richardson [PHI-79], Freksa [FRE-80] (celui-ci utilise plutôt des valeurs de vérité linguistiques représentées par des distributions de possibilités sur [0,1]), Baldwin [BAL-83a] [BAL-83b] et Umano [UMA-83] (qui utilise des valeurs de vérités définies par des distributions de possibilité). Ces différentes approches sont détaillées dans [TES-84].

En dehors des distributions de possibilités, nous devons mentionner les travaux de Wong [WON-82] qui propose une approche statistique (utilisant des distributions de possibilité) pour modéliser une information incertaine. L'approche proposée est fondée sur l'idée d'utiliser deux bases de données : une base de données idéale où toutes les valeurs des attributs sont connues avec précision, et une base de données courante où, pour chaque n-uplet, on ne connaît avec précision que quelques valeurs d'attributs. Les autres valeurs (manquantes ou erronées) ne sont connues que par des distributions de probabilité.

2.2.2. Système d'interrogation nuancé

Un système d'interrogation est considéré comme nuancé lorsqu'il est capable de traiter des requêtes contenant des valeurs nuancées ou lorsqu'il peut fournir une réponse nuancée.

Exemples de requêtes nuancées :

- donner les espèces de champignons à chapeau *blanchâtre*
- lister les employés *jeunes*
- trouver les personnes ayant une *grande* taille et pesant *environ 75kg*

Une requête nuancée peut être effectuée :

- soit sur une base de données précise [TAH-77] [HAM-86] [BOS-87],
- soit sur une base de données nuancée [UMA-82] [BAL-83a] [WON-82] [PRA-85] [TES-84].

Une réponse nuancée peut être fournie également sur les deux types de bases de données.

Nous allons dans ce paragraphe situer les différents travaux selon que la base de données est précise ou nuancée, puis nous terminons par une étude des problèmes d'interactivité dans les systèmes d'interrogations.

2.2.2.1. Requête nuancée sur une base de données précise

L'approche proposée par Bosc [BOS-87] et Hamon [HAM-86] consiste à étendre le langage SEQUEL (Structured English QUEry Langage). L'extension porte sur le système d'interrogation en redéfinissant les opérateurs relationnels et ensemblistes tels que la projection, la sélection, la jointure, l'union, l'intersection ... dans le cadre de la logique floue. Les agrégats permettant d'obtenir la somme, la moyenne ... sont également redéfinis.

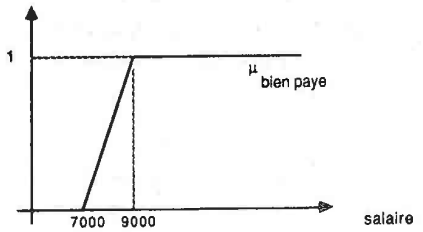
Le langage étendu appelé SEQUEL^F est fondé sur le bloc de base de SEQUEL : SELECT FROM WHERE. Il permet d'utiliser comme prédicats, dans la clause WHERE, des fonctions floues, par exemple (extrait de [HAM-86]) la question :

quels sont les employés bien payés ?

peut être exprimée par :

SELECT No-employe, Nom
WHERE *bien-payé* (salaire)

où *bien-payé* est une distribution de possibilité qui peut être définie par :



Le résultat d'une requête SEQUEL^F est une relation floue constituée de couples (n-uplet, coefficient de vraisemblance).

2.2.2.2. Requête nuancée sur une base de données nuancée

Après avoir présenté, au paragraphe 2.2.1.1., le modèle de Testemale [TES-84] pour représenter des domaines nuancés, nous allons étudier le système d'interrogation en accord

avec celui-ci.

Le système d'interrogation proposé par Testemale est fondé sur l'extension des principaux opérateurs de l'algèbre relationnelle afin de pouvoir adresser des requêtes nuancées sur la base de données nuancée, tout en utilisant la théorie des ensembles flous et la théorie des possibilités.

Parmi les opérateurs étendus, nous allons nous limiter à la présentation de l'opérateur, fondamental dans un système d'interrogation : l'opérateur de sélection qui est appelé sélection généralisée .

L'utilisation de distributions de possibilité pour représenter les valeurs nuancées induit une mesure de possibilité et une mesure de nécessité pour évaluer le fait qu'un n-uplet répond à une condition de sélection. Par conséquent, le résultat d'une sélection $\sigma(R,\mathfrak{C})$ (c'est à dire sélection appliquée à la relation R avec la condition $\mathfrak{C}$) est constitué de deux ensembles flous; l'ensemble des n-uplets de R qui satisfont possiblement la condition $\mathfrak{C}$ et l'ensemble des n-uplets qui satisfont nécessairement la condition $\mathfrak{C}$ [PRA-83a] [PRA-83b]. La condition $\mathfrak{C}$ peut être définie à partir de sous conditions combinées entre elles par des ET ou des OU.

Les conditions d'une sélection sont étendues afin d'accepter des comparateurs flous tels que *approximativement égal à* , *beaucoup plus grand que* ainsi que des valeurs floues telles que *grand* , *jeune, bonne, environ-10* qui sont définies par des distributions de possibilité.

Les mesures globales (possibilité Π et nécessité N qu'un n-uplet de la relation R soit compatible avec la condition $\mathfrak{C}$) sont obtenues par agrégation des mesures de possibilité Π_i et de nécessité N_i intermédiaires (correspondant aux sous conditions $\mathfrak{C}_i$ composant $\mathfrak{C}$) au moyen de l'opérateur MIN (Minimum).

En conclusion, le modèle proposé par Testemale [TES-84] nous paraît le plus complet parmi les approches citées , cependant nous émettons à son égard les réserves suivantes :

- la définition d'une distribution de possibilité pour chaque information nuancée nous semble très coûteuse et fastidieuse pour une application concrète, par exemple : les informations nuancées *environ-5, environ-6* ,... doivent être définies par des distributions de possibilités différentes,

- l'utilisation de l'opérateur MIN pour combiner les mesures de possibilité et de nécessité intermédiaires risque dans certains cas d'être trop stricte. En effet, un n-uplet ayant une seule mesure Π_i nulle et les autres mesures Π_j non nulles, sera écarté même si la condition $\mathfrak{C}_i$

(correspondant à la mesure Π_i=0) n'est pas sémantiquement très importante.

Notons enfin que les auteurs [UMA-82] [UMA-83] [BAL-82a] [BAL-83b] [WON-82] des approches citées au paragraphe 1.2.1. proposent également des systèmes d'interrogation en accord avec les modèles de représentation qu'ils avaient proposés.

2.2.2.3. Interactivité dans les systèmes d'interrogation nuancée

Le but de l'interactivité dans les systèmes d'interrogation nuancée est :

- de guider les utilisateurs indécis, par exemple si un utilisateur pose une question demandant des *véhicules*, un système interactif et coopératif guiderait l'utilisateur à préciser sa question en lui proposant d'autres termes pour formuler sa question, par exemple *voiture, moto, bus,....* ;

- d'aider les utilisateurs qui savent ce qu'ils veulent mais sont incapables de formuler leurs requêtes correctement du premier coup ;

- d'élargir la recherche, en cas de manque de réponses à la question posée, à des informations voisines qui conviendraient également à l'utilisateur. Par exemple, à un utilisateur voulant partir de Nancy à Nice en train entre 9 et 12 heures, un SGBD classique répondrait par une impossibilité, alors qu'un élargissement de la recherche donnerait d'autres solutions, comme par exemple des horaires d'avions ou de trains partant avant 9 heures du matin.

Parmi les solutions proposées pour résoudre les trois problèmes précédents, nous citons celle de Cuppens et Demolombe [CUP-87] [CUP-88] qui travaillent sur un système permettant de reconnaître les centres d'intérêt d'un utilisateur pour fournir des réponses coopératives, et le système EXPRIM [CRE-86] [HAL-88] [CRE-89] qui a pour objectif l'interrogation progressive d'une base d'images avec une aide à la reformulation des questions :

- Cuppens et Demolombe proposent une méthode pour fournir des informations supplémentaires intéressantes en réponse à des requêtes posées à une base de données relationnelle. Ces informations intéressantes sont déduites grâce à une base de connaissances contenant les règles codifiant le savoir-faire d'un expert habitué à fournir des réponses à des

utilisateurs occasionnels.

La base de données est décrite par des entités, des attributs, des relations et des *thèmes* . Les *thèmes* sont associés aux attributs et aux relations et permettent de regrouper les informations de la base de données qui appartiennent à un même champ sémantique. Par exemple, l'heure de départ et l'heure d'arrivée d'un vol appartiennent au même *thème* HORAIRE.

Les bases de données et de connaissances sont utilisées pour transformer la requête initiale afin d'y ajouter les informations supplémentaires à fournir à l'utilisateur. Par exemple, soit la requête suivante :

Quelle est l'heure de départ des vols Paris / New york qui partent entre 7 et 11 heures ?

Une réponse à la requête initiale pourrait être par exemple :

VOL	Heure de départ
AF001	10h00
AF001	11h00
AF015	10h30

Une réponse à la requête transformée, après utilisation du savoir-faire d'une personne travaillant dans une agence de voyage, pourrait être par exemple :

VOL	Heure de départ	Heure d'arrivée	Période	Prix/ff
AF001	10h00	08h45	01/01 27/09	27,715
AF001	11h00	09h45	28/09 31/12	27,715
AF015	10h30	14h55		

Cette réponse est plus complète et plus intéressante pour une personne qui doit, par exemple, prendre une correspondance. Elle pourrait être encore plus riche et plus coopérative si on y ajoutait : les villes des escales, les réductions qu'on peut avoir sur certains vols, etc ..

- Le système EXPRIM est réalisé au CRIN en vue d'aider l'utilisateur à interroger et à explorer une base d'images (stockée par exemple sur vidéodisque) par l'intermédiaire d'une base de descriptions.

L'exploration s'effectue à partir d'une requête initiale, formulée soit sous forme de mots par l'utilisateur, avec ou sans aide, soit par le système après que l'utilisateur ait choisi des images dans un "catalogue d'images" proposé par le système. Celui-ci procède ensuite au

modelage progressif et interactif de la requête, donnant un rôle important à la visualisation et à la sélection d'images par l'utilisateur. Il permet ainsi à l'utilisateur d'exprimer de mieux en mieux ses besoins et même, dans une certaine mesure, de les découvrir par approches successives.

Le processus d'interrogation consiste en l'itération d'une succession de trois phases non complètement distinctes :

+ recherche documentaire (phase dite "avant visualisation"), faisant largement appel à un thésaurus, destinée à sélectionner un ensemble d'images, sans chercher à trop éviter les "bruits" et parfois même en les générant;

+ visualisation interactive des images sélectionnées parmi lesquelles l'utilisateur effectue son choix;

+ exploitation des choix effectués lors de la visualisation (phase "après visualisation") , basée sur une sorte d'analyse discriminante des choix de l'utilisateur et destinée à découvrir de nouveaux critères de sélection d'images afin de les intégrer avec l'ancienne requête pour en créer une nouvelle ayant des chances de fournir de nouvelles images intéressantes.

En conclusion, nous pouvons constater que dans ces deux solutions, ni la requête initiale ni le processus de recherche ne permettent de manipuler des informations nuancées et qu'aucune évaluation de la proximité entre la réponse et la requête initiale n'est effectuée afin d'apprécier la qualité de la réponse.

3. EN INTELLIGENCE ARTIFICIELLE

On distingue entre autres, deux domaines d'application de l'Intelligence Artificielle :

- la reconnaissance des formes,
- les systèmes à base de connaissances.

Dans ce paragraphe, nous nous intéresserons uniquement aux systèmes à base de connaissances et en particulier aux systèmes experts tout en sachant que les problèmes liés à la gestion des informations nuancées se posent également en reconnaissance des formes [ROM-87][HIR-87] [FAR-86].

Nous mettons tout d'abord en évidence les problèmes se rattachant à la gestion des informations nuancées, ensuite nous terminons par un état de l'art des différentes approches proposées.

3.1. Limites des systèmes experts

Un système expert comporte essentiellement :

-***Une base de connaissances*** qui est un ensemble d'informations censé représenter le monde réel d'une application. On distingue dans une telle base :

+ "les faits décrivant des situations considérées soit comme établies (c'est à dire faits avérés),soit à établir (c'est à dire faits poursuivis ou hypothétiques)." ; [FAR-85]

+ "les règles qui représentent le savoir-faire des experts du domaine de l'application :elles indiquent quelles conséquences tirer ou quelles actions accomplir lorsque telle situation est établie ou est à établir ."; [FAR-85]

-***Un moteur d'inférence*** qui est un mécanisme d' accès et de traitement de la base de connaissances, utilisé pour répondre à une requête utilisateur ;

-***Un module de dialogue*** pour communiquer avec l'utilisateur.

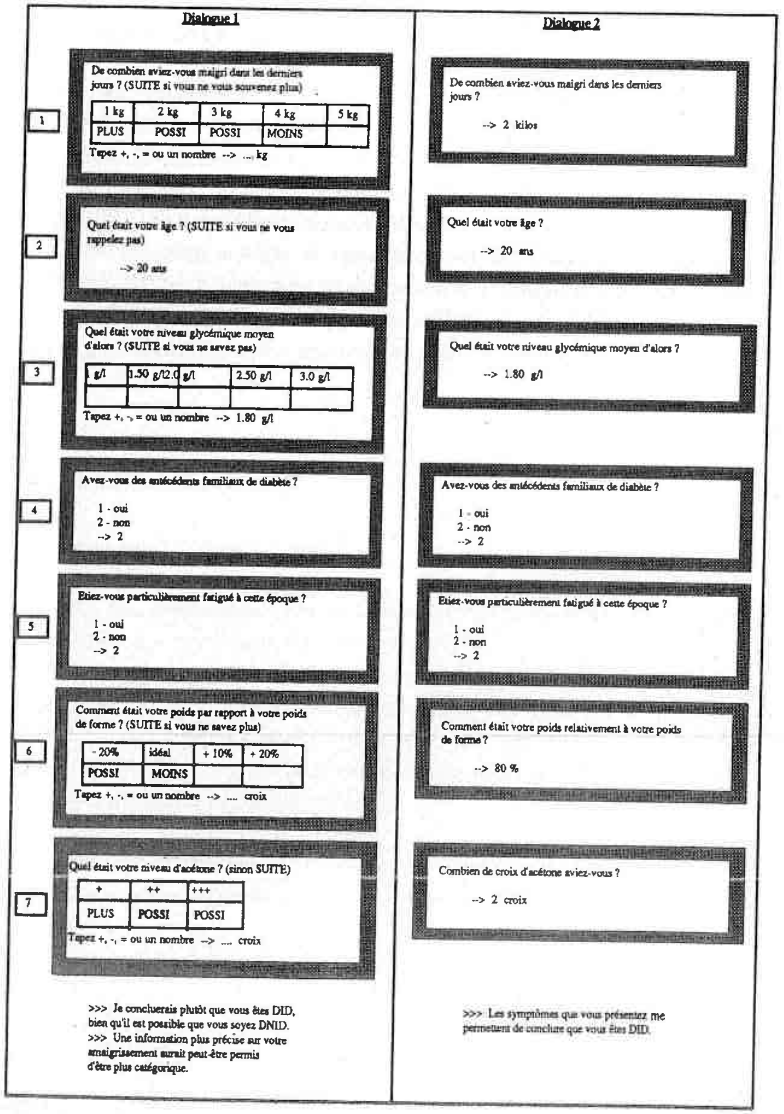
Dans la suite de ce paragraphe, nous allons étudier les défauts présentés par un système expert utilisant des faits et des règles certains et non vagues.

3.1.1. Faits et règles certains et non vagues

Nous allons illustrer , à travers quelques exemples, les problèmes que peuvent provoquer des faits et des règles certains et non vagues.

Exemple 1: Détermination du type de diabète (extrait de la thèse de Buisson [BUI-87])

Le but de cet exemple est de déterminer le type de diabète à partir d'un ensemble de données fournies par le malade et d'un ensemble de faits et de règles d'expertise exprimés par les médecins.



Notons que le dialogue 1 permet de fournir des informations imprécises par l'intermédiaire de tableaux prédéfinis de valeurs, tandis que le dialogue 2 n'utilise que des informations précises et exactes.

Le dialogue 1 est donc plus riche que celui du dialogue 2 car une certitude ou une précision sur la valeur d'un paramètre permet cependant d'avoir une indication sur ses bornes supérieures ou inférieures. Ainsi, en écrivant POSSI en face d'une valeur, on indique que celle-ci "est peut être" la valeur du paramètre. En écrivant PLUS, on indique que la vraie valeur "est plus" que la valeur donnée; en écrivant MOINS,on indique que "c'est moins".

D'autre part, la conclusion nuancée du dialogue 1 est plus satisfaisante et mieux adaptée à la réalité que la réponse du dialogue 2. En effet, le cas médical n'étant pas très typique, les deux diagnostics opposés DID,DNID ne sont pas à exclure, il est donc necessaire de nuancer la conclusion.

A travers cet exemple , il apparaît que le recueil des faits et des règles est associé à une imprécision et une subjectivité qui se traduisent par la difficulté de donner avec exactitude la valeur d'un paramètre (attribut).

En conclusion, notons que l'absence de nuances dans les faits et dans les règles d'un système expert risque de donner des déductions et des conclusions dangereuses dans certaines applications (notamment médicales) .

3.1.2. Insuffisance de la logique des propositions

• Dans la logique des propositions, les connaissances -faits et règles- sont représentées sous forme d'assertions logiques.Une assertion [BON-84], appelée proposition, est affectée de l'une des valeurs possibles VRAI ou FAUX .Il est possible de représenter des propositions complexes en utilisant les connecteurs logiques $\wedge$ (ET), V (OU), $\neg$ (NON), $\rightarrow$ (SI ALORS), $\leftrightarrow$ (SI ET SEULEMENT SI).

Pour l'écriture d'une règle on utilisera le vocabulaire **si** <prémisse> **alors** <conclusion>.

Le processus d'inférence de la logique des propositions repose sur les règles dites de

détachement suivantes :

- Le modus ponens $(p, (p \rightarrow q)) \rightarrow q$ où p et q sont des propositions,
- Le modus tollens($\neg q, (p \rightarrow q)) \rightarrow \neg p$.

Le modus ponens et le modus tollens, d'une part s'appliquent à des valeurs de vérité de propositions et d'autre part fournissent un seul résultat (VRAI ou FAUX) uniquement dans le cas où les deux prémisses sont simultanément vraies.Ceci est loin de rendre compte de la réalité et du raisonnement humain qui, face à une situation, fait évoluer l'information, qu'elle soit imprécise ou incertaine, et value ses conclusions sur une échelle plus fine que les deux valeurs VRAI ou FAUX afin de prendre en compte des concepts imprécis, incertains,...

Exemple : emprunté à la médecine :

(P: amaigrissement élevé → Q: acétone élevée), (P': amaigrissement trés élevé).
Que peut-on conclure sur Q ?

Il est évident qu'en appliquant le modus ponens, on ne conclurait rien sur Q car P et P' sont considérées comme deux propositions différentes, bien qu'elles soient incluses l'une dans l'autre (si on suppose que le sens "trés élevé" est inclus dans celui de "élevé").

La logique des propositions est donc très insuffisante pour la représentation et le raisonnement sur des informations nuancées.

Dans la suite, on étudiera un schéma d'inférence, proposé initialement par Zadeh, tenant compte des informations nuancées et qui sera une forme généralisée du modus ponens.

3.2. Etat de l'art

Dans la première partie de ce paragraphe, nous faisons un survol des différentes approches proposées pour la représentation et la manipulation des informations nuancées dans le domaine de l'Intelligence Artificielle et en particulier dans les Systèmes Experts. Une deuxième partie développe quelques exemples de Systèmes Experts intégrant une représentation de faits et/ou de règles imprécis et/ou incertains et un modèle de raisonnement adéquat.

3.2.1. Approches proposées

En raison de la difficulté de représenter et de manipuler des faits et/ou des règles imprécis et/ou incertains, diverses approches ont été proposées, parmi lesquelles on peut citer :

- les approches probabilistes [DED-76] [SKA-78] basées pour la majeure partie sur l'application du théorème de Bayes,
- les approches utilisant la théorie des sous-ensembles flous [ZAD-65] et la théorie des possibilités [DUB-87],
- l'approche basée sur les fonctions de croyance de Shafer [SHA-76] .

3.2.1.1. Approches probabilistes

Pendant longtemps, les approches probabilistes étaient les seules approches numériques au problème de l'inférence incertaine.

En raison du cadre normatif du théorème de Bayes, des chercheurs en Intelligence Artificielle ont éprouvé le besoin d'alternatives au modèle bayésien standard et ont proposé des modèles plus empiriques en particulier dans Mycin [SHO-75] ou dans Prospector [DUD-81] (voir aussi [FRI-81] [KAY-79]) .

L'application du théorème de Bayes nécessite de connaître les probabilités de chacuns des événements possibles mutuellement disjoints (ou indépendants), ainsi que les probabilités conditionnelles de la présence d'un événement E_i par rapport aux autres événements E_j ($i \neq j$). Par exemple, dans un système d'aide à la décision médicale, l'application de l'approche bayésienne nécessiterait de connaître la probabilité de chacun des diagnostics ainsi que les probabilités conditionnelles de la présence des signes pour chacun des diagnostics.

Du point de vue pratique, il est clair que le nombre élevé d'informations (mesures de probabilités) à fournir entraîne certaines complications dans une application réelle; à savoir l'identification, par l'utilisateur, de tous les événements indépendants et la donnée des mesures de probabilités précises alors que la connaissance humaine a tendance à être approximative et imprécise.

Pour résumer, l'approche probabiliste ne peut pas prendre en compte simplement et convenablement les informations nuancées à cause de son cadre trop normatif.

3.2.1.2. Approches : théories des possibilités et des sous ensembles flous

A partir de ces théories, de nombreux systèmes experts ont été développés intégrant des schémas de raisonnement avec des faits et des règles incertains et/ou imprécis. Parmi ces systèmes citons quelques systèmes experts médicaux comme SPHINX [FIE-81], PROTIS [SOU-82], CADIAG-2 [ADL-82], DIABETO [BUIS-87a], TOULMED [BUI-87b], le système MANAGER [ERN-82] appliqué au management, le système SPERIL [ISH-81] appliqué au génie civil, le sytème ELFIN [CLO-84] appliqué à la recherche de nappe pétrolière.

Des moteurs d'inférence généraux ont été également développés, pour manipuler l'incertitude et l'imprécision, sur les bases de la théorie des possibilités et des sous ensembles flous, par exemple le moteur TAIGER [FAR-86] qui a été appliqué à l'analyse financière, le moteur SPII [CLO-85] qui a été développé pour réaliser des systèmes comme ELFIN.

3.2.2. Quelques Systèmes Experts

La plupart des systèmes experts utilisant ou manipulant des informations imprécises et/ou incertaines sont issus du domaine médical.

Dans ce paragraphe, nous présentons quatre systèmes experts issus de ce domaine : Mycin, Toulmed, Protis, Sphinx. Les fondements des approches de Mycin et de Sphinx sont relativement empiriques, alors qu'ils sont plus théoriques pour les autres. Pour cette raison, nous proposons d'étudier un système de chaque catégorie : Mycin et Toulmed, puis nous nous contenterons de présenter les principales caractéristiques des autres systèmes.

3.2.2.1. MYCIN

Le nom de Mycin vient du suffixe commun à plusieurs agents antimicrobiens. Il a été développé entre 1972 et 1976 à l'Université de Stanford (USA) pour assister des médecins non spécialistes dans le diagnostic et la thérapeutique des maladies bactériennes du sang.

3.2.2.1.1. Représentation de la connaissance

Mycin manipule :

- des *contextes* : représentant les différents environnements ou entités relatifs au problème posé. Ils sont créés dynamiquement en cours de consultation, chaque contexte étant une instanciation d'un type de contexte prédéfini : les patients, les cultures bactériennes courantes , etc .....
- des *paramètres cliniques* : ils décrivent les caractéristiques du contexte. A chaque paramètre sont associées différentes connaissances, qui donnent les moyens et les méthodes pour le manipuler. Un paramètre peut être de trois type : monovalué, multivalué ou de type Oui/Non.

Pour représenter l'incertitude sur la valeur d'un paramètre, Mycin gère un **facteur de certitude** CF $\in$ [-1,1] (Certainty Factor). Soient <p> un paramètre de domaine D associé à un contexte <ctx>, et <val> une valeur de D, le facteur de certitude CF associé a la signification suivante :

CF = +1 : certitude que <val> est la valeur de (<ctx>,<val>),
CF = -1 : certitude que <val> n'est pas la valeur de (<ctx>,<val>),
CF = c (c>0) : degré de certitude que <val> est la valeur de (<ctx>,<val>),
CF = -c (c>0) : degré de certitude que <val> n'est pas la valeur de (<ctx>,<val>),
CF = 0 : on ne peut rien dire.

CF est défini comme la différence entre une mesure de croyance MB et une mesure de défiance MD, dont l'une des deux vaut 0 :

MB $\in$ [0,1] : degré de croyance en <val> comme valeur du paramètre,
MD $\in$ [0,1] : degré de doute en <val> comme valeur du paramètre,
MIN (MB,MD) = 0 : croire et douter sont mutuellement exclusif,
CF = MB-MD.

Cette représentation permet de mesurer l'incertitude sur une échelle continue [0,1] et non plus sur l'ensemble {vrai,faux}.

Les paramètres ne peuvent prendre leurs valeurs que sur un ensemble fini de valeurs

discrètes et précises. Par exemple, pour le paramètre âge, des faits/propositions comme "le malade a entre 14 et 16 ans" ou "la malade a environ 14 ans" ne peuvent être représentés.

Les règles manipulées par Mycin sont de la forme :

si <prémisse> **alors** <actions>

La partie <actions> peut comporter une ou plusieurs conclusions. A une conclusion est attaché un coefficient de certitude $CF_C \in [-1,+1]$ représentant un coefficient d'atténuation. Selon que CF_C est strictement positif ou négatif, la conclusion est favorisée ou défavorisée, autrement dit, une conclusion étant de la forme "la valeur du paramètre <p> du contexte <ctx> est égale à <val>" CF_C indique si la conclusion est en faveur ou en défaveur de la valeur <val>.

3.2.2.1.2. Etape d'inférence

a. Evaluation de la partie prémisse

La partie prémisse d'une règle a la structure suivante :

<prémisse> ≡ <condition> ET ..... ET <condition>
<condition-élémentaire>

<condition> ≡ <condition> OU ..... OU <condition>
<condition-élémentaire>

Pour déterminer si une règle est déclenchable ou non Mycin évalue sa partie prémisse. Cette évaluation rend un nombre compris entre 0 et 1 calculé en fonction de la structure de la partie prémisse.

. Cas d'une condition élémentaire isolée :

Il y a trois types de conditions élémentaires, on va se limiter dans ce paragraphe à l'étude du cas le plus simple (pour les autres cas voir [FAR85]) soit P(<contexte>,<paramètre>) où P est l'un des prédicats prédéfinis : CONNU, INCONNU, CERTAIN, INCERTAIN,

représentant les fonctions de test de la valeur d'un paramètre.

Pour évaluer ce type de condition on retient le plus grand des différents CF associés aux différentes valeurs possibles du paramètre, soit : CFmax. La valeur de la condition est alors calculée selon P :

CONNU : si CFmax > 0.2 alors 1 sinon 0
INCONNU : si CFmax <= 0.2 alors sinon 0
CERTAIN : si CFmax = 1alors 1 sinon 0
INCERTAIN : si CFmax < 1 alors 1 sinon 0.

. Cas de plusieurs conditions à conjuguer :

L'évaluation d'une conjonction ou d'une disjonction de conditions se fait de la manière suivante :

EVALUATION (<condition$_1$> ET ..... ET <condition$_n$>) =
MIN (EVALUATION (<condition$_1$>), ..., EVALUATION (<condition$_n$>))

EVALUATION (<condition$_1$> OU ..... OU <condition$_n$>) =
MAX (EVALUATION (<condition$_1$>), ..., EVALUATION (<condition$_n$>))

Une règle est déclenchable si le résultat de l'évaluation de la partie prémisse est supérieur à 0.2.

b. Evaluation de la partie conclusion

Lorsqu'une règle est déclenchable, l'évaluation d'une conclusion de la forme "la valeur du paramètre <p> du contexte <ctx> est égale à <val>, avec une certitude <CF_r>" amène à calculer un degré de certitude :

$CF = CF_r$ * EVALUATION (<prémisse>),

si $CF_r > 0$, alors CF > 0, et le déclenchement de la règle est en faveur de l'hypothèse :
VALEUR (<ctx>,<p>) = <val>;

si $CF_r < 0$, alors CF < 0, et le déclenchement de la règle conduit à douter de l'hypothèse :
VALEUR (<ctx>,<p>)=<val>.

c. Combinaison des conclusions de plusieurs régles

Si deux règles concluent sur une même hypothèse (<ctx>,<p>,<val>) avec deux facteurs de certitude CF1 et CF2, alors Mycin calcule un facteur de certitude cumulé de la façon suivante :

(1) $CF = CF1 + CF2 - CF1*CF2$ si $CF1*CF2 \geq 0$

(2) $CF = (CF1 + CF2) / (1 + \min(|CF1|,|CF2|))$ si $CF1*CF2 < 0$

Avec (1) et (2), on a $-1 \leq CF \leq 1$ et $|CF| \geq \max(|CF1|,|CF2|)$, ce qui exprime en présence de deux informations en faveur de la même hypothèse, Mycin renforce sa certitude, traduisant ainsi qu'arriver à la même conclusion par deux voies différentes est un argument de plus en faveur de celle-ci. Lorsque les informations proviennent de deux règles dont les conditions sont très dépendantes, le renforcement n'est pas toujours la bonne attitude, par exemple s'il existe la même règle deux fois dans la base, Mycin comptera toujours deux fois la même information.

3.2.2.1.3. Conclusion

- Mycin manipule des propositions incertaines, mais n'autorise pas d'imprécision sur leur contenu,
- les variables manipulées sont à domaines discrets seulement,
- la combinaison des règles n'est fondée sur aucune base mathématique. Cela risque d'entrainer des résultats absurdes après plusieurs inférences successives,
- l'utilisation de seuil pour l'évaluation de la partie prémisse d'une règle, est brutale. En effet, une condition est évaluée à 1 (resp. 0) dès que FC dépasse (resp. est inférieur) -aussi faiblement que ce soit- le seuil 0.2. Ainsi, une infime variation sur FC conduit à des conclusions opposées.

3.2.2.2. TOULMED

C'est un générateur de systèmes experts issu du développement du système expert DIABETO. Toulmed est capable de prendre en compte l'imprécision et l'incertitude des faits et des règles en utilisant le formalisme de la théorie des possibilités. Il a été développé en 1987 à l'université Paul Sabatier de TOULOUSE.

3.2.2.2.1. Représentation de la connaissance

Toulmed manipule des propositions et des règles. Les propositions sont de deux types : les propositions composées qui qualifient plusieurs variables et les propositions élémentaires qui en qualifient une seule. Pour représenter ses connaissances, Toulmed utilise la théorie des possibilités dont nous avons rappelé les principes de base au paragraphe 1.

a. Propositions élémentaires

"X est A (certitude c)" est la proposition la plus générale qui peut être à la fois imprécise (ou floue) et incertaine.

A représente les valeurs possibles pour la variable X et c est un nombre entre [0,1] qui représente le degré de certitude que "X est A" ; plus c est proche de 1 et plus on est certain que "X est A". Le cas extrême c=1 exprime que la proposition est certaine mais elle peut être imprécise.

En utilisant les concepts de la théorie des possibilités, Toulmed représente chaque proposition élémentaire par une distribution de possiblités π_X.

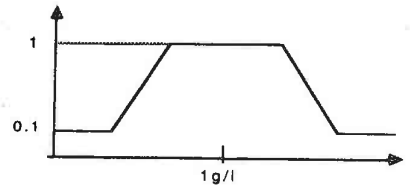
Dans le cas de la proposition P : "X est A (certitude c)", la distribution de possibilités associée est la suivante :

$$\forall u \in U,\ \pi_X(u) = \max(\mu_A(u), 1-c).$$

Exemple :

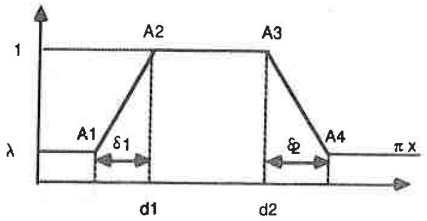
" la glycémie est très probablement d'environ 1 g/l"

Cette proposition est à la fois imprécise et incertaine, la distribution de possibilités associée est alors :



Le qualificatif "très probablement" a été traduit en degré de certitude de 0.9.

En conclusion, pour représenter ses conditions élémentaires, Toulmed utilise une distribution de possibilités de la forme générique suivante :



Cette représentation exprime :

- le niveau d'incertitude : $\lambda = 1-c$,
- les valeurs de U qui sont tout à fait possible : A2A3,
- les valeurs de U de possibilité minimale : A1A2,A3A4,
- les valeurs de U qui sont impossibles : $\forall u \in U$ tel que $u<d1-\delta1$ ou $u>d2+\delta2$.

En ce qui concerne les distributions de possibilités définies sur un domaine discret $\{v_1,...,v_n\}$, elles sont représentées par un n-uplets : $\{(v_1, \nu_1); ... ; (v_n, \nu_n)\}$, où ν_i est le degré de possibilité de la valeur v_i.

b. Propositions composées

Une proposition composée est une combinaison de propositions élémentaires. La combinaison peut être une conjonction ET ou une disjonction OU. Pour représenter une proposition composée, Toulmed utilise les distributions de possibilités associées à chacune des propositions élémentaires.

En cas de conjonction ET, la proposition "X_1 est A_1 ET X_2 est A_2" est représentée par :

$$\forall (u_1,u_2) \in U_1xU_2 , \ \pi_{X1,X2}(u_1,u_2) = \text{Min}(\pi_{X1}(u_1), \pi_{X2}(u_2)).$$

En cas de disjonction OU, la proposition "X_1 est A_1 OU X_2 est A_2" est définie par :

$$\forall (u_1,u_2) \in U_1xU_2 , \ \pi_{X1,X2}(u_1,u_2) = \text{Max}(\pi_{X1}(u_1), \pi_{X2}(u_2)).$$

c. Règles

Toulmed manipule trois types de règles :

- les règles à condition non floue et à conclusion quelconque, éventuellement incertaine, qui s'écrivent : " si X est A alors Y est B (certitude c) ",
- les règles à condition floue et à conclusion non floue, éventuellement incertaine, qui s'écrivent : " plus X est A, plus on est certain que Y est B (certitude c) ",
- les règles à condition floue et à conclusion floue, sans niveau d'incertitude, qui s'écrivent : " si X est A alors Y est B".

Pour représenter ces trois types de règles, Toulmed utilise la notion de distribution de possibilités conditionnelle [DUB-87]. Une distribution de possibilités conditionnelle $\pi_{Y/X}$ permet de représenter le lien de causalité entre deux variables X et Y définies respectivement sur les univers U et V. Un tel lien exprime des restrictions sur la valeur de Y quand on suppose que X vérifie certaines conditions.

$\pi_{Y/X}$ est définie de $VxU \rightarrow [0,1]$; telle que $\pi_{Y/X}(v,u)$ représente le degré de possibilité que Y ait la valeur v, sachant que X a la valeur u.

La définition de $\pi_{Y/X}$ dépend du type de règles :

- règles de type 1 : $\forall (v,u) \in VxU$, $\pi_{Y/X}(v,u) =$ si $u \in A$ alors $\mu_B(v)$ sinon 1,
- règles de type 2 : $\forall (v,u) \in VxU$, $\pi_{Y/X}(v,u) = Max(\mu_B(v), 1-\mu_A(u), 1-c)$,
- règles de type 3 : $\forall (v,u) \in VxU$, $\pi_{Y/X}(v,u) =$ si $\mu_B(v) \geq \mu_A(u)$ alors 1 sinon $\mu_B(v)$.

3.2.2.2.2. Etape d'inférence

Pour propager l'imprécision et l'incertitude des données au travers des règles, Toulmed utilise le Modus-Ponens généralisé introduit par Zadeh [ZAD-79].

Le Modus-Ponens généralisé proposé dans TOULMED correspond au schéma de raisonnement suivant :

X est A'		(certitude c_A attachée à la donnée)
si X est A alors	Y est B	(certitude c_B attachée à la règle)
	Y est B'	(certitude $c_{B'}$ attachée à la conclusion)

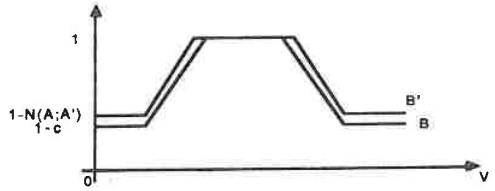
A, B, A' et B' sont des ensembles flous qui caractérisent uniquement l'aspect imprécis des propositions. En revanche, c_A, c_B et $c_{B'}$ sont des coefficients compris entre 0 et 1 qui traduisent l'aspect incertain.

