

WALID ALI

**SYSTÈME D'INDEXATION ET DE RECHERCHE DES
DOCUMENTS MULTIMÉDIAS**

Mémoire
présenté
à la faculté des études supérieures
de l'Université Laval
pour l'obtention
du grade de Maître ès Sciences (M.Sc)

Département d'informatique
FACULTÉ DES SCIENCES ET DE GÉNIE
UNIVERSITÉ LAVAL

MAI 2001

© Walid Ali, 2001



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-65397-8

Canada

Résumé

La généralisation de l'utilisation de moyens informatiques dans un nombre toujours croissant de secteurs de l'activité humaine a aujourd'hui pour conséquence la production d'un volume considérable d'informations et de documents sous forme papier ou sous forme électronique. Du fait de l'augmentation continue des capacités de stockage des systèmes informatiques, ces documents sont de plus en plus souvent conservés. En conséquence de cette disponibilité croissante des documents, diverses techniques de recherche et de collecte d'information ont vu le jour. Les systèmes d'indexation et de recherche d'informations qui existent actuellement sont spécialisés suivant les types de l'information à indexer ou à chercher

Dans ce mémoire nous avons exploré les solutions existantes en matière d'indexation de documents multimédias. Cette étude a permis de montrer que la plupart des systèmes existants se basent sur une suite de mots-clés n'ayant pratiquement aucun liens entre eux.

Ensuite, nous avons étudié MPEG7, un langage standard de description de documents multimédias. Il semble en effet prometteur mais n'est pas encore totalement spécifié officiellement.

Ensuite, nous avons envisagé quelques solutions pour bâtir un système d'indexation, en se basant sur les informations des deux phases précédentes. Plusieurs principes se sont avérés possibles et ont été étudiés : l'indexation basée sur les mots-clés a été écartée car tous les modèles existants utilisent ce principe et les résultats n'étaient pas assez convaincants ; l'utilisation des graphes conceptuels semblait beaucoup plus prometteuse, dans le sens où des relations de sens sont constituées entre les mots de la requête. Ce principe permet d'améliorer considérablement la pertinence des documents recherchés. Par contre, la mise en œuvre d'une interface pour traiter l'ensemble des mots et des relations semblait trop complexe du point de vue des usagers. Ainsi, une autre approche a été choisie : l'utilisation des expressions spécifiques à un domaine d'application qui résume l'ontologie servant à l'indexation et la recherche des documents de ce domaine.

L'approche basée sur la notion de concepts/expressions a été développée dans le but de présenter aux indexeurs un processus d'indexation simple et uniforme pour tout type de documents, et aux chercheurs une recherche plus efficace au niveau de l'élimination du bruit dans le résultat de recherche pour une requête donnée. Les interfaces d'indexation et de recherche sont facilement manipulables et elles ne sont pas compliquées. Le point fort de cette approche est qu'elle ne se base pas sur un dictionnaire de mots-clés sans relations entre eux, mais sur une ontologie de concepts/expressions reliés entre eux par différents types de relations. Ces relations servent principalement dans le processus de recherche pour fournir un résultat plus précis et pour éliminer le maximum de bruit.

Au niveau développement logiciel, les modules d'indexation et de recherche des documents ont été développés. Dans une prochaine étape, l'application doit être évaluée par les utilisateurs potentiels.

Québec, Mai 2001

Mr Walid Ali
Étudiant

Mr Bernard Moulin
Directeur de recherche

Dédicace

C'est avec un immense plaisir que je dédie ce travail

A mon cher père pour tous les sacrifices qu'il a consenti pour moi.

A ma chère mère et la perle de mon cœur.

A mes frères pour leur soutien tant moral que matériel.

A ma petite sœur en lui souhaitant beaucoup de chance et de réussite.

A tous mes amis et amies, qu'ils trouvent ici l'expression de mon estime.

A ma famille et à tous ceux que j'aime et qui ont souhaité ma réussite.

Walid Ali.

Remerciement

C'est avec pleine gratitude que je tiens à remercier mon directeur de recherche à l'Université Laval Monsieur Bernard Moulin et mon co-directeur Mr Sylvain Delisle pour leurs conseils judicieux et leurs directives qui m'ont servi de meilleure aide tout au long de la période d'encadrement.

Effectivement je remercie tous les membres de l'équipe de recherche du projet ICEM/SE et tous les membres du Centre de Recherche en Géomatique qui m'ont offert le matériel adéquat pour la rédaction de ce mémoire, ainsi que le réseau GÉOIDE pour son support financier.

Mes remerciements vont également à tous les membres du département d'informatique de l'Université Laval ainsi que tous les enseignants pour la clarté de leur enseignement.

Walid Ali.

Sommaire

Résumé

Liste des tableaux

Liste des figures

Chapitre 1: Introduction générale

| | |
|------------------------------------|---|
| 1-1- Motivations..... | 1 |
| 1-2- Domaine de l'application..... | 3 |
| 1-3 – Problématique..... | 5 |
| 1-4-Objectifs..... | 5 |
| 1-5-La démarche suivie..... | 6 |
| 1-6-Organisation du mémoire..... | 7 |

Chapitre 2: État de l'art de l'indexation et de la recherche des documents

| | |
|---|----|
| 2-1-Introduction..... | 9 |
| 2-2-La recherche d'information..... | 9 |
| 2-2-1 - L'indexation des documents textuels..... | 10 |
| 2-2-1-1- Présentation..... | 10 |
| 2-2-1-2- Les méthodes d'indexation..... | 12 |
| a) Les méthodes statistiques..... | 12 |
| b) Les méthodes linguistiques..... | 14 |
| 2-2-2- La recherche des documents textuels (la mise en correspondance)..... | 16 |
| 2-2-2-1- Présentation..... | 16 |
| 2-2-2-2 - Les modèles de mise en correspondance directe..... | 17 |
| a) Le modèle booléen conventionnel..... | 18 |
| b) Le modèle Vectoriel conventionnel (Vector space Model)..... | 21 |
| c) Le modèle booléen étendu..... | 25 |
| d) Le modèle probabiliste..... | 25 |
| e) Le modèle logique..... | 26 |
| f) Le modèle logique basé sur la logique floue..... | 28 |
| 2-2-2-3- Les modèles de mise en correspondance indirecte | 31 |
| a) La mise en correspondance assistée par des clusters: | |

| | |
|--|-----------|
| | 31 |
| b) La mise en correspondance basée sur les clusters..... | 32 |
| 2-2-2-4- La mise en correspondance mixte..... | 37 |
| 2-2-3- L'indexation et la recherche des documents multimédia..... | 40 |
| 2-3-La recherche des documents sur Internet..... | 42 |
| 2-3-1- La collecte des documents..... | 42 |
| 2-3-2- L'indexation des documents..... | 43 |
| 2-3-3- La recherche des documents..... | 44 |
| 2-3-3-1- Les Différents Modes d'Interrogation..... | 45 |
| 2-3-3-2- Méthodes de Fonctionnement Interne..... | 49 |
| 2-4 Conclusion..... | 50 |

Chapitre 3: L'exploration des solutions existantes

| | |
|---|-----------|
| 3-1- Introduction..... | 51 |
| 3-2- Les systèmes d'indexation et de recherche des documents textuels..... | 52 |
| 3-2-1- Présentation des systèmes d'indexation et de recherche de documents textuels..... | 52 |
| a) ZyIndex..... | 53 |
| b) Spirit..... | 54 |
| c) RetrievalWare..... | 55 |
| d) ConText..... | 56 |
| e) AltaVista (www.av.com)..... | 57 |
| f) Excite et Excite France (www.excite.com et www.excite.fr)..... | 59 |
| g) Google (www.google.com)..... | 60 |
| h) HotBot (www.hotbot.com)..... | 61 |
| i) InfoSeek et InfoSeek France (www.infoSeek.com)..... | 63 |
| j) Northern Light (www.northernlight.com)..... | 65 |
| k) Voilà (www.voila.fr)..... | 66 |
| 3-2-2- Bilan sur les systèmes d'indexation et de recherche des documents textuels..... | 67 |
| 3-3- Contexte et enjeu de l'indexation et de recherche des images..... | 68 |
| 3-3-1- Présentation des systèmes d'indexation et de recherche d'images..... | 69 |
| a) Les systèmes étudiés..... | 70 |
| b)Les différents types de systèmes..... | 70 |
| c) Les termes employés..... | 73 |

| | |
|---|------------|
| 3-3-2- Les fiches des systèmes..... | 74 |
| 3-3-3- Bilan concernant les systèmes d'indexation et de recherche d'images..... | 75 |
| 3-4- Présentation du standard MPEG-7..... | 81 |
| 3-4-1 - Présentation générale du standard MPEG-7..... | 81 |
| 3-4-2 - La terminologie du standard..... | 84 |
| 3-4-3 - Les éléments du standard..... | 85 |
| 3-4-4 - La mét-a-structure de la description des documents dans le standard MPEG-7..... | 87 |
| 3-4-5- Bilan sur MPEG-7..... | 100 |
| 3-5-Conclusion..... | 102 |

Chapitre 4: Les techniques proposées et la technique retenue

| | |
|---|------------|
| 4-1 – Introduction..... | 103 |
| 4-2 – la technique basée sur la notion des mots-clés..... | 103 |
| 4-2 – 1 - Présentation de la technique..... | 103 |
| 4-2 – 2 - Limites de la technique..... | 104 |
| 4- 3 – La technique purement sémantique..... | 108 |
| 4- 3 – 1 - Présentation de la technique..... | 108 |
| 4- 3 – 2 - La représentation des connaissances..... | 109 |
| 4- 3 – 2 – 1 - Les réseaux sémantiques..... | 110 |
| 4- 3 – 2 – 2 - Les graphes conceptuels..... | 111 |
| 4- 3 – 2 – 2 – 1 - Les concepts et les référents..... | 113 |
| 4- 3 – 2 – 2 – 2 - La grille de type de concepts..... | 114 |
| 4- 3 – 2 – 2 – 3 - Les relations conceptuelles..... | 116 |
| 4- 3 – 3 - L'utilisation des graphes conceptuels dans le système: | 119 |
| 4- 3 – 4 - Les ontologies utilisées par le système..... | 129 |
| 4- 3 – 5 - Limites de la technique..... | 131 |
| 4- 4 – La technique basée sur les expressions..... | 132 |
| 4- 4 – 1 – Motivations..... | 132 |
| 4- 4 – 2 - Présentation de la technique..... | 132 |
| 4- 5 - La solution retenue..... | 140 |
| 4- 6 – Conclusion..... | 140 |

Chapitre 5 : La conception de la solution (Volet théorique)

| | |
|---|------------|
| 5- 1 – Introduction..... | 142 |
| 5- 2 - Présentation de la solution..... | 142 |
| 5- 3 - La structure du système d'indexation et de recherche des documents multimédias..... | 142 |

| | |
|---|------------|
| 5- 3 – 1 - L'indexation des documents multimédias..... | 144 |
| 5- 3 – 2 - La recherche des documents multimédia..... | 149 |
| 5- 3 – 2 – 1 - La recherche dans la base de données du système | 149 |
| 5- 3 – 2 – 2 - La recherche des documents sur le réseau Internet | 154 |
| 5- 3 – 3 - La gestion de l'ontologie du système..... | 161 |
| a) Avantage de l'utilisation d'une ontologie..... | 161 |
| b) La structure de treillis..... | 162 |
| c) Les règles de construction..... | 164 |
| d) Les opérations de gestion de la hiérarchie..... | 165 |
| e) Les autres types de relations..... | 172 |
| f) Exemple d'ontologies..... | 173 |
| 5- 4 - La structure de la base de données de l'application..... | 176 |
| 5- 5 – Conclusion..... | 178 |

Chapitre 6 :La réalisation de la solution (Volet pratique)

| | |
|---|------------|
| 6- 1 – Introduction..... | 179 |
| 6- 2 - L'environnement de réalisation de l'application..... | 179 |
| 6- 3 – Conception et construction des interfaces personne-machine | 180 |
| 6- 3 – 1 - La fenêtre principale de l'application..... | 180 |
| a) Les éléments du menu principal de l'application..... | 180 |
| b) Les éléments de la barre d'outils de l'application..... | 183 |
| 6- 3 – 2 - Les écrans du module d'indexation..... | 184 |
| 6- 3 – 3 - Les écrans du module de recherche des documents..... | 188 |
| a) La recherche classique des documents multimédias..... | 188 |
| b) La recherche avancée des documents multimédias..... | 192 |
| c) La recherche des documents sur Internet..... | 197 |
| 6- 3 – 4 - Les écrans du module de gestion de la base de données du système..... | 200 |
| 6- 3 – 5 - Les écrans du module d'administration des ontologies du système..... | 207 |

Chapitre 7 :La réalisation de la solution (Volet pratique)

| | |
|------------------------------------|------------|
| 7- 1 – Introduction..... | 215 |
| 7- 2 – Expérimentation..... | 215 |
| 7- 3 – Évaluation..... | 216 |
| 7- 4 – Évaluation..... | 220 |

| | |
|---|------------|
| Conclusion générale et perspectives..... | 221 |
| Bibliographie..... | 223 |
| Annexe A: Liste des principaux algorithmes | |
| Annexe B: L'ontologie test du système | |

Liste des tableaux

| | |
|---|------------|
| Tableau 3.1 : caractéristiques des systèmes d'indexation et de recherche des images présentés précédemment | 80 |
| Tableau 5.1 : Tableau des moteurs de recherche sur Internet..... | 158 |
| Tableau 5.2 : Tableau des catégories des moteurs de recherche sur Internet..... | 160 |

Liste des figures

| | |
|--|-----|
| Figure 2.1: Le processus d'indexation..... | 11 |
| Figure 2.2: Recherche assistée par les clusters..... | 32 |
| Figure 2.3: Recherche basée sur les clusters..... | 33 |
| Figure 2.4: La mise en correspondance mixte..... | 38 |
| Figure 2.5: Présentation des résultats (Vivisimo)..... | 39 |
| Figure 3.1: Mode d'indexation (RetrievalWare)..... | 55 |
| Figure 4.1: Présentation des résultats (Altavista: Recherche simple)..... | 105 |
| Figure 4.2: Présentation des résultats (Altavista: Recherche avancée).... | 106 |
| Figure 4.3: Présentation des résultats (Copernic)..... | 107 |
| Figure 4.5: Exemple de réseau sémantique simple..... | 110 |
| Figure 4.6: Le graphe conceptuel représentant la phrase : « <i>David est allé à Boston en bus</i> » | 112 |
| Figure 4.7: Un exemple d'une ontologie de concepts..... | 115 |
| Figure 4.8: Les relations conceptuelles primitives..... | 117 |
| Figure 4.9: Les relations conceptuelles ensemble de départ : Starter set | 117 |
| Figure 4.10: Les relations conceptuelles définies : Defined..... | 117 |
| Figure 4.11: Graphe conceptuel représentant le texte de l'exemple précédent..... | 121 |
| Figure 4.12: Fenêtre d'indexation des documents..... | 122 |
| Figure 4.13: Menu de la gestion des objets..... | 122 |
| Figure 4.14: Menu de la gestion des actions..... | 122 |
| Figure 4.15: Menu de la gestion des relations..... | 123 |
| Figure 4.16: Interface d'ajout d'un objet..... | 124 |

| | |
|---|-----|
| Figure 4.17: Interface d'ajout d'une action..... | 125 |
| Figure 4.18: Interface d'ajout d'une relation..... | 128 |
| Figure 4.19: L'ontologie des concepts..... | 130 |
| Figure 4.20: L'ontologie des actions..... | 131 |
| Figure 4.21: L'ontologie des relations..... | 131 |
| Figure 4.22: Formation d'une expression composée à partir de concepts unitaires et des autres expressions..... | 134 |
| Figure 4.23: Indexation basée sur les expressions..... | 135 |
| Figure 4.24: Recherche basée sur les expressions..... | 136 |
| Figure 4.25(a): Indexation basée sur la notion des expressions..... | 138 |
| Figure 4.25 (b): Recherche basée sur la notion des expressions..... | 139 |
| Figure 5.1: La structure du système d'Indexation et de recherche des documents multimédias..... | 143 |
| Figure 5.2: La structure du sous-système de recherche des documents multimédias..... | 143 |
| Figure 5.3 : Le fonctionnement du sous-système d'indexation des documents multimédias..... | 145 |
| Figure 5.4 : Le fonctionnement du sous-système de recherche dans la base de données..... | 150 |
| Figure 5.5 : Le fonctionnement du sous-système de recherche sur Internet..... | 155 |
| Figure 5.6: Exemple de treillis..... | 163 |
| Figure 5.7: Exemple d'incompatibilité..... | 163 |
| Figure 5.8 : Exemple de hiérarchie..... | 166 |
| Figure 5.9 : Ajout d'un élément..... | 167 |
| Figure 5.10 : Ajout d'une relation..... | 168 |
| Figure 5.11 : Ajout d'une relation d'incompatibilité..... | 169 |

| | |
|--|-----|
| Figure 5.12 : Suppression d'une relation hiérarchique..... | 170 |
| Figure 5.13 : Suppression d'une relation hiérarchique..... | 171 |
| Figure 5.14 : Suppression d'une relation d'incompatibilité..... | 172 |
| Figure 5.15 : Suppression d'un élément..... | 173 |
| Figure 6.1 : Menu principal et barre d'outils..... | 180 |
| Figure 6.2 : Menu d'indexation des documents et de gestion de l'ontologie..... | 180 |
| Figure 6.3 : Menu de recherche des documents..... | 181 |
| Figure 6.4 : Menu de gestion des documents..... | 181 |
| Figure 6.5 : Menu de gestion des visualiseurs..... | 181 |
| Figure 6.6 : Menu de gestion des moteurs de recherche sur Internet..... | 182 |
| Figure 6.7 : Menu d'aide de l'application..... | 182 |
| Figure 6.8 : Menu quitter l'application..... | 182 |
| Figure 6.9 : La fenêtre d'ajout des documents..... | 185 |
| Figure 6.10 : Fenêtre d'indexation d'un document (1)..... | 186 |
| Figure 6.11 : Fenêtre d'indexation d'un document (2)..... | 187 |
| Figure 6.12: Fenêtre de recherche des documents..... | 188 |
| Figure 6.13 : Fenêtre de présentation des documents en résultat..... | 189 |
| Figure 6.14 : Fenêtre de recherche (Requête plus spécifique)..... | 190 |
| Figure 6.15 : Fenêtre de recherche (Filtrage du résultat)..... | 191 |
| Figure 6.16 : Fenêtre de recherche (Résultat négatif)..... | 191 |
| Figure 6.17 : Fenêtre de recherche avancée..... | 192 |
| Figure 6.18: Fenêtre de recherche avancée (Les expressions correspondantes à la requête (1))..... | 193 |
| Figure 6.19: Fenêtre de recherche avancée (Les expressions correspondantes à la requête (2))..... | 194 |

| | |
|--|-----|
| Figure 6.20 : Fenêtre de recherche avancée (Présentation des documents : Recherche AND)..... | 195 |
| Figure 6.21 : Fenêtre de recherche avancée (Présentation des documents : Recherche OR)..... | 195 |
| Figure 6.22 : Fenêtre de recherche avancée (Présentation des documents : Recherche OR : Avec filtrage)..... | 196 |
| Figure 6.23: Fenêtre de recherche avancée (Résultat négatif)..... | 196 |
| Figure 6.24 : Recherche sur Internet..... | 197 |
| Figure 6.25 : Résultats de la recherche sur Internet..... | 198 |
| Figure 6.26 : Recherche sur Internet par catégorie de moteurs de recherche..... | 199 |
| Figure 6.27 : Recherche sur Internet par catégorie de moteurs de recherche..... | 199 |
| Figure 6.28 : Fenêtre de la liste des documents..... | 200 |
| Figure 6.29 : Fenêtre de la liste des documents filtrés..... | 201 |
| Figure 6.30 : Ajout d'un document au système..... | 203 |
| Figure 6.31 : Fenêtre de la liste des visualiseurs mise à jour..... | 204 |
| Figure 6.32 : Ajout d'un visualiseur..... | 205 |
| Figure 6.33 : Liste des moteurs de recherche..... | 206 |
| Figure 6.34: Fenêtre de «Login» de l'administrateur de l'ontologie..... | 207 |
| Figure 6.35: Fenêtre d'affichage de l'ontologie du système..... | 208 |

| | |
|---|-----|
| Figure 6.36: Fenêtre d'ajout d'un concept/expression à l'ontologie du système..... | 209 |
| Figure 6.37: Fenêtre de présentation de l'ontologie après mise à jour..... | 210 |
| Figure 6.38 : Fenêtre d'affichage des relations de l'ontologie..... | 211 |
| Figure 6.39: Fenêtre d'ajout des relations à l'ontologie des relations..... | 212 |
| Figure 6.40: Fenêtre d'affichage de l'ontologie des relations mise à jour | 213 |
| Figure 6.41: Fenêtre d'ajout d'une relation entre concepts/expressions | 214 |

Chapitre 1

Introduction générale

Ce premier chapitre introduit de façon générale notre mémoire de maîtrise intitulé « Réalisation d'un Système d'Indexation et de Recherche des Documents Multimédia : SIRDM».

Dans la première section, nous présentons les motivations de notre recherche. La deuxième section présente, brièvement, le domaine de l'application dans lequel s'inscrit ce travail de recherche, ainsi que le projet de recherche dans lequel s'intègre ce travail. La section qui suit présente, en bref, la problématique de notre travail. Ensuite, nous présentons les objectifs que nous avons poursuivis dans ce travail. Après la présentation des objectifs, nous présentons la démarche suivie pour atteindre ces objectifs. Finalement, ce chapitre se conclut par une présentation sommaire de l'organisation du mémoire.

1 – 1 - Motivations:

La généralisation de l'utilisation de moyens informatiques dans un nombre toujours croissant de secteurs de l'activité humaine a aujourd'hui pour conséquence la production d'un volume considérable d'informations et de documents sous forme papier ou sous forme électronique. Par exemple, une réunion peut donner lieu à un
1

compte-rendu sur papier produit à l'aide d'un traitement de texte correspondant donc à un fichier électronique; une opération réalisée par l'intermédiaire d'un guichet automatique produit un enregistrement dans le système informatique de la banque concernée; un simple appel téléphonique correspond à une transaction stockée dans une base de données par l'opérateur de télécommunications...etc [RejmanFaltings 97]. Souvent, les données et les documents ainsi produits, que ce soit sous forme papier ou électronique, ne sont qu'un sous-produit d'une activité dont ils ne constituent pas la finalité principale. Du fait de l'augmentation continue des capacités de stockage des systèmes informatiques, ces documents sont de plus en plus souvent conservés. En conséquence de cette disponibilité croissante des documents, diverses techniques de recherche et de collecte d'information ont vu le jour [RejmanFaltings 97]. Pour accéder à ces différents documents et données stockés, divers services de documentation et de recherche documentaire ont été mis en place. Ces services représentent à l'heure actuelle « un savoir faire en matière de structuration / organisation de l'information et de la connaissance ». Ils fournissent aussi plusieurs méthodes pour chercher et accéder à ces informations. Ils permettent aussi au chercheur un gain de temps considérable dans sa recherche en économisant son passage dans les rayons d'une bibliothèque ou son déplacement dans un centre d'information pour chercher tel document ou telle publication [Trigano 94].

Actuellement, l'information n'est pas seulement textuelle, mais elle peut exister sous plusieurs formats plus ou moins complexes (ce que nous appelons l'information *multimédia*). Vue l'homogénéité de cette information, il faut toujours essayer de perfectionner les systèmes d'indexation et de recherche existants ou bien de créer d'autres systèmes pour supporter ces types d'information.

Les systèmes d'indexation et de recherche d'informations qui existent actuellement sont spécialisés suivant les types de l'information à indexer ou à chercher. Une fois, ces systèmes réalisés, ils restent figés, et ne peuvent manipuler que les types d'informations pour lesquels ils ont été conçus. Dans le domaine de l'informatique, de nouveaux formats d'informations apparaissent tous les jours. Pour cette raison, les chercheurs dans le domaine d'indexation et de recherche des documents, et les développeurs des systèmes ont pensé à créer d'autres approches d'indexation et de

recherche d'information qui peuvent être générales pour tout type ou format d'information.

1 – 2 - Domaine de l'application:

Notre recherche est appliquée dans le cadre d'un projet intitulé «*Interface Cartographique pour l'Exploitation Multidimensionnelle des indicateurs de Santé Environnementale sur le World Wide Web (ICEM / SE)*». Ce projet est sous la responsabilité de plusieurs chercheurs du CRG (Centre de Recherche en Géomatique) et du département d'informatique au sein du laboratoire d'informatique cognitive de l'Université Laval.

L'objectif du projet ICEM/SE est de collecter des informations et des données de natures diverses (santé, environnementale, cartes, documents médiatiques, etc ...) de différentes sources (rapports, documents cartographiques, reportages, sites web, etc...) et de les grouper dans un entrepôt de données, ce que nous appelons «Data warehouse». Ces données disponibles depuis le web seront exploitées à partir d'une interface SOLAP (Spatial On-Line Analytical Processing) qui est utilisée pour l'exploitation des données, répondre aux requêtes spatiales et accéder aux informations sur le contenu des bases de données et des limites d'interprétation des résultats, etc.

Les informations utilisées en santé environnementale sont collectées par différents organismes, dont plusieurs sont externes au réseau de la santé. Elles constituent une documentation de base pour les intervenants régionaux en santé environnementale. Plusieurs de ces informations comportent une dimension spatiale qu'il est important de visualiser. Ainsi, à moyen et long terme, l'intégration des données utilisées dans le domaine de la santé environnementale dans un système d'information à référence spatiale pourrait contribuer à :

- Assurer un accès convivial et opérationnel aux principales banques de données (sanitaires, socio-économiques, sociales, environnementales et

administratives) pour la MSSS (Ministère de la Santé et des Services Sociaux), les régies régionales par le biais des directions de santé publique, de même qu'aux autres partenaires du réseau (Centre de Toxicologie du Québec, et Centre Anti-Poison... etc)

- Uniformiser le traitement des données (validation des données de base, périodes, méthodes d'analyses statistiques) et de maintenir un haut niveau de qualité méthodologique et de crédibilité.
- Produire une représentation cartographique des données concernant la mortalité et morbidité, pour certains problèmes de santé potentiellement reliés à des contaminations environnementales, sur une base régionale (par les régies) et provinciale.
- Rendre accessible au Ministère de la Santé et des Services Sociaux (MSSS), aux régies régionales et à leurs directions de santé publique, diverses informations sur les activités menées par les équipes régionales en santé environnementale, et d'en suivre l'évolution dans le temps.
- Réaliser, de manière économique, l'analyse régionale des caractéristiques environnementales, sociales et sanitaires.
- La réalisation de bilans de l'état de santé de la population en rapport avec les risques biologiques, chimiques et physiques, sur une base régionale (par les régies) et provinciale.
- Répondre rapidement à des demandes courantes comme la vérification sommaire d'agrégats géographiques suspectés dans une région donnée, l'étude des associations pour d'autres problèmes de santé qui présentent des déterminants ou des vecteurs communs, ou la réalisation de bilans thématiques.

Nous remarquons que la quantité d'informations et des données collectée est énorme. En plus, cette information peut se présenter sous plusieurs formats, et elle peut comporter une dimension spatiale. Pour réaliser les objectifs du projet ICEM/SE, il faut bien gérer cette énorme quantité d'informations et de données. Notre travail consiste à concevoir et réaliser un système capable de gérer cette

quantité d'informations, de données et de documents afin que les utilisateur du projet peut y accéder facilement et rapidement.

1 – 3 - Problématique:

Les informations et les données à gérer sont de différents formats et types. Le système à concevoir pour gérer ces informations doit utiliser une approche unique, bien que les informations à gérer soient de formats hétérogènes. Toutes les approches qui existent actuellement en matière d'indexation et de recherche d'information sont spécifiques à des types bien déterminés d'informations et données. En plus, ces approches ne sont pas très efficaces en matière de pertinence des résultats qu'elles présentent. Nous n'avons pas une approche qui peut indexer et chercher toute une panoplie de documents de différents types et formats en utilisant une seule technique.

1 - 4 -Objectifs:

L'objectif général de notre mémoire est de présenter une nouvelle approche d'indexation et de recherche des documents de tous formats. Cette approche doit être indépendante du format ou type de l'information à indexer ou à chercher. Ainsi, elle doit être plus efficace que les approches existantes actuellement en terme d'élimination du bruit¹ du résultat dans la phase de recherche des documents.

Plus spécifiquement, les objectifs visés sont :

1 - Présenter une nouvelle approche qui sera plus efficace que les approches existantes en ce qui concerne l'indexation ou la recherche de l'information.

2 - Présenter une approche indépendante du type ou du format de l'information à indexer ou à chercher. Cette approche doit être la même quel que soit le type ou la format d'information ou du document à indexer ou chercher.

¹ Le bruit est l'ensemble des documents présentés dans le résultat de la requête d'un utilisateur mais qui ne répondent pas à cette requête.

3 - Mettre en œuvre cette approche par un système informatique qui s'intègre dans le projet ICEM/SE. Ce système doit être aisément manipulable par l'utilisateur et doit présenter des résultats plus efficaces que les autres systèmes existants actuellement.

4 - Évaluer cette approche en considérant les réactions des utilisateurs par rapport au système réalisé. Ces utilisateurs ont déjà utilisé des système d'indexation et de recherche classiques.

1 – 5 - La démarche suivie:

Dans un premier temps, nous avons fait une revue de littérature et l'apprentissage des notions fondamentales concernant les domaines pertinents à l'étude (la recherche documentaire, l'indexation, la mise en correspondance...).

L'étude sur la notion de la recherche d'information ou la recherche documentaire nous a permis d'avoir une compréhension des diverses approches existantes en ce qui concerne l'indexation et la recherche des documents.

Une autre étude a été faite sur différents systèmes d'indexation et de recherche existants faite pour analyser la mise en pratique des différentes approches théoriques d'indexation et de recherche des documents. Un bilan concernant ces solutions a été fait. Dans ce bilan nous avons décrit les limites des systèmes existants.

En analysant ces différentes approches mises en œuvre par les systèmes d'indexation et de recherche existants, nous avons constaté des insuffisances évidentes. Aussi nous avons considéré plusieurs techniques visant à combler ces insuffisances. Parmi les techniques que nous avons envisagées, l'une n'a pas été adoptée à cause de la difficulté de sa mise en œuvre, une autre a été rejetée en raison de la difficulté de manipulation par les utilisateurs potentiels. Une troisième technique a été acceptée. Cette dernière approche comble bien les insuffisances des

différentes approches existantes. De plus, elle peut être facilement mise en pratique et aisément manipulée par l'utilisateur.

Finalement, nous avons conçu et réalisé un système informatique d'indexation et de recherche des documents multimédia basé sur la technique retenue.

Enfin, la démarche proposée a été évaluée en considérant les réactions des utilisateurs et des spécialistes en matière d'indexation et de recherche des documents. Nous avons aussi pris en compte quelques remarques et préoccupations pertinentes relevées par ces utilisateurs.

1 – 6 - Organisation du mémoire:

Après ce premier chapitre introductif, nous avons organisé notre mémoire comme suit :

Chapitre2 : Nous présentons l'état de l'art de la recherche d'information dans un fonds documentaire. En premier lieu nous présentons la notion de recherche d'information. Ensuite nous présentons les fonctions d'indexation et de recherche d'information pour les documents textuels et pour les documents multimédia. Enfin, nous montrons comment se fait l'indexation et la recherche sur le réseau Internet qui est considéré comme une gigantesque bibliothèque virtuelle.

Chapitre3 : Nous présentons les solutions que nous avons explorées pendant notre étude bibliographique, afin de déterminer la possibilité d'intégration de l'une des solutions dans notre application. Nous avons présenté aussi la problématique de l'indexation des images et des autres documents non textuels. Nous finissons le chapitre par une présentation du standard MPEG-7 qui s'intéresse à une description standard des documents multimédia.

Chapitre4 : Nous présentons les approches existantes pour l'indexation et la recherche des documents que ce soit pour les systèmes de recherche documentaires

dans des bibliothèques locales ou sur le réseau Internet. Nous présentons aussi les limites de cette approche. Ensuite, nous présentons les différentes approches que nous avons envisagé pour combler les limites de cette approche. Nous présentons aussi les avantages et les inconvénients de chacune de ces approches. Enfin, nous présentons l'approche retenue pour concevoir notre système.

Chapitre5 : Nous présentons tous les aspects théoriques de notre solution. Nous présentons la structure du système que nous avons conçu, celle de l'ontologie sur laquelle il se base, ainsi que la structure de la base de données qu'il utilise.

Chapitre6 : Dans ce chapitre, nous concrétisons toutes nos études théoriques par une application. Nous présentons l'environnement de développement de cette application et ses interfaces personne-machine.

Enfin, nous terminons notre mémoire par une conclusion générale en présentant les limites et les perspectives de notre travail.

Chapitre 2

État de l'art de l'indexation et de la recherche des documents

2 - 1 – Introduction:

Dans ce chapitre, nous faisons une revue bibliographique sur la recherche d'information ou ce que nous appelons aussi la recherche documentaire. En fait, deux processus essentiels de la recherche documentaire sont à présenter: le processus d'*indexation* et celui de la *recherche ou la mise en correspondance*. Au début de ce chapitre nous nous intéressons seulement aux documents textuels. Ensuite, nous présentons comment adapter ces deux processus pour les documents multimédia et les problèmes constatés au niveau de ces types des documents. Enfin, nous présentons comment se fait l'indexation et la recherche des documents sur le réseau Internet et nous clôturons le chapitre par une conclusion.

2 - 2 - La recherche d'information:

Actuellement, la recherche d'information est divisée en deux catégories :

La première catégorie est la recherche d'information structurée (données) dans une base de données (hiérarchique, réseau, relationnelle ou orientée objet...). Ce type de

recherche est appelé aussi «interrogation de base de données», et il est piloté par un système de gestion de base de données [Leloup 98].

La seconde catégorie est la recherche d'informations non structurées (textes, images, vidéos, sons...) dans une base documentaire. Ce type de recherche est appelé aussi «recherche par le contenu» [Leloup 98].

Dans ce chapitre, nous allons nous intéresser seulement à la seconde catégorie, c'est à dire à l'indexation et la recherche des informations non structurées sous ses différentes formes.

La recherche documentaire fait partie de la recherche d'information. La recherche documentaire permet à l'utilisateur de rechercher l'information dans un fond documentaire, c'est à dire dans le contenu d'un ensemble de documents [Lamirel 97].

Deux processus importants sont à distinguer. Le premier concerne la représentation synthétique du contenu des documents du fond documentaire, c'est ce que nous appelons «*l'indexation*». Le second consiste en «*la mise en correspondance*» des requêtes des utilisateurs avec la base documentaire [Leloup 98]. Dans ce qui suit, nous allons présenter différentes techniques d'indexation et de recherche des documents textuels, puis nous verrons les caractéristiques de l'indexation et de la recherche des documents multimédia. Nous allons aussi voir comment ces deux processus se font sur le réseau Internet.

2 - 2 - 1 - L'indexation des documents textuels:

2 – 2 – 1 – 1 – Présentation:

L'indexation consiste à présenter le document par un ensemble d'informations qui résument son contenu d'une manière intelligente pour pouvoir le retrouver facilement et rapidement. Cet ensemble d'information s'appelle *base d'index* [Leloup 10]

98][Lamirel 97]. Pour pouvoir réussir la phase d'indexation, il faut bien décrire le document à indexer [Leloup 98].

L'indexation des documents est une étape très importante dans le processus de recherche. En effet, la qualité de la recherche dépend de la qualité de l'indexation. Une «bonne» indexation devrait permettre de constituer un fichier inversé regroupant les informations pertinentes des documents de base. Pour cela, il faut tout d'abord parvenir à extraire correctement ces informations.

Nous présentons le processus d'indexation par la figure suivante (Voir Figure numéro 2.1):

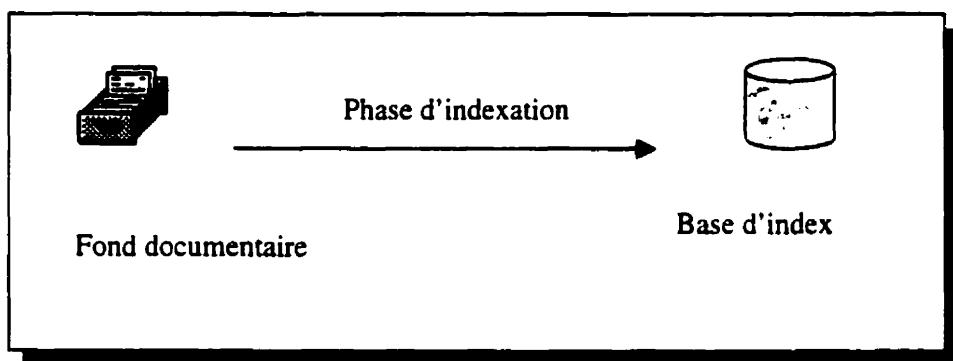


Figure numéro 2.1 : Le processus d'indexation

Exemple :

Prenons comme exemple le fond documentaire suivant:

D1 : «...le cancer des seins chez les femmes de 15 ans et plus dans la région de Québec et la région de Montréal... et précisément à Laval (Montréal dans le Québec)...»

D2 : «...la province d'Ontario se trouve dans le sud du Canada...»

D3 : «...le tourisme dans la province du Québec est un secteur très important....»

D4 : «...l'université Laval se trouve dans la ville de Québec...»

Nous pouvons indexer les documents par les mots soulignés dans ces mêmes documents. L'indexation peut être automatique grâce à un programme qui parcourt les textes et détecte les mots-clés (en se référant à une liste de mots appelée «dictionnaire des mots clés») ou bien manuellement par une personne qui analyse les documents et extrait les mots-clés pertinents.

2 – 2 – 1 – 2 - Les méthodes d'indexation:

Deux axes de recherche ont dominé l'indexation. Les méthodes de ces deux axes s'appliquent seulement aux documents textuels. Le premier axe se base sur la statistique et le deuxième se base sur la linguistique.

a) Les méthodes statistiques:

Le premier axe est une tentative de saisir la sémantique du texte par des méthodes quantitatives ou statistiques. Ces méthodes consistent à détecter, au moyen de divers indices numériques, des mots ou groupes de mots supposés plus «significatifs» que d'autres dans un corpus de données (un exemple d'indice est le nombre d'occurrences d'un mot dans un document textuel). Ces indices ne sont que des poids attribués aux différents termes d'indexation (par exemple la fréquence d'apparition d'un terme dans un document) [Belhassen 99][Kammoun 97].

Plusieurs travaux de recherche ont montré que la seule considération de ces indices n'est pas suffisante pour décider si nous avons bien indexé le document par les bons termes [Belhassen 99][Kammoun 97].

Exemple : Considérons le corpus de documents suivant:

D1 : «...le cancer des seins chez les femmes de 15 ans et plus dans la région de Québec et la région de Montréal... et précisément à Laval (Montréal dans le Québec)...»

D2 : «...la province d'Ontario se trouve dans le sud du Canada...»

D3 : «...le tourisme dans la province du Québec est un secteur très important...»

D4 : «...l'université Laval se trouve dans la ville de Québec...»

Ici, nous pouvons indexer le document D1 par le mots-clé «Québec» car ce mot s'est trouvé deux fois dans le même document. Donc le document sera bien indexé par le mot-clé «Québec» plus que le document D4 qui contient ce mot une seule fois.

La base d'index aura la forme suivante :

...

| | |
|------------|------------|
| Cancer | D1 |
| Sein | D1 |
| Femme | D1 |
| Region | D1 |
| Quebec | D1, D3, D4 |
| Montreal | D1 |
| Laval | D1, D4 |
| Province | D2, D3 |
| Ontario | D2 |
| Sud | D2 |
| Canada | D2 |
| Tourisme | D3 |
| Secteur | D3 |
| Universite | D4 |
| Ville | D4 |

...

L'avantage de cette approche d'indexation qui se base sur des méthodes statistiques est sa facilité de mise en œuvre. Ce type d'indexation peut être totalement

automatique. Mais l'inconvénient est qu'on n'est pas sûr d'obtenir si une «bonne» indexation en se basant sur ce genre d'informations pour indexer le document. Par exemple, l'indice «nombre d'occurrences des mots» peut ne pas être significatif (un mot peut apparaître plusieurs fois mais il n'est pas pertinent par rapport aux sujets traités par le document).

b) Les méthodes linguistiques:

Le second axe de recherche repose sur une méthode plus proche du langage naturel. Cette méthode est une description du contenu du document à partir de l'aspect syntaxique du texte contenu dans ce document (exemple : la construction grammaticale du texte) afin d'extraire des unités du langage (mots, groupes de mots, mots en relations syntagmatique...). Les méthodes linguistiques consistent à déterminer la nature des relations syntaxiques entre les termes du texte. Ces relations traduisent donc une certaine logique nécessaire à la construction du sens de certains textes. Ces méthodes font appel généralement à différents traitements linguistiques : le découpage du texte en mots, l'analyse morphologique de ces mots, la reconnaissance d'expressions idiomatiques, l'analyse syntaxique...[Trigano 94]

L'analyse morphologique vérifie si un mot appartient bien à la langue. Si oui, il lui attribue des propriétés linguistiques qui serviront dans la suite du traitement. Sinon, le mot est erroné et on le corrige ou bien c'est un néologisme et il est ajouté au dictionnaire du système [Trigano 94].

La reconnaissance des expressions idiomatiques permet l'identification en tant qu'unités insécables de suites de mots contenant des séparateurs. Il s'agit d'expressions dont le sens ne peut se réduire simplement du sens des parties [Trigano 94].

L'analyse syntaxique permet la levée des ambiguïtés grammaticales (un même mot peut avoir plusieurs fonctions syntaxiques) et l'établissement des relations de

dépendance entre mots (complément de nom, sujet verbe, verbe complément d'objet direct, etc...) par filtrage morphosyntaxique [Trigano 94].

Ensuite, le document peut être indexé en se basant sur les unités résultantes de tous ces traitements.

Nous pouvons dire que ce type de méthode est encore complètement absent dans les systèmes de recherche documentaires ou les moteurs de recherche existant actuellement [Belhassen 99][Kammoun 97].

Exemple : Considérons le corpus de documents suivant:

D1 : «...le cancer des seins chez les femmes de 15 ans et plus dans la région de Québec et la région de Montréal... et précisément à Laval (Montréal dans le Québec)...»

D2 : «...la province d'Ontario se trouve dans le sud du Canada...»

D3 : «...le tourisme dans la province du Québec est un secteur très important...»

D4 : «...l'université Laval se trouve dans la ville de Québec...»

Ici, nous pouvons indexer le document D1 par toute une unité «Cancer des seins» au lieu du mot-clé «Cancer». Le document D4 peut être bien indexé par l'unité «université Laval» au lieu du mot-clé «Laval» qui peut se référer à une municipalité dans la région de Montréal par exemple.

L'avantage de cette approche d'indexation qui se base sur des méthodes linguistiques se présente clairement dans la phase de recherche des documents. Mais, l'inconvénient majeur est la difficulté de sa mise en œuvre par des systèmes informatique. De plus, ce type d'indexation ne peut pas être totalement automatique.

La base d'index aura la forme suivante :

...

| | |
|--------------------|------------|
| Cancer | D1 |
| Sein | D1 |
| Femme | D1 |
| Region | D1 |
| Quebec | D1, D3, D4 |
| Montreal | D1 |
| Laval | D1, D4 |
| Province | D2, D3 |
| Ontario | D2 |
| Sud | D2 |
| Canada | D2 |
| Tourisme | D3 |
| Secteur | D3 |
| Universite | D4 |
| Ville | D4 |
| «Region Quebec» | D1 |
| «Universite Laval» | D4 |

...

2 – 2 – 2 - La recherche des documents textuels (la mise en correspondance):

2 - 2 - 2 - 1 – Présentation:

La mise en correspondance constitue la fonction centrale de toute recherche documentaire. Celle-ci regroupe l'ensemble des opérations de comparaison effectuées par le système entre une requête ou une question de l'utilisateur et les documents de la base interrogée. La mise en correspondance est appelée aussi «la gestion des recherches» [Lamirel 97][Kammoun 97]. Elle s'effectue par l'analyse de la question 16

de l'utilisateur et la génération d'une requête par le système de recherche, qui l'applique aux fiches d'index. La fonction calcule aussi le nombre de documents trouvés et elle les trie éventuellement selon leur degré de pertinence. Pour pouvoir retrouver les documents qui répondent au mieux à une requête exprimant le besoin de l'utilisateur, plusieurs modèles ont été mis en œuvre. Ainsi, nous pouvons considérer *les modèles booléen et vectoriel conventionnel* qui appartiennent à la catégorie de la mise en correspondance directe, ainsi que d'autres modèles qui en constituent des extensions tels que le *modèle probabiliste*, le modèle *logique*, le modèle *booléen étendu*, le modèle *basé sur la logique floue*. Il existe d'autres modèles qui appartiennent à la catégorie de la mise en correspondance indirecte ou à base de clusters [Leloup 98][Lamirel 97].

Dans ce qui suit nous allons présenter ces différentes catégories de mise en correspondance, ainsi que les modèles de chacune de ces catégories.

2- 2 - 2 – 2 - Les modèles de mise en correspondance directe:

La mise en correspondance directe, comme son nom l'indique, se fait entre la requête de l'utilisateur et la base d'index des documents directement [Leloup 98][Lamirel 97]. Nous y trouvons plusieurs modèles dont quelques uns ont été utilisés depuis longtemps en matière de recherche documentaire. Dans ce qui suit, nous allons présenter en détails les modèles les plus utilisés dans la mise en correspondance directe à savoir *le modèle booléen conventionnel* et *le modèle vectoriel conventionnel*. Les autres modèles comme *le modèle booléen étendu*, *le modèle probabiliste*, *le modèle logique*, *le modèle logique basé sur la logique floue* vont être présentés brièvement [Leloup 98][Lamirel 97]. Ensuite, nous allons présenter les modèles de mise en correspondance indirecte comme *le modèle basé sur les clusters* et *le modèle assisté par les clusters*.

a) Le modèle booléen conventionnel:

Le modèle booléen conventionnel fait partie des tous les premiers modèles utilisés dans la recherche documentaire. Au niveau de ce modèle il faut représenter les documents sous forme d'une conjonction des termes dits «descripteurs» ou «mots clés» que nous voulons rechercher. Ce modèle connecte les termes de la requête par des descripteurs booléens: et (and), ou (or) et sauf (not). Ces opérateurs permettent de structurer la description de la requête et d'exprimer les dépendances mutuelles entre termes et descripteurs. Ces opérateurs booléens peuvent alors être interprétés par des opérateurs ensemblistes sur les listes inverses des documents associés aux termes de la requête. Nous substituons donc l'opérateur booléen «et» (And) par l'intersection, l'opérateur «ou» (or) par l'union, et l'opérateur «sauf» (not) par l'opérateur de complémentarité dans l'ensemble des documents [Leloup 98][Lamirel 97] [Schweighofer 99].

- * L'opérateur « et (and) » est utilisé pour associer un contexte à des critères de recherche très généraux. Il s'agit de définir un concept complexe par une conjonction de sous-concepts.
- * L'opérateur « ou (or) » est utilisé pour définir des classes de synonymes ou quasi-synonymes.
- * L'opérateur « sauf (not) » a le même type d'effet que l'opérateur « et (and) », mais agit sur le résultat déjà obtenu à l'aide d'autres critères de recherche. Cet opérateur peut engendrer un coût non raisonnable (en terme de taille de résultat et en terme de temps de calcul). De plus il n'a pas une interprétation claire.

Avantages :

- Simplicité conceptuelle et simplicité de mise en œuvre. En fait, ce modèle est très simple, il s'appuie sur l'algèbre booléenne et les opérations ensemblistes correspondantes [Leloup 98][Lamirel 97].
- Plus adapté à une interrogation en ligne de fonds documentaires de taille importante [Leloup 98][Lamirel 97].

Inconvénients :

- Aucune valeur de pertinence des descripteurs n'est prise en compte. Ceci veut dire que nous connaissons si le document répond à la requête de l'utilisateur, mais nous ne savons pas s'il est peu, ou très pertinent. En fait, ce modèle ne peut pas distinguer un document qui comprend quelques termes de la requête de celui qui contient plusieurs termes [Leloup 98][Lamirel 97].
- Les documents fournis en réponse à une requête ne sont pas ordonnés par ordre de pertinence [Leloup 98][Lamirel 97].

Exemple : Prenons le corpus des documents suivant :

D1 : «...le cancer des seins chez les femmes de 15 ans et plus dans la région de Québec et la région de Montréal... et précisément à Laval (Montréal dans le Québec)...»

D2 : «...la province d'Ontario se trouve dans le sud du Canada...»

D3 : «...le tourisme dans la province du Québec est un secteur très important...»

D4 : «...l'université Laval se trouve dans la ville de Québec...»

La base d'index aura la forme suivante :

...

| | |
|------------|------------|
| Cancer | D1 |
| Sein | D1 |
| Femme | D1 |
| Region | D1 |
| Quebec | D1, D3, D4 |
| Montreal | D1 |
| Laval | D1, D4 |
| Province | D2, D3 |
| Ontario | D2 |
| Sud | D2 |
| Canada | D2 |
| Tourisme | D3 |
| Secteur | D3 |
| Universite | D4 |
| Ville | D4 |

...

Supposons que l'utilisateur a fourni les trois requêtes suivantes :

R1: «Québec»

R2: «Tourisme Quebec»

R3: «Universite Laval»

Les documents résultats de ces requêtes sont les suivants:

«Québec» : D1 + D3 + D4

«Tourisme Quebec» : $(D1+ D3+D4) \cap D3 = D3$

car D1 et D2 ne parlent pas du «tourisme».

«Universite Laval» : $(D4) \cap (D1+D4) = D4$

car D1 ne parle pas de «l'Université Laval».

Nous remarquons ici que ce principe de mise en correspondance ou de recherche n'est pas trop efficace en matière d'efficacité de résultat. Il fournit des documents qui peuvent ne pas répondre aux besoins de l'utilisateur. Ces documents constituent ce que nous appelons «du bruit».

b) Le modèle Vectoriel conventionnel (Vector space Model):

Ce modèle est l'un des modèles les plus anciens dans la recherche documentaire. Il représente une alternative au modèle booléen. Ce modèle introduit des poids de pertinence associés aux termes descripteurs en utilisant différentes fonctions de mise en correspondance [Leloup 98][Lamirel 97] [Schweighofer 99].

Ce modèle se base sur la notion de vecteurs. Il prend comme hypothèse que chaque document peut être représenté par un vecteur dont les coordonnées non nulles correspondent aux termes descripteurs de ce dernier. Ainsi, la comparaison de deux documents s'opère en comparant les vecteurs associés à chacun d'eux. D'une manière similaire, les requêtes proposées par l'utilisateur sont représentées par des vecteurs dans un espace vectoriel. Les coordonnées non nulles correspondent aux termes descripteurs du document ou de la requête [Leloup 98][Lamirel 97] [Schweighofer 99].

Les concepts de base de ce modèle sont définis comme suit:

D : Une collection des documents : $D_1, D_2, \dots, D_i, \dots, D_n$

T : L'univers des termes des descripteurs des documents d'une collection D .

$V(T)$: L'espace vectoriel de dimension $|T| = n \in \mathbb{N}$ avec $|T|$ défini comme le nombre d'éléments contenus dans l'univers T .

V_i : Le vecteur des termes descripteurs du document D_i .

W_{ik} : La valeur du poids de pertinence (weight) du terme descripteur T_k dans le document D_i .

Si T_k ne fait pas partie des termes descripteurs du document D_i , W_{ik} devra être

considéré comme nul. Si T_k fait partie des termes descripteurs du document D_i , W_{ik} pourra être considéré soit comme la valeur de l'unité (1). Étant donné que le nombre de coordonnées non nulles dans le vecteur de description d'un document est en général très inférieur à la dimension du vecteur V_i , nous choisissons souvent de représenter ce vecteur sous forme résumée par une liste de doublets ou couples de la forme (descripteur, valeur), ainsi le vecteur qui représente le document D_i sera le suivant: $V_i = ((t_1, W_{i1}), (t_2, W_{i2}), \dots, (t_p, W_{ip}))$ [Leloup 98] [Schweighofer 99].

Une requête est également représentée par un vecteur similaire à celui du document: Un vecteur dont les coordonnées non nulles représentent les critères de recherche (les mots de la requête). Les valeurs de ces coordonnées représentent le degré d'importance (ou d'intérêt de l'utilisateur) pour celles-ci. Nous représentons ainsi la requête sous la forme résumée d'une liste de doublets ou de couples (critère, valeur) comme suit : $VR = ((C_1, V_1), (C_2, V_2), \dots, (C_m, V_m))$ [Leloup 98][Schweighofer 99].

Les documents et les requêtes sont représentés dans ce modèle d'une manière unique par des vecteurs. Mettre en correspondance une requête et un ou plusieurs documents revient à faire des calculs mathématiques entre les vecteurs représentant la requête et ceux représentant les documents. Ce calcul aboutit à des mesures et indicateurs de mise en correspondance comme la «similarité» entre requête et document (ou entre documents). Un autre indicateur qui est plus connu est «le produit scalaire» des vecteurs en question qui donne une idée de la similarité entre la requête et les documents (ou entre les documents eux-mêmes) [Leloup 98][Schweighofer 99].

Avantages:

- Ce modèle est simple à mettre en œuvre [Leloup 98][Schweighofer 99].
- En utilisant toutes ces mesures de similarité entre les vecteurs des documents et ceux des requêtes, il est possible d'obtenir de véritables classements des pertinences des documents [Leloup 98][Schweighofer 99].

Inconvénients:

- Absence de lien entre les descripteurs de la requête (lien logique ou autre lien) [Leloup 98][Schweighofer 99].

Exemple : Prenons le corpus de documents suivant:

D1 : «...le cancer des seins chez les femmes de 15 ans et plus dans la région de Québec et la région de Montréal... et précisément à Laval (Montréal dans le Québec)...»

D2 : «...la province d'Ontario se trouve dans le sud du Canada...»

D3 : «...le tourisme dans la province du Québec est un secteur très important...»

D4 : «...l'université Laval se trouve dans la ville de Québec...»

En utilisant la modélisation vectorielle, la base d'index aura la forme suivante :

| t_k / D | D1 | D2 | D3 | D4 | |
|------------|----|----|----|----|----------|
| | | | | | |
| Cancer | 1 | 0 | 0 | 0 | |
| Sein | 1 | 0 | 0 | 0 | |
| Femme | 1 | 0 | 0 | 0 | |
| Region | 1 | 0 | 0 | 0 | |
| Quebec | 1 | 0 | 1 | 1 | |
| Montreal | 1 | 0 | 0 | 0 | |
| Laval | 1 | 0 | 0 | 1 | |
| Province | 0 | 1 | 1 | 0 | $V(T)$ |
| Ontario | 0 | 1 | 0 | 0 | W_{ik} |
| Sud | 0 | 1 | 0 | 0 | |
| Canada | 0 | 1 | 0 | 0 | |
| Tourisme | 0 | 0 | 1 | 0 | |
| Secteur | 0 | 0 | 1 | 0 | |
| Universite | 0 | 0 | 0 | 1 | |
| Ville | 0 | 0 | 0 | 1 | |

Nous présentons les requêtes des utilisateurs sous forme de vecteurs. Par exemple la requête «Tourisme Quebec» se présente comme suit

| V_m / C_i | Tourisme | Quebec |
|-------------|----------|--------|
| | | |
| Cancer | 0 | 0 |
| Sein | 0 | 0 |
| Femme | 0 | 0 |
| Region | 0 | 0 |
| Quebec | 0 | 1 |
| Montreal | 0 | 0 |
| Laval | 0 | 0 |
| Province | 0 | 0 |
| Ontario | 0 | 0 |
| Sud | 0 | 0 |
| Canada | 0 | 0 |
| Tourisme | 1 | 0 |
| Secteur | 0 | 0 |
| Universite | 0 | 0 |
| Ville | 0 | 0 |
| | | |

Maintenant, pour savoir quel document répond mieux à cette requête, nous pouvons calculer par exemple «le produit scalaire» entre le vecteur de la requête et tous les vecteurs des documents. Le document qui dispose du produit scalaire le plus élevé avec le vecteur de la requête est le document le plus similaire et le plus pertinent à cette requête.

Comme exemple de système qui implante ce modèle, nous avons le système de recherche sur Internet «AltaVista : <http://www.altavista.com> ou <http://www.av.com>».

c) Le modèle booléen étendu:

Ce modèle a été proposé comme une alternative au modèle booléen et au modèle vectoriel conventionnel pour remédier aux inconvénients de ces deux modèles. Ce modèle n'a pas connu beaucoup d'applications pratiques. Pourtant celui-ci représente un cadre de comparaison théorique intéressant entre différents modèles conventionnels de mise en correspondance [Leloup 98][Lamirel 97].

Ce modèle est un cas particulier des techniques statistiques vectorielles et probabilistes. Le principe est d'une part de conférer aux termes ou descripteurs de recherche de l'équation booléenne des poids, et d'autres part, d'interpréter les opérateurs de l'équation comme des distances entre requêtes et documents [Leloup 98][Lamirel 97]. Il s'agit là d'un compromis entre une technique d'appariement exact qui est la technique booléenne, et une technique d'appariement approché qui est la technique statistique [Leloup 98][Lamirel 97].

d) Le modèle probabiliste:

Bien qu'il repose sur les bases théoriques les plus solides et qu'il ait subi de nombreuses améliorations, ce modèle a été pratiqué par quelques communautés de recherche d'information à cause de la difficulté de sa mise en œuvre [Leloup 98][Schweighofer 99].

Dans ce modèle on part du principe que les termes ou les descripteurs sont distribués différemment dans les documents pertinents et dans les documents non pertinents. On décide qu'un document est pertinent ou non pour une requête en se basant sur un calcul de probabilité dit de pertinence. Dans le calcul de cette probabilité, nous appliquons quelques théorèmes de la théorie probabiliste comme le «théorème de Bayes». Nous notons que la pertinence d'un document par rapport à une requête donnée est dépendante de celles des autres documents pour la requête [Leloup 98][Schweighofer 99].

De plus, ce modèle propose un classement des documents par ordre de pertinence. Ce classement de pertinence des documents pour une requête est optimal, à partir du moment où il est possible de faire le meilleur usage possible de l'information disponible afin de bien estimer les probabilités de pertinence [Leloup 98][Schweighofer 99].

e) Le modèle logique:

Dans ce modèle, un document est représenté comme une conjonction logique de termes ou mots-clés non pondérés [Leloup 98][Schweighofer 99]. Dans ce modèle, nous représentons un document comme suit :

$$d = t_1 \wedge t_2 \wedge \dots \wedge t_n$$

Une requête est une expression logique quelconque de termes. Nous pouvons utiliser les opérateur «et» (\wedge), «ou» (\vee) et «non» (\neg). Nous pouvons représenter une requête dans ce modèle comme suit:

$$q = (t_1 \wedge t_2) \vee (t_3 \wedge \neg t_4)$$

Pour qu'un document corresponde à une requête, il faut que l'implication suivante soit valide:

$$d \Rightarrow q.$$

Cette évaluation peut être aussi définie de la façon suivante: Un document est représenté comme un ensemble de termes, et une requête comme une expression logique de termes. La correspondance $R(d, q)$ entre une requête et un document est déterminée de la façon suivante:

$$R(d, t_i) = 1 \text{ si } t_i \in d; 0 \text{ sinon.}$$

$$R(d, q_1 \wedge q_2) = 1 \text{ si } R(d, q_1) = 1 \text{ et } R(d, q_2) = 1; 0 \text{ sinon.}$$

$$R(d, q_1 \vee q_2) = 1 \text{ si } R(d, q_1) = 1 \text{ ou } R(d, q_2) = 1; 0 \text{ sinon.}$$

$$R(d, \neg q_1) = 1 \text{ si } R(d, q_1) = 0; 0 \text{ sinon.}$$

Nous pouvons remarquer plusieurs problèmes dans ce modèle :

1. La correspondance entre un document et une requête est soit 1, soit 0. En conséquence, le système détermine un ensemble de documents non-ordonnés comme réponse à une requête. Il n'est pas possible de dire quel document est plus pertinent qu'un autre. Cela crée beaucoup de problèmes aux usagers, car ils doivent encore fouiller dans cet ensemble de documents non-ordonnés pour trouver des documents qui les intéressent. C'est difficile dans le cas où beaucoup de documents répondent aux critères de la requête [Leloup 98][Schweighofer 99].

2. Tous les termes dans un document ou dans une requête étant pondérés de la même façon simple (0 ou 1), il est difficile d'exprimer qu'un terme est plus important qu'un autre dans leur représentation. Ainsi, un document qui décrit en détail "informatique", mais mentionne un peu "commerce" se trouve être représenté par la liste {informatique, commerce} dans laquelle les deux termes deviennent aussi importants l'un que l'autre. Cela ne correspond pas à ce que nous souhaitons avoir [Leloup 98][Schweighofer 99].

3. Le langage d'interrogation est une expression quelconque de la logique de propositions (un terme étant une proposition). Cela offre une très grande flexibilité aux usagers pour exprimer leurs besoins. Cependant, un problème pratique est que les usagers manipulent très mal les opérateurs logiques, les mots "et" et "ou" ne correspondent pas tout à fait aux opérateurs logique \wedge et \vee . Par exemple, quelqu'un qui cherche des documents en tapant les mots «cancer de la peau» «cancer des seins» peut en réalité vouloir chercher des documents traitant soit du «cancer de la peau» soit du «cancer des seins». Ici, la juxtaposition des mots doit plutôt être traduite par un OU (\vee). En partie à cause de cela, les expressions logiques données par un usager correspondent souvent mal à son besoin. La qualité de la recherche souffre en conséquence [Leloup 98][Schweighofer 99].

Il faut cependant remarquer que ce modèle logique standard n'est utilisé que dans très peu de systèmes de nos jours.

f) Le modèle logique basé sur la logique floue:

Cette extension au modèle logique standard vise à tenir compte de la pondération des termes dans les documents. Du côté requête, elle reste toujours une expression logique classique [Leloup 98][Lamirel 97].

Avec cette extension, un document «d» est représenté comme un ensemble de termes pondérés comme suit:

$$d = \{ \dots, (t_i, a_i), \dots \}$$

Avec :

t_i : Terme.

a_i : Pondération.

L'évaluation d'une requête peut prendre plusieurs formes. L'une d'elles est la suivante:

$$R(d, t_i) = a_i$$

$$R(d, q_1 \wedge q_2) = \min(R(d, q_1), R(d, q_2)).$$

$$R(d, q_1 \vee q_2) = \max(R(d, q_1), R(d, q_2)).$$

$$R(d, \neg q_1) = 1 - R(d, q_1).$$

Dans cette évaluation, les opérateurs logiques «And» \wedge et «Or» \vee sont évalués par «min» et «max» respectivement. C'est une des évaluations classiques proposées par L. Zadeh dans le cadre des ensembles flous [KraftBuell 83][Radecki 79][SaltonFoxAH 83][Zadeh 65]. Cependant, cette évaluation n'est pas parfaite. Par exemple, nous n'avons pas $R(d, q \wedge \neg q) \equiv 0$ et $R(d, q \vee \neg q) \equiv 1$. Ainsi, beaucoup d'autres formes d'évaluation ont été proposées. Une des formes est l'évaluation Lukasiewicz [WallerKraftH 79] qui est la suivante:

$$R(d, t_i) = a_i$$

$$R(d, q_1 \wedge q_2) = R(d, q_1) * R(d, q_2).$$

$$R(d, q_1 \vee q_2) = R(d, q_1) + R(d, q_2) - R(d, q_1) * R(d, q_2).$$

$$R(d, \neg q_1) = 1 - R(d, q_1).$$

Dans cette évaluation, nous voyons que les deux parties d'une conjonction ou d'une disjonction contribuent de façon symétrique, contrairement à celle de Zadeh [KraftBuell 83][Radecki 79]. Cependant, elle a le même problème qui est $R(d, q \wedge \neg q) \neq 0$ et $R(d, q \vee \neg q) \neq 1$. En plus, $R(d, q \wedge q) \neq R(d, q) \neq R(d, q \vee q)$.

Si nous comparons ces extensions avec le modèle logique standard, il est assez facile de voir les avantages. Le plus important est qu'on peut mesurer le degré de correspondance entre un document et une requête dans $[0, 1]$. Ainsi, nous pouvons ordonner les documents dans l'ordre décroissant de leur correspondance avec la requête. L'usager peut parcourir cette liste ordonnée et décider où s'arrêter. Au niveau de la représentation, nous avons également une représentation plus raffinée. Nous pouvons exprimer dans quelle mesure un terme est important dans un document [KraftBuell 83][Radecki 79].

Ces évaluations ont été proposées à la fin des années 1970 et au début des années 1980. Maintenant, ces extensions sont devenues standards: la plupart des systèmes booléens utilisent un de ces modèles étendus. Parmi les extensions proposées, nous citons «la reformulation de la requête de l'utilisateur» et «L'association d'une importance aux termes de la requête de l'utilisateur».

a) Reformulation de la requête

Il a été observé qu'une requête qui est une longue conjonction est très difficile à satisfaire. Dans bien des cas, la réponse est vide. À l'autre extrême, une longue disjonction est très facile à satisfaire. Nous aurons alors beaucoup de documents en réponse. C'est surtout pour résoudre le premier problème que nous voulons reformuler la requête initiale [KraftBuell 83][Radecki 79].

Dans le cas d'une longue requête en conjonction, si un document satisfait la plupart des termes de la requête, nous pouvons penser qu'il satisfait en partie le besoin de l'usager. Au lieu de ne donner aucune réponse à l'usager, il vaut mieux donner un ensemble de documents partiellement satisfaisants. Ainsi, le processus de reformulation consiste à examiner le nombre de documents en réponse. S'il est très

faible, on peut assouplir la requête initiale en supprimant un terme [KraftBuell 83][Radecki 79]. Par exemple, soit une requête initiale $q = t_1 \wedge t_2 \wedge t_3 \wedge \dots \wedge t_{10}$. Si aucun document n'a été trouvé pour cette requête, nous pouvons l'assouplir en la suivante:

$$\begin{aligned} q' &= (t_2 \wedge t_3 \wedge t_4 \wedge \dots \wedge t_{10}) \vee \\ &(t_1 \wedge t_3 \wedge t_4 \wedge \dots \wedge t_{10}) \vee \\ &\dots \\ &(t_1 \wedge t_2 \wedge t_3 \wedge \dots \wedge t_9) \end{aligned}$$

Cette reformulation peut continuer si nécessaire.

Cette façon de faire n'est pas unique. Une autre façon possible de reformuler consiste à supprimer le terme qui est le plus difficile à satisfaire (celui qui correspond au moins de documents). Nous pouvons en imaginer encore d'autres. En effet, ces assouplissements sont justifiés seulement par des besoins pratiques. Ils n'ont pas besoin de justification théorique [KraftBuell 83][Radecki 79].

b) Associer une importance aux termes de la requête

Certains auteurs suggèrent d'associer une importance à chaque terme de la requête pour que l'usager puisse différencier des termes très importants de ceux qui le sont moins. Nous pouvons voir certaines propositions dans les articles de Waller, Kraft et Reddecki [Leloup 98][Schweighofer 99] [KraftBuell 83][Radecki 79] [Zadeh 65].

Cependant, ces propositions n'ont jamais gagné du terrain dans la pratique. La raison principale est qu'il est difficile de comprendre le sens de la pondération associée à un terme. Par exemple, pour une expression comme (t, b) dans une requête, nous pouvons comprendre que b est un seuil, b est un facteur de multiplication (i.e. son évaluation est celle de t multipliée par b), ou un facteur de division. Il n'y a jamais eu de consensus là-dessus. Nous avons mentionné qu'il est déjà difficile pour un usager moyen d'utiliser correctement les opérateurs logiques. Il est impensable qu'il puisse mettre une pondération correcte en plus [Leloup 98][Schweighofer 99] [KraftBuell 83][Radecki 79] [Zadeh 65].

2- 2 - 2 - 3 - Les modèles de mise en correspondance indirecte:

Au niveau de la mise en correspondance indirecte, la correspondance se fait entre la requête de l'utilisateur et des classes de documents appelées *clusters*. Nous trouvons deux catégories de mise en correspondance : la mise en correspondance assistée par les clusters et la mise en correspondance basée sur les clusters. Ces deux types de mise en correspondance sont utilisés dans les systèmes d'indexation et de recherche thématiques et non à base de mots-clés. Ces deux catégories sont présentées dans les sous-sections suivantes [Leloup 98][Schweighofer 99].

a) La mise en correspondance assistée par des clusters:

D'après Lamirel [Lamirel 97] ce type de recherche s'effectue en deux étapes :

• Etape 1: Mise en correspondance entre requête et l'ensemble des clusters (classes de documents) : Cette étape permet de sélectionner le cluster le plus pertinent par une mise en correspondance entre les profils (le thème ou le domaine) correspondant aux différents clusters. Si on utilise la mise en correspondance basée sur le modèle vectoriel, il s'agit de la mise en correspondance entre le vecteur profil de chaque cluster et le vecteur requête [Lamirel 97].

• Etape 2: Mise en correspondance entre requête et l'ensemble des documents du cluster choisi. Dans cette étape on cherche les documents appartenant au cluster trouvé et répondant le mieux aux besoins d'information de la requête. Ces documents seront classés par ordre de pertinence si on utilise un modèle de mise en correspondance qui permet de calculer la pertinence des documents [Lamirel 97].

Cette approche est facile à mettre en œuvre car elle utilise un seul niveau du cluster, elle est appelée aussi approche «mono-niveau». Elle est présentée par la figure suivante (Voir Figure numéro 2.2).

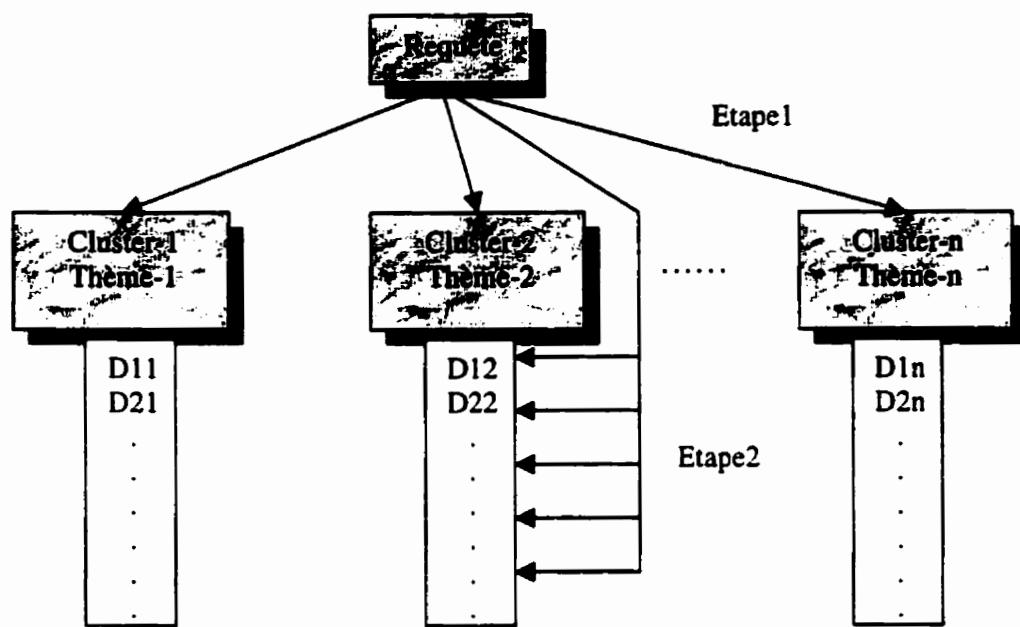


Figure numéro 2.2: Recherche assistée par les clusters

Avantages:

- Cette approche est facile à mettre en œuvre car elle comporte un seul niveau [Leloup 98].

Inconvénients :

- Cette approche utilise un seul niveau, ce qui restreint sa mise en œuvre en pratique [Leloup 98].
- Le document stocké sous un cluster (classe) peut ne pas traiter exactement du thème de la classe, mais nous avons été obligés de l'insérer sous cette classe car nous avons un seul niveau [Leloup 98].

b) La mise en correspondance basée sur les clusters:

Ce type de recherche est basé sur l'utilisation d'une classification hiérarchique des documents qui forment ainsi un arbre de recherche. Cette classification définit un arbre de clusters à n niveaux. Les clusters-feuilles représentent les documents élémentaires. Le cluster-racine, de niveau 0 dans l'arbre, contient l'ensemble des documents. Dans ce type de classification l'ensemble des documents rattachés à un

cluster de niveau k dans l'arbre des clusters, est nécessairement inclus dans l'ensemble des documents rattachés à son cluster-père de niveau k-1, exception faite pour le cluster racine qui n'a pas de cluster-père. Les documents les plus pertinents pour une requête sont considérés comme étant ceux appartenant au cluster qui a lui-même été déterminé comme le plus pertinent pour cette requête, ceci en parcourant l'arbre de recherche à partir de sa racine [Leloup 98]. Cette stratégie est plus difficile à mettre en œuvre car elle utilise une classification à n niveaux. Elle est présentée dans la figure suivante (Voir Figure numéro 2.3).

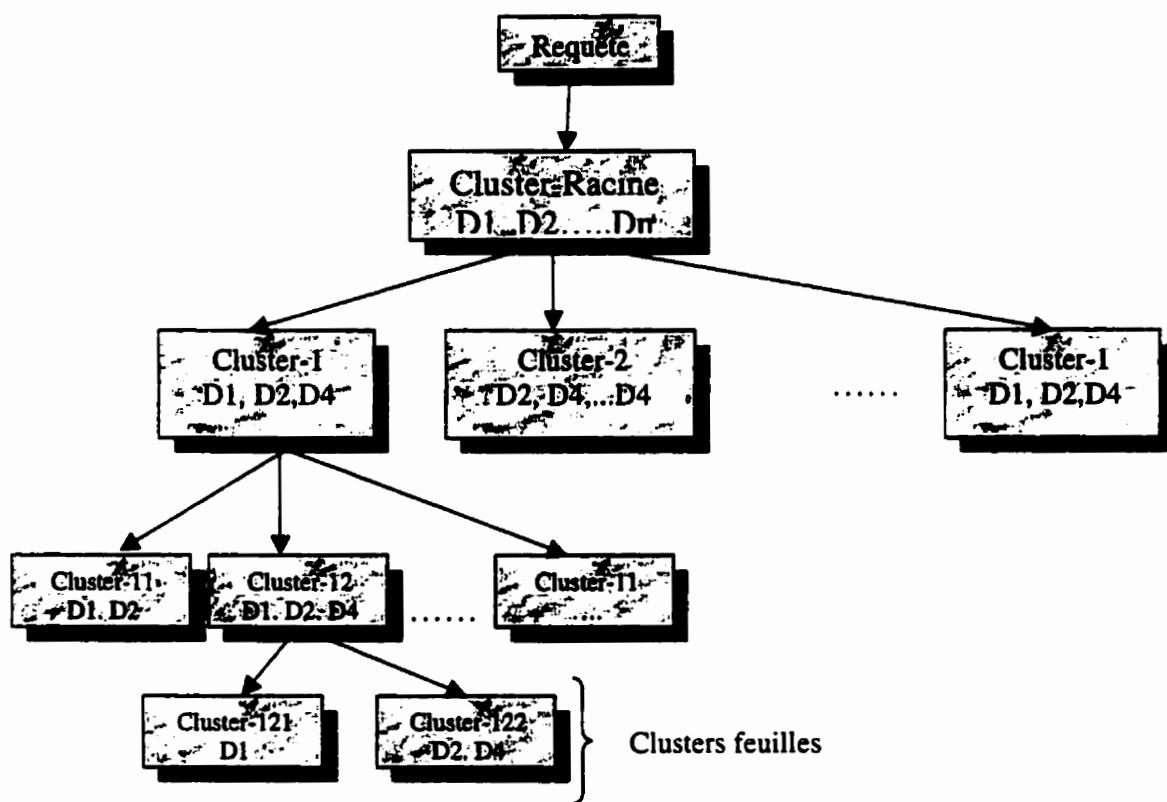


Figure numéro 2.3 : Recherche basée sur les clusters

La mise en correspondance ou la recherche à base des clusters est par nature mieux adaptée dans le cadre des recherches larges, comme la recherche thématique ou la recherche exploratoire.

Nous donnons comme exemple le système de recherche sur Internet qui utilise cette approche, qui est le moteur de recherche Yahoo. Ce système est un moteur thématique. Il présente les documents sous plusieurs classes et sous-classes de différents niveaux.

Avantages:

- Le document stocké sous un cluster (classe) traite exactement le thème correspondant à la classe. Il correspond aussi à tous les thèmes antécédents [Leloup 98].

Inconvénients :

- Cette approche est un peu difficile à mettre en œuvre car elle comporte plusieurs niveaux [Leloup 98].

Exemple :

Comme exemple de système de recherche qui se base sur la notion de clusters, nous avons le moteur de recherche sur Internet Yahoo (www.yahoo.com). Ce moteur de recherche présente à l'utilisateur une liste de classes ou domaines. Lorsque l'utilisateur clique sur l'une de ces classes, le moteur lui présente toutes ses sous-classes avec le nombre de documents contenus. L'utilisateur peut descendre dans la hiérarchie des classes jusqu'à arriver aux documents contenus sous une classe feuille.

Dans ce qui suit, nous présentons quelques classes et sous-classes du moteur de recherche Yahoo.

Le premier niveau présenté par Yahoo contient les classes suivantes:

Arts & Humanities

Literature, Photography...

Business & Economy

B2B, Finance, Shopping, Jobs...

Computers & Internet

Internet, WWW, Software, Games...

Education

College and University, K-12...

Entertainment

Cool Links, Movies, Humor, Music...

Government

Elections, Military, Law, Taxes...

Health

Medicine, Diseases, Drugs, Fitness...

News & Media

Full Coverage, Newspapers, TV...

Recreation & Sports

Sports, Travel, Autos, Outdoors...

Reference

Libraries, Dictionaries, Quotations...

Regional

Countries, Regions, US States...

Science

Animals, Astronomy, Engineering...

Social Science

Archaeology, Economics, Languages...

Society & Culture

People, Environment, Religion...

Si l'utilisateur clique sur la classe «*Health*», les sous-classes de cette classe apparaissent, ainsi que le nombre de documents contenus sous chacune de ces sous-classes. La liste de ces sous-classes est la suivante:

- [Alternative Medicine \(502\) NEW!](#)
- [Business to Business@](#)
- [Chats and Forums \(54\)](#)
- [Children's Health \(183\) NEW!](#)
- [Conferences \(18\)](#)
- [Death and Dying@](#)
- [Dentistry@](#)
- [Disabilities@](#)
- [Diseases and Conditions \(7864\) NEW!](#)
- [Education \(62\) NEW!](#)
- [Emergency Services \(249\)](#)
- [Employment \(114\) NEW!](#)
- [Environmental Health \(200\)](#)
- [First Aid \(12\)](#)
- [Fitness \(188\) NEW!](#)
- [General Health \(89\)](#)
- [Health Administration \(66\)](#)
- [Health Care \(346\) NEW!](#)
- [Health Sciences \(26\)](#)
- [Men's Health \(36\)](#)
- [Mental Health \(703\)](#)
- [Midwifery \(57\)](#)
- [News and Media \(204\)](#)
- [Nursing \(455\) NEW!](#)
- [Nutrition \(216\) NEW!](#)
- [Organizations \(21\)](#)
- [Pet Health@](#)
- [Pharmacy \(1194\) NEW!](#)
- [Procedures and Therapies \(305\)](#)
- [Public Health and Safety \(770\)](#)
- [Reference \(99\)](#)
- [Reproductive Health \(700\)](#)
- [Senior Health \(86\) NEW!](#)
- [Sexuality@](#)
- [Shopping and Services@](#)
- [Teen Health \(23\)](#)
- [Traditional Medicine \(196\) NEW!](#)
- [Travel Health and Medicine \(23\)](#)

- **Hospitals and Medical Centers** (44)
- **Institutes** (34)
- **Law@**
- **Long Term Care** (108)
- **Medicine** (5159) NEW
- **Web Directories** (51)
- **Weight Issues** (85) NEW
- **Women's Health** (159)
- **Workplace** (65) NEW

Si l'utilisateur clique sur la sous-classe «Disease and Condition», qui contient 7864 documents, ses sous-classes apparaissent. Parmi ces sous-classes nous trouvons la sous-classe «Cancer».

- **Allergies** (47)
- **Anxiety Disorders** (35)
- **Autoimmune Diseases** (35)
- **Back and Neck Injuries** (16)
- **Birth Defects** (31)
- **Blood Disorders** (17)
- **Bone Diseases** (12)
- **Cancers** (575) NEW
- **Circulation Diseases** (19)
- **Dental Conditions** (6)
- **Depressive Disorders** (53)
- **Digestion and Nutrition Disorders** (34)
- **Dissociative Disorders** (15)
- **Ear Conditions** (7)
- **Eating Disorders** (43)
- **Eye Conditions** (35)
- **Food Allergies** (18)
- **Foodborne Illnesses** (21)
- **Genetic Disorders** (105)
- **Heart Diseases** (82)
- **Hormonal Disorders** (7)
- **Impetigo** (4)
- **Impulse Control Disorders** (7)
- **Infectious Diseases** (67)
- **Institutes** (12)
- **Intestinal Diseases** (13)
- **Kidney Diseases** (50)
- **Language Disorders** (7)
- **Leukodystrophies** (5)
- **Liver Diseases** (27)
- **Mental Health** (44)
- **Metabolic Diseases** (13)
- **Mood Disorders** (14)
- **Neurological Disorders** (67)
- **Organizations** (125)
- **Personality Disorders** (7)
- **Phobias** (12)
- **Pregnancy Complications** (11)
- **Prion Diseases** (4)
- **Prostate Diseases** (10)
- **Registries** (10)
- **Respiratory Diseases** (53)
- **Sexual Disorders** (5)
- **Shopping and Services@**
- **Skin Conditions** (43) NEW
- **Sleep Disorders** (31)
- **Sports Injuries** (22)
- **Tropical Diseases** (17)
- **Vestibular Disorders** (7)
- **Web Directories** (11)
- **Usenet** (10)

Si l'utilisateur clique sur la sous-classe «Cancer» toutes les sous-classes de cette sous-classe apparaissent. Nous trouvons une classe qui s'appelle «Cancer Cluster»...

- [Bladder Cancer@](#)
- [Bone Cancer@](#)
- [Books@](#)
- [Brain Tumors@](#)
- [Breast Cancer@](#)
- [Cancer Clusters \(11\)](#)
- [Cervical Cancer@](#)
- [Chemotherapy-related Hair Loss@](#)
- [Clinical Trials \(14\)](#)
- [Clinics and Practices@](#)
- [Colon Cancer@](#)
- [Companies@](#)
- [Events \(23\)](#)
- [Fundraisers@](#)
- [Gynecologic Cancers@](#)
- [Hodgkin's Disease@](#)
- [Institutes \(136\) NEW](#)
- [Journals@](#)
- [Kidney Cancer@](#)
- [Laryngeal Cancer@](#)
- [Leukemia@](#)
- [Liver Cancer@](#)
- [Lung Cancer@](#)
- [Lymphoma@](#)
- [Myeloma@](#)
- [Neuroblastoma@](#)
- [Newsletters \(4\)](#)
- [Oncology@](#)
- [Oral Cancer@](#)
- [Organizations \(195\) NEW](#)
- [Ovarian Cancer@](#)
- [Pain Management \(7\)](#)
- [Pancreatic Cancer@](#)
- [Personal Experience \(49\)](#)
- [Prostate Cancer@](#)
- [Registries \(13\)](#)
- [Retinoblastoma@](#)
- [Skin Cancer@](#)
- [Testicular Cancer@](#)
- [Therapies \(20\)](#)
- [Thyroid Cancer@](#)
- [Web Directories \(8\)](#)
- [Usenet \(2\)](#)

Si l'utilisateur clique sur la classe «Cancer Cluster», il va aboutir à toutes ses sous-classes. La procédure se répète jusqu'à arriver à la liste des documents sous la dernière sous-classe feuille.

2- 2 - 2 – 4 - La mise en correspondance mixte:

C'est une approche de mise en correspondance qui se base partiellement sur la mise en correspondance directe et sur une approche à base de clusters ou assistée par des clusters. En effet, ces deux types de mise en correspondance offrent des intérêts complémentaires. La mise en correspondance directe reste plus sélective donc plus précise pour les fortes valeurs de pertinence que la mise en correspondance à base de clusters ou assistée par les clusters, alors que cette dernière fournit des résultats plus homogènes et d'une meilleure cohérence thématique. Pour cela, les chercheurs dans ce domaine ont eu l'idée de combiner ces deux types de mises en correspondance afin de profiter pleinement de leurs avantages respectifs [Leloup 98][Lamirel 97].

Le principe de la mise en correspondance mixte est représenté à la Figure numéro 2.4. Le traitement d'une requête R est effectué en parallèle selon les deux types de mises en correspondance. Les résultats Re1 de la mise en correspondance directe de la requête R avec l'ensemble D des documents de base, et Re2, le résultatat de la mise en correspondance de la requête R avec l'ensemble C des clusters des documents de la base, servent ensuite de base à l'élaboration du résultat final. Pour l'élaboration finale du résultat on va utiliser un nombre de fois bien déterminé soit le premier type de mise en correspondance soit le deuxième type [Leloup 98][Lamirel 97].

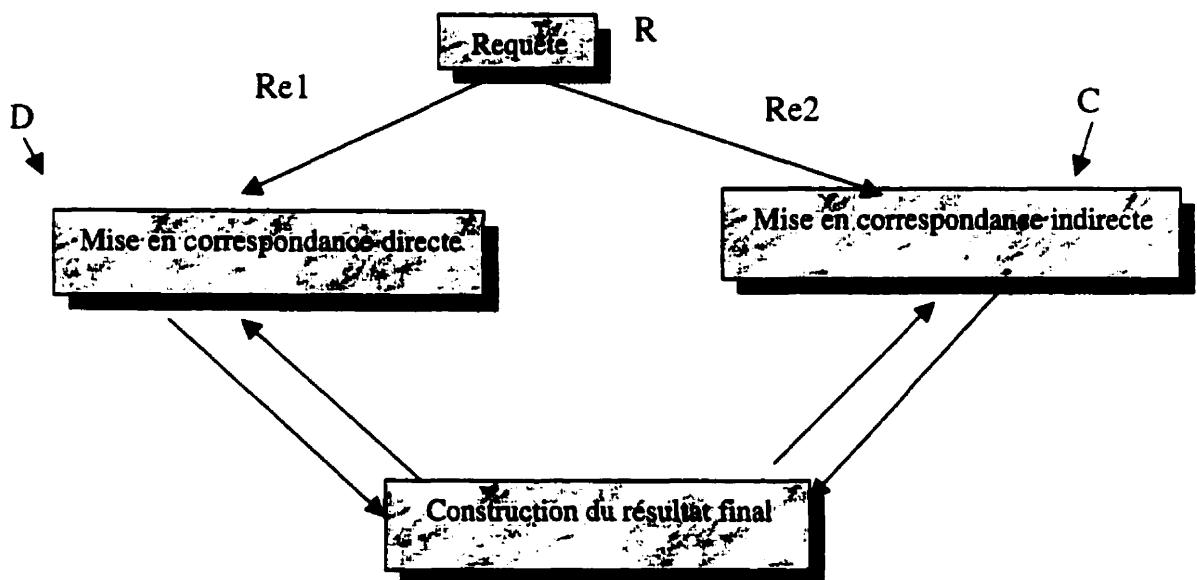


Figure numéro 2.4 : La mise en correspondance mixte

Exemple:

Comme exemple de système de recherche qui se base sur le principe de mise en correspondance mixte dans sa recherche, nous avons le moteur de recherche sur Internet intitulé «vivisimo» (<http://www.vivisimo.com>). Ce système fait partie des systèmes de recherche les plus puissants actuellement grâce à sa nouvelle approche de présentation des résultats de recherche à ses utilisateurs. Vivisimo n'est pas un moteur

de recherche, mais un métamoteur de recherche. *Vivisimo* fonctionne sur Internet, il présente dans sa page web principale une zone de texte dans laquelle l'utilisateur peut taper sa requête (une liste de mots). Dès que l'utilisateur lance la recherche, *Vivisimo* analyse cette requête, la transmet aux différents moteurs de recherche sur lesquels il se base. En récupérant les résultats, il construit dynamiquement une hiérarchie de concepts. Un exemple de hiérarchie est présenté à la figure numéro 2.5 pour la requête «Cancer de la peau». La hiérarchie a cette requête comme racine, et tous les concepts et les thèmes similaires ou proches comme nœuds (fils immédiats ou descendants) de la hiérarchie. Sous les nœuds feuilles, nous trouvons les documents reliés à ce nœud. Un document peut se trouver sous plusieurs nœuds de la hiérarchie.

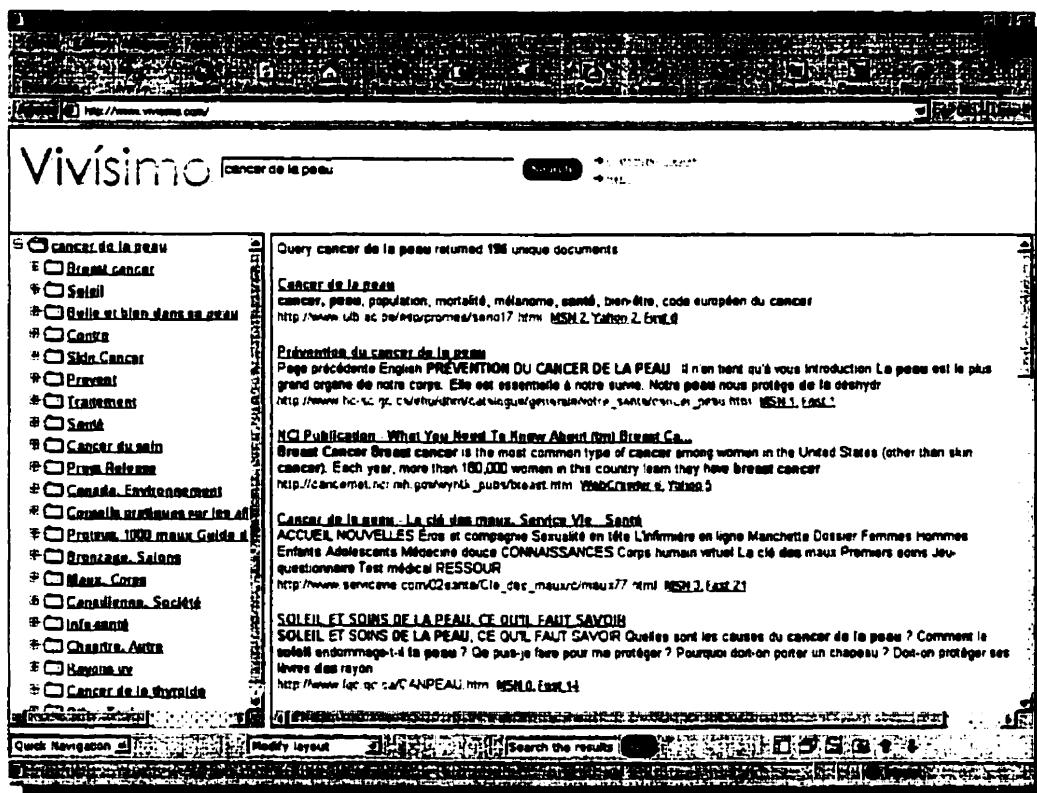


Figure numéro 2.5: Présentation des résultats (Vivisimo)

2- 2 - 3 – L'indexation et la recherche des documents multimédia:

Actuellement, les documents multimédia occupent une place prépondérante dans les informations que nous manipulons. Nous avons des images fixes ou animées, des séquences vidéos, des photographies, des peintures et des dessins numériques. Pour cette raison, dans notre travail, nous ne nous intéressons pas seulement à l'indexation et la recherche des documents textuels, mais nous allons considérer aussi d'autres documents multimédia comme les images fixes, animées, les séquences vidéos et les bandes sonores. À date, le domaine d'indexation et de recherche des documents multimédia est encore relativement jeune si on le compare à celui de l'indexation et de la recherche des textes, mais il est en pleine effervescence [Vézina 99]. À partir de la fin des années 70, de nombreuses institutions ont voulu mettre en valeur leurs collections visuelles et sonores. Toutefois, ces projets soulevèrent de nombreux problèmes vis-à-vis de l'indexation et de la recherche du matériel audiovisuel. Ces problèmes découlent de la nature même de ces documents, de leur organisation et de leur stockage [Berinstein 98].

Dans la majorité des applications d'indexation et de recherche des informations multimédia que ce soient des images – fixes ou animées- ou des séquences audio ou vidéos, l'indexation et la recherche ne se basent pas sur le contenu de ces documents, mais plutôt sur des techniques utilisées pour les documents textuels. Des exemples de ces systèmes seront donnés dans le chapitre suivant. Jusqu'à maintenant, ces techniques restent les plus directes et les plus efficaces pour trouver les images, les vidéos ou autres informations multimédia [Stix 97]. Pour utiliser les mêmes techniques, utilisées pour les documents textuels, pour indexer et chercher des informations multimédia, un expert analyse et décrit les informations multimédia par des annotations textuelles. Ensuite, les systèmes d'indexation et de recherche des documents textuels sont utilisés pour indexer et chercher des mots clés dans ces fichiers textuels associés aux documents multimédia, et le résultat de recherche n'est qu'un lien vers le fichier multimédia correspondant [Berinstein 98].

Ces techniques basées sur la recherche textuelle et la recherche par mots clés peuvent être améliorées en introduisant les techniques de traitement du langage naturel par exemple pour mieux garantir la pertinence des documents fournis en résultat [Vézina 99].

Généralement les informations textuelles sont insuffisantes pour décrire des images ou des séquences vidéos ou tout autre type de documents multimédia. Pour cette raison de nombreux efforts ont été faits pour créer des applications qui essaient d'indexer et de chercher les documents multimédia en se basant sur leurs contenus et leurs caractéristiques visuelles. Par exemple, on utilise des techniques de traitement des images et de reconnaissance des formes pour le traitement d'images ou des séquences audios ou vidéos. Dans le cas des images (l'information non textuelle multimédia la plus commune) les applications restent encore très limitées à cause de plusieurs problèmes dont le principal est l'automatisation des tâches d'indexation et de recherche [Berinstein 98]. Certaines de ces applications permettent aux utilisateurs de spécifier la requête en donnant des exemples d'images ou des formes dessinées, c'est ce qu'on appelle «la recherche par l'exemple», ou de sélectionner des caractéristiques des images à chercher (comme la couleur, la texture et le contour par exemple), et l'application va se charger de chercher les images similaires à l'image donnée ou sélectionnée à partir d'un catalogue d'images, ou les images qui contiennent les formes données ou encore celles qui supportent les caractéristiques données par l'utilisateur [Vézina 99]. Ces applications sont très limitées, et elles ne sont efficaces que dans des domaines spécifiques comme le domaine de l'imagerie médicale et la télédétection par exemple où les images sont souvent similaires et l'utilisateur a une idée des formats que peuvent contenir les images. Dans ces domaines spécifiques il est facile de modéliser les besoins des utilisateurs, ce qui facilite la création d'outils automatiques d'analyse d'images [Berinstein 98]. Dans le chapitre suivant, nous allons parler plus en détail de quelques applications de ce genre.

2- 3 – La recherche des documents sur Internet:

Une analogie revient souvent lorsque nous parlons d'Internet: qui est comparé à une gigantesque bibliothèque dans laquelle l'Internaute pourrait trouver tout ce qu'il cherche et bien plus encore - et c'est ce "bien plus encore" qui pose au chercheur pressé ou à la recherche précise plus de problèmes qu'il n'apporte de réponses [Herlin 97].

En fait de bibliothèque, il s'agit bel et bien d'une médiathèque. Ce terme s'est substitué ces dernières années à celui de bibliothèque quand il s'agissait encore de parler de lieux réels, les bibliothèques et centres de documentation. Ces bâtiments devenaient donc des lieux de stockage et de mise à disposition de documents multimédias, le mot s'écrivant alors avec un tiret. Les disques, les cassettes audio ou vidéo, bientôt les logiciels venaient rejoindre les livres et les revues traditionnels. Pour tous ces documents il fallait créer des outils qui permettent aux utilisateurs de les trouver car, de la même manière que les supports se diversifiaient, leur nombre augmentait [Herlin 97]. Ces outils s'appellent des «moteurs de recherche».

Dans cette partie, nous allons voir comment se fait l'indexation et la recherche des documents sur le réseau Internet.

L'indexation et la recherche des documents sur Internet se fait en se basant sur des techniques textuelles. En fait, les pages web ne sont que des textes écrits sous le format (HTML HyperText Markup Langage) interprétés par un logiciel appelé «navigateur» [DTI 97].

2 – 3 – 1 - La collecte des documents:

Dans le contexte actuel, nous pouvons dégager deux méthodes essentielles de collecte des documents : **manuelle** ou **automatique**. La première apporte une valeur ajoutée de sélection, validation et catégorisation des ressources. Dans cette approche la mise à jour est beaucoup moins rapide et la couverture beaucoup moins large. L'approche de collecte automatique des documents permet une mise à jour rapide des 42

documents collectés, mais elle ne donne pas le choix pour trancher si le document est suffisamment pertinent ou non [DTI 97]. Les modules de collecte automatique sont appelés «robots» ou «spiders» ou «crawlers».

2 – 3 – 2 - L'indexation des documents:

L'indexation des documents collectés est une étape très importante dans le processus de recherche. En effet, de la qualité de l'indexation dépend la qualité de la recherche.

L'indexation des documents sur Internet, se fait en utilisant les techniques textuelles. Ceci veut dire qu'un document Internet sera indexé par une liste de mots-clés. Actuellement, l'approche la plus utilisée pour indexer un document se base sur les méthodes statistiques. Donc une «bonne» indexation devrait permettre de constituer un fichier inverse (liste des mots-clés retenus, avec pour chacun d'eux les documents dans lesquels il apparaît) regroupant les termes pertinents des documents de la base. Pour cela, il faut tout d'abord parvenir à extraire correctement les mots du document, «tagger» les mots (reconnaitre leur genre/nombre, leur fonction grammaticale), «lemmatiser» les mots (chaque mot est normalisé : belle, belles, beau, beaux seront normalisés en beau; marcher, marchèrent, marcheras seront normalisés en marcher, ...), éliminer les mots vides, extraire les mots «pertinents» d'un document... [DTI 97]. Il existe plusieurs techniques d'indexation sur Internet. Dans ce qui suit nous présentons quelques unes de ces méthodes.

Nous rappelons qu'un «robot» est un logiciel d'exploration qui visite en continu les pages Web et les indexe de manière automatique, en fonction des mots-clés qu'ils contiennent. Cette indexation automatique s'effectue le plus souvent selon l'une ou plusieurs des méthodes suivantes :

* *Indexation par mots-clés contenus dans l'URL (Uniform Ressource Locator) du document HTML (HyperText Markup Langage) :*

Il s'agit du mode d'indexation le plus fréquent. Le «robot» répertorie automatiquement les mots-clés en lisant l'adresse Internet du site.

* *Indexation par mots-clés dans les titres et le premier sous-titre ou paragraphe du site:*

Le «robot» lit uniquement les mots-clés qui apparaissent dans le titre et les premières phrases de la page Web.

* *Méthode du «Scoping» :*

Le «robot» retient le mot qui apparaît le plus fréquemment dans l'ensemble du document.

• *Méthode des balises «Meta-tags» :*

Quelques moteurs de recherche utilisent les balises *Meta-tags* insérées dans le code source HTML du document. Il s'agit de la méthode la plus sûre d'indexation dans la mesure où le «robot» indexe les mots-clés, qui donnent une courte description du document, choisis par l'auteur du site lui-même.

• *Méthode de la «link-popularity» :*

Certains «robots» calculent automatiquement le nombre de liens hypertextes existant sur le réseau et renvoyant au site visité. Il indexera ainsi en priorité les sites faisant l'objet du plus grand nombre de référencement par liens hypertextes.

2 – 3 – 3 - La recherche des documents:

La recherche des documents a pour but de faire ressortir les documents de la base les plus pertinents pour la question posée. Nous distinguons deux aspects importants dans cette étape : le mode de formulation de la requête par l'utilisateur (requête par liste de mots-clés, requête booléenne, requête en langage naturel), et le méthode interne utilisée par le moteur de recherche (recherche booléenne simple, recherche booléenne raffinée, recherche en langage naturel). Dans ce qui suit, nous allons

présenter les différents modes d'interrogation existants, puis, les méthodes de fonctionnement interne des moteurs de recherche [DTI 97].

2 – 3 – 3 – 1- Les Différents Modes d'Interrogation:

Trois types d'interrogation sont envisageables : L'interrogation booléenne, l'interrogation par liste de mots et l'interrogation en langage naturel.

1. **L'interrogation Booléenne :** Elle se base sur le modèle booléen de mise en correspondance. Cette interrogation est en fait le type d'interrogation le plus basique, et surtout le plus simple à implémenter. Il consiste à formuler une question avec une liste de termes séparés par des opérateurs logiques (AND, OR, NOT), et à rechercher les documents correspondant à cette requête. Par exemple, pour la question "(moteur AND recherche) OR (search AND engine)", le système doit fournir comme réponse tous les documents de la base contenant les deux termes "moteur" et "recherche". A l'heure actuelle, un grand nombre des systèmes disponibles sur Internet fonctionne suivant ce type de recherche (Alta Vista (www.av.com), Excite (www.excite.com), Galaxy (galaxy.einet.net), HotBot (www.hotbot.com), InfoSeek (www.infoseek.com), Lycos (www.lycos.com), Magellan (www.mckinley.com), OpenText (search.opentext.com), WebCrawler (www.webcrawler.com), WWWorm (guano.cs.colorado.edu/www/)). Quelques raffinements sont fournis par certains systèmes, afin de palier aux limitations de la recherche booléenne :
 - **La combinaison des opérateurs :** Elle permet d'effectuer des recherches un peu plus complexes que celles proposées par plusieurs systèmes de recherche. À titre d'exemple nous avons Open Text (search.opentext.com), WebCrawler (www.webcrawler.com) ou WWWorm (guano.cs.colorado.edu/www/) qui ne permettent que d'utiliser un seul opérateur entre les différents mots. Ainsi, vous pouvez rechercher "Moteur AND recherche AND search AND engine", ou "Moteur OR recherche OR search OR engine", mais vous n'avez aucun moyen d'effectuer une recherche avec une question du type "(Moteur AND recherche) OR (search AND engine)". C'est une

limite très contraignante pour effectuer une recherche booléenne efficace, et c'est donc un point très important pour un système booléen de disposer de cette fonctionnalité [DTI 97].

- Le **parenthésage** des expressions permet de compliquer un peu plus les requêtes, et permettra ainsi à l'utilisateur averti d'effectuer des recherches complexes. Cette possibilité, qui pourtant semble basique et tout à fait logique dans un contexte booléen n'est cependant disponible que dans les moteurs de recherche Alta Vista (www.av.com) et Excite (www.excite.com) [DTI 97].
- La **troncature** (automatique et/ou manuelle) permet de rechercher des sous-chaînes. L'opérateur de troncature (généralement "*") remplace un ensemble de caractères afin d'effectuer des recherches plus larges. Nous distinguons trois types de troncatures, la troncature «*droite*» (la plus commune), la troncature «*gauche*», et enfin la troncature «*interne*». La troncature droite permet d'effectuer une recherche en utilisant le début d'un mot, afin d'obtenir les différentes formes dérivées du mot (et d'autres mots n'ayant aucun rapport avec ce que recherche l'utilisateur!). Par exemple, si vous recherchez des documents sur *les chats*, vous pouvez saisir la requête "chat*" afin d'obtenir les documents contenant les termes chat, chats, chatte, chattes, chaton, chatons, Le résultat va cependant être surprenant, puisque les documents contenant les termes français chatoiement, chatouille, chatterton vous seront également retournés. Mais ça ne s'arrête pas là, puisque le World Wide Web contient essentiellement des documents de langue anglaise, vous obtiendrez de nombreux documents contenant les mots chat (bavardage), ou chattel (bien mobilier), qui n'ont rien à voir avec votre recherche. Les troncatures gauche et interne fonctionnent de la même manière, elles permettent respectivement de rechercher des chaînes sans spécifier le début du mot, ou une partie interne du mot. Tous ces opérateurs sont à manipuler avec précaution. En effet, en lançant une recherche avec troncature dans un environnement multilingue comme le WWW, le

bruit devient très rapidement trop important pour pouvoir exploiter les résultats obtenus [DTI 97].

- Les opérateurs de proximité et d'adjacence. Ce type d'opérateurs a pour but de contraindre le système à rechercher des mots se trouvant proches l'un de l'autre, et donc étant supposés avoir des liens syntaxiques et/ou sémantiques. Ainsi, en posant la question "moteur ADJ recherche", vous allez éliminer un certain nombre de documents concernant la recherche dans le domaine des moteurs automobiles ou aéronautiques. Mais il ne faut pas s'y tromper, cet opérateur ne permet pas de spécifier qu'il doit exister un lien syntaxique entre les mots. Il va considérer qu'un document est pertinent si les mots moteur et recherche ne se trouvent pas séparés dans le texte par plus de 2, 3, 4, 5 ... 10 mots. La limite du nombre de mots est soit déterminée arbitrairement par le système, soit spécifiable par l'utilisateur suivant les moteurs de recherche. La recherche de syntagmes nominaux (phrases) vous permet de rechercher une chaîne de caractère exacte. Par exemple, la question "moteur de recherche" ne vous retournera comme résultat que les documents contenant exactement la chaîne «moteur de recherche» [DTI 97].

Il n'est pas besoin d'être un spécialiste des systèmes de recherche d'information pour se rendre compte des limites des systèmes booléens. En fait, leur fonctionnement interne se résume à une simple recherche de chaîne de caractères. Que cette chaîne soit ou non syntaxiquement correcte, peu importe (Essayez d'ailleurs de poser des questions n'ayant aucune signification aux moteurs de recherche actuels, vous serez étonnés des résultats. Des questions comme "aaaaa", "aieru", ou ":-)" ont de grandes chances de vous retourner des résultats !!!), si elle est présente dans le document, le système la retrouvera. L'utilisation de requêtes booléennes peut rapidement devenir complexe, et finalement, sur Internet, qui utilise ces opérateurs pour effectuer des recherches? Les documentalistes certainement, les informaticiens peut-être, l'utilisateur non spécialiste sûrement pas! [DTI 97].

2. La Recherche par liste de mots (ou pseudo Langage Naturel) permet de se passer d'utiliser un langage d'interrogation pour effectuer une recherche. Ces dernières sont donc plus simples à formuler. Plus simples, mais également plus imprécises, et donc plus bruitées. En effet, ce mode d'interrogation souvent proposé par les moteurs (AliWeb, AltaVista, EuroFerret, Excite, Galaxy, HotBot, InfoSeek, Lycos, Magellan, OpenText, WebCrawler, WWWorm) en complément de la recherche booléenne n'est en fait qu'une surcouche logicielle de cette dernière. La requête de l'utilisateur est retranscrite par le système en une expression booléenne suivant un schéma précis pré-établi (ET implicite, OU implicite, troncature droite implicite, etc...) [DTI 97].

En somme, même si la recherche à partir de tels systèmes est a priori plus simple que la recherche à partir d'une requête booléenne, il est néanmoins nécessaire que l'utilisateur connaisse les traitements de reformulation de la question effectués par le moteur. Il pourra ainsi adapter sa requête au fonctionnement interne de ce dernier et il aura une meilleure compréhension de résultats obtenus [DTI 97].

3. La recherche en langage naturel, de loin la mieux adaptée au texte va permettre à l'utilisateur de formuler une question totalement libre (en langage naturel = le langage commun de "*tous les jours*") pour effectuer sa recherche. Une telle recherche nécessite une indexation et une recherche "*intelligente*" mettant en oeuvre des modules de traitements linguistiques élaborés. Actuellement, d'après nos études, il existe un seul moteur de recherche sur le réseau Internet disposant d'une telle puissance de traitement du langage. Ce moteur est intitulé «Delphes» de la compagnie «Delphes Technologies» (http://www.delphesintl.com/newdelphes/02francais/portal_FR.asp). Ce moteur permet à l'utilisateur de spécifier sa requête en langage naturel. Il élimine les termes vides de la requête. Ensuite, il présente les documents résultats qui contiennent les termes intéressants de cette requête.

2 – 3 – 3 –2 - Méthodes de Fonctionnement Interne:

Dans cette section, nous présentons les méthodes utilisées par les moteurs de recherche sur Internet pour la recherche d'information. Nous envisageons ici deux types de fonctionnement interne des moteurs de recherche : la recherche booléenne, et les systèmes vectoriels.

1. La Recherche Booléenne, comme nous l'avons évoqué est un système basique de recherche. Il repose sur une simple comparaison de chaînes de caractères. Le moteur recherche dans son fichier inversé (fichier d'index) les mots correspondant à ceux de la requête, tout en respectant les opérateurs séparant les différents termes. L'interrogation est réalisée en exprimant le besoin de l'utilisateur par une fonction booléenne de descripteurs. Ainsi, les documents réponses sont ceux dont l'index fournit la réponse «vrai» à cette fonction. Il faut noter que pour ces systèmes, deux méthodes de fonctionnement interne sont envisageables :

- o La recherche booléenne simple qui se contente de retrouver les documents contenant les chaînes de caractères correspondant à la requête de l'utilisateur en respectant les restrictions des différents opérateurs. A défaut d'opérateur entre les mots, ce système va utiliser un opérateur implicite qui sera en général le OU logique afin de ne pas trop restreindre la recherche et retourner une liste vide de documents. Il est cependant utile de savoir que quelques systèmes tels que Lycos ou HotBot utilisent l'opérateur ET comme opérateur implicite [DTI 97].
- o La recherche booléenne avec dégradations successives de la question. Cette méthode consiste à reformuler la question de l'utilisateur avec des opérateurs ET entre les termes, et à effectuer plusieurs dégradations de la question de l'utilisateur en insérant des opérateurs permettant d'élargir la recherche (typiquement OU, troncature) [DTI 97].

Nous constatons donc qu'un système de recherche booléen n'est pas en mesure d'évaluer la pertinence d'un document par rapport à une question. Il n'y

a que deux états possibles, le document répond à la question, ou il n'y répond pas. Le moteur de recherche retourne donc une liste de documents réponses pertinents à la question, mais ils ne sont pas triés les uns par rapport aux autres. Plusieurs méthodes sont utilisées par les moteurs booléens actuels pour pallier ce manque et fournir une mesure de pertinence : nombre d'occurrences des mots de la question dans le document, présence ou absence des mots de la question dans le titre du document, popularité du site auquel appartient le document, etc. Toutes ces méthodes semblent assez approximatives et leurs fondements sont assez douteux [DTI 97].

2. Les Systèmes Vectoriels sont basés sur le modèle vectoriel conventionnel que nous avons présenté précédemment.

2- 4 – Conclusion:

Nous avons présenté dans ce chapitre une revue rapide des principales approches d'indexation et de recherche des documents textuels ou multimédias. Nous avons présenté également le processus d'indexation et de mise en correspondance (recherche) sur le réseau Internet.

Pour voir les approches mises en pratique par les systèmes actuels d'indexation et de recherche des documents textuels et multimédia, nous allons explorer, dans le chapitre suivant, quelques uns de ces systèmes dont la majorité existent sur Internet. Un bilan sur ces différents systèmes va être fait. Dans la deuxième partie du chapitre suivant, nous allons présenter le standard MPEG-7 qui est un standard d'annotation des documents multimédia. Également un bilan concernant ce standard sera fait.

Chapitre 3

L'exploration des solutions existantes

3- 1 – Introduction:

Dans le chapitre précédent, nous avons présenté une revue bibliographique à propos de l'indexation et de la recherche des documents. Nous avons vu aussi comment se fait l'indexation et la recherche sur le réseau Internet. Nous rappelons que notre objectif est de concevoir et de réaliser un système d'indexation et de recherche de documents de tous formats. Nous avons étudié plusieurs systèmes dans le but de savoir si nous pourrons améliorer, intégrer ou adopter des techniques proposées par ces systèmes. Dans la première partie du présent chapitre, nous présentons les systèmes que nous avons étudiés. Nous commençons par présenter des systèmes indépendants d'indexation et de recherche des documents textuels, nous étudions aussi ceux qui existent sur le réseau Internet et qui permettent l'indexation et la recherche dans les pages web de ce réseau. Puis, nous présentons quelques systèmes d'indexation et de recherche d'images, les images qui présentent le format non textuel (visuel) le plus évident d'un document.

Après la présentation de tous ces systèmes, nous faisons un bilan pour présenter leurs limites.

Dans la deuxième partie de ce chapitre, nous présentons un standard intitulé «MPEG-7» qui permet de présenter les documents textuels ou multimédia en se basant sur une structure unique. Ce standard est très intéressant, mais, il présente des limites.

3- 2 – Les systèmes d'indexation et de recherche des documents textuels:

Actuellement, il existe plusieurs systèmes d'indexation et de recherche des documents de format textuel, dont la plupart se trouvent sur le réseau Internet. Ceux qui ne sont pas sur le réseau Internet s'intitulent «*Systèmes de gestion documentaire*», ils fonctionnent sur des fonds documentaires locaux comme par exemple ceux qui existent dans des bibliothèques. Ils peuvent indexer et chercher des documents de plusieurs formats textuels tels que ASCII, word, RTF,...etc. Ceux qui sont sur le réseau Internet sont connus sous le nom de «robots» (dans le cas des systèmes d'indexation) et «moteurs de recherche» (dans le cas des systèmes de recherche ou de mise en correspondance). Ces systèmes indexent et recherchent des pages web à travers tout le réseau Internet en se basant sur le code HTML (HyperText Markup Langage) de ces pages.