A partir de ce dernier schéma de raisonnement, Toulmed propose une représentation de la proposition " Y est B' " pour chacun des types de règles cités ci-dessus :

- pour les règles de type 1 :

$$\forall v \in V, \mu_{B'}(v) = Max(1- N(A;A'), \mu_B(v)),$$

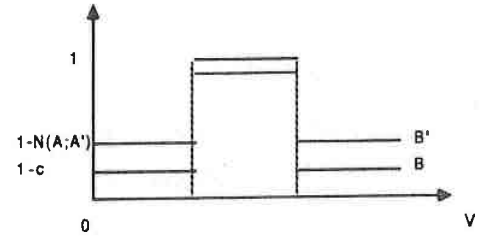
ainsi, B' s'obtient en ajoutant à B le niveau d'incertitude 1- N(A;A'),



plus N(A;A') est faible (c'est à dire A' n'est pas très compatible avec A), et plus le résultat est incertain. Le cas extrême N(A;A')=0 mène à un résultat complétement indéterminé B' = V.

- pour les règles de type 2 :

$$\forall v \in V, \mu_{B'}(v) = \text{si } v \in B \text{ alors 1 sinon Max } (1- N(A;A'),1-c),$$



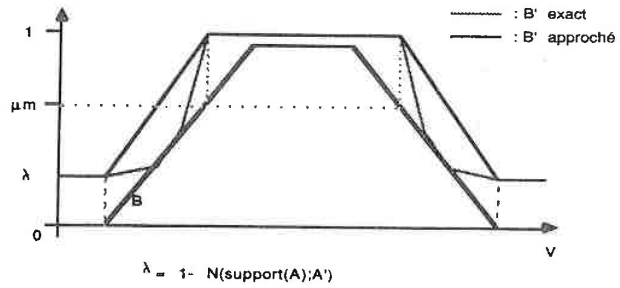
là également le niveau d'incertitude 1-N(A;A') est ajouté.

- pour les règles de type 3 , Toulmed se contente de donner une bonne approximation de la forme de B' en ne s'intéressant essentiellement qu'aux valeurs de B' qui sont tout à fait possibles (c'est à dire les valeurs du noyau (cf §1.1) de B') et aux valeurs de possibilité minimale. Ces deux ensembles de valeurs sont établis par :

+ $v \in$ noyau (B') $\Leftrightarrow \mu_B(v) \geq \mu_m = Inf_{u \in \text{ noyau } (A')} \mu_A(u)$

+ $\mu_{B'}$ est minimale sur V - support (B) (cf §1.1), et vaut : $\lambda = 1\text{-N(support (A);A')}$,

Entre ces deux ensembles de valeurs, $\mu_{B'}$ est linéaire par morceaux, mais Toulmed l'assimile à un segment de droite :



λ = 1- N(support(A);A')

Pour le reste, le raisonnement approché de Toulmed est un ensemble de techniques et de stratégies; par exemple :

- la prise en compte de l'importance relative des différentes conditions d'une règle,
- l'emploi d'une stratégie particulière pour combiner les informations partielles apportées par plusieurs règles,
- l'utilisation d'un mécanisme de retour-arrière permettant de remettre en cause les données qui sont à l'origine des conflits.

3.2.2.2.3. Conclusion

Toulmed est fondé sur la théorie des possibilités, ce qui lui permet de représenter à la fois l'imprécision et l'incertitude au niveau des faits comme les conditions et les conclusions des règles.

Il permet de manipuler des variables à domaines continus ou discrets, utilise le Modus-Ponens généralisé comme schéma de raisonnement, et donne la possibilité de pondérer les conditions de la partie prémisse.

3.2.2.3. Autres systèmes

a. PROTIS

Protis est un système expert médical fournissant des conseils thérapeutiques à des médecins généralistes. Il manipule des règles de déductions floues (FR) qui ont la forme suivante :

FR : <condition> → <décision> (e,r)

où e,r ∈ [0,1], appelés respectivement "degré de suggestion" et "degré de rejet" de la décision.

Pour représenter les faits et les conditions, Protis utilise le concept de distribution de possibilités. Les distributions de possibilités sont uniquement à domaine continu.

Lors de l'étape d'inférence, Protis évalue la compatibilité entre un fait F_i et une condition élémentaire C_i en calculant une mesure de possibilité Π_i et une mesure de nécessité N_i.

Si deux conditions C_i et C_j sont vérifiées par les faits avec les degrés Π_i, N_i et Π_j, N_j, Protis évalue les degrés de compatibilité de $C_i \wedge C_j$ et de $C_i \vee C_j$ de la façon suivante :

$C_i \wedge C_j$: $\Pi = \text{Min}(\Pi_i,\Pi_j)$, $N = \text{Min}(N_i,N_j)$

$C_i \vee C_j$: $\Pi = \text{Max}(\Pi_i,\Pi_j)$, $N = \text{Max}(N_i,N_j)$

En appliquant ces formules à tous les goupes C_i connectés par $\wedge$ ou $\vee$, Protis calcule les deux mesures globales Π et N caractérisant la compatibilité de la partie <condition> d'une règle FR avec les faits connus.

Protis combine alors les deux poids e et r avec les deux mesures Π et N, et calcule deux nombres :

$\alpha = \text{Max}(0,\Pi-(1-e))$: degré d'évocation de la <décision> par la <condition>,

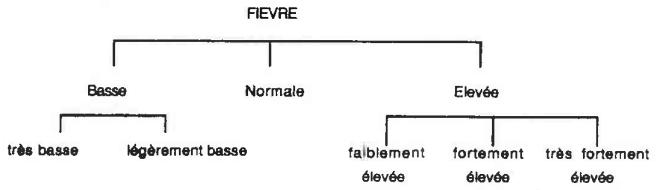
$\beta = \text{Min}(1,N+(1-r))$: degré de non rejet de la <décision> par la <condition>.

En conclusion, Protis est bien fondé mathématiquement à l'exception de certaines formes de combinaisons (par exemple $\Pi = \text{Min}\ \Pi_i$) qui manque de souplesse. Il ne permet de manipuler que des informations imprécises et non incertaines. Les distributions de possibilités associées aux informations imprécises sont uniquement à domaines continus.

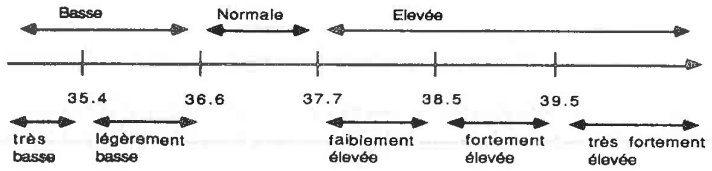
b. SPHINX

Sphinx est un système expert médical d'aide au diagnostic des ictères et au traitement du diabète. Il manipule des objets appelés *entités* et *attributs* , qui sont tout à fait comparable aux contextes et paramètres cliniques de Mycin.

A un couple (entité, attribut), Sphinx associe ce qu'il nomme une *variable sémantique* . Les valeurs possibles d'une telle variable sont organisées en arborescence ; par exemple les valeurs possibles de la variable Fièvre sont organisées de la façon suivante :



A chacune des valeurs correspond un intervalle sur une échelle continu :



On peut remarquer immédiatement la discontinuité de l'attribution de la valeur d'une telle variable. En effet, pour deux valeurs proches de part et d'autre d'un seuil, Sphinx attribuera deux valeurs symboliques différentes, et les raisonnements effectués à partir de là pourront conduire à deux résultats complétement différents.

Les règles manipulées par Sphinx, ont un sens tout à fait identique aux règles de Protis et elles ont la forme suivante :

règles d'évocation : <signe> —— évoque (e) ——> <contexte>

règles de rejet : <signe> —— rejette (r) ——> <contexte>

<signe> est une conjonction de conditions portant sur des variables sémantiques. Chacune de ces conditions est une proposition "X est I_i" , où X est une variable et I_i un intervalle de l'arborescence du domaine de X.

Pour calculer la compatibilité entre une condition I_j et un fait I_i, Sphinx utilise une solution empirique consistant à calculer un nombre t appelé *degré de conformité sémantique* de la façon suivante :

$t = 1$ si $I_i = I_j$,

$t = 0$ si I_i et I_j n'ont pas de prédécesseur commun dans l'arbre autre que la racine,

$t = 1 - (1 / (1+n)^2)$ si I_j est fils de I_i ,

$t = 1 - (1 / (1+n))$ sinon

où n est le niveau dans l'arbre (racine = 0) du prédecesseur commun à I_i et I_j .

Le nombre t a un sens similaire aux mesures Π ou N : plus t tend vers 1, plus la compatibilité de "X est I_i" avec "X est I_j" est totale.

On peut noter une disymétrie dans le calcul de t par rapport aux deux arguments I_i, I_j , ainsi que l'aspect pessimiste (t=0) du raisonnement lorsque les deux intervalles I_i et I_j sont au premier niveau de l'arbre et ayant une frontière en commun.

A partir du nombre t et des nombres e et r, Sphinx calcule un degré d'évocation α pour une règle d'évocation et un degré de rejet β pour une règle de rejet :

$$\alpha = \text{Max}\ (t - (1-e), 0),$$

$$\beta = \text{Min}\ (t + (1-r), 1).$$

En conclusion, Sphinx manipule des règles incertaines et des propositions imprécises (en utilisant des intervalles) mais non floues , par exemple "la température est d'environ 37" . Le calcul du degré de compatibilité d'un fait avec une condition est empirique et donne parfois des résultats non conformes à ceux qu'on obtiendrait en appliquant la théorie des possibilités.

3.2.3. Conclusion

Pour chacun des systèmes présentés ci-dessus nous avons souligné ses possibilités d'expression de l'imprécis et de l'incertain. Plus précisément :

- Mycin manipule des propositions qui peuvent être incertaines, imprécises mais non floues (l'imprécision sur la valeur d'une variable est désignée simplement par un sous ensemble ordinaire de valeurs),
- Toulmed manipule des propositions qui peuvent être imprécises, floues et / ou incertaines,
- Protis manipule des faits et des conditions de règles flous, mais qui ne peuvent être incertains, par contre les décisions sont incertaines,
- Sphinx manipule des propositions qui peuvent être imprécises mais non floues,quant à l'évocation des contextes elle est incertaine.

La prise en compte de l'imprécision, du flou et de l'incertitude est traduite , dans chacun des systèmes, par des concepts différents définis sur une échelle graduée : degrés de croyance et de défiance de Mycin, degrés d'évocation et de rejet de Protis et Sphinx, les mesures de possibilité et de nécessité de Toulmed.

Parmi tous les formalismes présentés dans ces systèmes, la théorie des possibilités nous paraît le cadre le plus intéressant pour représenter, et l'imprécision (et le flou), et l'incertitude, grâce aux concepts de distribution de possibilité et de nécessité. Avec cette théorie on peut vérifier que les différentes mesures de confiance (degrés de croyance et de défiance ...) dérivent toutes de l'une des deux mesures de possibilité et de nécessité [BUI-87b].

4. CONCLUSION

De cette étude sur l'état actuel des recherches en matière de systèmes de gestion d'informations et de connaissances nuancées nous tirons les conclusions suivantes :

- La théorie des possibilités est utilisée aussi bien dans des approches "bases de données" que dans des "systèmes experts".
- Parmi les différentes méthodes de représentation de l'incertain et de l'imprécis c'est la théorie des possibilités qui offre un cadre de raisonnement le mieux formalisé.
- Les solutions proposées en intelligence artificielle mettent l'accent sur le raisonnement approximatif (incertain ou imprécis) mais ne proposent pas de systèmes d'interrogation nuancée.
- Les systèmes d'interrogation nuancée issus du domaine des bases de données se limitent pour la plupart à un aspect de gestion d'informations nuancées.
- Dans toutes les solutions proposées, en intelligence artificielle comme en bases de données, le nombre de coefficients, de fonctions et de distributions à fournir est très élevé et aucun moyen n'est offert au concepteur d'une application pour le diminuer.

La solution proposée dans cette thèse utilise comme cadre formel la théorie des possibilités, elle se situe entre l'approche "bases de données" et l'approche "systèmes experts" et offre des moyens pour générer une part importante des coefficients et des fonctions nécessaires à la mise en oeuvre de la théorie.

Chapitre III CONNAISSANCES NECESSAIRES A LA GESTION DES INFORMATIONS NUANCEES

Chapitre III CONNAISSANCES NECESSAIRES A LA GESTION DES INFORMATIONS NUANCEES

Le problème de base qui se pose lors du développement d'un modèle ou d'une méthode est celui de la structuration et de la représentation des connaissances du domaine d'application choisi, que ce soit en Science naturelle, en Médecine ou en Archéologie ou en tout autre domaine où les connaissances sont riches et complexes.

Dans ce travail, le domaine auquel nous nous sommes intéressés, en priorité, est celui des sciences naturelles (Zoologie, Botanique, Mycologie, ...) car les connaissances y présentent les particularités suivantes :

- Il existe en général une bonne structuration du domaine se présentant sous la forme d'une classification hiérarchique par exemple en familles, genres et espèces, dans laquelle chaque classe est définie par des caractères macroscopiques ou microscopiques prenant leurs valeurs dans des domaines finis.
- La description des valeurs possibles prises par les caractères d'une classe est toujours très riche et très nuancée, nous entendons par là, qu'un caractère d'une espèce est rarement défini par une valeur précise mais au contraire par une liste de valeurs dans laquelle est associée à chaque valeur une information complémentaire que nous avons appelée "nuance" (cf chapitre 1).
- La classification comporte en général de nombreuses exceptions qui expliquent qu'il est presque toujours nécessaire de recourir à un spécialiste du domaine pour identifier un élément.

Le modèle que nous proposons est une tentative d'intégration de ces trois aspects.

Dans ce chapitre nous présentons tout d'abord une vue d'ensemble du contenu type d'une base de connaissances associée à un domaine des sciences naturelles. Ensuite, nous détaillons les connaissances spécifiques à notre approche : le thésaurus des termes, les fonctions caractéristiques et les règles. Enfin, nous donnons le modèle retenu qui utilise un formalisme "objet" [MAS-89].

1. CONTENU ET ORGANISATION DE LA BASE DE CONNAISSANCES

La figure n°1 est une représentation possible et partielle à l'aide du modèle Entité-Association du schéma conceptuel d'une base de connaissances multimédia associée à un domaine des sciences naturelles.

Cette base de connaissances contient les informations suivantes :

- la représentation de la classification (espèce, groupe d'espèces),
- la définition des caractères utilisés pour distinguer les espèces les unes des autres,
- les domaines de valeurs de ces caractères et leur définition en extension pour certains,
- un thésaurus entre les valeurs de certains domaines contenant des liens de généricité ou de voisinage,
- une description nuancée des espèces,
- un ensemble de règles se rapportant à un ou plusieurs éléments du schéma et exprimant soit une connaissance non modélisable, soit des stratégies à appliquer lors de l'identification de l'espèce à laquelle appartient un exemplaire observé,
- une bibliothèque de fonctions caractéristiques exprimant la sémantique attachée par les spécialistes à une information nuancée,
- une base d'images comprenant des photos représentant les espèces et des schémas explicatifs des caractères et de leurs valeurs,
- une base de textes en langage naturel décrivant les espèces.

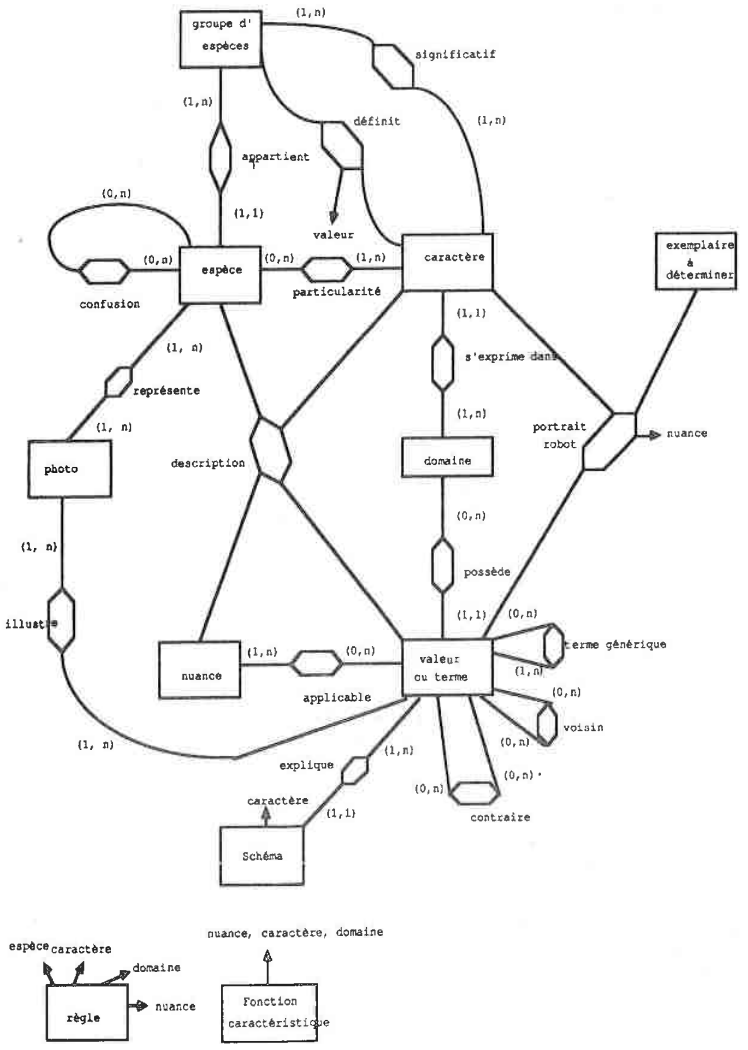


Figure1 Schéma conceptuel d'une base de connaissances multimédia en sciences naturelles

LEGENDE :


: entité


: association entre entités

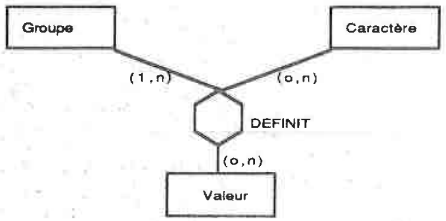
: indique d'autres entités faisant partie de l'association. Nous avons utilisé cette notation pour éviter d'encombrer le schéma.

Commentaire :

Nous avons utilisé dans ce premier schéma le modèle Entité-Association (E-A) tel qu'il est présenté dans la méthode MERISE [TAR-83]. Nous supposons le lecteur familier de ce modèle et ne justifierons donc que les associations non binaires. Les cardinaux de ces associations n'ont pas été portés sur le schéma général afin de ne pas l'alourdir; nous les donnons ci-dessous avec les explications concernant la sémantique de ces associations.

Association DEFINIT :

Elle donne pour chaque groupe d'espèces les caractères et les valeurs associées qui sont spécifiques du groupe.



- Pour un groupe il y a plusieurs caractères définissant ce groupe et pour chacun de ces caractères une ou plusieurs valeurs possibles.
- Un caractère peut n'être spécifique à aucun groupe ou au contraire à plusieurs. Lorsqu'il spécifique il l'est avec une ou plusieurs valeurs. De même que pour les valeurs.

Exemple emprunté à la mycologie :

Le fait que les amanites aient en commun :

- des lamelles blanches,

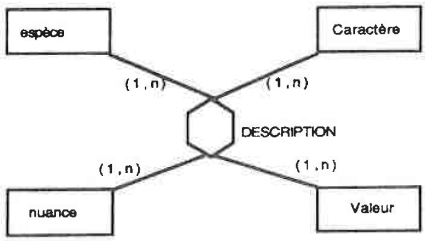
- des lamelles libres,

entraîne que le groupe des amanites sera représenté par deux triplets :

- (amanite, couleur des lamelles, blanches)
- (amanite, allure des lamelles, libres)

Association DESCRIPTION :

Elle définit pour chaque espèce les valeurs nuancées que peuvent prendre leurs caractères.



Exemple emprunté à la mycologie :

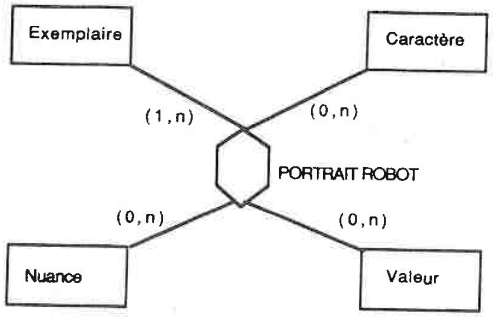
Le fait que l'amanite phalloïde ait un chapeau qui est vert généralement, jaunâtre parfois et blanc rarement sera représenté par les trois quadruplets suivants :

- (amanite phalloïde, couleur chapeau, verte, généralement)
- (amanite phalloïde, couleur chapeau, jaunâtre, parfois)
- (amanite phalloïde, couleur chapeau, blanche, rarement)

Comme pour l'association DEFINIT, tous les cardinaux maximaux sont de type n. En effet, aucune restriction n'existe sur l'apparition des valeurs et des nuances dans la description des espèces.

Association PORTRAIT ROBOT :

Elle est l'équivalent de l'association DESCRIPTION pour la représentation des valeurs données à certains caractères par l'utilisateur.



Dans cette association les cardinaux minimaux sont égaux à 0 car il est évident que tous les caractères, nuances ou termes ne sont pas utilisés par les utilisateurs.

Comme nous l'avons indiqué au début du paragraphe *ce schéma ne constitue qu'une représentation possible et limitée des connaissances* à mettre en place dans un système d'aide à l'identification nuancée.

L'utilisation d'un modèle E-A étendu aux abstractions de généralisation-spécialisation et d'agrégation [SMI-77], mettrait mieux en évidence les objets principaux :

- les descriptions d'espèces, de groupe d'espèces ou de portrait robot,
- les caractères,
- les nuances,
- les domaines,
- le thésaurus représenté par l'ensemble des entités et des associations TERME, TERME GENERIQUE, VOISIN et CONTRAIRE

Cette approche a l'avantage de ne pas morceler la description des espèces en de nombreux nuplets indépendants. C'est cette approche que nous avons retenue pour mettre en évidence les différents objets du modèle que nous décrivons dans ce chapitre.

Lorsque cette base de connaissances est utilisée pour identifier l'espèce à laquelle appartient un élément réel observé, il lui est associé une base de faits contenant le portrait robot de cet élément tel que l'a observé l'utilisateur. Ce portrait robot est exprimé, lui aussi, le plus souvent sous une forme nuancée puisque l'observation est souvent difficile dans la réalité.

Nous précisons dans ce chapitre les composants de cette base de connaissances, spécifiques à notre approche :

- **le thésaurus des termes** représentés par l'entité *terme* d'une part et par les associations *voisin, terme générique* et *contraire* d'autre part,
- **les fonctions caractéristiques** associées aux informations nuancées,
- **la base de règles ,**
- **le modèle de description des espèces et du portrait robot.**

2. THESAURUS DES TERMES

La notion de thésaurus est empruntée au domaine de la documentation [SLY-86] où un thésaurus des termes est un ensemble structuré de concepts destinés à représenter de manière univoque le contenu des documents et des questions, et à assister l'utilisateur dans l'indexation des documents et des questions; les concepts sont extraits d'une liste finie, établie à priori mais qui quelquefois peut être progressivement complété; seuls les termes figurants dans cette liste peuvent être utilisés pour indexer les documents et les questions; l'assistance à l'utilisateur est apportée par la structure sémantique du thésaurus, c'est à dire par les différents types de relations pouvant exister entre les termes : relations de synonymie, de généricité, d'association.

Exemple :

Par exemple le terme "SERVICE DE DOCUMENTATION " pourra être décrit dans un thésaurus :

SERVICE DE DOCUMENTATION
équivalent : système documentaire
générique : système d'information
spécifique : système de signalement
associé : banque de données bibliographiques

2.1. Pourquoi un thésaurus

Dans les systèmes documentaires, le thésaurus est essentiellement utilisé pour :

- **indexer les documents** : le documentaliste n'utilise que les termes appelés descripteurs pour décrire un document. Ainsi, le documentaliste doit traduire tous les concepts trouvés dans les documents par des descripteurs du thésaurus,
- **faciliter la formulation des questions** : grâce à sa structure sémantique, le thésaurus permet :
  . de trouver plus aisément les descripteurs à utiliser à partir de l'expression en clair des concepts de la question,
  . d'étendre la recherche à d'autres concepts, plus spécifiques, plus génériques, ou simplement associés à ceux de la question.

Le thésaurus que nous utilisons présente les particularités suivantes :

- nous n'avons pas un thésaurus unique mais un ensemble de petits thésaurus appelés micro-thésaurus. Un micro-thésaurus est associé à un domaine. Un micro-thésaurus comprend l'ensemble des termes du domaine et l'ensemble des associations existant entre ces termes (comme par exemple les associations *voisin* et *regroupement* de la figure 1).
- souvent les liens entre les termes du thésaurus sont quantifiés pour pouvoir générer automatiquement les fonctions caractéristiques (ou distributions de possibilité) associées aux termes du thésaurus (cf §3.2.).

Dans notre approche, nous faisons appel au thésaurus lors des deux étapes suivantes :

- à la transformation de la requête utilisateur. Au cours de cette étape nous remplaçons les termes employés par l'utilisateur et appartenant au thésaurus par des termes plus spécifiques ou plus génériques. Le but de cette transformation est d'étendre ou de restreindre le champ de la recherche aux autres termes proches du terme employé par l'utilisateur,
- à la transformation d'une donnée de la base des connaissances. Nous procédons à cette transformation de la même façon que pour la requête utilisateur, afin d'augmenter le champ d'intersection entre la requête et la donnée.

Nous reviendrons plus en détail sur la réalisation de ses transformations en chapitre 4.

2.2. Contenu du thésaurus

Le thésaurus que nous utilisons est composé d'un ensemble de micro-thésaurus représentant les différents domaines. Par exemple : *Lieu de récolte*, *Couleur* , ...

Un micro-thésaurus est l'ensemble de termes que l'on rencontre dans les descriptions des espèces pour les caractères prenant leurs valeurs dans le domaine. Les termes sont liés entre eux par des relations ou liens sémantiques.

Dans notre approche, nous avons retenu pour l'instant trois types de liens sémantiques qui nous sont utiles :

- ***voisin***,
- ***contraire***,
- ***générique-spécifique***.

. Lien *voisin* (noté ~)

Une relation de voisinage entre deux termes exprime que leurs significations sont très proches l'une de l'autre et qu'il sera possible de les remplacer l'un par l'autre aussi bien dans la description des espèces que dans l'expression du portrait robot.

Pour exprimer la proximité sémantique de deux termes on introduit dans la relation "voisin" un coefficient compris entre 0.5 et 1.

Exemple

Dans le domaine "lieu de récolte" associé au caractère "écologie" pour les champignons on trouve :

voisin (prés, paturages, 0. 9)

noté également : prés $\sim_{0.9}$ paturages

Note:

L'expérience a montré que la fermeture transitive du lien *voisin* détériore les performances et qu'il fallait se limiter au voisinage "direct", c'est à dire le lien de voisinage n'est utilisé qu'à un seul niveau. Par exemple : soient les termes t1, t2, t3, t 4 et t5 tel que :

$t1 \sim t2$, $t2 \sim t3$, $t1 \sim t4$ et $t4 \sim t5$

Les termes considérés comme voisins de t1 sont alors seulement t2 et t4.

. Lien *contraire* (noté #)

La relation ***contraire*** traduit le fait qu'un terme est sémantiquement opposé à un autre et qu'il est donc impossible de les substituer l'un à l'autre.

Nous n'avons pas jugé nécessaire d'attacher un coefficient à ce type de relation. On aurait pu donner un coefficient compris entre 0 et 0.5, mais nous n'avons pas rencontré de situation où cela se justifiait.

Exemple : Dans le même domaine que précédemment on trouve :
contraire (sol:sec, sol:humide)
noté : sol:sec # sol:humide.

. Lien *générique-spécifique*

Cette relation correspond à la relation classique existant dans tous les thésaurus des systèmes documentaires. Elle permet de regrouper sous un terme dit "générique" un ensemble de termes qui expriment une notion voisine mais de manière plus précise : les termes spécifiques.

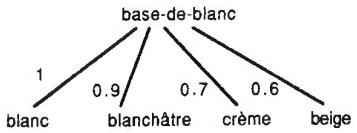
De même que dans les relations de type ***voisin***, il est possible dans certains cas d'affecter aux relations des coefficients compris entre 0 et 1 définissant la spécificité ou le degré de ressemblance ou de voisinage d'un terme vis à vis de son générique.

Exemple :

Le domaine "couleur" associé au caractère "couleur du chapeau" des champignons comprend un ensemble de termes génériques représentant des bases de couleurs : base-de-blanc, base-de-vert, base-de-jaune, etc ... A chaque terme générique est associé une liste de couleurs plus précises, par exemple pour base-de-blanc on trouve :

gén-spé (base-de-blanc, blanc, 1)
gén-spé (base-de-blanc, blanchâtre, 0. 9)
gén-spé (base-de-blanc, crème, 0. 7)
gén-spé (base-de-blanc, beige, 0. 6)

que l'on représente également de façon graphique comme ci-dessous

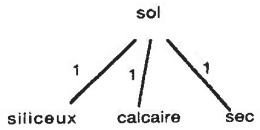


Le coefficient exprime une sorte de distance entre le terme générique et le terme spécifique. *Beige* est plus "éloigné" de la notion générale de couleur blanche que crème ou blanchâtre, c'est ce qu'expriment les coefficients 0.6 , 0.7 et 0.9 .

Notons que l'ensemble des spécifiques d'un même générique peuvent être :

- **liés sémantiquement** : on peut les remplacer les uns par les autres car ils sont proches sémantiquement comme par exemple les couleurs Blanc et Blanchâtre.
- **non-liés sémantiquement** : ils ne peuvent pas être substitués les uns par les autres, sauf s'il existe entre eux une relation de type ***voisin***.

Exemple :



Dans cette portion de réseau, sol:siliceux, sol:calcaire et sol:sec sont **non-liés sémantiquement** et ne pourront donc pas être substitués l'un à l'autre.

Dans certains cas, comme pour les formes ou les couleurs, les termes spécifiques d'un même générique peuvent être **liés sémantiquement** par une relation d'ordre entre les valeurs spécifiques d'un même générique , par exemple les couleurs peuvent être ordonnées par la relation "plus claire que" .

3. FONCTIONS CARACTERISTIQUES

Dans l'approche que nous avons adoptée, l'utilisateur a la possibilité d'employer des informations nuancées dans la description du portrait robot -correspondant à la requête- et dans les descriptions des espèces -correspondant à la base de données-.

Ayant choisi comme cadre formel pour la gestion des informations nuancées la théorie des possibilités, nous sommes amenés à considérer toute information nuancée comme un ensemble flou. Un ensemble flou étant défini par une fonction caractéristique, comme nous l'avons vu au paragraphe 1 du chapitre II, nous avons défini un mode de calcul des fonctions caractéristiques propre aux informations nuancées que nous avons retenues.

Rappelons qu'une information nuancée est pour nous un couple (<valeur> <LN>) où LN est une liste de nuances et où la valeur peut être elle même entachée d'imprécision.

Pour obtenir la fonction caractéristique associée à une information de ce type, nous introduisons successivement :

- les fonctions caractéristiques associées aux nuances,
- les fonctions caractéristiques associées aux valeurs imprécises ou vagues,
- les fonctions caractéristiques associées aux couples (<valeur> <LN>),
- les fonctions caractéristiques associées aux listes de couples (<valeur> <LN>).

Avant de définir ces différents niveaux de fonctions caractéristiques, nous introduisons une classification des informations nuancées inspirée de Hamon [HAM-87].

3.1. Classification des informations nuancées

La classification des informations nuancées repose sur les ensembles intervenant dans les définitions des fonctions caractéristiques associées. Dans cette classification nous distinguons les deux classes suivantes :

- Les transformations nuancées,
- Les modifications nuancées.

3.1.1. Transformations nuancées

On regroupe dans cette classe les informations nuancées auxquelles sont associées des fonctions caractéristiques μ qui, à partir d'un ensemble quelconque U (ou d'un produit cartésien d'ensembles), donnent le coefficient (entre 0 et 1) de ressemblance des différents

éléments u de U avec l'information nuancée :

$$U \rightarrow [0,1]$$

$$u \rightarrow \mu(u)$$

Les transformations nuancées servent à exprimer les deux classes d'informations nuancées suivantes :

- Les termes imprécis, flous ou vagues, tels que :
  . *grand* en fonction de la taille (T) d'une personne :

$$T \rightarrow [0,1]$$

$$t \rightarrow \mu_{grand}(t)$$

  . *jeune* en fonction de l'âge (A) d'une personne :

$$A \rightarrow [0,1]$$

$$a \rightarrow \mu_{jeune}(a)$$

Ces termes imprécis sont utilisés pour représenter un ensemble flou particulier dans un domaine donné.

- Les relations nuancées, telles que :
  . *beaucoup plus grand que* (*bpgq*) relativement à la taille (T) :

$$T \times T \rightarrow [0,1]$$

$$(t_1,t_2) \rightarrow \mu_{bpgq}(t_1,t_2)$$

  . *semblable* relativement à la taille (T) :

$$T \times T \rightarrow [0,1]$$

$$(t_1,t_2) \rightarrow \mu_{semblable}(t_1,t_2)$$

La classe des relations nuancées regroupe les nuances qui traduisent une condition relative d'un terme par rapport à un autre.

3.1.2. Modifications nuancées

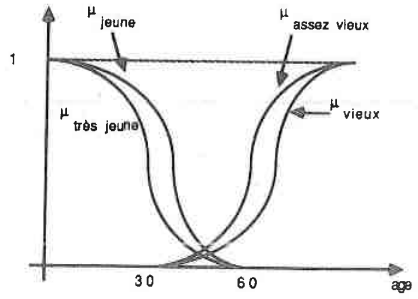
Ce sont les nuances qui "déforment" un coefficient de ressemblance. L'ensemble de départ de la fonction caractéristique associée est l'intervalle [0,1] de même que l'ensemble d'arrivée :

$$[0,1] \rightarrow [0,1]$$
$$x \rightarrow \mu(x)$$

Cette classe recouvre en réalité deux types de nuance :

- Les modificateurs nuancés, tels que : *très, peu, extrêmement,* qui permettent de représenter de nouveaux ensembles flous dérivés de l'ensemble flou initial.

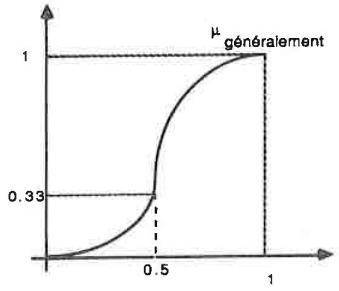
Par exemple, soient les expressions suivantes : (*jeune* et *très jeune*) d'une part et (*vieux* et *assez vieux*) d'autre part. Les termes *jeune* et *vieux* sont des transformations nuancées qui à partir de l'âge d'une personne fournissent un coefficient de ressemblance. Les informations nuancées *très jeune* et *assez vieux* se décomposent en une application des modificateurs *très* et *assez* sur les coefficients de ressemblance obtenus précédement.



On peut remarquer sur cette figure que les modificateurs nuancés *très* et *assez* ont dérivé de nouvelles fonctions à partir des fonctions caractéristiques initiales attachées aux termes *jeune* et *vieux* .

- Les quantificateurs nuancés , tels que : *la plupart , généralement , presque tous , rarement* , expriment le cardinal d'ensembles, désignent la "normalité", c'est-à-dire que l'ensemble considéré est le complémentaire d'un ensemble d'exceptions.

Exemple :
soit le quantificateur nuancé *généralement* , on peut définir la fonction caractéristique associée de la façon suivante :



Le quantificateur nuancé *généralement* définit un ensemble flou de proportions : plus grande est la proportion, mieux le terme *généralement* s'applique , et donc plus grand est le coefficient de ressemblance. Celui-ci, pour la proportion 0.5 de l'ensemble *généralement* , serait par exemple 0.33 .

3.2. Calcul des fonctions caractéristiques

Dans le modèle que nous présentons, nous considérons les informations nuancées comme des ensembles flous et par conséquent, nous les représentons par des fonctions caractéristiques.

Une fonction caractéristique est associée :
- soit à une valeur imprécise,
  Exemple : *grand , jeune , base-de-blanc , jaunâtre , ...*
- soit à une nuance,

Exemple : *très* , *environ* , *peu* , ...

- soit à un couple (valeur , nuance),

Exemple : (10 , *environ*) , (*blanchâtre* , *généralement*)

- soit à une liste de couples (valeur , nuance),

Exemple : ((*blanc* , *souvent*) ; (*jaune* , *parfois*))

Nous allons présenter dans la suite de ce paragraphe comment une fonction caractéristique est calculée pour chacune des situations ci-dessus.

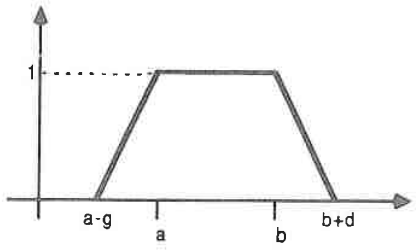
3.2.1. Fonction caractéristique associée à une valeur imprécise

Le support de définition de la fonction caractéristique associée à une valeur imprécise peut être un domaine continu (par exemple : le domaine des âges des personnes) ou un ensemble discret de valeurs (par exemple : l'ensemble des couleurs).

Nous allons étudier comment nous définissons la fonction caractéristique associée à une valeur imprécise en fonction de son domaine.

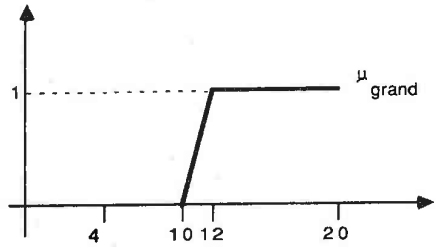
3.2.1.1. Cas du domaine continu

Une valeur imprécise appartenant à un domaine continu est définie par une fonction caractéristique qui est elle même spécifiée à l'aide d'une fonction trapézoïdale [CAY-82]. Elle est en général donnée sous la forme d'un quadruplet (a b g d) tel que :



Exemple :

Soit la valeur imprécise *grand* attachée au caractère "taille du chapeau" de domaine [4,20]. La fonction caractéristique μ_{grand} est représentée par :



Toutes ces fonctions caractéristiques sont définies par l'expert en fonction du caractère et du domaine ; par exemple la valeur imprécise *grand* attachée au caractère "diamètre des pores" de domaine [1,4] n'aura pas la même définition que celle attachée au caractère "taille du chapeau". Les fonctions ainsi définies sont stockées dans une bibliothèque de fonctions prédéfinies.

3.2.1.2. Cas d'un domaine discret

Une fonction de ce type est définie à l'aide d'un ensemble flou qui est lui même spécifié par une liste de couples (valeur , coefficient) où le coefficient (entre 0 et 1) mesure le degré d'appartenance de la valeur à cet ensemble flou.

Exemple :

Soit la valeur vague *base-de-blanc* associée au caractère "couleur du chapeau", la fonction caractéristique $\mu_{base\text{-}de\text{-}blanc}$ est définie par :

((base-de-blanc 1) (blanc 1) (blanchâtre 0.9) (crème 0.7) .....)

Une telle fonction est soit générée à partir du thésaurus lorsque le domaine discret auquel appartient la valeur imprécise y est représentée, soit définie par l'expert puis rangée dans la bibliothèque lorsque le domaine discret ne se prête pas à une représentation par thésaurus mais à une représentation par une structure d'ensemble.

Génération des fonctions caractéristiques à partir du thésaurus :

Le thésaurus contient à la fois des termes imprécis et des termes précis utilisés pour décrire les espèces et le portrait robot. Nous proposons dans ce paragraphe une méthode pour générer la fonction caractéristique associée à un terme du thésaurus.

Pour définir la fonction caractéristique μ_t associée au terme t, nous avons établi des règles de génération qui tiennent compte du type de la relation liant t aux autres termes du thésaurus.

Pour cela, notons :

- TG(t) : l'ensemble des termes génériques directs du terme t.
- TS(t) : l'ensemble des termes spécifiques directs du terme t.
- TV(t) : l'ensemble des termes tv liés au terme t par la relation "VOISIN" .
- TF(t) : l'ensemble des termes tf ayant le même terme générique direct que t.

et

- $E_{TG} = \{(tg, coef(t,tg)) / tg \in TG(t) \}$ où coef est le coefficient attaché à l'arc liant t à tg
- $E_{TS} = \{(ts, coef(t,ts)) / ts \in TS(t) \}$ où coef est le coefficient attaché à l'arc liant t à ts
- $E_{TV} = \{(tv, coef(t,tv)) / tv \in TV(t) \}$ où coef est le coefficient attaché à l'arc liant t à tv
- $E_{TF} = \{(tf, coef(t,tf)) / tf \in TF(t) \}$ où coef(t,tf)= Dist (t,tf)

où Dist est une fonction calculant la distance sémantique entre deux termes spécifiques d'un même terme générique. *Voici un exemple de calcul de distance que nous avons établie et que nous modulons en fonction de nos expérimentations* :

(1) $Dist = Max(O, Min_{tg \in TG(t) \cap TG(tf)} (\mu_{tg}(tf) - 1/2 |\mu_{tg}(t) - \mu_{tg}(tf)|)$,

il est évident que $E_{TF} = \emptyset$ lorsque les termes sont non liés sémantiquement .

Remarques:

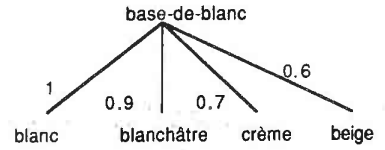
1- On pourrait établir au niveau du thésaurus un lien supplémentaire (une relation) entre le terme t et chaque terme tf (du même générique) qui permettrait alors d'obtenir directement le coefficient coef(t,tf) attaché à la relation liant t et tf. Dans un premier temps, nous retenons la solution du calcul de coef(t,tf) par la formule donnée (1) ci-dessus.

2- Il existe des thésaurus où les termes pourraient être ordonnés entre eux par une relation d'ordre. Par exemple, les couleurs pourraient être ordonnées par la relation "*plus claire que*". Une telle relation d'ordre permettrait alors de calculer les coefficients coef(t,tf) en décrémentant, par exemple, le coefficient coef(t,tg) de la distance séparant t à tf multipliée par un seuil préétabli.

La fonction caractéristique μ_t est représentée par une liste de couples (terme coef) donnée par l'union des ensembles E_{TG}, E_{TS}, E_{TV} et E_{TF} définis ci-dessus. μ_t s'écrit alors :

$$\mu_t = \{(t,1)\} \cup E_{TG} \cup E_{TS} \cup E_{TV} \cup E_{TF}$$

Exemple : Soit le micro-thésaurus suivant :

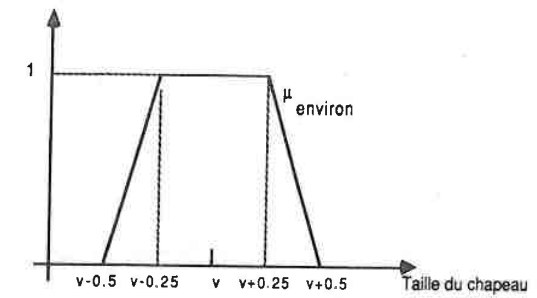


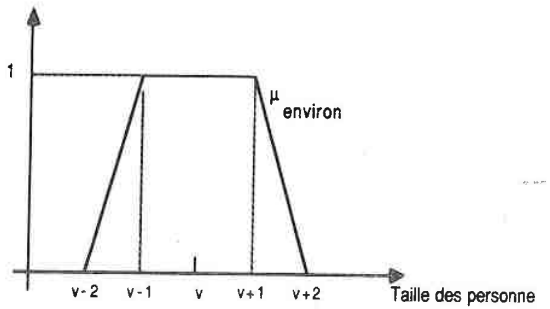
La fonction caractéristique associée au terme "blanc" est :

μ_{blanc} = ((blanc 1) (base-de-blanc 1) (blanchâtre 0.85) (crème 0.55) (beige 0.4))

3.2.2. Fonction caractéristique associée à une nuance

A chaque nuance n utilisée, est associée une fonction caractéristique μ_n donnée par l'expert. La définition de la fonction caractéristique est fonction du domaine et du caractère auxquels elle s'applique. Par exemple, les fonctions caractéristiques associées à la nuance *environ* pour les caractères *Taille du chapeau d'un champignon* et *Taille d'une personne* n'auront pas les mêmes définitions car les domaines de définition sont différents :





v étant une valeur quelconque du domaine du caractère choisi.

Dans notre solution les fonctions caractéristiques ainsi définies sont paramétrées par les valeurs du domaine auquel elles s'appliquent ce qui n'est pas le cas dans certains travaux [BOS-87] [TES-84] qui définissent, pour une même nuance, autant de fonctions caractéristiques différentes que de valeurs différentes.

Note :

Dans le système que nous proposons, l'expert a la possibilité de définir ses propres fonctions caractéristiques pour des nouveaux termes ou nuances dont il aurait besoin et que le système ne connaitrait pas encore.

3.2.3. Fonction caractéristique associée à un couple (valeur,nuance)

La définition de la fonction caractéristique $\mu_{<v,n>}$ associée au couple (valeur , nuance) (v,n) dépend du type de la nuance n (suivant que c'est une transformation ou une modification) et du type de la valeur (suivant qu'elle est précise ou imprécise).

- Cas où la nuance est une transformation :

La définition de la fonction caractéristique $\mu_{<v,n>}$ associée au couple (valeur, nuance) (v,n) est différente suivant que la valeur est précise ou imprécise.

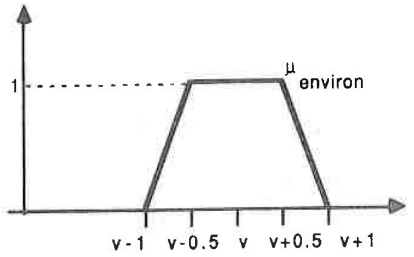
En effet, lorsque la valeur est précise, la fonction μ_{nv} est de la forme :

$$(1) \qquad \mu_{<v,n>}(v') = \mu_n(v,v') \qquad v' \in Dc$$

où Dc est le domaine du caractère c auquel est affecté le couple (v,n).

Exemple :

soit le couple (5 , environ) attaché au caractère "taille du chapeau", et la fonction $\mu_{environ}(v,*)$ associée à une valeur v et représentée par le schéma ci-dessous :



La fonction caractéristique $\mu_{environ-5}$ associée au couple est: $\mu_{environ-5}(v') = \mu_{environ}(5,v')$.

En revanche, lorsque la valeur est imprécise , soit μ_v sa fonction caractéristique, la fonction μ_{nv} doit être définie par une combinaison des fonctions μ_n et μ_v et nous proposons de la calculer par la formule empirique suivante :

(2) $\mu_{<v,n>}(v') = \sup_{v'' \in Dc} \min(\mu_n(v',v''), \mu_v(v'))$ $v' \in Dc$

- Cas où la nuance est une modification :

D'après la classification des nuances présentée au paragraphe §3.1., la fonction caractéristique μ_v donne le coefficient de ressemblance entre toute valeur v' du domaine Dc et la valeur imprécise v, quant à la nuance n associée à la valeur v et représentée par sa fonction caractéristique μ_n, elle doit agir sur le coefficient de ressemblance entre toute valeur v' et v soit en l'atténuant soit en le renforcant.

Ainsi, la fonction caractéristique $\mu_{<v,n>}$ doit être définie par une composition des fonctions μ_n et μ_v et nous proposons de l'établir par la formule suivante :

$$\mu_{<v,n>}(v') = \mu_n(\mu_v(v')) \qquad v' \in Dc$$

Cette formule généralise les deux cas où la valeur est précise ou non (cf exemple §3.1.2.).

3.2.4. Fonction caractéristique associée à une liste de couples (valeur , nuance)

Souvent les caractères sont décrits par des listes de couples (valeur , nuance) ; en général données sous la forme :

$Lv = (C_1, C_2, \ldots, C_p)$ où $C_i = (Valeur_i, Nuance_i)$ i=1, p

Exemple : Soit le caractère "couleur du chapeau" , une description de ce caractère pour une espèce peut être : (blanc , parfois) (vert , généralement)

La fonction caractéristique associée à une liste Lv est une combinaison des fonctions caractéristiques μ_i associées aux couples C_i (cf §3.2.3.) . En théorie, cette combinaison des μ_i utilise les opérateurs "Max" et "Min" suivant que les couples C_i sont connectés par un "OU" ou par un "ET", par conséquent μ_{Lv} est donnée par :

- cas d'un connecteur OU $\mu_{Lv}(v) = \mathbf{Max}(\mu_i(v))$ i = 1,p
- cas d'un connecteur ET $\mu_{Lv}(v) = \mathbf{Min}(\mu_i(v))$ i = 1,p

Remarque :

Il existe cependant des cas où le type de connexion (ET , OU) peut être déduit grâce à

des heuristiques (ou règles) tenant compte du caractère du type de domaine (discret ou continu) et des valeurs; par exemple les valeurs d'un caractère défini sur un domaine continu sont forcément liées par un "OU" (exclusif) de même les termes du thésaurus liés par une relation *contraire* sont obligatoirement connectés par un "OU" exclusif.

4. BASE DE REGLES

Dans notre approche, la base de règles exprime toutes sortes d'information qu'il n'est pas possible ou souhaitable de modéliser suivant un modèle de données classique comme le modèle entité association.

Parmi ces règles nous en distinguons deux types :

- des règles générales d'utilisation du thésaurus pour éviter par exemple d'arriver sur des incohérences dans l'utilisation des liens *contraires* :

Exemple :

Soient t1 et t2 deux termes contraires t1 # t2 , alors tout terme t' associé à t1 est contraire de tout terme t" associé à t2.

Voici un exemple (extrait du thésaurus des *Odeurs)* de résultat de cette règle :

SI *Odeur:agréable # Odeur:désagréable*
Odeur:agréable ~ Odeur:bonne
Odeur:désagréable ~ Odeur:mauvaise
ALORS *Odeur:bonne # Odeur:mauvaise*

- des règles attachées aux caractères ou aux espèces :

Exemples : de règles empruntées à la mycologie

. si l'appareil sporifère est du type aiguillons alors il ne peut y avoir de volve ,

. si l'appareil sporifère est du type lamelle et si la couleur du chapeau est autre que violette alors la description doit comporter une information sur la volve et sur l'anneau .

- Toutes les espèces de la famille des amanites ont une volve se présentant de différentes manières.

Ces règles permettent de déduire des informations à partir d'autres ou bien de vérifier la cohérence des descriptions - ce sont en quelque sorte des contraintes d'intégrité

statiques -,

- des règles de stratégie sur lesquelles repose le processus d'identification. Elles sont utilisées principalement pour :

. extraire des sous-ensembles de données d'après des caractères discriminants et fiables. Par exemple, le caractère "allure des lames" est l'un des caractères les plus stables chez certaines familles de champignons,

. discriminer entre les données résultant d'une phase d'identification lorsque le nombre de celles-ci dépasse un certain seuil. La discrimination entre les données peut être, par exemple, une série de questions sur les caractères discriminants non renseignés par l'utilisateur ,

. effectuer des retours-arrière en cas d'échec du processus d'identification pour éventuellement écarter des caractères qui ont influé sur l'élimination de certaines données proches du portrait-robot décrit par l'utilisateur. Par exemple :

si pour une espèce, l'anneau est "fugace" alors ne pas tenir compte de l'observation faite sur ce caractère par un utilisateur s'il dit ne pas en avoir observé .

Ces règles s'apparentent à des contraintes d'intégrité dynamiques.

5. MODELE DE DESCRIPTION DES DONNEES ET DU PORTRAIT-ROBOT

Elaborer un modèle de représentation ou de description de connaissances consiste à trouver des structures informatiques qui en permettent un stokage et une utilisation efficaces par des mécanismes d'exploitation.

Nous avons le choix entre un modèle issu du monde des Bases de Données (BD) ou un modèle issu de celui de l'Intelligence Artificielle (IA). En fait, nous avons retenu un modèle de type OBJET qui, à l'origine, était plutôt utilisé en IA mais qui, maintenant, est de plus en plus utilisé en BD et sur lequel reposeront les futurs SGBDs [BAN-88].

Pour justifier ce choix, nous faisons un très rapide tour d'horizon des modèles de chacun des domaines BD et IA.

5.1. Modèles de description existants

De nombreux travaux ont été consacrés à la modélisation des données et des connaissances, que ce soit en IA ou en BD. Nous nous contenterons ici de citer des ouvrages de synthèse auxquels le lecteur peut se reporter : [LAU-82] [CAL-83] [PIN-81] pour l'intelligence artificielle et [DEL-82] [DAT-77] [ULL-80] pour les bases de données.

5.1.1. En Base de Données

Il existe trois niveaux de représentation d'une base de données, qui sont le niveau interne, le niveau conceptuel et le niveau externe [ANS-75].

Le niveau interne décrit l'organisation physique des données sur les organes périphériques de l'ordinateur.

Le niveau conceptuel décrit l'organisation logique des données. Le schéma conceptuel, qui décrit en termes abstraits cette organisation, est le résultat d'une modèlisation du monde réel, où les objets sont classés en catégories et désignés par des noms. La spécification du schéma conceptuel est un processus fondamental dans la conception d'une base de données, qui doit respecter les conventions liées à un modèle de données. Un modèle de données est un outil formel pour comprendre et interpréter le monde réel; il permet de regrouper les objets en classes d'objets de nature identique, de nommer ces classes et de décrire les associations existant entre ces classes.

Enfin, le niveau externe correspond à la vision d'une partie du schéma conceptuel par un groupe d'utilisateurs.

Rappelons que dans les SGBDs, il existe trois principaux modèles de données : le modèle hiérarchique, le modèle réseau et le modèle relationnel.

Le modèle relationnel est le modèle le plus "conceptuel". Nous entendons par là que c'est celui qui permet une expression et une compréhension du contenu de la base de données de façon indépendante de ses caractéristiques physiques d'implantation. Cependant, l'expérience a montré qu'il n'était pas facile de représenter directement, avec ce modèle, un univers réel. C'est pourquoi, il a été proposé un certain nombre de modèles appelés modèles sémantiques,

qu'il est conseillé d'utiliser avant d'exprimer la réalité à modéliser sous formes de relations. Le modèle Entité Association [CHE-76] est le plus connu de ces modèles .

Depuis, de nombreux autres modèles sémantiques [HUL-87] ont été proposés dans l'objectif de mieux prendre en compte la structure d'objets complexes [ABI-87]. Récemment, on assiste à une nouvelle orientation qui consiste à choisir comme modèles de données des modèles issus des langages objets qui étaient plutôt utilisés en génie logiciel et en intelligence artificielle.

De nombreux travaux actuellement s'intèressent à la construction de nouveaux SGBDs reposant sur ces langages. Dans ces nouveaux modèles, on privilégie la représentation de la structure des objets et de leur comportement vis à vis de l'aspect relations entre objets ou classes d'objets. Ces langages étant issus de génie logiciel "et de l'intelligence artificielle" nous les présentons au paragraphe suivant.

Parmi les trois modèles de données, aucun ne propose des fonctions de manipulation des informations nuancées. Seules quelques propositions d'extensions du modèle relationnel (vues au chapitre II) ont été faites dans ce sens. Dans le chapitre IV, nous nous placerons dans le cadre du modèle relationnel pour proposer des schémas de relations permettant de tenir compte des informations nuancées et de résoudre le problème d'identification en présence de ces informations. Nous utiliserons ensuite une représentation de type "OBJET" présentant l'avantage de ne pas morceler en de nombreux nuplets la description des phénomènes réels.

5.1.2. En Intelligence Artificielle

En intelligence aritificielle , une polémique opposa, au cours des années 70, les tenants d'une représentation procédurale à ceux qui optaient pour une vision déclarative. Le but recherché par la dernière approche est de spécifier un savoir indépendamment des contraintes et des méthodes d'utilisation; on cherche à résoudre une question de type "quoi" alors qu'une procédure exprime par essence un flot d'informations et traduit "comment" transite la connaissance.

Puis sont apparues des représentations mixtes, telles que les "frames" [MIN-75], utilisant des procédures et des déclarations. La polémique opposa alors le formalisme relationnel au formalisme objet. Ce problème abordé au paragraphe 5.1.2.2. nous amènera à l'étude des représentations orientées objets.

Selon l'opposition procédural / déclaratif, les principales représentations des connaissances en intelligence artificielle sont purement déclaratives : la logique des prédicats, les réseaux sémantiques ou les règles de production. Parmi ces approches déclaratives, nous allons nous limiter à l'étude de la dernière approche "règles de production" car elle est la plus utilisée. Pour une présentation plus détaillée de ces approches voir [VIG-85].

5.1.2.1. Les règles de production

Les règles de production ont été utilisées dans les systèmes de production pour la première fois par Post en 1943.

Une règle de production est la spécification d'une action conditionnelle de la forme :

si <prédicat> **alors** <action>
antécédent conséquence

Les règles de production sont organisées sous forme d'un réseau c'est à dire que la conséquence d'une règle peut être l'antécédent d'une autre. Elles permettent de représenter la connaissance sous forme de petits modules indépendants.

Comme le montre Rechenmann [REC-84], la modularité des règles entraîne une dispersion des éléments de connaissance constituant chaque règle de production. Les notions de classification hiérarchique et de contexte ne peuvent donc pas être utilisées.

Enfin, cette représentation des connaissances répond mal au besoin de hiérarchiser les connaissances manipulées et d'exprimer des informations nuancées.

5.1.2.2. Les représentations orientées objets

Dans le formalisme relationnel classique [DEL-82], un objet n'existe pas en tant que tel mais seulement comme participant à un ensemble d'énoncés dispersés dans la base de connaissances.

A l'opposé, dans le formalisme objet, les entités ont une existence réelle et un espace propre dans la base de connaissances. Les représentations en objets structurés sont nées de la

conjonction d'idées de diverses sources. Elle s'inspirent des "schémas" résultats d'études en psychologie par Bartlett, des "frames" de Minsky [MIN-75], des "scripts" de Schank [SCH-77], des "objets" utilisés dans les langages orientés objets comme SMALLTALK [GOL-83] ou des types abstraits.

Les entités rassemblent la connaissance associée à un objet. Le terme d'objet recouvre plusieurs notions. Il peut désigner un objet au sens physique, mathématique ou au sens du domaine concerné. Des informations descriptives et déclaratives au sujet de la dynamique de l'objet décrit sont rassemblées dans le même concept .

Parmi les principales caractéristiques de la représentation à l'aide d'objets, on peut citer les trois suivantes :

- une représentation déclarative qui repose sur la notion d'objet. Le monde est vu comme un ensemble d'objets autonomes, chacun ayant une existence réelle dans la base de connaissances.
- chaque objet comporte à la fois une composante statique et une composante dynamique. On utilise ainsi une puissante combinaison de déclaratif et de procédural.
- les notions de hiérarchie d'objets et d'héritage jouent un rôle essentiel dans l'écriture et la manipulation de la base de connaissances.

L'ouvrage de Masini G. et al. [MAS-89] fournit une présentation détaillée et complète de ces types de représentation.

5.2. Modèle de description proposé

Nous présentons dans ce paragraphe nos choix de description et de représentation des données et du portrait robot.

Le mode de description des données que nous avons choisi est essentiellement inspiré des représentations orientées objets que nous proposons d'améliorer sur les points suivants :

- faciliter la description des données (ou objets) en permettant l'utilisation des informations nuancées exprimées à l'aide de termes du langage naturel,
- prendre en compte les notions d'importance des caractères (ou attributs) dans la description des objets,

- permettre un raisonnement nuancé et fournir une évaluation de distance entre les objets plutôt qu'une réponse binaire trop stricte.

Nous avons choisi ce type de représentation car les informations des domaines d'application auxquels nous nous intéressons sont déjà bien **structurées** et **hiérarchisées** dans la réalité. Par exemple, en Mycologie, les espèces sont regroupées en familles possédant des caractéristiques communes. De même en médecine il existe une classification des maladies établie par l'Organisation Mondiale de la Santé. De plus nous sommes en présence d'univers où les objets ont des structures complexes.

Enfin, le choix d'un modèle orienté objet nous permet d'inclure facilement dans la représentation, les informations particulières à notre approche c'est à dire :

- les nuances que nous conservons selon leur mode d'expression en langue naturelle,
- les coefficients d'importance accordés aux caractères du portrait robot et ceux des espèces.

Le schéma Entité-Association présenté en début de ce chapitre (paragraphe 1) correspond à un schéma externe possible de la base de connaissances. De ce schéma, nous avons extrait les classes objets suivantes :

- **DESCRIPTION** qui regroupe les descriptions des espèces et des groupes d'espèces,
- **PORTRAIT ROBOT** qui correspond à la description de l'exemplaire à identifier,
- **CARACTERE** qui assemble les caractères décrivant une espèce ou un groupe d'espèces,
- **DOMAINE** qui regroupe les domaines de définition des différents caractères,
- **TERME** qui regroupe les termes ou valeurs d'un domaine de définition,
- **NUANCE** qui regroupe les nuances qui peuvent être associées aux termes.

Dans la suite de ce paragraphe, nous ferons une présentation succincte des quatre premières classes d'objets. Les classes NUANCE et TERME ayant déjà été étudiées aux paragraphes précédents. Les choix d'implantation de ces classes, dans le langage de programmation retenu pour le prototype réalisé, seront donnés dans le chapitre V.

5.2.1. Les DESCRIPTIONS

A chaque type de description est associé l'ensemble des caractères ayant un sens pour ce type considéré, et à chacun de ces caractères est associée une liste de couples (VALEUR, NUANCE). L'ensemble de la description est complété par un caractère appelé IMAGE prenant comme valeurs des numéros d'images ou des séquences d'images, sur un vidéodisque, par exemple, illustrant l'objet associé à la description.

Ainsi, tout type description aura la structure suivante :

{<nom de la description>
SUPER-CLASSE : attribut indiquant la classe mère,
AIDE : un texte de définition en clair pour l'utilisateur,
LCS : attribut contenant la liste des caractères ayant un sens pour la description,
DEF : attribut contenant une liste de la forme $(C_i\ LVN_i)$ où
* C_i est le nom du caractère i
* LVN_i est une liste de la forme $(V^i_j\ LN^i_j)$ où
. V^i_j est la valeur j associée à C_i
. LN^i_j est la liste des nuances associées à V^i_j
}

Dans l'attribut **LCS** (Liste des Caractères Significatifs) chaque caractère est muni d'un poids appelé poids de pertinence et de fiabilité . Nous reviendrons sur le rôle de ces poids dans le chapitre suivant.

Exemple : emprunté à la Mycologie

Nous décrivons dans cet exemple la famille des BOLETS :

{ *BOLETS*
SUPER-CLASSE : *CHAMPIGNON*
AIDE : *"Nom générique pour les champignons à tubes muni d'un pied et dont la chair n'est pas coriace "*
LCS : *((couleur-appareil-sporifère 1) (couleur-chapeau 1) ...)*
DEF : *((couleur-appareil-sporifère ((blanc généralement) (jaune rarement)))*
(couleur-chapeau .....))
}

5.2.2. Le PORTRAIT ROBOT

Le modèle de description du portrait robot est une liste de la forme $(C_i\ P_i\ LVN_i)$ où

* C_i est le nom du caractère i donné par l'utilisateur,
* P_i est un poids de confiance, entre 0 et 1, que l'utilisateur accorde à son information,
* LVN_i est une liste de même type que celle ci-dessus pour décrire les descriptions.

Exemple : emprunté à la mycologie
((couleur-chapeau 1 (jaunâtre) (blanchâtre))
(diamètre-chapeau 0.8 (6, environ)))

5.2.3. Les CARACTERES

A chaque caractère nous associons :

- ***son domaine de définition***,
- ***son type*** qui indique si le caractère est monovalué ou multivalué. Cette information va permettre de définir le type de lien logique (ET, OU, OU Exclusif) entre les valeurs,
- ***la liste des nuances*** qui lui sont applicables. Cette liste servira éventuellement à l'utilisateur pour décrire son portrait robot ou à l'expert qui introduit des descriptions,
- ***les numéros ou les séquences d'images*** sur vidéodisque ou autre support, qui illustrent le caractère et servent également comme aide à la description du portrait robot,
- ***son poids de pertinence*** qui indique à quel degré le caractère est fiable et descriminant. Ce poids peut changer d'une description à une autre. Il représente le poids par défaut dans le cas où il ne serait pas donné au niveau de la description,
- ***sa nature***. Cette information permet de savoir si le caractère peut être renseigné ou non par l'utilisateur. Les natures de caractère envisagées sont :

* *INFORMATIF*, i.e. le caractère apporte de l'information, soit à l'utilisateur soit au système, mais ne peut être renseigné par l'utilisateur, par conséquent il n'est pas utilisé en phase d'identification. Exemples de caractères informatifs : les références à des livres, à des images,

* *VISIBLE* , i.e. l'utilisateur peut renseigner le caractère à l'oeil nu. Seuls les

caractères de cette nature seront utilisés pendant la phase d'identification.

* *INVISIBLE* , i.e. le caractère porte sur un élément microscopique comme la forme des spores par exemple.

En résumé, la structure de la classe des caractères est la suivante :

{ <nom de caractère>
SUPER-CLASSE
DOMAINE
TYPE
NUANCES
IMAGES
POIDS
NATURE
}

Exemple : emprunté à la Mycologie
Nous décrivons dans cet exemple le caractère ALLURE DU CHAPEAU :

{ *ALLURE DU CHAPEAU*
SUPER-CLASSE : *CARACTERE*
DOMAINE : *FORME-CHAPEAU*
TYPE : *MULTIVALUE*
NUANCES : *(TRES , GENERALEMENT, RAREMENT , ...)*
IMAGES : *(2 , 4)*
POIDS : 1
NATURE : *VISIBLE*
}

5.2.4. Les DOMAINES

Un domaine est décrit par :
- ***son type*** DISCRET ou CONTINU,
- ***sa définition***, i.e. l'ensemble des valeurs composant le domaine. La définition du domaine dépend de son type :
  + dans le cas d'un domaine continu, la définition est soit :
    * un intervalle de valeurs, représenté par un couple (<Borne Inf>,<Borne Sup>)
    * un type prédéfini ENTIER, REEL, ...
  + dans le cas d'un domaine discret, la définition est soit :
    * un ensemble discret de valeurs : {<valeur$_1$>, ... , <valeur$_n$>}
    * un microthésaurus, le domaine est alors défini par les termes du thésaurus qui sont liés au noeud racine du microthésaurus par un lien de type "spécifique". L'avantage de l'utilisation des microthésaurus comme domaine est double : il évite la duplication des termes d'une part, et permet d'autre part à l'utilisateur (en cas d'aide) de naviguer dans le thésaurus afin de mieux exprimer son besoin.

La structure de la classe DOMAINE est donc :

{ <nom du domaine>
SUPER-CLASSE
TYPE
DEFINITION
}

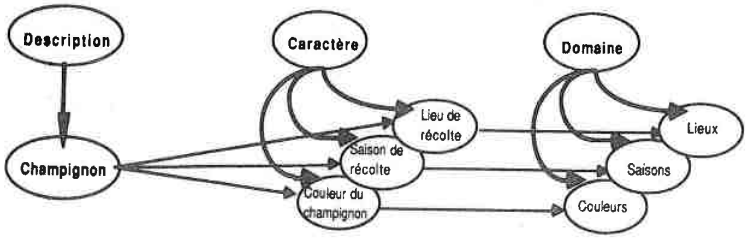
Exemples :

{ 2:10
SUPER-CLASSE : *DOMAINE*
TYPE : *CONTINU*
DEFINITION : (2 10)}

{ *COULEUR*
SUPER-CLASSE : *DOMAINE*
TYPE : *DISCRET*
DEFINITION : *THESAURUS-COULEURS* }

6. CONCLUSION

Nous venons de faire l'inventaire des différents types de connaissances nécessaires à la représentation et à la gestion des informations nuancées. Une partie de ces connaissances est représentée par des classes d'objets dont la figure ci-dessous schématise les liens qui existent entre elles :



LEGENDE :

a comme sous-type

a comme attribut

Dans le chapitre suivant, nous montrons comment ces connaissances sont utilisées dans le processus d'identification nuancée.

Chapitre IV IDENTIFICATION NUANCEE

Chapitre IV IDENTIFICATION NUANCEE

L'identification d'un objet observé parmi un ensemble d'objets connus est un problème auquel on est souvent confronté dans la réalité. Par exemple :

- en médecine : identifier une maladie à partir d'un certain nombre de signes observés,
- en science naturelle : identifier l'espèce à laquelle appartient un exemplaire observé ,
- en jutice : identifier un malfaiteur à partir d'une description vague.

L'objet observé par l'utilisateur est souvent décrit d'une manière pauvre et nuancée sous la forme d'un portrait robot, quant aux objets connus, ils sont en général décrits de manière plus ou moins riche mais néanmoins nuancée.

Dans le paragraphe 1, nous situons le problème à résoudre vis à vis des approches bases de données et intelligence artificielle.

Dans le paragraphe 2, nous exposons notre solution. Cette solution s'appuie sur les connaissances introduites précédemment et en particulier sur le mode de calcul des fonctions caractéristiques que nous avons définies au chapitre précédent. Elle consiste à appliquer la théorie des possibilités pour déterminer les éléments de la base de connaissances les plus "proches" de l'exemplaire à déterminer. Afin d'améliorer encore le résultat de l'identification nuancée, nous avons adapté l'application de la théorie des possibilités de deux manières :

- en prenant en compte les pondérations sur les caractères, que ce soit dans les descriptions des espèces ou dans l'exemplaire à identifier,
- en incorporant la solution dans un processus coopératif et progressif du type processus EXPRIM [CRE-86] d'interrogation de bases d'images.

1. POSITION DU PROBLEME

1.1. En Base de Données

Résoudre le problème d'identification nuancée à l'aide d'un système de gestion de base de données consiste à chercher la réponse à *une requête nuancée* - correspondant au portrait

robot - *dans une base de données nuancées* - correspondant aux descriptions des espèces connues -.

Nous montrons ici que sous l'angle de l'algèbre relationnelle ce problème nécessite la mise en œuvre d'un nouvel opérateur appelé : **division relationnelle nuancée** que nous introduisons en montrant tout d'abord que si le portrait robot (i.e. la requête) est décrit d'une manière précise et si les descriptions des espèces (i.e. les données) le sont également, l'identification est une division relationnelle.

Remarque :

Nous avons choisi dans ce paragraphe une représentation des espèces et du portrait-robot utilisant le modèle relationnel afin de mettre en évidence le type de problème auquel correspond l'identification. Ceci n'est pas une contradiction avec le modèle de représentation orienté objet présenté au chapitre précédent mais simplement un support local de réflexion.

1.1.1. Rappel : division relationnelle

. Définition [DEL-82] :

Soient R une relation définie par deux attributs X et Y, notée R(X,Y) et S une relation définie par l'attribut Y, notée S(Y).

La division relationnelle de R(X,Y) par S(Y), notée **R÷S** est une relation définie sur l'attribut X par :

$$R \div S = \{x / \forall y \in S(Y), (x,y) \in R(X,Y)\}$$

où (x,y) est le n-uplet formé des éléments x et y.

. Exemple (extrait de [DEL-82])

Etant donné :

- une relation R de schéma R(Pièce,Fournisseur) définie par le tableau de valeurs suivant :

R	Fournisseur	Pièce
	pierre	*vis*
	paul	*boulon*
	jacques	*écrou*
	paul	*vis*
	pierre	*boulon*
	jacques	*boulon*
	pierre	*écrou*

- et une relation S de schéma S(Pièce) définie par le tableau de valeurs suivant :

S	Pièce
	vis
	boulon

L'opération R÷S détermine l'ensemble des fournisseurs qui approvisionnent au moins les *vis* et les *boulons*, soit {*pierre* , *paul* }.

1.1.2. L'identification en l'absence de nuances est une division relationnelle

En l'absence de nuances les espèces peuvent être décrites par une relation Desc, de schéma :

Desc (nom-espèce, nom-caractère, valeur)

Un triplet (e,c,v) signifie : l'espèce e peut prendre la valeur v pour le caractère c.

Cette façon de décrire les espèces par des triplets correspond à la théorie A.O.V. <Attribut, Objet, Valeur> introduite par Feldman et Rovner [FEL-69].

Le portrait robot à identifier peut également être représenté par une relation PR,de schéma :

PR (nom-caractère, valeur)

Un couple (c,v) signifie : l'exemplaire à identifier décrit par le portrait robot présente la valeur v pour le caractère c.

Etant donné ces deux relations, l'identification de l'exemplaire parmi l'ensemble des espèces consiste à trouver toutes les espèces qui contiennent au moins tous les couples (c,v) de la relation PR, autrement dit, à trouver une relation E de schéma E(nom-espèce) définie comme suit :

$$E = \{e / e \in Desc[\text{nom-espèce}] \text{ et } \forall (c,v) \in PR ; (e,c,v) \in Desc\}$$

où Desc[nom-espèce] est la projection de Desc sur l'attribut "nom-espèce".