3 - 2 – 1 – Présentation des systèmes d'indexation et de recherche de documents textuels:

Dans ce qui suit, nous présentons quelques uns de ces systèmes. Dans les sous-sections de 'a' jusqu'à 'd', nous présentons les systèmes d'indexation et de recherche des documents textuels. Ces systèmes sont dis 'indépendants' car ils fonctionnent avec leurs propres bases de données et avec leurs propres formats textuels. Pour ces systèmes, nous allons étudier les formats pris en compte, l'aspect indexation et l'aspect recherche. Puis, dans les sous-sections de 'e' jusqu'à 'k', nous présentons les systèmes d'indexation et de recherche des pages web sur le réseau Internet. Ces

systèmes indexent et cherchent des pages web en se basant sur le code textuel HTML. Pour ces systèmes, nous allons étudier l'aspect recherche et l'aspect présentation des résultats, car l'indexation se base sur le même principe pour tous ces systèmes: l'utilisation des mots-clés et de méthodes statistiques.

a) **ZyIndex:**

ZyIndex est un système d'indexation et de recherche conçu par la société «*ZyLab Technology*». *ZyIndex* comporte deux modules intitulés «*ZyBuild*» et «*ZyFind*» respectivement modules d'indexation et module de recherche. Ce système permet d'indexer des documents textuels dans un environnement client-serveur [Leloup 98].

* **Mode d'indexation:**

Le système *ZyIndex* indexe les informations disponibles sur un réseau local sans les dupliquer et conserve la trace de leurs localisations grâce aux chemins d'accès des fichiers natifs. *ZyBuild* construit les indexes et permet de programmer leur mise à jour de manière périodique, séparément pour chaque index. *ZyFind* interroge les index, retrouve les fichiers concernés, les affiche en indiquant la position des termes pertinents. *ZyIndex* indexe les documents textuels. Tous les index sont positionnels et gardent trace de la position de mots dans le texte, par rapport au numéro de ligne et numéro de page du document, selon le paramétrage, des positions dans les phrases et les paragraphes. *ZyIndex* indexe également le titre du document – qui sera l'information présentée en recherche – à partir de critères permettant de le localiser dans les documents [Leloup 98].

* **Modes de recherche:**

Les modes de recherche proposés par *ZyIndex* sont, en standard, la recherche booléenne, la troncature et la recherche de proximité... La recherche booléenne a été présentée dans le chapitre précédent, ainsi que la troncature. La recherche de proximité se fait par distance fixe ou variable dans une plage donnée, avec ou sans

ordre spécifié, ainsi que dans une ligne, une phrase ou un paragraphe. Il est également possible d'effectuer une recherche sur des documents multilingues en précisant le nombre de langues contenues par rapport à une liste de langues [Leloup 98].

b) Spirit:

Spirit est né au début des années 80 de travaux concernant la recherche d'information au niveau Européen et par la *CEA* (*Commissariat à l'Energie Atomique*). Depuis 1993, il a été industrialisé et commercialisé par la société *T-GID* qui a maintenu des partenariats avec les organismes de recherche et le *CEA* pour faire évoluer le produit. *Spirit* est un moteur d'indexation et de recherche de documents qui fonctionne dans un environnement client-serveur et qui assure le stockage des documents. *Spirit* fonctionne en architecture Client-serveur, multibase et multiserveurs. Les différents services du moteur peuvent être exécutés sur des serveurs différents : indexation en utilisant des dictionnaires sur plusieurs bases de données, recherche et utilisation des règles de reformulation, stockage de la base de données documentaires...etc [Lamirel 97]

Spirit traite les documents sous la forme d'un corps de texte et d'une notice comportant titre, auteur, date. Les textes sont stockés par *Spirit* en ASCII étendu (sur 8 bits). Les dictionnaires utilisés par *Spirit* sont très riches et comportent respectivement environ 500.000 entrées en français, 1.300.000 en allemand et 300.000 en anglais [Lamirel 97].

*** Mode d'indexation:**

L'indexation effectuée par *Spirit* se fait par mots-clés. Il utilise la technique statistique pour indexer les documents. *Spirit* utilise plusieurs dictionnaires pendant l'indexation, ces dictionnaires sont généraux et peuvent être enrichis par l'utilisateur.

*** Mode de recherche:**

La recherche proposée par *Spirit* s'effectue à l'aide de mots-clés. Ces mots peuvent être combinés par des opérateurs booléens comme AND et OR.

c) RetrievalWare:

Ce moteur d'indexation et de recherche a été conçu et utilisé par la compagnie «Excalibur Technologies». Cette compagnie développe et commercialise une gamme de produits d'indexation et de recherche d'information dont *RetrievalWare* fait partie.

Les documents à indexer ou chercher avec *RetrievalWare* peuvent être structurés en champs, repérés par des balises qui sont indexées séparément et donc accessibles comme tels. Ils sont manipulés en format ASCII étendu [Leloup 98].

* Mode d'indexation:

Le processus d'indexation des documents s'effectue à travers différents modules selon le principe dit de pipeline comme le montre la figure suivante (Figure numéro 3.1) :

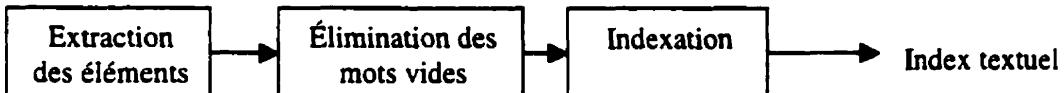


Figure numéro 3.1 Mode d'indexation (RetrievalWare)

La première étape est l'identification des éléments contenus dans le texte, qui consiste à éclairer le texte en mots, avec reconnaissance des dates et des nombres dans le corps du texte. La deuxième étape consiste à éliminer les mots-vides² en préparation de l'indexation. La dernière étape est celle d'indexation qui utilise une technique de fichiers inversés, avec une structure de B-Tree dans laquelle les entrées d'index (mots, dates, nombres) sont référencées ainsi que leur position absolue dans le texte [Leloup 98].

² Un mot vide est un mot qui n'est pas pertinent pour l'indexation et la recherche. Comme exemple de mot vide nous avons : le, la, les, une

*** Modes de recherche:**

Le mode de recherche proposé par *RetrievalWare* pour le texte est la recherche booléenne. La recherche booléenne est classique et s'applique aux mots de la requête fournie par l'utilisateur du système [Leloup 98].

d) ConText:

ConText est une option du SGBDR (Système de Gestion de Base de Données Relationnelle) Oracle qui permet la recherche textuelle. C'est le successeur des deux précédents outils mis au point par Oracle, «*TextRetrieval*» et «*TextServer*». *ConText* est disponible pour les versions Oracle 7 et Oracle 8 sur les plates-formes sur lesquelles fonctionne le SGBDR et notamment SUN/Solaris, IBM/AIX, HP-UX, Windows NT...

Les documents à indexer ou à chercher peuvent être stockés dans des tables du SGBDR ou à l'extérieur. Le stockage dans les tables du SGBDR peut être global- les textes sont stockés dans une colonne d'une table, chaque document représentant une ligne- ou selon un schéma dit «maître-détail», le texte des documents est éclaté dans plusieurs lignes de détail, elles-mêmes liées à une ligne maître [Leloup 98].

*** Mode d'indexation:**

ConText permet une indexation positionnelle du texte, en indiquant la position absolue des mots ou termes dans le texte. Les caractères accentués sont remplacés en indexation par les caractères correspondants non accentués. Une liste de mots vides par défaut est également fournie et peut être enrichie pour une application donnée [Leloup 98].

*** Modes de recherche:**

Les modes de recherche proposés sont variés : Recherche booléenne et de proximité par l'opérateur NEAR, pondération des termes de recherche, utilisation de la troncature gauche, droite, milieu, recherche phonétique (par langue) et recherche floue (tolérance par rapport à l'orthographe essentielle). La recherche booléenne présente aussi un opérateur particulier (ACCUMULATE) qui équivaut à un OU mais avec prise en compte des fréquences de termes trouvés dans les documents pour le calcul du rang de pertinence. Le nombre d'occurrences trouvées peut être fourni avant de lancer effectivement la recherche par un parcours des index [Leloup 98].

Les résultats de recherche peuvent être présentés en calculant le rang de pertinence à partir des pondérations des termes de recherche, sur une base statistique qui privilégie les termes rares dans l'index et la densité des termes de la question dans les documents obtenus en réponse. Le nombre de réponses peut également être limité ainsi que le nombre d'occurrences trouvées dans le lot résultat.

e) AltaVista (www.av.com):

Taille : 140 Millions de pages

Mise à jour : 1 j à 6 semaines

AltaVista a été lancé par la société «*Digital Equipment*» en décembre 1995, société rachetée par «*Compaq*» en 1998. Il se veut le plus complet des moteurs de recherche et effectue une indexation du texte intégral : plus de 140 millions de pages trouvées dans les serveurs Web. L'index est réactualisé en fonction de la stabilité des sites, déterminée par le robot «*Scooter*» à partir des visites précédentes. D'autre part il indexe en temps réel 4 millions de messages de plus de 14 000 groupes de nouvelles (News Groups).

*** Recherche:**

Alta Vista offre deux modes de recherche pour interroger les serveurs web :

* **Recherche simple:** Pas d'opérateurs booléens, la requête est une suite de mots. Nous disposons des signes '+' pour imposer un terme, '-' pour l'exclure. Le système est sensible aux minuscules et majuscules. Il ignore les mots trop fréquents. Altavista propose 25 langues de recherche.

* **Recherche avancée:** Il faut obligatoirement utiliser les opérateurs «AND», «OR», «NOT» ou «NEAR» en majuscule pour combiner plusieurs termes. Les parenthèses sont utilisées lorsqu'il y a plusieurs opérateurs. Un signe '+' accolé à gauche d'un terme le déclare obligatoire, alors que le signe '-' indique un terme à refuser. Le logiciel tient compte de la casse des caractères. Il tient compte aussi de la structure des documents en s'appuyant sur les balises HTML: nous pouvons limiter la recherche aux mots du titre, aux URL, aux liens contenus dans un document. Enfin, la limite par date d'entrée des documents dans la base de données est présente.

AltaVista vérifie l'orthographe des termes de recherche pour l'anglais, le français, l'espagnol et l'italien.

Ce système offre aussi des recherches spécialisées :

Keyword ➔ Trouve les pages contenant ces mots-clés.

anchor:text ➔ Trouve les pages contenant le terme dans un lien hypertexte.

applet:class ➔ Trouve les pages contenant une applet Java.

domain:domainname ➔ Trouve les pages contenant le domaine spécifié.

host:name ➔ Trouve les pages d'un ordinateur.

image:filename ➔ Trouve les pages contenant des images dont le nom de fichier contient le terme.

link:URLtext ➔ Trouve les pages pointant vers l'URL spécifiée.

text:text ➔ Trouve les pages contenant le texte spécifié n'importe où.

title:text ➔ Trouve les pages contenant le texte dans le titre de page.

url:text ➔ Trouve les pages contenant ayant le terme dans les URL.

Un choix de 25 langues permet de limiter les résultats d'une recherche aux seules pages publiées dans la langue spécifiée.

*** Résultats:**

Les documents sont triés par pertinence en fonction des termes du champ «ranking». Le format d'affichage donne un titre (lien hypertexte vers le site), l'URL, une douzaine de mots du texte, la taille du fichier, la langue du document, la date d'entrée dans la base de données d'*Alta Vista* .

f) Excite et Excite France (www.excite.com et www.excite.fr):

Taille : 55 Millions de pages

Mise à jour : 1 j à 6 semaines

Lancé à la fin de l'années 1995, Excite a proposé une nouvelle version de son moteur de recherche et revendique la première place en volume : la base de données pointe vers plus de 55 millions de pages Web.

La base de données est mise à jour chaque semaine. Les visiteurs peuvent personnaliser le service dès la page d'accueil et choisir les nouvelles et sources d'information qui apparaîtront automatiquement à chaque visite sur le site. Nouvelles, sports, météo locale, cotes de la Bourse et horoscope figurent parmi les différentes options possibles du service nommé «All About You».

Excite a acquis *Magellan* (magellan.excite.com) en juillet 1996 et *WebCrawler* (www.webcrawler.com) en novembre 1996. Ces derniers continuent en tant que services séparés. «Global Excite» pointe vers l'Australie, la Chine, la France, l'Allemagne, l'Italie, le Japon, la Hollande, la Suède et U.K.

Excite utilise une méthode dite «Intelligent Concept Extraction» pour analyser un site et déterminer les mots-clés à indexer, en cherchant des concepts (ou des thèmes) plutôt que des mots. Il utilise un algorithme pour trouver des thèmes, il n'indexe donc

pas le texte complet et ne tient pas compte des balises META. Le concept de chaînes (channels) structure un répertoire sélectif de 140 000 sites. Une «chaîne» fédère dans une même page diverses informations relatives à un même thème.

*** Recherche:**

La recherche se fait soit par mots-clés, soit par concepts : La première méthode ne ramène que les documents contenant les termes de la recherche alors que la seconde peut donner des documents contenant de l'information liée à ces termes. Le producteur prétend que les premiers documents sont identiques dans les deux cas.

Le moteur utilise une technique dite floue (fuzzy AND) en combinant les opérateurs AND et OR pour une recherche sur plusieurs termes. Une recherche plus précise est possible par «Power Search». Nous disposons alors d'un formulaire permettant d'éviter l'utilisation explicite des opérateurs booléens et des parenthèses.

*** Résultats:**

Les résultats sont classés par ordre de pertinence, avec un titre, un résumé mais sans l'URL. Il est possible de cliquer sur un icône pour relancer la recherche vers des documents similaires. Il est possible de trier les résultats par site web. Le moteur de recherche Excite est en passe de développer une nouvelle interface graphique de présentation des résultats de recherche en 3D. Développé en Java, le site devrait permettre de visualiser sur une seule page les réponses principales au centre, et les concepts approchant en orbite.

g) Google (www.google.com):

Taille : 100 Millions de pages

Mise à jour : chaque semaine

Dernier né des moteurs de recherche de l'Université de «Stanford», *Google* grossit très vite et se distingue par deux aspects :

- * Son mode de classement des résultats (PageRank)
- * Le fait qu'il archive toutes les pages html indexées

Google est le seul outil qui garde sur disque l'ensemble des pages qu'il indexe, constituant ainsi un système d'archivage inexistant par ailleurs.

*** Recherche:**

L'interface est très simple et se distingue des pages d'accueil des autres systèmes où il est souvent difficile de se retrouver. La requête est une suite de termes séparés par un espace. L'opérateur par défaut est le AND et nous disposons du signe '-' correspondant à l'opérateur SAUF.

Google invite à affiner une recherche en ajoutant d'autres termes dans la requête. Il n'est pas possible de rechercher dans un sous ensemble de la base de données.

*** Résultats:**

La barre rouge exprimée en pourcentage correspond au classement du document basé sur le nombre de liens pointant sur lui. La barre devient mauve pour les autres résultats d'un même serveur, car ces derniers sont regroupés. Le tri ne dépend pas des mots de la question contrairement aux autres moteurs de recherche.

h) HotBot (www.hotbot.com):

Taille : 110 Millions de pages

Mise à jour : 1 j à 2 semaines

Lancé en mai 1996, ce service est un partenariat entre le magazine électronique «HotWired» et l'Université de «Berkeley».

La base de données du système *HotBot* pointe vers plus de 110 millions de documents. HotBot indexe le texte complet des pages HTML.

*** Recherche:**

Nous utiliserons ici aussi de préférence le mode «More search options» pour formuler les requêtes. Nous bénéficions ainsi de plusieurs critères rarement trouvés ailleurs :

- * Un choix parmi 9 langues
- * Une limite par date d'entrée dans la base permet, comme avec *AltaVista*, d'envisager des recherches à intervalle régulier sur une même question.
- * L'option media type permet de préciser la recherche d'une image, d'un document audio ou de fichiers de type VRML, Acrobat, JavaScript, Java etc...
- * La localisation géographique des sites recherchés
- * La possibilité de recherche de mots proches, est réservée à l'anglais.

*** Résultats:**

Le classement des résultats donnés par le score, repose sur les critères suivants par ordre d'importance : mots du titre, mots inclus dans les balises META et fréquence des mots dans le corps du texte.

Hotbot a introduit la technologie «*Direct Hit*» qui améliore la sélection de sites pertinents en analysant les choix effectués par les millions d'utilisateurs. «*Direct Hit*» est une nouvelle technologie inventée par la compagnie «*Direct Hit*» (<http://www.directhit.com>). Cette technologie espionne les recherches des utilisateurs des moteurs de recherche et mémorise leurs comportements.

Les moteurs de recherche comme AltaVista, Lycos ou Google effectuent, de façon classique, leurs investigations en texte intégral dans leurs index composés de plusieurs centaines de millions de pages Web. Le système de classement des résultats est le plus

souvent proche d'un outil à l'autre : présence du mot demandé dans le titre de la page, dans le texte visible, dans les balises META, etc. Bref, tout ça est devenu très classique. Pour bien comprendre comment il fonctionne, analysons le comportement "classique" d'un internaute devant un moteur de recherche : il va sur la page d'accueil, saisit un ou plusieurs mots dans un formulaire, consulte la page de résultats proposée (sur laquelle plusieurs liens sont indiqués, classés par ordre de pertinence), il choisit l'un d'entre eux, va sur le site correspondant, le consulte. Si cette page ne lui convient pas, il revient sur la page de résultats du moteur (par le bouton "précédent" du navigateur), choisit un autre lien parmi ceux proposés, etc. jusqu'à ce qu'il ait trouvé son bonheur.

DirectHit va, en fait, profiter de cet aspect comportemental de l'internaute pour tenter de trouver les pages les plus "populaires", c'est-à-dire le plus souvent cliquées, sur un moteur de recherche et ainsi améliorer ainsi leur classement. Il fonctionne, en règle générale, en tâche de fond sur un moteur existant. A chaque consultation d'un internaute, il va noter sur quel lien il a cliqué et quel était le rang (le classement) de ce lien. Il calcule ensuite combien de temps l'utilisateur met avant de revenir sur la page de résultats. Si il ne revient pas, il en "déduit" que le site proposé était a priori pertinent. Son adresse sera alors mieux classée dans les résultats suivants, lors d'une interrogation sur le même mot clé. Et ainsi de suite, les interrogations et la façon d'interroger et de naviguer des internautes vont alors enrichir la base de données de *DirectHit*.

DirectHit est le seul système qui tienne compte, pour le classement des pages qu'il propose, du comportement "humain" de l'internaute. De plus, et ce n'est pas négligeable, il est très complexe à "spammer" (obtenir par une technique quelconque - assimilée à une fraude - un meilleur positionnement sur le moteur de recherche). C'est pour celà que les moteurs utilisent de plus en plus largement ce concept, qui est sans conteste appelé à un bel avenir !

i) InfoSeek et InfoSeek France (www.infoSeek.com):

Taille : 50 Millions de pages

Mise à jour : 1 j à 8 semaines

InfoSeek a été lancé début 1995. Au printemps 1996, «Ultraseek» a profondément fait évoluer ce service en passant de 2 millions à 50 millions d'URL. Il permet de rechercher l'information dans les serveurs web, Gopher, FTP, les groupes de News (FAQ comprises) et des sites évalués. La mise à jour varie entre 1 jour et 8 semaines.

InfoSeek est localisé dans 11 pays. Il indexe le texte intégral des documents trouvés sur les serveurs visités.

*** Recherche:**

Le traitement de la question présente des caractéristiques intéressantes :

- * *Infoseek* recherche automatiquement les variations des termes grâce à une troncature implicite à droite (exemple : photography, photographer, photographs)
- * L'ordre des termes a son importance : c'est indispensable pour rechercher des mots composés ou des morceaux de phrases.

En recherche simple, nous posons une question en entrant les termes de recherche sans opérateur booléen ni caractère de troncature. Le logiciel ne recherche pas les mots vides. Cependant les majuscules sont prises en compte : c'est utile pour la recherche des noms propres. Pour préciser une question, nous utiliserons les règles suivantes :

- Un nom propre : mettre la première lettre en majuscule: Exemple: «Orson Wells»
- Deux noms propres : mettre une virgule entre : Exemple: «Laurel, Hardy»
- Séparés par un tiret pour une proximité forte : Exemple: «laser-printer ISO-9000»
- Entre crochets si l'ordre est indifférent : Exemple: «[WWW search]»
- Obligatoirement un mot : coller un signe plus devant : Exemple: «chip +Motorola»

- Un mot et éviter un autre : coller un signe moins devant : Exemple: «python - Monty»
- Penser aux synonymes : Exemple: «CD-ROM, CDROM, cdrom»

*** Résultats:**

Le résultat d'une recherche est trié et les documents les plus pertinents apparaissent en tête.

Les facteurs de tri les plus importants sont :

- Les termes de recherche présents dans le titre et au début du document sont privilégiés,
- La fréquence des termes de recherche dans le document.

Cependant la taille de l'Internet conduit souvent à une liste importante de réponses. *InfoSeek* utilise une technique brevetée pour différencier les pages riches en information des pages pauvres. De plus il propose une liste de sujets à explorer en rapport avec la question ou la requête.

Parallèlement à la recherche directe, *InfoSeek* propose 18 catégories ce qui permet d'interroger un sous-ensemble de la base de données. Mais ces catégories sont créées automatiquement par reconnaissance de vocabulaire et ne sont donc pas toujours très pertinentes.

j) Northern Light (www.northernlight.com):

Taille : 80 Millions de pages

Mise à jour : 2 à 4 semaines

Ce système, lancé le 12 Août 1997, propose une recherche dans une base de données de 30 millions de pages indexées du web et dans une «Collection Spéciale» de documents issus de quelques 5 000 sources (journaux, livres, magazines, bases de

données, dépêches d'agences) introuvables sur l'Internet. La recherche est gratuite mais les articles sont payants (1 à 4\$).

*** Recherche:**

Dans la recherche simple l'opérateur AND est implicite. Nous disposons aussi des opérateurs «OR», «NOT» et des «parenthèses» pour utiliser plusieurs opérateurs dans la même question (requête). Une autre syntaxe consiste à placer le signe '+' ou '-' à gauche du terme pour imposer sa présence ou son absence dans les réponses comme sur d'autres moteurs de recherche.

*** Résultats:**

L'aspect novateur de ce système est le classement des documents trouvés dans des dossiers constitués automatiquement en fonction des réponses. Un dossier peut lui-même être constitué de sous-dossiers. Quatre types existent : thèmes, types de documents, source, langue. Ces renseignements se retrouvent dans l'affichage de chaque réponse avec la date de publication. Dans chaque dossier final, les réponses sont triées par pertinence.

Pour promouvoir les capacités de recherche de son moteur, *Northern Light* vient de lancer un service spécialisé sur la recherche d'informations économiques et financières. «Industry Search» (www.northernlight.com/industry.html) permet la recherche de renseignements sur des entreprises dans 26 catégories avec des limitations de dates des documents et du type d'information: communiqués de presse, revue de produits, offres d'emploi...etc

k) Voilà (www.voila.fr):

Taille : 6,5 Millions de pages en français

Mise à jour : chaque semaine

Lancé en juillet 1998, *Voilà* est la suite du moteur «Echo» racheté par «France Telecom». Sous forme de portail, *Voilà* regroupe un moteur de recherche sur le web francophone, un annuaire intégré (QuiQuoiOù), les actualités, la météo, les annuaires pages jaunes, pages blanches, les marques, les rues commerçantes...etc

Le service s'internationalise avec l'ouverture de «Voila.com» qui propose une recherche mondiale pour 5 pays en plus de la France.

*** Recherche:**

Voila recherche des documents dans une base de données de plus de 6 500 000 de pages web en langue française mise à jour chaque semaine. Les sites inscrits manuellement sont indexés sous 15 jours. Il offre deux possibilités de recherche: Une recherche simple et une recherche avancée.

*** Résultats:**

Dans les résultats de recherche, nous trouvons en tête les documents présents dans l'annuaire «QuiQuoiOù» et identifiés par une loupe. Le classement des réponses tient compte du contenu des balises META et de la place des mots dans le document HTML. Nous pouvons limiter l'affichage aux réponses indexées dans les 15 derniers jours ou choisir un regroupement par sites. *Voilà* effectue aussi une recherche par mots-clés dans une base de plus de 200.000 adresses e-mail francophones.

3 - 2 - 2- Bilan sur les systèmes d'indexation et de recherche des documents textuels:

En étudiant tous ces systèmes, nous avons dégagé les constats suivants:

* Les systèmes 'indépendants' d'indexation et de recherche des documents textuels utilisent leurs propres bases de données. Ces bases de données utilisent des structures internes relatives aux systèmes. Ces systèmes utilisent leurs propres formats textuels

qui peuvent être plus ou moins complexes. De plus, ces systèmes indexent les documents en se basant sur le principe des mots-clés. Ils cherchent les documents en se basant sur le même principe.

* Pour les systèmes d'indexation et de recherche sur le réseau Internet ils n'indexent et cherchent que les documents web (ils traitent un seul format : HTML). Nous n'avons aucune idée sur les formats des bases de données utilisés par ces systèmes. De plus nous ne pouvons pas y accéder. Ces systèmes se basent sur la notion des mots-clés dans leurs processus d'indexation et de recherche.

Nous rappelons que notre objectif est la conception d'un système d'indexation et de recherche des documents de tout format, et non seulement le format textuel. Pour cette raison, nous avons étudié quelques systèmes d'indexation et de recherche de documents non textuel. Le premier format de documents non textuels est le format image. Dans ce qui suit, nous présentons le contexte d'indexation et de recherche des images, puis nous présentons quelques systèmes d'indexation et de recherche de ce format particulier de documents.

3 - 3 - Contexte et enjeu de l'indexation et de recherche des images:

Les images sont des éléments caractéristiques des nouveaux média, qu'ils soient informatiques ou non. L'ère visuelle est en pleine expansion et profite de l'informatique pour s'épanouir, grâce à la numérisation. Les bases de données résultant de cette numérisation possèdent de réels avantages : le fait de pouvoir offrir à tout le monde et instantanément ce qui était auparavant sur support magnétique ou papier, donc difficilement transportable ; ensuite, le gain en place d'archivage est important puisque la conservation des pellicules ou des photos demande rapidement de gros volumes [Vézina 99].

D'un autre côté, cette conservation numérique apporte son lot de problèmes. Si l'ensemble des fichiers peut être rangé de façon structurée dans une base de données, il est souvent difficile d'y identifier des informations pertinentes pour une recherche

particulière [Vézina 99]. Il est alors difficile d'exploiter ces bases avec efficacité, que ce soit d'une façon locale ou distante. Pour le moment, le système d'indexation et de recherche le plus couramment utilisé consiste à associer quelques mots clés à chaque image stockée et à ne faire les recherches que sur ces mots. Le besoin de réaliser l'indexation et la recherche de ces images par leur contenu, et non plus seulement sur une information externe à ces images, est de plus en plus évident. Il devient donc crucial de pouvoir définir des techniques automatiques (ou au moins semi-automatiques) d'indexation par le contenu de ces bases pour une consultation ou une manipulation aisées de tels matériels multimédia. Le facteur automatique prend ici tout son sens car un tel type d'indexation et de recherche éviterait le caractère subjectif d'une description humaine d'une image. Il est aussi notoire que l'indexation manuelle et la recherche de milliers d'images est très coûteuse en temps et donc en argent, tout en n'assurant pas forcément une bonne efficacité [Nastar 99].

Ce problème d'indexation et de recherche des images est un véritable défi. Il faut être capable de résoudre plusieurs aspects fondamentaux: extraction pertinente et fiable d'informations relatives au contenu, organisation et représentation efficaces des éléments extraits, et recherche optimisée des objets de la base. De plus, les consultations et les utilisations de ces bases multimédia peuvent être de nature variable selon l'origine et l'objet de la requête émise [Vézina 99][NastarBoujema 99]. Une des difficultés principales pour réaliser un tel système tient au fait qu'il existe une séparation très nette entre la donnée d'une image (sous forme de pixels) et la description (par des mots ou même des schémas) du contenu de ces signaux et images. Ces questions nécessitent donc des efforts importants d'étude pour apporter des solutions tangibles dont les retombées techniques et économiques sont satisfaisantes.

3- 3 – 1 - Présentation des systèmes d'indexation et de recherche d'images:

Devant la nécessité de trouver rapidement un système efficace d'indexation et de recherche d'images, de nombreux laboratoires de recherche ont développé des

prototypes. Certains travaux ont débouché sur des logiciels commerciaux, et d'autres sont encore en développement [Vézina 99][NastarBoujemaa 99]. L'étude de ces systèmes existants s'est appuyée en grande partie des recherches Internet, où de nombreux sites font état de ce problème d'indexation. Les informations concernant les divers systèmes sont fournies en général par les personnes réalisant ces systèmes, donc ne sont pas toujours très objectives. De plus, lorsque des démonstrations sont accessibles, les bases d'images sont spécifiques aux développeurs et nous ne pouvons pas réaliser nos propres tests de performance. Ainsi, la comparaison entre tous ces systèmes est délicate.

a) Les systèmes étudiés:

Les systèmes existants sont nombreux et plus ou moins documentés. Cette étude s'est donc attachée à parcourir l'ensemble de ce qui se fait à l'heure actuelle dans le cadre de l'indexation et de la recherche d'images. Tous les systèmes n'ont pas été étudiés, et l'ensemble des fonctionnalités n'a pas été répertorié.

Deux aspects sont à étudier:

- L'indexation : tout ce qui porte sur l'extraction du contenu des images et son stockage dans la base de donnée
- La recherche (mise en correspondance) : tout ce qui porte sur les requêtes effectuées par l'utilisateur pour retrouver une image

b) Les différents types de systèmes:

Au niveau de l'*indexation* des fichiers, deux types de systèmes prédominent actuellement :

- L'indexation manuelle par champs alphanumériques (nombre ou texte) : les champs correspondent à des caractéristiques particulières de l'image (Nom, date, photographe, description...).
- L'indexation par le contenu : méthode plus récente, elle travaille à partir des informations visuelles de l'image (les pixels). Les caractéristiques extraites des images sont généralement de plusieurs ordres, et sont issues des techniques de traitement du signal (transformées diverses, filtres...) :
 - Les histogrammes de couleur représentant le nombre de pixels ayant la même couleur, ce qui traduit l'ensemble des couleurs et leur proportion dans l'image.
 - Les textures représentant les arrangements spatiaux de pixels, et les caractéristiques locales particulières des images.
 - Les formes représentant les zones homogènes extraites par une reconnaissance des contours.
 - Les concepts prenant en compte les objets représentés dans l'image (zones homogènes ayant des caractéristiques propres) avec leurs relations (disposition spatiale, importance).

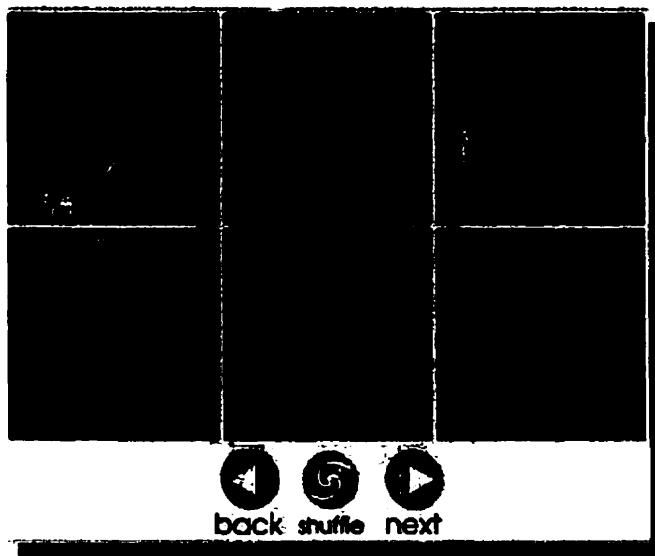
Dans les deux cas, les résultats sont retranscrits en caractères alphanumériques, pour pouvoir utiliser les moteurs actuels de recherche des bases de données, qui sont très performants. En effet, ces moteurs sont développés depuis des dizaines d'années, possèdent une bonne fiabilité et sont dans l'ensemble bien optimisés.

Pour la *recherche d'informations*, nous pouvons noter plusieurs méthodes, s'appuyant sur des principes complètement différents :

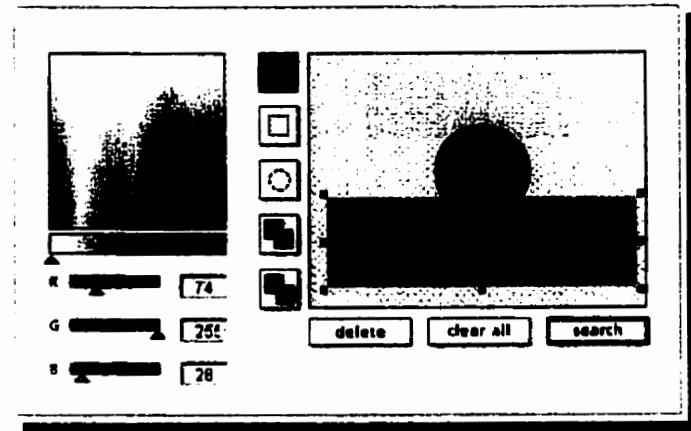
- Par mots-clés : la recherche porte sur un ou plusieurs champs (texte ou nombre) que l'utilisateur doit compléter. Le système recherche alors les images ayant rapport à ces mots-clés.

| | | |
|--|---|------------------------------------|
| Photo Caption | <input type="button" value="contains"/> <input type="text"/> | <input type="button" value="any"/> |
| Subject | <input type="button" value="begins with"/> <input type="text"/> | <input type="button" value="any"/> |
| Category | <input type="button" value="equals"/> <input type="text"/> | <input type="button" value="any"/> |
| Location | <input type="button" value="begins with"/> <input type="text"/> | <input type="button" value="any"/> |
| Date of Photo | <input type="button" value="equals"/> <input type="text"/> | Example: 1995-01-31 |
| Perspective | <input type="button" value="equals"/> <input type="text"/> | <input type="button" value="any"/> |
| Photo ID | <input type="button" value="equals"/> <input type="text"/> | Example: 7700 3262 3317 0010 |
| DWR ID | <input type="button" value="equals"/> <input type="text"/> | Example: 8310-742 |
| Film Format | <input type="button" value="equals"/> <input type="text"/> | Example: 2-1/4 negative |
| <input type="button" value="Search"/> <input type="button" value="Clear"/> | | |

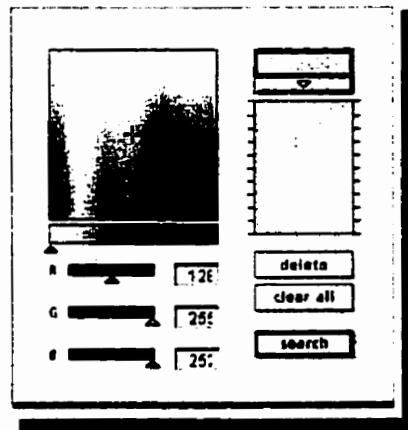
- Par l'exemple : la recherche se base sur une image choisie par l'utilisateur. Le choix s'effectue généralement en naviguant dans une série d'images aléatoires. Le système extrait alors les caractéristiques visuelles de l'image et recherche les images similaires, sur un des critères (couleur, texture, forme)



- Par le dessin : la recherche se base sur un croquis dessiné par l'utilisateur.



- Par la couleur : la recherche se base sur des histogrammes définis par l'utilisateur



- Par la navigation : l'utilisateur navigue dans une structure où toutes les images sont rangées par catégories ayant rapport au sujet de l'image.

c) Les termes employés:

Différents termes sont employés dans les fiches de résumé des systèmes, en voici la signification :

- Les opérateurs de recherche : dans les recherches par mots-clés, ce sont les associations de plusieurs requêtes différentes pour affiner le résultat. Nous pouvons répertorier plusieurs types d'opérateurs :
 - Booléens : et, ou, sauf (ex : voiture ET bleu)

- De proximité : près, avant, après (ex : bleu PRÉS voiture)
 - De consultation : dans, derniers (ex : bleu DANS couleur)
 - Arithmétiques :>,<,>=,<=,entre (ex : date > 14/05/1998)
-
- Les masques et les troncatures : ce sont des caractères de substitution qui sont également utilisés dans les recherches par mots-clés.

Le masque (symbole « ? ») permet de remplacer un et un seul caractère. Par exemple, la question ?ar? nous donne les réponses « Marc », « mars » et « part ». Le masque peut être utilisé à gauche, au centre ou à droite d'un mot et peut être répété autant de fois que souhaité.

La troncature est représentée par le symbole « * ».

3- 3 - 2 - Les fiches des systèmes:

Les systèmes sont résumés en fiches synthétiques apportant les caractéristiques principales de chaque système.

Ces fiches sont organisées comme suit :

- Nom du système
- Description du système : (Nom, fabricants/chercheurs, Domaines d'application)
- Fonctionnalités d'indexation
- Fonctionnalités de recherche
- Liens WWW concernant le système
- Résumé portant sur le logiciel en lui-même et sur ses apports dans le domaine de l'indexation des images.

Dix fiches sont présentées mais d'autres systèmes existent. Un certain nombre d'autres systèmes n'ont pas été présentés car ils n'apportent aucune nouveauté par

rapport à ceux étudiés ou alors la documentation n'était pas assez conséquente pour en établir un quelconque bilan.

3 - 3 - 3 - Bilan concernant les systèmes d'indexation et de recherche d'images:

Il faut voir ces logiciels de deux façons. D'un côté, il y a les applications commerciales destinées à être vendues et les applications de recherche. Les applications commerciales doivent avoir des résultats efficaces et des interfaces conviviales. Ainsi, des systèmes comme «Phraséa», largement utilisés par les professionnels de la photo (agences de presse notamment), ne prennent pas du tout en compte les caractéristiques visuelles des images. Cela peut se comprendre car les photos sont gérées par un groupe restreint de personnes (les journalistes de l'agence), qui savent extraire les informations propres à leur profession. De plus, «Phraséa» dispose de nombreux outils de recherche par mots-clés, donc possède une souplesse d'utilisation très efficace pour ce type de collection. Par contre, ces bases d'images seront sûrement difficilement exploitable par un utilisateur quelconque n'ayant rien à voir avec la profession.

Parallèlement à ce type de système, nous avons ceux qui se basent sur le contenu de l'image («Surfimage», «Blobworld»). Ils s'avèrent pratiques lorsque le contenu sémantique de l'image n'est pas recherché. Par exemple lorsque nous avons une idée de l'image que nous cherchons, il suffit d'en faire un croquis ou de décider des couleurs que nous souhaitons avoir dans l'image et de lancer la recherche. Par contre, si une image précise est recherchée, alors ce type de système est très fastidieux à utiliser.

Des variantes sont possibles : une navigation sommaire permet de rechercher un domaine ou une catégorie d'image, puis de rechercher une image par similitude de couleur, de forme, de texture («VisualSeek», «Netra»). Par contre, cela reste fastidieux pour rechercher une image particulière. De plus, le classement effectué limite la souplesse du système. Cela se passe bien pour des paysages ou des animaux,

mais reste ambigu pour des images d'actualités par exemple, pour lesquelles des classifications sont multiples sont possibles.

Dans cette étude, nous pouvons remarquer que les performances actuelles se portent sur une association de mots-clés et de caractéristiques visuelles («Virage», «Cypress», «QBIC»). Elle permet en effet d'obtenir de bons résultats pour un utilisateur quelconque. Par exemple, si l'utilisateur cherche des images spécifiques à «l'assassinat du président Kennedy», il n'a d'autre choix actuellement que d'effectuer sa recherche par mots-clés. Ensuite, une fois qu'il possède sa collection spécifique au sujet recherché, il doit encore faire des choix. C'est alors que la recherche par les caractéristiques visuelles va intervenir. S'il cherche une image extérieure quelconque, il choisira des images ayant plutôt un fort contraste et des couleurs assez claires (jour de beau temps), et s'il désire un gros plan de la voiture, alors il pourra dessiner un croquis de la scène qu'il souhaite visualiser.

Ce type de système se trouve très adapté et très souple d'utilisation, car il reprend les avantages des mots-clés et de la caractérisation par le contenu. Les mots-clés permettent de rechercher une image spécifique et les critères visuels une série ou un type d'images.

De plus, c'est à ce niveau que nous trouvons des APIs disponibles («Virage», «QBIC»). Ce sont des applications indépendantes qui peuvent être utilisées par d'autres programmes. Ainsi, il est possible de développer ses propres systèmes en utilisant les traitements et les développements effectués par «Virage» et «IBM» dans l'indexation et la recherche des images. Il faut toutefois noter que la plupart des indexations où des mots-clés sont présents restent manuelles ou semi-automatiques. En effet, la spécificité d'une image n'est pas encore extraite de façon automatique.

Certains systèmes comme «Photobook» travaillent dans ce sens en caractérisant l'image par des mots-clés génériques issus de concepts. Ces concepts (voiture, immeuble, rue, ville...) sont définis par leurs caractéristiques visuelles, et le système essaie de les retrouver dans l'ensemble des images au cours de l'indexation. La spécificité relative à une indexation manuelle n'est cependant pas atteinte. Les

concepts utilisés restent simples et ne permettent pas beaucoup de précision. Par exemple, un système de ce type n'a encore aucun moyen de distinguer par exemple le type de fleur qui se présente sur une image.

C'est l'orientation que prennent actuellement les recherches dans l'indexation et la recherche d'images. Les systèmes extraient les différents éléments d'une image, forment des relations entre ces objets (disposition spatiale, inclusion, hiérarchie), caractérisent ces objets (couleur, texture, forme), et bâtissent un modèle conceptuel de l'image. A partir de là, des programmes devraient identifier des objets de par leur caractéristiques (fleur, voiture, chien) et éventuellement créer de nouveaux concepts si le type n'est pas reconnu par le système. Ces systèmes sont donc reliés aux bases de données de connaissances et d'apprentissage. Comme exemple de ces systèmes, nous avons les systèmes *Cypress-Chabot*, *ImageMiner* et *Netra*.

Dans le tableau suivant ([tableau numéro 3.1](#)), nous résumons les caractéristiques des systèmes d'indexation et de recherche des images présentés précédemment.

| Nom | Adresser Internet | Type de document(s) | Méthode d'indexation | Méthode de recherche |
|--|---|---------------------------------|---|---|
| Phraséa | http://www.virage.com/ | Images, textes, vidéos, sons... | <ul style="list-style-type: none"> - Indexation textuelle basée sur la notion des mots-clés (indexation d'un texte descriptif relié au document multimédia) | <ul style="list-style-type: none"> - Recherche basée sur la notion des mots-clés (recherche textuelle) |
| Virage | http://www.virage.com/ | Images, vidéos, sons... | <ul style="list-style-type: none"> - Indexation des images par extraction des caractéristiques visuelles (histogrammes des couleurs, textures, formes...) - Indexation des vidéos par extraction des différents plans (l'indexation est faite pour chaque plan en se basant sur ses caractéristiques visuelles). - Indexation des sons en analysant les paroles (reconnaissance de paroles). <p>Pour chaque document le système associe un texte descriptif au moment de l'indexation.</p> | <ul style="list-style-type: none"> - Recherche textuelle sur les documents vidéos et sonores. - Recherche par caractéristiques visuelles ou textuelles des images. |
| Cypress (anciennement Chabot) | http://elib.cs.berkeley.edu/cypress.html | Images | Indexation visuelle qui se base sur l'histogramme de couleurs pour indexer visuellement le contenu de l'image. | Recherche en se basant sur les caractéristiques visuelles (couleurs choisies par l'utilisateur qui veut chercher des images contenant ces couleurs). |
| BlobWorld | http://elib.cs.berkeley.edu/photos/blobworld | Images | <ul style="list-style-type: none"> - Indexation de base effectuée sur le classement des images par mots-clés. - Indexation visuelle : Segmentation des images en régions homogènes (couleurs+textures). | <ul style="list-style-type: none"> - Recherche par mots-clés. - Recherche par l'exemple qui se base sur les régions homogènes de l'image et sur certains critères d'importance de ces régions |

| | | | |
|---------------------------|---|--------|--|
| | | | (couleurs, textures, formes, positions). |
| ImageMiner | http://www.tzi.uni-bremen.de/BV/ImageMine.html/home.html | Images | <ul style="list-style-type: none"> - Indexation basée sur le contenu : Couleurs, texture et forme des objets. - Indexation automatique avec la possibilité d'ajouter des descriptions attachées aux images (indexation par mots-clés). |
| Netra | http://vivaldi.ece.ucsb.edu/demos.html#Netra | Images | <ul style="list-style-type: none"> - Indexation basée sur le contenu : Couleur, texture, forme, position. Segmentation de l'image en régions homogènes. - Classement des images en domaines. |
| PhotoBook-FourEyes | http://vismod.www.media.mit.edu/~tpminku/photobook/ | Images | <ul style="list-style-type: none"> - Indexation basée sur le contenu : Couleur, texture et forme. L'indexation est manuelle, mais elle peut être assistée par le choix d'une image de la collection pour appliquer les mêmes critères de cette image pour les images à indexier. - Utilisation d'un texte descriptif pour chaque image (indexation textuelle). |
| QBIC | http://www.wqbic.almaden.ibm.com/ | Images | <ul style="list-style-type: none"> - Indexation basée sur le contenu : histogramme des couleurs, forme et textures. - Utilisation d'un texte accompagnant l'image. |

| | | | | |
|-------------------|---|-------------------|---|---|
| | | | | recherche. - Recherche par la couleur : Définition des pourcentages des couleurs que l'on veut voir apparaître. - Recherche sommaire par mots-clés. |
| SurfImage | http://www-rocq.inria.fr/cgi-bin/imedia/surfimage.cgi | Images. | - Indexation basée sur les critères visuels. | - Indexation basée sur le contenu : histogramme des couleurs, formes et textures. - Indexation textuelle : Utilisation des mots accompagnant l'image. - Emploi d'agents automatiques pour indexer les images présentées sur Internet. |
| VisualSeek | http://www.ctr.columbia.edu/~jrsmith/ | Images et vidéos. | - Recherche par l'exemple : Proposition d'images aléatoires puis affinement du choix. | - Recherche par l'exemple : Utilisation des critères de couleurs ou de texture. - Recherche par le dessin : Utilisation d'un croquis (forme et couleur) comme base de recherche.. - Recherche par la couleur : Définition des histogrammes de couleurs que l'on veut voir apparaître. - Recherche par mots-clés. |

Tableau 3.1 : Les caractéristiques des systèmes d'indexation et de recherche des images

D'après l'étude de tous ces systèmes d'indexation et de recherche, que ce soient des pages web sur Internet, des images ou d'autres documents visuels et audiovisuels, nous avons constaté que chaque système a ses propres caractéristiques. Il existe des systèmes qui se basent sur la notion des mots-clés pour l'indexation et la recherche, d'autres sur des aspects visuels et d'autres sur la notion de mots-clés et l'aspect visuel en même temps. Il y a des systèmes que nous pouvons utiliser localement (nous les téléchargeons et nous les utilisons) et il y en a d'autres que nous ne pouvons utiliser que sur le réseau Internet....etc

En conclusion, nous ne pouvons ni améliorer, ni adapter, ni intégrer l'un de ces systèmes pour notre travail à cause des contraintes mentionnées ci-dessus. Dans notre travail, nous voulons utiliser une seule approche d'indexation et de recherche pour tous les formats de documents à indexer ou à rechercher. Nous voulons, aussi, utiliser une approche plus efficace que celle basée sur les mos-clés et nous voulons gérer nos documents et notre propre base de données documentaires.

Pour pouvoir utiliser une seule approche d'indexation et de recherche de documents de tous formats, il faut présenter ces documents par une seule structure, en utilisant un seul langage.

Pendant notre étude bibliographique, nous avons exploré un standard intitulé «MPEG-7» qui est un noyau standardisé permettant la description des documents de tous formats en utilisant une seule structure. La description de ces documents se base sur un seul langage qui est le langage XML (eXtended Markup Langage).

3- 4 - Présentation du standard MPEG-7:

3- 4 – 1 - Présentation générale du standard MPEG-7:

Le standard MPEG-7 connu sous le nom «Multimedia Content Description Interfaces» a pour but de proposer un noyau standardisé permettant la description du contenu des données audiovisuelles dans les environnements multimédias. MPEG-7

étend les fonctionnalités limitées des solutions propriétaires actuelles dans l'identification des contenus, notamment en ajoutant de nouveaux types de données. En d'autres termes, MPEG-7 spécifie un ensemble standard de descripteurs qui peuvent être utilisés pour décrire des types variés d'information multimédia. MPEG-7 spécifie également des structures prédéfinies pour les descripteurs et leurs relations, en même temps que des éléments de définition pour ses propres structures. Ces structures sont appelées des schémas de description (DS) [OIN 2000]. La définition de nouveaux schémas de description est réalisée en utilisant un langage spécial, le langage de définition des descriptions (DDL ou Description Definition Langage), qui fait également partie du standard. La description peut être associée au contenu lui-même pour permettre des recherches rapides et efficaces [OIN 2000].

Les descripteurs MPEG-7 ne dépendent pas de la façon dont a été codé ou stocké le contenu. Il est possible de créer une description MPEG-7 d'un film analogique ou d'un journal imprimé sur papier.

Voici un exemple qui nous permet d'illustrer les deux niveaux extrêmes de description :

- Bas niveau : forme, taille, texture, couleurs, trajectoire, position des objets
- Haut niveau : « C'est une scène comportant un chien gris qui aboie près d'un arbre »

Le niveau d'abstraction est relié à la façon dont les caractéristiques sont extraites : la plupart des caractéristiques de bas-niveau peuvent être extraites complètement automatiquement, tandis que les caractéristiques de haut-niveau demandent plus d'intervention humaine.

En plus d'une définition du contenu, il est nécessaire d'avoir d'autres informations reliées aux données multimédia :

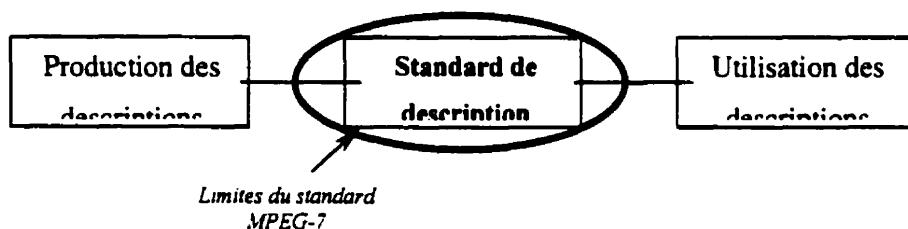
- La forme ou le type de codage du fichier
- Les conditions d'accès du support : droits d'auteur, prix...
- La classification : classement parental, numéro de catégorie...
- Des liens vers des supports pertinents

- Le contexte : l'événement de l'enregistrement

Dans de nombreux cas, il est intéressant d'associer des descriptions textuelles aux supports audiovisuels. Cependant, il faut bien faire attention que la description reste aussi indépendante que possible du langage employé.

La façon dont les données MPEG-7 sont utilisées pour répondre aux requêtes est en dehors du standard. Par ce principe, tout type de support audiovisuel peut être retrouvé par tout type de recherche. Cela signifie, par exemple, qu'un support vidéo peut être retrouvé à partir de ses caractéristiques de ses séquences d'images et des aspects sonores reliés à ce vidéo [OIN 2000]... C'est au moteur de recherche de faire correspondre la requête à la description MPEG-7.

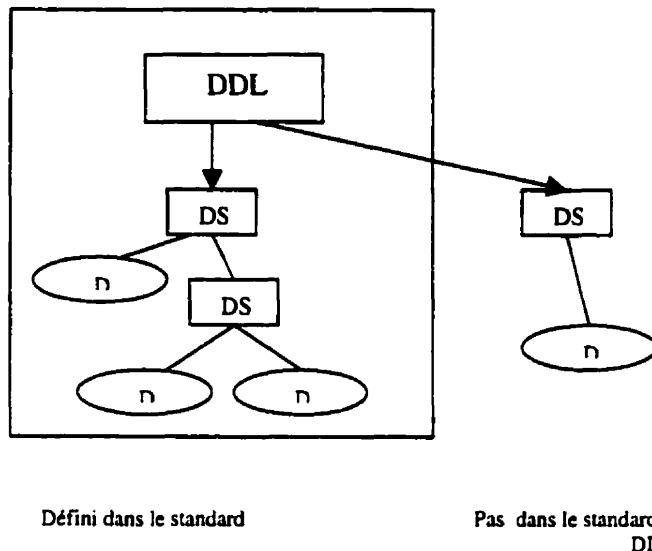
Nous pouvons résumer par le schéma suivant l'étendue du standard par rapport à ses applications.



En fait, le standard permet d'assurer une compatibilité entre les descriptions de documents, mais ne s'occupe pas de la réalisation des logiciels permettant de gérer ces descriptions. Cela permet de garder une certaine flexibilité et une étendue maximale pour les descriptions. Cependant, tant que la norme ne sera pas stabilisée, les outils de traitement des descriptions seront incomplets, donc peu performants par rapport aux avantages offerts par le standard (les outils n'exploitent pas les avantages du standard à cent pour cent) [OIN 2000].

Un grand avantage du standard est l'extensibilité des descriptions : le DDL donne la possibilité de créer des schémas de descriptions qui peuvent servir de base à la

réalisation de nouveaux descripteurs. Les structures des documents peuvent donc être de diverses formes et de nouveaux types de documents peuvent apparaître, sans que la norme ne devienne obsolète. Le schéma ci-dessous résume l'extensibilité du standard.



3- 4 – 2 - La terminologie du standard:

Dans le but d'assurer une structure de description commune à tous les types des documents multimédia, facilement manipulables par les outils de gestion de ces documents, MPEG-7 définit les notions suivantes:

- *Les descripteurs (D)* : Ce sont des présentations de caractéristiques des éléments existants dans les documents. Ils définissent la syntaxe et le sens de chaque représentation de ces caractéristiques [OIN 2000].
 - *Les schémas descripteurs (SD)* : Ils spécifient la structure et le sens des relations entre leurs composantes, qui peuvent être soit des descripteurs soit des schémas descripteurs [OIN 2000].
 - *Un langage de définition des descriptions (LDD)* : Il permet la création de nouveaux descripteurs ou schémas descripteurs. Il permet également d'étendre et de modifier les descripteurs et les schémas descripteurs existants [OIN 2000].

- *Les outils et les systèmes* qui permettent de générer les descripteurs et les schémas descripteurs du standard MPEG-7, qui permettent de les gérer, les manipuler..etc.
[OIN 2000]

Nous pourrons noter deux termes supplémentaires employés dans le standard :

- *Valeur d'un descripteur* : instance d'un descripteur pour un ensemble de données.

Remarque: les valeurs sont combinées avec les schémas de descriptions pour former une description

- *Description* : consiste en un SD et en un ensemble de valeurs de descripteurs décrivant la donnée.

3- 4 – 3 - Les éléments du standard:

Le standard MPEG-7 standardise les descripteurs et les schémas descripteurs des éléments visuels, audio et multimédia (qui sont des éléments ni purement visuels, ni purement audio). Il standardise aussi des éléments de base pour décrire le contenu audiovisuel du média. La description des documents multimédia se fait suivant cinq points de vue différents. Ces points de vue sont les suivants :

- Création et production
- Média
- Utilisation
- Aspects structurels
- Aspects conceptuels

Les trois premiers points de vue s'adressent principalement aux informations reliées à la *gestion du contenu*. Tandis que les deux derniers points de vue s'intéressent à l'aspect *description du contenu*.

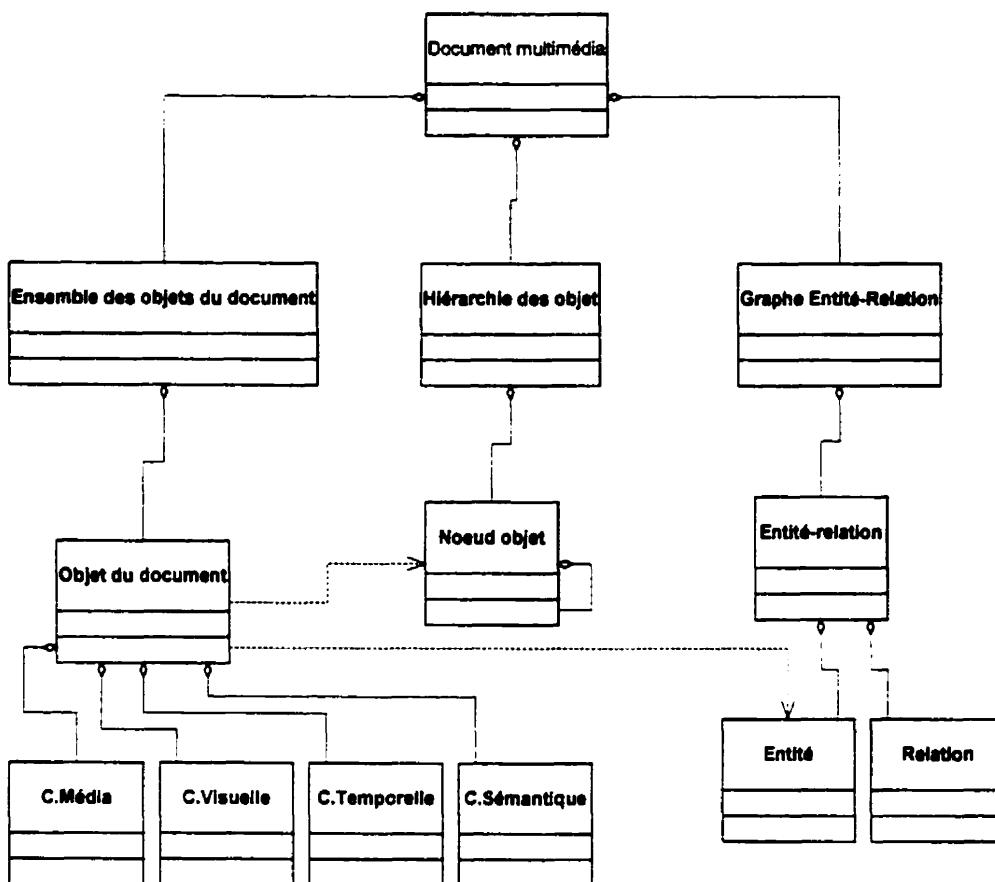
Ces cinq points de vue sont présentés dans le tableau suivant :

| Ensemble des éléments | Fonctionnalité |
|------------------------|---|
| Création et production | Des méta-informations qui décrivent la création et la production du contenu, elles décrivent le titre, le créateur, le but de la création... |
| Utilisation vie | Des méta-informations reliées à l'utilisation du contenu : Elles comportent les droits d'accès, des informations financières, des droits de publication... Ces informations peuvent faire l'objet de changement durant la durée de vie du contenu audio-visuel. |
| Média stockage : | Ces informations décrivent les caractéristiques de Format, éléments pour identifier le média... |
| Aspects structurels | Des descriptions d'un point de vue contenu : Ces informations décrivent les segments qui peuvent représenter des composantes spatiales, temporelles ou spatio-temporelles du contenu audio-visuel. Chaque segment peut être décrit par les caractéristiques suivantes (la couleur, la texture, la forme, le mouvement, d'autres caractéristiques audio...) et quelques informations sémantiques élémentaires. |
| Aspect Conceptuels | Des descriptions du contenu audiovisuel d'un point de vue conceptuel. Ces informations ne sont pas indiquées dans les documents techniques du standard MPEG, car elles sont en cours de standardisation. |

En fonction des applications, quelques unes de ces descriptions peuvent ne pas être prises en compte.

3- 4 – 4 – La méta-structure de la description des documents dans le standard MPEG-7:

Nous avons dit qu'à l'aide du standard MPEG-7, nous pouvons décrire n'importe quel document textuel ou audiovisuel en nous basant sur une même structure à l'aide d'un seul langage (XML). Dans cette section, nous présentons la méta-structure qui permet de présenter un document multimédia (en utilisant la notation UML), puis nous présentons comment cette méta-structure peut être instanciée pour des formats bien spécifiques (textes, images, vidéos..)



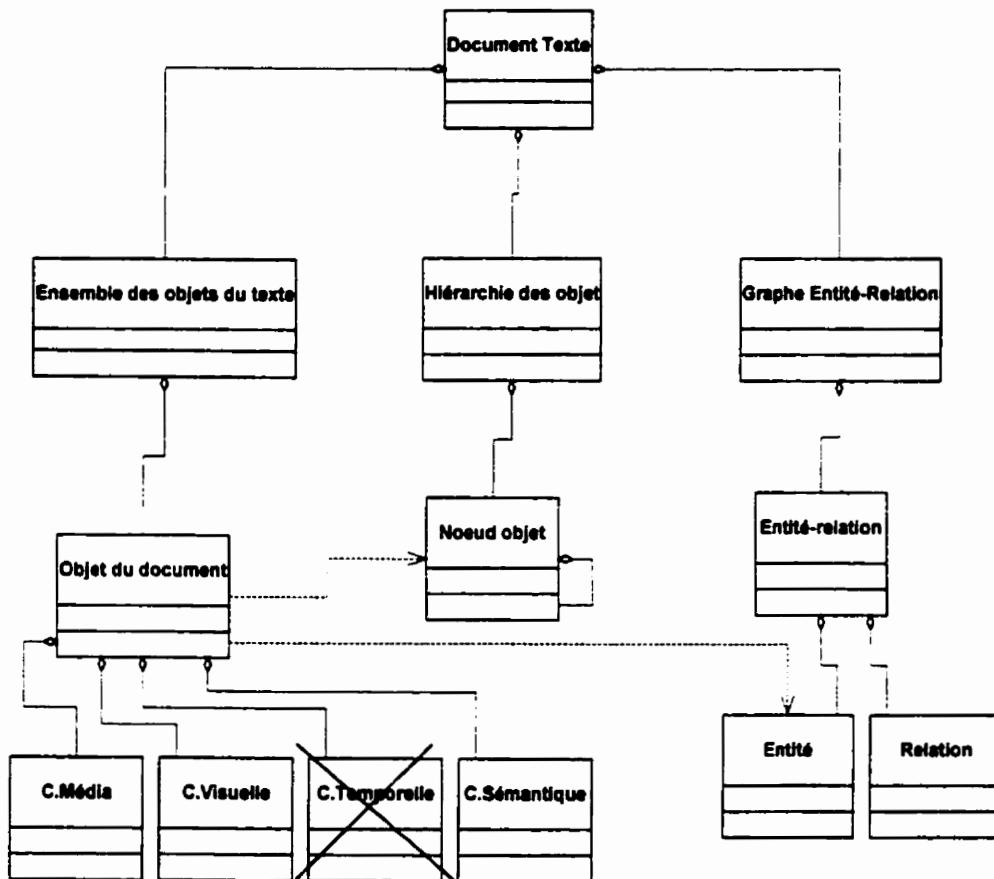
Type de la relation:

- Spatiale
- Structurelle
- Conceptuelle (Sémantique)
- Temporelle

Dans ce diagramme, nous nous basons sur les différents éléments présentés par le standard MPEG-7 pour décrire un document multimédia (les éléments: Création et production, média, utilisation, aspects structurels et aspects conceptuels). En plus des caractéristiques multimédia et structurel, un document peut être décrit en se basant sur son contenu visuel. Un document peut être décrit à l'aide de plusieurs objets organisés en une forme hiérarchique et qui ont des relations entre eux. Les objets sont décrits par une structure hiérarchique et les relations entre ces objets vont être décrites par une structure de graphe entité-relation (les entités sont les objets qui décrivent les documents et les relations sont les relations entre ces objets (spatiales, structurelles, conceptuelles, temporelles...)).

Cette structure est générale, nous pouvons l'instancier pour tout genre de documents. Dans les diagrammes suivant, nous instancions ce diagramme pour les documents textuels, images fixes, images animées et vidéos.

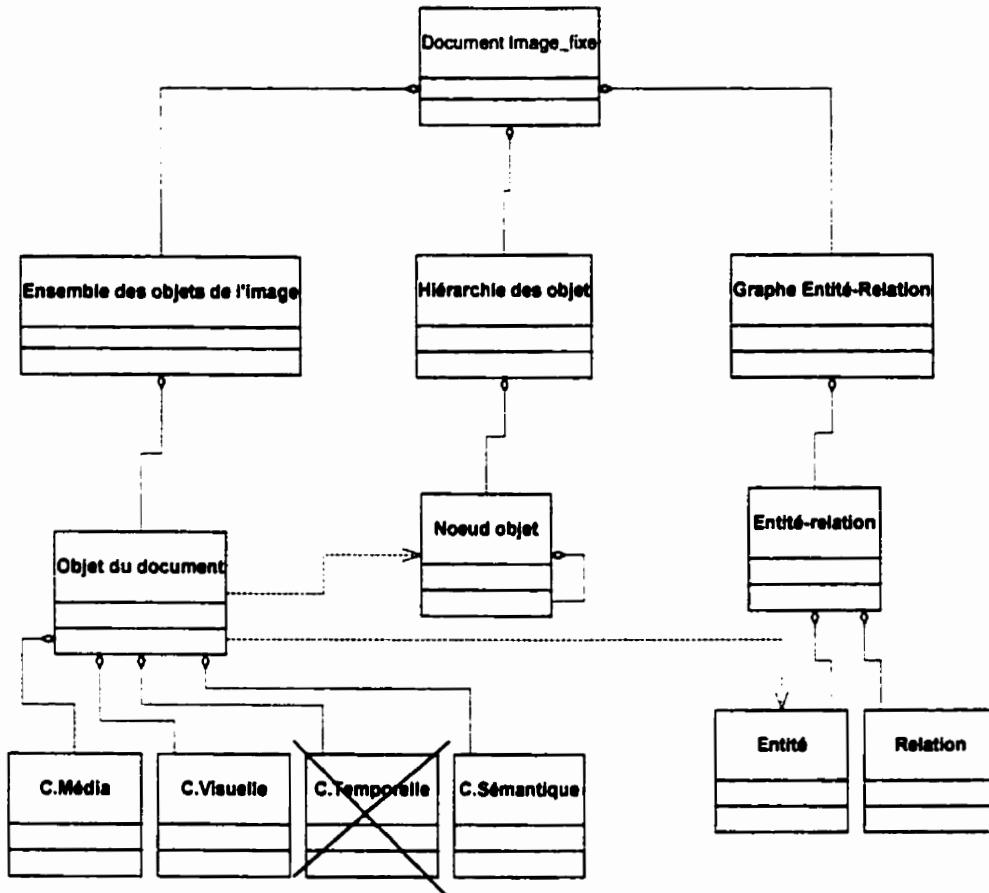
Instanciation de la mét-a-structure pour le document Texte : schéma descripteur du document Texte :



Type de la relation:

- Spatiale
- Structurelle
- Conceptuelle (Sémantique)

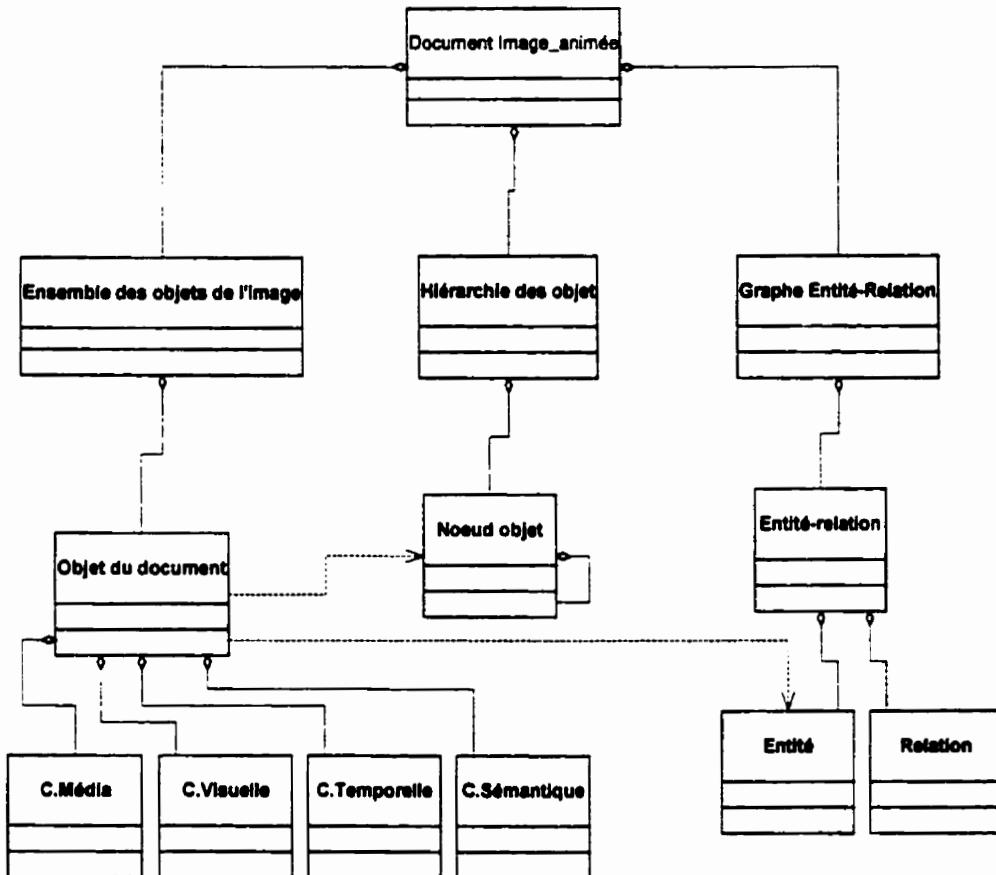
Instanciation de la mét-astructure pour le document Image_fixe (ou Frame): schéma descripteur du document Image_Fixe (Frame):



Type de la relation:

- Spatiale
- Structurelle
- Conceptuelle (Sémantique)

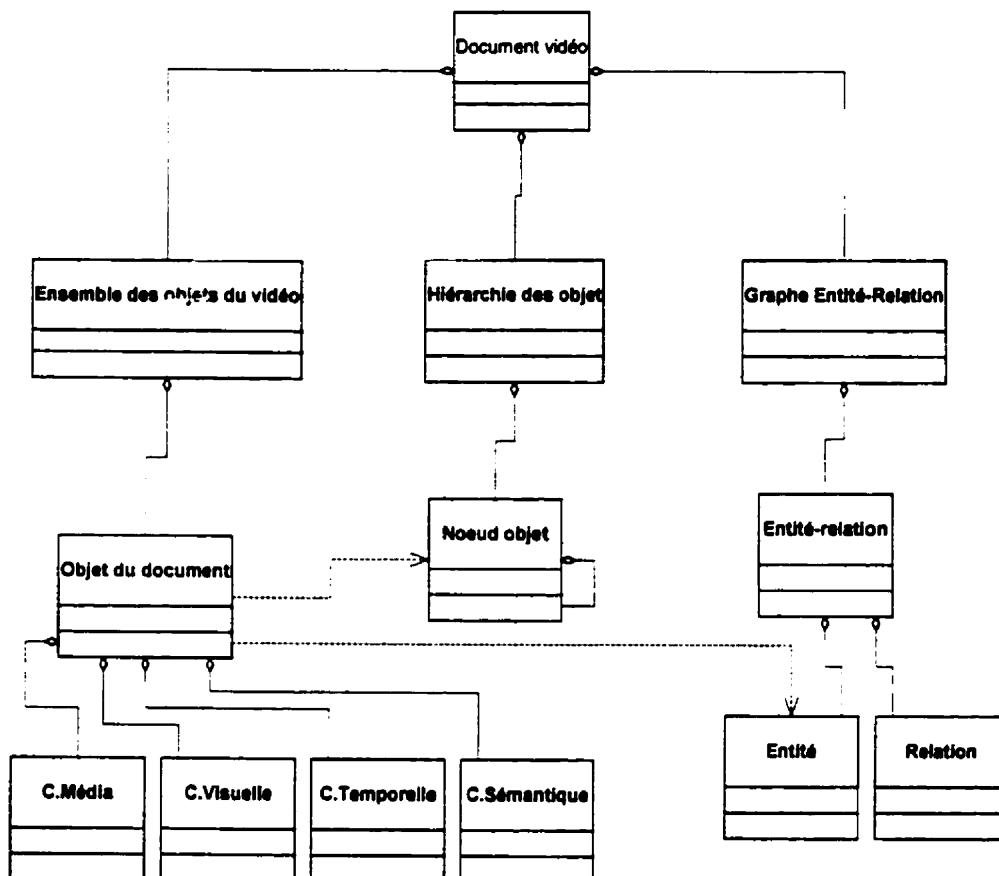
**Instanciation de la mét-a-structure pour le document Image_Animée:
schéma descripteur du document Image_Animée :**



Type de la relation:

- Spatiale
- Structurelle
- Conceptuelle (Sémantique)
- Temporelle

Instanciation de la mét-astructure pour le document vidéo : schéma descripteur du document vidéo :



Type de la relation:

- Spatiale
- Structurelle
- Conceptuelle (Sémantique)
- Temporelle

Types des relations utilisées par le standard MPEG-7:

Nous avons vu, dans les diagrammes de représentation des documents présentés précédemment, qu'un document est représenté par une liste d'objets structurés sous forme d'une hiérarchie, et reliés par des relations de différents type. Le standard MPEG-7 présente une liste initiale de relations qui les organise sous une forme hiérarchique [OIN 2000]. La liste de ces relations est présentée dans le tableau suivant:

| | |
|---------------------------------------|--|
| Relation spatiale Directionnelle | TopOf BottomOf RightOf LeftOf UpperLeftOf UpperRightOf LowerLeftOf LowerRightOf |
| Topologique | AdjacentTO NeighboringTo Nearby Within Contain |
| Relation Sémantique (Conceptuelle) | RelativeOf BelongTo PartOf RelatedTo SameAs IsA ConsistOf |
| Relation Temporelle Directionnelle | Before After ImmediatelyBefore ImmediatelyAfter |
| Topologique | CoBegin CoEnd Parallel Sequential Overlap Within |

| | |
|-----------------------|-------------------|
| | Contain Nearby |
| Relation Structurelle | |

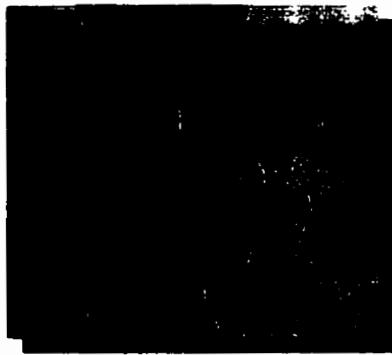
Les caractéristiques des objets :

Selon MPEG-7, chaque objet décrivant le document a des caractéristiques que nous devons les définir. Les caractéristiques présentées dans ce standard sont présentées dans le tableau suivant :

| | |
|------------|---|
| Media | File Format File Size Color Representation Resolution Data File Location Modality Transcoding Author Date Of Creation |
| Visuelle | Color (Color Histogram, Dominant Color, Color Coherence Vector, Visual Sprite Color) Texture (Tamura, MSAR, Edge Direction Histogram, DCT coefficient Energin, Visual Sprite Texture) Position Size Shape Orientation Motion Editing Effect Camera Motion |
| Sémantique | Texte Annotation Who What Object What Action Why When Where |
| Temporelle | Start Time End Time Duration |

Exemple général:

Comme exemple, nous prenons l'image présentée dans la figure suivante. Nous allons décrire cette image en nous basant sur notre diagramme instancié pour le document **Image_fixe**.



La liste des objets de l'image :

Objet 0 : L'image

Objet 1 : La personne ou l'enfant

Objet 3 : L'instrument musical

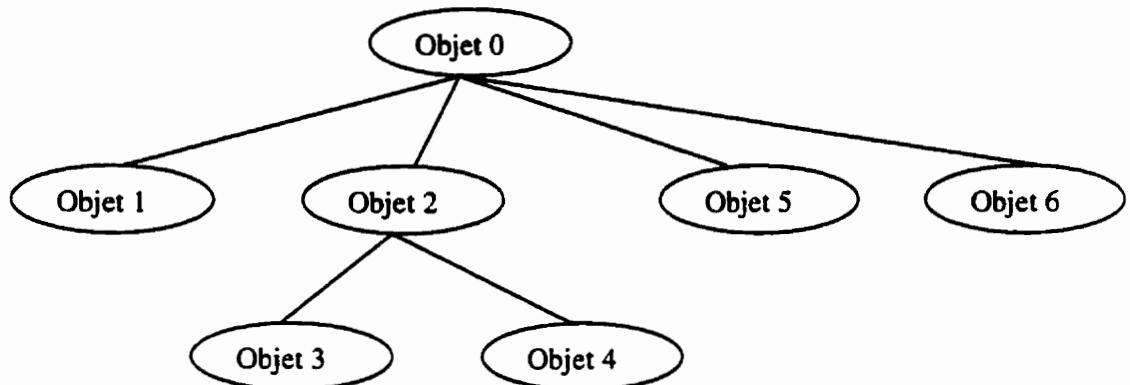
Objet 4 : Violent

Objet 5 : L'archet

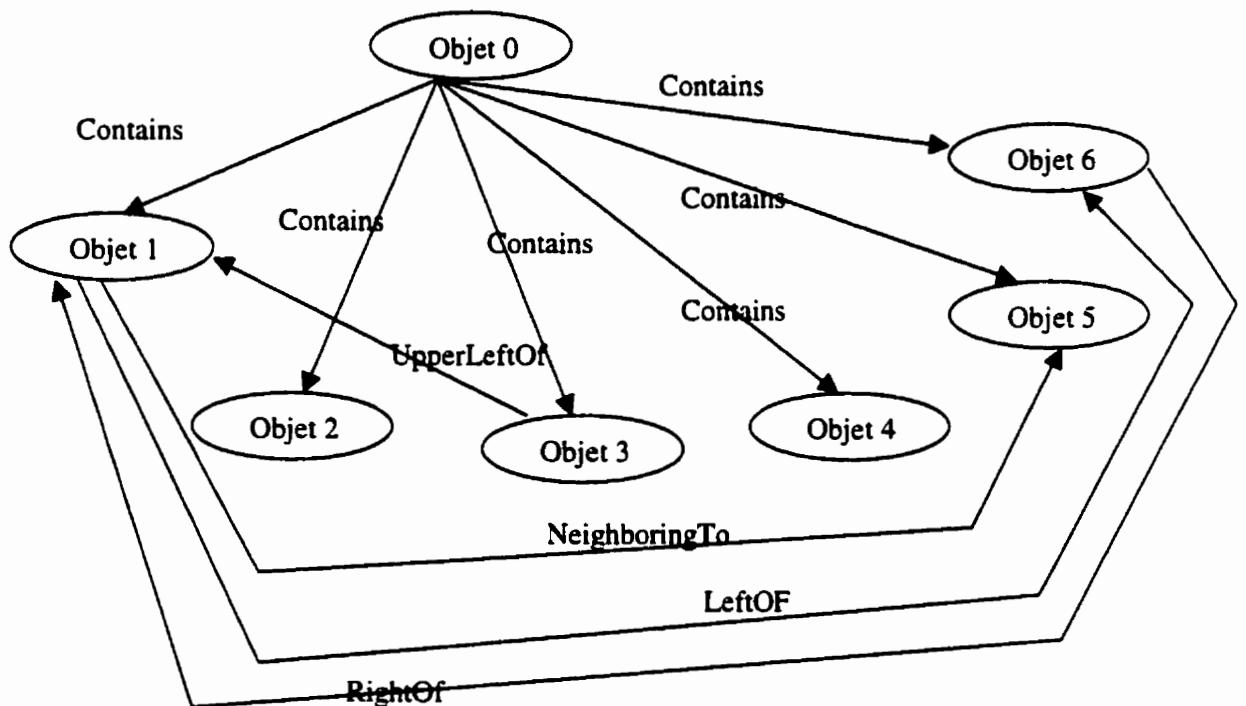
Objet 6 : La rue

Objet 7 : La fontaine

La hiérarchie des objets : Type Physique

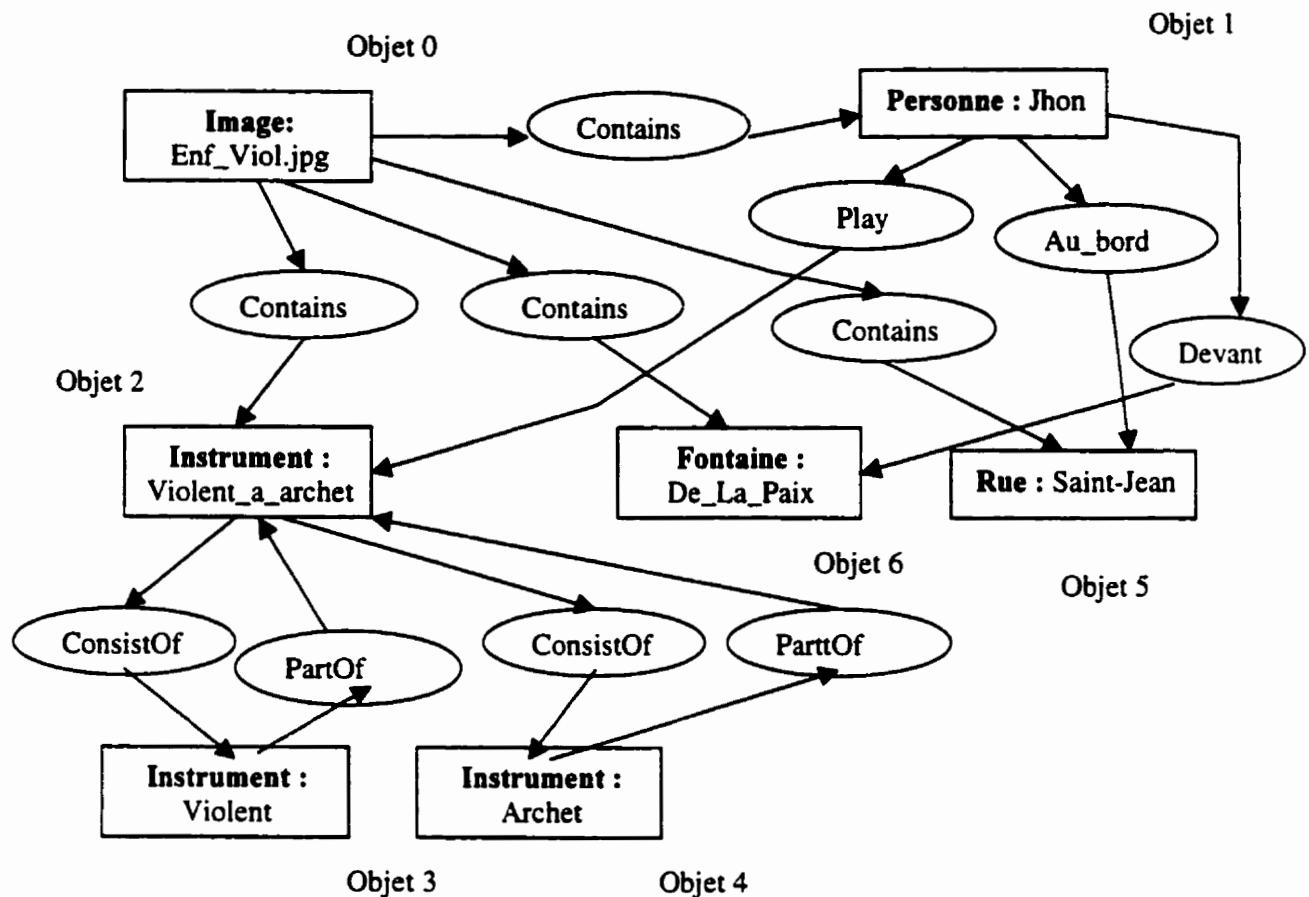


Le graphe Entité-Relation Spatial :



Nous pouvons représenter ce graphe, à l'aide du formalisme des graphes conceptuels, c'est à dire à l'aide des concepts et des relations conceptuelles.

Le graphe Entité-Relation Sémantique (Conceptuel) :



Les caractéristiques des objets:

* Objet 0 : L objet Image

Media

File Format: GIF

File Size: 45 K

Color Representation: Histogramme

Resolution: 45 x 45

Data File Location : c :\Data\Image\

Modality Transcoding:

Author:Ali

Date Of Creation:24/01/2000

Visuelles

Color (Color Histogram, Dominant Color, Color Coherence Vector, Visual Sprite Color):

Texture (Tamura, MSAR, Edge Direction Histogram, DCT coefficient Energin, Visual Sprite Texture):

Position: Vide

Size: Vide

Shape: Rectangle

Orientation: Vide

Motion: Vide

Editing Effect: Vide

Camera Motion: Vide

Semantiques

Texte Annotation:

Who: Enfant

What Object: Violent

What Action: Jouer

Why: Vide

When: un samedi

Where: devant la fontaine

Temporelles

Start Time: Vide

End Time: Vide

Duration: Vide

Remarque: Pour chaque objet, nous devons définir ces caractéristiques, et nécessairement il y a des caractéristiques qui n'existent pas pour quelques objets, comme par exemple les caractéristiques temporelles pour l'objet image. Nous pouvons bénéficier des descripteurs définis par le standard MPEG-7 pour enrichir les descripteurs de nos objets.

3- 4 - 5 - Bilan sur MPEG-7:

Le DDL n'est pas encore bien défini et reste au stade de l'ébauche car toutes les exigences requises initialement ne sont pas remplies par les propositions actuelles. Par contre, le langage XML (Extended Markup Language) a été choisi comme syntaxe du DDL MPEG-7. XML offre la possibilité d'avoir un langage avec une sémantique orientée-objet, et de créer des contraintes structurelles, relationnelles, ainsi que de définir des types de données particulières.

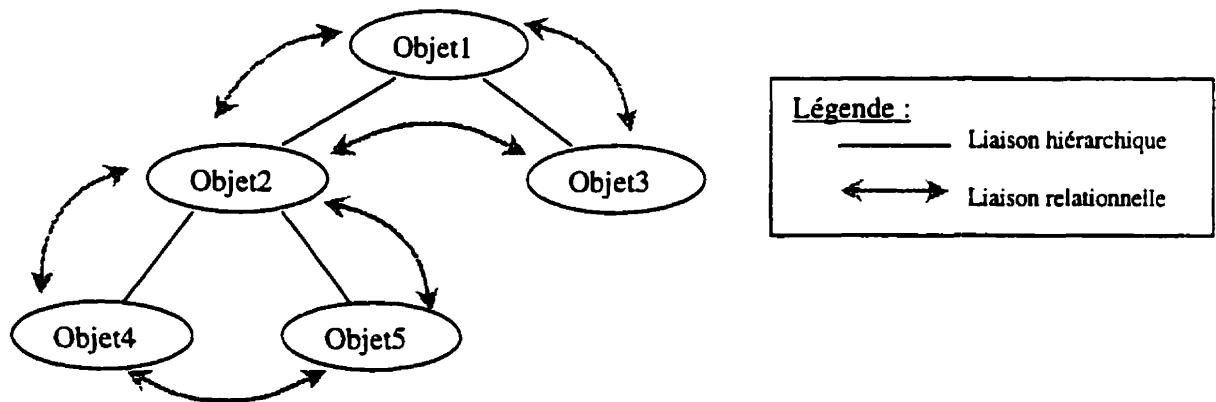
La structure générale d'une description se basant sur XML peut être résumée par la syntaxe suivante :

```
<objet1
  <objet2
    caractéristiques de l'objet2
    <objet4
      caractéristiques de l'objet4
      <relations de l'objet4
      </relations
    </objet4
    <objets5
      caractéristiques de l'objets5
      <relations de l'objets5
      </relations
    </objets5
    <relations de l'objet 2
    </relations
  </objet2
  <objet3
    caractéristiques de l'objet3
    <relations de l'objet 3
    </relations
  </objet3
  caractéristiques de l'objet1
  <relations de l'objet 1
  </relations
</objet1
```

Ce bout de code XML décrit un document multimédia en se basant sur le standard MPEG-7. Selon ce standard, nous avons dit qu'un document multimédia est représenté par une liste d'objets organisés en une structure hiérarchique, qui ont des relations entre eux et tels que chacun de ces objets possède ses propres caractéristiques. Dans ce code XML nous décrivons ces objets en utilisant des balises correspondant à chacun de ses objets, des balises correspondant aux relations existant entre les objets et des balises pour décrire les caractéristiques de chaque objet. Les balises sont imbriquées afin de représenter la structure hiérarchique des objets qui décrivent le document.

Les objets décrits peuvent être de toutes sortes, que ce soit un document ou une partie de document. Cette structure permet de définir des hiérarchies et des relations.

Pour l'exemple donné ci-dessus, nous aurons la hiérarchie des objets suivante:



Ainsi, une description assez avancée d'éléments multimédias peut être effectuée.

La syntaxe du standard MPEG-7 est donc bâtie sur les schémas XML. Cela permet de bénéficier des avantages de XML (lisibilité, extensibilité), tout en spécifiant une structure de données propre au standard MPEG-7 : le langage de définition des descriptions, les schémas de description et les descripteurs. Bien que cette structure ne

soit pas complètement définie, certains documents et travaux nous apportent des informations concernant la syntaxe même des descripteurs [OIN 2000].

3- 5 – Conclusion:

Dans ce chapitre, nous avons étudié quelques systèmes d'indexation et de recherche de documents textuels, images ou multimédia. La majorité de ces systèmes existe sur le réseau Internet. Nous avons constaté que nous ne pouvons pas utiliser ces systèmes ou les intégrer dans notre travail car ils sont plus ou moins dépendants ou spécifiques à certains types de documents. Nous avons aussi étudié un standard intitulé «MPEG-7» qui permet de présenter des documents de différents formats en utilisant une seule structure et un langage unique. Mais, malheureusement, ce standard n'était pas totalement défini lors de notre recherche.

En étudiant tous ces systèmes et ce standard, nous avons dégagé plusieurs techniques et approches d'indexation et de recherche des documents multimédia. Dans le chapitre suivant, nous allons présenter quelques approches les plus utilisées et nous allons choisir l'approche sur laquelle va se baser notre système.

Chapitre 4

Les techniques proposées et la technique retenue

4- 1 – Introduction:

Pendant l'étude des systèmes d'indexation et de recherche d'information textuelle ou multimédia, un certain nombre de techniques d'indexation et de recherche a été dégagé. Dans ce chapitre, nous présentons la technique la plus utilisée par ces systèmes: la technique basée sur la notion des mots-clés, nous présentons aussi ses limites. Également, nous présentons les techniques que nous avons envisagées pour notre système à savoir la technique purement sémantique et celle basée sur la notion des expressions. Enfin, nous présentons la technique que nous allons retenir pour notre système.

4- 2 – La technique basée sur la notion des mots-clés:

4- 2 – 1 - Présentation de la technique:

La technique d'indexation et de recherche basée sur des mots-clés est la technique la plus facile et la plus utilisée par la plupart des systèmes d'indexation et de recherche existants. Cette technique se base sur les étapes suivantes:

- Une notice est constituée pour chaque document (cette notice contient entre autre tous les mots-clés décrivant le document).
- Extraction des mots-clés pertinents de la notice en se référant par rapport à un dictionnaire de descripteurs.
- Un fichier index est construit. Il contient tous les mots-clés et permet une lecture optimisée pour faire une recherche plus rapide sans parcourir tout le document.

Pour assurer une bonne recherche dans les systèmes qui utilisent cette technique, il faut bien suivre des règles pour choisir les-mots clés de la requête.

Les mots-clés doivent correspondre *réellement* à une recherche et non pas à un thème de recherche. Par exemple il est peu probable qu'un utilisateur fasse une recherche avec le mot «*logiciel*», le résultat va être énorme car ceci est très large, car il y a plusieurs types de logiciels (des logiciels de messageries électroniques, des logiciels de comptabilité, de gestion...etc).

4- 2 – 2 - Limites de la technique:

La plupart des outils existants d'indexation et de recherche des documents utilisent la technique basée sur les mots-clé dans leurs processus d'indexation et de recherche des documents. Cette technique présente une limite, car elle fournit aux utilisateurs des résultats non pertinents dans plusieurs cas. En fait, le résultat de la recherche comporte un bruit énorme (nombre de documents non pertinents par rapport à la requête).

Pour présenter cette limite, nous présentons deux exemples de systèmes qui utilisent la technique d'indexation et de recherche en question.

Exemple 1:

Prenons comme exemple le système de recherches le plus utilisé dans le monde et qui exploite cette technique: le système de recherche sur Internet intitulé *Altavista*

(<http://www.altavista.com> ou <http://www.av.com>). Prenons la requête suivante : « visite urgences quebec asthme ». Le résultat de cette requête, présenté dans l'écran de la figure numéro 4.1, contient un bruit énorme. Certains documents-résultats parlent des «visites touristiques au Québec», d'autres parlent du «Québec» en général...etc. En fait, le système affiche en résultat tous les documents qui contiennent soit le mot «visite», soit le mot «urgences», soit le mot «Québec», soit le mot «asthme» par l'utilisation d'un «OU» exclusif implicite dans la requête.

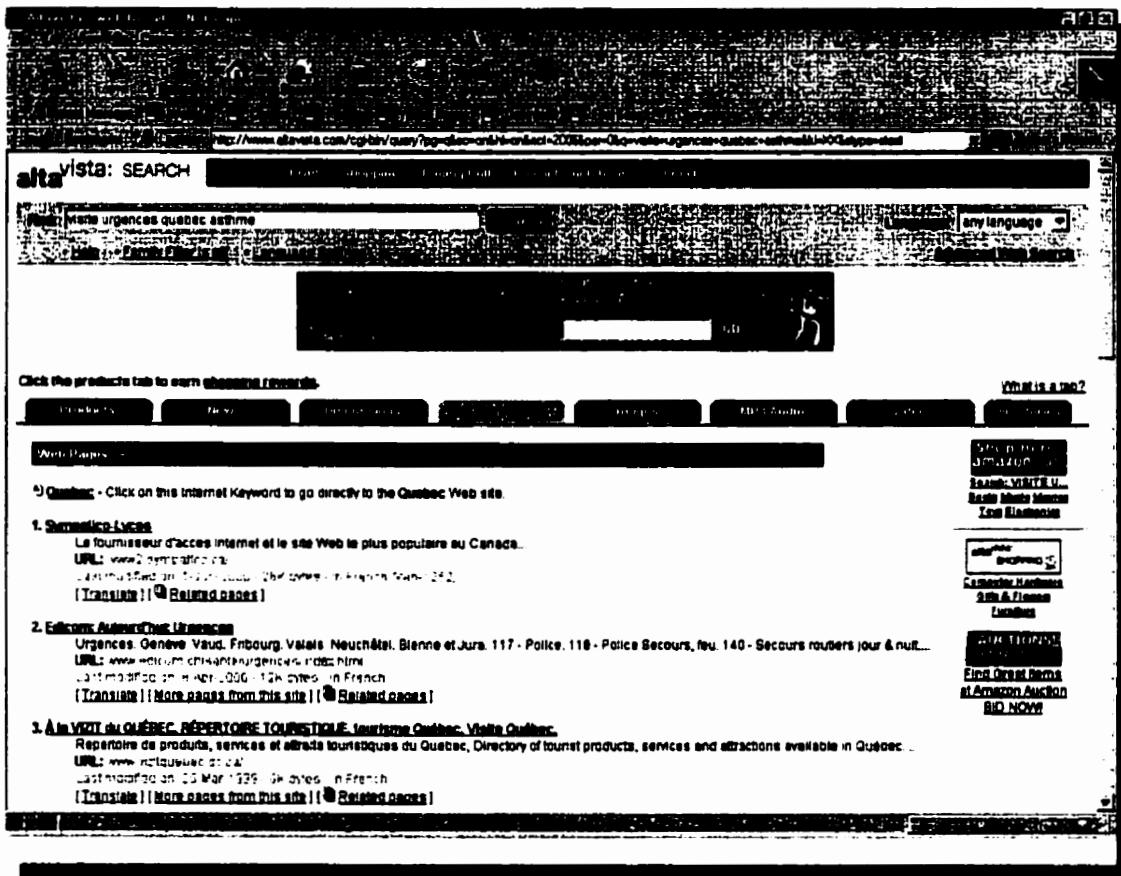


Figure numéro 4.1: Présentation des résultats (Altavista: Recherche simple)

Pour palier cet inconvénient, et dans le but de présenter un résultat plus pertinent à l'utilisateur, les développeurs du moteur de recherche Altavista ont ajouté une fonctionnalité intitulée «*recherche avancée*». Ce type de recherche n'est qu'une recherche logique qui introduit une liaison logique entre les mots. Dans l'écran de la figure numéro 4.2, nous donnons la requête suivante au moteur Altavista : «visite

AND urgences AND Québec AND asthme». Par cette liaison logique entre les mots-clés de la requête, le système affiche en résultat tous les documents qui contiennent ces mots-clés, mais là aussi, les documents, contiennent ces mots-clés avec un lien logique et aucun lien sémantique. Par exemple, nous pouvons avoir en résultat des documents qui parlent des «visites touristiques», du «Québec», des «urgences aux hôpitaux de Genève» et de «la maladie de l'asthme en Afrique». En plus, le système affiche en fin de liste tous les documents qui contiennent l'un de ces mots-clés.

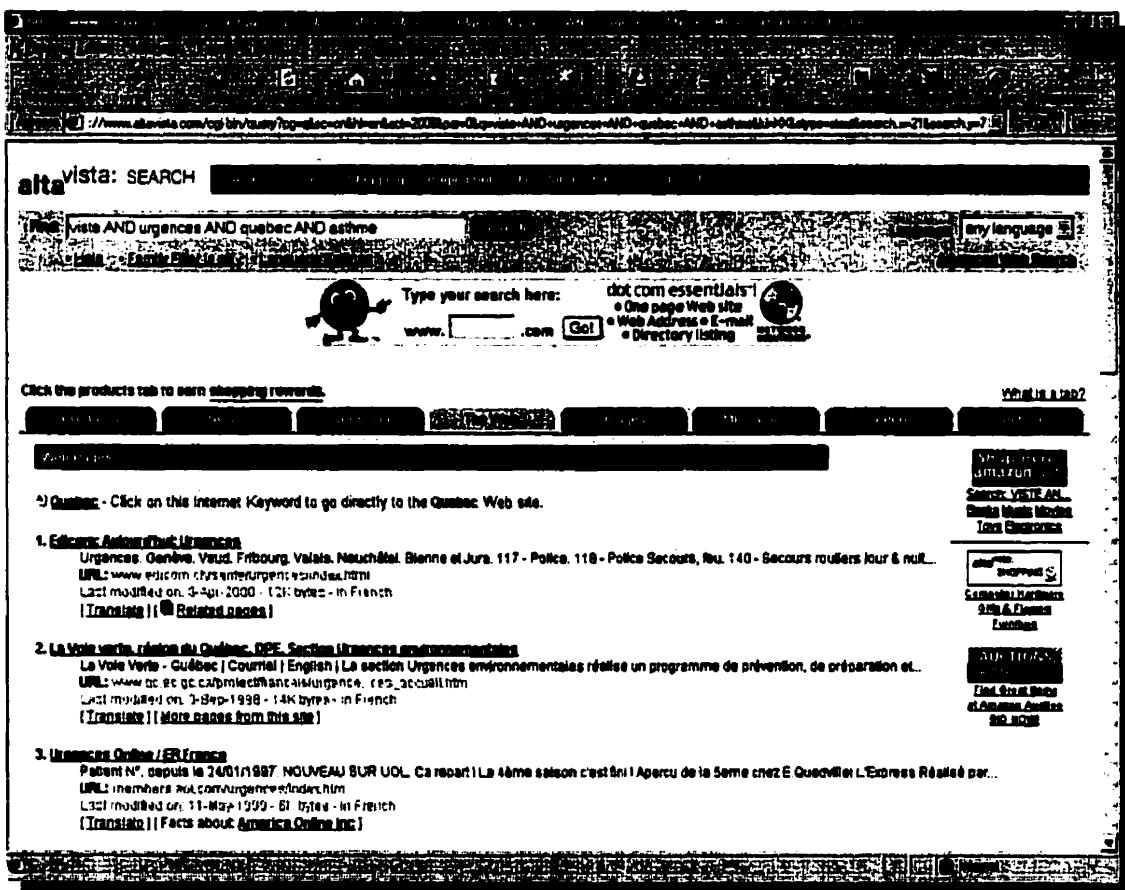


Figure numéro 4.2: Présentation des résultats (Altavista: Recherche avancée)

Exemple 2:

La limite de la technique basées sur les mots-clés au niveau de la précision du résultat pour l'utilisateur reste réelle, avec un autre exemple de moteur de

recherche ou agent de recherche intitulé «Copernic» (<http://www.copernic.com>). En fait, ce système n'est qu'un métamoteur de recherche, c'est-à-dire qu'il prend en charge la requête de l'utilisateur et il l'envoie à plusieurs moteurs de recherche qui utilisent la technique basée sur les mots-clés, et ensuite ces moteurs lui rendent le résultat. *Copernic* récupère le résultat, élimine les redondances, et affiche les documents pertinents dans l'ordre des moteurs de recherche qui lui ont fourni le résultat. En fait, nous tombons sur le même problème qui est le problème du bruit, où il y a plusieurs documents qui n'ont aucun lien avec la requête de l'utilisateur (voir figure numéro 4.3). Ce dernier, après avoir récupéré le résultat, devra le filtrer manuellement, et bien sûr, le temps mis pour ce filtrage dépend directement du nombre de documents en résultat.

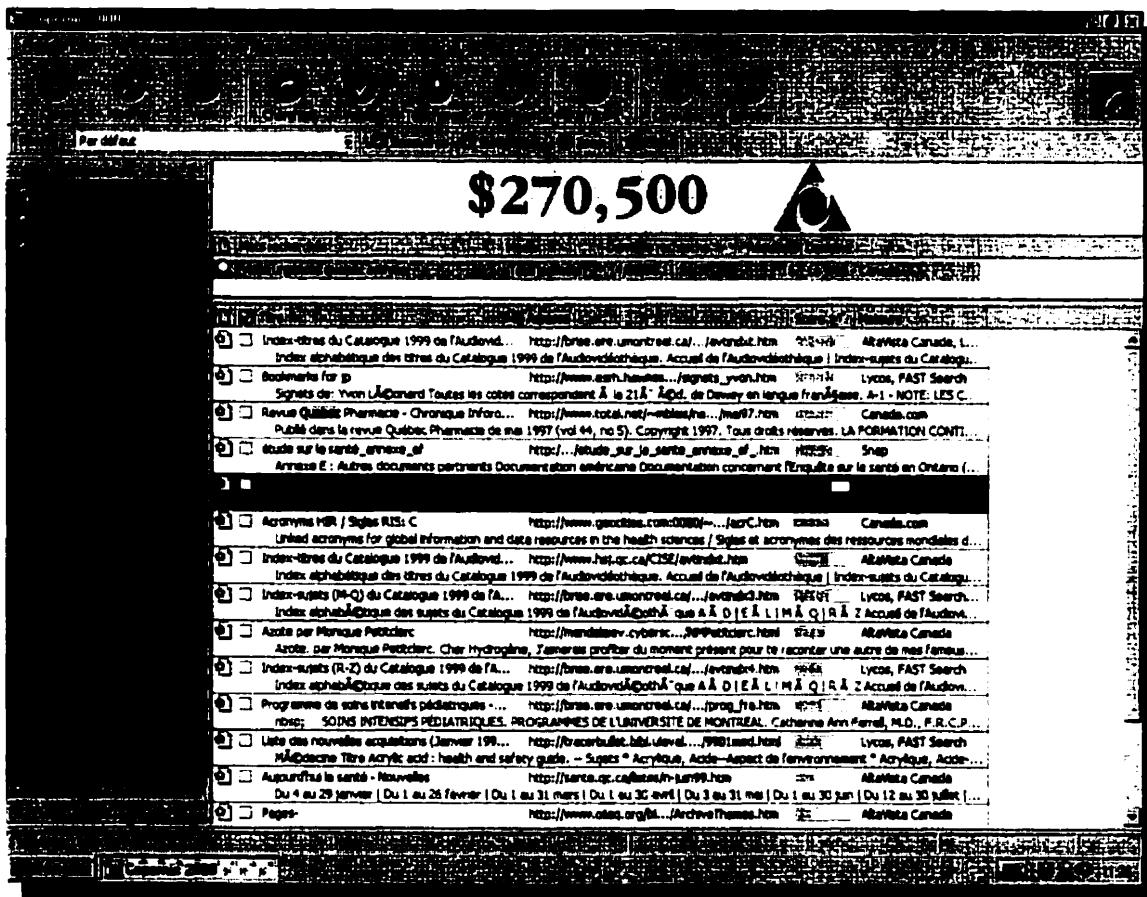


Figure numéro 4.3: Présentation des résultats (Copernic)

La technique basée sur les mots-clé n'est donc pas suffisante, et il est plus judicieux d'utiliser une autre technique plus intelligente, en se basant sur le contenu sémantique des documents. Pour utiliser cette technique, nous devons avoir une structure sémantique pour représenter les documents et pour les rechercher par la suite. Ceci peut être très intéressant pour filtrer les documents en résultat. Un système se basant sur la sémantique est plus efficace au niveau de l'indexation des documents, et aussi au niveau de la recherche. Un système permet d'indexer les documents, non pas par une liste de mots-clés séparés et indépendants, mais par une liste de mots ou concepts reliés entre eux par des relations appelées «*relations sémantiques*». La recherche de ces documents va se baser sur le même principe.

Dans la section suivante, nous allons présenter cette technique purement sémantique.

4- 3 – La technique purement sémantique:

4- 3 – 1 - Présentation de la technique:

Étant donné l'inconvénient de la technique présentée dans la section précédente, et dans le but de présenter à l'utilisateur un système plus efficace que ceux qui existent actuellement en terme de minimisation du bruit dans le résultat d'une requête, la technique basée sur les mots-clés a été mise de côté. La raison est que cette technique n'introduit aucun aspect sémantique ni au cours de l'indexation, ni au cours de la recherche. Pour introduire une sémantique entre les mots qui indexent les documents (indexation) ou les mots qui forment la requête (recherche), nous devons représenter ces documents ou ces requêtes en utilisant une structure et un langage bien déterminé. Nous allons utiliser les langages de représentation des connaissances et les structures sur lesquelles se basent ces langages.

Avant de présenter comment nous pouvons introduire cette structure dans notre système pour ajouter une couche sémantique dans notre processus d'indexation et de

recherche, la représentation des connaissances et les structures des graphes conceptuels et des réseaux sémantiques vont être présentés.

4- 3 – 2 - La représentation des connaissances:

Le problème de la représentation des connaissances consiste à trouver une correspondance entre un monde extérieur et un système symbolique.

Les connaissances que les ordinateurs manipulent habituellement sont d'ordre numérique. Les connaissances symboliques (i.e. qui utilisent des symboles) sont alors stockées dans des fichiers texte [Proux 97]. Par exemple, la phrase «Robert est allé à Paris» peut être représentée comme une simple chaîne de caractères ou d'une manière plus structurée :

* Chaîne de caractères : A stocker dans des fichiers textes par exemple. Un problème se présente lorsque nous cherchons des informations pour répondre à une question, comme par exemple «qui est allé à Paris?».

* Représentation incluant des éléments de signification de la phrase :

ACTION : Aller

AGENT : Robert

SOURCE : ?

DESTINATION : Paris

TEMPS : Passé

MOYEN : ?

Nous remarquons que cette représentation est beaucoup plus «appropriée» que la précédente pour répondre à des questions sur les faits enregistrés.

De même qu'il n'y a pas de langue universelle de programmation, il n'existe pas de formalisme «idéal» pour représenter les connaissances. Nous présentons deux formalismes de représentation de connaissances à savoir les «réseaux sémantiques» et «graphes conceptuels».

4- 3 – 2 - 1 - Les réseaux sémantiques:

Ce type de représentation a été à l'origine élaboré dans le domaine de la psychologie. Il s'agissait de rendre compte de la façon dont les êtres humains classent et mémorisent les concepts. L'idée a très vite été reprise en intelligence artificielle, l'objectif étant alors de symboliser les relations existant entre les différents concepts. Formellement, les réseaux sémantiques sont représentés par des graphes orientés, les nœuds symbolisant basiquement les concepts, et les arcs les relations existant entre ces concepts [Proux 97].

De façon plus précise, les nœuds peuvent représenter à la fois les concepts (exemple : oiseau, humain,...) mais aussi des actions (exemple : manger, dormir,...) ou des situations. De la même façon les arcs peuvent symboliser plusieurs types de relations. Nous pouvons noter les relations de hiérarchisation (représentées par les nœuds «est un» comme par exemple «un rossignol 'est un' oiseau»), de particularisation (qui sont l'inverse des liens de hiérarchisation : un oiseau «peut être un» rossignol), d'équivalence (une voiture «est équivalente» à une automobile), de contraste psychologique (beau-laid...) ou matériel (lumineux-obscur), de partie (un doigt «est une partie» de la main), de succession (mardi «succède à » lundi), de position (à droite, en haut de....), etc (voir [figure numéro 4.5](#)) [Sowa 84][Trigano 94][Proux 97].

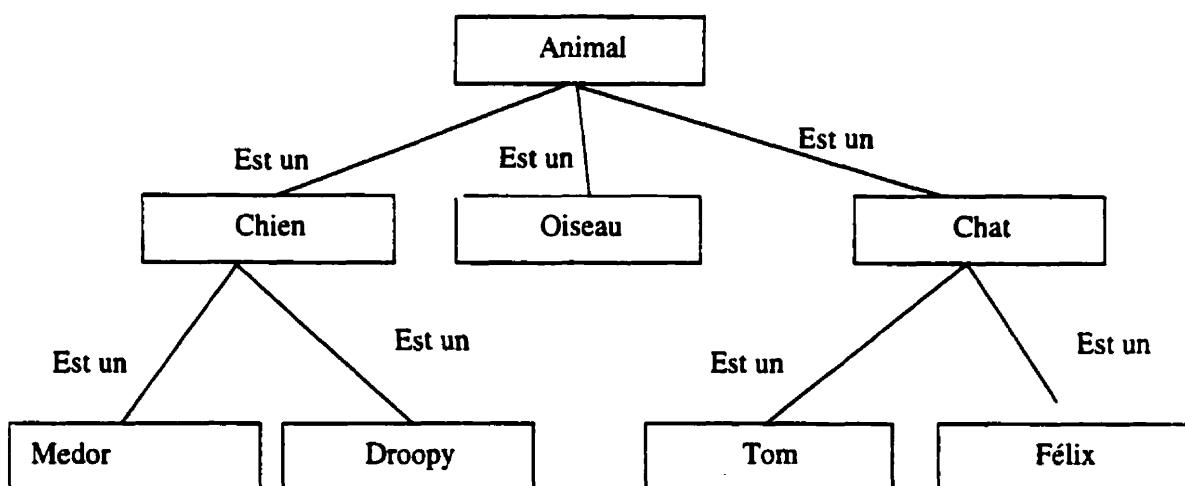


Figure numéro 4.5: Exemple de réseau sémantique simple

Le problème de ce type de représentation, et qui d'ailleurs est commun à un grand nombre de systèmes, concerne la polysémie. Un même mot peut correspondre, suivant ses emplois, à plusieurs nœuds du réseau. Il faut alors être capable de détecter, en fonction de la phrase à quelle unité conceptuelle nous avons affaire.

Le formalisme de réseaux sémantiques n'est pas le seul dans le monde de représentation des connaissances; il existe d'autres formalismes. Dans la section suivante, nous présentons un autre formalisme important de représentation des connaissances qui est le formalisme des graphes conceptuels.

4- 3 – 2 – 2 - Les graphes conceptuels:

Un «*graphe conceptuel*» est un système de la logique dont le but est de représenter le sens des mots ou des phrases dans une forme logiquement précise, qui peut être facilement compréhensible par l'être humain et aisément interprétable par l'ordinateur. Les graphes conceptuels peuvent servir comme un langage intermédiaire pour passer des formalismes orientés vers l'ordinateur à des langages naturels orientés vers les êtres-humains et vice versa. Par leur représentation graphique simple, ils peuvent servir comme une conception formelle mais compréhensible des langages naturels [Sowa 84].

Un graphe conceptuel est un diagramme qui représente la sémantique d'une phrase. Les éléments conceptuels sont *les concepts* (représentés par des rectangles) et *les relations conceptuelles* (représentées par des cercles).

Dans la figure numéro 4.6, nous avons un graphe conceptuel qui représente la phrase suivante : «*David est allé à Boston en bus*». A partir de cette phrase, nous avons quatre concepts qui sont [PERSONNE : David] [CITE : Boston] qui représentent deux instances des concepts PERSONNE et CITE; [ALLER] qui représente une instance non spécifiée du concept ALLER et [BUS] qui représente une instance non spécifiée du concept BUS. Les relations conceptuelles de ce graphe sont AGENT qui veut dire que «David» est l'agent de ALLER, c'est à dire que c'est lui qui a fait l'action d'aller, la relation DEST qui veut dire que «Boston» est la

destination, et INSTR qui représente le fait que le «Bus» est l'instrument de voyage [Sowa 84].