Par définition de l'opérateur division de l'algèbre relationnelle (cf §1.1.1), E est le résultat de la division de la relation Desc par la relation PR.

1.1.3. L'identification en présence de nuances est une division relationnelle nuancée

En présence de nuances, soit dans les descriptions des espèces, soit dans le portrait robot, soit dans les deux, l'identification ne consiste plus à trouver les descriptions des espèces qui présentent, pour tous les caractères, les mêmes valeurs que celles du portrait robot mais qui présentent des valeurs "proches".

Nous disons , dans le cas où Desc ou PR, ou les deux, présentent des nuances, que **E est le résultat de la division nuancée de Desc par PR**. Donnons la définition de cette opération dans le cas où Desc et Pr ont les schémas suivants :

Desc (nom espèce, nom de caractère, valeur, nuance)

PR (nom de caractère, valeur, nuance)

PR pouvant présenter plusieurs valeurs pour le même caractère.

$$\mathbf{E(nom\ espèce) = \{e / e \in Desc[nom\_espèce]\ et\ \forall (c, v, n) \in PR,}$$

$$\mathbf{(e, c, v', n') \in Desc\ et\ proche\ (v, v', n, n')\ \}}$$

La proximité entre "e" et le portrait robot de l'exemplaire à identifier dépend de l'ensemble des nuances n et n' mises en jeu dans l'opération de division nuancée. La relation "proche" permet d'évaluer la proximité entre deux couples (valeur , nuance).

Dans le paragraphe 2 nous présentons une méthode de mise en œuvre de ce nouvel opérateur où la relation "proche" sera traduite par les deux mesures de possibilité et de nécessité de la théorie de possibilités.

Il est intéressant d'avoir situé le type de problème auquel correspond l'identification vis à vis de l'approche relationnelle. En effet, tout spécialiste des SGBDs relationnels sait que la division relationnelle est l'opération la plus dificile à exprimer dans un langage relationnel et que c'est également une opération très lourde au point de vue temps d'exécution qui nécessite donc une étude de l'optimisation de sa mise en oeuvre dans chaque cas particulier.

Notre approche ayant pour conséquence une augmentation de la complexité de

l'opération d'identification qui n'est plus une simple division relationnelle mais une division relationnelle nuancée, il n'était absolument pas envisageable d'utiliser un SGBD relationnel pour sa mise en oeuvre.

1.2. En Intelligence Artificielle

Résoudre le problème d'identification nuancée en intelligence artificielle consiste à trouver un schéma de raisonnement capable de mettre en correspondance un fait nuancée - correspondant au portrait robot - avec une règle nuancée - correspondant à la description d'une espèce -.

Nous avons déjà justifié au paragraphe 2 du chapitre 2 le besoin d'un mode d'inférence plus général que le modus ponens classique, afin de propager des informations nuancées à travers les règles. Le modus ponens généralisé introduit par Zadeh [ZAD-79] (cf chapitre 2 §3.2.2.2.2) répond à ce besoin.

Des systèmes comme TOULMED (cf chapitre 2 §3.2.2.2.2) utilisent ce schéma de raisonnement. Dans notre approche, nous l'avons adapté à notre type de représentation des connaissances (cf chapitre 3) et étendu afin de tenir compte des pondérations dans les faits et dans les règles.

2. METHODE DE RESOLUTION PROPOSEE

Après avoir défini au paragraphe 5 du chapitre précédent un modèle de description capable d'accueillir des informations nuancées, nous proposons ici un processus d'identification en accord avec ce modèle. Ce processus basé sur le filtrage flou [CAY-82] [UMA-79] [DUB-87] [VIG-85] et la théorie des possibilités, permet d'identifier un exemplaire défini par un portrait robot parmi un ensemble d'espèces eux mêmes décrits de manière nuancée.

2.1. Filtrage

2.1.1. Filtrage classique

Cette technique classique en intelligence artificielle permet d'extraire d'une base l'ensemble des descriptions d'espèces correspondant à un portrait robot (i.e. le filtre) [CHA-83] [WAR-78].

Les procédés ordinaires de filtrage sont fondés sur une conception rigide de la similarité entre les caractères du portrait robot et ceux d'une description d'espèce. La similarité globale est calculée comme une combinaison logique des similarités atomiques binaires.

Aucune imprécision ou incertitude n'est permise dans l'expression des caractères. Un langage restreint et strict est imposé pour la spécification des connaissances.

Les processus d'identification classiques n'utilisent pas ou peu la sémantique des informations. Ils se contentent d'une mise en correspondance exacte et syntaxique entre les valeurs des caractères. De tels processus rendent des distances binaires (0 ou 1) et ne permettent pas un raisonnement nuancé. Des solutions admissibles sont ainsi rejetées, faute de nuances, ce qui peut être la cause d'un échec global du processus.

2.1.2. Filtrage flou

Les processus de filtrage flou permettent de manipuler et d'exprimer des connaissances nuancées facilitant ainsi la communication homme-machine. Le résultat du filtrage flou est une note entre 0 et 1 qui évalue la distance entre le portrait robot (i.e. le filtre) et la description d'une espèce (i.e. la donnée). Les processus de filtrage flou développés jusqu'à présent [BUI-87] [VIG-85] [DUB-87] [TES-84] sont fondés sur la théorie des ensembles flous et sur la théorie des possibilités.

2.2. Identification nuancée

Dans la suite de ce paragraphe, nous utiliserons les notations du modèle de description présenté au chapitre précédent paragraphe 5 et que nous rappelons ci-dessous :

- structure d'une description

{<nom de la description>

SUPER-CLASSE : attribut indiquant la classe mère,

AIDE : un texte de définition en clair,

LCS : attribut contenant la liste des caractères ayant un sens pour la description,

DEF : attribut contenant une liste de la forme $(C_i\ LVN_i)$ où

* C_i est le nom du caractère i
* LVN_i est une liste de la forme $(V^i_j\ LN^i_j)$ où
  . V^i_j est la valeur j associée à C_i
  . LN^i_j est la liste des nuances associées à V^i_j

}

Dans l'attribut **LCS** (Liste des Caractères Significatifs) à chaque caractère est attaché un poids appelé poids de pertinence et de fiabilité . Nous reviendrons sur le rôle de ces poids dans la suite de ce chapitre.

- la structure du portrait robot

Le modèle de description du portrait robot est une liste de la forme $(C_i\ P_i\ LVN_i)$ où

* C_i est le nom du caractère i donné par l'utilisateur,
* P_i est un poids de confiance, entre 0 et 1, que l'utilisateur accorde à son information,
* LVN_i est une liste de même type que celle ci-dessus pour décrire les descriptions.

Le processus d'identification nuancée que nous avons mis au point est constitué de deux étapes principales : **le filtrage hiérarchique** et **le filtrage élémentaire**. Dans la première étape, le système vérifie que le portrait robot se situe bien dans la hiérarchie/classe de la description. Lors de la seconde étape, des mesures de compatibilités sont évaluées, entre le portrait robot et la description, à partir des valeurs des caractères.

2.2.1. Filtrage hiérarchique

Le filtrage hiérarchique repose sur une comparaison des structures du portrait robot et de la description. Cette comparaison est une mise en correspondance entre la liste des noms des caractères (soit LCR^{PR}) renseignés au niveau du portrait robot et la liste des noms des

caractères (soit LCS^D) définissant la description.

La mise en correspondance réussit lorsque la liste LCR^{PR} est incluse dans la liste LCS^D, le processus rend alors la description D comme résultat possible et procède à une deuxième comparaison entre la liste LCR^{PR} et la liste des noms des caractères (soit LCR^D) renseignés dans la description (ce sont les caractères "visibles" de la liste DEF de la description). Cette deuxième étape de comparaison permet de savoir si un filtrage flou peut être effectué entre le portrait robot et la description ou s'il faut descendre dans la hiérarchie afin de procéder à d'autres filtrages hiérarchiques.

En cas d'échec global du processus, c'est à dire si aucune description n'a été retenue par le filtrage hiérarchique, le système reprend le processus après avoir éliminé les caractères du portrait robot qui sont les moins fiables. Cet aspect du processus sera repris dans le paragraphe 2.3. de ce chapitre.

2.2.2. Filtrage élémentaire

Si le filtrage hiérarchique ne conduit pas à un échec, le processus calcule, pour chaque description retenue, une distance globale entre celle-ci et le portrait robot. Cette distance globale est obtenue par une combinaison de distances élémentaires entre les caractères respectifs du portrait robot et de la description retenue.

Dans le but d'utiliser la théorie des possibilités, nous avons représenté les valeurs nuancées des caractères par des fonctions caractéristiques ou distributions de possibilité (nous ne faisons pas de distinction entre les deux dans le cadre de ce travail) (cf chapitre 3). Ainsi, la distance entre le portrait robot PR et une description D est donnée par les deux mesures de possibilité Π et de nécessité N (cf chapitre 2).

Le résultat du filtrage élémentaire est alors, un ensemble R tel que :

(0) $\mathbf{R = \{(D,\Pi,N) / D \in BD, \Pi(PR,D) \geq Seuil1 \text{ et } N(PR,D) \geq Seuil2\}}$

où

- BD est la Base des Descriptions
- Seuil1 et Seuil2 sont deux nombres entre 0 et 1 fixés au départ.

Les mesures globales Π et N sont obtenues par agrégation de mesures élémentaires Π_i et N_i. Π_i et N_i mesurent la distance de compatibilité entre les valeurs nuancées du caractère C_i

du portrait robot et celles du caractère C_i de la description.

L'agrégation des mesures Π_i et N_i dépend de la prise en compte ou de la non prise en compte des pondérations sur les caractères, qu'elles proviennent du portrait robot ou de la description. Nous allons étudier séparément ces deux cas.

2.2.2.1. Sans pondération des caractères

La mesure Π (resp N) est obtenue par une combinaison des mesures de possibilité Π_i (resp nécessité N_i). Une mesure Π_i évalue le degré d'intersection entre l'ensemble flou des valeurs compatibles avec LVN^D_i (la liste des valeurs attachées au caractère C_i dans D) et l'ensemble flou des valeurs compatibles avec LVN^{PR}_i (la liste des valeurs attachées au caractère C_i dans PR). En revanche, une mesure de nécessité N_i évalue un degré d'inclusion de l'ensemble flou des valeurs compatibles avec LVN^D_i dans l'ensemble flou des valeurs compatibles avec LVN^{PR}_i.

La combinaison des mesures Π_i (resp N_i) pour calculer la mesure globale Π (resp N) utilise l'opérateur "Min" [DUB-87] qui permet de préserver la sémantique des mesures de possibilité et de nécessité ainsi que la sémantique de l'identification nuancée dont les descriptions résultats doivent vérifier tous les caractères du portrait robot. On a alors les résultats suivants :

(1) $\Pi = \mathbf{Min}_{i=1,n}\, \Pi_i$ où n est le nombre de caractères dans PR

(2) $N = \mathbf{Min}_{i=1,n}\, N_i$

Soit Dom_i le domaine des valeurs du caractère C_i, et μ^{PR}_i (resp μ^D_i) la fonction caractéristique associée à la liste des valeurs attachées au caractère C_i dans le portrait robot (Resp dans la description D).

Les mesures Π_i et N_i sont alors respectivement définies suivant les résultats de Zadeh et Dubois [ZAD-78] [DUB-87] par :

(3) $\Pi_i = \mathbf{Sup}_{v \in Dom_i}\, \mathbf{Min}\ (\mu^{PR}_i(v), \mu^D_i(v))$

(4) $N_i = \mathbf{Inf}_{v \in Dom_i}\, \mathbf{Max}\ (\mu^{PR}_i(v), 1 - \mu^D_i(v))$

Remarque :

L'agrégation en (1) et (2) des mesures élémentaires Π_i et N_i par l'opérateur "Min" reflète une conjonction des caractères du portrait robot. Dans le cas où ce dernier correspond à une disjonction de caractères l'opérateur "Min" devient "Max" en (1) et (2). On peut bien sûr se poser le problème des caractères structurés en arborescence ET/OU qui se traduirait dans ce cas par une combinaison de "Min" et de "Max". Mais comme nous nous posons le problème d'identification et non les problèmes d'interrogation d'une base de données, nous supposons alors que le portrait robot est une conjonction de caractères.

2.2.2.2. Avec pondération des caractères

La solution précédente de calcul des mesures Π et N suppose que les différents caractères de la description ont tous la même importance. En réalité , certains caractères sont plus discriminants et plus stables dans le temps donc plus pertinents que d'autre; pour cela nous accordons (ou plutôt l'expert accorde) un poids de pertinence à chacun des caractères C_i d'une description donnée (cf chapitre 3 § 5.2.1.). Un poids de pertinence par défaut est spécifié au niveau de la définition du caractère (cf chapitre 3 §5.2.3.).

Au chapitre 3, paragraphe 5, le modèle de représentation proposé permet d'introduire également des poids dans le portrait robot. Un poids p_i exprime la confiance accordée par l'utilisateur aux valeurs nuancées qu'il a données au caractère C_i .

Notons p^{PR}_i et p^{D}_i les poids accordés respectivement au caractère C_i du portrait robot PR et au caractère C_i de la description D. Ces poids doivent vérifier les hypothèses énumérées ci-dessous; soient respectivement $p_1,...,p_n$ les poids accordés aux caractères C_1 ,...,C_n :

i) $\forall$ i, $p_i \in [0,1]$

ii) p_i est d'autant plus grand que C_i est pertinent

iii) $Max_{i=1,n}\, p_i = 1$ (condition de normalisation)

NB : Les caractères jugés les plus pertinents sont pondérés à 1.

En tenant compte des poids p^{PR}_i et p^{D}_i accordés aux différents caractères C_i , les nouvelles mesures Π et N sont alors données par :

(5) $\Pi = Min_{i=1,n}\, Max\,(1-p^{PR}_i\,,\, 1-p^{D}_i\,,\, \Pi_i)$

(6) $N = Min_{i=1,n}\, Max\,(1-p^{PR}_i\,,\, 1-p^{D}_i\,,\, N_i)$

Les mesures Π et N expriment à quel point on est certain que la partie importante (défin par les poids p^{D}_i) d'une description D satisfait la partie certaine (définie par les poids p^{PR}_i) du portrait robot. Notons que si tous les poids p^{PR}_i sont égaux à 1 (i.e. toutes les informations données dans le portrait robot sont certaines) et si tous les poids p^{D}_i sont égaux à 1 (i.e. tous les caractères de la description sont pertinents), on retrouve les formules (1) et (2). Lorsqu' un poids $p^{D}_i = 0$ (resp $p^{PR}_i = 0$) , c'est à dire le caractère C_i n'est pas du tout pertinent (resp les valeurs nuancées associées au caractère C_i sont incertaines), le caractère C_i n'est pas pris en compte dans le calcul des mesures Π et N .

Remarque :

Si l'on rapproche les formules (5) et (6) des égalités (1) et (2), on peut affirmer que les schémas d'agrégation proposés dans les formules (5) et (6) fournissent respectivement des mesures de possibilité et de nécessité. Pour cela, nous posons :

(7) $\Pi^*_i = Max\,(1-p^{PR}_i\,,\, 1-p^{D}_i\,,\, \Pi_i)$

(8) $N^*_i = Max\,(1-p^{PR}_i\,,\, 1-p^{D}_i\,,\, N_i)$

Par conséquent, les formules (5) et (6) deviennent :

(9) $\Pi = Min\ \Pi^*_i$

(10) $N = Min\ N^*_i$

(voir Dubois [DUB-86] pour une preuve dans le cas où on n'utilise que les poids p^{PR}_i).

On peut également rapprocher les formules (7) et (8) des égalités (3) et (4). En effet, à partir des formules (7) et (8) nous avons les égalités suivantes (voir preuve en annexe) :

(11) $\Pi^*_i = Sup_{v \in Domi}\, Min\,(\mu^*_i{}^{PR}(v)\,,\, \mu^*_i{}^{D}(v))$

(12) $N^*_i = Inf_{v \in Domi}\, Max\,(\mu^*_i{}^{PR}(v)\,,\, 1-\mu^*_i{}^{D}(v))$

où

(13) $\mu^*_i{}^{PR}(v) = Max\,(1\text{-}p^{PR}_i\,,\,\mu^{PR}_i(v))$

(14) $\mu^*_i{}^{D}(v) = Max\,(1\text{-}p^{D}_i\,,\,\mu^{D}_i(v))$

Conclusion :

Un des avantages de la prise en compte des poids pour évaluer la compatibilité entre le portrait robot et les descriptions, est d'éviter d'éliminer des descriptions n'ayant qu'une seule mesure élémentaire $\prod_i$ nulle correspondant à un caractère non pertinent.

L'application des formules ci-dessus dans un cas réel, engendre un très grand nombre de calculs pour la plupart inutiles. Dans le système réalisé, nous utilisons la pertinence des caractères pour diminuer le nombre de calculs. En effet, nous exploitons les caractères dans l'ordre des pertinences décroissantes de façon à arrêter les calculs dès qu'un caractère pertinent et certain C_i ($p^D_i = 1$, $p^{PR}_i = 1$) a sa mesure de possibilité nulle ($\prod_i = 0$).

Notons qu'en cas d'échec, où aucune description n'étant suffisamment proche du portrait robot, il peut être intéressant de recommencer la recherche en éliminant un ou plusieurs caractères. Le choix des caractères à éliminer peut être fait parmi les caractères pertinents ayant une mesure $\prod_i$ égale à 0. Pour cela, il est intéressant d'intégrer la solution dans un processus de recherche permettant des retours-arrière comme nous le proposons au paragraphe suivant.

2.3. Intégration de l'identification nuancée dans un processus de recherche progressif et coopératif : le processus EXPRIM

L'équipe EXPRIM du CRIN (Centre de Recherche d'Informatique de Nancy) a proposé un processus d'interrogation progressif et coopératif d'une base d'images appelé le processus EXPRIM [CRE-85] que nous avons déja présenté au paragraphe 2.2.2.3 du chapitre 2. Ce processus a été mis en oeuvre dans plusieurs systèmes :

- le système MYCOMATIC, développé par O. Foucaut, J.F. Foucaut et P. Bourret [FOU-88], d'aide à l'identification de champignons qui est à l'origine de cette thèse,
- le système RIVAGE, développé par G. Halin [HAL-89], utilise des méthodes d'apprentissage pour essayer de comprendre et de représenter les besoins de l'utilisateur

pendant une phase d'interrogation d'une base d'images,
- le système BIRDS, développé par A. David [DAV-89], a pour objectif de trouver une méthodologie pour construire des scénarios d'EAO fondés sur l'image.

Ces trois systèmes ne comportent comme gestion d'informations nuancées que :

- la possibilité d'utiliser des termes généraux qui sont remplacés totalement par la liste de leurs spécifiques dans un thésaurus,
- la possibilité de fournir des portraits robots incomplets en ne répondant pas à certaines questions.

Nous allons montrer comment notre méthode d'identification nuancée s'intègre dans ces systèmes et les avantages qu'elle y apporte.

Pour cela, nous reprenons le processus d'identification progressif et interactif proposé par O. Foucaut et al. [FOU-89] pour les systèmes multimédia d'aide à l'identification de phénomènes naturels et nous y situons notre solution.

2.3.1. Le processus d'identification progressif et interactif

2.3.1.1. Utilisation d'une base d'images

L'utilisation d'une base d'images dans le processus d'identification a un double objectif :

- aider l'utilisateur à choisir parmi les descriptions trouvées après une phase d'identification, en lui proposant de regarder les images associées à ces descriptions. L'illustration des résultats par l'image permet de rendre compte totalement de tous les aspects morphologiques d'une description ainsi que de toutes les nuances, ce qui n'est pas le cas d'une illustration par le texte,
- aider l'utilisateur à formuler son portrait robot en lui affichant des schémas explicatifs. Ces schémas sont de deux types :
  * les schémas explicatifs dont le but est d'illustrer un terme particulier utilisé (par exemple "Lamelles décurrentes" en mycologie),
  * les schémas morphologiques reproduisant des allures ou formes courantes pour certains caractères.

Remarque :

Les images manipulées peuvent être soit numérisées soit stockées sur un vidéodisque. Dans le premier cas les images sont visualisées sur l'écran de l'ordinateur, dans le second cas elles sont affichées sur un moniteur couleur connecté à un lecteur vidéodisque. A la place du moniteur couleur, il serait plus agréable et plus efficace d'utiliser un poste de travail du type de l'imageur développé par l'agence SYGMA puis par la SEP [HUD-85] . Ce poste permet de visualiser, simultanément, jusqu'à 16 images, il permet également de trier un ensemble d'images, en constituant des piles d'images. Un tel outil permettrait à l'utilisateur de mieux comparer les photos d'espèces proposées, de revisualiser une photo particulière, etc ...

2.3.1.2. Le processus de détermination de MYCOMATIC

MYCOMATIC est un système de gestion et d'interrogation de connaissances mycologiques intégrant textes et images. La fonction de détermination est la plus importante du système, elle a pour objectif d'aider un utilisateur à déterminer l'espèce à laquelle appartient un exemplaire qu'il a observé ou récolté. C'est cette fonction que nous proposons d'étendre aux informations nuancées.

Le processus de détermination proposé, dans MYCOMATIC, est voisin du processus de recherche d'images dans une photothèque couplée à une base documentaire tel qu'il a été défini à l'origine dans le projet EXPRIM [CRE-83].

La figure ci-dessous illustre ce processus de détermination.

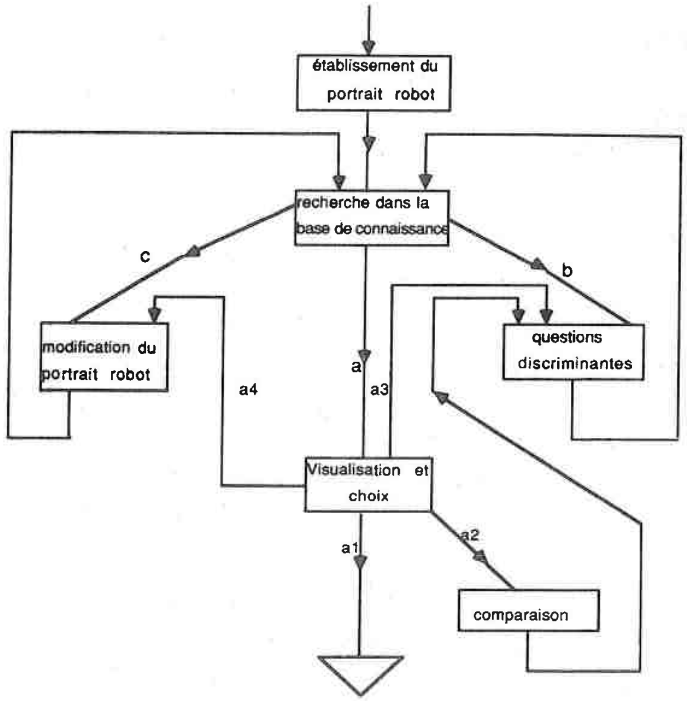


Figure : Processus de détermination

Ce processus consiste en une alternance de phases de description et de phases de visualisation d'images. L'utilisateur fournit tout d'abord son portrait-robot. Le système transforme le portrait-robot en une description interne en appliquant des règles générales et des règles spécifiques au domaine ou aux termes utilisés.

Le système confronte cette description interne à celle des espèces de la base de connaissances. Trois situations peuvent alors se produire :

a) le nombre d'espèces trouvé est correct, ni trop grand, ni nul. Le seuil limite dépend du domaine. Dans MYCOMATIC, le seuil est placé à 6. Dans ce cas le système passe en phase de visualisation, l'utilisateur peut alors consulter à la fois les photos des espèces trouvées et leurs descriptions textuelles.

b) le nombre d'espèces trouvé est trop grand. Le système pose alors des questions supplémentaires ou demande des précisions sur les caractères décrits trop généralement et reprend la recherche.

c) le nombre d'espèces trouvé est nul. Le système propose à l'utilisateur de modifier son portrait robot.

Dans une phase de visualisation, l'utilisateur, après avoir confronté son exemplaire avec les photos et les descriptions des espèces retenues, indique celles qu'il retient et celles qu'il élimine. A l'issue d'une phase de visualisation, quatre situations sont possibles :

a1) Une seule espèce est retenue : la détermination est supposée réussie ;

a2) Deux espèces sont retenues, la fonction "comparaison" aide l'utilisateur à étudier leurs caractères différents ;

a3) Plus de deux espèces sont retenues, le système recherche les caractères différenciant ces espèces et pose des questions les concernant ;

a4) Aucune espèce n'est retenue, le système propose à l'utilisateur de modifier le portrait-robot.

2.3.2. Le processus d'identification nuancée

La solution que nous proposons apporte au processus précédent les améliorations suivantes :

- classement des résultats par proximité décroissante,
- possibilité de retour-arrière avec déduction automatique des caractères:
  + soit à éliminer dans le cas où aucune description n'est trouvée après une phase d'identification,
  + soit à renseigner lorsque le nombre de descriptions trouvées est trop important. Le choix des caractères à renseigner est effectué parmi les caractères les plus pertinents.

Indépendamment de ces améliorations, notre approche apporte aux systèmes précédents les deux avantages suivants :

- possibilité de décrire les espèces ou les phénomènes réels de manière nuancée en conservant les nuances sous la forme de termes en langue naturelle,
- possibilité d'utiliser également ces mêmes nuances dans le portrait-robot.

2.3.2.1. Classement des résultats

Après une phase d'identification nuancée, on dispose d'un ensemble de descriptions, chacune d'elles étant caractérisée par une mesure de possibilité (Π) et une mesure de nécessité (N). Pour l'utilisateur, il est intéressant d'obtenir les descriptions qui donnent les meilleures mesures vis à vis du portrait robot. Pour cela, il faut établir une méthode de classement entre les couples (Π,N) associés aux différentes descriptions D obtenues.

Le choix d'une méthode de classement peut être guidé par les remarques suivantes :

- la mesure de nécessité est plus importante que la mesure de possibilité,
- si pour chaque description on obtient une mesure de nécessité nulle et une mesure de possibilité égale à 1, c'est que le portrait robot est exprimé d'une manière précise par rapport aux descriptions disponibles.

La méthode de classement que nous avons retenue est celle proposée par Dubois et Prade [DUB-87] qui utilisent à leur tour l'ordre de Pareto connu sous le nom de l'ordre produit. Dans l'ordre de Pareto, le classement de deux couples (Π_1,N_1) et (Π_2,N_2) se fait de la façon suivante :

$$(\Pi_1,N_1) > (\Pi_2,N_2) \Leftrightarrow (\Pi_1>\Pi_2 \text{ et } N_1 \geq N_2) \text{ ou } (\Pi_1 \geq \Pi_2 \text{ et } N_1> N_2)$$

L'ordre de Pareto étant seulement partiel, il existe cependant des situations où $(\Pi_1>\Pi_2$ et $N_1< N_2)$ ou $(\Pi_1<\Pi_2$ et $N_1> N_2)$. Pour remédier à ce problème, Dubois et Prade ont envisagé alors une mesure de précision égale à $(\Pi - N)$ associée à chaque description D telle que : $\Pi - N$ est d'autant plus proche de zéro que D est pertinente par rapport au portrait robot. Ainsi, dans la situation où $(\Pi_1>\Pi_2$ et $N_1< N_2)$ et sachant que pour tout couple (Π,N) Π est supérieur à N ($\Pi > N$ voir chapitre 2), on a alors $\Pi_1>\Pi_2>N_2>N_1$, ce qui indique que la description D_2 associée au couple (Π_2,N_2) est la meilleure du point de vue de la précision que de la nécessité.

2.3.2.2. Retour-arrière avec déduction automatique

A l'issue d'une étape d'identification, trois situations peuvent se présenter :

- ***Cas 1 : aucune description possible n'est trouvée.*** Dans ce cas le système effectue un retour-arrière pour éliminer un ou plusieurs caractères. Le choix des caractères à éliminer est fait parmi les caractères pertinents C_i (p_i =1) ayant une mesure Π_i égale à 0. Puis le processus d'identification est relancé.

- ***Cas 2 : un très grand nombre de descriptions possibles est trouvé.*** Le système ne peut pas visualiser toutes ces descriptions, pour cela des compléments d'information sont demandés sur les caractères pertinents non renseignés dans le portrait robot. En tenant compte de ce supplément d'information, le système essaye de discriminer entre les descriptions trouvées. Ce processus est itéré jusqu'à ce que le nombre de descriptions possibles soit inférieur à un seuil donné.

- ***Cas 3 : le nombre de descriptions trouvées est inférieur à un seuil donné.*** Pour chacune des descriptions trouvées, le système visualise la ou les images qui lui sont associées et demande à l'utilisateur s'il reconnaît son portrait robot. A la fin de cette étape, trois situations peuvent se présenter à nouveau :

+ ***Cas 3.1 : l'utilisateur n'a reconnu son portrait robot dans aucune des descriptions proposées.*** Le système effectue, là également, un retour-arrière pour éliminer les caractères du portrait robot dont le poids de confiance est inférieur à un seuil. Nous envisageons également de diminuer les seuils d'acceptation des mesures Π et N .

+ ***Cas 3.2 : l'utilisateur hésite entre deux ou plusieurs descriptions trouvées*** , le système passe alors dans une phase de discrimination. Le rôle de cette phase est de chercher s'il existe entre les descriptions en litige des caractères pertinents (poids =1) qui peuvent les discriminer; si oui, le système demande des compléments d'information sur les caractères ainsi trouvés comme dans le cas 2.

+ ***Cas 3.3 : l'utilisateur a reconnu son portrait robot,*** il peut alors demander la visualisation de tous les caractères de la description avec leurs valeurs. A l'issue de la visualisation, l'utilisateur peut remettre en cause sa décision, ce qui replace le système dans le cas 3.1.

2.3.3. Deux propositions supplémentaires pour l'amélioration des systèmes d'identification nuancée

Dans l'avenir il serait possible d'améliorer encore ces systèmes d'identification nuancée de deux manières :

- en permettant à l'utilisateur de décrire son portrait robot uniquement à partir d'images,
- en générant le portrait robot et les descriptions nuancées à partir de textes en langage naturel.

Les idées émises dans ce paragraphe, constituent une première approche de ces problèmes qui demeurent pour l'instant ouverts.

2.3.3.1. Décrire le portrait robot à l'aide d'images

Dans ce paragraphe, nous nous situons dans le cadre des domaines où les descriptions peuvent être illustrées par des images. Par exemple en médecine, la dermatologie où les maladies sont en général illustrées par des photos. Dans de tels domaines, une façon agréable de décrire le portrait robot serait d'utiliser les images associées aux descriptions et aux termes du domaines. Le système pourrait proposer des images à l'utilisateur comme le propose M. Crehange [CRE-88] qui choisirait certaines d'entre elles en "nuançant" ces choix.

Exemple :

L'utilisateur pourrait par exemple répondre : *mon portrait robot ressemble **un peu** à l'image 1 mais il est **très proche** de l'image 5 ...*

A partir de ce choix, le système essayerait de générer une description du portrait robot en s'appuyant sur les descriptions associées aux images choisies par l'utilisateur et en interprétant les nuances.

Les façons dont le système choisirait les images à proposer pourrait être variée :

- ce pourrait être une arborescence prédéterminée d'images, dans laquelle l'utilisateur chemine. L'arborescence pourrait correspondre à la hiérarchie des espèces,
- ce pourrait être aussi une sélection aléatoire, qui donnerait à l'utilisateur la possibilité de "feuilleter" la base comme on le ferait dans un catalogue ou un manuel,
- ce pourrait être encore une sélection guidée par la connaissance qu'a le système de

l'utilisateur, de l'utilisation, ...

Il est certain que pour cette phase une visualisation d'images à l'aide d'un imageur serait une aide extrêmement puissante, tant par la densité de visualisation que par le caractère évocateur et comparatif de la vision simultanée d'un ensemble d'images.

2.3.3.2. Générer le portrait robot et les descriptions à partir de textes en langage naturel

Il est possible d'envisager dans l'avenir d'utiliser dans les systèmes d'identification nuancée une fonction de compréhension et d'interprétation de la langue naturelle. Cette fonction permettrait d'utiliser les descriptions des phénomènes réels telles qu'elles se présentent dans les ouvrages scientifiques. Il suffirait de fournir au système une liste des caractères pouvant figurer dans les descriptions et quelques modèles de descriptions nuancées respectant le modèle de données du système. Le système pourrait alors, à la manière des systèmes d'indexation automatique, générer les descriptions nuancées. Le spécialiste n'aurait plus alors qu'à jouer un rôle de contrôle. Les thésaurus pourraient être élaborés également de manière assistée.

3. RESUME DU PROCESSUS D'IDENTIFICATION NUANCEE

En résumé, pour mettre en place un système d'identification nuancée dans un domaine d'application donné il faut effectuer les tâches suivantes :

- identifier les caractères décrivant les types (espèces) de phénomènes,
- construire ou reprendre la structure hiérarchique existant entre les types,
- déterminer le domaine de valeur de chaque caractère,
- définir les microthésaurus associés aux domaines discrets éventuels,
- faire l'inventaire des nuances utilisables pour chaque domaine de valeur et chaque caractère,
- établir la bibliothèque des fonctions caractéristiques initiales associées à chaque nuance,
- définir un jeu d'essai pertinent afin de choisir les seuils de tolérance pour les mesures Π et N,
- choisir, si le domaine d'application s'y prête, les images à associer aux descriptions

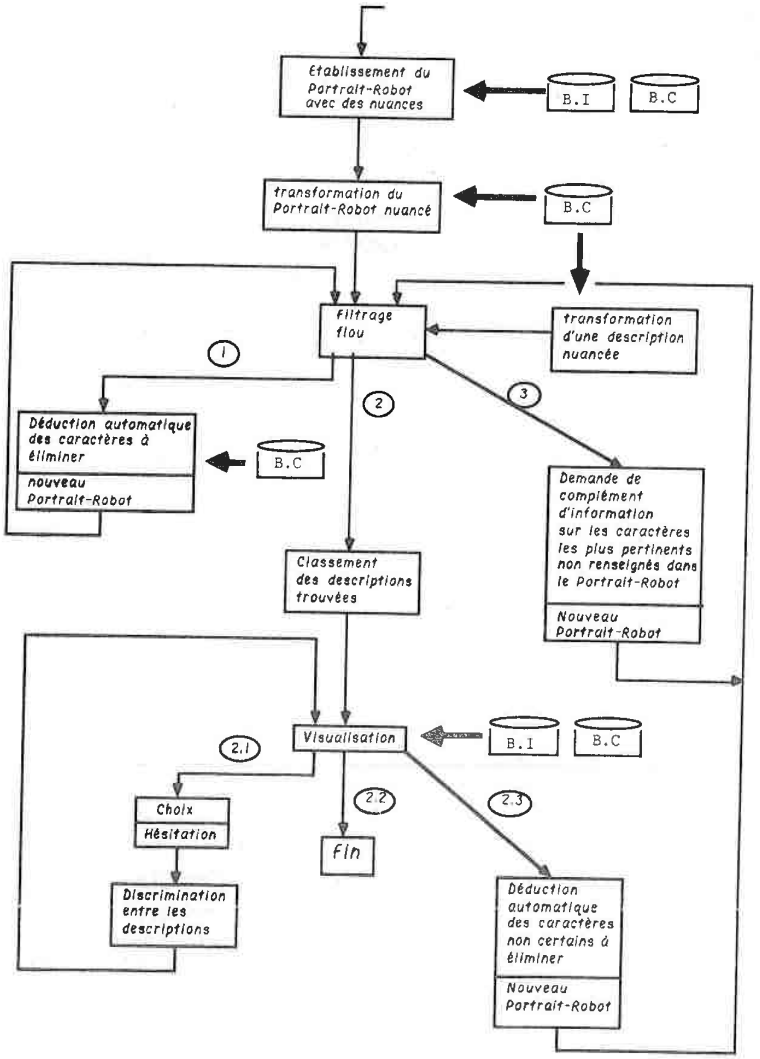
textuelles,

- déterminer les règles spécifiques au domaine d'application,
- établir les descriptions nuancées des types de phénomènes de référence,
- mettre en place un système d' "AUDIT" de façon à conserver une trace de toute interrogation du système pour aider ensuite à son amélioration.

Ces tâches sont un préalable à l'utilisation du système d'identification nuancée que nous avons réalisé et que nous présentons au chapitre suivant.

Elles pourraient être partiellement automatisées ou assistées dans un système plus général ou méta-système qui permettrait de mettre en place facilement des systèmes d'identification nuancée pour des domaines d'application divers. Une première spécification d'un tel système général a été effectuée par J.Y. Landrac [LAN-88] dans son mémoire de DEA. La poursuite de ce travail est en cours actuellement dans l'équipe EXPRIM.

La figure ci-dessous résume quant à elle le processus d'identification nuancée :



PROCESSUS D'IDENTIFICATION NUANCEE

LEGENDE :

B.I. : Base des Images.