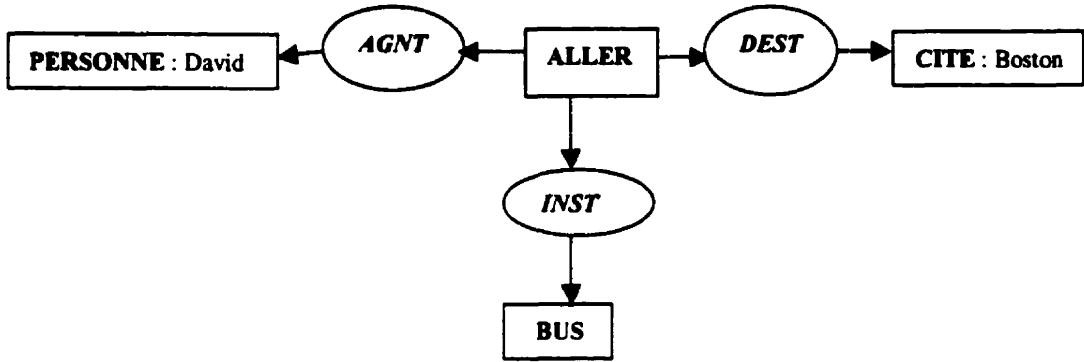


Figure numéro 4.6: Le graphe conceptuel représentant la phrase : «*David est allé à Boston en bus*»

Les graphes conceptuels sont des structures logiques : nous pouvons les transformer directement vers d'autres représentations logiques, telles que le calcul des prédicats. Dans la logique des prédicats, les concepts PERSONNE, ALLER, CITE et BUS par les fonction PERSONNE(x), ALLER (x), CITE (x), BUS (x). Les relations conceptuelles sont transformées en des prédicats avec un seul argument pour chaque flèche du graphe conceptuel. Les relations conceptuelles AGENT, DEST et INSTR deviennent AGENT (x , y), DEST (x , y), INSR (x , y). Le premier argument représente la flèche entrante, et le deuxième est pour la flèche sortante dans le diagramme du graphe conceptuel [Sowa 84]. Le graphe conceptuel de la figure numéro 4.6 peut être présenté par la formule suivante dans le calcul des prédicats du premier ordre :

$$\begin{aligned}
 & (\exists x)(\exists y)(\text{PERSONNE}(\text{David}) \wedge \text{ALLER}(x) \wedge \text{CITE}(\text{Boston}) \wedge \text{BUS}(x) \\
 & \quad \wedge \text{AGENT}(x, \text{David}) \wedge \text{INSTR}(x, y) \wedge \text{DEST}(x, \text{Boston}))
 \end{aligned}$$

Cette formule peut être lue comme suit : «Il existe un x et un y où David est une PERSONNE, x est une instance de ALLER, Boston est une CITE, y est un BUS,

l'agent AGENT de x est David, l'instrument INSTR de x est y, et la destination DEST de x est Boston».

A coté de ces représentations graphique et logique, nous pouvons aussi représenter les graphes conceptuels sous forme linéaire, car sous leur forme graphique nous ne pouvons pas les schématiser facilement. De plus, ils occupent beaucoup d'espace et nous ne pouvons pas les manipuler par la machine. Dans la représentation linéaire, nous utilisons des crochets pour présenter des éléments conceptuels et des parenthèses pour les relations conceptuelles [Sowa 84]. Ainsi, nous pouvons représenter le graphe conceptuel de la phrase «David est allé à Boston» par la ligne suivante :

[PERSONNE : David] ← (AGENT) → [ALLER] → (DEST) → [CITE : Boston].

Si le graphe a des branches complexes, nous ne pouvons pas le représenter sur une seule ligne, mais sur plusieurs. Dans le graphe conceptuel de la figure numéro 4.6, la relation conceptuelle ALLER est attachée à trois concepts ce qui rend la représentation du graphe non linéaire. Ainsi nous pouvons représenter la phrase «David est allé à Boston par bus» d'une manière non linéaire comme suit :

[ALLER] -

(AGENT) → [PERSONNE : David]

(DEST) → [CITE : Boston]

(INSTR) → [BUS].

4- 3 – 2 – 2 – 1 - Les concepts et les référents:

Les concepts représentent le sens des mots. Comme les mots, ils se réfèrent aux entités, aux actions, aux propriétés ou aux événements dans le monde. Pour distinguer les types des concepts, des référents, et des individus spécifiques, le rectangle qui représente le concept dans le graphe conceptuel est divisé en deux parties : Un champ «type» à gauche et un champ «référent» à droite. Par exemple, le concept

[PERSONNE : David] est un concept individuel avec un type PERSONNE et un référent «David». Dans l'exemple précédent, les concepts [BUS] et [ALLER] sont appelés des «concepts génériques» parce qu'ils n'ont pas identifié un référent, ils spécifient seulement le champ type du concept [Sowa 84].

Le champ référent peut être représenté par des symboles au lieu des noms pour représenter des quantificateurs ou des noms pluriels ou des ensembles.

Exemples : * Quantificateur «Tous les chiens mangent de la viande»

[CHIEN : \forall] \leftarrow (AGENT) \rightarrow [MANGER] \leftarrow (PATIENT) \rightarrow [VIANDE]

* Ensemble «David et Caroline mangent»

[PERSONNE : {David, Caroline}] \leftarrow (AGENT) \rightarrow [MANGER]

Il existe d'autres symboles de référents pour représenter d'autres types de concepts comme l'interrogation, les autres quantificateurs (le quantificateur existentiel par exemple), les littéraux.

4- 3 – 2 – 2 – 2 – La grille de type de concepts:

Les types des concepts sont organisés en une grille de type en se basant sur leur niveau de généralité. Dans la figure numéro 4.7 nous montrons un exemple de hiérarchie avec un type universel (T) tout en haut de la hiérarchie. Au dessous de T , il existe les sous-types les plus généraux ENTITY et SITUATION. Sous le sous-type ENTITY nous trouvons les objets physiques, les abstractions et les types de données. Sous le sous-type SITUATION nous trouvons les processus, les états, les événements et les actions. En descendant dans la hiérarchie, les types deviennent de plus en plus spécialisés [Martin 96][MartinEklund 97].

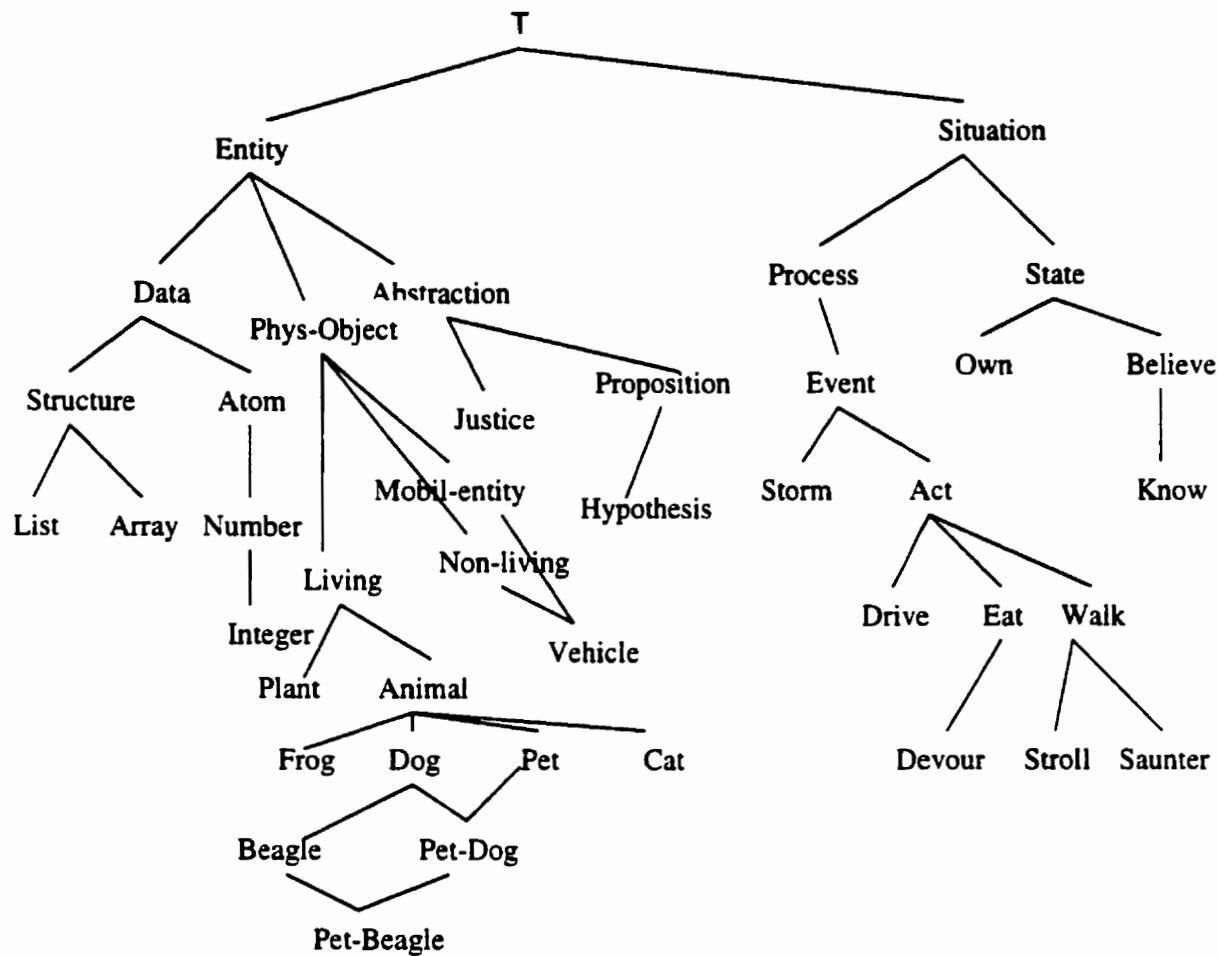


Figure numéro 4.7: Un exemple d'une ontologie de concepts

Cette grille de type ne présente pas un «arbre», car certains types ont plus qu'un super-type immédiat. Par exemple le type PET-DOG est un sous-type de DOG et de PET. Cette grille de type représente un graphe acyclique. En introduisant la notion de l'ordre dans la grille, la théorie des graphes conceptuels définit cette grille de type comme un treillis. Pour convertir ce graphe acyclique en un treillis, il faut introduire un nœud au plus bas niveau \perp appelé «type absurde». Il n'existe aucune chose qui peut être une instance de \perp , mais ce type est utilisé pour compléter la représentation sous forme de treillis. Chaque type est un sous-type de **T** et un supertype de \perp . En se basant sur cette structure, nous pouvons appliquer les opérations du treillis sur nos types telles que le supertype commun minimal ou le sous-type commun maximal, la conformité, la dénotation...[Sowa 84]

4- 3 – 2 – 2 – 3 - Les relations conceptuelles:

Comme son nom l'indique, une relation conceptuelle représente une certaine relation entre les concepts. Quand une relation conceptuelle relie deux concepts, elle montre qu'il existe un certain lien entre leurs référents. Dans la représentation formelle des graphes conceptuels, nous trouvons trois types de relations conceptuelles : Les relations conceptuelles « primitives » (Primitives), les relations conceptuelles « ensemble de départ » (starter set) et les relations conceptuelles « définies » (Defined) [Sowa 84].

- ***Les relations conceptuelles primitives : Primitives*** (voir figure numéro 4.8): Il n'existe une seule relation primitive qui est la relation LINK. Toutes les autres relations peuvent être définies à partir d'elle.
- ***Les relations conceptuelles ensemble de départ : Starter set*** (voir figure numéro 4.9): Ce type de relations conceptuelles est très utilisé. Chaque relation dans cet ensemble de départ a une correspondance dans les langages naturels et vice versa.
- ***Les relations conceptuelles définies : Defined*** (voir figure numéro 4.10): Chaque relation utilisée dans la logique, les bases de données relationnelles ou les diagrammes entité-relation peut être définie comme étant une relation conceptuelle de ce type. Les relations définies peuvent ne pas avoir automatiquement une correspondance dans les langages naturels.

Dans les trois diagrammes qui suivent, nous représentons la phrase «Le chat chasse la souris» de trois manières, en nous basant à chaque fois sur un type de relation.

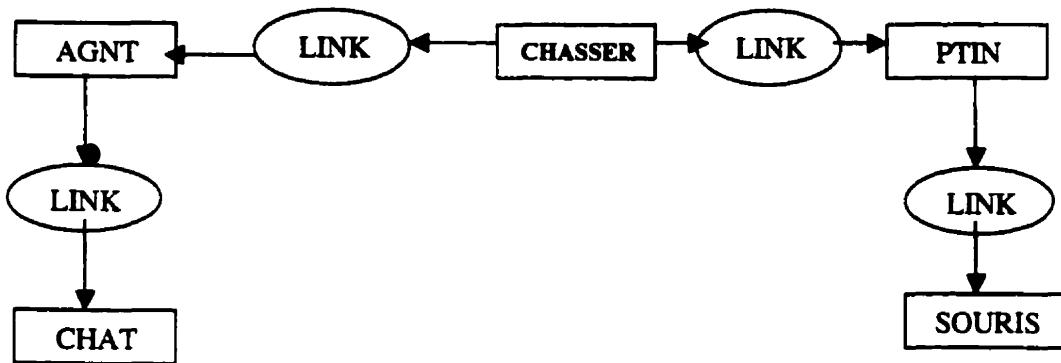


Figure numéro 4.8: Les relations conceptuelles primitives



Figure numéro 4.9: Les relations conceptuelles ensemble de départ :
Starter set

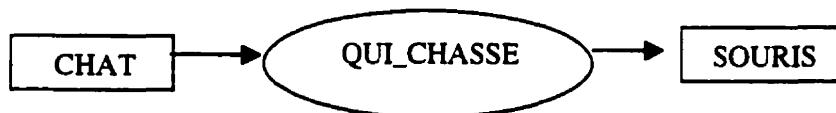


Figure numéro 4.10: Les relations conceptuelles définies : Defined

Les relations conceptuelles de l'ensemble de départ ont des noms standards dans le formalisme des graphes conceptuels. A titre d'exemple, nous présentons quelques exemples de relations conceptuelles de type «Starter set» :

- *Les relations thématiques* : Elles montrent l'action ou l'état. Elles sont exprimées par un verbe et elles sont reliées aux entités qui sont exprimées par des sujets, objets ou autres compléments. Voici quelques relations possibles :

AGNT : agent

PTNT : Patient

STAT : Etat

EXPR : Expérience

RCPT : Récipient

INST : Instrument

DEST : Destination

RSLT : résultat

- *Les relations spatiales* : Elles incluent les localisations simples LOC, ainsi que les localisations spécifiques IN, ON, ABOVE...etc
- *Les relations mathématiques* : Elles incluent la mesure MEAS, supérieure à >, inférieure à <, égal =, non égal /=, moyenne AVG, les arguments d'une fonction ARG1 et ARG2...etc
- *Les relations des attributs* : Elles incluent la relation générale ATTR ainsi que d'autres relations spécifiques comme CHRC pour «caractéristique» et PART pour «a une partie»...etc
- *Les métarélations* : Elles décrivent comment les types des concepts et les relations sont reliés les uns aux autres dans le graphe conceptuel. Elles incluent le type KIND, le sous-type SUBT, la description DSCR, la représentation REPR.

L'ensemble de départ des relations conceptuelles doit être fixé dès le début, et chaque nouvelle relation doit être définie.

Le vocabulaire des concepts comme PERSONNE, BUS ...et des relations comme AGENT, ALLER... dérive d'une ontologie de types prédéfinis. Une «ontologie» n'est qu'un dictionnaire des termes qui peuvent être utilisés comme vocabulaire pour modéliser la connaissance. Une telle ontologie peut être organisée sous forme d'un treillis [Sowa 84].

Il est important de signaler que la structure idéale pour représenter les connaissances est le langage naturel, mais la complexité de manipulation automatique de ce langage nous a mené à manipuler les connaissances en utilisant des structures proches du langage naturel comme la structure des graphes conceptuels.

4- 3 – 3 - L'utilisation des graphes conceptuels dans le système:

Le système de recherche et d'indexation va utiliser la structure des graphes conceptuels dans les deux processus d'indexation et de recherche.

L'indexation consiste en l'analyse du document. Cette analyse n'est pas faite pour conserver la totalité de l'information contenue dans le document, mais seulement certaines parties de l'information pertinente qui permettra à un utilisateur de retrouver le document. Pour pouvoir réussir la phase d'indexation, il faut bien décrire le document à indexer.

Intuitivement, lorsqu'une personne lit un texte, regarde une image ou une séquence vidéo, ou bien écoute une séquence audio, elle ne se souvient habituellement que des objets, relations entre ces objets, actions et évènements importants. Pour assurer une indexation efficace dans le but de retrouver le document par la suite, il ne faut pas le structurer sous forme de mots-clé séparés. Il semble plus adéquat de se référer à des éléments pertinents dans les documents tels que des objets, actions ou évènements. C'est ce que nous allons tenter avec la technique proposée dans cette section. En fait, dans le processus d'indexation que nous envisageons, l'utilisateur sélectionnerait le document, puis commencerait à ajouter les objets pertinents, les relations conceptuelles entre ces objets, ainsi que les actions ou les évènements liés à ce document. La façon idéale pour décrire ce document de cette manière consisterait à utiliser des phrases en langage naturel. Mais, pour éviter la complexité du traitement du langage naturel, il faut trouver une interface simple d'utilisation et se rapprochant suffisamment du langage naturel. L'interface que nous proposons contient des champs qui doivent être remplis par l'utilisateur pour introduire les sujets des actions, les objets des actions qui décrivent une phrase, etc. Les objets introduits par l'utilisateur chargé de l'indexation sont traités comme des concepts de graphes conceptuels, tandis que les relations entre ces objets et que les actions sont manipulés comme des relations conceptuelles. Le document va être indexé à l'aide d'un ou plusieurs graphes conceptuels qui vont être sauvegardés dans un fichier joint au fichier du document.

Dans la technique proposée, nous nous appuyons sur la capacité qu'a l'usager de faire une analyse grammaticale des phrases servant à indexer les documents. Ceci, afin de déterminer le sujet, le verbe, le complément d'objet,...etc. Nous avons choisi cette approche comme une première approximation de détermination de graphes conceptuels.

Voici un exemple d'indexation d'un document selon cette technique:

Exemple: Indexation d'un bout de texte en utilisant la notion des graphes conceptuels.

Soit le texte suivant : «Une étude montre que 50 personnes mâles âgées de moins de 40 ans et qui ont le cancer des poumons travaillent dans l'industrie chimique dans la région de Gaspésie.»

Les mots-clés que nous pouvons extraire de ce texte sont les suivants: Nombre, mâle, poumon, cancer, chimique, industrie, Gaspésie, région, âge, 40, 50 ...etc

Le graphe conceptuel qui représente ce texte est représenté dans la figure numéro 4.11:

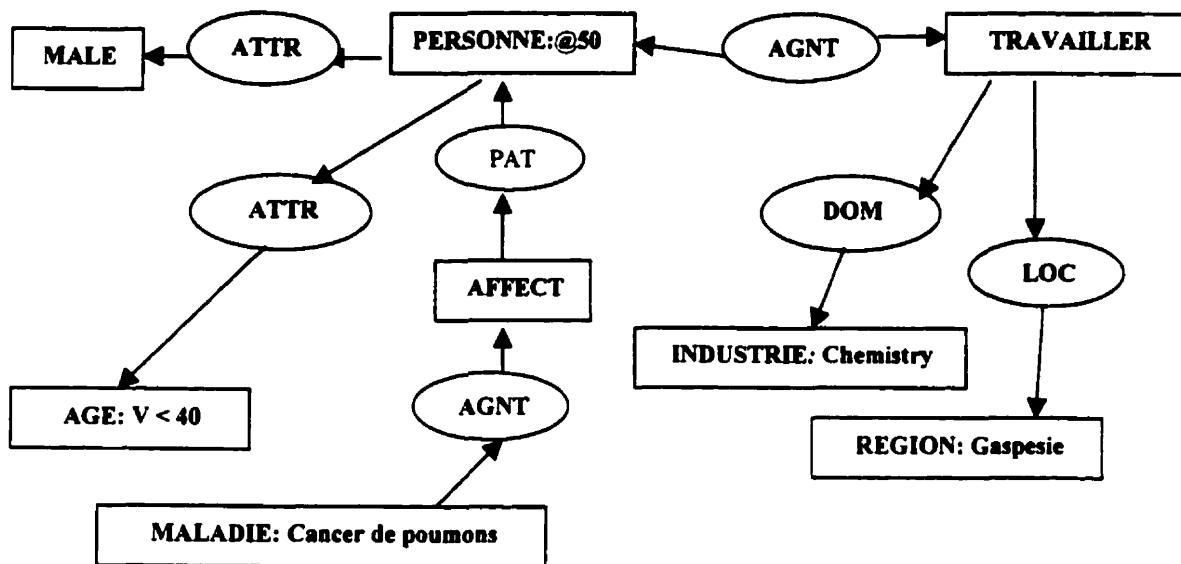


Figure numéro 4.11: Graphe conceptuel représentant le texte de l'exemple précédent

Dans le processus d'indexation, l'indexeur va introduire les objets, les actions, et enfin, des relations pour construire sa structure d'indexation qui est une structure similaire au graphe conceptuel. Les objets vont être traités comme des concepts, les actions et les relations comme des relations conceptuelles dans la structure des graphes conceptuels. Les objets ou les concepts, les actions et les relations conceptuelles sont sélectionnés à partir des ontologies correspondantes. En sélectionnant les objets, les actions ou les relations, l'utilisateur peut introduire sa propre entité (objet, action ou relation), cet ajout est bien sûr contrôlé par l'administrateur du système pour assurer l'intégrité des ontologies en question.

En pratique, nous avons développé des maquettes d'interface pour concrétiser la démarche proposée et pour évaluer les réactions de nos usagers. Voici les principaux menus et écrans proposés:

a) Indexation des documents :

Pour indexer un document multimédia, l'utilisateur lance le module d'indexation des documents et l'écran de la figure numéro 4.12 s'affichera:



Figure numéro 4.12 : Fenêtre d'indexation des documents

Pour indexer un document, il faut ajouter des objets, des actions et des relations entre ces objets pour construire l'index de ce document. La gestion des objets, des relations et des actions est présentée dans les fenêtres qui suivent.



Figure numéro 4.13: Menu de la gestion des objets

A travers le menu de cette forme l'utilisateur peut ajouter, modifier ou supprimer des objets de la structure qui permet d'indexer le document en question (voir [figure numéro 4.13](#)).



Figure numéro 4.14: Menu de la gestion des actions

A travers le menu de cette forme, l'utilisateur peut ajouter, modifier ou supprimer des actions de la structure qui permet d'indexer le document en question (voir [figure numéro 4.14](#)).



Figure numéro 4.15: Menu de la gestion des relations

A travers le menu de cette forme, l'utilisateur peut ajouter, modifier ou supprimer des relations de la structure qui permet d'indexer le document en question (voir [figure numéro 4.15](#)).

- Ajout d'un objet:

Si l'utilisateur désire ajouter un objet à la structure d'indexation du document, il aura en résultat la fenêtre ci-dessous : Par exemple, si l'utilisateur veut ajouter un objet personne qui s'appelle John, il choisira dans la structure de l'ontologie le concept personne et entrera le nom de l'objet en question, et pèsera sur le bouton '>>' pour l'ajouter à la liste des objets courants de la structure d'index du document. L'utilisateur peut ajouter autant d'objets qu'il veut, et peut finir cette opération en cliquant sur le bouton 'Fin'. La liste des objets courants contient la liste des objets qui sont ajoutés par l'utilisateur : Un objet a la forme suivante : 'Personne-> Homme : John' (voir [figure numéro 4.16](#)).

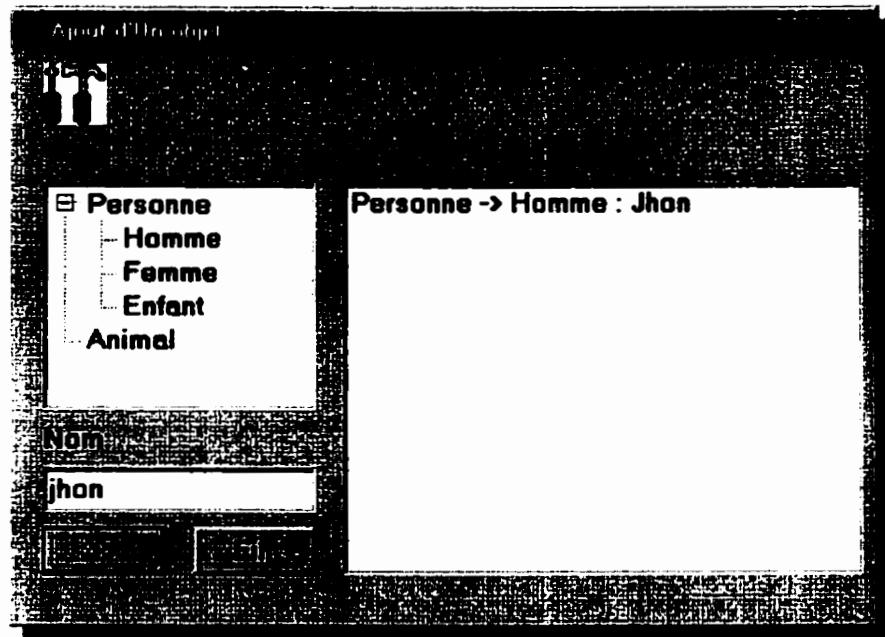


Figure numéro 4.16: Interface d'ajout d'un objet

- Ajout d'une action:

Pour ajouter une action, l'utilisateur doit choisir l'opération dans le menu déroulant présenté précédemment et la fenêtre suivante s'affichera.

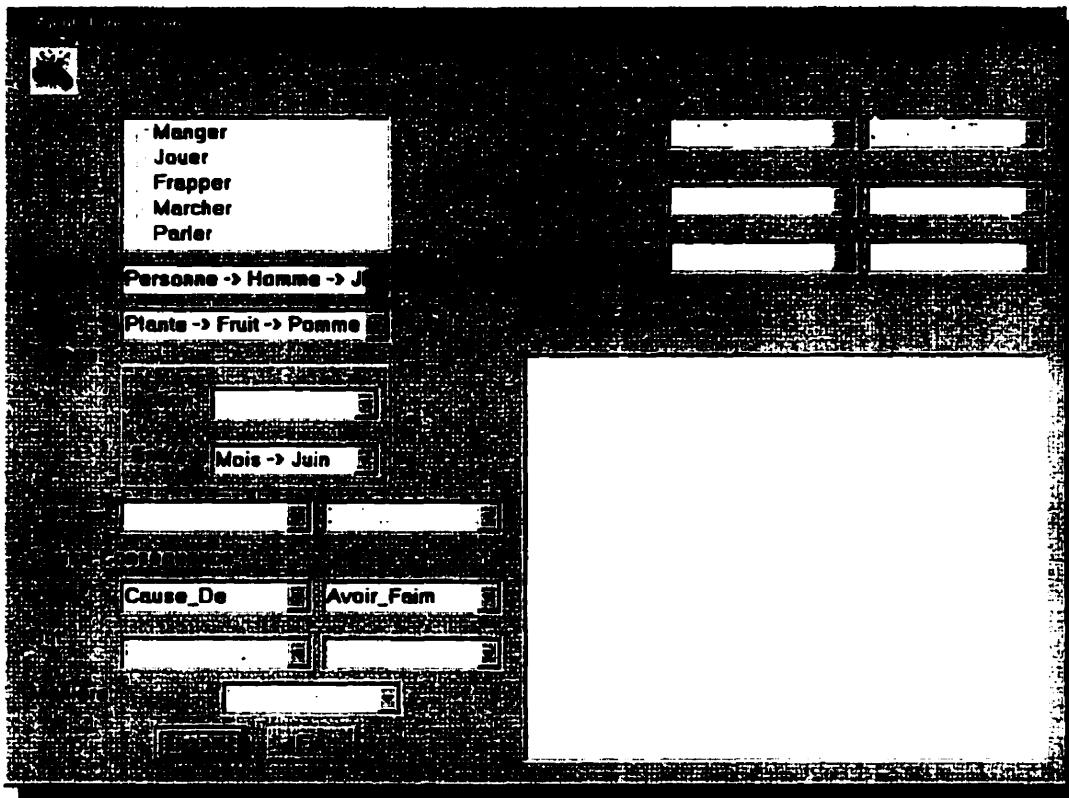


Figure numéro 4.17: Interface d'ajout d'une action

Par exemple, pour ajouter l'action suivante : «John mange une pomme car il a faim». Dans ce cas l'utilisateur va ajouter les objets essentiels pour cette action, comme l'objet Personne avec la valeur John, l'objet Fruit avec la valeur Pomme... Puis, dans cette fenêtre, il va choisir l'action Manger dans l'ontologie des actions, et il commencera à remplir les autres champs paramètres de l'action (voir figure numéro 4.17).

Les différents champs de l'action peuvent être présentés comme suit :

- Type de l'action : Ce champ représente le type de l'action en question. Les différents types sont définis dans l'ontologie des actions. Comme exemple, on a le type de l'action Manger. Pour une action, on peut associer un nom ou une instance. Par exemple dans le cas d'un document où on a deux actions

identiques, on peut les différencier par leurs instances, pour un type Manger, on peut associer les instances Manger1 et Manger2.

- **Sujet de l'action** : Pour une action, on peut avoir un sujet, qui est l'acteur de l'action. Ce sujet n'est qu'un objet qui a fait l'action.
- **Objet de l'action** : Pour une action, on peut avoir un objet, qui est l'objet sur lequel l'action est faite.
- **Temps de l'action** : Une action peut avoir un temps qui peut être un instant, une durée ou un temps par rapport une autre action :
Par rapport à un instant ou une période : Avant , Pendant, Après....
Par rapport à une autre action : Avant que, Pendant que, après que, jusqu'à ce que...
- **Lieu de l'action** : Une action peut avoir un lieu, le lieu n'est qu'un objet associé à l'action : Devant, Sur, Au dessus, Au dessous...
- **Cause-Conséquence de l'action** : Une action peut avoir une cause ou une conséquence qui est une autre action : Cause de , Conséquence de...
- **But de l'action** : Une action peut avoir un but : qui est une autre action : Dans le but de, Afin de...
- **Manière de l'action** : Une action peut avoir une manière : Lentement, Doucement, Fort...
- **Moyen de l'action** : Une action peut avoir un moyen, ce moyen n'est qu'un objet : A l'aide de, Par, En utilisant...
- **Comparaison** : Pour comparer une action par rapport à une autre action: Plus fort que, Moins fort que, Autant que....

- Concession : Pour faire des concessions par rapport d'autres actions : quoique, Bien que, Malgré, Alors que, tandis que....

Remarque :

Les noms des champs présentés ci-dessus ont été choisis par nous-mêmes afin d'utiliser des notions familières à l'usager. Bien entendu, d'un point de vue linguistique ces termes ne sont pas assez précis.

Nous pouvons ajouter d'autres champs, comme le champ de condition : Si, Condition d'une action par rapport à une autre action.

L'utilisateur peut valider son action en appuyant sur le bouton '>>'. L'action en question sera ajoutée à la liste des actions courantes de la structure d'index du document. L'utilisateur peut ajouter autant d'actions qu'il veut, et peut finir cette opération en cliquant sur le bouton 'Fin'

- **Ajout d'une relation:**

Pour ajouter une relation conceptuelle entre deux objets, l'utilisateur doit choisir l'opération dans le menu déroulant présenté précédemment et la fenêtre ci-dessous s'affichera (voir figure numéro 4.18). En fait une relation conceptuelle ne peut être définie qu'entre deux objets. Alors l'utilisateur choisit dans la première liste déroulante le premier objet, puis il choisit le type de la relation et enfin dans la deuxième liste le deuxième objet. Les types des relations sont prédéfinis dans une ontologie de relations. Dans la liste déroulante qui permet à l'utilisateur d'ajouter le premier objet ou le deuxième, nous trouvons les objets déjà ajoutés par l'utilisateur dans la fenêtre d'ajout des objets. L'utilisateur a le choix de mettre un nouvel objet grâce à une entrée 'Autre...' dans la liste déroulante, ce qui provoque l'affichage de la fenêtre d'ajout des objets à la structure d'indexation du document. Les relations, une fois ajoutées en appuyant sur le bouton '>>', sont mises dans la liste des relations courantes (voir figure numéro 4.18). Ces relations ont la forme suivante :

Exemple d'une relation conceptuelle entre deux objets : «*Josef est le mari de Marie*».

L'utilisateur doit ajouter un objet «Personne -> Homme -> Josef» et un objet «Personne -> Femme -> Marie», et après il ajoute la relation «Est_le_mari_de». La relation s'affichera comme suit (voir figure numéro 4.18):

Personne->Homme : Josef * Est_le_mari_de * Personne->Femme-> : Marie

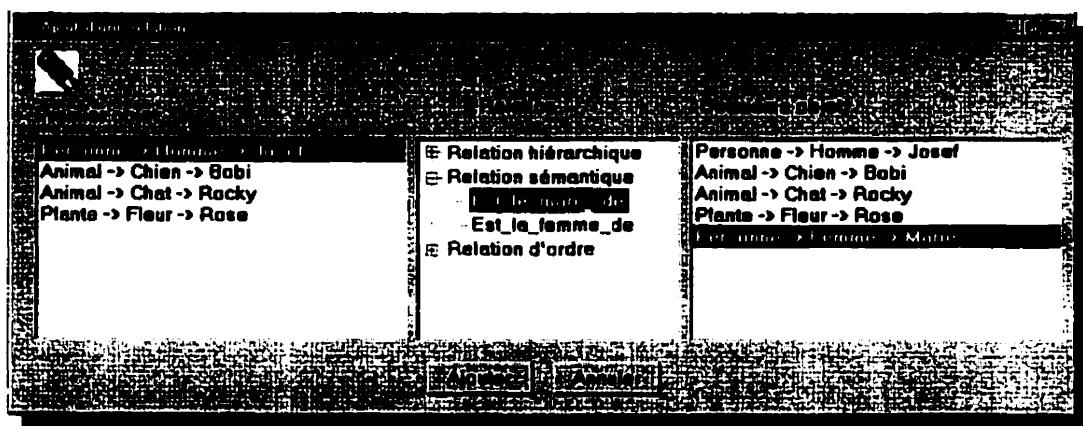


Figure numéro 4.18: Interface d'ajout d'une relation

b) Recherche des documents :

Dans le processus de recherche des documents ou simplement dans le processus du matching (ou appariement), nous adoptons le même principe. En fait, l'utilisateur introduit sa requête sous forme d'un ensemble d'objets, de relations entre ces objets et d'actions, grâce à une interface similaire à celle de l'indexation. La requête va être manipulée comme un ou plusieurs graphes conceptuels. Le processus du matching n'est qu'une correspondance du ou des graphes conceptuels de la requête et de ceux qui indexent le document.

Dans le processus de recherche, nous allons utiliser la même approche que celle utilisée dans l'indexation. En fait la requête de l'utilisateur va être posée sous forme d'une liste de champs à remplir par l'utilisateur et non simplement par une liste de mots-clé.

4- 3 – 4 – Les ontologies utilisées par le système:

Nous avons indiqué précédemment qu'une ontologie est nécessaire pour manipuler les concepts et les relations conceptuelles. L'idée serait de posséder une ontologie globale, par exemple celle de la langue française. Mais, vu la difficulté de la mise en œuvre de cette ontologie, il est préférable d'utiliser seulement un noyau de cette ontologie.

a) L'ontologie des concepts: (Voir figure numéro 4.19)

L'ontologie des concepts utilisée par le système est un sous-ensemble de l'ontologie de la langue française. Elle est structurée en une structure de treillis. A titre d'exemple, voici ci-dessous une portion de notre ontologie de test. La structure complète est en cours d'étude.

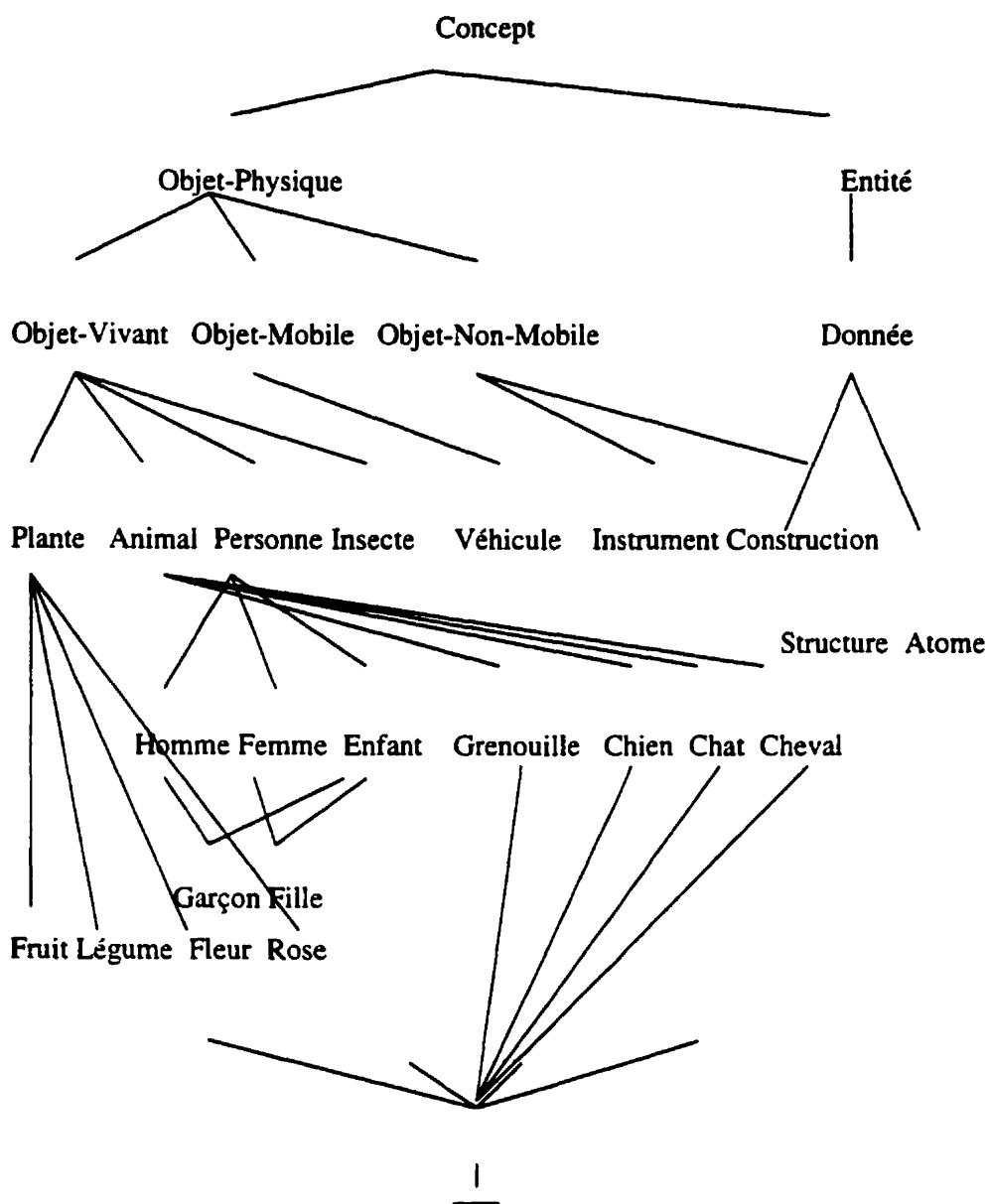


Figure numéro 4.19: L'ontologie des concepts

b) L'ontologie des actions: (Voir figure numéro 4.20)

L'ontologie des actions utilisée par le système dérive aussi de la langue française. Elle est structurée aussi en une structure de treillis. A titre d'exemple, voici ci-dessous une portion de cette ontologie. La structure complète est aussi en cours d'étude.

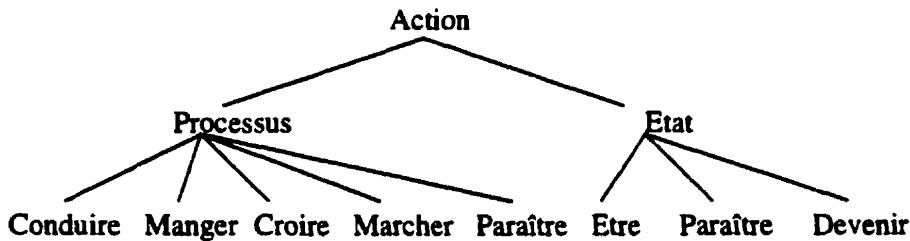


Figure numéro 4.20: L'ontologie des actions

c) L'ontologie des relations: (Voir figure numéro 4.21)

L'ontologie des relations conceptuelles utilisée par le système dérive aussi de la langue française. Elle est structurée en une structure d'arbre. Encore une fois, voici ci-dessous une portion de cette ontologie. La structure complète est aussi en construction.

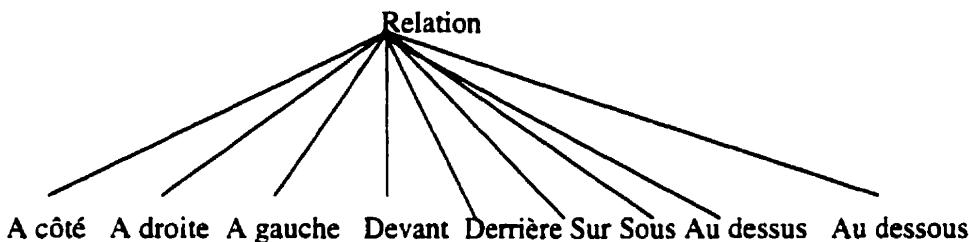


Figure numéro 4.21: L'ontologie des relations

4- 3 – 5 - Limites de la technique:

Cette technique est très intéressante pour le domaine d'indexation et de recherche des documents car elle est proche du langage naturel et aussi car elle est unique pour indexer ou chercher plusieurs types de documents. Mais le problème que nous avons rencontré se situe au niveau de la programmation de cette structure et aussi au niveau de son utilisation. En fait, d'après les différentes interfaces présentées précédemment, pour indexer un document, il faut introduire plusieurs objets, plusieurs relations entre ces objets, diverses actions. Nous pouvons prévoir que pour un document vidéo qui

comporte plusieurs scènes et objets, l'indexation sera pénible. De même pour la recherche, l'utilisateur doit introduire sa requête sous forme d'objets et relations entre les objets. Cette technique n'est pas la meilleure si nous voulons concevoir un système qui doit répondre rapidement et efficacement aux requêtes des utilisateurs.

4- 4 – La technique basée sur les expressions:

4- 4 – 1 – Motivations:

Les deux techniques présentées précédemment (par mots-clés et basée sur la création des graphes conceptuels) sont générales. Elles s'appliquent à n'importe quel type de documents (image, texte, vidéo, etc.) pour n'importe quel domaine. Cependant, de nombreuses limitations à ces modèles ont été mises en évidence, notamment en terme d'ergonomie des interfaces. Les deux types d'interfaces permettent de traiter des cas généraux sans prendre en compte la spécificité du domaine d'application. Ainsi, les interfaces peuvent être difficiles à utiliser pour certains domaines d'application. Par exemple, si un utilisateur emploie toujours le même type d'expressions propres à son domaine, alors il est fastidieux pour lui de ressaisir à chaque fois son expression pour rechercher ou indexer un document. C'est dans cette optique que nous avons essayé de trouver une autre technique et un autre type d'interface permettant de simplifier la tâche d'indexation et de recherche de documents.

4- 4 -2 - Présentation de la technique:

Si l'utilisateur est un professionnel dans un domaine particulier, il fait appel à un vocabulaire spécialisé qu'il utilise pour indexer et classer ses documents. Le problème qui apparaît assez vite est celui des expressions. En effet, le vocabulaire utilisé prend en compte les termes pour leur sens propre mais ne prend pas en compte le sens des expressions qui peuvent exister dans le domaine. Par exemple, l'expression «cul de sac» ne veut pas dire le «fond d'un sac» mais une «impasse», et si cette expression n'est pas prise en compte dans l'ontologie, alors des problèmes de pertinence entre la requête et le résultat de la recherche peuvent apparaître. Ainsi, les expressions du

domaine doivent être intégrées dans ce vocabulaire. Dans ce qui suit, nous présentons brièvement cette technique qui sera détaillée dans le cinquième chapitre de ce mémoire.

Voici une représentation schématique du principe d'indexation et de recherche par expressions. Les cercles représentent les mots et les liaisons les liens sémantiques entre les mots.

- Formation des expressions à partir des concepts (expressions unitaires et des autres expressions): (voir figure numéro 4.22)

Dans cette étape nous construisons l'ontologie des expressions de notre système. Les expressions sont construites au fur et à mesure à partir des concepts unitaires et des expressions existantes déjà dans l'ontologie des expressions (par exemple l'expressions «Cancer du sein» est construite à partir des expressions «Cancer» et «Sein»). C'est l'indexeur qui choisit la relation qui relient les expressions pères pour former la nouvelle expression.

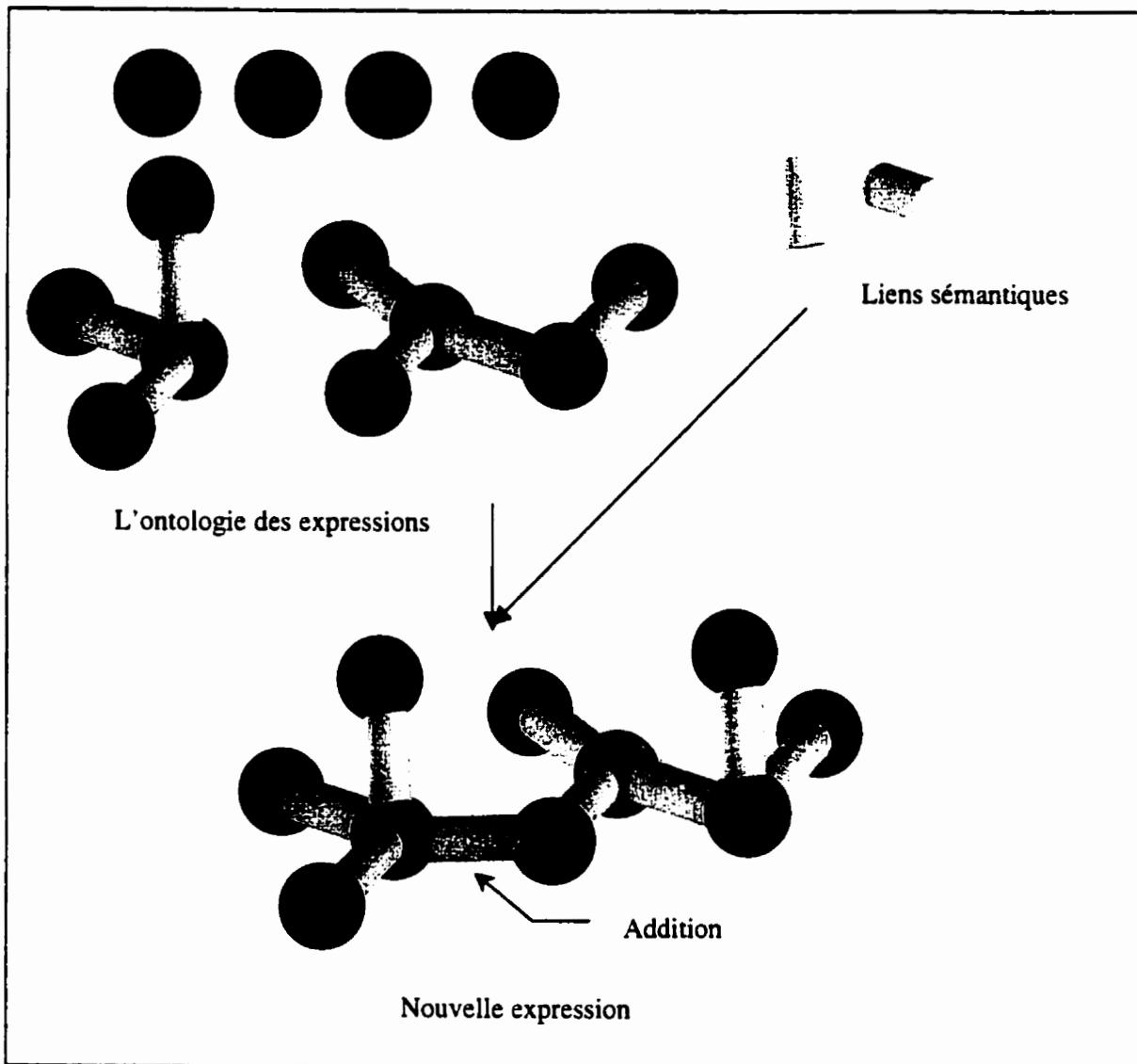


Figure numéro 4.22: Formation d'une expression composée à partir de concepts unitaires et des autres expressions

- L'indexation et la recherche par des expressions : (Voir [figure numéro 4.23](#))

Dans cette étape, nous indexons les documents en nous basant sur les expressions existant déjà dans l'ontologie. A chaque document, nous associons un ensemble des expressions (unitaires ou composées) qui décrivent le mieux le sujet traité par le document en question. C'est indexation de fait manuellement.

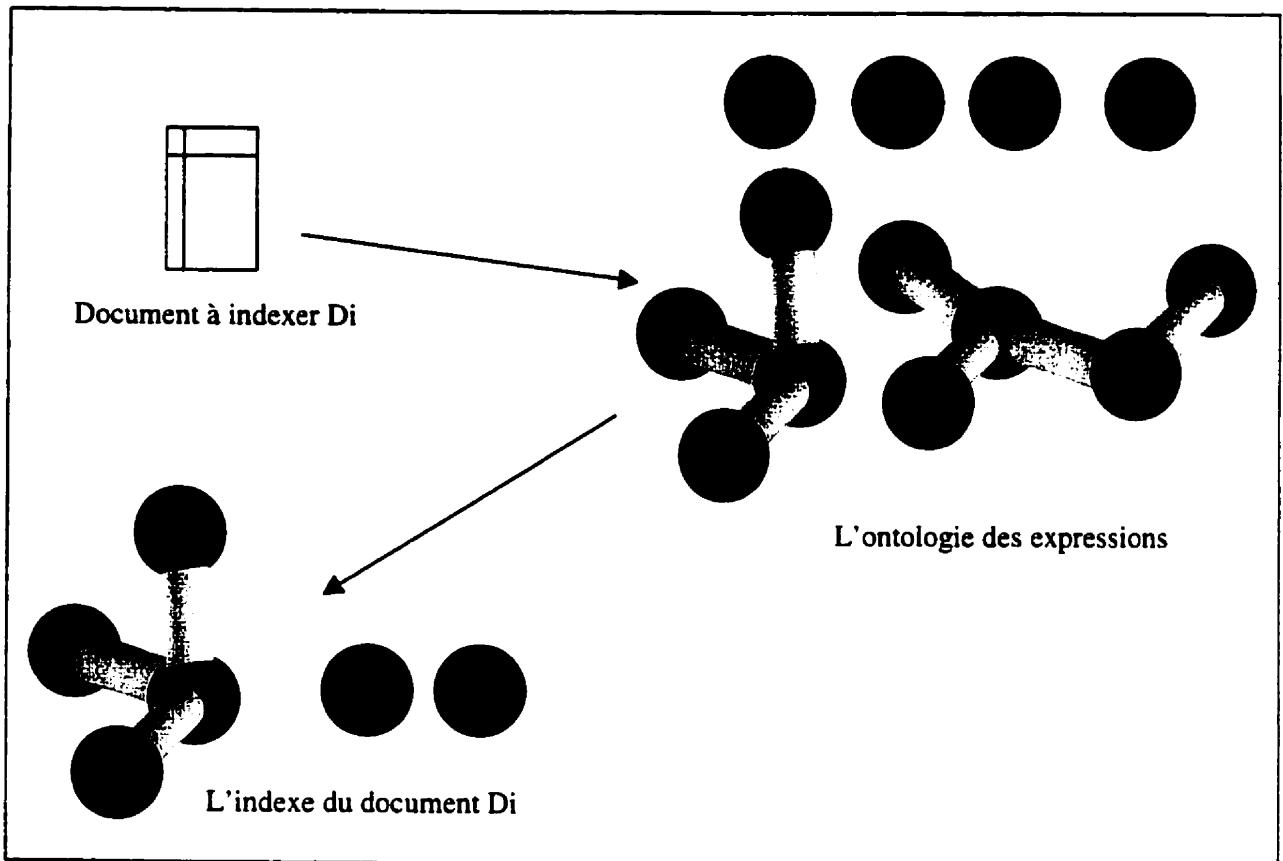


Figure numéro 4.23: Indexation basée sur les expressions

- La recherche par des expressions : (Voir figure numéro 4.24)

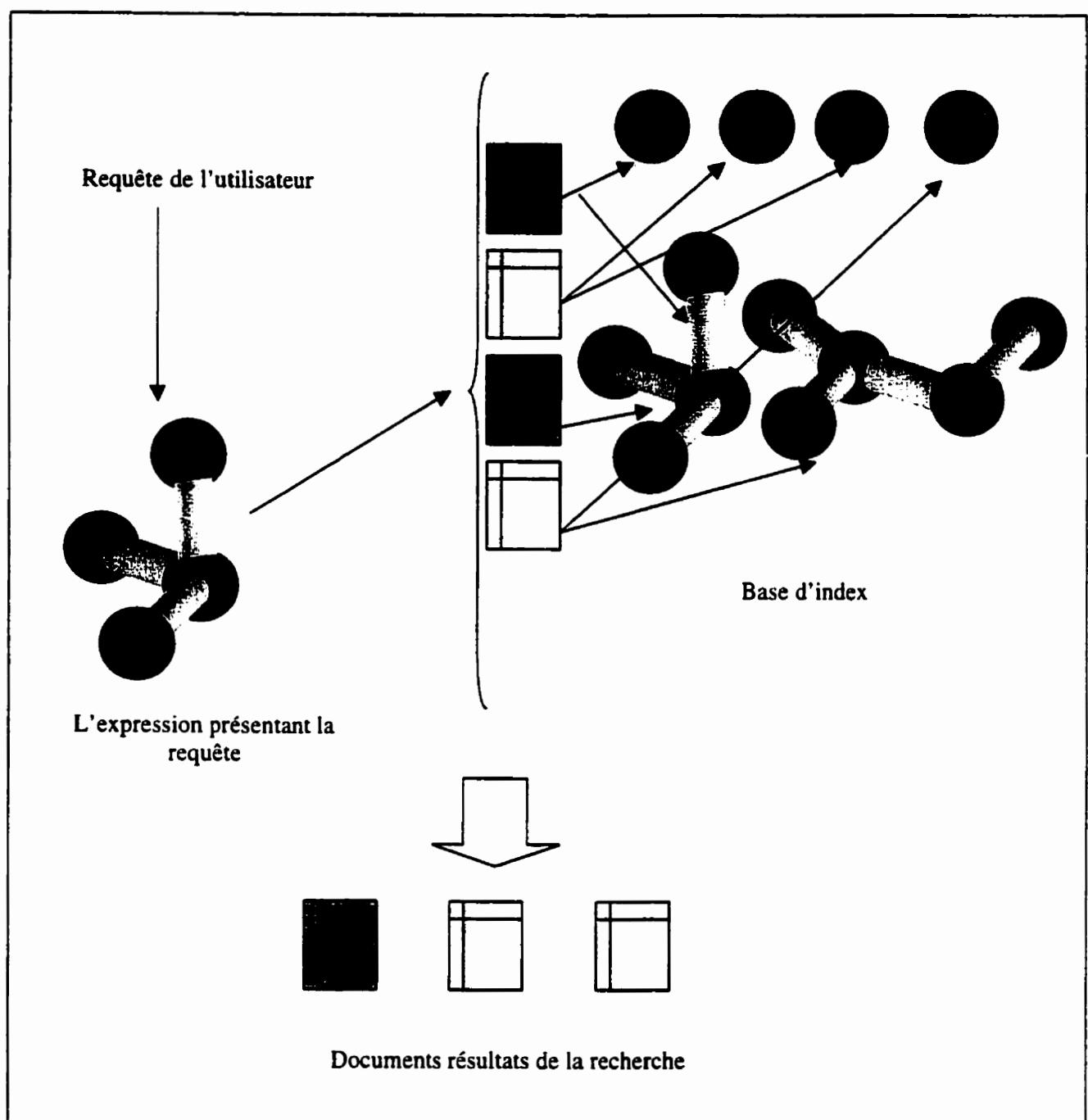


Figure numéro 4.24: Recherche basée sur les expressions

Dans la recherche basée sur les expressions, nous allons considérer la requête de l'utilisateur comme étant une expression entière. Le système va mettre en correspondance cette expression requête avec toutes les expressions de la base des documents.

expressions. Le résultat de l'appariement donne les documents indexés par l'expression requête ou bien par l'une de sous-expressions de cette dernière. Par exemple, si la requête de l'utilisateur est «Cancer» le résultat sera la liste de tous les documents qui parlent du «Cancer», du «cancer de la peau», du «Cancer du sein»...etc. Par contre si la requête de l'utilisateur est «Cancer du sein» le résultat sera la liste des documents qui parlent exactement du «Cancer du sein», «Cancer du sein chez les femmes de plus de 15 ans»...; mais pas les documents qui parlent du «Cancer de la peau» par exemple.

En utilisant cette technique, des nouvelles interfaces doivent aussi être mises en place. Dans ce qui suit, nous allons présenter les interfaces proposées pour l'indexation et la recherche basées sur les expressions.

a) Interfaces de l'indexation: (Voir figure numéro 4.25 (a))

L'indexation par expressions a pour but de simplifier l'interface d'indexation. C'est à dire que le ou les premiers termes entrés par l'utilisateur lui donne un choix d'expressions du domaine contenant ces termes. Ce procédé est couramment utilisé pour les aides en ligne. L'utilisateur choisit donc l'expression appropriée et réalise ainsi son indexation. Plusieurs expressions peuvent servir à indexer le fichier, la manipulation est alors réalisée plusieurs fois ce qui permet à l'utilisateur de remplir le fichier d'indexation concernant le document.