B.C. : Base des Connaissances regroupant : le Thésaurus, la Base des Descriptions, la Base des Caractères et la Base des Nuances.

→ : enchainement des modules de traitement

⇒ : lecture dans la base de connaissances

-=-=-=-=-=-=-=-

1 : Aucune description possible n'est trouvée .

2 : Le nombre de descriptions trouvées est inférieur à un seuil donné.

3 : Un très grand nombre de descriptions possibles sont trouvées.

2.1 : L'utilisateur hésite entre deux ou plusieurs descriptions trouvées.

2.2 : L'utilisateur a reconnu son portrait-robot.

2.3 : L'utilisateur n'a reconnu son portrait-robot dans aucune des descriptions proposées.

Chapitre V REALISATION ET VALIDATION SYSTEME FIMS

Chapitre V REALISATION ET VALIDATION SYSTEME FIMS

L'approche présentée dans les chapitres précédents a été expérimentée et validée sur un système réalisé en langage LE_LISP [CHAI-86] sur une station de travail SUN 360. Nous avons baptisé ce système FIMS (Fuzzy Information Management System).

Ce système comprend pour l'instant essentiellement la fonction d'identification nuancée, les fonctions de gestion et de calcul des fonctions caractéristiques, les fonctions de calcul des mesures de possibilité et de nécessité et des fonctions simples de gestion et de manipulation de la base de connaissances.

Le domaine d'application que nous avons choisi est la mycologie. Ce domaine présente en effet la particularité d'être très riche en informations nuancées. Nous avons réutilisé une partie de la base de connaissances du système MYCOMATIC [FOU-88] réalisé auparavant dans l'équipe EXPRIM.

Dans ce chapitre, nous présentons successivement : le domaine d'application choisi, la description du système et les résultats obtenus. On trouvera en annexe de larges extraits de la base de connaissances et des programmes.

1. DOMAINE D'APPLICATION CHOISI : LA MYCOLOGIE

L'identification des genres et des espèces en Mycologie est fondée sur des critères définis à partir d'observations. Ces observations, en général, ont la particularité d'être nuancées.

Les critères utilisés en Mycologie pour l'identification sont de cinq types :

- Macroscopiques :

Observables à l'oeil nu, ce sont par exemple les dimensions, les formes, les couleurs, etc ...

- Ecologiques :

Ils définissent les conditions générales d'apparition du champignon. Ce sont la

localisation, le sol, la végétation, l'habitat et la saison.

- Organolepltiques :

Ce sont l'odeur et la saveur.

- Microscopiques :

Ils nécessitent la pratique des coupes ou des préparations et l'observation au microscope; on peut citer par exemple la structure de la chair et les pigments du revêtement.

- Macrochimiques :

Ils consistent à l'étude des réactions que provoquent certains produits chimiques sur la chair ou la cuticule du champignon étudié.

L'approche que nous avons exposée aux chapitres précédents permet de tenir compte des cinq critères ci-dessus. En effet, avec notre modèle, tout caractère appartenant à l'un des trois premiers types de critères est considéré comme de nature "visible" (cf. Chapitre 3 §5) et par conséquent utilisé dans le processus d'identification, quant aux caractères appartenant aux deux derniers types de critères, ils seront de nature dits "invisibles".

Dans notre base de connaissances actuelle, nous n'avons que des caractères de nature "visible" car le système MYCOMATIC s'adresse à des amateurs qui n'ont pas la possibilité en général d'observer des caractères microscopiques ou macrochimiques.

L'expérience des mycologues a permis de dégager les particularités suivantes sur les caractères utilisés pour décrire les espèces :

- Ils varient avec l'âge de l'espèce.
- Ils font l'objet de nombreuses exceptions même au sein de la même famille.
- Ils sont difficiles à exprimer, à tel point qu'ils sont souvent décrits et interprétés différement selon les ouvrages de mycologie.

Ces particularités ont pour conséquence une utilisation très importante des nuances dans les descriptions des espèces.

Pour une description plus complète des différents caractères mycologiques on pourra consulter l'ouvrage de Romagnesi [ROM-53].

Afin de donner au lecteur une idée de l'importance des nuances en mycologie et de l'apport de notre modèle, par rapport à un modèle de représentation classique n'utilisant que des valeurs précises, nous donnons ci-dessous pour une même espèce de champignons trois descriptions :

- une description en langue naturelle extraite d'un ouvrage de mycologie,

- la description figurant dans le système MYCOMATIC,
- la description nuancée utilisée dans FIMS.

On remarquera les précisions apportées par l'introduction des nuances; tout en notant la difficulté de rendre compte de la richesse du texte en langue naturelle.

La description en langue naturelle du Tricholoma virgatum *(extraite du livre "petit atlas des champignons" de Romagnesi [ROM-53]:*

NOM VULGAIRE : français : tricholome vergeté.

Ce Tricholome écailleux n'est pas très rare, à la fin de l'été et en automne, dans les forêts feuillues, surtout de hêtres, et de conifères mélangés, plutôt sur les sols plus ou moins acides.

Le chapeau (4 - 8 cm) est d'abord conique pointu, et conserve un mamelon assez distinct dans le type ; dans la variété sciodes (Secr.) (= Tr. murinaceum au sens de Quélet), il est convexe, surbaissé, et le mamelon n'est qu'indiqué ; il est gris brun, gris violacé ou nuancé de rosâtre, plus foncé sur le mamelon, et ailleurs rayé de longues fibres qui se détachent sur le fond un peu plus pâle ; mais dans la variété, on peut observer aussi quelques écailles apprimées. Le pied (5-10x0.4-1.2 cm) est ferme, blanc, ou en tout cas bien plus clair que le chapeau, rayé de longues fibres, sans trace de cortine.

La chair est assez ferme, blanche ou un peu teintée localement. L'odeur est insignifiante, tout au plus un peu terreuse ; mais le caractère essentiel qui distingue cette espèce de toutes ses voisines, est la saveur âcre, que, comme chez les Russules, elle manifeste après quelques moments de mastication. Les lamelles sont moyennement serrées, assez épaisses, blanches ou grisonnantes avec l'arête souvent ponctuée de noir sur les jeunes.

Spores blanches, non amyloïdes, 6.5-9x5-6.5 µ courtement elliptiques, presque arrondies lisses. Cellules stériles de l'arête des lames en massue, souvent emplies d'un suc noirâtre. Cuticule filamenteuse.

Cette espèse est à rejeter, à cause de sa saveur âcre.

La description du Tricholoma Virgatum dans le système MYCOMATIC :

TYPE DE L'APPAREIL SPORIFERE	:	Lamelles
COULEUR DE L'APPAREIL SPORIFERE	:	gris, blanc
VOLVE à la BASE DU PIED	:	NON
PRESENCE DE L'ANNEAU	:	NON
ALLURE GENERALE	:	allure 2, allure 1, allure 8, allure 13
ALLURE DES LAMELLES ou PLIS	:	émarginées ou échancrées
COULEUR DU CHAPEAU	:	gris, violacé

ALLURE DU CHAPEAU	:	conique
REVETEMENT DU CHAPEAU	:	fibrilleux, soyeux
PRESENCE DE LAIT	:	NON
SAVEUR DE LA CHAIR OU DU LAIT	:	âcre
PRESENCE D'UNE CORTINE	:	NON
CHAIR_GRENUE_CASSANTE	:	NON
ODEUR_CARACTERISTIQUE	:	faible ou nulle
CARACTERISTIQUE DE LA CHAIR	:	blanche
ALLURE DU PIED	:	cylindrique
COULEUR DU PIED	:	blanc
SAISON DE RECOLTE	:	automne
LIEU DE RECOLTE	:	conifères, feuillus
PARTICULARITE	:	saveur âcre
REACTION CHIMIQUE SPECIALE	:	sans objet
COMESTIBILITE	:	à rejeter
REFERENCES BIBLIOGRAPHIQUES	:	Rom_168b, Phi_35
CONFUSIONS POSSIBLES	:	Groupes des petits gris, Tricholoma sciodes
SYNONYMES COURANTS	:	Tricholome vergeté
IMAGES DU VIDEODISQUE	:	341, 342

La description du Tricholoma Virgatum dans notre système FIMS :

```
:champignon
((super_classe )
(def
    (lait 1 (non))
    (cortine 1 (non))
    (chair_grenue_cassante 1 (non)))

:tricholoma
((super_classe :champignon)
(def
    (volve 1 (non))
    (anneau 1 (non))
    (type_app_sporif 1 (lamelles))
```

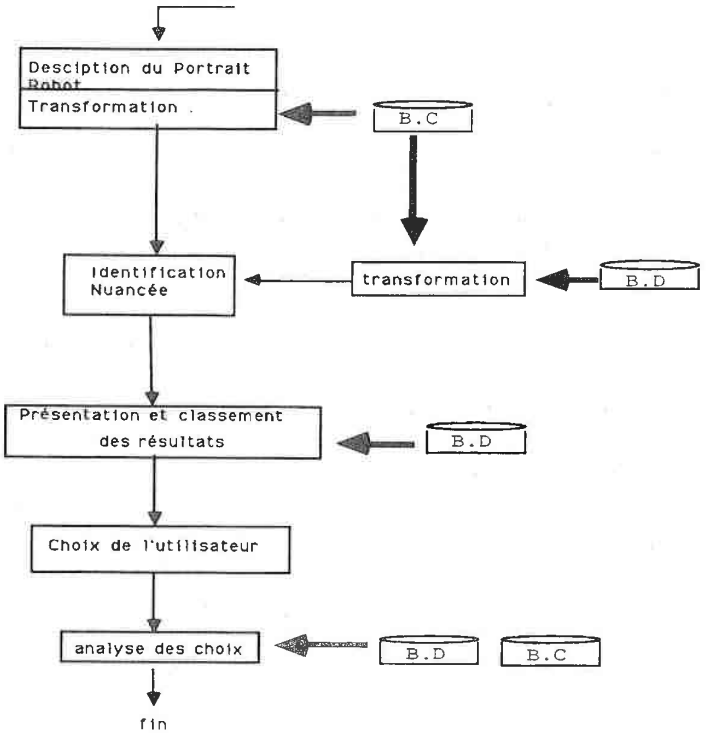


```
    (odeur_nature 0 (quelconque))
    (odeur:intensite .8 (moyenne))))

:virgatum
((super_classe :tricholoma)
(def
    (allure .8 (all_1) (all_2) (all_8) (all_13))
    (coul_app_sporif .8 (gris) (blanc))
    (diam_chapeau .9 ((4 8)))
    (coul_chapeau 1 (gris) (violace))
    (allure_chapeau .9 (conique jeune) (mamelonne toujours))
    (revet_chapeau 1 (fibrilleux) (soyeux))
    (allure_lamelles .9 (emarginees) (echancrees) (serrees moyennement) (epaisses assez) )
    (diam_pied .9 ((0.4 1.2)))
    (haut_pied .9 ((5 10)))
    (coul_pied .6 (blanc raye))
    (allure_pied .7 (cylindrique))
    (odeur:intensite 1 (faible) (nulle))
    (couleur_chair .8 (blanc) (grisatre))
    (lieu_recolte .8 (sol_acide plutot) (coniferes) (feuillus) (hetres surtout))
    (saison_recolte 1 (automne) (ete fin))
    (saveur_chair 1 (acre))
```

2. DESCRIPTION DU SYSTEME FIMS

Nous présentons ici l'architecture simplifiée du système FIMS. Sur la figure ci-dessous nous n'avons volontairement pas fait figurer le "retour arrière" (cf chapitre 4 §3) afin de ne pas surcharger le schéma et afin de ne mettre en évidence que les principaux modules. Le module réalisant le "retour arrière" sera décrit dans le paragraphe 2.2. de ce chapitre.



PROCESSUS ACTUEL

Legende :

Module de traitement

B.D Base des descriptions des espèces

B.C Base de connaissances regroupant les définitions des nuances, le thésaurus, les caractères et leurs domaines de définition.

Indique l'enchaînement des modules

Indique les lectures dans B.C. ou dans B.D.

D'un côté on a les différents modules de traitement (*Transformation, identification nuancée, etc ...*) et de l'autre côté les différentes connaissances manipulées (*B.D., B.C.*).

Les différents modules seront décrits dans le paragraphe 2.2., quant aux connaissances elles sont présentées dans le paragraphe suivant 2.1.

2.1. Représentation des connaissances manipulées

La base de connaissances (B.C.) (cf figure ci-dessus) englobe en réalité quatre autres types de connaissances : la base des nuances, la base des caractères, la base des domaines de définition des caractères et le thésaurus, que nous avons décrits au troisième chapitre.

Nous exposons dans ce paragraphe nos choix de représentation pour chacun de ces types de connaissances ainsi que pour la base de descriptions des espèces (B.D.).

Dans l'environnement de programmation choisi (LE_LISP), nous avons représenté toute connaissance intervenant dans la structure du système par un objet qui est lui même soit une classe d'objets soit une instance d'une classe.

Chaque classe d'objets est représentée par un "package" au sens LE_LISP. Cette notion permet de définir une hiérarchie entre des objets. L'accés à un "package" se fait à partir du "package" racine noté # et par une suite de noms de "package" séparés par des deux points ":".

Etant donné qu'au chapitre 3 nous avons décrit l'ensemble des classes d'objets, nous nous contenterons dans la suite de ce paragraphe de donner quelques exemples pour chacune des classes présentes dans le système.

La classe C_THESAURUS

La classe C_THESAURUS regroupe l'ensemble des thésaurus manipulés. Chaque thésaurus est réprésenté par un "package" regroupant l'ensemble de ses termes. A chaque terme est associé son type de dépendance (sémantiquement liés ou non sémantiquement liés), la liste de ses termes génériques et éventuellement celle de ses termes spécifiques, la liste de ses termes contraires et celle de ses termes voisins.

:couleur:base-blanc

((lien_sem t)

(ts (:couleur:blanc 1) (:couleur:creme .6) (:couleur:blanchâtre 1)))

:consistance_chair:fragile

((lien_sem ())

(tv (:consistance_chair:tendre .8)) (tc (:consistance_chair:tenace)))

; dans le cas des liens contraires on a fait le choix de ne pas utiliser de coefficients.

La classe C_NUANCE

Cette classe regroupe l'ensemble des définitions données aux nuances et aux valeurs vagues.

Pour toute nuance ou valeur vague, on donne son type (*LIMITE, MODIFICATEUR, TRANSFORMATION*) et sa fonction caractéristique.

La fonction caractéristique peut être soit un quadruplet soit une "lambda expression" LE_LISP. Le quadruplet représente une fonction caractéristique trapézoïdale quant à la "lambda expression" c'est une fonction LE_LISP définissant la fonction caractéristique associée à une nuance ou une valeur vague.

Illustrons ces propos par quelques définitions de nuances et de valeurs vagues tous extraits de la base des connaissances :

:environ

((type trans)

(FC (lambda (a b c d) (list (- a (/ a 20)) (+ b (/ b 20)) (c 1.5) (* d 1.5)))))*

; la fonction caractéristique associée à cette nuance est une "lambda expression" qui prend

; en paramètre une fonction caractéristique définie par le quadruplet (a b c d), et retourne

; une nouvelle fonction caractéristique dérivée de celle passée en paramètre.

:grand

((type trans) (FC (10 20 3 3))

; *grand* étant une valeur vague définie sur un domaine continu, sa définition est alors

; donnée par la fonction caractéristique trapézoïdale *(10 20 3 3)*. Cette définition dépend

; du domaine auquel la nuance s'applique ; ici il s'agit de "*grand*" appliquée à la taille du chapeau.

:très

((type mod)

(FC (lambda (a b) (list a (sqrt b)))))

; la nuance *très* est de type modificateur, la fonction caractéristique associée est une

; "lambda expression" qui prend en paramètre un couple de type *(valeur coefficient)* et qui

; retourne en résultat le couple formé de*valeur* et de la racine carrée de *coefficient*..

La classe C_DOMAINE

Chaque domaine de la classe C_DOMAINE est défini par son type (CONT pour continu ou DISC pour discret) et par sa définition propement dite. Cette définition est donnée :

- dans le cas où le domaine est de type continu, par un intervalle de valeurs,
- dans le cas où le domaine est de type discret, par le nom d'un noeud du thésaurus ou par un ensemble discret de valeurs.

:couleur

((type DISC) (def #:c_thésaurus:couleur))

; la définition du domaine *couleur* est donnée par le thésaurus des couleurs.

:allure_chapeau

((type DISC) (def (plan bombe irregulier conique))

; la définition du domaine *allure_chapeau* est donnée en extension sous forme d'une liste

; de valeurs.

:2-20

((type CONT) (def 2 20))

; lorsqu'il s'agit d'un domaine continu, on donne sa borne inférieure et sa borne

; supérieure.

La classe C_CARACT

Dans le système actuel les caractères ne sont pas hiérarchisés ; ils sont tous au même niveau (ie. ils sont "mis à plat"). La critique que l'on peut faire à ce choix est la perte des

caractères agrégés. Nous pensons remédier à cette insuffisance dans la prochaine version de notre système en permettant par exemple de définir le caractère "odeur" comme étant l'agrégation des deux sous-caractères "intensité" et "nature". Dans la version actuelle de FIMS nous avons deux caractères distincts "odeur-intensité" et "odeur-nature".

Exemples de description des caractères extraits de la base de connaissances actuelle :

```
:coul_chapeau                ; c'est le caractère couleur du chapeau
   ((DOM couleur)            ; domaine de définition est couleur
    (TYPE mu)                ; c'est un caractère multivalué
    (NUANCE (généralement surtout souvent rarement parfois))
    (POIDS 0.9)              ; poids par défaut
    (NATURE visible))

:diam_chapeau                ; c'est le caractère diamètre du chapeau
   ((DOM 2-20)               ; domaine de définition est l'intervalle [2,20]
    (TYPE mu)                ; c'est un caractère multivalué
    (NUANCE (généralement environ)
    (POIDS 0.8)              ; poids par défaut
    (NATURE visible))
```

La classe C_FAMILLE

La classe C_FAMILLE conserve toutes les descriptions d'espèces ou de familles d'espèces. La représentation pratique d'une description en LE_LISP est très proche de la structure d'une description telle qu'on l'a décrite au chapitre 3. En effet, une description est représentée par une liste contenant quatre informations : la première indique la famille d'espèces "mère", la deuxième donne la liste des caractères renseignés avec les valeurs associées, la troisième indique la liste des caractères significatifs pour la description et la quatrième contient éventuellement la liste des sous familles et des espèces si la description est une famille d'espèces. La troisième information est générée automatiquement à partir de la deuxième ainsi que la quatrième qui est générée à partir de la première pendant une phase de chargement de la base de connaissances.

Les descriptions nuancées des espèces ont été introduites à partir des descriptions du

système MYCOMATIC dans lesquelles un spécialiste a introduit des nuances à partir des textes en langue naturelle figurant dans les ouvrages de mycologie.

Nous allons donner quelques exemples de description ne contenant que les deux premières informations qui sont les seuls indispensables à la saisie de la description :

```
:amanita
   ((super_classe :champignon)
    (def     (type_app_sporif 1 (lamelles))
             (coul_app_sporif 1 (blanc))
             (volve 1 (oui))
             (anneau 1 (oui))
             (saison_recolte 1 (été) (automne))
             (allure_lamelles 1 (libres))))

:vaginata
   ((super_classe :amanita)
    (def     (diam_chapeau .8 ((4 12)))
             (coul_chapeau 1 (gris) (fauve) (blanc) (orange))
             (revet_chapeau 1 (strie))
             (anneau 1 (non))
             (diam_pied .8 ((1.5 2)))
             (haut_pied .8 ((13 20)))
             (coul_pied .6 (gris) (fauve) (blanc) (orange))
             (allure_pied .7 (elance))
             (carac_chair 1 (fragile) (mince))
             (lieu_recolte .8 (feuillus) (conifères))
             (saison_recolte 1 (printemps fin) (été) (automne))
             (image 1 ((29 31)))))
```

Remarques :

- Les familiers du LE_LISP pourront constater que tous les objets manipulés sont représentés par des "a-listes" (i.e. une liste de couples (<clé> <sexp>) où <sexp> est une S-expression LE_LISP). Cette structure offre l'avantage d'être dynamique par conséquent

facile à modifier.

- Tous les objets manipulés sont amenés en mémoire par un module de chargement. Pendant cette phase de chargement les liens des instances et des sous-classes vers leurs classes sont créés dynamiquement (mise à jour de l'attribut SUPER-CLASSE), la liste des caractères significatifs (l'attribut LCS) des descriptions est générée automatiquement. Finalement les liens entre les termes génériques et les termes spécifiques du thésaurus sont également établis.

2.2. Modules de traitement

Nous donnons dans ce paragraphe le principe des différents modules du processus global plus les fonctions LE_LISP qui les réalisent.

Nous commençons d'abord par les principaux modules, à savoir :

- le module de transformation des informations nuancées,
- le module d'identification nuancée,

puis nous terminons par un résumé des modules assurant le "retour arrière".

Nous nous contenterons dans ce paragraphe de donner une description des fonctions LE_LISP, les programmes sources sont donnés en annexe 2.

2.2.1. Module de transformation des informations nuancées

Le but du module de transformation des informations nuancées appelé F_TRANSFORME est de remplacer toute information nuancée, présente dans une description d'espèce ou de portrait robot, par la fonction caractéristique adéquate.

Le passage des informations nuancées aux fonctions caractéristiques repose sur les méthodes de calcul et de génération des fonctions caractéristiques décrites au chapitre 3.

Le principe de ce module est de traiter successivement chaque caractère de la description décrit sous la forme d'un triplet (<nom caractère> <poids> LVN) où LVN est une liste d'informations nuancées sous forme de couples (<valeur> <nuance>). Le traitement d'un caractère fait appel aux deux modules F_GENERE_DISCRET et F_GENERE_CONTINU suivant que le type du domaine du caractère est respectivement discret ou continu.

Dans le cas où le domaine est représenté par un thésaurus, on a réalisé une fonction LE_LISP F_GENERE_RS qui permet de générer automatiquement la fonction caractéristique associé à un terme du thésaurus.

On décrit ci-dessous les fonctions LE_LISP qui réalisent le mécanisme de transformation.

(F_GENERE_CONTINU caractere)
; CARACTERE : (Nom_caract poids (Val1 Nuance1) ... (Valn Nuancen))
; BUT : Cette fonction génère la fonction caractéristique associée au caractère passé en
; paramètre et défini sur un domaine continu.
; RESULTAT : (Nom_caract poids (BI1 BS1 PG1 PD1) .... (BIn BSn PGn PDn))

(F_GENERE_DISCRET caractere)
; CARACTERE : (Nom_caractere poids (Val1 Nuance1) ... (Valn Nuancen)).
; BUT : générer la fonction caractéristique associée à un caractère de domaine discret.
; RESULTAT : une liste de la forme (Nom_caractere poids (Val1 Coef1) .. (Valq coefq))

(F_GENRE_RS terme)
; TERME : terme du thésaurus dont on veut générer la fonction caractéristique.
; BUT : génère la fonction caractéristique associée au terme passé en paramètre.
; RESULTAT : une liste de couples (<terme> <coefficient>).

(F_NUANCE fc nuance l_nuance)
; FC : cas continu -> c'est un quadruplet (a b c d)
; cas discret -> c'est un couple (terme coef)
; NUANCE : soit Nil (pas de nuance) soit le nom de la nuance donné dans le p. robot.
; L_NUANCE : c'est la liste définie au niveau d'un caractère qui associe au nom de la
; nuance dans le portrait robot sa nuance synonyme pour laquelle le système
; possède une fonction caractéristique.
; BUT : transformer la fonction caractéristique FC en lui appliquant la fonction
; caractéristique de la nuance passée en paramètre.
; RESULTAT : de même type que le paramètre FC.

(F_TRANSFORME L_pr)

; L_PR : ((<nom_caractere_1> <pds1> (<val1> <nuance1>) ... (<valn> <nuancen>))
; .....
; (<nom_caractere_p> <pdsp> (<val1> <nuance1>) ...(<valn> <nuancen>)))
; BUT : remplace tous les termes nuancés du Portrait Robot L_PR par les fonctions
; caractéristiques correspondantes.
; RESULTAT : une liste de la forme :
; ((<nom_caractere_1> <pds1> <Fc1>) .... (<nom_caractere_p> <pdsp> <Fcp>))
; ou` <Fci> est
; - (<val1> <coef1>) ... (<valn> <coefn>) si Domaine du caractere est DISCRET
; - (<bi1> <bs1> <pg1> <pd1>) .. (idem) si domaine du caractere est CONTINU

2.2.2. Module d'identification nuancée

Ce module qui constitue en fait le noyau du système gère le module précédent, calcule les mesures de possibilité et de nécessité, assure les processus de filtrage hiérarchique et élémentaire et gère la trace des descriptions d'espèces pour lesquels on a trouvé une mesure de possibilité nulle après le filtrage élémentaire. La consultation de cette trace permettra à l'utilisateur de comprendre pourquoi certaines espèces ont été éliminées.

2.2.2.1.Module de calcul des mesures de possibilié et de nécessité

L'objectif de ce module est de calculer les mesures de possibilité et de nécessité entre deux fonctions caractéristiques FC1 et FC2. Pour cela, on a distingué deux situations selon que le domaine de définition des fonctions caractéristiques est discret ou continu. Dans le premier cas, nos calculs portent sur des listes de couples (<valeur> <coefficient>) et dans le second cas sur des quadruplets.

Les fonctions LE-LISP que nous avons développées pour réaliser les différents calculs et et dont nous donnons une description ci-dessous, sont utilisables pour n'importe quelle application nécessitant la gestion d'informations nuancées.

(F_POSS_CONTINU fc1 fc2)

; FC1, FC2 : sont 2 quadruplets représentant des fonctions caractéristiques trapézoïdales.
; BUT : calculer la mesure de possibilité entre les deux fonctions FC1 et FC2.
; RESULTAT : est la mesure de possibilité entre FC1 et FC2.

(F_NESS_CONTINU fc1 fc2)

; FC1, FC2 : sont 2 quadruplets représentant des fonctions caractéristiques trapézoïdales.
; BUT : calculer la mesure de nécessité entre les deux fonctions FC1 et FC2 .
; RESULTAT : est la mesure de nécessité entre FC1 et FC2.

(F_POSS_DISCRET fc1 fc2)

; FC1, FC2 : sont 2 listes de couples (<valeur> <coefficient>) représentant des fonctions
; caractéristiques de 2 valeurs vagues définies sur des domaines discrets.
; BUT : calculer la mesure de possibilité entre les deux fonctions FC1 et FC2
; RESULTAT : est la mesure de possibilité entre FC1 et FC2.

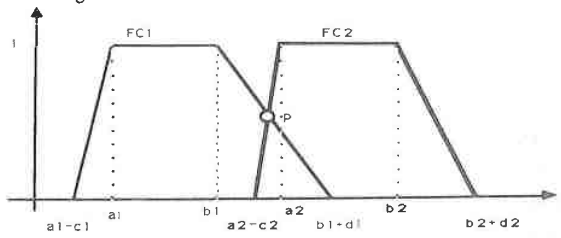
(F_NESS_DISCRET fc1 fc2)

; FC1, FC2 : sont 2 listes de couples (<valeur> <coefficient>) représentant des fonctions
; caractéristiques de 2 valeurs vagues définies sur des domaines discrets.
; BUT : calculer la mesure de nécessité entre les deux fonctions FC1 et FC2
; RESULTAT : est la mesure de nécessité entre FC1 et FC2.

Note 1 :

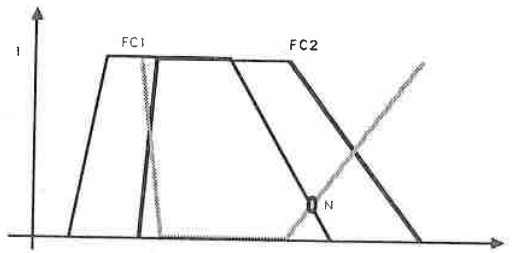
Les fonctions F_POSS_CONTINU et F_NESS_CONTINU ont pour arguments des quadruplets. Soient : FC1=(a1 b1 c1 d1) et FC2=(a2 b2 c2 d2).

Dans le cas de figure suivant :



(F_POSS_CONTINU FC1 FC2) retourne l'ordonnée du point P, soit :
Max (0,1 - (a2-b1)/(c2+d1)).

Dans le cas de figure suivant :



(F_NESS_CONTINU FC1 FC2) retourne l'ordonnée du point N, soit :
Min (1,Max (0, (b1-b2+d1)/(d1+d2))

Note 2 :

Les fonctions F_POSS_DISCRET et F_NESS_DISCRET ont pour arguments des listes de couples. Soient : FC1=((a_1 $coef^1_1$) .. (a_n $coef^1_n$)) et FC2= ((b_1 $coef^2_1$) .. (b_p $coef^2_p$)).

(F_POSS_DISCRET FC1 FC2) calcule :

Sup_k Min ($coef^1_k$,$coef^2_k$)

(F_NESS_DISCRET FC1 FC2) calcule :

Inf_k Max ($coef^1_k$, 1-$coef^2_k$)

2.2.2.2. Modules de filtrage hiérarchique et élémentaire

Les processus de filtrage hiérarchique et élémentaire constituent les principales fonctions de ce module.

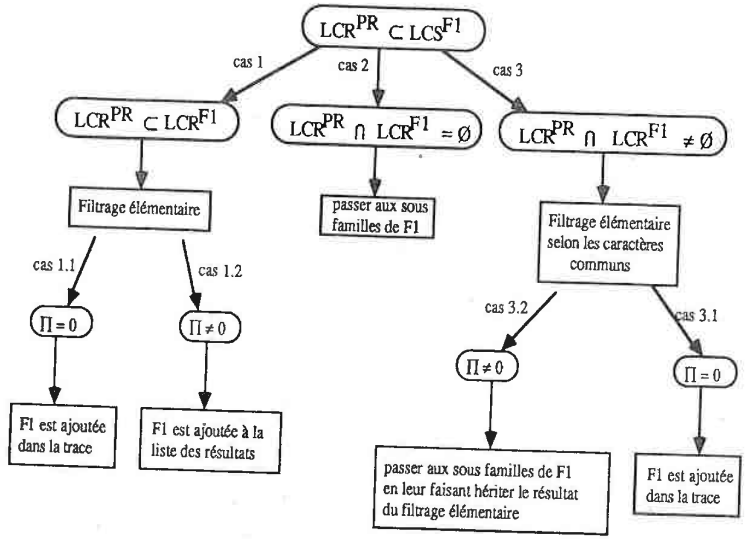
Le premier permet, à partir de la description du portrait robot et d'une liste LF de descriptions d'espèces ou de familles d'espèces, de filtrer toute description F de la liste LF telle que la liste des caractères renseignés dans le portrait robot soit incluse dans la liste des caractères renseignés dans la description F. Le but de ce procédé est d'éliminer des branches

de la hiérarchie des descriptions. Au cours de ce filtrage les mesures de possibilité et de nécessité sont calculées pour les caractères renseignés au niveau de la description F. En cas de "surcharge", au niveau d'une sous famille F' de F, c'est à dire qu'il existe parmi les caractères renseignés dans F' des caractères déjà renseignés dans F (i.e. F' est une exception dans la famille F), les mesures de possibilité et de nécessité sont recalculées en fonction des nouvelles valeurs figurant dans F'.

Quant au processus de filtrage élémentaire, il effectue des comparaisons entre les valeurs associées aux caractères du portrait robot et de ceux des descriptions issues du filtrage hiérarchique, puis calcule les mesures de possibilité et de nécessité.

Le schéma ci-dessous explique l'imbrication du filtrage hiérarchique et du filtrage élémentaire :

- Soit PR le portrait robot et soit LCR^{PR} la liste des noms des caractères décrits dans PR.
- Soit F1 une description de famille ou d'espèce décrite par :
  - . LCS^{F1} : la liste des noms des caractères ayant un sens pour cette description. Cette liste est l'union de tous les noms des caractères décrits dans les sous-familles.
  - . LCR^{F1} : la liste des noms des caractères valorisés/communs à toutes les sous-familles ou instances de F1.
  - . DEF^{F1} : la liste des poids et des valeurs des caractères de LCR^{F1}.



IMBRICATION DU FILTRAGE HIERARCHIQUE ET DU FILTRAGE ELEMENTAIRE

Legende :

- ***cas 1*** : tout caractère de PR est renseigné dans F1. On procède alors au filtrage élémentaire qui retourne une mesure de possibilité Π et une mesure de nécessité N associées à F1. Aucun autre filtrage hiérarchique n'est effectué dans la hiérarchie de F1.

  . ***cas 1.1*** : la mesure de possibilité est nulle ; il existe parmi les caractères comparés un ou plusieurs ayant une mesure de possibilité nulle. Dans ce cas on ajoute F1 dans la trace.

  . ***cas 1.2*** : tous les caractères comparés ont une mesure de possibilité non nulle. F1 est ajoutée à la liste des résultats.

- ***cas 2*** : aucun caractère de PR n'est renseigné dans F1. Sachant à priori que les

caractères de PR ont un sens pour F1 ($LCR^{PR} \subset LCS^{F1}$) ; c'est à dire que tous les caractères de PR sont renseignés au niveau des sous familles de F1, on procède alors au filtrage hiérarchique parmi les sous familles de F1.

- ***cas 3*** : quelques caractères de PR sont renseignés dans F1. On compare les valeurs des caractères communs à PR et F1 à l'aide d'un filtrage élémentaire qui retourne une mesure de possibilité Π et une mesure de nécessite N.

  . ***cas 3.1*** : idem le cas 1.1

  . ***cas 3.2*** : tous les caractères communs à PR et à F1 ont une mesure de possibilité non nulle. Dans ce cas, on poursuit le filtrage hiérarchique parmi les sous familles de F1 jusqu'à ce que tous les caractères de PR soient comparés avec ceux des sous familles de F1.Toute sous famille F' de F1 hérite le résultat du filtrage élémentaire entre les caractères communs à PR et F1.

La liste des fonctions LE_LISP ci-dessous illustre la réalisation pratique de ce module.

(F_IDENTIFIE pr lf seuil1 seuil2)

; PR : Portrait robot sous forme d'une liste de couple
; (<nom_caractere> (<valeur1> <nuance>) ... (<valeurp> <nuance>))
; LF : Liste de noms d'objets de la hiérarchie des descriptions à partir desquels on
; commence l'identification. EX: (#:c_famille:amanita) cela veut dire que
; l'identification ne se fera que parmi les amanites.
; SEUIL1, SEUIL2 : seules les espèces ayant une mesure de possibilité >seuil1
; et une mesure de nécessité >seuil2 seront acceptées
; BUT : identification nuancée.
; RESULTAT : la liste des descriptions classées suivant l'ordre de Pareto.

(F_FILTRE_HIERARCHIQUE pr lf lcarcalc)

; PR : portrait robot .
; LF : liste des noms de familles ou d'espèces qu'on va comparer au PR.
; LCARCAlC: liste des caractères déja calculés (dans les super_classes des familles de lf),
; les éléments de LCARACT sont de la forme (<car> <pds> <poss> <ness>),
; <pds> est le poids qui apparait dans les super_classes).
; RESULTAT : une liste de triplets (<nom_famille/espe`ce> <poss> <ness>)

(F_FILTRE_ELEMENTAIRE probot espece)
; PROBOT : ((nom_caractere1 poids1 (val11 nuance11) ... (val1n nuance1n))
; .....
; (nom_caracterep poidsp (valp1 nuancep1) ... (valpq nuancepq)))
; ESPECE : (nom_espece <idem probot>).
; BUT : comparer le portrait robot et l'espèce.
; RESULTAT : (TRACE nom_espece poss ness)
; où TRACE : ((nom_caractere pds poss ness) ...)

2.2.3. Autres modules

- Le module *présentation et classement des résultats* permet simplement d'ordonner les espèces trouvées suivant l'ordre de Pareto (cf chapitre 4 §2.3.2.1.) puis de les présenter à l'utilisateur.

- Le module *choix de l'utilisateur* : permet à l'utilisateur de choisir parmi les espèces proposées celles qui ressemblent le plus à son exemplaire à identifier. C'est dans ce module que seraient utilisées les images. Remarquons, qu'il peut, comme dans MYCOMATIC, refuser toutes les propositions du système ou en retenir plusieurs.

- Le module *Analyse des choix* permet en cas d'échec (toutes espèces propsées sont refusées) d'effectuer un "retour arrière" grâce à la trace construite pendant le module d'identification nuancée. Cette trace contient la liste des espèces qui ont été éliminées et pour chacune d'elles, la liste des mesures de possibilité et de nécessité calculées pour chaque caractère. Les espèces y sont ordonnées par ordre croissant du nombre de mesures de possibilité nulles. De cet ordre et des poids de confiance donnés par l'utilisateur dans le portrait robot, il est possible de déterminer les caractères qui ont peut être conduits à l'élimination d'espèces à tort. La méthode consiste alors à faire abstraction progressivement de ces caractères afin de proposer d'autres espèces.