L'index associé à la recherche peut se faire de deux façons :

- Le début de l'expression entrée sert à extraire les expressions proposées à l'utilisateur

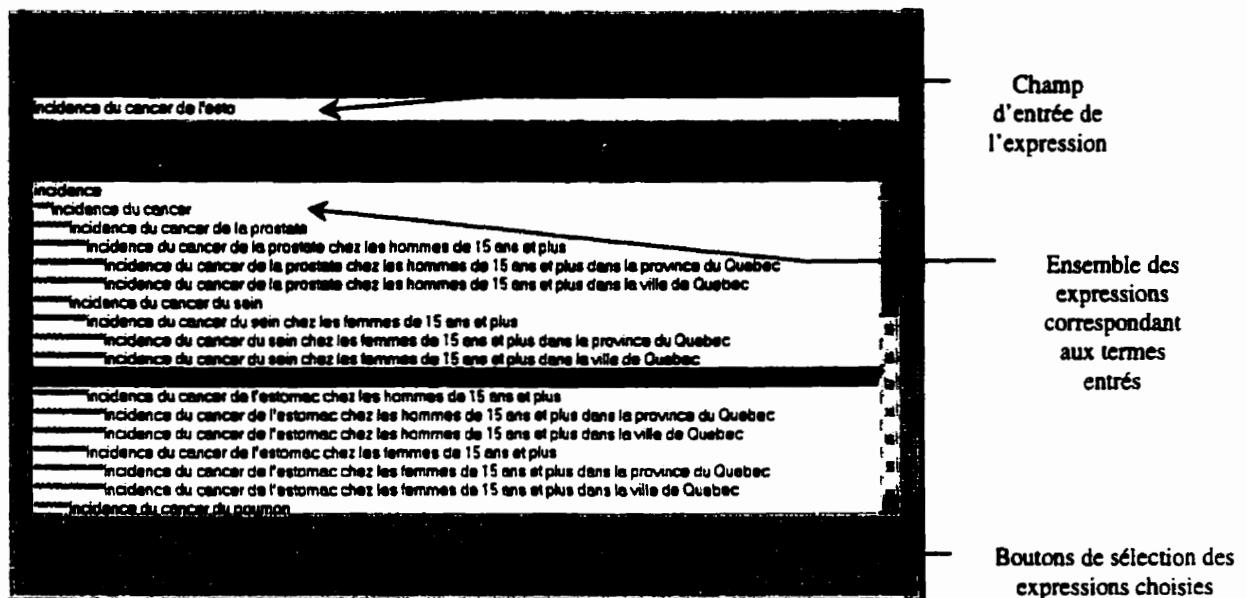


Figure numéro 4.25(a): Indexation basée sur la notion des expressions

L'avantage de cette interface est de pouvoir visualiser l'ensemble des expressions du domaine et éventuellement de choisir son expression.

- Les termes entrés servent à extraire les expressions proposées à l'utilisateur

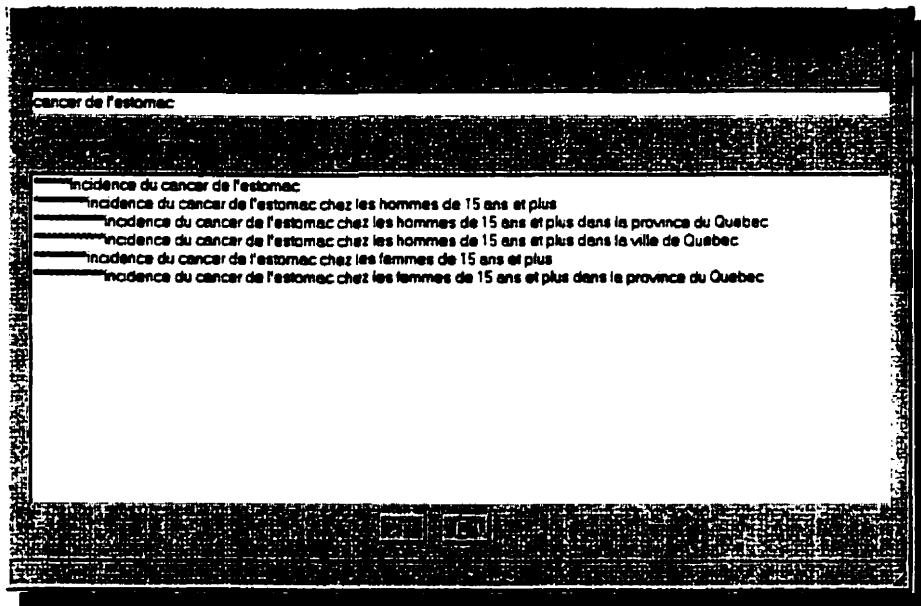


Figure numéro 4.25 (b): Recherche basée sur la notion des expressions

La restriction des expressions proposées à l'utilisateur est réalisée avec les termes entrés. Même si les termes ne sont pas en ordre dans les expressions, la restriction des expressions rend le choix plus pertinent et plus rapide. De plus, il reste aussi la possibilité de choisir son expression sans entrer aucun terme puisqu'il n'y a aucune restriction au départ, donc toutes les expressions sont affichées. Ce type d'interface est à privilégier, même si la programmation est plus difficile.

b)Interface de la recherche:

La recherche de documents peut se faire de la même façon. L'utilisateur donne au système les expressions sur lesquelles il veut faire une recherche, puis le système les traduit et fait la correspondance avec les fichiers d'indexations. Par exemple, si l'utilisateur recherche un document concernant « le cancer du poumon chez les hommes », alors le système retrouvera le document indexé par l'expression « l'incidence du cancer du poumon chez les hommes de plus de 15 ans ».

Avec l'interface choisie, l'utilisateur peut même entrer «Cancer poumons hommes», et le système lui retrouvera aussi le document.

L'avantage de cette nouvelle technique est d'être bâtie sur l'ontologie du domaine tout en ayant un ensemble d'expressions pré-déterminées. Cela permet de simplifier l'indexation.

4- 5 - La solution retenue:

Parmi les techniques que nous avons examinées, la première (par mots-clés) est facile à programmer et facile à manipuler par l'utilisateur, mais elle n'est pas efficace en terme de pertinence des résultats. La deuxième (par graphes conceptuels) est très efficace en terme de résultats de recherche, mais, sa programmation et son utilisation sont fastidieuses. La troisième technique est particulièrement intéressante pour notre domaine, elle est facilement manipulable pour l'indexation et pour la recherche. Cette technique présente des interfaces facilement manipulables par l'indexeur et l'utilisateur. Ces interfaces sont très proches que celles utilisées dans l'approche basée sur les mots-clés. Dans la conception et la réalisation de notre système, nous allons adopter cette dernière technique (basée sur les expressions).

4- 6 – Conclusion:

Dans ce chapitre, nous avons présenté la technique d'indexation et de recherche qui est utilisée actuellement par la plupart des systèmes d'indexation et de recherche. Cette technique est basée sur la notion des mots-clés. Cette technique présente beaucoup de bruit dans le résultats de la recherche. Nous avons présenté aussi deux autres techniques que nous avons envisagées pour notre système. La première est essentiellement sémantique et elle se base sur la structure des graphes conceptuels pour représenter les documents à indexer (indexation) ou les requêtes des utilisateurs (recherche). Cette technique a été rejetée car elle s'est avérée trop complexe à manipuler pour les utilisateurs. La troisième technique se base sur les expressions. L'utilisation des expressions est très intéressante pour notre domaine car la majorité des documents de ce domaine peuvent être représentés par des expressions plus ou moins complexes. L'avantage c'est que nous avons un domaine spécialisé pour lequel nous pouvons utiliser un vocabulaire spécialisé (ontologie) et donc retenir des expressions typiques. Dans notre travail, nous allons utiliser cette troisième technique

pour bâtir notre système. Dans le chapitre suivant, nous présentons cette technique en détail. Nous présentons aussi la structure de notre système et de ses sous-systèmes, la structure de l'ontologie sur laquelle il se base, ainsi que la structure de la base de données qu'il utilise.

Chapitre 5

La conception de la solution (Volet théorique)

5- 1 – Introduction:

Dans ce chapitre, nous présentons en détail notre solution, ainsi que le système d'indexation et de recherche des documents multimédia mis en œuvre. En premier lieu, nous présentons la structure générale de ce système. Puis, nous présentons en détail la structure de chacun de ses sous-systèmes, enfin, nous présentons la structure de sa base de données.

5- 2 - Présentation de la solution:

5- 3 - La structure du système d'indexation et de recherche des documents multimédia:

La structure du système peut être divisée en trois sous-systèmes:

- Sous-système d'indexation des documents multimédia.
- Sous-système de recherche des documents multimédia.
- Sous-système de gestion des ontologies.

La figure suivante ([figure numéro 5.1](#)) présente cette structure :

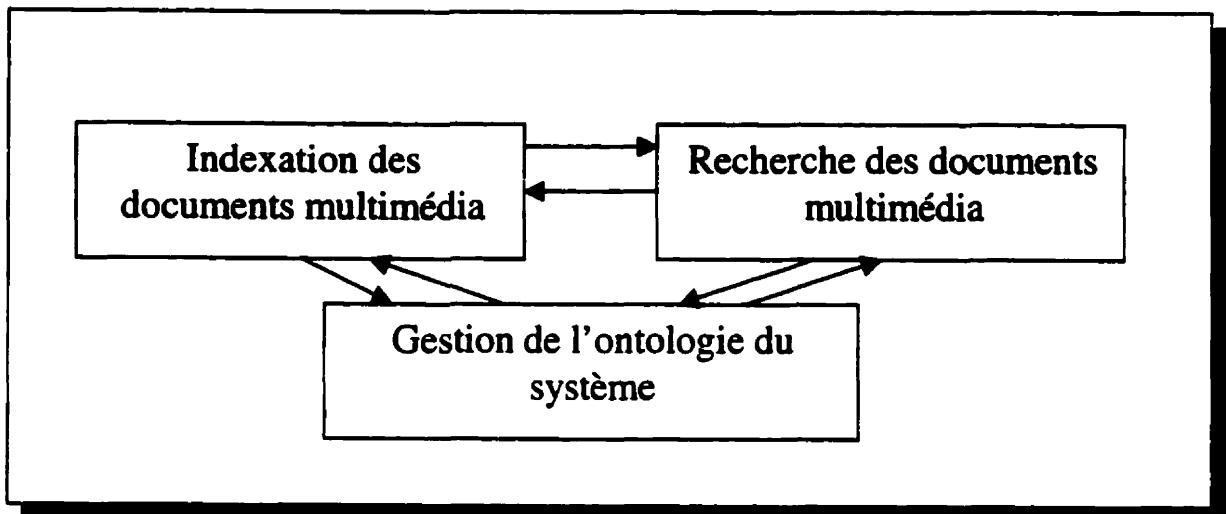


Figure 5.1: La structure du système d'Indexation et de recherche des documents multimédia

La structure du sous-système de recherche des documents multimédia est la suivante ([voir figure numéro 5.2](#)):

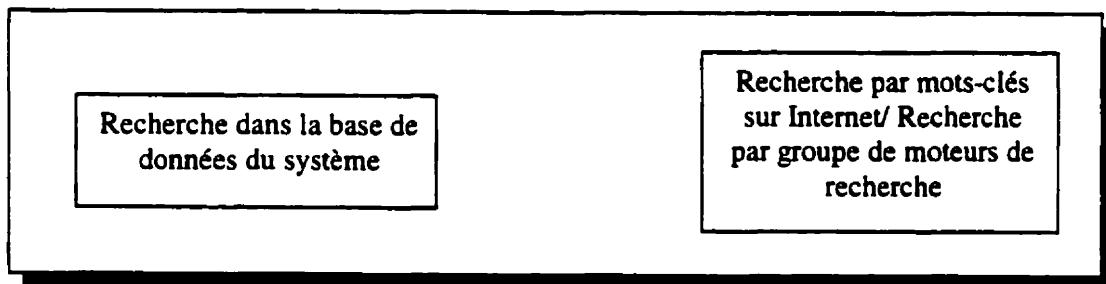


Figure 5.2: La structure du sous-système de recherche des documents multimédia

Dans les sections suivantes, nous présentons en détail chacun de ces sous-systèmes.

5- 3 – 1 - L'indexation des documents multimédias:

Le sous-système d'indexation des documents permet l'indexation des documents multimédia en utilisant la technique basée sur les expressions. Ce sous-système permet d'indexer tout type de document (textes, images fixes, images animées, des séquences audio, des séquences vidéos...) en utilisant cette unique technique. Du fait que le sous-système permet d'indexer des documents textuels et visuels en se basant sur une seule technique, l'indexation sera loin d'être automatique. La raison est que la technique se base sur la notion des concepts et des expressions que nous ne pouvons pas extraire des documents visuels, audio ou multimédia.

En nous basant sur cette technique, nous indexons les documents non pas par des mots-clés, mais par des concepts qui peuvent être unitaires ou composés (expressions).

Le fonctionnement du système d'indexation se présente dans la figure numéro 5.3.

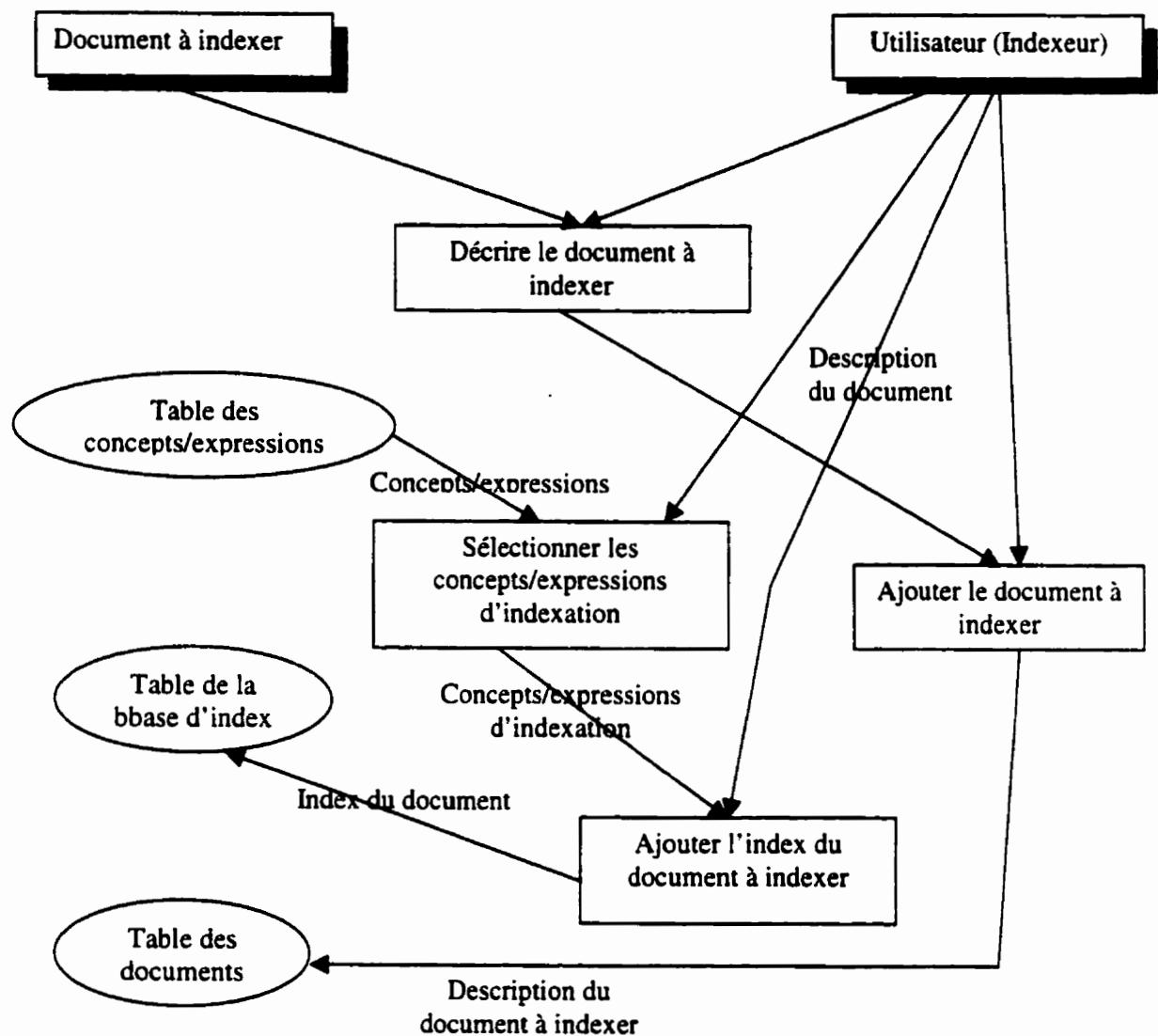


Figure numéro 5.3 : Le fonctionnement du sous-système d'indexation des documents multimédia

Dans la figure numéro 5.3, pour indexer un document, l'indexeur doit suivre les étapes suivantes :

- Faire une brève description du document à indexer.
- Consulter tous les concepts/expressions de l'ontologie du système, puis sélectionner ceux par lesquels cet utilisateur veut son document.
- Ajouter les concepts/expressions d'indexation à la table des indexes et ajouter le document à la table des documents.

Pour illustrer cette technique prenons un exemple:

Exemple:

Soit le corpus des documents suivant:

D1 : «...le cancer des seins chez les femmes de 15 ans et plus dans la région de Québec et la région de Montréal... et précisément à Laval (Montréal dans le Québec)...»

D2 : «...la province d'Ontario se trouve dans le sud du Canada...»

D3 : «...le tourisme dans la province du Québec est un secteur très important...»

D4 : «...l'université Laval se trouve dans la ville de Québec...»

En utilisant l'approche par mots-clés, la base d'index aura la forme suivante :

...

| | |
|------------|------------|
| Cancer | D1 |
| Sein | D1 |
| Femme | D1 |
| Region | D1 |
| Quebec | D1, D3, D4 |
| Montreal | D1 |
| Laval | D1, D4 |
| Province | D2, D3 |
| Ontario | D2 |
| Sud | D2 |
| Canada | D2 |
| Tourisme | D3 |
| Secteur | D3 |
| Universite | D4 |
| Ville | D4 |

...

Si nous utilisons la technique d'indexation qui se base sur les expressions, la base d'index aura la forme suivante :

| | | |
|--------------------|------------|---------------------------------|
| ... | | |
| Cancer | D1 | |
| Sein | D1 | |
| Femme | D1 | |
| Region | D1 | |
| Quebec | D1, D3, D4 | |
| Montreal | D1 | |
| Laval | D1, D4 | Concepts unitaires |
| Province | D2, D3 | |
| Ontario | D2 | |
| Sud | D2 | |
| Canada | D2 | |
| Tourisme | D3 | |
| Secteur | D3 | |
| Universite | D4 | |
| Ville | D4 | |
| ... | | |
| Cancer des seins | D1 | |
| Region de quebec | D1 | |
| Region de montreal | D1 | |
| Province d'ontario | D2 | |
| Sud du Canada | D2 | |
| Province du quebec | D3 | Concepts composés (expressions) |
| Universite Laval | D4 | |
| Ville de quebec | D4 | |
| Tourisme a quebec | D3 | |
| | | |

Comme le montre la structure de la base d'index ci-dessus, le premier champ de cette base n'est pas un simple dictionnaire de mots-clés, mais c'est un ensemble d'expressions unitaires ou composées. Cet ensemble d'expressions est sélectionné à partir de l'ontologie des expressions du système.

5- 3 – 2 - La recherche des documents multimédias:

Dans le sous-système de recherche des documents multimédia, nous avons deux modules de recherche. Le premier module s'intéresse à la recherche dans les documents stockés dans la base de données du système et qui sont indexés par l'administrateur de notre système. Le deuxième module s'intéresse à la recherche sur le réseau Internet. La description de ces deux modules est présentée dans les deux sous-sections suivantes.

5- 3 – 2 – 1 - La recherche dans la base de données du système:

Ce module permet la recherche des documents dans la base de données du système. Dans ce module de recherche, nous utilisons la technique de recherche basée sur la notion des expressions.

Dans notre système, tous les documents sont indexés par des concepts/expressions. Dans le processus de recherche, nous allons nous baser sur le même principe. Ceci veut dire que nous allons considérer la requête de l'utilisateur comme une expression, puis nous allons faire la correspondance (matching) entre cette expression-requête et les expressions par lesquelles sont indexés les documents de la base du système.

Le fonctionnement du système de recherche dans la base de données se présente dans la figure numéro 5.4.

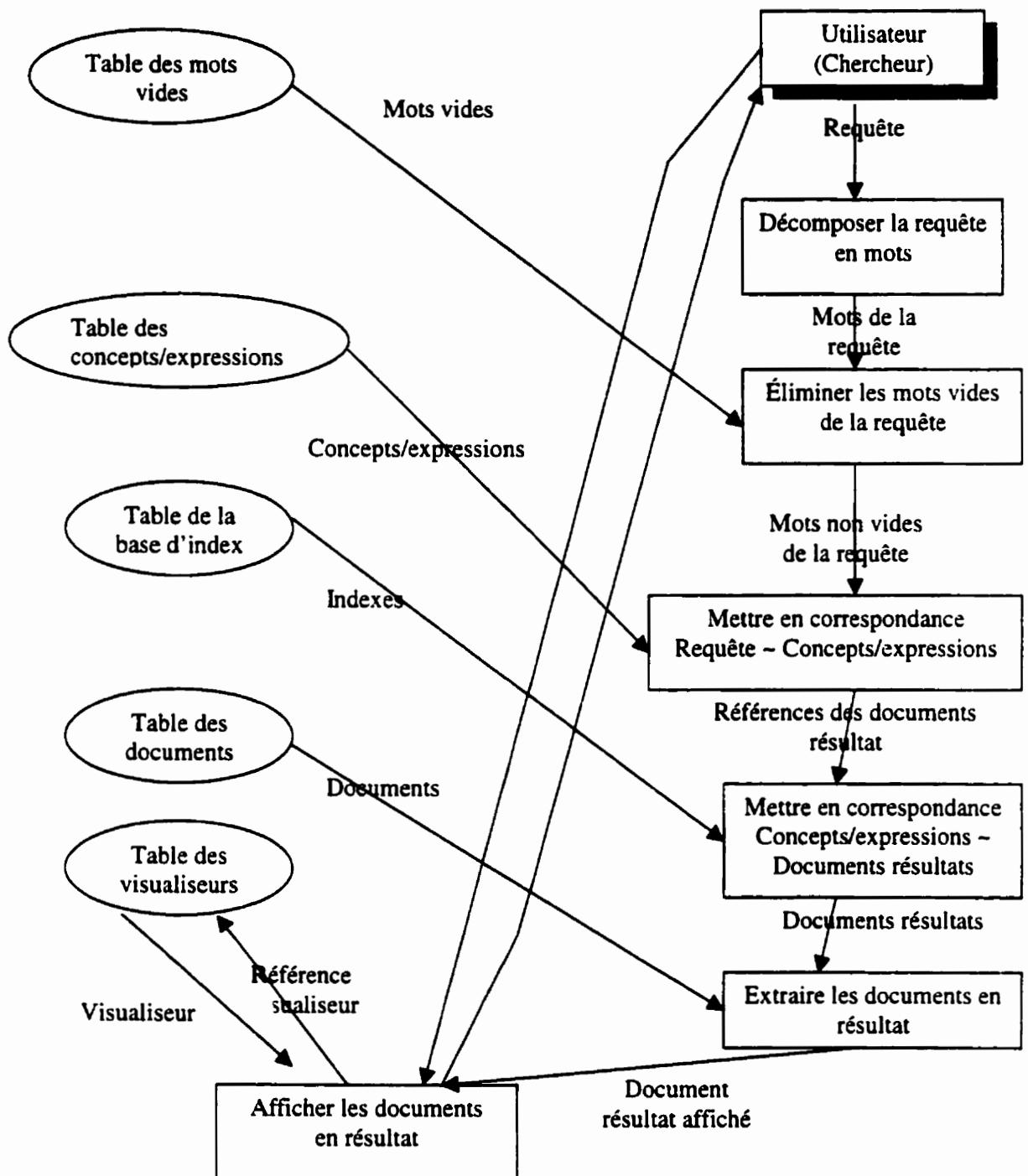


Figure numéro 5.4 : Le fonctionnement du sous-système de recherche dans la base de données

Dans la figure numéro 5.4, pour effectuer la recherche de documents dans la base de données du système, l'utilisateur commence par taper sa requête de recherche.

Au début, le système prend en charge cette requête, la décompose en mots-séparés, élimine les mots-vides de cette requête en se référant à la liste des mots vides dans la base de données. Puis, le système met en correspondance cette requête avec des concepts/expressions pour faire sortir ceux qui répondent à la requête de l'utilisateur. A partir de ces concepts/expressions résultat, le système se réfère à la table des indexés pour dégager les références des documents indexés par les concepts/expressions résultat. Ensuite, le système construit la liste des documents résultats en se référant aux références des documents résultats et à la table des documents dans la base de données. Enfin, le système affiche à l'utilisateur l'ensemble des documents en résultat. Lorsque l'utilisateur veut visualiser un de ces documents résultat, le système va se référer à la table des visualiseurs de la base de données pour visualiser le document en utilisant le visualiseur correspondant (par exemple visualiser un document Word dans le logiciel Word, ou une page web dans un navigateur).

Prenons un exemple pour présenter notre technique. Soit le corpus des documents suivant :

D1 : «...le cancer des seins chez les femmes de 15 ans et plus dans la région de Québec et la région de Montréal... et précisément à Laval (Montréal dans le Québec)...»

D2 : «...la province d'Ontario se trouve dans le sud du Canada...»

D3 : «...le tourisme dans la province du Québec est un secteur très important...»

D4 : «...l'Université Laval se trouve dans la ville de Québec...»

Si nous utilisons la technique basée sur la notion des mots-clés, la base d'index aura la forme suivante :

...

| | |
|------------|------------|
| Cancer | D1 |
| Sein | D1 |
| Femme | D1 |
| Region | D1 |
| Quebec | D1, D3, D4 |
| Montreal | D1 |
| Laval | D1, D4 |
| Province | D2, D3 |
| Ontario | D2 |
| Sud | D2 |
| Canada | D2 |
| Tourisme | D3 |
| Secteur | D3 |
| Universite | D4 |
| Ville | D4 |

...

Supposons que l'utilisateur fournit les trois requêtes suivantes:

R1: «Québec»

R2: «Tourisme Québec»

R3: «Université Laval»

Les résultats de ces requêtes sont les suivants:

R1: «Québec» : D1 + D2 + D3

R2: «Tourisme Québec» : (D1+ D2+D3) \cup D3 = D1+D2+D3

Or D1 et D2 ne parlent pas du tourisme.

R3: «Université Laval» : (D4) \cup (D1+D4) = D1+D4

Or D1 ne parlent pas de l'Université Laval.

Remarquons bien la présence d'un bruit énorme pendant le processus de recherche en se basant sur la technique classique par mots-clés.

Maintenant, si nous utilisons la technique d'indexation et de recherche basée sur la notion d'expressions, la base d'index aura la forme suivante :

| | |
|------------|------------|
| ... | |
| Cancer | D1 |
| Sein | D1 |
| Femme | D1 |
| Region | D1 |
| Quebec | D1, D3, D4 |
| Montreal | D1 |
| Laval | D1, D4 |
| Province | D2, D3 |
| Ontario | D2 |
| Sud | D2 |
| Canada | D2 |
| Tourisme | D3 |
| Secteur | D3 |
| Universite | D4 |
| Ville | D4 |

Concepts unitaires

| | |
|--------------------|----|
| ... | |
| Cancer des seins | D1 |
| Region de quebec | D1 |
| Region de montreal | D1 |
| Province d'ontario | D2 |
| Sud du Canada | D2 |
| Province du quebec | D3 |
| Universite Laval | D4 |
| Ville de quebec | D4 |
| Tourisme a quebec | D3 |

Concepts composés (expressions)

....

Supposons que l'utilisateur fournit les cinq requêtes suivantes :

R1: «Québec»

R2: «Tourisme Québec»

R3: «Université Laval»

R4: «Province»

R5: «Province d'Ontario»

Les résultats de ces requêtes sont les suivants:

R1: «Québec» : D1 + D2 + D3

R2: «Tourisme Québec» : (D3) (bruit = 0)

R3: «Université Laval» : (D4) (bruit = 0)

R4: «Province» : (D2+D3) \cup D2 = D2 + D3 (bruit = 0)

R5: «Province d'Ontario» : (D2) (bruit = 0)

Nous remarquons qu'en utilisant cette technique le bruit est nul. Tant que l'utilisateur spécifie la requête tant que les documents sont plus filtrés pour répondre exactement à la requête de cet utilisateur. Dans notre exemple, lorsque l'utilisateur tape la requête «Province d'Ontario» il aura en résultats le document D2 qui parle de la province d'Ontario et non pas le document D3 qui parle bien d'une «province» mais qui est «la province du Québec».

5- 3 – 2 – 2 - La recherche des documents sur le réseau Internet:

Ce module se charge de la recherche des documents sur Internet. Si l'utilisateur de l'application veut lancer sa requête sur le réseau Internet, il peut le faire à partir de notre système. Nous avons conçu un module qui permet de lancer un moteur, parmi une liste donnée de moteurs de recherche, en utilisant la requête de l'utilisateur comme paramètre.

Le fonctionnement du sous-système de recherche sur Internet est représentée dans la figure numéro 5.5.

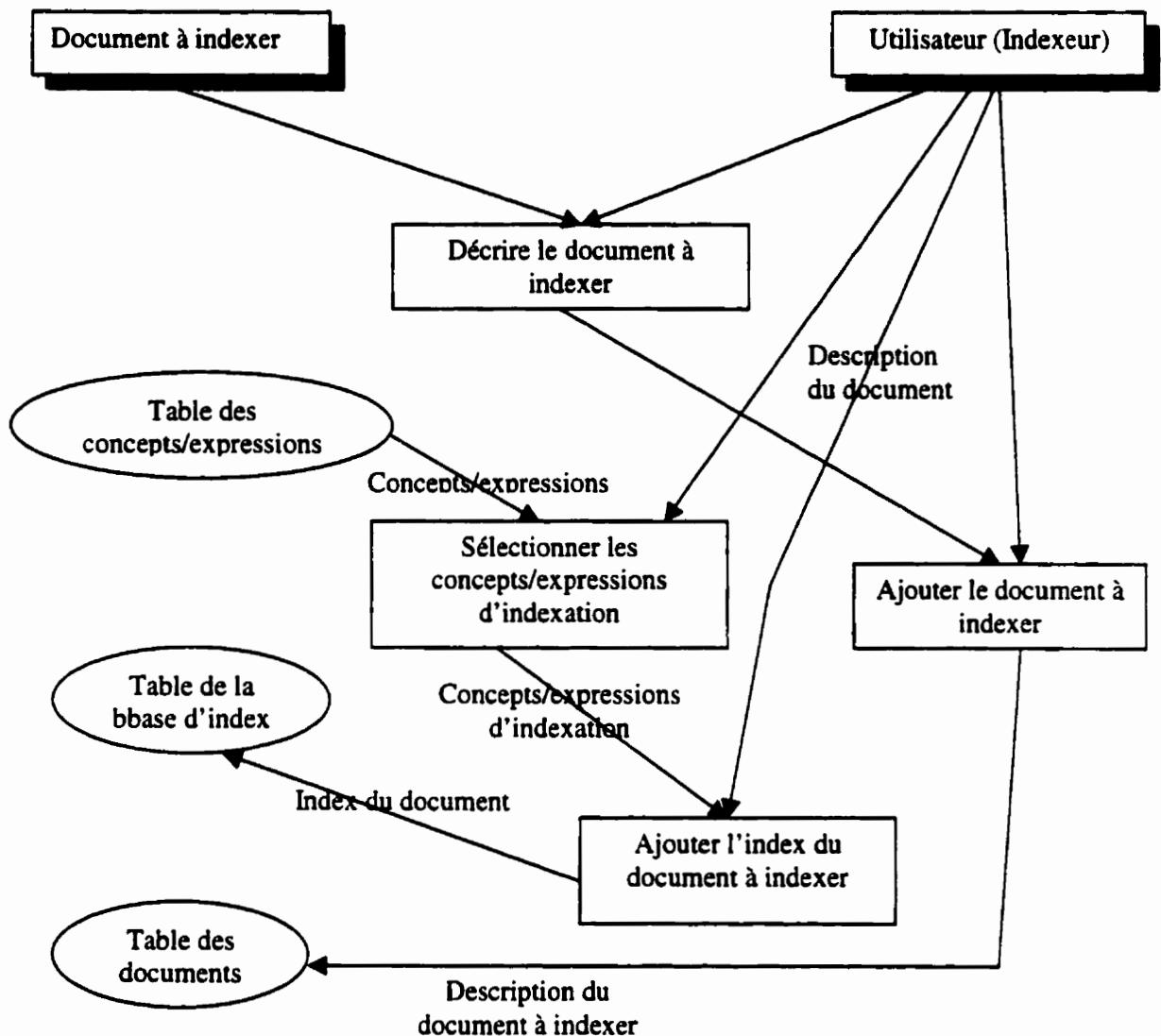


Figure 5.5 : Le fonctionnement du sous-système de recherche sur Internet

Dans la figure numéro 5.5, pour effectuer la recherche des documents sur le réseau Internet à partir du système, l'utilisateur commence par taper sa requête, puis il sélectionne le moteur de recherche qu'il veut lancer. Le système lance le moteur de recherche sélectionné en utilisant la requête comme paramètre. Plus en détail, le système prend en charge la requête de l'utilisateur, la décompose en mots, élimine les stop-words et les mots de marque, puis effectue une recherche dans les bases de données correspondantes.

mots vides de la requête en se référant par rapport à la table des mots vides dans la base de données du système et insère des opérateurs booléens entre les mots de cette requête pour pouvoir effectuer une recherche avancée sur Internet. La liste des moteurs de recherche présentée à l'utilisateur se trouve dans la table des moteurs de recherche dans la base de données du système. Le système lance le moteur de recherche sélectionné avec la requête en paramètre dans un navigateur externe.

Dans la base de données du système, nous avons toute une liste de moteurs de recherche les plus connus sur le réseau Internet. Cette liste de moteurs se présente dans le tableau suivant (voir tableau numéro 5.1):

| Rechercheur | |
|-------------------------|--|
| Altavista France | www.altavista.fr |
| Yahoo France | www.yahoo.fr |
| Excite France | www.excite.fr |
| Voilà | www.voila.fr |
| Google | www.google.com/ |
| Nomade | www.nomade.fr/ |
| Trouvez | www.trouvez.com/ |
| Altavista | www.av.com ou www.altavista.com ou www.altavista.digital.com |
| Altavista Canada | www.altavista.ca |
| Yahoo | www.yahoo.com |
| WebCrawler | www.webcrawler.com |
| Lycos | www.lycos.com |
| InfoSeek | www.infoseek.com ou www.go.com |
| HotBot | www.hotbot.com |
| Excite | www.excite.com |
| Lokace | www.lokace.com |
| Bonjour Mamma | www.mamma.com/bonjour.html |
| All the web | www.alltheweb.com/ |
| MetaFind | www.metacrawler.com/index_metafind.html |
| AudioFind | www.audiofind.com/search.html |
| StreamSearch | www.streamsearch.com/musichome.asp |
| StreamSearch | www.streamsearch.com/musichome.asp |
| MapFinder | ella.slis.indiana.edu/~jfieber/mapfinder/ |
| Mégagiciel | www.megagiciel.com |

| | |
|--------------------|--|
| Télécharger | www.telecharger.com/ |
| Deja | www.deja.com/ |
| WhoWhere | french.whowhere.lycos.com/ |

Tableau numéro 5.1 : Tableau des moteurs de recherche sur Internet

Remarque: Cette liste n'est pas exhaustive car le module permet à l'administrateur d'ajouter un nouveau moteur à la base du système, de modifier les paramètres d'un moteur ou de supprimer carrément un moteur.

Dans la recherche sur Internet, et afin de mieux aider l'utilisateur dans sa recherche, nous avons catégorisé les moteurs de recherche par catégories. Sous chaque catégorie, nous avons mis une liste bien déterminée de moteurs de recherche. Donc, l'utilisateur, avant de lancer sa requête, peut sélectionner la catégorie de moteur de recherche qu'il veut lancer, puis la liste des moteurs de recherche sous cette catégorie apparaît. L'utilisateur peut ainsi taper sa requête, sélectionner le moteur correspondant de cette catégorie et le lancer. La liste initiale de ces catégories ainsi que la liste des moteurs sous chaque catégorie sont les suivantes (voir tableau numéro 5.2):

| <u>WEB FRANCOPHONE</u> | |
|----------------------------------|---|
| Altavista France | www.altavista.fr |
| Yahoo France | www.yahoo.fr |
| Excite France | www.excite.fr |
| Voilà | www.voila.fr |
| Google | www.google.com/ |
| Nomade | www.nomade.fr/ |
| Trouvez | www.trouvez.com/ |
| <u>WEB MONDIAL</u> | |
| Altavista | www.av.com ou www.altavista.com ou www.altavista.digital.com |
| Altavista Canada | www.altavista.ca |
| Yahoo | www.yahoo.com |
| WebCrawler | www.webcrawler.com |
| Lycos | www.lycos.com |
| InfoSeek | www.infoseek.com ou www.go.com |
| HotBot | www.hotbot.com |
| Excite | www.excite.com |
| Lokace | www.lokace.com |
| Bonjour Mamma | www.mamma.com/bonjour.html |
| All the web | www.alltheweb.com/ |
| MetaFind | www.metacrawler.com/index_metafind.html |
| <u>TEXTES ET ARTICLES</u> | |
| <u>IMAGES ET PHOTOS</u> | |

| | |
|-----------------------------------|--|
| <u>AUDIO</u> | |
| AudioFind | www.audiofind.com/search.html |
| StreamSearch | www.streamsearch.com/musichome.asp |
| <u>VIDEO</u> | |
| StreamSearch | www.streamsearch.com/musichome.asp |
| <u>BASE DE DONNEES</u> | |
| <u>CARTES</u> | |
| MapFinder | ella.slis.indiana.edu/~jieber/mapfinder/ |
| <u>LOGICIEL</u> | |
| Mégagiciel | www.megagiciel.com |
| Télécharger | www.telecharger.com/ |
| <u>LISTES DE DIFFUSION</u> | |
| <u>FORUM DE DISCUSSION</u> | |
| Deja | www.deja.com/ |
| <u>ADRESSE EMAIL</u> | |
| WhoWhere | french.whowhere.lycos.com/ |
| <u>PAGES JAUNES</u> | |

Tableau numéro 5.2 : Tableau des catégories des moteurs de recherche sur Internet

Le système donne le choix à l'administrateur ou l'utilisateur d'ajouter son propre groupe de moteurs de recherche, ainsi que les moteurs sous ce groupe.

5- 3 – 3 - La gestion de l'ontologie du système:

a) Avantage de l'utilisation d'une ontologie:

Il est important de rappeler, que notre système se base sur une ontologie de concepts/expressions et une ontologie de relations conceptuelles entre ces concepts.

Ces deux ontologies ne sont pas figées et elle peuvent être enrichies au fur et à mesure par l'administrateur du système. Pour cela nous prévoyons un module destiné à l'administration de ces deux types d'ontologies.

Ces ontologies prennent une forme de treillis.

L'ontologie des concepts/expressions comporte des concepts unitaires ou des concepts/expressions liés par des liens de parenté. En plus de ces relations de parenté, ces concepts/expressions sont liés par différentes relations qui se trouvent dans la hiérarchie des relations. L'ontologie des relations comporte les différentes types de relations qui peuvent exister entre les concepts de l'ontologie des concepts.

L'introduction de différents types de relations entre les concepts/expressions permet une indexation et recherche plus intéressantes que l'indexation et la recherche qui se basent sur les mots-clés. Prenons un exemple.

- Dans le cas d'indexation et recherche basées sur la notion des mots-clés : Si nous indexons un document par le mot «cancer». Si nous introduisons la requête «cancer et sein», le document n'apparaîtra pas dans le résultat car il n'a pas été indexé par les mots «cancer» et «sein».

- Dans le cas d'indexation et recherche qui se basent sur une ontologie, les documents indexés par le concept «cancer» vont apparaître dans le résultat de recherche de la requête «cancer du sein» par le fait que le concept «cancer» détient une relation de parenté avec le concept «cancer de sein». Ainsi pour obtenir une indexation plus fine, il est recommandé d'employer des expressions plus précises.

Dans l'ontologie nous pouvons introduire d'autres types de relations. Un exemple de relation est la relation de synonymie. Prenons un autre exemple montrant l'utilité de cette relation.

- Dans le cas de l'indexation et recherche basées sur la notion des mots-clés : Si nous indexons un document par les mots «États Unis», ce document n'apparaîtra pas dans le résultat de la requête «USA».

- Dans le cas d'indexation et recherche qui se basent sur une ontologie les documents indexés par le concept «Tass Unis» vont apparaître dans le résultat de la recherche de la requête «USA» car le concept «États Unis» et «USA» détiennent une relation de synonymie entre eux.

b) La structure de treillis:

Pour gérer une ontologie, il faut tout d'abord gérer sa structure de treillis.

La structure de l'ontologie est une structure de treillis, donc elle doit accepter le fait qu'un élément peut avoir plusieurs pères.

Dans la représentation d'un treillis, les flèches représentent une relation (père → fils) (voir figure numéro 5.6)

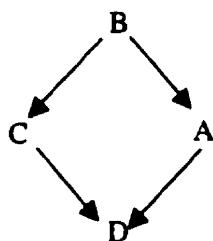


Figure 5.6 :Exemple de treillis

Cette structure permet d'implanter l'héritage, c'est-à-dire qu'un élément fils « hérite » des propriétés du père. Dans l'exemple ci-dessus (voir figure numéro 5.6), D possède les caractéristiques de C et de A. Et comme C (ou A) descend de B, D hérite des caractéristiques de B.

A côté de cette relation d'héritage, nous avons ajouté un autre type de relations dans la structure de l'ontologie : la relation d'incompatibilité.

Lorsqu'un élément A est relié à un autre élément B par une relation d'incompatibilité, A ne peut pas être ni un descendant ni un ascendant de B et vice versa. En plus, les éléments A et B ne peuvent pas avoir des descendants communs. Dans l'exemple de la figure numéro 5.7, il existe une relation d'incompatibilité entre le concept «*être humain*» et le concept «*plante*»: une personne ne peut pas être en même temps «*être humain*» et «*plante*».

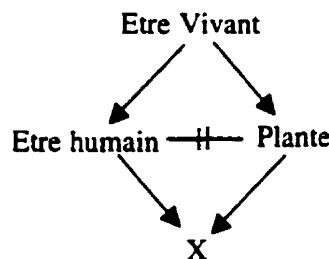


Figure 5.7 :Exemple d'incompatibilité

Cette relation est symbolisée par le symbole : $\perp\!\!\!\perp$

L'élément X ne peut donc pas hériter en même temps de ces deux concepts. Le graphe tel qu'il est représenté n'est donc pas correct (une flèche arrivant à X est en trop).

Avec les deux relations d'héritage et d'incompatibilité, il est ainsi possible de bâtir une ontologie de base assez complexe.

Des règles de construction et diverses opérations ont été définies, pour rendre l'ontologie homogène.

c) Les règles de construction:

Les relations de hiérarchie sont orientées (Père → Fils). Nous en déduisons plusieurs propriétés :

- Si X est le **père** de Y, alors Y est le **fils** de X

$$X \rightarrow Y$$

- Si X est le **père** de Y et Y le **père** de Z, alors Z est le **descendant** de X (propriété de transitivité de la relation) et X est l'**ascendant** de Z

$$X \rightarrow Y \rightarrow Z$$

- X ne peut être ni père, ni fils de lui-même (Incohérence 1) (Non reflexivité)

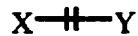


- Le **père** Y d'un élément X ne peut pas être le **descendant** de X, et a fortiori ne peut pas être le **fils** de X. (Incohérence 2) (Graphe acyclique)



Les relations d'incompatibilité ne sont pas orientées et concernent deux éléments. Là aussi, plusieurs propriétés sont à prendre en compte :

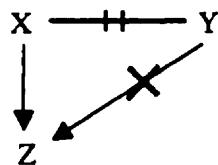
- Si X est incompatible avec Y, alors Y est incompatible avec X



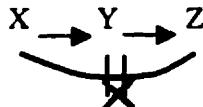
- X ne peut pas être incompatible avec lui-même (Incohérence 3)



- Si X est incompatible avec Y et que Z est le fils de X, alors Z est incompatible avec Y (Soit Y ne peut descendre de Y) (Incohérence 4)



- Si Z est un descendant de X, il ne peut y avoir incompatibilité entre X et Z. (Incohérence 5)



d) Les opérations de gestion de la hiérarchie:

Considérant toutes les règles définies précédemment, il est possible de définir les opérations possibles sur une structure gérant les relations d'héritage et d'incompatibilité. Prenons un exemple de hiérarchie construite en respectant les règles définies ci-dessus (voir figure numéro 5.8):

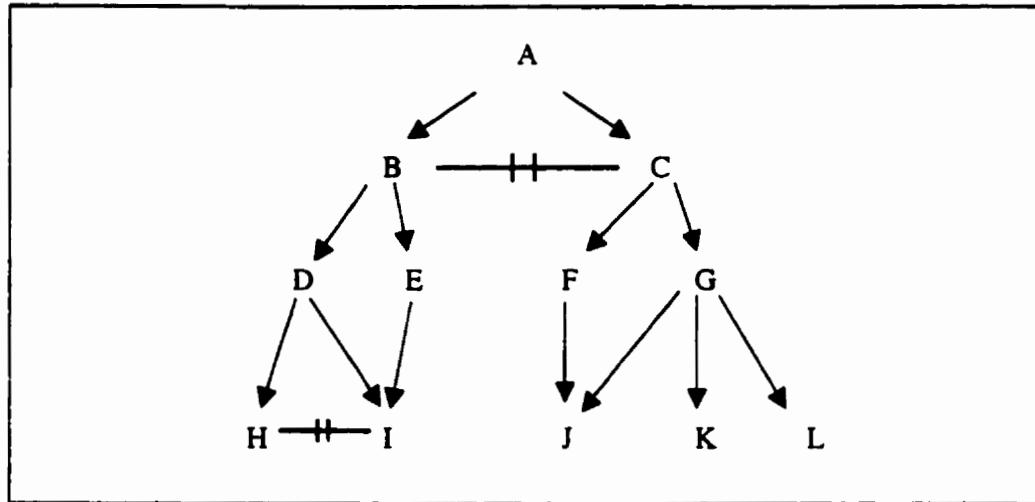


Figure 5.8 : Exemple de hiérarchie

Dans la figure numéro 5.8, l'élément A est le père de l'ontologie, c'est-à-dire que tous les éléments de l'ontologie sont des descendants de A, et tant que l'ontologie existe, cet élément existe.

* **Ajout d'élément** (voir figure numéro 5.9) :

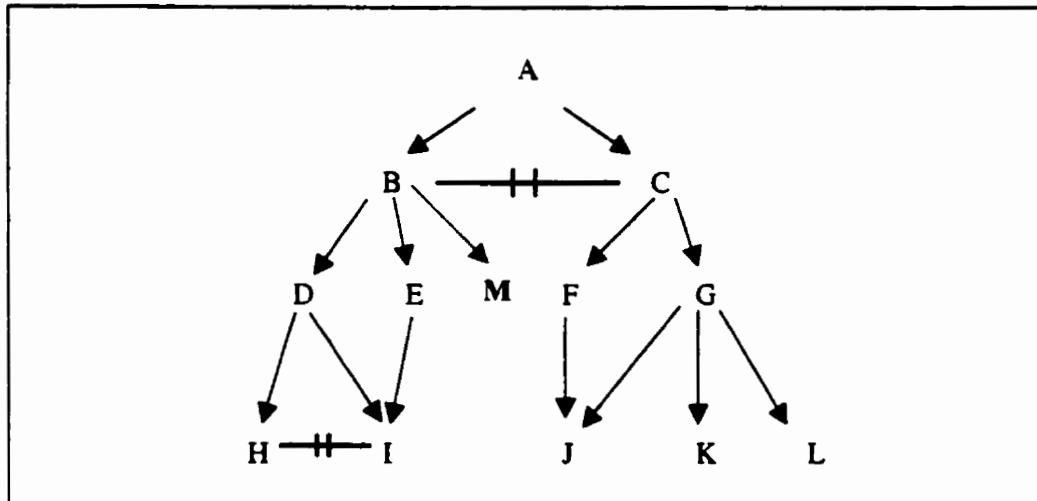


Figure 5.9 : Ajout d'un élément

L'ajout d'un nouvel élément M dans la hiérarchie ne pose pas de problème, dans la mesure où on connaît le père (ici B). Il suffit de créer l'objet M et d'associer une relation de hiérarchie entre le père B et le fils M.

Test à effectuer avant l'opération :

- Vérifier que M n'existe pas déjà dans l'ontologie.

* Ajout de relations hiérarchiques (voir figure numéro 5.10):

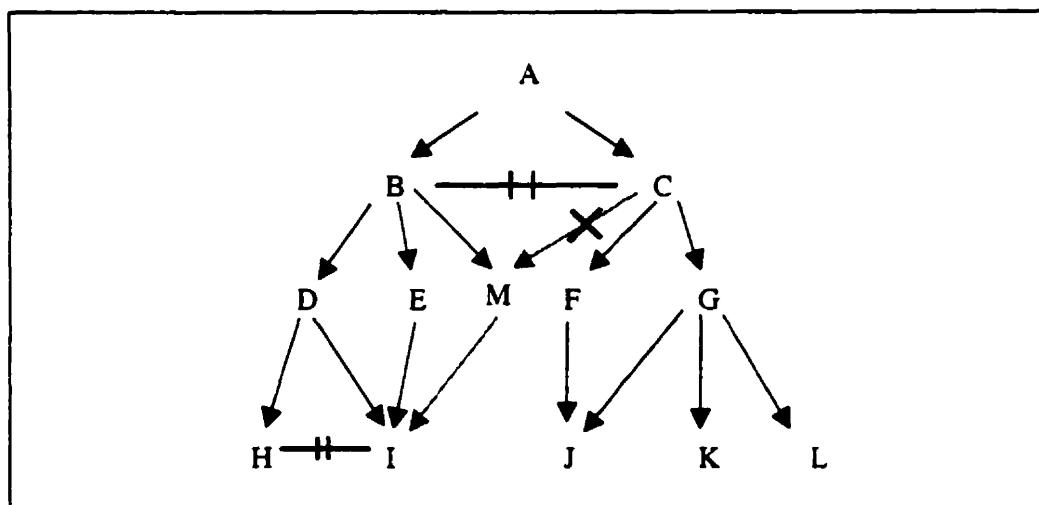


Figure 5.10 : Ajout d'une relation

L'ajout d'une relation demande plus de vérifications, que ce soit au niveau de la hiérarchie en elle-même qu'au niveau des incompatibilités.

Tests à effectuer avant l'opération :

- L'origine et l'extrémité de la relation sont des éléments différents.
(découle de l'incohérence 1)
- La relation n'existe pas déjà
- Le père pour la relation n'est pas déjà un descendant du fils de la relation
(découle de l'incohérence 2)

- Les éléments liés ne descendent pas d'une incompatibilité (découle de l'incohérence 4)

Dans l'exemple ci-dessus, M descend de B avec B et C incompatibles, donc la relation entre C et M est impossible.

* Ajout de relations d'incompatibilité (voir figure numéro 5.11):

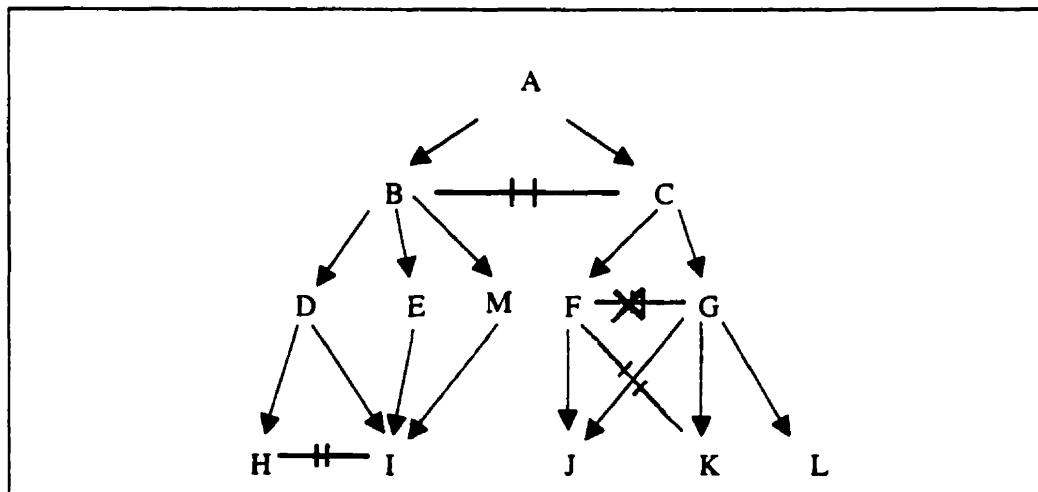


Figure 5.11 : Ajout d'une relation d'incompatibilité

L'ajout d'une incompatibilité est délicate, car il faut vérifier qu'il n'y ait pas de descendants communs aux deux éléments incompatibles.

Tests à effectuer avant l'opération :

- L'origine et l'extrémité de la relation sont des éléments différents.
(découle de l'incohérence 3)
- L'incompatibilité n'existe pas déjà
- Les éléments n'ont pas de relations hiérarchiques entre eux.
(découle de l'incohérence 5)

- Les éléments n'ont pas de descendants communs
(découle de l'incohérence 4)

Dans l'exemple ci-dessus, une incompatibilité entre F et G est impossible car ils ont J comme descendant commun.

Par contre, une incompatibilité entre F et K est possible car ils n'ont aucun lien hiérarchique et aucun descendant commun.

* Suppression de relation hiérarchique (voir figures numéro 5.12 et 5.13):

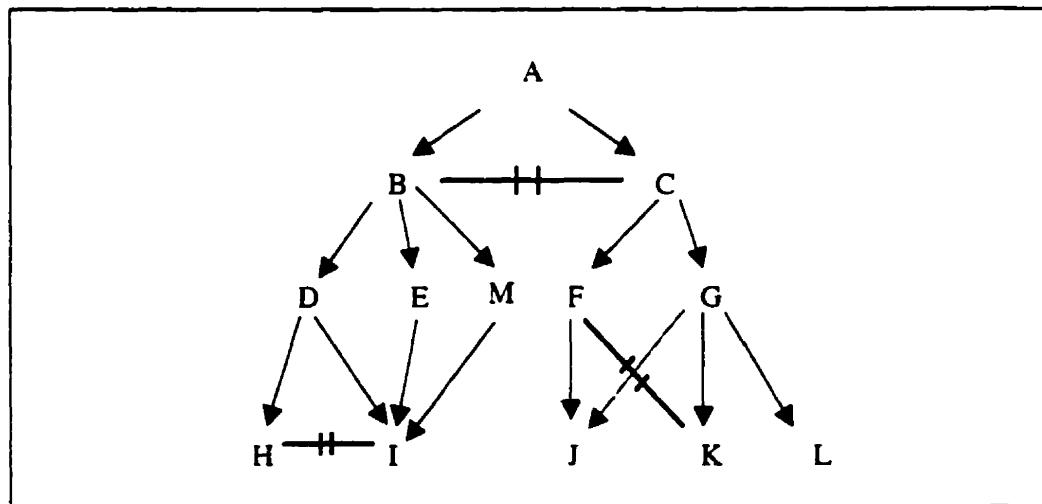


Figure 5.12 : Suppression d'une relation hiérarchique

La suppression d'une relation ne pose généralement pas de problème, sauf si le fils de la relation n'a qu'un seul père, auquel cas la suppression de la relation supprimerait également le fils.

Tests à effectuer avant la suppression d'une relation : vérifier que le fils possède plus d'un parent

Si un élément ne possède qu'un parent et que l'utilisateur souhaite supprimer la relation qui le lie avec son parent, il sera proposé à l'utilisateur de supprimer de

l'ontologie l'élément en question ou de le rendre premier fils de l'ontologie, c'est-à-dire qu'une relation de hiérarchie sera créée entre l'élément et le père de l'ontologie.

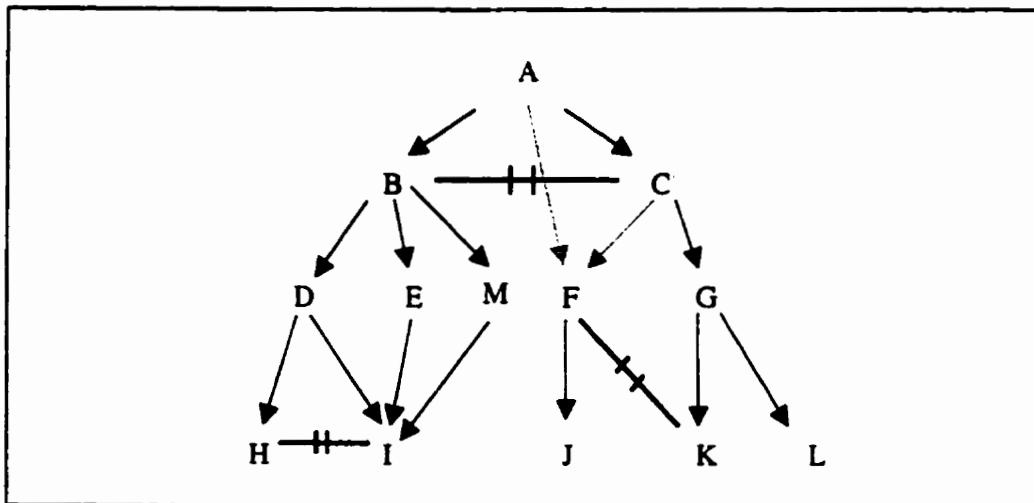


Figure 5.13 : Suppression d'une relation hiérarchique

Dans l'exemple ci-dessus, la relation entre C et F est supprimée et une relation entre A et F est créée.

* **Suppression de relation d'incompatibilité (voir figure numéro 5.14):**

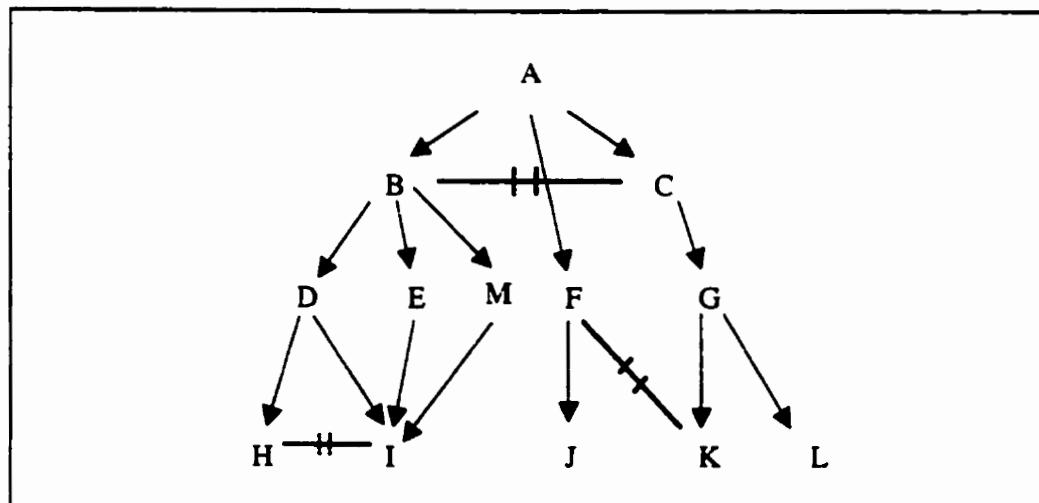


Figure 5.14 : Suppression d'une relation d'incompatibilité

La suppression d'une incompatibilité est toujours possible et n'impose pas de test, puisque chaque élément garde au moins une relation hiérarchique et ne peut donc pas être exclu de l'ontologie.

Test à effectuer avant la suppression d'une incompatibilité : aucun

* Suppression d'un élément (voir figure numéro 5.15):

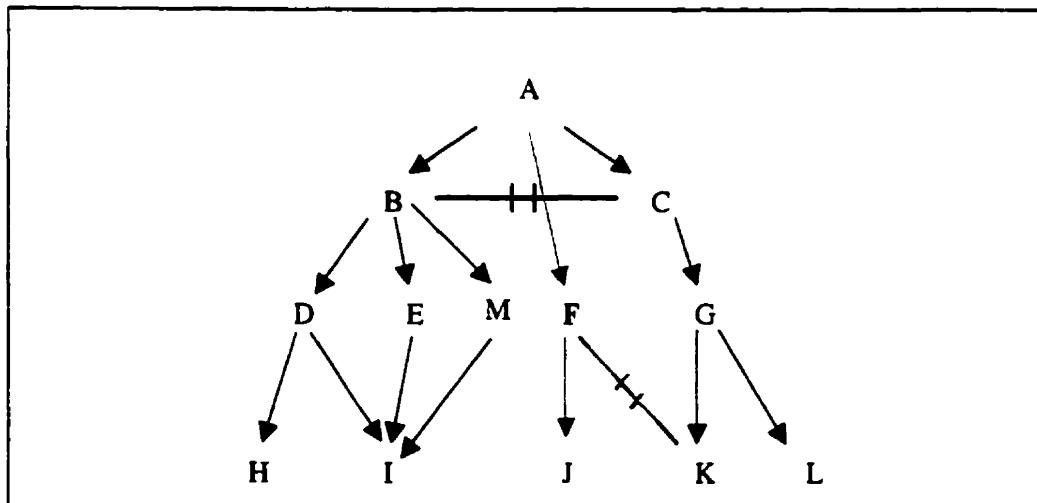


Figure 5.15 : Suppression d'un élément

La suppression d'objet est assez délicate à cause de la hiérarchie. En effet, toutes les relations qui se rapportent à l'objet doivent être supprimées. Il faut donc vérifier pour chaque relation les critères que nous avons définis pour une suppression de relation.

Dans le cas présent (voir figure numéro 5.15) , la suppression de F entraîne la suppression des relations ($A \rightarrow F$) et ($F \rightarrow J$). Il faut également vérifier que J a encore un parent. Si J n'a plus de parent, il faut le supprimer de l'ontologie, ou le mettre comme premier fils de l'ontologie, selon le choix de l'utilisateur.

Tests à effectuer :

- Vérifier que l'objet à supprimer n'est pas le père de la hiérarchie

- Effectuer les tests sur la suppression des relations (hiérarchiques et d'incompatibilité) liées à l'objet

e) Les autres types de relations

Une ontologie de concept/expression peut contenir des concepts similaires, pour cela, une autre relation a été incluse dans la définition de la hiérarchie, la relation de synonyme.

Cette relation n'influe pas directement sur la structure de l'ontologie, mais permet d'éviter des incohérences par la suite. En effet, plusieurs termes ou expressions peuvent être associés à un même sens. Les termes 'personne' et 'individu' en sont un bon exemple. Si deux concepts sont définis séparément tout en ayant le même sens, il pourrait arriver que chacun soit descendant de concepts incompatibles. Si par la suite une relation de synonyme est créée entre les deux, l'ontologie deviendrait incohérente.

La solution proposée consiste à ne définir qu'un concept et à lui associer des synonymes. Ces synonymes peuvent être des concepts, mais ils peuvent aussi être des termes ou des expressions équivalentes au niveau du sens.

Pour donner à notre ontologie un caractère plus général, nous ne pouvons nous limiter seulement à ces relations définies préalablement, mais nous sommes allés un peu plus loin. Alors nous avons pensé à d'autres relations. Ces relations peuvent aussi être nécessaires dans la définition de la hiérarchie. Par exemple, il existe des relations d'appartenance (« le chien de Pierre », de caractéristique (« une pomme rouge »), de position géographique (« le chat à-côté-de l'arbre »)...

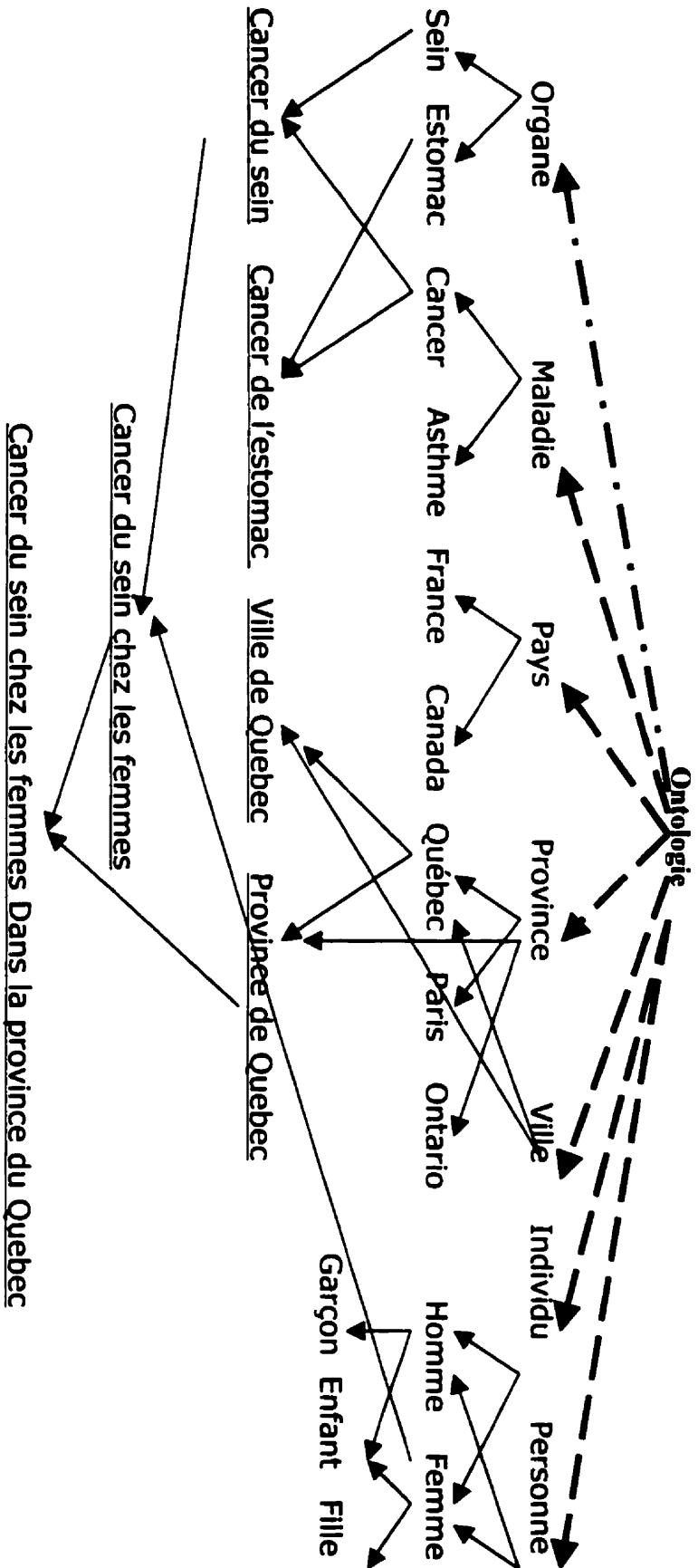
Ces relations peuvent être orientées mais ne sont pas transitives, ce qui évite d'avoir des tests avant de les créer ou de les supprimer. Le seul test consiste à vérifier si la relation existe avant de la recréer inutilement.

Étant donné que ces relations ne posent pas de problèmes particuliers et ont toutes des comportements similaires, il est prévu de donner à l'utilisateur la possibilité de créer ses propres relations, spécifiques à son domaine, et donc à son ontologie.

Une contrainte a toutefois été donnée à ces relations : une relation ne peut relier que deux éléments de l'ontologie à la fois. Si une relation met en jeu plus de deux éléments, il faut la diviser en plusieurs sous-relations mettant en jeu uniquement deux éléments.

f) Exemple d'ontologies :

Exemple d'une ontologie de concepts/expressions :



Cancer du sein chez les femmes Dans la province du Quebec

Lien Direct ou indirect

← Lien Direct

Exemple d'une ontologie de relations :

Logique

ET
OU
Ou exclusif

Ordre

Différence

Contraire à
Sans intersection avec
Opposé à

Inégalité

Inférieur
Inférieur ou égal à (nombre)
Supérieur
Supérieur ou égal à (nombre)

Similitude

Égal à
Équivalent à
Proche de

Spécialisation

Instance de
Sous-catégorie de

Spatiale

Topologique

Contenu dans
Composé de
Union de
Intersection de

Directionnelle

A droite de
A gauche de
A l'extérieur de
A l'intérieur de
Dans
Derrière
Sur
Dessous

Devant

Temporelle

Topologique

A l'intérieur de

Pendant

Instant de

Période de

égale

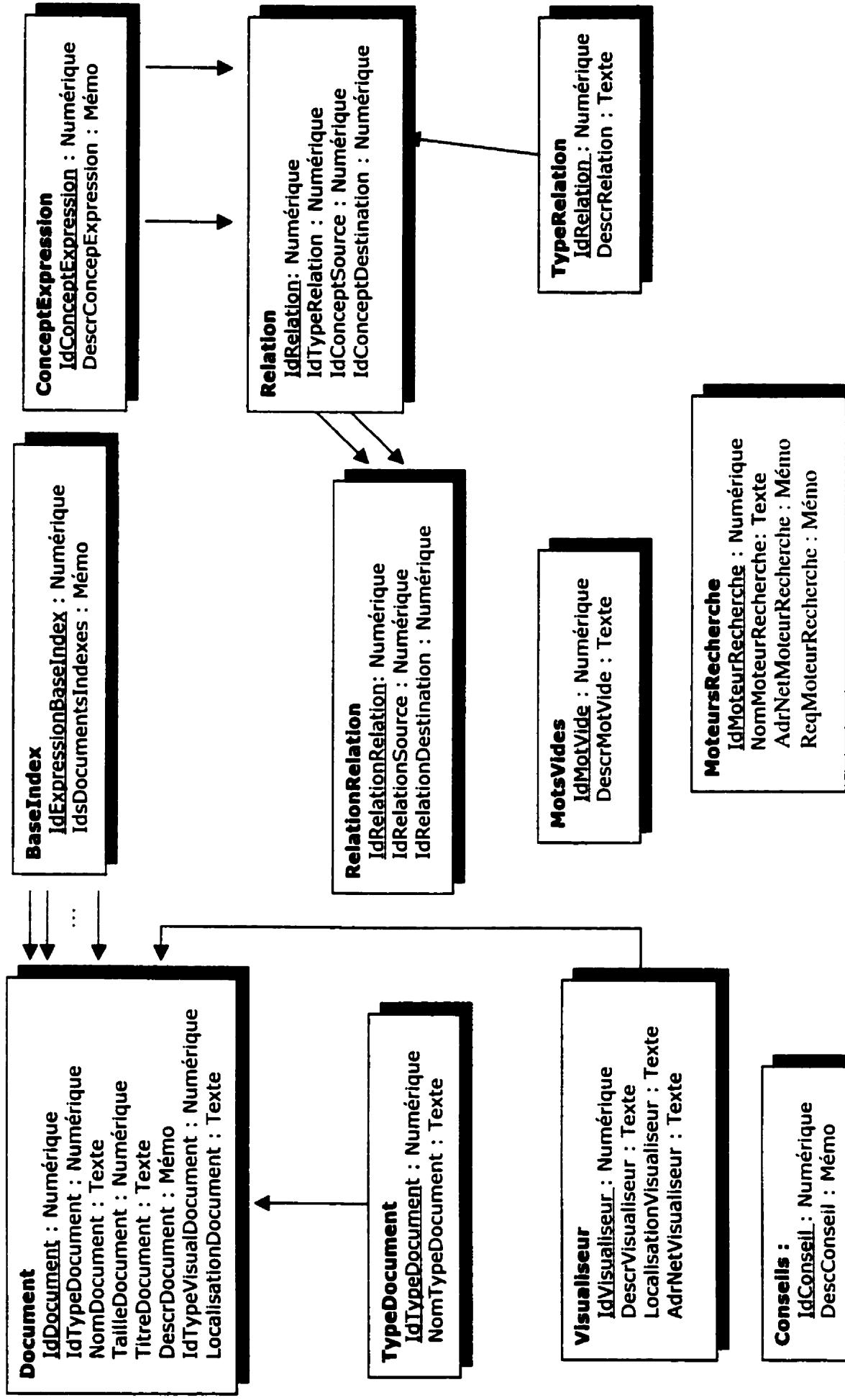
Directionnelle

Avant

Aprè

5- 4 - La structure de la base de données de l'application:

Le système utilise une base de données Access créée en utilisant le logiciel Access 2000 de MicroSoft. La structure de la base de données est la suivante :



5- 5 – Conclusion:

Dans ce chapitre nous avons présenté l'aspect conception de notre système d'indexation et de recherche des documents multimédias. Nous avons présenté la structure du système, celles de ses sous-systèmes (indexation, recherche dans la base de données, recherche sur Internet et gestion des ontologies) ainsi que la structure de la base de données sur laquelle il se base. Dans le chapitre suivant, nous présentons l'aspect développement de notre système ainsi que les différentes interfaces de ce dernier.

Chapitre 6

La réalisation de la solution (Volet pratique)

6- 1 – Introduction:

Après la conception du système d'indexation et de recherche des documents multimédias, nous présentons la phase de réalisation.

Dans ce chapitre, nous décrivons l'environnement de réalisation du système. Ensuite, nous présentons les interfaces personne-machine de l'application réalisée. Ces interfaces vont être présentées successivement pour les modules d'indexation des documents, de recherche des documents (dans la base de données ou sur le réseau Internet) et de gestion des ontologies du système. Enfin, nous clôturons le chapitre par une conclusion.

6- 2 - L'environnement de réalisation de l'application:

Le *Système d'Indexation et de Recherche des Documents Multimédias (SIRDM)* a été réalisé sous l'environnement *C++ Builder Version 5.0* de la compagnie *Borland*, sous le système d'exploitation *Windows NT 4.1*. La base de données du système a été créée à l'aide du logiciel *Access version 2000* de *MicroSoft*.

6- 3 – Conception et construction des interfaces personne-machine:

Pour le prototype du système, des maquettes sous forme papier ont été d'abord utilisées pour la conception des interfaces personne-machine. Ces maquettes ont été améliorées au fur et à mesure des interactions avec les futures usagers. Ensuite, un prototype exécutable a été construit. Dans ce qui suit, nous présentons les interfaces personne-machine de l'application.

6- 3 – 1 - La fenêtre principale de l'application:

La fenêtre principale de l'application est présentée dans la figure numéro 6.1. Elle comporte le menu général et la barre d'outils.

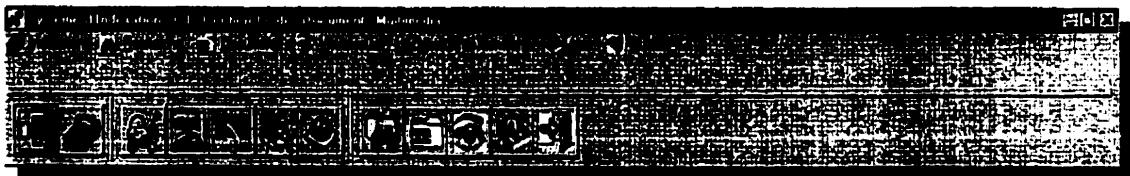


Figure 6.1 : Menu principal et barre d'outils

a) Les éléments du menu principal de l'application:

Le premier élément du menu s'intitule «*Indexation*» (voir figure numéro 6.2). Il concerne l'aspect indexation des documents et l'aspect administration des ontologies du système.

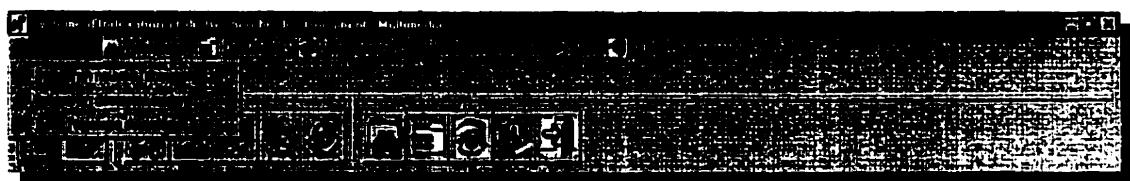


Figure 6.2 : Menu d'indexation des documents et de gestion de l'ontologie

Le deuxième élément du menu concerne l'aspect «Recherche des documents». Il est intitulé «*Recherche*» (voir figure numéro 6.3). En cliquant sur cet élément, l'utilisateur accède à la fenêtre de recherche classique des documents multimédia, à la fenêtre de recherche avancée et aux fenêtres de recherche sur Internet.



Figure 6.3 : Menu de recherche des documents

L'élément suivant concerne la gestion des documents. Il est intitulé «*Documents*» (voir figure numéro 6.4). En cliquant sur cet élément, l'administrateur peut accéder à la fenêtre de gestion des documents du système.

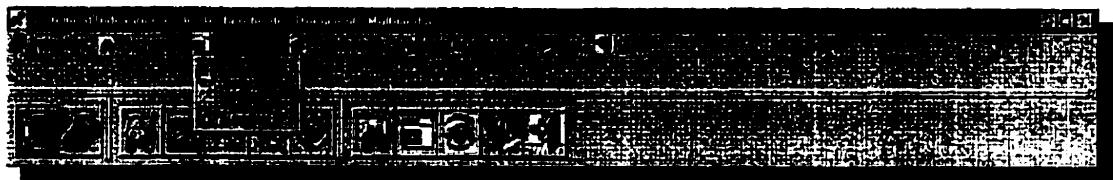


Figure 6.4 : Menu de gestion des documents

Le quatrième élément s'appelle «*Visualiseurs*» (voir figure numéro 6.5). Il permet à l'administrateur de l'application d'accéder à la fenêtre de gestion des visualiseurs sur lesquels se base l'application.

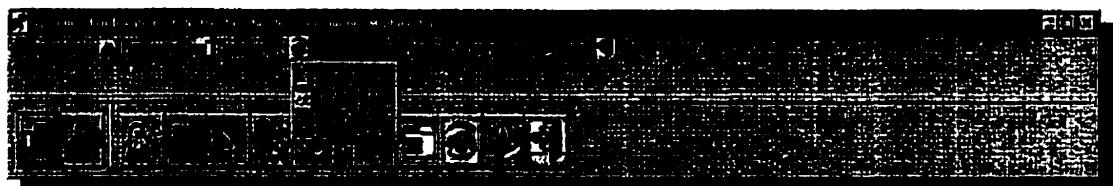


Figure 6.5 : Menu de gestion des visualiseurs

L'élément suivant s'intitule «*Moteur de recherche*» (voir figure numéro 6.6). Il permet à l'administrateur de gérer les différents moteurs de recherches d'Internet utilisés par l'application.



Figure 6.6 : Menu de gestion des moteurs de recherche sur Internet

L'élément suivant s'intitule «*Aide*» (voir figure numéro 6.7). Il permet à l'utilisateur d'accéder aux différentes formes d'aide de l'application (l'aide générale, la fenêtre des conseils du jour, les astuces, etc...)

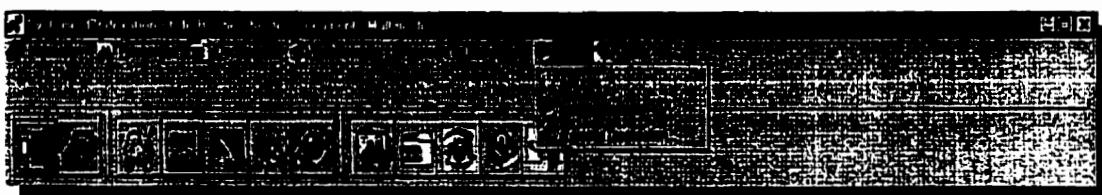


Figure 6.7 : Menu d'aide de l'application

Le dernier élément permet à l'utilisateur de quitter l'application. Il est intitulé «*Quitter*» (voir figure numéro 6.8).

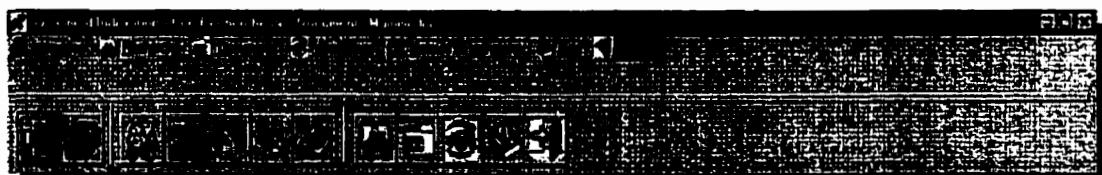


Figure 6.8 : Menu quitter l'application

b) Les éléments de la barre d'outils de l'application:

La barre d'outils comporte plusieurs groupes de boutons.

Le premier groupe comporte deux boutons :

-  Le premier bouton de ce groupe permet d'accéder aux ontologies du système pour les gérer.
-  Le deuxième bouton de ce groupe est le bouton d'indexation des documents.

Le deuxième groupe s'intéresse à l'aspect «recherche sur le réseau Internet», il comporte les boutons suivants:

-  Le bouton d'accès à la liste des moteurs de recherche sur lesquels le système se base pour lancer des requêtes de recherche sur le réseau Internet.
-  Le deuxième bouton de ce groupe est un raccourci vers le navigateur «Internet Explorer» pour lancer des moteurs de recherche ou des pages web.
-  Le bouton suivant est un raccourci vers le navigateur «Netscape» pour lancer des moteurs de recherche ou des pages web.
-  Ce bouton permet d'accéder à la fenêtre de recherche sur le web.

Le troisième groupe s'intéresse à l'aspect «recherche des documents de la base de données du système». Il contient les boutons suivants:



- Le bouton «recherche des documents». Lorsque l'utilisateur clique sur ce bouton, la fenêtre de recherche des documents dans la base de données s'affichera.



- Le bouton «liste des documents». Lorsque l'utilisateur clique sur ce bouton, la fenêtre qui permet d'afficher la liste des documents de la base de données du système s'affichera.



- Le bouton «liste des visualiseurs». En cliquant sur ce bouton, la fenêtre qui permet d'afficher la liste des visualiseurs stockés dans la base de données du système apparaît.

Un quatrième groupe s'intéresse à d'autres fonctionnalités de l'application. Il comporte les boutons suivants:



- Le bouton d'aide générale de l'application. Lorsque l'utilisateur clique sur ce bouton, il peut accéder au système d'aide général de l'application.



- Le bouton «quitter». En cliquant sur ce bouton, l'utilisateur quitte l'application.

Dans ce qui suit, nous présentons les interfaces personne-machine des différents modules de l'application. Nous commençons par présenter les interfaces du module d'indexation.

6- 3 – 2 - Les écrans du module d'indexation:

Dans cette section nous présentons les différentes interfaces du module d'indexation des documents multimédias.

Avant d'indexer un document, ce document doit exister dans la base de données du système pour y accéder et pour savoir quel sujet ce document traite afin de pouvoir l'indexer adéquatement.

Pour indexer un document, l'indexeur doit ajouter le document ou sa référence à la base de données du système. La fenêtre d'ajout du document ou de sa référence au système se présente dans la fenêtre de la figure numéro 6.9.

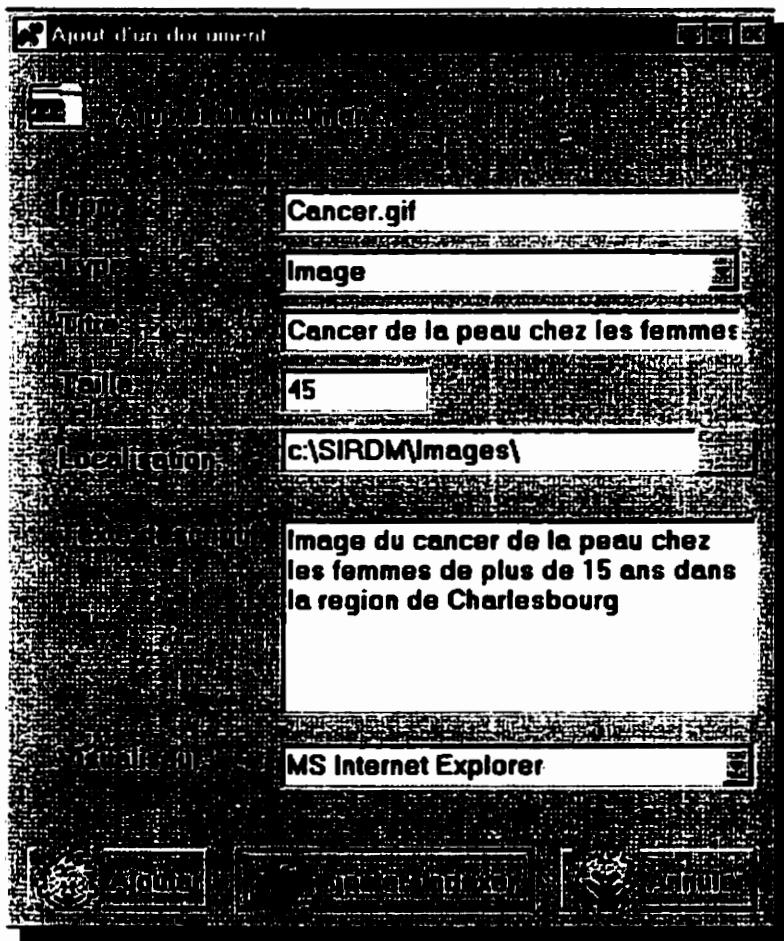


Figure 6.9 : La fenêtre d'ajout des documents

Dans cette fenêtre:

- ◆ Le bouton «Ajouter» permet d'ajouter un nouveau document dans la base du système, sans l'indexer.

- ◆ Le bouton «Ajouter+Indexer» permet d'ajouter un document à la base du système, puis il permet d'ouvrir la fenêtre d'indexation de la figure numéro 6.10, pour indexer ce document.
- Le bouton «Annuler» permet d'annuler l'indexation du document.

Pour aider l'indexeur à indexer le document par les concepts/expressions de l'ontologie du système, le système affiche à l'indexeur tous les concepts/expressions contenant les mots entrés par ce dernier. Par exemple, si l'indexeur veut indexer le document par le concept/expression « Cancer de la peau » il commence par taper le mot « Cancer », le système lui affiche immédiatement tous les concepts/expressions de l'ontologie qui contiennent ce mot. L'utilisateur peut sélectionner les concepts/expressions par lesquels il veut indexer son document et les ajouter dans la liste des concepts/expressions retenus pour l'indexation (voir figure numéro 6.10).

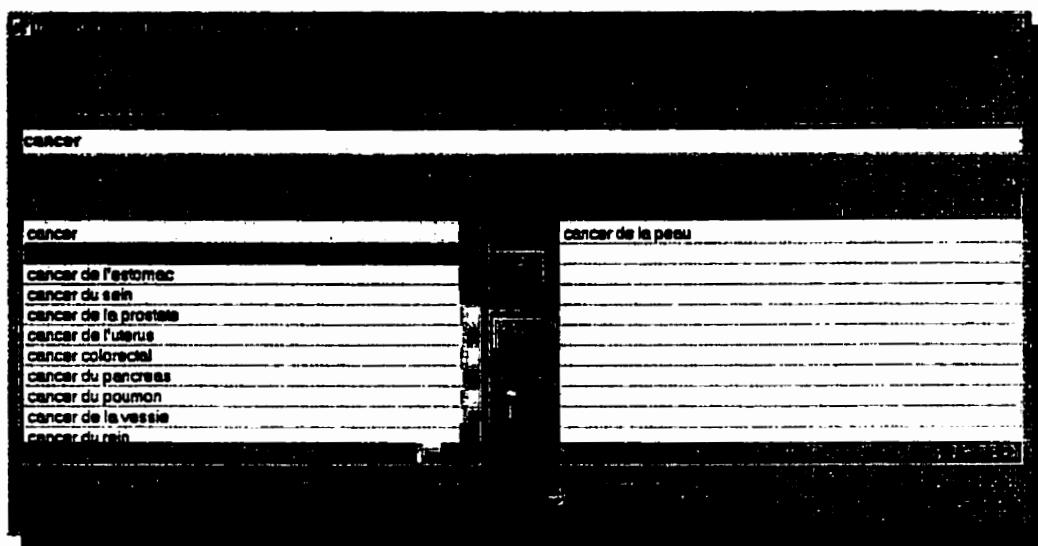


Figure 6.10 : Fenêtre d'indexation d'un document (1)

Dans cette fenêtre:

- ◆ Le bouton «Indexer» permet d'indexer le document en utilisant les concepts/expressions qui existent dans la liste des concepts/expressions retenus pour l'indexation.

◆ Le bouton «Annuler» permet d'annuler l'opération d'indexation du document. Il permet de fermer la fenêtre d'indexation.



◆ Le bouton qui contient l'icône permet d'ajouter l'expression sélectionnée dans la liste des expressions à la liste des concepts/expressions retenus pour l'indexation.



◆ Le bouton qui contient l'icône permet de retirer une expression de la liste des concepts/expressions retenus pour l'indexation.



◆ Le bouton qui contient l'icône permet de vider la liste des concepts/expressions retenus pour l'indexation.

L'indexeur peut indexer son document en utilisant autant de concepts/expressions qu'il veut. Une fois qu'il a sélectionné un ou plusieurs concepts/expressions d'indexation, il peut taper d'autres mots et d'autres concepts/expressions apparaissent au fur et à mesure. Cet utilisateur peut ajouter le nombre des concepts/expressions qu'il veut à la liste des concepts/expressions retenus pour l'indexation (voir figure numéro 6.11).

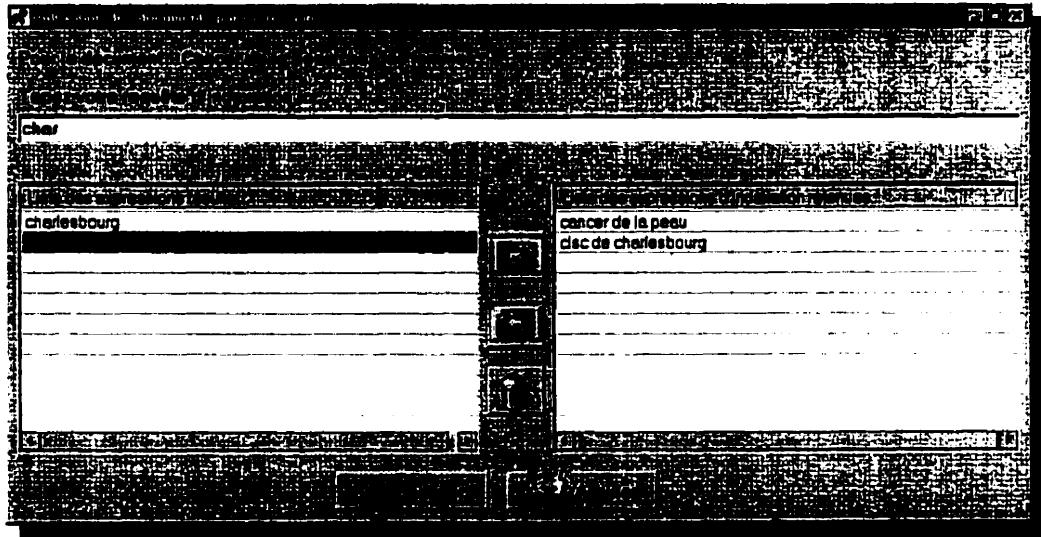


Figure 6.11 : Fenêtre d'indexation d'un document (2)

Une fois qu'il a fini, il peut cliquer sur le boutons «indexer» pour indexer le document par les concepts/expressions choisis, ou bien sur le bouton «annuler» pour annuler l'indexation.

6- 3 – 3 - Les écrans du module de recherche des documents:

a) La recherche classique des documents multimédias:

Si les documents sont indexés par les concepts/expressions de l'ontologie, les utilisateurs du système vont chercher ces documents en entrant des requêtes. La fenêtre de recherche des documents est présentée par la figure numéro 6.12:

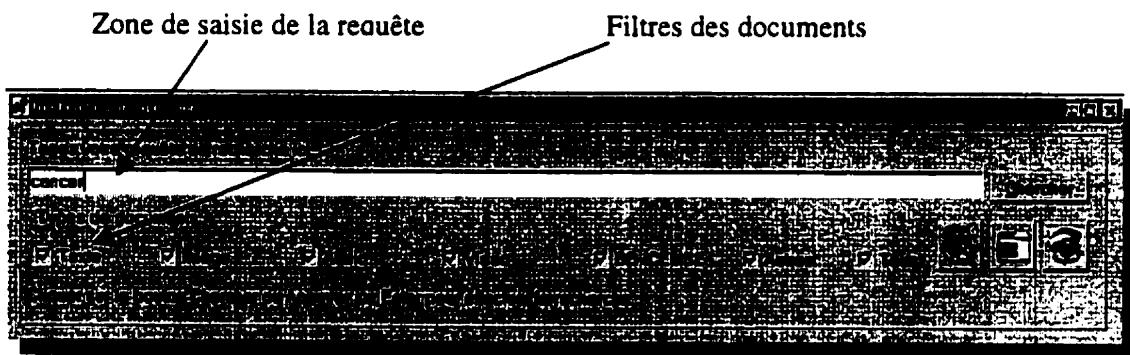


Figure 6.12: Fenêtre de recherche des documents

Dans cette fenêtre :

● Le bouton «Chercher» permet de lancer la recherche des documents dans la base de données du système.

● Le bouton qui contient l'icône permet d'afficher la fenêtre de recherche des documents sur le réseau Internet.

● Le bouton qui contient l'icône permet d'afficher la fenêtre qui présente la liste des documents stockés dans la base de données du système.

● Le bouton qui contient l'icône permet d'afficher la fenêtre qui présente la liste des visualiseurs du système.

Dans cette fenêtre, l'utilisateur entre sa requête dans la zone de saisie correspondante, puis il clique sur le bouton «Chercher». Le système effectue sa recherche dans la base d'index du système et présente tous les documents qui répondent à la requête fournie par cet utilisateur. Par exemple, si l'utilisateur tape le mot «Cancer» il aura en résultat 50 documents qui répondent à sa requête (voir figure numéro 6.13).

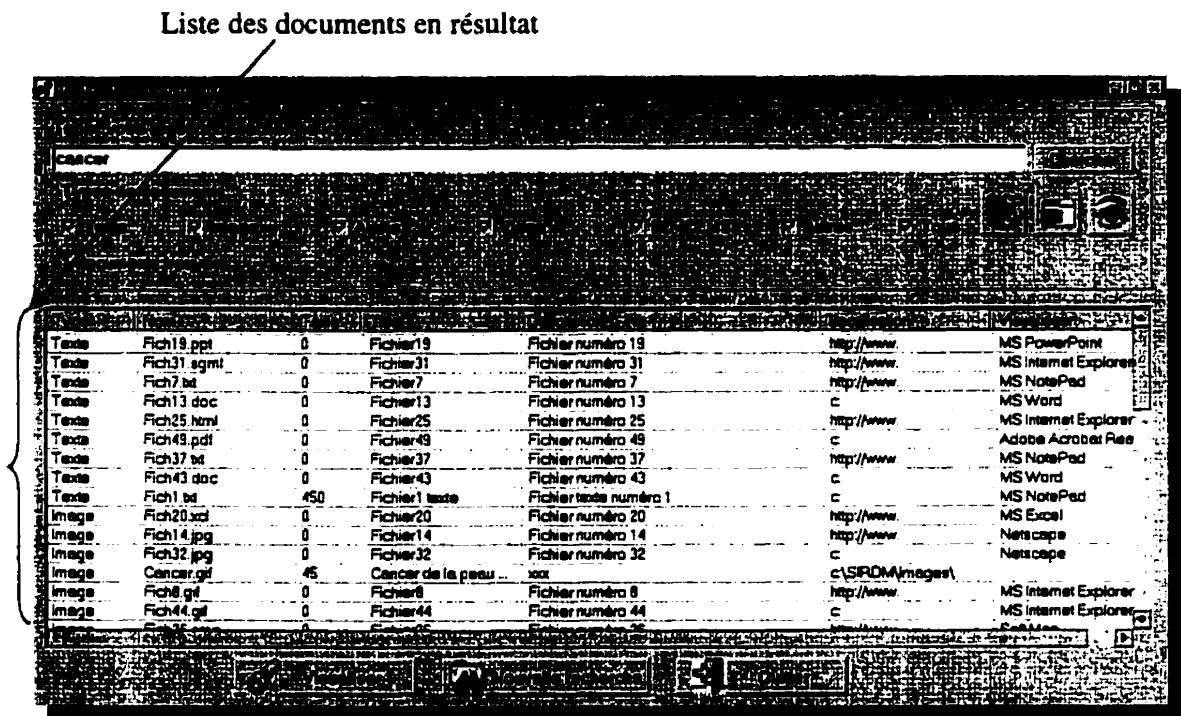


Figure 6.13 : Fenêtre de présentation des documents en résultat

Dans cette fenêtre:

- Le bouton «Visualiser» permet de visualiser le document sélectionné par l'utilisateur dans la liste des documents-résultat en utilisant le visualiseur correspondant.
- Le bouton «Nouvelle Recherche» permet d'effectuer une nouvelle recherche.
- Le bouton «Quitter» permet de quitter l'application.

Le système indexe et cherche les documents en se basant sur les ontologies des concepts/expressions et des relations. Il utilise les concepts/expressions de l'ontologie pour chercher les documents qui répondent exactement à la requête de l'utilisateur. Plus que l'utilisateur spécifie sa requête, plus le nombre de documents en résultat est restreint. Par exemple, dans la figure numéro 6.14, si l'utilisateur entre la requête «Cancer de la peau» il aura en résultat 17 documents au lieu de 50 documents obtenus pour la requête «Cancer» dans le cas présenté dans la figure numéro 6.13.

The screenshot shows a Windows-style application window titled "Recherche de la peau sur la peau". The main area displays a table of search results:

| Type | Nom du document | Taille | Nom du fichier | Description | Protocole | Application |
|--------------|-----------------|--------|----------------|-------------------|-------------|----------------------|
| Texte | Fich25.html | 0 | Fichier25 | Fichier numéro 25 | http://www. | MS Internet Explorer |
| Image | Fich14.jpg | 0 | Fichier14 | Fichier numéro 14 | http://www. | Netscap |
| Image | Fich26.bmp | 0 | Fichier26 | Fichier numéro 26 | http://www. | Soft Map |
| Audio | Fich31.wav | 0 | Fichier33 | Fichier numéro 33 | http://www. | MS Media Player |
| Audio | Fich39.wav | 0 | Fichier39 | Fichier numéro 39 | c | MS Media Player |
| Video | Fich28.avi | 0 | Fichier28 | Fichier numéro 28 | c | MS Media Player |
| Video | Fich34.avi | 0 | Fichier34 | Fichier numéro 34 | http://www. | MS Media Player |
| Base de d... | Fich11.bd | 0 | Fichier11 | Fichier numéro 11 | c | |
| Base de d... | Fich35.bd | 0 | Fichier35 | Fichier numéro 35 | c | |

Figure 6.14 : Fenêtre de recherche (Requête plus spécifique)

En plus de la fonction de recherche des documents, le système fournit à l'utilisateur une fonction de filtrage des documents présentés en résultat. Ce filtrage est effectué en se basant sur le type des documents demandés. Par exemple, dans la figure numéro 6.15, si l'utilisateur entre la requête «Cancer» et filtre les documents de type Texte, nous trouvons 41 documents au lieu de 50 de la fenêtre présentée dans la figure numéro 6.13.

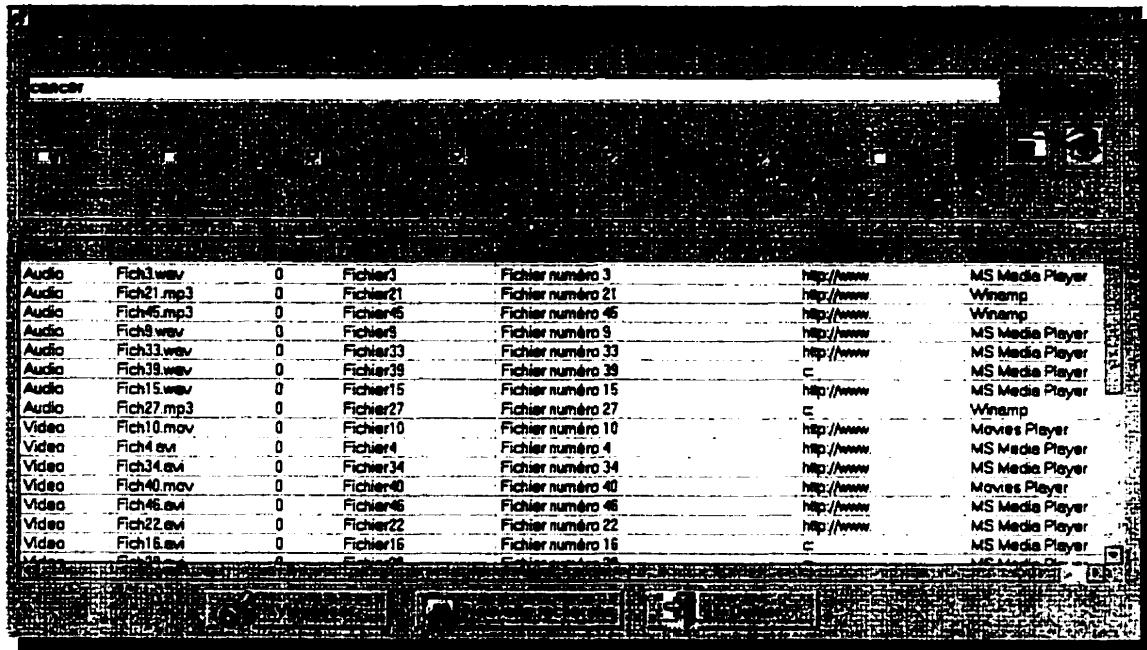


Figure 6.15 : Fenêtre de recherche (Filtrage du résultat)

Parfois, il arrive que l'utilisateur ne trouve pas de document qui répond à sa requête. Dans le cas, nous disons que le résultat est négatif. Alors la fenêtre de la figure numéro 6.16 s'affichera:

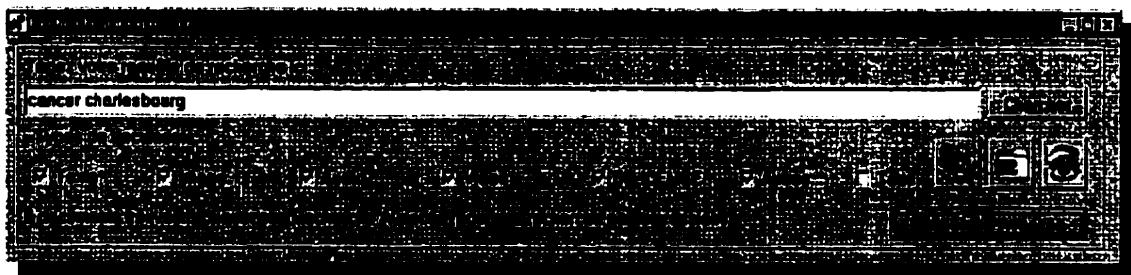


Figure 6.16 : Fenêtre de recherche (Résultat négatif)

Dans le cas de résultat négatif, le système donne la main à l'utilisateur pour effectuer un autre type de recherche intitulé «recherche avancée».

Dans la figure numéro 6.16, le bouton «Recherche avancée» permet d'afficher la fenêtre de recherche avancée du système. Le principe de la recherche avancée est présenté en détail dans la section suivante.

b) La recherche avancée des documents multimédias:

La fenêtre de recherche avancée est présentée dans la figure numéro 6.17. Lorsque cette fenêtre est activée à partir de la fenêtre de recherche classique, la zone de la requête contient la requête déjà entrée par l'utilisateur dans la fenêtre de recherche classique.

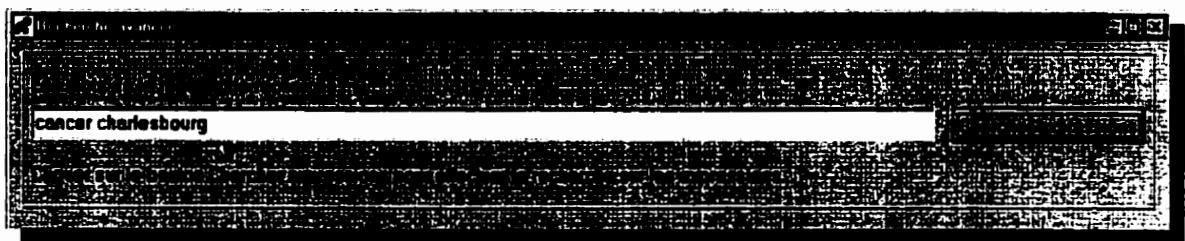


Figure 6.17 : Fenêtre de recherche avancée

Dans cette fenêtre, l'utilisateur clique sur le bouton «Chercher expressions» pour chercher les concepts/expressions qui peuvent être formés à partir des mots de la requête posée (voir figure numéro 6.18).

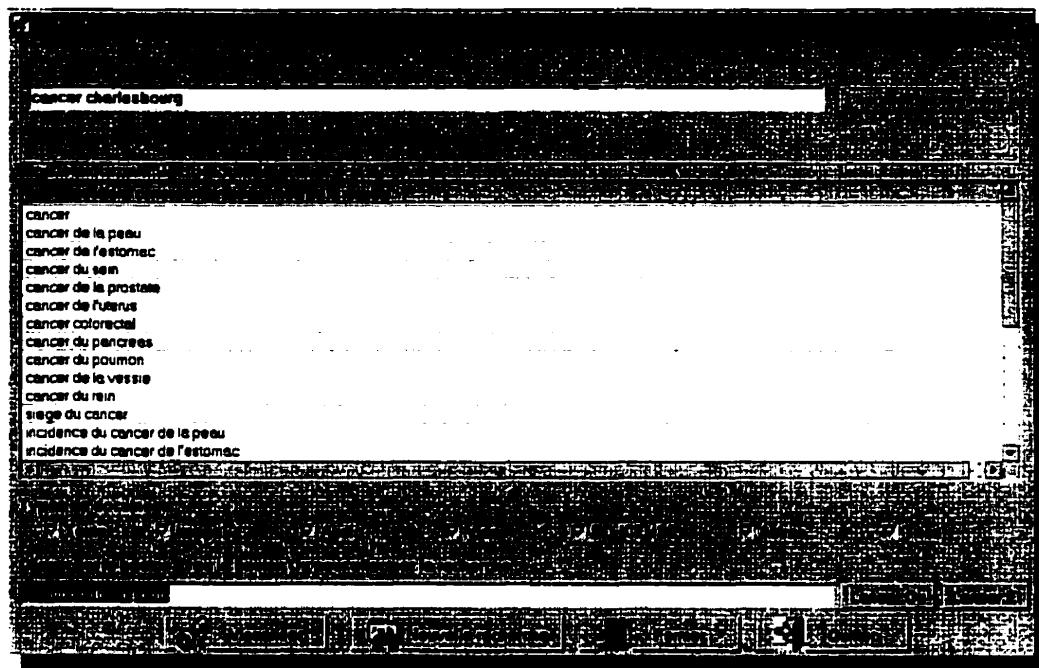


Figure 6.18: Fenêtre de recherche avancée (Les expressions correspondantes à la requête (1))

Dans cette fenêtre, pour la requête «cancer charlesbourg» tous les concepts/expressions comportant «cancer» et tous les concepts/expressions comportant «charlesbourg» vont apparaître.

L'utilisateur sélectionne les expressions de l'ontologie pour reformer sa requête de recherche en bas de la fenêtre. Il clique sur les concepts/expressions qu'il désire ajouter et ces derniers s'ajoute dans la zone de texte de la requête formulée (voir [figure numéro 6.18](#)). L'utilisateur peut sélectionner autant de concepts/expressions qu'il veut (voir [figure numéro 6.19](#)).

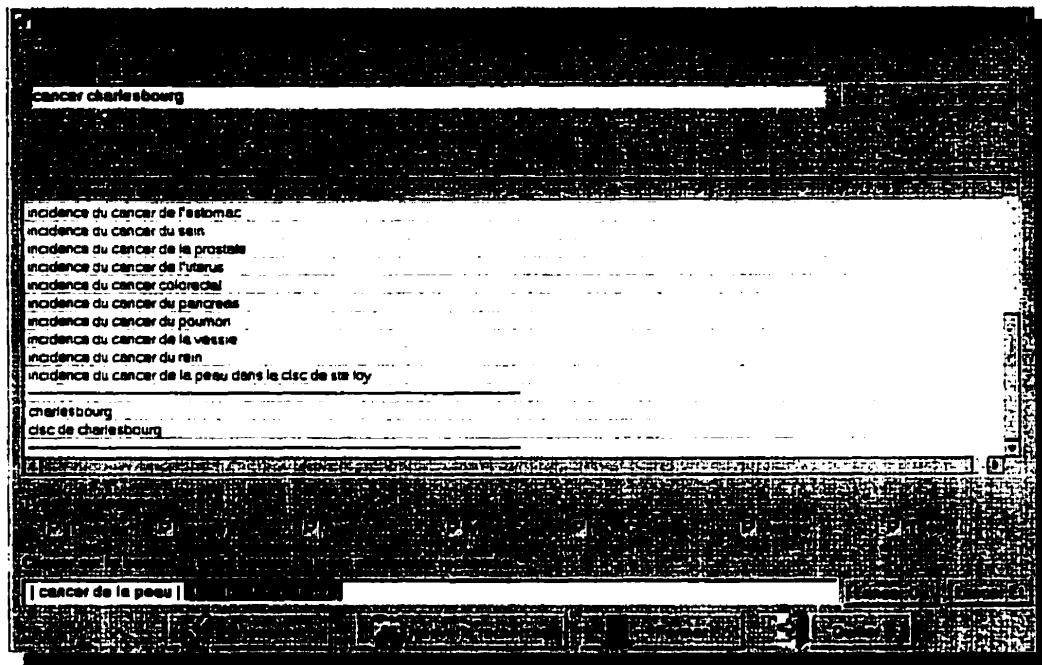


Figure 6.19: Fenêtre de recherche avancée (Les expressions correspondantes à la requête (2))

Une fois qu'il a sélectionné les concepts/expressions pertinents pour reformuler sa requête, l'utilisateur a deux choix : Soit il cherche les documents indexés par tous les concept/expressions à la fois (opérateur ET (And) (voir figure numéro 6.20), soit il veut chercher tous les documents indexés par l'un des concepts/expressions sélectionnés (opérateur OU (Or) (voir la figure numéro 6.21).

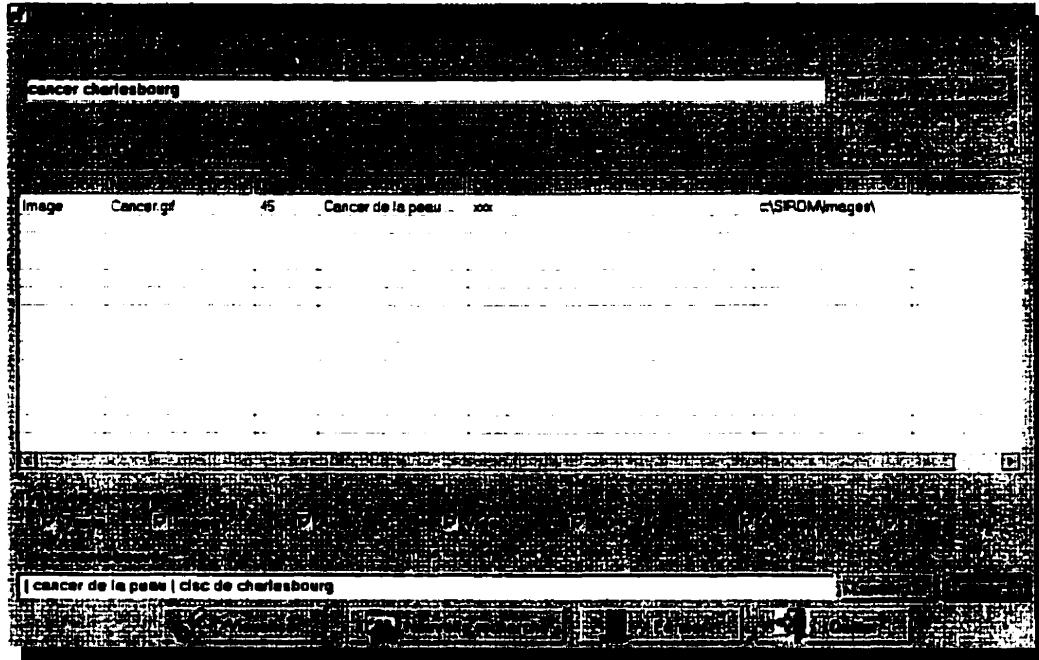


Figure 6.20 : Fenêtre de recherche avancée (Présentation des documents : Recherche AND)

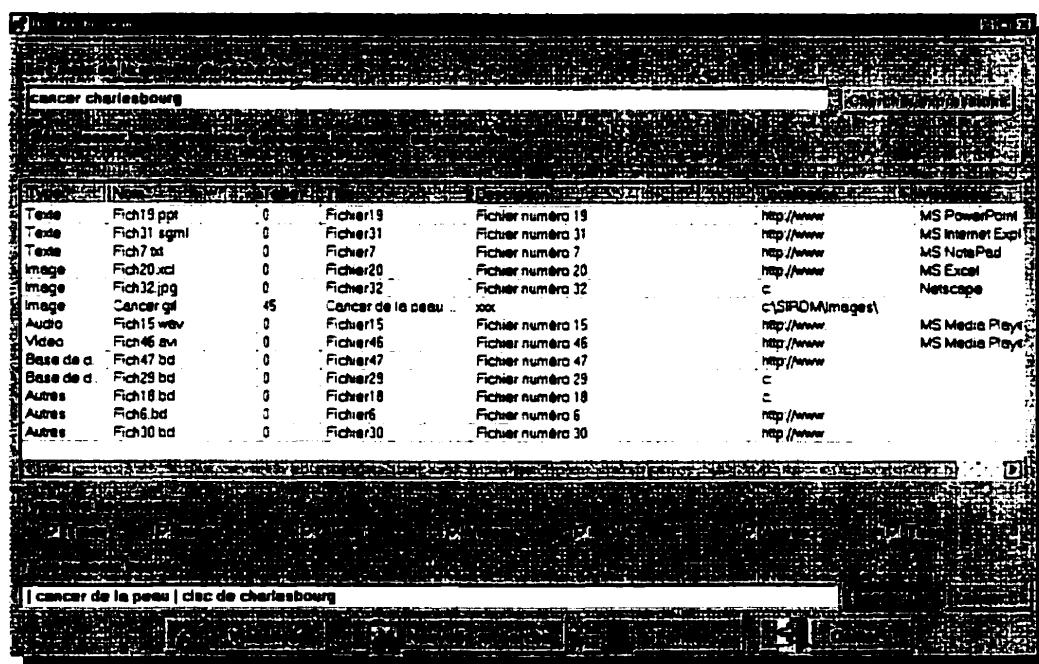


Figure 6.21 : Fenêtre de recherche avancée (Présentation des documents : Recherche OR)

Dans le cas de la recherche avancée, l'utilisateur dispose toujours de la fonction de filtrage des documents en résultat. Dans la figure numéro 6.22, nous présentons le résultats de la figure numéro 6.21 filtré sur les documents textuels.