Les différentes possibilités de "retour arrière" décrites en fin du chapitre 4 §3 ne sont pas toutes intégrées au système actuel. Nos efforts ont porté, jusqu'à présent, plus sur la construction d'une base de connaissances cohérente et complète et sur le choix de jeux d'essai afin de tester nos différentes heuristiques et formules développées tout au long de cette thèse.

3. RESULTATS

Le système a été testé par un spécialiste du domaine. Celui-ci a fourni des portraits robots volontairement vagues et incomplets correspondant à des espèces précises.

Pour chaque portrait robot, le spécialiste a établi à l'avance la liste des espèces pouvant lui correspondre et a comparé cette liste à celle trouvée par le système FIMS.

Enfin, le même portrait robot a été soumis au système MYCOMATIC afin de comparer les résultats obtenus par les deux systèmes.

Lorsque la liste préétablie par le spécialiste ne correspondait pas à celle donnée par le système, nous avons cherché à l'aide de la trace de calcul des fonctions caractéristiques et des différents coefficients d'où provenaient les différences. Elles étaient le plus souvent dues à des oublis ou à des erreurs dans le thésaurus ou à des poids d'importance mal adaptés.

3.1. Exemples d'exécution

On donne ci-dessous deux exemples d'identification nuancée, on en trouvera d'autres en annexe.

3.1.1. Exemple 1

- PORTRAIT ROBOT EN LANGUE NATURELLE

"J'ai ramassé un champignon jaunâtre qui sent très mauvais, sous des sapins,".

- RESULTAT PREVU PAR LE SPECIALISTE

Tricholoma Saponaceum ou *Tricholoma Sulfureum*,
d'après le contenu de la base de connaissances de FIMS.

- DETERMINATION A L'AIDE DE MYCOMATIC

MYCOMATIC permet de rentrer le portrait robot en choisissant les caractères pour lesquels l'utilisateur désire donner des informations. Pour chacun de ces caractères, MYCOMATIC propose de choisir dans une liste de termes génériques. Ainsi, la description du portrait robot dans MYCOMATIC est donnée sous la forme :

COULEUR CHAPEAU : *base-de-jaune*
ODEUR : *désagreable*
LIEU DE RECOLTE : *conifères ou résineux*

MYCOMATIC contenant 210 espèces de champignons, trouve une cinquantaine d'espèces pouvant correspondre à ce portrait robot. Parmi celles-ci, seules nous intéressent les espèces figurant dans la base de connaissances de FIMS soit :

Tricholoma Rutilans, Tricholoma Equestre,
Tricholoma Sulfureum, Tricholoma Saponaceum.

- IDENTIFICATION A L'AIDE DE FIMS :

Le modèle proposé par notre système FIMS (cf. chapitre 4 §5) permet de décrire le portrait robot de la façon suivante :

((COUL_CHAPEAU 1 (jaunatre))
(ODEUR_NATURE .8 (mauvais très))
(LIEU_RECOLTE 1 (sapins)))

Voici un extrait d'exécution du processus d'identification de FIMS :

3.1.2. Exemple 2

- PORTRAIT ROBOT EN LANGUE NATURELLE

"J'ai ramassé, fin novembre sous des conifères, en montagne, un champignon ayant un chapeau grisâtre, rouge et écailleux, un pied d'environ 6 cm et n'ayant pas de volve".

- RESULTAT PREVU PAR LE SPECIALISTE

Tricholoma Pardinum ou Tricholoma Imbricatum,
d'après le contenu de la base de connaissances de FIMS.

- DETERMINATION A L'AIDE DE MYCOMATIC

Description du portrait robot dans MYCOMATIC :

COULEUR CHAPEAU : *base-de-rouge, base-de-gris*
LIEU DE RECOLTE : *conifères*
VOLVE : *non*
REVETEMENT DU CHAPEAU : *rugueux, accidenté*
SAISON DE RECOLTE : *automne, hiver*

Résultats retournés par MYCOMATIC :

Tricholoma Aurantium, Tricholoma Imbricatum, Tricholoma Scalpturatum,
Tricholoma Sciodes, Tricholoma Album, Tricholoma Vaccinum,
Tricholoma Rutilans, Tricholoma Pardinum, Tricholoma Terreum.

- IDENTIFICATION A L'AIDE DE FIMS :

Description du portrait robot dans FIMS :

((COUL_CHAPEAU 1 (grisatre) (rouge))
(VOLVE 1 (non))
(HAUT_PIED .6 (6 environ))
(REVET_CHAPEAU .8 (ecailleux))
(SAISON_RECOLTE 1 (novembre fin))
(LIEU_RECOLTE 1 (coniferes) (montagne)))

Nous présentons ci-dessous deux extraits d'exécution du processus d'identification de FIMS ; le premier donne toutes les espèces obtenues et le deuxième ne montre que les espèces ayant une mesure de possibilité supérieure à 0.3 :

3.1.3. Commentaires

On fera les remarques suivantes sur le processus de détermination de MYCOMATIC :

- MYCOMATIC impose de décrire le portrait robot à l'aide de termes génériques exclusivement, obligeant ainsi l'utilisateur à remplacer les termes spécifiques par des termes génériques ce qui n'est pas toujours évident,
- MYCOMATIC ne donne pas la possibilité à l'utilisateur d'accorder des poids de confiance aux informations qu'il a fournies,
- MYCOMATIC ne donne pas le classement des résultats,
- MYCOMATIC ne permet pas d'exprimer les nuances.

En revanche, FIMS apporte une amélioration sur tous ces points. L'utilisateur final appréciera en particulier le classement des espèces même si parfois il n'est pas très significatif.

3.2. Conclusion

Les différents tests effectués jusqu'à présent en présence d'un spécialiste, nous ont permis de porter un premier jugement, sur la validité de notre approche, qui est fort encourageant et qui va dans le sens des résultats attendus de nos diverses heuristiques et formules. Grâce à des discussions avec ce spécialiste, nous avons déja apporté certaines améliorations à notre système ; la principale de ces améliorations a été de modifier le calcul de la mesure de nécessité pour la raison suivante : nous avions introduit la mesure de nécessité comme étant le degré d'inclusion de la description de l'espèce dans celle du portrait robot. Pour obtenir, avec cette méthode, l'espèce en résultat avec une mesure de nécessité non nulle, il faudrait que la description du portait robot englobe celle de l'espèce, ce qui n'est que très rarement le cas puisque la description de l'espèce est toujours plus riche que celle du portrait robot. Pour remédier à ce problème nous avons calculé la mesure de nécessité comme le degré d'inclusion de la description du portrait robot dans celle de l'espèce et non l'inverse.

Cependant l'expérience a montré lors de la mise au point du logiciel MYCOMATIC qu'il était absolument nécessaire d'effectuer les tests sur ce type de système en situation réelle ; c'est-à-dire avec des utilisateurs amateurs et dans le cadre d'une exposition de champignons. Seul ce type de test, que nous n'avons pas pu encore matériellement réaliser, permettra une

mise au point plus fine des heuristiques, des différents coefficients du thésaurus et des poids attachés aux caractères.

4. CONCLUSION

Le système FIMS a été réalisé indépendamment du domaine d'application afin de pouvoir l'appliquer à d'autres domaines que la Mycologie.

La programmation de FIMS a été faite en langage LE_LISP version 15.2, ce qui facilite sa portabilité sur toute machine possédant ce langage.

Cependant, la réalisation d'un système d'identification demeure délicate à cause des difficultés rencontrées pour constituer la base de connaissances en particulier pour établir les définitions des fonctions caractéristiques associées aux informations nuancées, les poids attachés aux caractères et les poids des liens sur le thésaurus.

Pour remédier à ces difficultés et améliorer notre système, nous pensons réaliser des fonctions pour guider l'utilisateur à modifier ou à définir de nouvelles fonctions caractéristiques. Ces fonctions seraient fondées sur des techniques d'apprentissage et sur des heuristiques.

CONCLUSION

Dans cette thèse, nous avons proposé une solution globale au problème de l'identification d'un phénomène mal défini dans un domaine décrit par des connaissances nuancées.

Cette solution comprend trois éléments :

- un modèle de représentation des connaissances nuancées,
- une méthode de détermination des objets ressemblant au phénomène à identifier,
- un processus d'identification dans un système possédant une base de données multimédia.

LE MODELE DE REPRESENTATION :

Il présente les particularités suivantes :

- une ou plusieurs nuances, exprimées en langue naturelle, peuvent être associées à chaque valeur prise par un caractère d'un objet,
- les caractères ou les attributs des objets peuvent être monovalués ou multivalués,
- les domaines des caractères peuvent être des intervalles ou des ensembles finis de type quelconque,
- à chaque domaine discret peut être associé un micro-thésaurus dans lequel existent des liens divers (généricité, synonymie, opposition) entre les termes. Ces liens peuvent être munis de coefficients exprimant certaines "distances sémantiques" entre les termes,
- des poids d'importance ou de confiance peuvent être associés à chaque caractère aussi bien dans la description des objets de référence que dans la description du phénomène à identifier.

LA METHODE D'IDENTIFICATION :

Elle repose sur la théorie des possibilités. Cependant nous avons amélioré l'application de cette théorie sur deux plans :

- diminution très importante du nombre de fonctions caractéristiques à fournir par le spécialiste du domaine grâce à l'introduction d'heuristiques permettant soit de les générer à

partir des micro-thésaurus soit de les calculer à partir d'autres déjà définies par composition ou par transformation,

- possibilité d'exprimer certaines connaissances complexes sous la forme de règles prises en compte dans le calcul final des mesures de possibilité et de nécessité.

LE PROCESSUS D'IDENTIFICATION :

Le processus d'identification est du type du processus défini dans le projet EXPRIM. Il permet une identification interactive et progressive au cours de laquelle alternent des phases de filtrage, d'affichage de résultats, d'observation d'images et de consultation de textes. En cas d'échec, nous proposons une stratégie de "retour-arrière" qui s'appuie sur les poids des caractères.

REALISATION : SYSTEME FIMS

Le système FIMS réalisé a eu pour objectif principal de valider le modèle et la méthode. Le choix du langage LE-LISP a permis une représentation hiérarchisée des connaissances sous une forme très proche de leur modélisation conceptuelle. D'autre part sa souplesse d'utilisation a facilité la mise au point des heuristiques.

Nous avons eu la chance de pouvoir valider nos propositions sur un domaine réel riche en nuances, la Mycologie, pour lequel préexistait dans l'équipe un autre logiciel d'aide à la détermination; les nuances y étaient peu prises en compte. Nous avons pu ainsi mettre en évidence concrètement les avantages apportés par notre solution par rapport à la solution précédente. Ces avantages sont :

- l'expression des nuances en "langue naturelle" dans la description des espèces,
- la possibilité offerte à l'utilisateur de décrire son exemplaire à identifier avec des termes vagues et des nuances,
- le classement par proximité décroissante des espèces proposées par le système,
- l'amélioration de la phase de "retour-arrière" par la prise en compte des pondérations sur les caractères.

Les résultats obtenus sont très satisfaisants et ont été contrôlés par un spécialiste du domaine.

ORIGINALITE DU TRAVAIL :

En résumé, notre solution apporte, vis à vis des systèmes antérieurs (cf. chapitre 2) permettant une gestion des informations nuancées, les améliorations suivantes :

- expression des nuances en langue naturelle aussi bien dans les éléments de référence que dans les phénomènes à identifier,
- génération automatique de nombreuses fonctions caractéristiques grâce à des heuristiques et au thésaurus,
- possibilité d'exprimer certaines connaissances complexes sous la forme de règles prises en compte dans le calcul final des mesures de possibilité et de nécessité,
- pondération des caractères descriptifs du phénomène à identifier et des éléments de référence, permettant une identification progressive et interactive,
- expérimentation sur une application réelle très riche en nuances.

PERSPECTIVES

Comme l'a montré la synthèse bibliographique du chapitre 2, ce travail peut avoir des prolongements et des applications aussi bien dans le domaine des bases des données que dans celui de l'intelligence artificielle.

Dans le domaine des bases données il serait intéressant d'introduire le modèle de représentation des informations nuancées dans un prototype de SGBD reposant sur un langage à objets.

Dans le domaine de l'intelligence artificielle, c'est plutôt l'aspect introduction de règles dans les méthodes de calculs des mesures de possibilité et de nécessité qui mériterait d'être inclus dans un générateur de systèmes experts.

Dans un avenir proche, nous nous proposons de continuer ce travail dans les deux voies suivantes :

- poursuite de l'étude de la sémantique des nuances pour affiner la typologie donnée au chapitre 1, par exemple les nuances temporelles (*en vieillissant, jeune*),
- poursuite de la validation sur de nouveaux domaines d'application en particulier dans les systèmes d'aide à la décision dans le domaine de la gestion des entreprises,
- gestion indépendante de la base de règles,

- introduction de la méthode d'identification nuancée dans un méta-système d'aide à l'identification de phénomènes par le texte et l'image réalisé dans un langage à objets.

A long terme nous pensons étendre le système sur les deux points décrits au chapitre 4 :

- la description du portrait robot à l'aide d'images,
- la génération assistée des descriptions des phénomènes et du portrait robot à partir de textes en langue naturelle.

REFERENCES

[ABI-87] ABITEBOUL S., GRUMBACH S.
"Bases de données et objets structurés".
TSI, vol. 6, n° 5, pp. 387-404, 1987.

[ADL-82] ADLASSNIG K.P., KOLARZ G.
"CADIAG-2, Computer-assisted medical diagnosis using fuzzy subsets".
In : Approximate Reasoning in Decision Analysis, (M.N. Gupta, E. Sanchez, eds)
North-Holland, pp. 219-247, 1982.

[AND-86] ANDRES V., DUBOIS D., PRADE H., TESTEMALE C.
"Notion d'importance relative dans les requêtes à une base de données imprécises.
Expérimentation sur micro-ordinateur".
Actes journées AFCET Bases de données "SGBD sur micro-ordinateurs", La Rochelle, 1986.

[ANS-75] ANSI / SPARC
"Study group on data base management systems : interim report".
FDT 7:2, ACM, New-York, 1975.

[BAL-83a] BALDWIN J.F.
"A Fuzzy relational inference language for expert Systems".
Proc. 13th IEEE Int. Symp. on multiple-valued Logic, Kyoto, Japan, pp. 416-423, 1983.

[BAL-83b] BALDWIN J.F.
"Knowledge engineering using a fuzzy relational inference language".
Proc. IFAC Symposium on Fuzzy Information, Knowledge Representation & Decision
Analysis, Marseille, pp. 15-20, july 19-21, 1983

[BAN-88] BANCILHON F.
"Object-Oriented Database Systems".
ACM Symposium on Principles of Database Systems. Austin,Texas, pp. 1-11, Mars 1988.

[BON-84] BONNET A.
"L'intelligence artificielle. Promesses et Réalités".
Inter Editions, 1984

[BOS-85] BOSC P. et al
"Une extension de SEQUEL pour permettre l'interrogation floue"
Actes AFCET, Dijon, 14-15 Nov. 1985.

[BOS-87] BOSC P. et al.
"Fuzzy querying with SQL : extensions and implementation aspects"
Fuzzy sets and systems., 1988

[BUC-82a] BUCKLES B.P., PETRY F.E.
"A fuzzy representation of data for relational databases".
Fuzzy sets and systems, vol. 7, pp. 213-226, 1982.

[BUC-82b] BUCKLES B.P., PETRY F.E.
"Fuzzy databases and their applications".
Fuzzy information and decision Process. (M.M. Gopta, E. Sanchez, E.D.S.)
North-Holland, pp. 361-371, 1982.

[BUC-83] BUCKLES B.P., PETRY F.E.
"Extension of the fuzzy database with fuzzy arithmetic".
Proc. IFAC Symposium, Fuzzy information, Knowledge Representation & Decision Analysis, Marseille, pp. 409-414, July 19-21, 1983.

[BUC-84] BUCHANAN B.G., SHORTLIFFE E.H.
"Rule-Based Expert Systems : The MYCIN Experiments of the Stanford Heuristic Programming Project".
Addison-Wesley, Reading, 1984.

[BUI-85] BUISSON J.C., FARRENY H., PRADE H.
"Un système expert en diabétologie accessible par minitel"
5ème journées ADI-AFCET Systèmes Experts, Avignon, Mai 1985.

[BUI-87a] BUISSON J.C., FARRENY H., PRADE H.
"Dealing with imprecision and uncertainty in the expert system DIABETO III".
Univ. P. Sabatier, Toulouse, Rapport L.S.I., n° 266, pp. 26-42, février 1987

[BUI-87b] BUISSON J.C.
"TOULMED, un générateur de systèmes-experts qui prend en compte l'imprécision et l'incertitude des connaissances".
Thèse soutenue à l'INP, Toulouse, Novembre 1987.

[CAL-83] Mc CALLA G., CERCONE N.
"Approaches to knowledge representation"
IEEE Computer Society, October 1983.

[CAY-82] CAYROL M., FARRENY H., PRADE H.
"Fuzzy pattern matching".
Kybernetes vol. 11, pp. 103-116, 1982.

[CHA-83] CHARNIACK, RIESBECK, Mc DERMOTT
"Artificial Intelligence Programming".
Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1983.

[CHAI-86] CHAILLOUX J. et al
" Le manuel de référence de LE_LISP version 15.2 de l'INRIA".
INRIA Domaine de Voluceau Rocquencourt 78 153 Le Chesnay Cedex, Novembre 1986.

[CHE-76] CHEN P.P.
"The entity-relationship model : toward a unified view of data".
Transaction Database Systems, ACM, vol. 1, n°1, pp. 9-36, 1976.

[CLO-84] MARTIN-CLOUAIRE R.
"Une approche système-expert et théorie des possibilités appliquées en géologie pétrolière".
Thèse de 3e cycle, Univ. P. Sabatier, Toulouse, octobre 1984

[CLO-85] MARTIN-CLOUAIRE R., PRADE H.
"SPII-1, un moteur d'inférences simple capable de traiter des informations imprécises ou incertaines".
Actes cognitiva 85, Paris, juin 1985

[COD-79] CODD E.F.
"Extending the database relational model to capture more meaning".
ACM, vol. 4, n° 4,pp. 397, 1979.

[CRE-86] CREHANGE M. et al
"Les structures de données dans le projet EXPRIM".
Congrès INFORSID, Fontevraud, mai 1986.

[CRE-88] CREHANGE M.
"Bases d'images et Intelligence Artificielle".
Dans "Images et Vidéodisque", Documentation Française, DBMIST, 1988.

[CRE-89] CREHANGE M, HALIN G.
"Image progressive retrieval from a videodisk : a machine learning problem".
SPIE Congress, Orlando (Floride), March 1989.

[CUP-87] CUPPENS F., DEMOLOMBE R.
"Cooperative answering : a methodology to provide intelligent access to databases".
Technical Report, ONERA-CERT, 1987.

[CUP-88] CUPPENS F., DEMOLOMBE R.
"Comment reconnaître les centres d'intérêts pour fournir des réponses coopératives".
Actes BD3, pp. 257-278, Benodet, 1988.

[DAT-77] DATE D.J.
"An Introduction to Data Bases Systems".
Addison. wesley, 1977.

[DAV-89] DAVID A., THIERY O., CREHANGE M.
"Intelligent Hypermedia in Education".
ICCAL (International Conférence on Computer. Assisted learning), Dallas, USA, Mai 1989.

[DED-76] DE DOMBAL F.I., GREMY F.
"Decision making and medical care : can information science help ? ".
Proc. IFIP conference, Dijon, 1976.

[DEL-82] DELOBEL C., ADIBA M.
"Bases de données et systèmes relationnels"
Edit. Dunod, 1982.

[DUB-86] DUBOIS D., PRADE H., TESTEMALE C.
"Weighted fuzzy pattern matching".
Actes journée Nat. sur les ensembles flous, la théorie des possibilités et leurs applications, Toulouse, Juin 1986.

[DUB-87] DUBOIS D., PRADE H.
"Théorie des possibilités.Applications à la représentation des connaissances en Informatique".
2ème édition, Masson, 1987.

[DUB-88] DUBOIS D., PRADE H.
" Properties of measures of information in evidence and possibility theories".
Fuzzy Sets and Systems, 1988.

[DUD-81] DUDA R. GASCHNIG J., HART P.
"Model design in the PROSPECTOR consultant system for mineral exploration".
Expert Systems in the Micro-Electronic Age,
Edinburgh Univ. Press, pp. 153-167, 1981.

[ERN-82] ERNST C.
"Le modèle de raisonnement approché du système MANAGER".
BUSEFAL, n° 9, L.S.I, Univ. P. SABATIER, TOULOUSE, pp. 93-99, 1982.

[EXP-83] Equipe EXPRIM
"Rapport final de conventions dans le cadre du PRC BD3".
Rapport interne CRIN, n° 87-R-096, Nancy, 1987.

[FAR-85] FARRENY H.
"Les Systèmes Experts, principes et exemples",
Cepadues ed., Toulouse, juin 1985.

[FAR-86] FARRENY H., PRADE H., WYSS E.
"Approximate reasoning in a rule-based expert system using possibility theory : a case study",
In : Information Processing 86, North-Holland,
Proc. 10th world IFIP Conf., pp. 407-413, Dublin, Sept. 1986.

[FEL-69] FELDMAN J.A., ROVNER P.D.
"An algol base associative language"
Communication ACM, pp. 439-449, 1969.

[FIE-81] FIESCHI M.
"SPHINX : un système-expert d'aide à la décision en médecine",
Thèse d'Etat en Biologie Humaine, Université d'Aix-Marseille, mars 1983.

[FIE-84] FIESCHI M.
"Intelligence Artificielle en Médecine".
Masson, Coll. Méthodes + Programmes, 1984.

[FOU-85] FOUCAUT O., FOUCAUT J.F.
"Mycomatic : Système Expert pour la reconnaissance des champignons supérieurs".
Rapport interne CRIN n° 85-R-069, Univ. Nancy 1, 1985.

[FOU-87a] FOUCAUT O., MOUADDIB N., CREHANGE M.
"Les Informations nuancées, une approche de leur gestion. Application à la Mycologie".
Congrès INFORSID, Lyon, 1987.

[FOU-87b] FOUCAUT O.
"Mycovision : système de gestion et d'interrogation de connaissances mycologiques intégrants textes et images".
Rapport interne CRIN n°87-R-118, Univ. Nancy I, 1987.

[FOU-88] FOUCAUT O. et al.
"Système expert d'aide à l'identification d'espèces naturelles intégrant textes et images : applications à la mycologie".
1ère Journée Internationale "Applications de l'Intelligence Artificielle à l'Agriculture, à l'Agrochimie".
Caen, France, Septembre 1988.

[FRE-80] FREKSA C.,
"L-Fuzzy-an A.I. language with linguistic modification of patterns".
AISB Conf. Amsterdam, 1980.

[FRI-81] FRIEDMAN L.
"Extended plausible inference".
Proc. 7th. Ind. Joint. Conf. Artificial Intelligence, Vancouver, pp. 487-495, Août 1981.

[GOL-83] GOLDBERG A., ROBSON D.
"Smaltalk 80. The language and its implementation".
Addison Wesley Publishing Compagny, 1983.

[GRA-79] GRANT J.
"Partial values in a tabular database".
Information Processing Letters, vol. 9, n° 2, pp. 97-99, 1979.

[HAA-77] HAAR R.L.
"A fuzzy relational data base system".
University of Maryland. Computer Center. TR-586. Sept. 1977.

[HAL-88] HALIN G., MOUADDIB N., FOUCAUT O., CREHANGE M.
"Semantics of user interface for image retrieval : possibility theory and learning techniques applied on two prototypes".
Proc. Conf. RIAO 88, Boston (USA), Mars 1988.

[HAL-89] HALIN G.
"Apprentissage pour la recherche interactive et progressive d'image : processus EXPRIM et prototype RIVAGE".
Thèse d'université, Nancy 1, Soutenance courant Sept. 1989.

[HAM-86] HAMON G.
"Extention d'un langage d'interrogation de base de données en vue de l'utilisation de questions imprécises".
Thèse, INSA, Juin 1986.

[HIR-87] HIRCH G.
"Equations de relation floue et mesures d'incertain en reconnaissance de formes".
Thèse d'état, Univ. Nancy 1, Avril 1987.

[HUD-85] HUDRISIER H.
"L'imageur Documentaire".
Le documentaliste, vol. 22, n° 4-5, pp. 155-160, Juillet-Octobre 1985.

[HUL-87] HULL R., KING R.
"Semantic Database Modeling : Survey, Applications and Research Issue".
ACM Computing Surveys, vol. 18, n° 3, pp. 202-260, 1987.

[IMI-81] IMIELINSKI T., LIPSKI W.
"On representing incomplete information in a relational database".
Proc. 7th. Int. Conf. on Very Large Databases, Cannes, France, Sept. 1981.

[IMI-84] IMIELINSKI T., LIPSKI W.
"Incomplete information in relational Database".
Journal of the Association for Computing Machinery, vol. 31, n° 4, pp. 761-791, 1984.

[ISH-81] ISHIZUKA M., FU K.S., YAO J.T.P.
"Inexact inference for rule-based damage assessment of existing structures".
Proc. 7th Int. Joint Conf. on Art. Intelligence, pp. 837-842, Vancouver, 1981.

[JAE-79] JAEGERMANN M.
"Information storage and retrieval systems with incomplete information".
Fund. Inform. 2, pp. 141-166, 1979.

[KAY-79] KAYSER D.
"Vers une modélisation du raisonnement approximatif".
Proc. of the colloquium "Représentation des connaissances et raisonnement dans les sciences de l'Homme", St. Maximin, Sept. 1979.

[LAN-88] LANDRAC J.Y.
"Maquette de méta-système d'identification d'oobjets".
Rapport de DEA, CRIN, Université de Nancy 1, Juin 1988.

[LAU-82] LAURIERE J.L.
"Représentation et utilisation des connaissances".
1ère partie, TSI, vol. 1, n° 1, 1982.

[LIP-79] LIPSKI W.
"On semantic issues connected with incomplete information databases".
ACM Trans. Database Systems, vol. 4, n° 3 pp. 262-296, 1979.

[LIP-81] LIPSKI W.
"On databases with incomplete information".
J. of Association for Computing Machinery, vol. 28, n° 1, pp. 41-70, 1981.

[MAS-89] MASINI G., et al
"Les langages à objets".
Inter-édition, à paraître, 1989.

[MIN-75] MINSKI M.
"A framework for representing knowledge".
In "The psychology of computer vision".
Ph. winston Eds, Mc Graw Hill, 1975.

[MOU-88] MOUADDIB N., FOUCAUT O.
"Gestion des informations nuancées dans une base de connaissances en sciences naturelles. Utilisation de la théorie des possibilités".
Congrès BD3, Bénodet, 1988.

[MOU-89] MOUADDIB N., FOUCAUT O.
" Les informations nuancées dans les systèmes d'informations. Application à une base de connaissances en sciences naturelles ".
Congrès "Les systèmes d'informations" organisé par SFBA, Corse, France, Mai 1989.

[MOU-89] MOUADDIB N., HALIN G.
"Deux modèles sémantiques pour la mise en correspondance entre requêtes et documents".
Congrès INFORSID, Nancy, France, Mai 1989.

[PET-79] PETERSON R. et al
"Guide des oiseaux d'Europe".
Delachaux et Niestlé, 1979.

[PHI-79] PHILIPS R.J. et al
"An Architectural relational database".
Computer-Aided-Design, vol. 11, n° 4, 1979.

[PIN-81] PINSON S.
"Représentations des connaissances dans les systèmes Experts".
RAIRO informatique, vol. 15, n° 4, 1981.

[PRA-82a] PRADE H.
"The connection between Lipski's approach to incomplete information data bases and Zadeh's possibility theory".
Proc. Int. Conf. Systems Methodology, Washington, pp. 402-408, Janv. 1982.

[PRA-82b] PRADE H.
"Possibility sets, fuzzy sets and their relation to ukasiewiez logic".
Proc. 12th. Symp. on Multiple-Valued Logic, Paris, pp. 223-227, May 1982.

[PRA-82c] PRADE H.
"Modèles mathématiques de l'imprécis et de l'incertain en vue d'application au raisonnement naturel".
Thèse d'Etat, Univ. P. Sabatier, Toulouse, 1982.

[PRA-83a] PRADE H.
"Représentation d'informations incomplètes dans une base de données à l'aide de la théorie des possibilités".
Proc. Convention Informatique Latine 83, Barcelone, Spain, June 1983.

[PRA-83b] PRADE H., TESTEMALE C.
"Generalizing database relational algebra for the treatement of incomplete/uncertain informations and vague queries".
2nd NAFIP Workshop, Schenectady, June 1983.

[PRA-84] PRADE H.
"Lipski's approach to incomplete information databases restated and generalized in the setting of Zadeh's possibility theory".
Information systems, vol. 9, n° 1, 1984.

[PRA-85] PRADE H.
"Reasoning with fuzzy default values".
Proc. 15th IEEE Int. Symp. Multiple - Valued Logic, pp. 191-197, Kingston, Ontario 1985.

[REC-84] RECHENMANN F.
"Intelligence Artificielle et construction de modèles dynamiques".
Intelligence Artificielle et Productique, 2ième symposium et exposition international 84, Paris, Nov. 1984.

[ROM-53] ROMAGNESI H., KUHNER R.
"Flore analytique des champignons supérieurs (Agarics, Bolets, Chanterelles)".
Masson 1953.

[ROM-63] ROMAGNESI H.
"Petit atlas des champignons".
Bordas, 1963.

[ROMA-87] ROMARY L.
"Quelques problèmes liés à l'incertain en reconnaissance de la parole".
Rapport de DEA, Univ. Nancy 1, Juin 1987.

[SCH-77] SCHANK R.C., ABELSON R.
"scripts, plans, goals and understanding".
Lawrence Erlbaum Assoc., N.J., 1977.

[SHA-76] SHAFER G.
"A Mathematical Theory of Evidence".
Princeton University Press, 1976.

[SHO-75] SHORTLIFFE E.H., BUCHANAN B.G.
"A model of inexact reasoning in medicine".
Mathematical Biosciences, n° 23, 1975.

[SKA-78] SKALA H.L.
"On many-valued logics, fuzzy sets, fuzzy logics and their applications".
Fuzzy sets and Systems. vol. 1, Number 2, North Holland, 1978.

[SLY-86] Van SLYPE G., MANIEZ J.
"Les langages documentaires".
Les éditions d'organisations, 1986.

[SMI-77] SMITH J.M., SMITH D.C.P.
"Data base abstractions : Aggregation ans Generalization".
ACM Transaction on Data Base Systems, vol. 2, n° 2, 1977.

[SOU-82] SOULA G., SANCHEZ E.
"Soft deduction rules in medical diagnosis processes".
In : Approximate Reasoning in Decision Analysis, pp. 77-88, North Holland, 1982.

[SOU-83] SOULA G. et al.
"PROTIS, a fuzzy deduction rule system : application to the treatment of diabetes".
Proc. MEDINFO 83, Amsterdam, 1983.

[TAH-77] TAHANI V.
"A conceptual framework for fuzzy query processing - A step toward very intelligent database systems".
Information Processing & Management, vol. 13, pp. 289-303, 1977.

[TAR-83] TARDIEU H. et al
"La méthode MERISE : principes et outils".
Ed. d'Organisation, 1983.

[TES-84] TESTEMALE C.
"Un système de traitement d'informations incomplètes ou incertaines dans une base de données relationnelle".
Thèse de 3 ème cycle, Univ. P. Sabatier, Toulouse, Juin 1984.

[ULL-80] ULLMAN J.D.
"Principles of database systems".
Computer science. Press, 1980.

[UMA-79] UMANO M., MIZUMOTO M., TANAKA K.
"A system for fuzzy reasoning".
Proc 6th, IJCAI, Tokyo, Août 1979.

[UMA-82] UMANO M., FREEDOH O.
"A fuzzy database system".
In : Fuzzy Information and Decision Processes, pp. 339-347, North-Holland, 1982.

[UMA-83] UMANO M.
"Retrieval from fuzzy database by fuzzy relational algebra".
Proc. IFAC Symposium on Fuzzy Information Knowledge Representation & Decision Analysis, Marseille, July 1983.

[VIG-85] VIGNARD P.
"Un mécanisme d'exploitation à base de filtrage flou pour une représentation des connaissances Centrées objets".
Thèse de 3ème cycle, Grenoble, Juin 1985.

[WAT-78] WATERMAN D.A., HAYES-ROTH F.
"Pattern-Directed inference systems".
Academic Press, New-York, 1978.

[WON-82] WONG E.A.
"Statistical approach to incomplete information in database systems".
ACM Trans. on Database Systems, vol. 7, n° 3, pp. 470-488, 1982.

[ZAD-65] ZADEH L.A.
"Fuzzy sets"
Information and control. Vol. 8, pp. 338-353, 1965.

[ZAD-75] ZADEH L.A. et al
"Fuzzy sets and their applications to cognitive and decision processes".
Academic Press, Inc. New York, 1975.

[ZAD-78] ZADEH L.A.
"Fuzzy sets as basis for a theory of possibility".
Fuzzy sets ans systems. Vol. 1, n° 1, pp. 3-28, 1978.

[ZAD-79] ZADEH L.A.
"A theory of approximate reasoning".
Mach. Intell., vol. 9, PP. 149-194, New-York, Elsevier, 1979.

[ZAD-83] ZADEH L.A.
"A computationnal approach to fuzzy quantifiers in natural languages".
Comp. and Maths with Appls. vol. 9, n° 1, pp. 149-184, 1983.

ANNEXES

ANNEXE 1 : Extraits de la base de connaissances

ANNEXE 2 : Extraits des programmes

ANNEXE 3 : Exemples d'exécution

ANNEXE 1

Extraits de la base de connaissance

Nous présentons dans cette annexe quelques exemples de représentation interne des connaissances décrites au troisième chapitre.

```
:champignon
((super_classe )
 (def
    (lait 1 (non))
    (cortine 1 (non))
    (chair_grenue_cassante 1 (non)))

:amanita
((super_classe :champignon)
 (def
    (type_app_sporif 1 (lamelles))
    (coul_app_sporif 1 (blanc))
    (volve 1 (oui))
    (anneau 1 (oui))
    (saison_recolte 1 (ete) (automne))
    (allure_lamelles 1 (libres))
    (odeur_nature 0 (quelconque))
    (allure .8 (all_1) (all_3))))

:vaginata
((super_classe :amanita)
 (def
    (anneau 1 (non))
    (diam_chapeau .9 ((4 12)))
    (coul_chapeau 1 (gris) (fauve) (blanc) (orange))
    (revet_chapeau 1 (strie marge))
    (diam_pied .9 ((1.5 2)))
```

```
    (haut_pied .9 ((13 20)))
    (coul_pied .6 (gris) (fauve) (blanc) (orange))
    (allure_pied .7 (elance))
    (carac_chair 1 (fragile) (mince))
    (lieu_recolte .8 (feuillus) (coniferes))
    (saison_recolte 1 (printemps fin) (ete) (automne))))

:pantherina
((super_classe :amanita)
 (def
    (allure_lamelles .7 (libres) (serrees))
    (diam_chapeau .9 ((6 15) environ))
    (coul_chapeau 1 (brun generalement) (bistre generalement)  (blanchatre rarement))
    (revet_chapeau 1 (verruqueux))
    (diam_pied .9 ((0.5 1.5) environ))
    (haut_pied .9 ((5 15) environ))
    (coul_pied .6 (blanc))
    (allure_pied .9 (bulbeux))
    (revet_pied 1 (bourrelets_blancs))
    (lieu_recolte .8 (coniferes) (feuillus))
    (saison_recolte 1 (ete fin) (automne))
    (image 1 (60) (61) (62) (63) (64))))

:phalloides
((super_classe :amanita)
 (def
    (volve 1 (membraneuse))
    (coul_app_sporif .6 (blanc) (verdatre legerement))
    (diam_chapeau .9 ((5 15) environ))
    (coul_chapeau 1 (vert) (jaune) (verdatre) (blanc parfois))
    (revet_chapeau 1 (vergete souvent) (fibrilleux) (strie marge))
    (allure_lamelles 1 (libres) (serrees assez))
    (diam_pied .9 ((0.8 2) environ))
    (haut_pied .9 ((5 11) environ))
    (allure_pied .7 (bulbeux))
```



```
    (revet_pied 1 (tigre) (floconneux))
    (lieu_recolte .8 (feuillus))
    (image 1 (44) (45) (46) (47) (48) (49) (50) (51) (52) (53))))

:citrina
((super_classe :amanita)
 (def
    (allure .8 (all_1) (all_2) (all_3))
    (diam_chapeau .9 ((5 10)))
    (coul_chapeau 1 (jaune souvent) (blanc parfois))
    (revet_chapeau 1 (squameux) (verrues))
    (diam_pied .9 ((0.8 1.5)))
    (haut_pied .9 ((5 12)))
    (allure_pied .7 (bulbeux gros))
    (odeur:nature .7 (pomme_de_terre_crue))
    (odeur:intensite .7 (moyenne))
    (image 1 (82) (83) (84) (85) (86) (866))))

:verna
((super_classe :amanita)
 (def
    (allure .8 (all_3))
    (diam_chapeau .9 ((5 12)))
    (coul_chapeau 1 (blanc) (blanchatre) (creme))
    (allure_chapeau 1 (mamelonne) (conique) (etale))
    (revet_chapeau 1 (lisse))
    (diam_pied .9 ((1 1.5)))
    (haut_pied .9 ((9 12)))
    (coul_pied .8 (blanc))
    (revet_pied 1 (lisse) (nu))
    (carac_chair 1 (mince))
    (lieu_recolte .8 (coniferes) (feuillus))
    (saison_recolte 1 (ete) (automne) (printemps))
    (image 1 (54) (55))))
```

```
:virosa
((super_classe :amanita)
 (def
    (diam_chapeau .9 ((4 10) environ))
    (coul_chapeau 1 (blanc))
    (allure_chapeau .7 (mamelonne))
    (revet_chapeau 1 (lisse))
    (diam_pied .9 ((0.6 1.5) environ))
    (haut_pied .9 ((8.5 15) environ))
    (coul_pied .8 (blanc))
    (revet_pied 1 (pelucheux souvent) (nu parfois))
    (carac_chair 1 (mince))
    (couleur_chair .6 (blanc))
    (lieu_recolte .8 (sol_acide principalement) (feuillus))
    (image 1 (56))))

:rubescens
((super_classe :amanita)
 (def
    (coul_app_sporif .8 (blanc) (rose) (rougeatre))
    (diam_chapeau .9 ((5 15)))
    (coul_chapeau 1 (brun) (rose) (rougeatre) (vineux ))
    (revet_chapeau 1 (verruqueux))
    (diam_pied .9 ((1 3.5)))
    (haut_pied .9 ((6 22)))
    (coul_pied .9 (blanc) (rougissant a_la_base))
    (allure_pied .7 (bulbeux))
    (couleur_chair 1 (rougissante))
    (lieu_recolte .8 (coniferes) (feuillus))
    (image 1 (69) (70) (71) (72) (73))))

:muscaria
((super_classe :amanita)
 (def
    (diam_chapeau .9 ((6 20)))
```



```
    (coul_chapeau 1 (rouge) (orange))
    (revet_chapeau 1 (verruqueux parfois) )
    (diam_pied .9 ((6 20)))
    (haut_pied .9 ((15 25)))
    (allure_pied .7 (bulbeux))
    (revet_pied 1 (bourrelets_blancs a_la_base))
    (carac_chair 1 (ferme))
    (couleur_chair .8 (blanc) (jaune sous_cuticule))
    (lieu_recolte .8 (bouleaux surtout) (coniferes parfois))
    (saison_recolte 1 (ete fin) (automne))
    (image 1 (75) (76) (77) (78) (79))))

:caesarea
((super_classe :amanita)
 (def
    (coul_app_sporif 1 (jaune franc))
    (diam_chapeau .9 ((8 20)))
    (coul_chapeau 1 (orange souvent) (jaune rarement))
    (revet_chapeau 1 (strie))
    (diam_pied .9 ((2 3)))
    (haut_pied .9 ((8 15)))
    (coul_pied .6 (jaune))
    (allure_pied .7 (bulbeux))
    (couleur_chair .8 (blanc) (jaune sous_cuticule))
    (lieu_recolte .8 (chenes) (chataigniers))
    (image 1 (40) (41) (42))))

:limacella_guttata
((super_classe :amanita)
 (def
    (allure .8 (all_2) (all_3))
    (volve 1 (non))
    (anneau 1 (oui))
    (coul_app_sporif .8 (blanc) (creme))
    (carac_sporee 1 (blanc))
```

```
    (diam_chapeau .9 ((5 10)))
    (coul_chapeau 1 (creme) (rosatre))
    (allure_chapeau .7 (convexe) (obtus) (conique))
    (revet_chapeau 1 (lisse) (brillant) (visqueux))
    (allure_lamelles 1 (libres) (serrees))
    (diam_pied .9 ((0.8 2.5)))
    (haut_pied .9 ((6 12)))
    (coul_pied .6 (blanc))
    (allure_pied .7 (bulbeux generalement) (cylindrique parfois))
    (revet_pied 1 (lisse) (ponctue_de_gouttes))
    (odeur:intensite .8 (forte))
    (lieu_recolte .8 (hetres) (coniferes))
    (image 1 (88) (89) (90))))

:tricholoma
((super_classe :champignon)
 (def
    (volve 1 (non))
    (anneau 1 (non))
    (type_app_sporif 1 (lamelles))
    (odeur_nature 0 (quelconque))
    (odeur:intensite .8 (moyenne))))

:aurantium
((super_classe :tricholoma)
 (def
    (allure .8 (all_1) (all_8))
    (coul_app_sporif .6 (blanc) (creme))
    (diam_chapeau .9 ((5 10) generalement))
    (coul_chapeau 1 (orange) (roux))
    (allure_chapeau .7 (convexe) (marge_enroulee))
    (revet_chapeau 1 (visqueux tres) (meches))
    (allure_lamelles .7 (emarginees) (echancrees))
    (diam_pied .9 ((1 2)))
    (haut_pied .9 ((5 10)))
```



```
    (coul_pied 1 (blanc net) (roux) (orange en_zebrures))
    (allure_pied .1 (cylindrique) (elance))
    (revet_pied 1 (tigre) (chine) (zebre))
    (odeur:nature .8 (farine) (concombre))
    (couleur_chair .8 (blanc))
    (saveur_chair 1 (amere))
    (lieu_recolte 1 (sol_calcaire toujours) (herbe) (lisiere_bois))
    (saison_recolte 1 (automne))
    (image 1 (349) (350))))

:flavobrunneum
((super_classe :tricholoma)
 (def
    (allure .8 (all_1) (all_2) (all_8))
    (coul_app_sporif 1 (jaune) (roussissant))
    (diam_chapeau .9 ((5 9)))
    (coul_chapeau 1 (brun) (roux))
    (allure_chapeau .7 (convexe))
    (revet_chapeau 1 (visqueux) (floconneux))
    (allure_lamelles .7 (emarginees) (echancrees) (decurrentes legerement))
    (diam_pied .9 ((0.8 1.5)))
    (haut_pied .9 ((7 15)))
    (coul_pied 1 (jaunatre))
    (allure_pied .7 (cylindrique))
    (revet_pied 1 (fibrilleux))
    (odeur:nature .8 (farine))
    (carac_chair 1 (ferme))
    (lieu_recolte .8 (sol_humide) (feuillus) (bouleaux))
    (saison_recolte 1 (automne))
    (image 1 (345))))

:terreum
((super_classe :tricholoma)
 (def
    (allure .8 (all_2) (all_6) (all_8))
```

```
(coul_app_sporif .8 (blanchatre) (gris))
(diam_chapeau .9 ((3 8)))
(coul_chapeau 1 (gris) (bistre))
(allure_chapeau .7 (conique) (mamelonne))
(revet_chapeau 1 (veloute) (mat) (duveteux))
(allure_lamelles 1 (emarginees) (echancrees))
(diam_pied .9 ((0.6 1.6)))
(haut_pied .9 ((4 8)))
(coul_pied .6 (blanc))
(allure_pied .7 (cylindrique))
(revet_pied 1 (lisse))
(odeur:intensite .9 (faible))
(saveur_chair 1 (doux))
(lieu_recolte 1 (coniferes exclusivement))
(saison_recolte 1 (automne) (hiver))
(image 1 (334) (335) (336))))

:pardinum
((super_classe :tricholoma)
(def
    (allure .8 (all_1) (all_6) (all_8) (all_9))
    (coul_app_sporif .8 (blanc) (creme) (jaunatre))
    (diam_chapeau .9 ((5 20)))
    (coul_chapeau 1 (gris) (brun) (bistre))
    (allure_chapeau .6 (grand) (mamelonne faiblement) (irregulier souvent))
    (revet_chapeau 1 (ecailleux))
    (allure_lamelles 1 (emarginees tres) (echancrees tres) (libres presque) (serrees))
    (diam_pied .9 ((1 4)))
    (haut_pied .9 ((5 8)))
    (coul_pied .6 (blanc))
    (allure_pied .7 (epais) (enorme))
    (odeur:nature .5 (farine))
    (odeur:intensite 1 (faible))
    (couleur_chair .6 (blanc))
    (lieu_recolte 1 (montagne exclusivement) (hetres) (coniferes))
```



```
    (saison_recolte 1 (automne) (ete))
    (image 1 (356) (357))))

:ustaloides
((super_classe :tricholoma)
(def
    (allure .8 (all_1) (all_2) (all_8))
    (cortine 1 (fugace))
    (coul_app_sporif .6 (blanc) (roux))
    (diam_chapeau .9 ((6 12)))
    (coul_chapeau 1 (brun) (chatain) (roux))
    (allure_chapeau .9 (marge_enroulee))
    (revet_chapeau 1 (visqueux tres) (fibrilleux))
    (allure_lamelles .8 (emarginees) (echancrees))
    (diam_pied .9 ((1.5 3)))
    (haut_pied .9 ((5 10)))
    (coul_pied .9 (roux) (blanc en_haut))
    (allure_pied .7 (cylindrique))
    (odeur:nature 1 (farine))
    (odeur:intensite 1 forte))
    (lieu_recolte .8 (feuillus))
    (saison_recolte 1 (automne))
    (image 1 (346) (347))))

:albobrunneum
((super_classe :tricholoma)
(def
    (allure .8 (all_1) (all_2) (all_8))
    (cortine 1 (fugace))
    (coul_app_sporif .6 (blanc) (roussatre))
    (diam_chapeau .9 ((5 8)))
    (coul_chapeau 1 (brun) (roux) (chatain))
    (allure_chapeau .7 (convexe) (conique))
    (revet_chapeau 1 (fibrilleux) (visqueux) (vergete))
    (allure_lamelles .7 (emarginees) (echancrees) (libres))
```

```
    (diam_pied .9 ((1 1.5)))
    (haut_pied .9 ((4 6)))
    (coul_pied .9 (roux) (blanc en_haut))
    (allure_pied .7 (cylindrique))
    (odeur:nature 1 (farine))
    (carac_chair 1 (ferme))
    (couleur_chair .8 (blanc))
    (lieu_recolte .8 (coniferes) (montagne) (epiceas))
    (saison_recolte 1 (automne))
    (image 1 (348))))

:imbricatum
((super_classe :tricholoma)
 (def
    (allure .8 (all_1) (all_2) (all_8))
    (coul_app_sporif .6 (blanc) (creme) (roux))
    (diam_chapeau .9 ((4 10)))
    (coul_chapeau 1 (brun) (roux))
    (allure_chapeau .7 (convexe) (conique))
    (revet_chapeau 1 (ecailleux) (sec))
    (allure_lamelles .7 (emarginees) (echancrees) (adnees))
    (diam_pied .9 ((1 2.5)))
    (haut_pied .9 ((4 10)))
    (coul_pied .6 (brun) (roux))
    (odeur:intensite 1 (faible) (nulle))
    (carac_chair 1 (ferme))
    (couleur_chair .6 (blanc))
    (saveur_chair .8 (amere))
    (lieu_recolte .8 (coniferes surtout) (feuillus parfois))
    (saison_recolte 1 (automne))
    (image 1 (354))))

:scalpturatum
((super_classe :tricholoma)
 (def
```



```
    (allure .8 (all_1) (all_2) (all_8) (all_13))
    (cortine 1 (oui parfois))
    (coul_app_sporif .6 (blanc) (gris) (jaunissant))
    (diam_chapeau .9 ((1 5)))
    (coul_chapeau 1 (gris) (brun))
    (revet_chapeau 1 (pelucheux) (ecailleux))
    (allure_lamelles .9 (emarginees) (echancrees))
    (diam_pied .9 ((0.3 0.8)))
    (haut_pied .9 ((3 4)))
    (coul_pied .6 (blanc))
    (allure_pied .7 (cylindrique))
    (odeur:nature 1 (farine net))
    (couleur_chair .9 (jaunissant a_la_putrefaction) (blanche))
    (saveur_chair 1 (farine))
    (lieu_recolte .8 (coniferes) (feuillus))
    (saison_recolte 1 (ete) (automne))
    (image 1 (337))))

:virgatum
((super_classe :tricholoma)
 (def
    (allure .8 (all_1) (all_2) (all_8) (all_13))
    (coul_app_sporif .8 (gris) (blanc))
    (diam_chapeau .9 ((4 8)))
    (coul_chapeau 1 (gris) (violace))
    (allure_chapeau .9 (conique jeune) (mamelonne toujours))
    (revet_chapeau 1 (fibrilleux) (soyeux))
    (allure_lamelles .9 (emarginees) (echancrees) (serrees moyennement) (epaisses assez) )
    (diam_pied .9 ((0.4 1.2)))
    (haut_pied .9 ((5 10)))
    (coul_pied .6 (blanc raye))
    (allure_pied .7 (cylindrique))
    (odeur:intensite 1 (faible) (nulle))
    (couleur_chair .8 (blanc) (grisatre))
    (lieu_recolte .8 (sol_acide plutot) (coniferes) (feuillus) (hetres surtout))
```

```
    (saison_recolte 1 (automne) (ete fin))
    (saveur_chair 1 (acre))
    (image 1 (341) (342))))

:sciodes
((super_classe :tricholoma)
 (def
    (allure .8 (all_1) (all_2))
    (coul_app_sporif .8 (blanc) (gris) (rose parfois) (noir))
    (diam_chapeau .9 ((3 10)))
    (coul_chapeau 1 (gris) (violace))
    (allure_chapeau .7 (conique) (mamelonne rarement))
    (revet_chapeau 1 (excorie) (soyeux))
    (allure_lamelles .9 (emarginees) (echancrees))
    (diam_pied .9 ((1 2)))
    (haut_pied .9 ((6 9)))
    (coul_pied .6 (blanc))
    (odeur:intensite 1 (faible) (nulle))
    (couleur_chair .6 (blanc))
    (saveur_chair 1 (acre))
    (lieu_recolte .8 (feuillus) (hetres surtout))
    (saison_recolte 1 (automne))
    (image 1 (344))))

:album
((super_classe :tricholoma)
 (def
    (allure .8 (all_1) (all_2) (all_8))
    (coul_app_sporif .8 (blanc) (creme))
    (diam_chapeau .9 ((3 9)))
    (coul_chapeau 1 (blanc parfois) (creme parfois) (roussatre surtout))
    (allure_chapeau .7 (bombe) (etale a_la_fin))
    (revet_chapeau 1 (lisse) (mat))
    (allure_lamelles .9 (emarginees) (echancrees))
    (diam_pied .9 ((0.5 1.5)))
```



```
    (haut_pied .9 ((5 8)))
    (coul_pied .6 (blanc) (creme) (roussatre))
    (allure_pied .7 (cylindrique))
    (odeur:nature .8 (desagreable souvent) (farine) (parfumee))
    (couleur_chair .6 (blanc))
    (saveur_chair 1 (desagreable))
    (lieu_recolte .8 (feuillus) (hetres surtout))
    (saison_recolte 1 (ete) (automne))
    (image 1 (318) (319))))

:saponaceum
((super_classe :tricholoma)
 (def
    (allure .8 (all_1) (all_2))
    (coul_app_sporif .8 (blanc) (creme) (rougissant parfois))
    (diam_chapeau .9 ((5 15)))
    (coul_chapeau 1 (jaunatre) (brun) (gris) (olive))
    (allure_chapeau .7 (mamelonne))
    (revet_chapeau 1 (marge_pale))
    (allure_lamelles .7 (adnees) (emarginees) (echancrees))
    (diam_pied .9 ((1 3.5)))
    (haut_pied .9 ((5 15)))
    (coul_pied .9 (rougissant lentement))
    (allure_pied .7 (base_en_pointe))
    (odeur:nature .8 (savon))
    (carac_chair 1 (epaisse))
    (couleur_chair .6 (rosissante))
    (saveur_chair .5 (agreable))
    (lieu_recolte .8 (coniferes) (feuillus))
    (saison_recolte 1 (ete) (automne))
    (image 1 ((331 333)))))

:portentosum
((super_classe :tricholoma)
 (def
```

(allure .8 (all_1) (all_2) (all_8) (all_9))
(coul_app_sporif .8 (blanc) (jaunatre souvent))
(diam_chapeau .9 ((4 15)))
(coul_chapeau 1 (gris) (violace) (jaunatre parfois))
(revet_chapeau 1 (visqueux) (lisse) (vergete) (fibrilleux))
(allure_lamelles .9 (emarginees) (echancrees) (espacees assez))
(diam_pied .9 ((1 2.5)))
(haut_pied .9 ((6 10)))
(coul_pied .6 (jaune) (blanc))
(allure_pied .7 (cylindrique))
(revet_pied 1 (lisse))
(odeur:nature 1 (farine))
(couleur_chair .8 (blanc) (jaune sous_cuticule) (brun sous_cuticule))
(saveur_chair 1 (farine) (doux))
(lieu_recolte .8 (coniferes surtout) (feuillus parfois))
(saison_recolte 1 (ete fin) (automne))
(image 1 (329) (330))))

:sulfureum
((super_classe :tricholoma)
(def
(allure .8 (all_2) (all_8))
(coul_app_sporif 1 (jaune) (verdatre) (soufre))
(diam_chapeau .9 ((3 8)))
(coul_chapeau 1 (jaune) (verdatre) (fauve) (olive) (soufre))
(revet_chapeau 1 (soyeux parfois) (glabre parfois) (mat surtout))
(allure_lamelles .8 (emarginees) (echancrees) (espacees))
(diam_pied .9 ((0.7 1)))
(haut_pied .9 ((6 10)))
(coul_pied .9 (jaune) (verdatre) (fauve) (olive) (soufre))
(allure_pied .7 (cylindrique))
(revet_pied 1 (raye))
(odeur:nature 1 (gaz))
(odeur:intensite 1 (forte))
(couleur_chair .9 (jaune))

(lieu_recolte .8 (coniferes) (feuillus))
(saison_recolte 1 (automne))
(image 1 (213) (322) (323))))

:columbetta
((super_classe :tricholoma)
(def
(allure .8 (all_1) (all_2) (all_8))
(coul_app_sporif .6 (blanc) (rouge rarement))
(diam_chapeau .9 ((5 10)))
(coul_chapeau 1 (blanc) (bleu parfois) (rose parfois))
(revet_chapeau 1 (fibrilleux) (soyeux))
(allure_lamelles .8 (emarginees) (echancrees))
(diam_pied .9 ((1 2.5)))
(haut_pied .9 ((5 7)))
(coul_pied 1 (blanc) (bleu parfois) (rose parfois))
(allure_pied .7 (cylindrique))
(odeur:intensite 1 (faible) (nulle))
(carac_chair 1 (ferme))
(couleur_chair .9 (blanc))
(lieu_recolte .8 (sol_siliceux) (coniferes) (feuillus))
(saison_recolte 1 (automne) (ete))
(image 1 (320) (321))))

:sejunctum
((super_classe :tricholoma)
(def
(allure .8 (all_1) (all_2) (all_8))
(coul_app_sporif .8 (blanchatre) (blanc) (jaunatre))
(diam_chapeau .9 ((5 10)))
(coul_chapeau 1 (jaune) (vert) (blanchatre rarement))
(revet_chapeau 1 (vergete) (fibrilleux) (visqueux peu))
(allure_lamelles .7 (libres presque) (emarginees) (echancrees) (espacees))
(diam_pied .9 ((1 2.5)))
(haut_pied .9 ((3 8)))

```
    (coul_pied .6 (jaune parfois) (blanc))
    (allure_pied .7 (cylindrique))
    (odeur:nature 1 (farine))
    (carac_chair .8 (epaisse))
    (couleur_chair .8 (jaune parfois) (blanche generalement))
    (saveur_chair 1 (farine) (amere parfois))
    (lieu_recolte .8 (coniferes) (feuillus))
    (saison_recolte 1 (ete) (automne))
    (image 1 (324) (325))))

:equestre
((super_classe :tricholoma)
 (def
    (allure .8 (all_1) (all_2) (all_8))
    (coul_app_sporif 1 (jaune) (citron) (soufre))
    (diam_chapeau .9 ((4 10)))
    (coul_chapeau 1 (jaune) (soufre))
    (revet_chapeau 1 (visqueux) (squameux))
    (allure_lamelles .9 (libres presque) (emarginees) (echancrees) (serrees))
    (diam_pied .9 ((0.8 2)))
    (haut_pied .9 ((4 10)))
    (coul_pied .8 (jaune) (soufre))
    (allure_pied .7 (cylindrique))
    (revet_pied 1 (lisse))
    (odeur:intensite 1 (faible) (nulle))
    (couleur_chair .8 (blanchatre) (jaune))
    (lieu_recolte .8 (coniferes) (sol_siliceux))
    (saison_recolte 1 (automne))
    (image 1 ((326 328)))))

:vaccinum
((super_classe :tricholoma)
 (def
    (allure .8 (all_1) (all_2) (all_8))
    (coul_app_sporif .8 (blanc) (creme) (roux))
```



```
    (diam_chapeau .9 ((4 9)))
    (coul_chapeau 1 (roux) (fauve))
    (allure_chapeau .9 (mamelonne) (marge_enroulee))
    (revet_chapeau 1 (ecailleux) (poilu) (meches))
    (allure_lamelles .7 (emarginees) (echancrees) (adnees))
    (diam_pied .9 ((0.8 2)))
    (haut_pied .9 ((4 5.5)))
    (coul_pied .9 (blanc en_haut) (roux))
    (allure_pied .7 (cylindrique))
    (revet_pied 1 (fibrilleux))
    (odeur:nature 1 (farine))
    (odeur:intensite 1 (faible))
    (couleur_chair .8 (rosatre) (blanc))
    (saveur_chair 1 (amere))
    (lieu_recolte .8 (coniferes generalement) (feuillus parfois))
    (saison_recolte 1 (ete) (automne))
    (image 1 (352) (353))))

:orirubens
((super_classe :tricholoma)
 (def
    (allure .8 (all_1) (all_2) (all_8))
    (coul_app_sporif .8 (blanc) (gris) (rose sur_l_arete) (rougissant en_vieillissant))
    (diam_chapeau .9 ((5 10)))
    (coul_chapeau 1 (gris) (brun) (noir) (rougeatre))
    (revet_chapeau 1 (ecailleux))
    (allure_lamelles .9 (emarginees) (echancrees))
    (diam_pied .9 ((0.8 1.8)))
    (haut_pied .9 ((4 7)))
    (coul_pied .6 (bleu_vert a_la_base) (blanc) (rouge parfois))
    (odeur:nature 1 (farine))
    (odeur:intensite 1 forte))
    (carac_chair 1 (epaisse))
    (couleur_chair .8 (blanc) (rougissante lentement))
    (saveur_chair 1 (farine))
```

(lieu_recolte .8 (feuillus) (hetres surtout) (sol_argileux souvent))
(saison_recolte 1 (automne))
(image 1 (338) (339))))

:lyophyllum_decastes
((super_classe :tricholoma)
(def
(allure .8 (all_11) (all_2))
(coul_app_sporif .8 (blanc) (creme))
(diam_chapeau .9 ((5 15)))
(coul_chapeau 1 (brun) (gris))
(revet_chapeau 1 (lisse) (fibrilleux))
(allure_lamelles .7 (adnees) (decurrentes peu))
(diam_pied .9 ((1 2.5)))
(haut_pied .9 ((6 28)))
(coul_pied .6 (blanc))
(allure_pied 1 (tenace))
(carac_chair 1 (ferme) (elastique))
(lieu_recolte .8 (lignicole) (coniferes) (feuillus))
(saison_recolte 1 (automne))
(image 1 (315) (316))))

:lepista_nuda
((super_classe :tricholoma)
(def
(allure .8 (all_2) (all_8))
(coul_app_sporif .8 (violet souvent) (brun parfois))
(diam_chapeau .9 ((2 12)))
(coul_chapeau 1 (violet souvent) (brun parfois))
(allure_lamelles .9 (emarginees) (echancrees))
(diam_pied .9 ((1 3)))
(haut_pied .9 ((3 10)))
(allure_pied .7 (bulbeux))
(odeur:nature 1 (fruite))
(carac_chair 1 (epaisse))

(couleur_chair .8 (violet))
(lieu_recolte .8 (feuillus) (coniferes))
(saison_recolte .8 (automne) (hiver) (printemps))
(image 1 (389) (390) (391))))

:lepista_personata
((super_classe :tricholoma)
(def
(allure .8 (all_3) (all_7) (all_8) (all_9))
(coul_app_sporif .8 (blanc) (brun))
(diam_chapeau .9 ((6 10)))
(coul_chapeau 1 (brun) (bistre) (gris) (beige))
(allure_chapeau .7 (convexe))
(revet_chapeau 1 (lisse))
(allure_lamelles .7 (emarginees) (echancrees) (decurrentes parfois))
(diam_pied .9 ((1.5 2)))
(haut_pied .9 ((4 7)))
(coul_pied 1 (violet))
(revet_pied .8 (fibrilleux))
(allure_pied .7 (court))
(odeur:nature 1 (agreable))
(carac_chair 1 (ferme))
(lieu_recolte 1 (pres toujours))
(saison_recolte 1 (automne fin) (hiver))
(image 1 (394) (395) (396))))

:lepista_luscina
((super_classe :tricholoma)
(def
(allure .8 (all_1) (all_2) (all_8) (all_9))
(coul_app_sporif .8 (blanc) (gris) (creme))
(diam_chapeau .9 ((3 10)))
(coul_chapeau 1 (gris) (brun) (beige))
(allure_lamelles .7 (emarginees) (echancrees) (decurrentes))
(diam_pied .9 ((1 2)))

(haut_pied .9 ((3 7)))
(coul_pied .6 (gris) (brun) (beige))
(revet_pied 1 (fibrilleux))
(odeur:nature 1 (farine))
(odeur:intensite 1 forte))
(carac_chair 1 (tendre))
(saveur_chair 1 (farine))
(lieu_recolte 1 (pres))
(saison_recolte 1 (automne))
(image 1 ((400 402)))))

:lepista_irina
((super_classe :tricholoma)
(def
(allure .8 (all_2) (all_8) (all_9))
(coul_app_sporif .8 (creme) (brun) (roussatre))
(diam_chapeau .9 ((5 8.5)))
(coul_chapeau 1 (creme) (brun) (roussatre))
(allure_chapeau .7 (mamelonne))
(revet_chapeau 1 (lisse))
(allure_lamelles .7 (emarginees) (echancrees) (decurrentes legerement))
(diam_pied .9 ((1 2.5)))
(haut_pied .9 ((7 10.5)))
(coul_pied .8 (blanchatre))
(revet_pied 1 (fibrilleux))
(odeur:nature 1 (iris))
(odeur:intensite 1 (forte))
(lieu_recolte .8 (feuillus) (hetres surtout) (clairieres))
(saison_recolte 1 (automne))
(image 1 ((397 399)))))

:tricholomopsis_rutilans
((super_classe :tricholoma)
(def
(allure .8 (all_1) (all_2) (all_8))

(coul_app_sporif .9 (jaune))
(diam_chapeau .9 ((5 20)))
(coul_chapeau 1 (pourpre) (jaunatre) (rougeatre) (vineux) (brun))
(allure_chapeau .9 (mamelonne) (convexe))
(revet_chapeau 1 (meches) (veloute))
(allure_lamelles .7 (adnees) (serrees) (emarginees) (echancrees) (libres presque))
(diam_pied .9 ((1 2)))
(haut_pied .9 ((6 9)))
(coul_pied .6 (pourpre) (jaunatre) (rougeatre) (vineux) (brun))
(allure_pied .7 (cylindrique))
(revet_pied 1 (meches))
(odeur:intensite .7 (faible) (nulle))
(couleur_chair .6 (jaune))
(lieu_recolte .8 (lignicole) (coniferes) (sur_souches))
(saison_recolte 1 (automne) (ete))
(image 1 (378) (379))))

:calocybe_gambosa
((super_classe :tricholoma)
(def
(allure .8 (all_2) (all_8) (all_9))
(coul_app_sporif .8 (blanchatre) (creme))
(diam_chapeau .9 ((5 15)))
(coul_chapeau 1 (blanc) (rose) (ocre))
(revet_chapeau 1 (mat))
(allure_lamelles .9 (emarginees) (echancrees) (serrees))
(diam_pied .9 ((1 4.5)))
(haut_pied .9 ((4 9)))
(coul_pied .8 (blanc) (rose) (ocre))
(allure_pied .7 (epais))
(odeur:nature 1 (farine))
(odeur:intensite 1 (forte))
(carac_chair 1 (ferme))
(saveur_chair 1 (farine))
(lieu_recolte .8 (herbe) (feuillus parfois) (friches) (buissons))

```
    (saison_recolte 1 (printemps surtout) (automne rarement))
    (image 1 ((312 314)))))

:hygrophorus_russula
((super_classe :tricholoma)
(def
    (allure .8 (all_6) (all_15))
    (coul_app_sporif .8 (blanc) (pourpre en_tache))
    (diam_chapeau .9 ((5 20)))
    (coul_chapeau 1 (rose) (vineux) (pourpre))
    (allure_chapeau .7 (convexe) (plan))
    (revet_chapeau 1 (visqueux peu))
    (allure_lamelles .7 (emarginees) (echancrees) (adnees) (epaisses))
    (diam_pied .9 ((1 3)))
    (haut_pied .9 ((3 8)))
    (coul_pied .6 (pourpre))
    (allure_pied .7 (cylindrique))
    (carac_chair 1 (epaisse))
    (couleur_chair .8 (blanc))
    (saveur_chair .5 (amere parfois))
    (lieu_recolte .8 (feuillus) (hetres))
    (saison_recolte 1 (ete) (automne))
    (image 1 (535))))
```


ANNEXE 2

Extraits des programmes

Il n'est pas possible de publier ici l'ensemble des programmes qui présente environ 50 pages de texte source LE_LISP. Nous nous contenterons de donner les fonctions de calcul des mesures de possibilité et de nécessité et les fonctions de génération des fonctions caractéristiques à partir du thésaurus.

```
; FONCTIONS DE CALCUL DES MESURES DE POSSIBILITE ET DE NECESSITE

(de F_POSS_CONTINU (fc1 fc2)
    ; FC1, FC2 : sont 2 quadruplets représentant des fonctions caractéristiques trapézoïdales.
    ; BUT : calculer la mesure de possibilite entre les deux fonctions FC1 et FC2.
    ; RESULTAT : est la mesure de possibilité entre FC1 et FC2.

(let ((d1 (- (cadr fc2) (car fc1))) (d2 (- (cadr fc1) (car fc2))))
 (cond
    ((>= (+ (- (cadr fc1) (car fc1)) (- (cadr fc2) (car fc2))) (max d1 d2)) 1)
        ; c'est le cas où les 2 fonctions FC1 et FC2 se recouvrent, la possibilité est alors 1
    ((> d1 d2) (F_poss_cont_bis fc1 fc2))
    ((> d2 d1) (F_poss_cont_bis fc2 fc1)) ) ))


(de F_POSS_CONT_BIS (fc1 fc2)
    ; FC1, FC2 : sont 2 quadruplets représentant des fonctions caractéristiques trapézoïdales.
    ; BUT : calculer la mesure de possibilite entre les deux fonctions FC1 et FC2 lorsque
    ;           celles-ci ne se recouvrent pas, cela revient à calculer leur point d'intersection.
    ; RESULTAT : est la mesure de possibilité entre FC1 et FC2.

(cond
    ((and (zerop (caddr fc2)) (zerop (cadddr fc1))) 0)
    (t (max 0 (difference
                (difference (/ (float (- (car fc2) (cadr fc1))) (float (+ (cadddr fc1) (caddr fc2))))
                1))))))
```

```
(de F_NESS_CONTINU (fc1 fc2)
  ; FC1, FC2 : sont 2 quadruplets représentant des fonctions caractéristiques trapézoïdales.
  ; BUT : calculer la mesure de nécessité entre les deux fonctions FC1 et FC2 .
  ; RESULTAT : est la mesure de nécessité entre FC1 et FC2.

  (cond
      ((and (zerop (caddr fc1)) (zerop (caddr fc2))
            (zerop (cadddr fc1)) (zerop (cadddr fc2)))
       (cond ((< (car fc2) (car fc1)) 0)
             ((>= (cadr fc1) (cadr fc2)) 1)
             (t 0)))
      ((and (zerop (cadddr fc1)) (zerop (cadddr fc2)))
       (cond ((< (cadr fc1) (cadr fc2)) 0)
             (t (min 1
                     (max 0 (/ (float (+ (caddr fc1) (car fc2) (difference (car fc1))))
                     (float (+ (caddr fc2) (caddr fc1)))))))))
      ((and (zerop (caddr fc1)) (zerop (caddr fc2)))
            (cond ((< (car fc2) (car fc1)) 0)
                  (t (min 1
                          (max 0 (/ (float (+ (cadddr fc1) (cadr fc1) (difference (cadr fc2))))
                                    (float (+ (cadddr fc2) (cadddr fc1)))))))))
      (t (min 1 (min
           (max 0 (/ (float (+ (caddr fc1) (car fc2) (difference (car fc1))))
                     (float (+ (caddr fc2) (caddr fc1)))))
           (max 0 (/ (float (+ (cadddr fc1) (cadr fc1) (difference (cadr fc2))))
                     (float (+ (cadddr fc2) (cadddr fc1)))))))))))


(de F_POSS_DISCRET (fc1 fc2)
  ; FC1, FC2 : sont 2 listes de couples (<valeur> <coefficient>) représentant des fonctions
  ;            caractéristiques de 2 valeurs vagues définies sur des domaines discrets.
  ; BUT : calculer la mesure de possibilite entre les deux fonctions FC1 et FC2
  ; RESULTAT : est la mesure de possibilité entre FC1 et FC2.

(cond ((null (cdr fc2))
       (min (cadar fc2)
```



```
            (cond ((assq (caar fc2) fc1)
                   (cadr (assq (caar fc2) fc1)))
                  (t 0))))
      (t (max (F_poss_discret fc1 (list (car fc2)))
              (F_poss_discret fc1 (cdr fc2))))))


(de F_NESS_DISCRET (fc1 fc2)
  ; FC1, FC2 : sont 2 listes de couples (<valeur> <coefficient>) représentant des fonctions
  ;            caractéristiques de 2 valeurs vagues définies sur des domaines discrets.
  ; BUT : calculer la mesure de nécessité entre les deux fonctions FC1 et FC2
  ; RESULTAT : est la mesure de nécessité entre FC1 et FC2.

(cond
  ((null (cdr fc2))
   (max     (difference 1 (cadar fc2))
            (cond ((assq (caar fc2) fc1) (cadr (assq (caar fc2) fc1)))
                  (t 0))))
  (t (min (F_ness_discret fc1 (list (car fc2)))
          (F_ness_discret fc1 (cdr fc2))))))


; FONCTIONS DE CALCUL ET DE GENERATION DES FONCTIONS  :CARACTERISTIQUES

(de F_GENERE_RS (terme)
  ; TERME : terme du thésaurus dont on veut générer la fonction caractéristique.
  ; BUT : génère la fonction caractéristique associée au terme passé en paramètre.
  ; RESULTAT : une liste de couples (<terme> <coefficient>).

(cons   (list terme 1)
        (append (cassoc 'ts (eval terme))
                (cassoc 'tv (eval terme))
                (when (car (cassoc 'lien_sem (eval terme)))
                      (append (cassoc 'tg (eval terme))
                              (mapcan 'f_genere_freres
                                      (cirlist terme) (cassoc 'tg (eval terme))))))))
```

```
(de F_GENERE_FRERES (terme tg)
   ;TG : terme générique du terme passé en paramètre : c'est un couple de la forme (<terme gen.> <coef.>)
   ;      ou` COEF est le coefficient du lien spécifique/générique entre TERME  et TG.
   ; TERME : terme du thésaurus dont on veut générer les termes frères.
   ; BUT : génère les termes spécifiques du terme générique TG .
   ; RESULTAT : une liste de couples (<terme> <coefficient>).

(when (car (cassoc 'lien_sem (eval (car tg))))
 (let ((l_ts (cassoc 'ts (eval (car tg)))))
    (setq l_ts (delete (assoc terme l_ts) l_ts))
       ; on a enlevé TERME de la liste des termes spécifiques car on l'a déja
       ; généré avec le coefficient 1
    (mapcar 'F_calc_coef (cirlist (cadr tg)) l_ts))))


(de F_CALC_COEF (coef_tg ts)
   ; COEF_TG : coeff. liant le terme générique au terme dont on veut générer la fonction caractéristique
   ; TS  : c'est un terme spécifique (<terme spe.> <coef_ts>) du terme générique
   ; BUT :  calculer le coeff. de ressemblance entre le terme spécifique et le terme dont on veut générer la FC
   ; RESULTAT :  MAX (0,(COEF_TS - (abs(COEF_TG - COEF_TS)/2)))

(append (list (car ts)) (list (max 0 (- (cadr ts) (/ (abs (- coef_tg (cadr ts))) 2))))))


(de F_GENERE_DISCRET (un_caractere)
   ; UN_CARACTERE : (Nom_caractere pds (Val1 Nuance1) ... (Valn Nuancen)).
   ; BUT : générer la fonction caractéristique associée à un caractère de domaine discret.
   ; RESULTAT : une liste de la forme (Nom_caractere poids (Val1 Coef1) ... (Valq coefq))

(let ((def_dom (f_def_domaine (car un_caractere)))
   (l_nuance (cassoc 'nuan (eval (symbol 'c_caract (car un_caractere))))))
   (append     (list (car un_caractere)) (list (cadr un_caractere))
               (F_enleve_double
                (apply 'append
                       (mapcar
                         (lambda (couple)
```



```
                (mapcar 'f_nuance
                        (f_fc_val_discret couple def_dom)
                        (cirlist (cadr couple))
                        (cirlist l_nuance)))) ; fin du (lambda ..)
          (cddr un_caractere)))))))


(de F_GENERE_CONTINU (caractere)
   ; CARACTERE : (Nom_caract poids (Val1 Nuance1) ... (Valn Nuancen))
   ; BUT :   Cette fonction génère la fonction caractéristique associée au caractère passé en
   ;         paramètre et défini sur un domaine continu.
   ; RESULTAT :  (Nom_caract poids (Bl1 BS1 PG1 PD1)  ....  (Bln BSn PGn PDn))

   (append (list (car caractere)) (list (cadr caractere))
      (let ((l_nuance (cassoc 'nuan (eval (symbol 'c_caract  (car caractere))))))
      (mapcar
         (lambda (couple)
           (f_nuance  (f_fc_val_continu couple)  (cadr couple)  l_nuance))
            ; application de la nuance sur le quadruplet representant la valeur
           (cddr caractere)))))


(de F_TYPE_DOMAINE (nom_caractere)
   ; BUT : renvoyer le type de domaine (DISCret ou CONTinu du caractère dont le nom est passé en
   ;       paramètre.

(car (cassoc 'type (eval (symbol 'c_domaine (car (cassoc 'DOM  (eval (symbol 'c_caract nom_caractere)))))))))


(de F_MEILLEURS_MESURES (C_poss_ness L_poss_ness)
   ; C_POSS_NESS  :  (mesure_possiblite  mesure_nécessite).
   ; L_POSS_NESS  :  c'est une liste de couple de type C_POSS_NESS.
   ; BUT : retourner le couple ayant la meilleure mesure de nécessite.

 (cond ((null L_poss_ness)  C_poss_ness)
   ((= (cadr C_poss_ness) (cadar L_poss_ness))
```

```
        ; les mesures de nécessite' sont égales
        (if (< (car C_poss_ness) (caar L_poss_ness))
         (F_meilleurs_mesures (car L_poss_ness) (cdr L_poss_ness))
         (F_meilleurs_mesures C_poss_ness (cdr L_poss_ness))))
    ( (< (cadr C_poss_ness) (cadar L_poss_ness))
      (F_meilleurs_mesures (car L_poss_ness) (cdr L_poss_ness)))
    ( (> (cadr C_poss_ness) (cadar L_poss_ness))
      (F_meilleurs_mesures C_poss_ness (cdr L_poss_ness)))))
```

```
(de F_ENLEVE_DOUBLE (fc)
    ; FC : ((Val1 Coef1) .... (Valn Coefn))
    ; BUT :    éliminer les couples (Val Coefi) ayant la même valeur Val en les remplacant par un seul
    ;          couple (Val Coef) ou COEF est le max des COEFi .
    ; RESULTAT : (Val coef).

 (let ((fc_sans_double ()))
    (while fc
     (ifn (assoc (caar fc) fc_sans_double)
      (newr fc_sans_double (list (caar fc) (F_min_max_coef (car fc) (cdr fc)))))
     (setq fc (cdr fc)))
    fc_sans_double))
```

```
(de F_MIN_MAX_COEF (couple L_couple)
    ; COUPLE :  (Val Coef)
    ; L_COUPLE : ((Val1 Coef1) .... (Valn Coefn))
    ; BUT :    combiner par un MAX ou un MIN ou .. tous les coef. des couples de L_couple ayant la même
    ;          valeur que le paramètre Couple. Le coef. résultat sera le nouveau Coef. attachée à la valeur Val.

 (let  ((couple1 (assoc (car couple) L_couple)))
     (ifn couple1  (cadr couple)
                   (max (cadr couple)
                   (F_min_max_coef couple1 (cdr (member couple1 L_couple)))))))
```



```
(de F_NUANCE (fc nuance l_nuance)
    ; FC :      cas continu -> c'est un quadruplet (a b c d)
    ;           cas discret -> c'est un couple (terme coef)
    ; NUANCE : soit Nil (pas de nuance) soit le nom de la nuance donné dans le probot.
    ; L_NUANCE :  c'est la liste définie au niveau d'un caractère dans laquelle on associe au nom utilisateur
    ;                le nom interne de la nuance.
    ; BUT : transformer la fonction caractéristique FC en lui appliquant la fonction caractéristique de la nuance
    ;       passée en paramètre.
    ; RESULTAT : de même type que le paramètre FC.

    (ifn nuance fc
        (ifn (cassoc nuance l_nuance) fc
          (apply (car (cassoc 'fc (eval (symbol 'c_nuance (car (cassoc nuance  l_nuance))))))
                 fc))))
```

```
(de F_FC_VAL_CONTINU (couple)
    ; COUPLE : est un couple (valeur nuance).
    ; BUT :    retourner la fonction caractéristique associée à la valeur de domaine continu.
    ; RESULTAT : quadruplet (a b c d).

    (cond ((numberp (car couple))   ; la valeur est précise
                (list (car couple) (car couple) 0 0))
          ((symbolp (car couple))
                (if  (oblist 'c_fc (car couple) )
                     ; alors la valeur est une valeur vague et on dispose de sa définition.
                     (f_att_objet (f_acc_objet 'c_fc (car couple)) 'fc)
                     ; sinon erreur
                     (erreur 3 (list (car couple)))))
          ((listp (car couple)) ; la valeur est un intervalle
                (append (car couple) '(0 0)))
          (t (erreur 3 (list (car couple))))))
```

```
(de F_FC_VAL_DISCRET (couple def_dom)
    ; COUPLE : est un couple (valeur nuance)
```

```
; DEF_DOM : est la définition du domaine, c'est soit le nom d'un thésaurus (EX: #:c_reseau:couleur) soit
;                une liste de valeurs dans le cas ou` le domaine n'est pas represente par un thesaurus.
; BUT : retourner la fonction caractéristique associée à la valeur de domaine discret.
; RESULTAT : une liste de couple (VAL COEF).

(cond   ((listp def_dom)
         (if (oblist 'c_fc (car couple) )
             ; alors la valeur est une valeur vague et on dispose de sa définition
           (f_att_objet (f_acc_objet 'c_fc (car couple)) 'fc)
             ; sinon la valeur est précise, on retourne alors la valeur avec  le coef. 1.
           (list (list (car couple) 1))))
        ((atomp def_dom)
         (if (oblist def_dom (car couple) )
             ; alors la valeur appartient au thesaurus associé
           (f_genere_rs (symbol def_dom (car couple)))
             ; sinon erreur
           (erreur 3 (list (car couple)))))
        (t (erreur 3 (list (car couple))))))
```

; FONCTIONS DE FILTRAGE HIERARCHIQUE ET ELEMENTAIRE

```
(de F_TRAITE_UN_CARACTERE (un_caractere)
    ; UN_CARACTERE : (Nom_Caractere <poids>  (Val1 Nuance1) ... )
    ; BUT :  générer la fonction caractéristique associée au paramètre UN_CARACTERE.
    ; RESULTAT : voir résultat de F_GENERE_DISCRET et F_GENERE_CONTINU.

 (let ((type_domaine (F_type_domaine (car un_caractere))))
    (cond ((equal type_domaine 'DISC)  ; domaine discret
                (F_genere_discret un_caractere))
          ((equal type_domaine 'CONT)  ; domaine continu
              (F_genere_continu un_caractere))
          (t (F_erreur 2)))))
```



```
(de F_TRANSFORME (L_pr)
    ; L_PR :   ((<nom_caractere_1> <pds1> (<val1> <nuance1>) ... (<valn> <nuancen>))
    ;                ..........
    ;              (<nom_caractere_p> <pdsp> (<val1> <nuance1>) ...(<valn> <nuancen>)))
    ; BUT : remplace tous les termes nuancés du Portrait Robot L_PR  par les fonctions caractéristiques
    ;         correspondantes.
    ; RESULTAT : une liste de la forme :
    ;     ((<nom_caractere_1> <pds1> <Fc1>) .... (<nom_caractere_p> <pdsp> <Fcp>))
    ;     ou` <Fci> est
    ;     - (<val1> <coef1>) ... (<valn> <coefn>) si Domaine du caractere est DISCRET
    ;     - (<bi1> <bs1> <pg1> <pd1>) .. (idem)  si domaine du caractere est CONTINU

(mapcar 'F_traite_un_caractere L_pr))


(de F_FILTRE_ELEMENTAIRE (probot espece)
    ; PROBOT : ((nom_caractere1 poids1 (val11 nuance11) ... (val1n nuance1n))
    ;                ..............
    ;              (nom_caracterep poidsp (valp1 nuancep1) ... (valpq nuancepq)))
    ; ESPECE : (nom_espece <idem probot>).
    ; BUT : comparer le portrait robot et l'espèce.
    ; RESULTAT : ( TRACE nom_espece poss ness)
    ;                où TRACE : ((nom_caractere pds poss ness) ... )

(F_compare_probot_espece
    (F_transforme probot)
    (cons (car L_espece) (F_transforme (cdr L_espece)))))


(de F_COMPARE_PROBOT_ESPECE (probot espece)
    ; PROBOT : ((<nom_caractere1 poids <FC1> ... <FCn>) ...)  voir F_transforme.
    ; ESPECE : (nom_espece (caract1 pds1 FC1 ... FCm) ... (caractj pdsj FCj1 ...))
    ; BUT :    comparer le portrait_robot transformé par la fonction F_transforme ci_dessus, avec l'espèce
    ;           transformé également par F_transforme.
    ; RESULTAT : ( TRACE nom_espece poss ness)  où TRACE : ((nom_caractere pds poss ness) ... )
```

```
(let ((L_caract_espece (cdr espece)) (tracecomp))
 (setq tracecomp
   ; on prend chaque caractère du portrait robot et on le compare avec le caractère ; correspondant de l'espèce.
  (mapcar
  (lambda (Un_caract_probot)
   (let ((domaine (f_type_domaine (car un_caract_probot)))
      (Un_caract_espece (assoc (car Un_caract_probot) L_caract_espece)))
    (cond      ((and (equal domaine 'DISC)
                (atomp (f_def_domaine (car Un_caract_probot)))
                (f_terme_contraire Un_caract_probot Un_caract_espece))
               ; c'est le cas où il existe des termes contraires dans la description du probot et
               ; dans celle de l'espèce, dans ce cas on retourne comme résultat une poss. et une
               ; ness. nulles

               (list (car Un_caract_espece) (cadr Un_caract_espece) 0 0))
              (t (f_compare_2_caracteres Un_caract_probot Un_caract_espece)))))
   probot))
 (cons    (cons (car espece) tracecomp)
       (cons (car espece) (f_combine_poss_ness (mapcar 'cddr tracecomp))))))
```

```
(de F_COMBINE_POSS_NESS (L_poss_ness)
   ; L_POSS_NESS : ((<poss.1> <ness.1>) ... (<poss.n> <ness.n>))
   ; BUT :    Combiner les mesures de possibilité et de nécessité pour extraire une mesure de
   ;          poss. et une mesure de ness. globales, on utilise l'opérateur Min.
   ; RESULTAT : un couple (<poss> <ness>)

   (apply 'mapcar (cons 'min l_poss_ness)))
```

```
(de F_COMPARE_2_CARACTERES (fc_caract1 fc_caract2)
   ; FC_CARACT1 et FC_CARACT2 sont les FC associées à 2 caractères, elles sont de
   ; l'une une des formes suivantes :
   ;  - (Nom_caract pds (Val1 Coef1) .... (Valn Coefn))  --> Domaine discret
   ;  - (Nom_caract pds (BI1 BS1 PG1 PD1) .....      ) --> Domaine continu
   ; RESULTAT : retourne comme resultat un quadruplet (nom_caractere poids poss ness)
```



```
(let ((type_domaine (F_type_domaine (car fc_caract1))) (l_poss_ness))
  (cond
   ((equal type_domaine 'DISC)
       (list
        (car fc_caract2) ; nom du caract
        (cadr fc_caract2) ; poids du caract ds l'espèce
        (max (F_poss_discret (cddr fc_caract1) (cddr fc_caract2))
             (- 1 (cadr fc_caract1))
             (- 1 (cadr fc_caract2)))
        (max (F_ness_discret (cddr fc_caract1) (cddr fc_caract2))
             (- 1 (cadr fc_caract1))
             (- 1 (cadr fc_caract2)))))
   ((equal type_domaine 'CONT)
   (setq l_poss_ness
         (mapcan
             (lambda (fc)
              (mapcar (lambda (fc1)
                                  (list (max (F_poss_continu fc fc1)
                                             (- 1 (cadr fc_caract1))
                                             (- 1 (cadr fc_caract2)))
                                        (max (F_ness_continu fc fc1)
                                             (- 1 (cadr fc_caract1))
                                             (- 1 (cadr fc_caract2)))))
                          (cddr fc_caract2)))
             (cddr fc_caract1)))
   (cons (car fc_caract2)
         (cons (cadr fc_caract2)
               (F_meilleurs_mesures (car l_poss_ness) (cdr L_poss_ness)))))))
```

```
(de F_IDENTIFIE (pr lf seuil1 seuil2)
   ; PR  : Portrait robot sous forme d'une liste de couple
   ;     (<nom_caractere> (<valeur1> <nuance>) ... (<valeurp> <nuance>))
   ; LF    : Liste de noms d'objets de la hiérarchie des descriptions à partir desquels on commence
   ;         l'identification. EX: (#:c_famille:amanita) cela veut dire que l'identification ne se fera que
```

```
;            parmi les amanites.
; SEUIL1, SEUIL2 : seules les espèces ayant une mesure de possibilité >seuil1  et une mesure de
;                  nécessité >seuil2 seront acceptées
; BUT : identification nuancée.
; RESULTAT : la liste des descriptions classées suivant l'ordre de Pareto.

(f_imp_resultat seuil1 seuil2 (f_classement (f_filtre_hierarchique pr lf ())))


(de F_COMPARE (pr objet lc lcarcalc)
; PR   : Portrait robot
; OBJET : une famille ou une instance de type c_familles avec tous les caractères de PR valués.
; LC   : liste de noms de caractères suivants lesquels PR et OBJET son comparés.
; LCARACT : liste des caractères hérités et déja calculés dans les super_classes ,
;           les éléments de LCARACT sont de la forme ( <car> <pds> <poss> <ness>),
;           <pds> est le poids qui apparait dans les super_classes).
; BUT  : comparer pr avec objet
; RESULTAT : retourner comme résultat la liste suivante :
;      (<trace> <nom_objet> <Mesure_Possibilite MP> <Mesure_Necessite MN>)
;      <trace>:(<nom_obj> (<nom_caract1> <pds1> <MP1> <MN1>) ...
;                         (<nom_caractq> <pdsq> <MPq> <MNq>))
;      <trace> sert pour la trace dans le cas ou` un MPi=0

(lets ((lcsv (f_lcsv objet)) (lncarcalc (mapcar 'car lcarcalc)) (lncc (f_inter lcsv lncarcalc))
       (resultat ()))
  (when lncc
      ; cela veut dire que parmi les caractères hérités il y'en a qui sont redéfinis dans OBJET. Par
      ; conséquent, il faut comparer à nouveau ces caractères. C'est ainsi que sont traitées les exceptions.
      (setq lc (f_union lc lncc))
      (while lncc
          (setq lcarcalc (remove (assoc (car lncc) lcarcalc) lcarcalc))
          (setq lncc (cdr lncc))))
  (setq resultat (f_compare_caractere lc pr objet))
  (cons      (car resultat)
             (cons (cadr resultat)
                   (f_combine_poss_ness (cons (cddr resultat) (mapcar 'cddr lcarcalc)))))))
```



```
(de F_COMPARE_CARACTERE (lcomp pr objet)
; LCOMP : liste de nom de caracte`res suivant lesquels on va comparer pr et objet
; PR, OBJET : voir la fonction f_compare
; BUT  : comparer le portrait robot pr et l'objet (famille ou instance) suivant la liste
;        LCOMP passée en paramètre.
; RESULTAT : idem fonction f_compare

  (f_filtre_elementaire (f_reduire pr lcomp)
                        (cons objet (f_reduire (cassoc 'def (eval objet)) lcomp))))


(de F_FILTRE_HIERARCHIQUE (pr lf lcarcalc)
; PR : portrait robot .
; LF : liste des noms de familles ou d'espèces qu'on va comparer au PR.
; LCARACT : liste des caractères hérités et déja calculés dans les super_classes ,les éléments de
;           LCARACT sont de la forme ( <car> <pds> <poss> <ness>),<pds> est le poids qui
;           apparait dans les super_classes). les éléments de LCARACT sont de la forme ( <car>
;           <pds> <poss> <ness>),<pds> est le poids qui apparait dans les super_classes).
; RESULTAT : une liste de triplets (<nom_famille!espe`ce> <poss> <ness>)

(lets ((objet (car lf)) (lcsv (f_LCSV objet)) (lcpr (mapcar 'car pr)) (resultat ()) (l_inter ())
       (lncarcalc (mapcar 'car lncarcalc)) (lc (f_differ lcpr lncarcalc)))
 (when lf
  (cond
   ((f_inclus lcpr lcsv)
      ; cas ou` l'objet possede tous les caractères du PR avec leurs valeurs dans ce cas on
      ; procède à la comparaison de l'objet et du PR.
    (setq resultat (f_compare pr objet lc lcarcalc))
    (cond
      ((equal (caddr resultat) 0)
      ; mesure de possibilité Nulle dans ce cas on garde une trace
       (setq trace (cons (append (car resultat) lcarcalc) trace))
       (f_filtre_hierarchique pr (cdr lf) lcarcalc))
      (t
        ; cas où la poss. est non nulle : on ajoute la famille en cours dans  le résultat (il faut cependant
        ; calculer les poss. et ness. globales qui tiennent compte de ce qui a déja été calculé dans les
```

```
        ; super_classes de objet...)
        (cons (cdr resultat)
              (f_filtre_hierarchique pr (cdr lf) lcarcalc))))) ; fin (f_inclus ... )

    ((null (f_inter lcpr lcsv))
        ; cas où les caractères de l'objet ne sont pas valués dans ce cas on passe aux sous-familles et aux
        ; instances.
        (append
         (f_filtre_hierarchique
             pr
             (f_filtre_ss_fam (cassoc 'sous_classe (eval objet)) lc)
             lcarcalc)
         (f_filtre_hierarchique pr (cdr lf) lcarcalc)))

    ((setq l_inter (f_inter lcpr lcsv) )
        ; cas où l'objet ne possède qu'une partie des caractères qui sont valués, dans ce cas il faut descendre
        ; aux sous-familles et aux instances. Avant de descendre aux sous-familles et aux instances, on
        ; compare d'abord le PR et l'objet suivant les caractères qu'ils ont en communs, si la mesure de
        ; possibilité résultat de cette comparaison est nulle on ne descend pas aux .. mais on ajoute l'objet
        ; dans la trace.
     (setq resultat (f_compare pr objet l_inter lcarcalc))
     (cond
      ((equal (caddr resultat) 0)
          (setq trace (cons (append (car resultat) lcarcalc) trace))
          (f_filtre_hierarchique pr (cdr lf) lcarcalc)) ; fin (equal ... )
     (t (setq new_pr (f_reduit_aliste pr (f_inter lcpr lcsv)))
       (append (f_filtre_hierarchique pr
                    (f_filtre_ss_fam (cassoc 'sous_classe (eval objet))
                                     (f_differ lcpr (f_union l_inter lncarcalc)))
                    (f_combine (cdar resultat) lcarcalc))
                   (f_filtre_hierarchique pr (cdr lf) lcarcalc)))))))


(de F_REDUIRE (pr l_caract)
    ; PR : portrait robot
    ; L_CARACT : liste de nom de caractères (<nom_caract1> ... <nom_caractp>)
```



```
    ; BUT :   réduire la description du portrait robot juste aux caractères donnés dans la liste
    ;         L_caract passée en paramètre.
    ; RESULTAT : idem PR.

(when pr
 (if (member (caar pr) l_caract) (cons (car pr) (f_reduire (cdr pr) l_caract))
   (f_reduire (cdr pr) l_caract))))


(de F_FILTRE_SS_FAM (l_ss_fam l_carac)
    ; L_SS_FAM : une liste de noms d'espèces ou de familles d'espèces.
    ; L_CARACT : liste de noms de caractères.
    ; BUT : retourne la liste des sous-familles qui ont dans leur LCS tous les caractères du portrait robot.
    ; RESULTAT : idem L_SS_FAM

(ifn l_ss_fam
        ()
        (if (f_inclus l_carac (cassoc 'lcs (eval (car l_ss_fam))))
          (cons (car l_ss_fam) (f_filtre_ss_fam (cdr l_ss_fam) l_carac))
          (f_filtre_ss_fam (cdr l_ss_fam) l_carac))))
```

ANNEXE 3

Exemples d'exécution

Nous présentons ici quelques exemples d'exécution du processus d'identification nuancée. Nous donnerons également quelques exemples de la trace après une étape d'identification nuancée.

```
= 12
?
? ; PORTRAIT ROBOT
? ; --------------
?
? pr
= ((coul_chapeau 1 (grisatre)) (haut_pied .8 (6 environ)) (lieu_recolte 1 (
coniferes)))
?
? (f_identifie pr '(#:c_famille:champignon) 0 0)

 RESULTATS
 ---------

 Nom Especes                                                    Poss.  Ness.
-------------------------------------------------------------------------------
#:c_famille:virgatum.....................................       0.80   0.00
#:c_famille:pardinum.....................................       0.80   0.00
#:c_famille:lyophyllum_decastes..........................       0.80   0.00
#:c_famille:saponaceum...................................       0.80   0.00
#:c_famille:terreum......................................       0.80   0.00
#:c_famille:portentosum..................................       0.80   0.00
#:c_famille:pantherina...................................       0.40   0.00
#:c_famille:sciodes......................................       0.20   0.00
#:c_famille:scalpturatum.................................       0.20   0.00
#:c_famille:vaginata.....................................       0.20   0.00
-------------------------------------------------------------------------------
Nombre d'especes trouvees  :   10
Nombre d'especes affichees :   10
-------------------------------------------------------------------------------
?
? 
```



```
= 12
?
? ; IMPRESSION DE LA TRACE
? ; ----------------------
?
? (f_imp_trace trace)

#:c_famille:columbetta
-----------------------
  (coul_chapeau 1 0 0)
  (haut_pied .8 1 1)
  (lieu_recolte 1 1 1)

? 1

#:c_famille:lepista_irina
--------------------------
  (coul_chapeau 1 0 0)
  (haut_pied .8 .2 .2)
  (lieu_recolte 1 .2 .2)

? 2

#:c_famille:lepista_nuda
-------------------------
  (coul_chapeau 1 0 0)
  (haut_pied .8 1 1)
  (lieu_recolte 1 1 1)

? 3

#:c_famille:hygrophorus_russula
--------------------------------
  (coul_chapeau 1 0 0)
  (haut_pied .8 1 1)
  (lieu_recolte 1 .2 .2)

? 4

#:c_famille:vaccinum
---------------------
  (coul_chapeau 1 0 0)
  (haut_pied .8 .2 .2)
  (lieu_recolte 1 1 1)

? 5

#:c_famille:tricholomopsis_rutilans
------------------------------------
  (coul_chapeau 1 0 0)
  (haut_pied .8 1 .2)
  (lieu_recolte 1 1 1)

? 6
```

```
#:c_famille:abobrunneum
-----------------------
  (coul_chapeau 1 0 0)
  (haut_pied .8 1 .2)
  (lieu_recolte 1 1 1)

? 7

#:c_famille:flavobrunneum
-------------------------
  (coul_chapeau 1 0 0)
  (haut_pied .8 .2 .2)
  (lieu_recolte 1 .2 .2)

? 8

#:c_famille:album
-----------------
  (coul_chapeau 1 0 0)
  (haut_pied .8 1 1)
  (lieu_recolte 1 .2 .2)

? 9

#:c_famille:equestre
--------------------
  (coul_chapeau 1 0 0)
  (haut_pied .8 1 1)
  (lieu_recolte 1 1 1)

? 10

#:c_famille:sulfureum
---------------------
  (coul_chapeau 1 0 0)
  (haut_pied .8 1 .2)
  (lieu_recolte 1 1 1)

? 11

#:c_famille:imbricatum
----------------------
  (coul_chapeau 1 0 0)
  (haut_pied .8 1 1)
  (lieu_recolte 1 1 1)

? 12
```



```
#:c_famille:aurantium
---------------------
  (coul_chapeau 1 0 0)
  (haut_pied .8 1 1)
  (lieu_recolte 1 1 1)

? 13

#:c_famille:calocybe_gambosa
----------------------------
  (coul_chapeau 1 0 0)
  (haut_pied .8 1 1)
  (lieu_recolte 1 .2 .2)

? 14

#:c_famille:sejunctum
---------------------
  (coul_chapeau 1 0 0)
  (haut_pied .8 1 1)
  (lieu_recolte 1 1 1)

? 15

#:c_famille:lepista_personata
-----------------------------
  (coul_chapeau 1 .8 0)
  (haut_pied .8 1 1)
  (lieu_recolte 1 0 0)

? 16

#:c_famille:muscaria
--------------------
  (coul_chapeau 1 0 0)
  (haut_pied .8 .2 .2)
  (lieu_recolte 1 .5 .5)

? 17

#:c_famille:caesarea
--------------------
  (coul_chapeau 1 0 0)
  (haut_pied .8 .2 .2)
  (lieu_recolte 1 .2 .2)

? 18
```

```
#:c_famille:virosa
-------------------
 (coul_chapeau 1 0 0)
 (haut_pied .8 .2 .2)
 (lieu_recolte 1 .2 .2)

? 19

#:c_famille:limacella_guttata
-------------------
 (coul_chapeau 1 0 0)
 (haut_pied .8 1 .2)
 (lieu_recolte 1 1 1)

? 20

#:c_famille:rubescens
-------------------
 (coul_chapeau 1 0 0)
 (haut_pied .8 1 .2)
 (lieu_recolte 1 1 1)

? 21

#:c_famille:verna
-------------------
 (coul_chapeau 1 0 0)
 (haut_pied .8 .2 .2)
 (lieu_recolte 1 1 1)

? 22

#:c_famille:phalloides
-------------------
 (coul_chapeau 1 0 0)
 (haut_pied .8 1 1)
 (lieu_recolte 1 .2 .2)

? 23
= ()
?
```



```
= 12
?
? ; PORTRAIT ROBOT
? ; --------------
?
? pr
= ((coul_chapeau 1 (vert)) (coul_app_sporif .8 (blanc)) (allure_lamelles 1 (
emarginees)) (odeur_nature .8 (farine)))
?
? (f_identifie pr '(#:c_famille:champignon) 0 0)

 RESULTATS
 ---------

 Nom Especes                                                    Poss.  Ness.
-------------------------------------------------------------------------------
#:c_famille:sejunctum.....................................      1.00   0.60
#:c_famille:saponaceum....................................      0.00   0.00
#:c_famille:columbetta....................................      0.35   0.00
#:c_famille:equestre......................................      0.20   0.00
#:c_famille:sulfureum.....................................      0.20   0.00
-------------------------------------------------------------------------------
Nombre d'especes trouvees  :    5
Nombre d'especes affichees :    5
-------------------------------------------------------------------------------
= ------------------------------------------------------------------------------
?
```

```
- 12
?
? ; IMPRESSION DE LA TRACE
? ; ----------------------
?
? (f_imp_trace trace)

#:c_famille:lepista_irina
-------------------------
  (coul_chapeau 1 0 0)
  (coul_app_sporif .8 .6 .2)
  (allure_lamelles 1 1 1)
  (odeur_nature .8 .8 .2)
  (odeur_nature .8 1 1)

? 1

#:c_famille:lepista_nuda
-------------------------
  (coul_chapeau 1 0 0)
  (coul_app_sporif .8 .2 .2)
  (allure_lamelles 1 1 1)
  (odeur_nature .8 1 .2)
  (odeur_nature .8 1 1)

? 2

#:c_famille:hygrophorus_russula
-------------------------
  (coul_chapeau 1 0 0)
  (coul_app_sporif .8 1 .5999999)
  (allure_lamelles 1 1 1)
  (odeur_nature .8 1 1)

? 3

#:c_famille:vaccinum
-------------------------
  (coul_chapeau 1 0 0)
  (coul_app_sporif .8 1 .6)
  (allure_lamelles 1 1 1)
  (odeur_nature .8 1 .6999999)
  (odeur_nature .8 1 1)

? 4

#:c_famille:virgatum
-------------------------
  (coul_chapeau 1 0 0)
  (coul_app_sporif .8 1 .5999999)
  (allure_lamelles 1 1 1)
  (odeur_nature .8 1 1)

? 5
```



```
#:c_famille:tricholomopsis_rutilans
-------------------------
  (coul_chapeau 1 0 0)
  (coul_app_sporif .6 .2 .2)
  (allure_lamelles 1 1 1)
  (odeur_nature .8 1 1)

? 6

#:c_famille:pardinum
-------------------------
  (coul_chapeau 1 0 0)
  (coul_app_sporif .8 1 .6)
  (allure_lamelles 1 1 1.)
  (odeur_nature .8 1 .6999999)
  (odeur_nature .8 1 1)

? 7

#:c_famille:albobrunneum
-------------------------
  (coul_chapeau 1 0 0)
  (coul_app_sporif .8 1 .5999999)
  (allure_lamelles 1 1 1)
  (odeur_nature .8 1 .6999999)
  (odeur_nature .8 1 1)

? 8

#:c_famille:lyophyllum_decastes
-------------------------
  (coul_chapeau 1 0 0)
  (coul_app_sporif .8 1 .6)
  (allure_lamelles 1 .3000001 .3000001)
  (odeur_nature .8 1 1)

? 9

#:c_famille:flavobrunneum
-------------------------
  (coul_chapeau 1 0 0)
  (coul_app_sporif .8 .4 .2)
  (allure_lamelles 1 1 1)
  (odeur_nature .8 1 .6999999)
  (odeur_nature .8 1 1)

? 10
```

```
#:c_famille:album
-----------------------
  (coul_chapeau 1 0 0)
  (coul_app_sporif .8 1 .6)
  (allure_lamelles 1 1 1)
  (odeur_nature .8 1 .6999999)
  (odeur_nature .8 1 1)

? 11

#:c_famille:ustaloides
-----------------------
  (coul_chapeau 1 0 0)
  (coul_app_sporif .8 1 .5999999)
  (allure_lamelles 1 1 1)
  (odeur_nature .8 1 .6999999)
  (odeur_nature .8 1 1)

? 12

#:c_famille:lepista_luscina
-----------------------
  (coul_chapeau 1 0 0)
  (coul_app_sporif .8 1 .6)
  (allure_lamelles 1 1 1)
  (odeur_nature .8 1 .6999999)
  (odeur_nature .8 1 1)

? 13

#:c_famille:imbricatum
-----------------------
  (coul_chapeau 1 0 0)
  (coul_app_sporif .8 1 .6)
  (allure_lamelles 1 1 1)
  (odeur_nature .8 1 1)

? 14

#:c_famille:sciodes
-----------------------
  (coul_chapeau 1 0 0)
  (coul_app_sporif .8 1 .5999999)
  (allure_lamelles 1 1 1)
  (odeur_nature .8 1 1)

? 15
```



```
#:c_famille:orirubens
-----------------------
  (coul_chapeau 1 0 0)
  (coul_app_sporif .8 1 .5999999)
  (allure_lamelles 1 1 1)
  (odeur_nature .8 1 .6999999)
  (odeur_nature .8 1 1)

? 16

#:c_famille:aurantium
-----------------------
  (coul_chapeau 1 0 0)
  (coul_app_sporif .8 1 .6)
  (allure_lamelles 1 1 1)
  (odeur_nature .8 1 .75)
  (odeur_nature .8 1 1)

? 17

#:c_famille:terreum
-----------------------
  (coul_chapeau 1 0 0)
  (coul_app_sporif .8 1 .2)
  (allure_lamelles 1 1 1)
  (odeur_nature .8 1 1)

? 18

#:c_famille:portentosum
-----------------------
  (coul_chapeau 1 0 0)
  (coul_app_sporif .8 1 .5999999)
  (allure_lamelles 1 1 1)
  (odeur_nature .8 1 .6999999)
  (odeur_nature .8 1 1)

? 19

#:c_famille:calocybe_gambosa
-----------------------
  (coul_chapeau 1 0 0)
  (coul_app_sporif .8 1 .2)
  (allure_lamelles 1 1 1)
  (odeur_nature .8 1 .6999999)
  (odeur_nature .8 1 1)

? 20
```

```
#:c_famille:scalpturatum
-----------------------
  (coul_chapeau 1 0 0)
  (coul_app_sporif .8 1 .5999999)
  (allure_lamelles 1 1 1)
  (odeur_nature .8 1 .6999999)
  (odeur_nature .8 1 1)

? 21

#:c_famille:lepista_personata
-----------------------
  (coul_chapeau 1 0 0)
  (coul_app_sporif .8 1 .5999999)
  (allure_lamelles 1 1 1)
  (odeur_nature .8 1 .2)
  (odeur_nature .8 1 1)

? 22

#:c_famille:amanita
-----------------------
  (coul_app_sporif .8 1 .5999999)
  (allure_lamelles 1 0 0)
  (odeur_nature .8 1 1)

? 23
- ()
?
```

NOM DE L'ETUDIANT : MOUADDIB Noureddine

NATURE DE LA THESE : Doctorat de l'Université de NANCY I en Informatique

VU, APPROUVE ET PERMIS D'IMPRIMER

NANCY, le 16 MAI 1989 n° 777

LE PRESIDENT DE L'UNIVERSITE DE NANCY I

M. BOULANGE

Université de Nancy I

Faculté des Sciences **en Informatique de Nancy**

Résumé :

Cette thèse présente une solution globale au problème de l'identification d'un phénomène ou d'un objet mal défini dans un domaine d'application décrit par des connaissances nuancées.

Cette solution comprend trois éléments :

- un modèle de représentation des connaissances nuancées,
- une méthode de détermination des objets ressemblant au phénomène à identifier,
- un processus d'identification dans un système possédant une base de données multimedia.

Le modèle de représentation des connaissances présente les particularités suivantes :

- une ou plusieurs nuances, exprimées en langue naturelle, peuvent être associées à chacune des valeurs prise par un caractère d'un objet,
- à chaque domaine de définition discret de caractère peut être associé un micro-thésaurus dont les liens (généricité, synonymie, opposition) peuvent être munis de coefficients exprimant certaines "distances sémantiques" entre les termes,
- des poids d'importance ou de confiance peuvent être associés à chaque caractère aussi bien dans la description des objets de référence que dans la description du phénomène à identifier.

La méthode d'identification repose sur la théorie des possibilités dont nous avons assoupli l'application en diminuant le nombre de fonctions caractéristiques à fournir, par le spécialiste du domaine d'application, grâce à l'introduction d'heuristiques permettant soit de les générer à partir des micro-thésaurus soit de les calculer à partir d'autres déjà définies par composition ou par transformation

Le processus d'identification permet une identification interactive et progressive au cours de laquelle alternent des phases de filtrage, d'affichage de résultats, d'observation d'images et de consultation de textes. En cas d'échec, nous proposons une stratégie de "retour-arrière" qui s'appuie sur les poids des caractères.

Mots clés :

Théorie des possibilités, Bases de Données, Intelligence Artificielle, Information nuancée,
Division relationnelle, Identification nuancée.