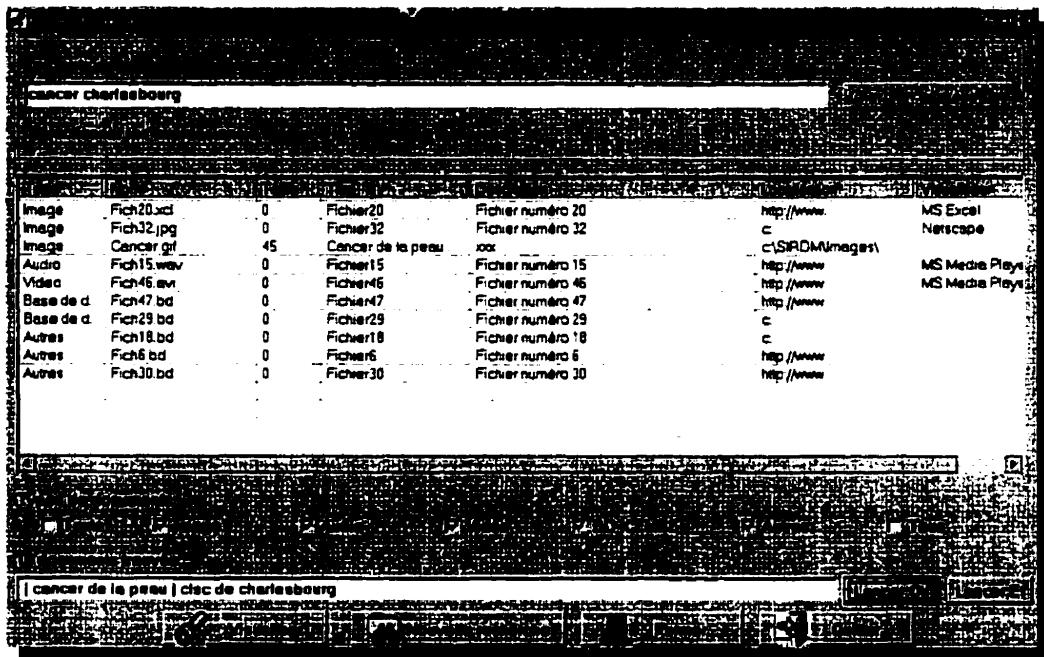


Figure 6.22 : Fenêtre de recherche avancée (Présentation des documents :
Recherche OR : Avec filtrage)

Si le résultat de recherche est négatif en utilisant la recherche avancée, le système présente la fenêtre numéro 6.23 pour donner la main à l'utilisateur pour retourner à la forme de recherche classique.

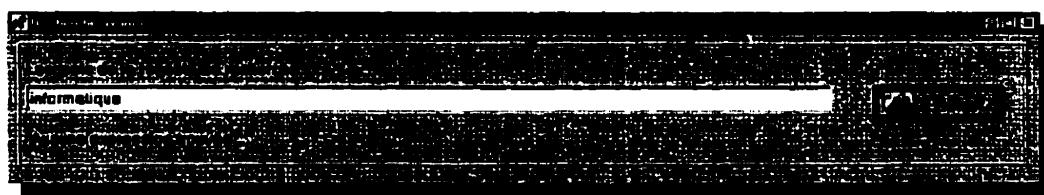


Figure 6.23: Fenêtre de recherche avancée (Résultat négatif)

Le système d'indexation et de recherche des documents multimédias ne cherche pas les documents dans sa base de données seulement, mais il peut chercher aussi les documents sur le réseau Internet. La recherche ne se fait pas directement sur le réseau, mais par l'intermédiaire des moteurs de recherche sur Internet.

c) La recherche des documents sur Internet:

La fenêtre de recherche sur le réseau Internet est présentée dans la figure numéro 6.24 :

L'utilisateur entre sa requête dans la zone de saisie spécifique, et sélectionne le moteur de recherche qu'il veut lancer avec la requête. Les références vers les moteurs de recherche présentés dans la liste dans cette fenêtre sont stockés dans la base de donnée du système.

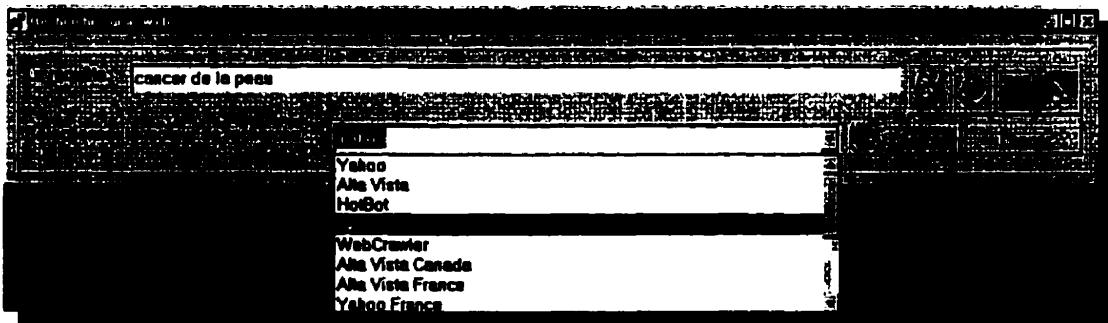


Figure 6.24 : Recherche sur Internet

Dans cette fenêtre :

- ◆ Le bouton qui contient l'icône : permet à l'utilisateur d'accéder à la fenêtre qui présente la liste des moteurs de recherche existant dans la base de données du système.
- ◆ Le bouton qui contient l'icône : Permet de lancer le navigateur «*Internet Explorer*» pour effectuer une recherche sur Internet indépendante du système.
- ◆ Le bouton qui contient l'icône : Permet de lancer le navigateur «*Netscape*» pour effectuer une recherche sur Internet indépendante du système.

Si l'utilisateur entre sa requête, sélectionne son moteur de recherche et clique sur le bouton «Chercher», la page web qui présente le résultat du moteur avec la requête en paramètre apparaîtra. A titre d'exemple, la figure numéro 6.25 présente les résultats présenté par le moteur «*Lycos*» avec la requête «Cancer de la peau».

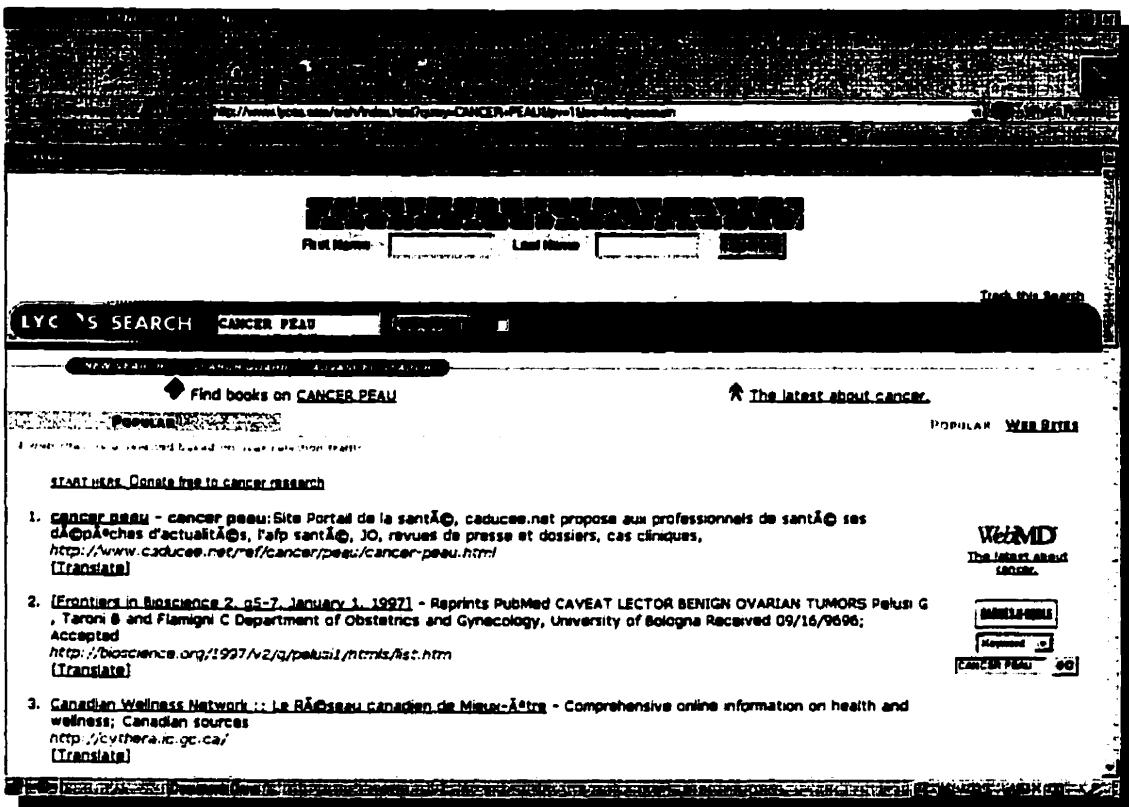


Figure 6.25 : Résultats de la recherche sur Internet

L'utilisateur peut lancer la page web principale du moteur de recherche sélectionné en cliquant sur le bouton «Lancer».

Le système offre aussi à l'utilisateur une recherche sur Internet par type de documents, en se basant sur les types de groupements des moteurs de recherches présentés dans le cinquième chapitre. Le système présente à l'utilisateur une liste des catégories de moteurs de recherche. L'utilisateur sélectionne une catégorie dans cette liste et la liste des moteurs de recherche de cette catégorie apparaissent. L'utilisateur

peut sélectionner un moteur et le lancer avec une requête qu'il tape dans la zone de saisie spécifique. A titre d'exemples, la figure numéro 6.26 présente la liste de moteurs de recherche sous la catégorie «web francophone» et la figure numéro 6.27 présente la liste de moteurs de recherche sous la catégorie «logiciel».

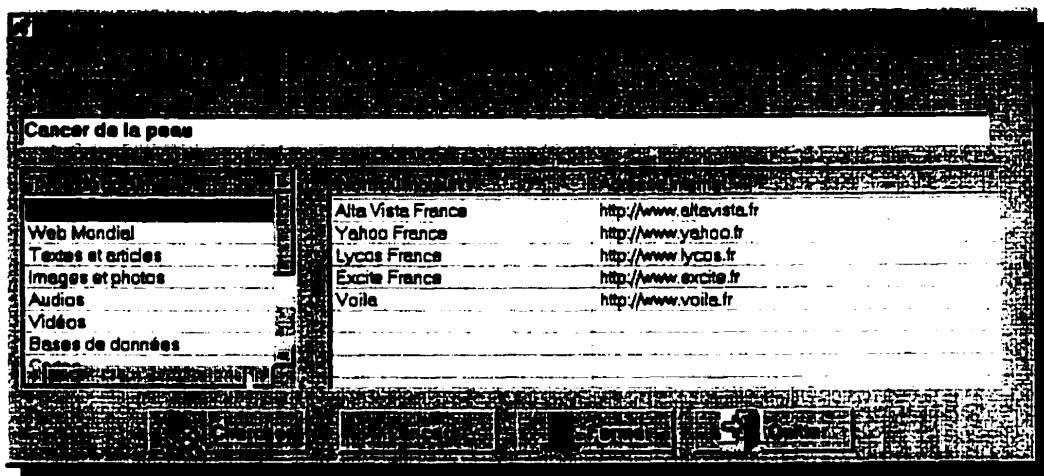


Figure 6.26 : Recherche sur Internet par catégorie de moteurs de recherche

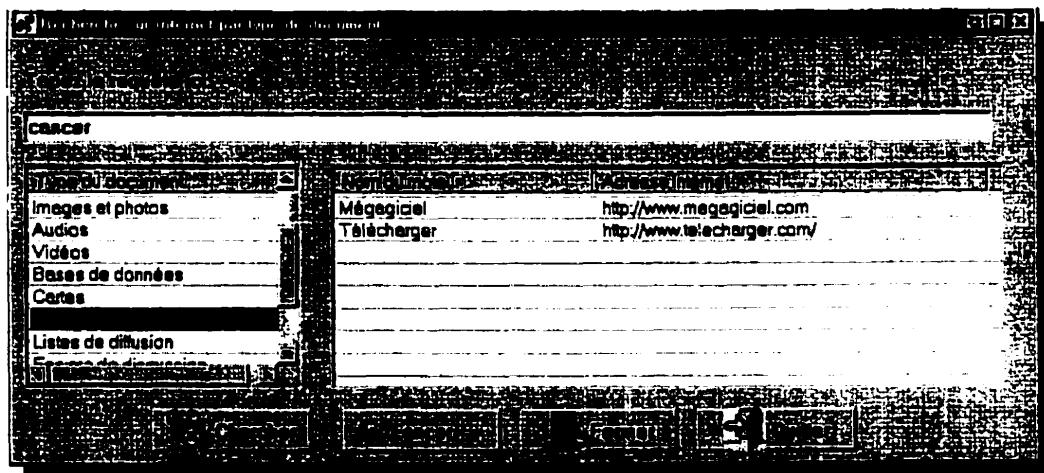


Figure 6.27 : Recherche sur Internet par catégorie de moteurs de recherche

6- 3 – 4 - Les écrans du module de gestion de la base de données du système:

Le système utilise une base de données *Access* de la compagnie *MicroSoft*. Cette base de données comporte plusieurs tables servant à stocker les données manipulées par l'application. Pour gérer cette base de données, nous avons conçu et réalisé toutes les fenêtres et les fonctionnalités nécessaires. Dans ce qui suit, nous présentons quelques une de ces fenêtres. La table la plus importante de la base de données est la table des documents, la fenêtre de la figure numéro 6.28 présente la fenêtre principale de gestion des documents. Cette fenêtre présente la forme à travers laquelle l'utilisateur peut consulter et accéder aux différentes fenêtres de gestion des documents.

| Type | Nom | Taille | Description | Emplacement | Application |
|------------|------------|--------|--------------------|------------------------|----------------------------|
| Texte | Fich1.txt | 450 | Fichier1 texte | Fichier texte numero 1 | c MS NotePad |
| Image | Fich2.bmp | 500 | Fichier2 image bmp | Fichier image numero 2 | http://www MS Paint |
| Audio | Fich3.wav | 0 | Fichier3 | Fichier numero 3 | http://www MS Media Player |
| Video | Fich4.avi | 0 | Fichier4 | Fichier numero 4 | http://www MS Media Player |
| Base de d. | Fich5.bd | 0 | Fichier5 | Fichier numero 5 | http://www |
| Autres | Fich6.xls | 0 | Fichier6 | Fichier numero 6 | http://www |
| Texte | Fich7.txt | 0 | Fichier7 | Fichier numero 7 | http://www MS NotePad |
| Image | Fich8.gif | 0 | Fichier8 | Fichier numero 8 | http://www MS Internet E |
| Audio | Fich9.wav | 0 | Fichier9 | Fichier numero 9 | http://www MS Media Player |
| Video | Fich10.mov | 0 | Ficher10 | Ficher numero 10 | http://www MS Media Player |
| Base de d. | Fich11.bd | 0 | Ficher11 | Ficher numero 11 | c Moves Plays |
| Autres | Fich12.xls | 0 | Ficher12 | Ficher numero 12 | c |
| Texte | Fich13.doc | 0 | Ficher13 | Ficher numero 13 | c MS Word |
| Image | Fich14.jpg | 0 | Ficher14 | Ficher numero 14 | http://www Netscape |

Figure 6.28 : Fenêtre de la liste des documents

Dans cette fenêtre :

- ◆ Le bouton qui contient l'icône permet d'afficher les documents de type texte dans la liste des documents.
- ◆ Le bouton qui contient l'icône permet d'afficher les documents de type image dans la liste des documents.
- ◆ Le bouton qui contient l'icône permet d'afficher les documents de type audio dans la liste des documents.

- ♦ Le bouton qui contient l'icône permet d'afficher les documents de type vidéo dans la liste des documents.
- ♦ Le bouton qui contient l'icône permet d'afficher les documents de type base de données dans la liste des documents.
- ♦ Le bouton qui contient l'icône permet d'afficher les documents de tous les types dans la liste des documents.

Par exemple, si l'utilisateur veut visualiser un type bien déterminé de documents (documents vidéos), il peut cliquer sur le bouton comportant l'icône vidéo et tous les documents vidéo apparaissent dans la liste (voir figure numéro 6.29).

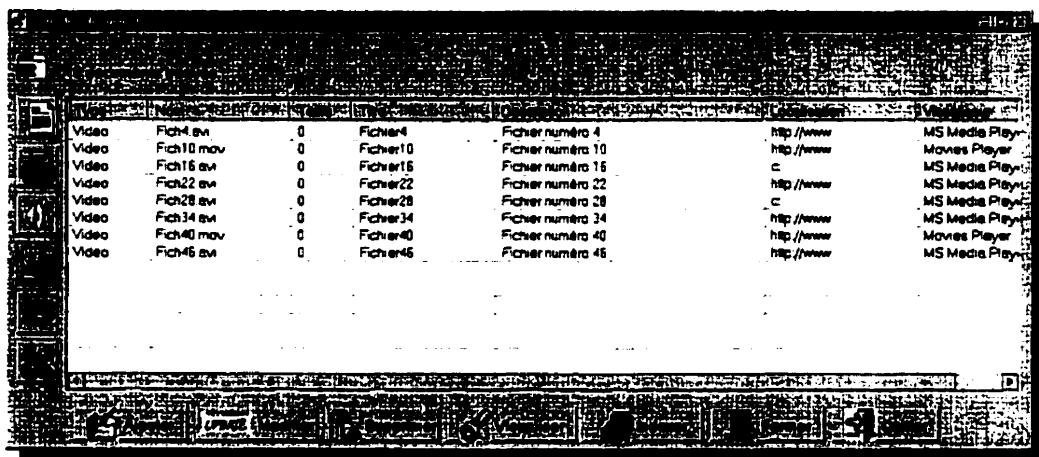


Figure 6.29 : Fenêtre de la liste des documents filtrés

Dans cette fenêtre :

- ♦ Le bouton qui contient l'icône permet d'ajouter un nouveau document dans la base de données.
- Le bouton qui contient l'icône permet de modifier le document sélectionné dans la liste des documents de cette fenêtre.
- ♦ Le bouton qui contient l'icône permet de supprimer le document sélectionné dans la liste des documents de cette fenêtre.

● Le bouton qui contient l'icône  permet de visualiser le document sélectionné dans la liste des documents de cette fenêtre.

◆ Le bouton qui contient l'icône  permet d'indexer le document sélectionné dans la liste des documents de cette fenêtre.

◆ Le bouton qui contient l'icône  permet de fermer la fenêtre de présentation des documents.

◆ Le bouton qui contient l'icône  permet de quitter l'application.

Gérer les documents de la base de données veut dire ajouter, modifier ou supprimer un ou plusieurs documents de la bse de données. La fenêtre qui permet d'ajouter un nouveau document à la base se présente dans la fenêtre numéro 6.30.

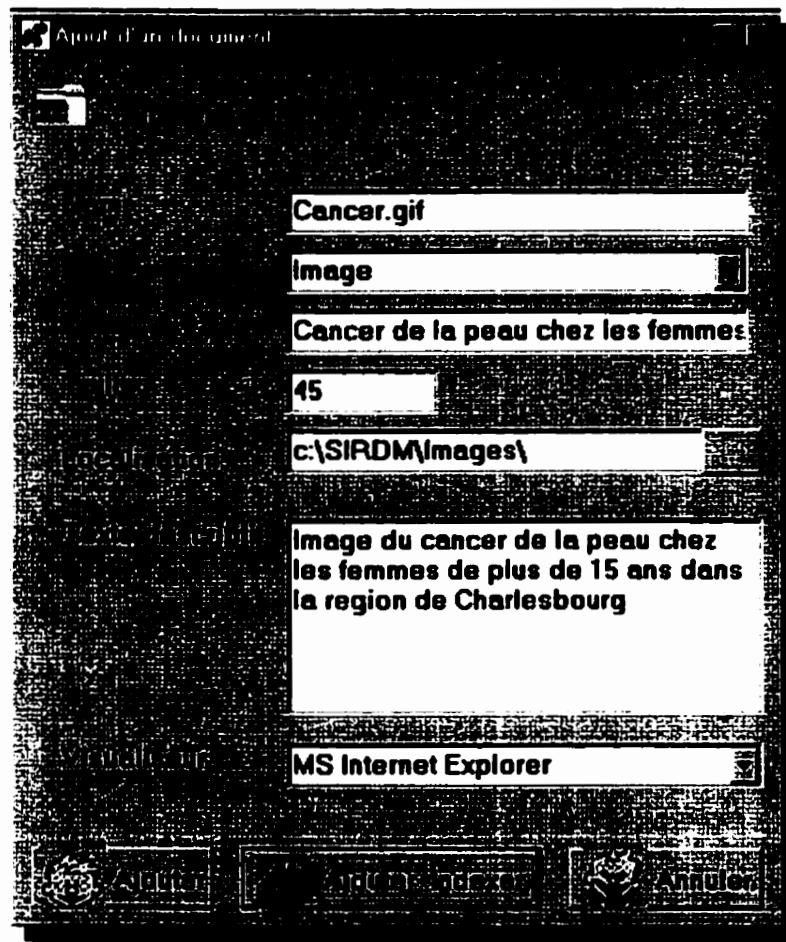


Figure 6.30 : Ajout d'un document au système

A travers cette fenêtre l'utilisateur peut ajouter, ou ajouter ou indexer un nouveau document.

Le système permet aussi de modifier et de supprimer un ou plusieurs documents de la base de données du système.

Comme le cas des documents, le système permet d'afficher la liste des visualiseurs du système aussi (Voir figure numéro 6.31).

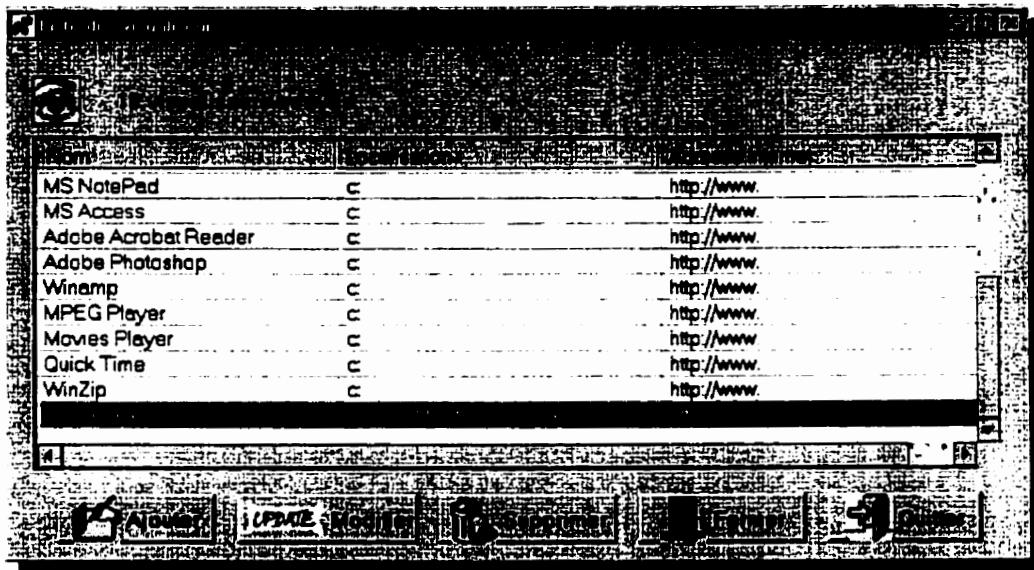


Figure 6.31 : Fenêtre de la liste des visualiseurs mise à jour

Dans cette fenêtre :

- ◆ Le bouton qui contient l'icône permet d'accéder à la fenêtre d'ajout d'un nouveau visualiseurs à la base de données du système.
- ◆ Le bouton qui contient l'icône permet de modifier le visualiseur sélectionné dans la liste de visualiseurs de cette forme.
- ◆ Le bouton qui contient l'icône permet de supprimer le visualiseur sélectionné dans la liste des visualiseurs de cette forme.
- ◆ Le bouton qui contient l'icône permet de fermer cette fenêtre.
- ◆ Le bouton qui contient l'icône permet de quitter l'application.

La fenêtre présentée dans la [figure numéro 6.32](#) permet d'ajouter un visualiseur à la liste des visualiseurs du système.

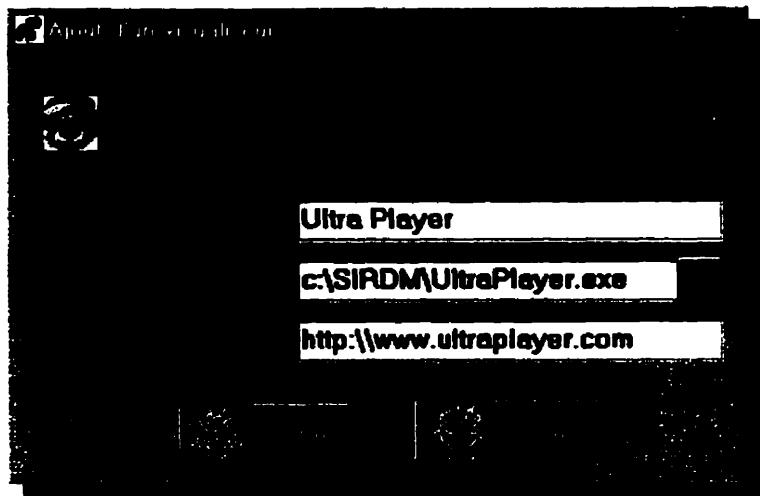


Figure 6.32 : Ajout d'un visualiseur

Dans cette forme, le bouton «Ajouter» permet l'ajout du visualiseur à la base de données du système et le bouton «Annuler» permet d'annuler l'ajout.

Le système permet de modifier ou de supprimer un ou plusieurs visualiseurs de sa base de données.

Comme le cas des documents et des visusliseurs, le système permet de gérer la table des moteurs de recherche qui sert pour la recherche dans le réseau Internet. La fenêtre numéro 6.33 permet d'afficher la liste de ces moteurs de recherche.

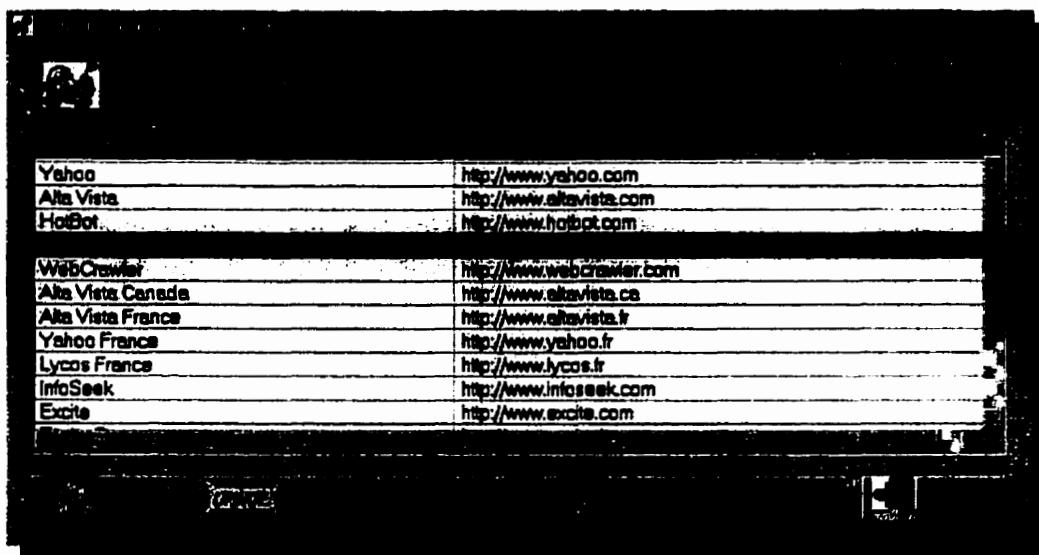


Figure 6.33 : Liste des moteurs de recherche

Dans cette fenêtre :

- ◆ Le bouton qui contient l'icône permet d'afficher la fenêtre d'ajout d'un nouvel enregistrement la table des moteurs de recherche à la base de données du système.
- ◆ Le bouton qui contient l'icône permet de modifier le moteur de recherche sélectionné dans la liste des moteurs de recherche de cette fenêtre.
- ◆ Le bouton qui contient l'icône permet de supprimer le moteur de recherche sélectionné dans la liste des moteurs de recherche de cette forme.
- ◆ Le bouton qui contient l'icône permet de fermer cette fenêtre.
- ◆ Le bouton qui contient l'icône permet de quitter l'application.

6- 3 – 5 - Les écrans du module d'administration des ontologies du système:

Ce module permet à l'administrateur de gérer les deux ontologies du système, à savoir l'ontologie des concepts/expressions et l'ontologie des relations.

Pour pouvoir gérer les ontologies du système, l'administrateur doit avoir un accès privilégié. Pour cette raison, lorsqu'il clique sur le bouton d'administration des ontologie, la fenêtre de «Login» de la figure numéro 6.34 s'affichera en demandant d'introduire un identifiant et un mot de passe administrateur.

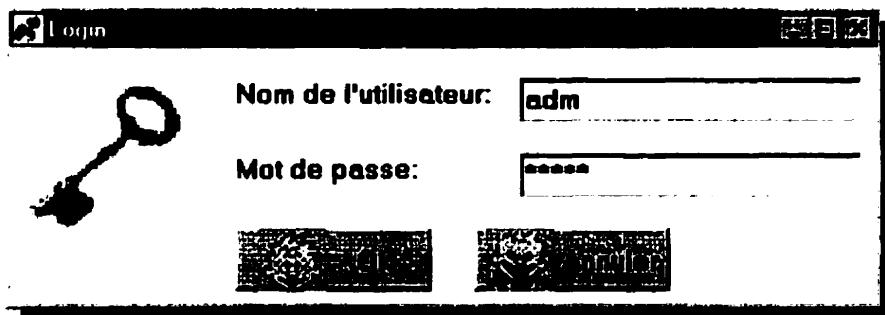


Figure 6.34 : Fenêtre de Login de l'administrateur de l'ontologie

Si l'identifiant et le mot de passe sont exacts, les fenêtres d'administration des deux ontologies peuvent être affichées, sinon un message d'erreur s'affichera. La première fenêtre qui s'affiche est la fenêtre de la figure numéro 6.35 qui affiche l'ontologie des concepts/expressions.

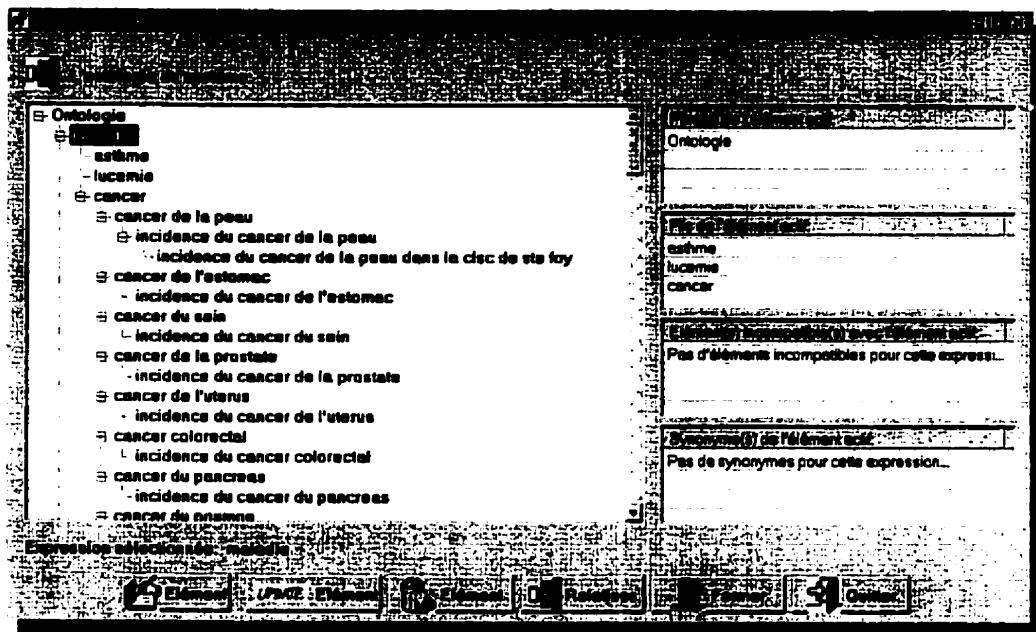


Figure 6.35 : Fenêtre d'affichage de l'ontologie du système

Dans cette fenêtre :

- ◆ Le bouton qui contient l'icône permet d'accéder à la fenêtre d'ajout d'un nouveau concept/expression pour ajouter un nouveau concept/expression à l'ontologie du système.
- ◆ Le bouton qui contient l'icône permet de modifier le concept/expression sélectionné dans cette fenêtre.
- ◆ Le bouton qui contient l'icône permet de supprimer le concept/expression sélectionné dans cette fenêtre.
- ◆ Le bouton qui contient l'icône permet d'accéder à la fenêtre de présentation de l'ontologie des relations du système.
- ◆ Le bouton qui contient l'icône permet de quitter la fenêtre.
- ◆ Le bouton qui contient l'icône permet de quitter l'application.

Lorsque l'utilisateur sélectionne l'un des concepts/expressions la fenêtre affichera à côté la liste des pères, des fils, des éléments incompatibles et des synonymes du concept/expression en question (voir figure numéro 6.35).

A travers ce module l'administrateur peut ajouter un nouveau concept ou une nouvelle expression à l'ontologie des concepts/expressions. La fenêtre d'ajout est présentée dans la figure numéro 6.36.

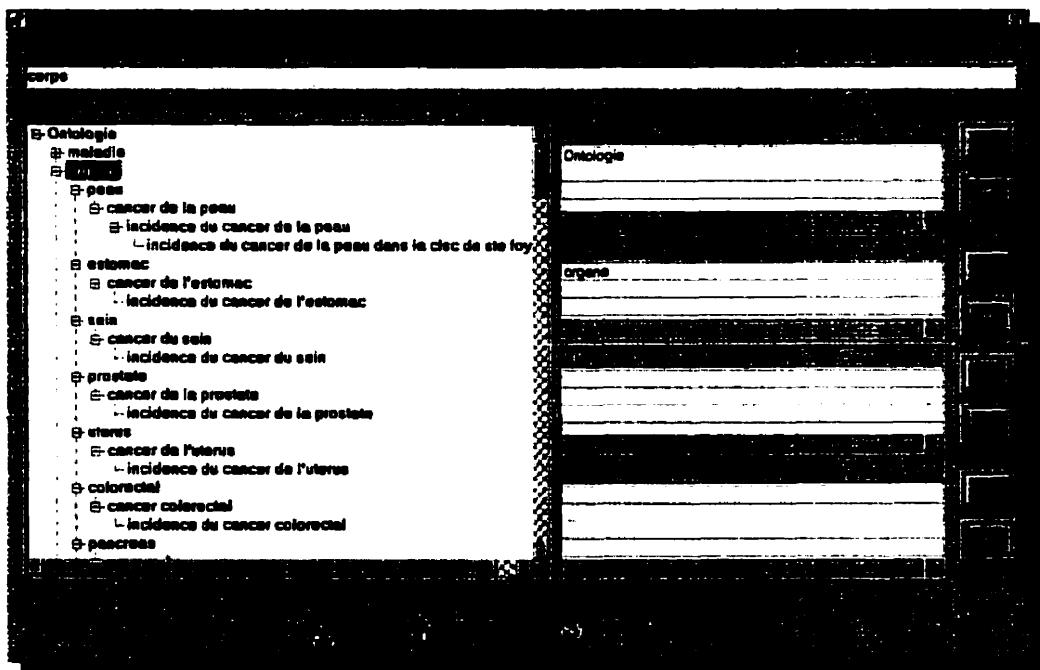


Figure 6.36 : Fenêtre d'ajout d'un concept/expression à l'ontologie du système

Dans cette fenêtre :

- Le bouton qui contient l'icône permet d'ajouter un concept/expression à une liste (liste des pères, des fils, des incompatibles ou synonymes).
- ◆ Le bouton qui contient l'icône permet de supprimer un concept/expression dans une liste (liste des pères, des fils, des incompatibles ou synonymes).

Si l'administrateur ajoute un nouveau concept/expression et le relie avec les quatre relations de base, le concept/expression s'ajoute immédiatement à l'ontologie, alors il paraîtra lorsque la fenêtre d'affichage de l'ontologie des concepts/expressions sera activée (Voir figure numéro 6.37).

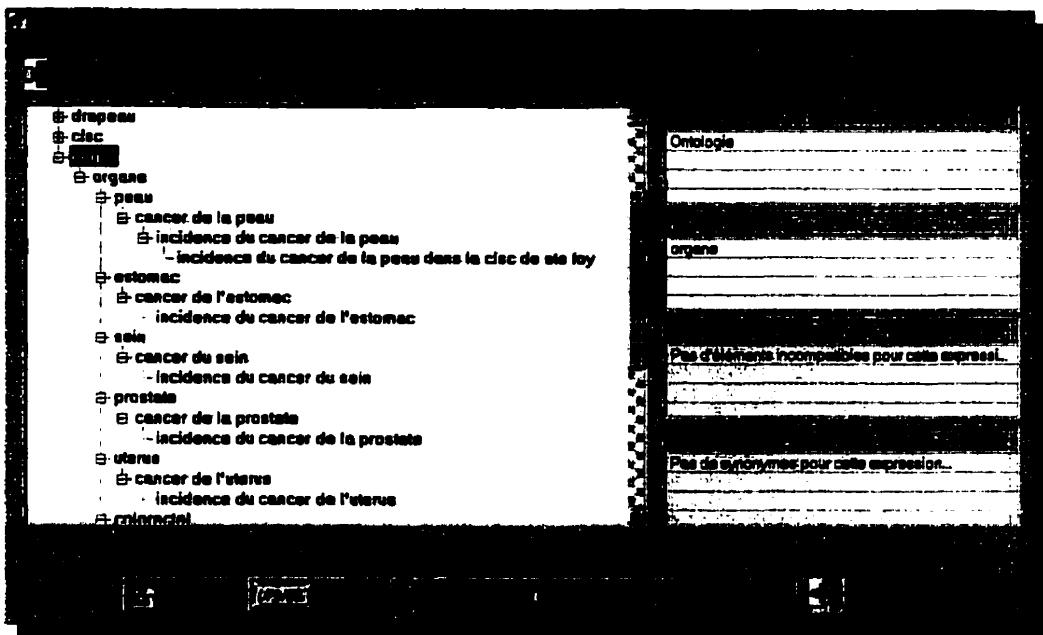


Figure 6.37 : Fenêtre de présentation de l'ontologie après mise à jour

Le système permet aussi de modifier ou de supprimer un concept/expression de l'ontologie des concepts/expressions en utilisant les boutons correspondants après avoir sélectionné ce concept/expression dans la hiérarchie.

Le système se base sur deux types d'ontologies, une ontologie de concepts/expressions et une ontologie de relations. Il permet aussi de présenter à l'administrateur l'ontologie des relations. L'affichage de l'ontologie des relations se fait à l'aide de la fenêtre numéro 6.38:

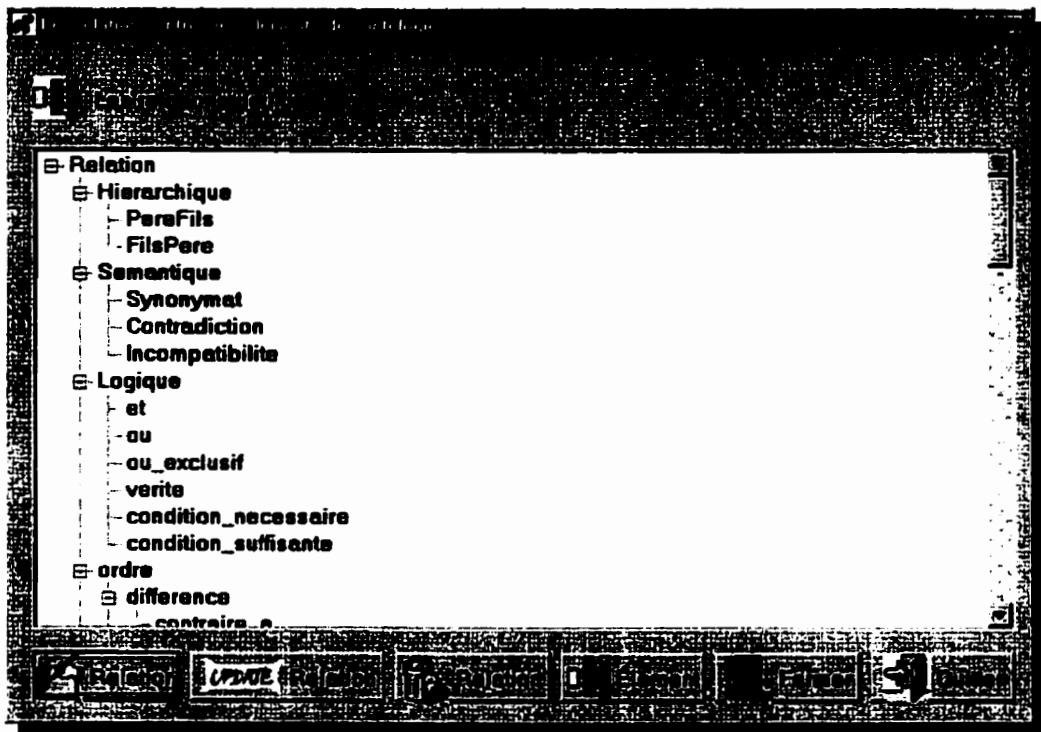


Figure 6.38 : Fenêtre d'affichage des relations de l'ontologie

Dans cette fenêtre :

- ◆ Le bouton qui contient l'icône permet d'accéder à la fenêtre d'ajout d'une relation à l'ontologie des relations du système.
- ◆ Le bouton qui contient l'icône permet de modifier la relation sélectionnée dans cette fenêtre.
- ◆ Le bouton qui contient l'icône permet de supprimer la relation sélectionnée dans cette fenêtre.
- ◆ Le bouton qui contient l'icône permet d'accéder à la fenêtre de présentation de l'ontologie des relations du système.
- ◆ Le bouton qui contient l'icône permet de fermer cette fenêtre.
- ◆ Le bouton qui contient l'icône permet de quitter l'application.

La fenêtre d'ajout d'une relation à l'ontologie des relations est présentée dans la figure numéro 6.39.

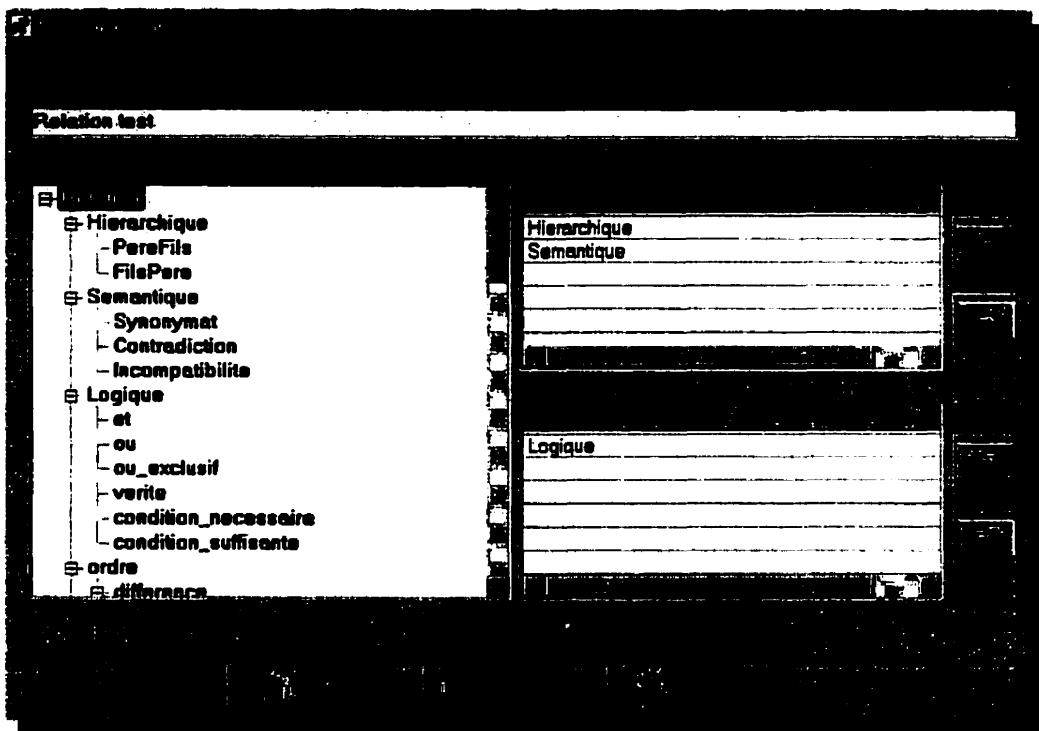


Figure 6.39 : Fenêtre d'ajout des relations à l'ontologie des relations

Dans cette fenêtre :

- ♦ Le bouton qui contient l'icône permet d'ajouter une relation à une liste (liste des pères ou des fils).
- ♦ Le bouton qui contient l'icône permet de supprimer une relation dans une liste (liste des pères ou des fils).

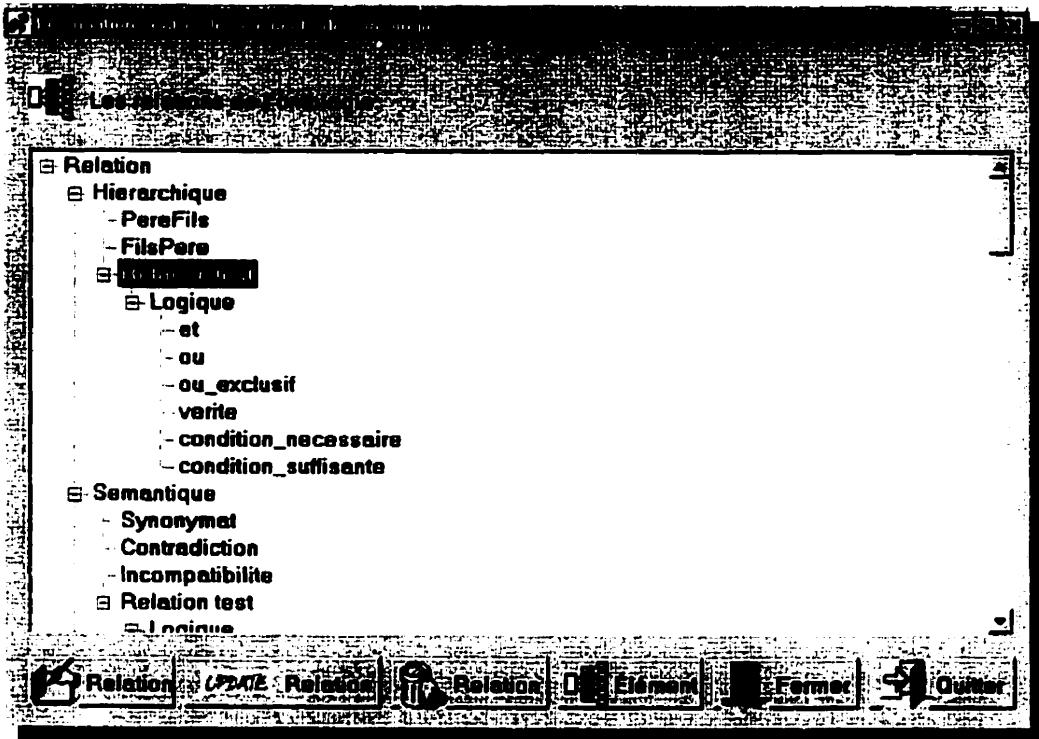


Figure 6.40 : Fenêtre d'affichage de l'ontologie des relations mise à jour

Le système permet aussi de modifier ou de supprimer une relation de l'ontologie des relations en utilisant les boutons correspondant après avoir sélectionné cette relation dans la hiérarchie.

Le module de gestion des ontologies du système permet d'ajouter d'autres relations entre les concepts/expressions de l'ontologie du système en se basant sur l'ontologie des relations du système. La seule contrainte est que l'utilisateur de ce module ne peut ajouter qu'une relation entre deux concepts/expressions seulement. S'il veut ajouter une relation entre trois concepts/expressions il sera obligé de l'ajouter deux à deux.

La figure numéro 6.41 présente la forme d'ajout d'une relation entre deux concepts/expressions

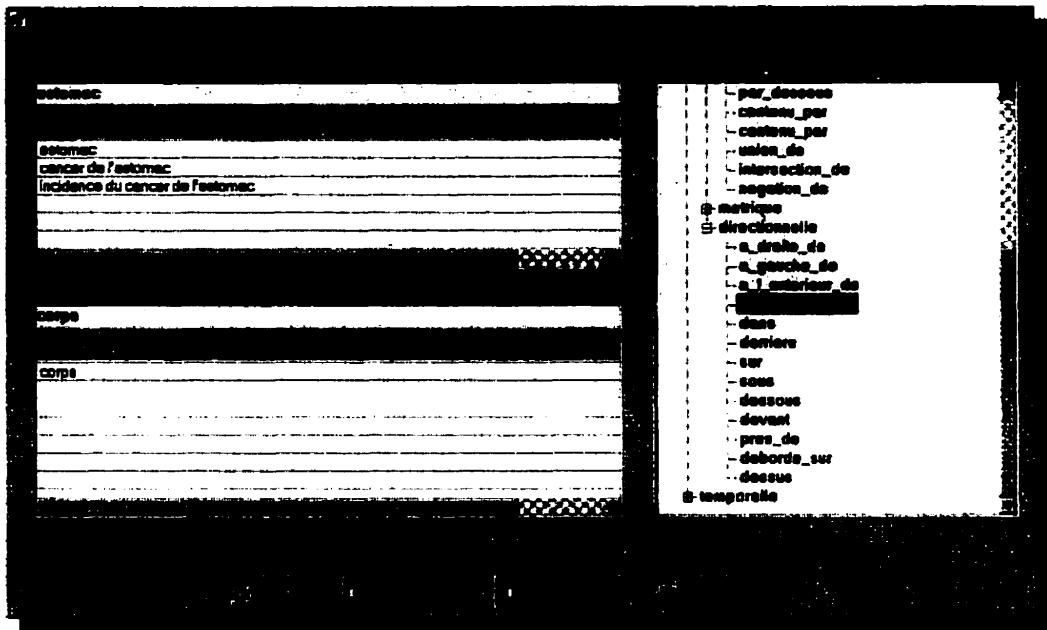


Figure 6.41 : Fenêtre d'ajout d'une relation entre concepts/expressions

6- 4 – Conclusion:

Dans ce chapitre nous avons présenté l'aspect pratique de notre recherche. Nous avons montré le fonctionnement principal de l'application en présentant l'enchaînement des écrans de notre maquette. Nous avons présenté aussi les écrans pour les différents modules de l'application.

Chapitre 7

Expérimentation et évaluation du système :

7 – 1 - Introduction

Dans ce chapitre nous présentons l'aspect expérimentation et l'évaluation de la démarche proposée. Dans une première section, nous présentons la phase d'expérimentation du système en se basant sur une vraie ontologie, et dans une deuxième section, nous présentons l'évaluation ainsi que quelques questions posées à propos du système ainsi que leurs réponses. Enfin, une conclusion clôture le chapitre.

7 – 1 - Expérimentation :

Nous rappelons que le système d'indexation et de recherche se base sur une ontologie de concepts/expressions dans les processus d'indexation et de recherche.

Nous avons expérimenté le système en utilisant une ontologie réelle du domaine de la santé environnementale. Cette ontologie se présente dans l'annexe B de ce mémoire.

Nous avons installé le système sur la machine de l'utilisateur responsable à la gestion de l'ontologie. Cet utilisateur a commencé par saisir les thèmes de l'ontologie du domaine de santé environnementale ainsi que leurs relations, et il nous a fourni beaucoup de commentaires concernant la manipulation de ce module. Ces commentaires vont être présentés dans la section suivante.

Les modules d'indexation et de recherche ont été expérimentés par nous même et pas encore par les utilisateurs potentiels. Pour ces modules nous n'avons pas de commentaires.

7 – 1 - Évaluation :

En ce qui concerne la saisie des concepts/expressions et leurs relations, l'utilisateur responsable de l'ontologie a posé plusieurs questions. Dans ce qui suit, nous présentons les commentaires et les questions les plus importantes et nous allons discuter chacun d'eux.

Au niveau de la saisie des données :

* *L'utilisateur a remarqué un gros problème de mémoire.*

Après avoir saisi un certain nombre de thèmes et de leurs relations (256 thèmes et 369 relations), l'utilisateur du système commence à avoir des problèmes de mémoire. Ce problème est la conséquence de l'utilisation des classes d'objets au niveau des thèmes et des relations entre les thèmes. Le système utilise des tableaux d'objets d'une ou deux dimensions, ce qui consomme beaucoup de ressources mémoire.

Pour résoudre ce problème, nous avons essayé d'utiliser les objets tableaux dont la taille peut changer dynamiquement (comme l'objet *vector* en Java), mais malheureusement ce type d'objet n'existe pas en utilisant C++ dans l'environnement *Builder de Borland*.

* *L'utilisateur aimerait pouvoir visualiser un niveau à la fois ou faire une recherche dans l'ontologie pour un thème.*

Ce problème a été résolu par l'introduction d'une fenêtre de recherche des thèmes dans l'ontologie. Cette fenêtre est appelée par l'utilisateur s'il veut chercher un thème bien particulier pendant la gestion de l'ontologie.

* *Comment peut-on dupliquer et sauvegarder ce fichier (le fichier qui contient les thèmes et les relations entre eux)?*

Nous avons fourni à l'utilisateur toute l'information concernant la sauvegarde des données, leur localisation et comment il peut les dupliquer à l'aide d'une fonction de duplication des données sur la machine.

- *On devrait pouvoir rentrer plusieurs thèmes à la fois (ex. tous les fils d'un thème donné). Actuellement les relations père, fils et relié ne fonctionnent que pour les thèmes préétablis.*

Le modèle de saisie de plusieurs thèmes et leur insertion sur un seul thème père a été développé. Il permet à l'utilisateur de saisir une liste de thèmes dans une liste, puis le programme vérifie si le thème est déjà existant ou non et il l'ajoute sur le thème père spécifié par l'utilisateur. Seulement les relations pères-fils vont être ajoutées. Si l'utilisateur veut ajouter une relation «relié» ou «synonymes», il doit utiliser le module de modification du thème correspondant ainsi que ses relations.

- *On devrait pouvoir charger un certain nombre de thèmes à partir d'une source quelconque. (ex. CIM-10).*

Le module d'ajout de plusieurs fils à un thème donne la main à l'utilisateur et lui permet d'importer une liste des thèmes fils qui existe dans une source (par exemple une table dans une base de donnée) et permet de les ajouter au dessous du thème sélectionné.

* *Comment peut-on indiquer une relation «relié» double pour un thème?*

La relation double est spécifiée dans la liste des thèmes reliés dans la fenêtre d'ajout de thèmes en cochant l'élément dans la liste des thème reliés. Si le thème est coché alors il aura une relation double (double sens) avec le thème sélectionné, sinon il aura une relation simple (un seul sens).

* *Pourquoi la longueur des champs thèmes est limitée?*

Le type de données utilisé pour le champ thème dans la table des thèmes est le type «Mémo» de Access. Mais avec les problèmes de mémoire rencontrés nous avons été obligés de changer le type en type «texte» de taille maximale 255 caractères. L'utilisateur ne doit pas saisir un thème qui dépasse 255 caractères, sinon il va perdre l'information.

D'autres questions ont été posées par l'utilisateur chargé de la gestion de l'ontologie. La liste de ces questions est la suivante :

Que se passera-t-il lorsque:

- * *deux pères ont deux fils différents qui ont les mêmes noms (eau/rejets polluants ; air/rejets polluants) ?*

Dans ce cas l'utilisateur doit utiliser des noms différents pour les deux fils.

- * *deux pères ont un fils commun (air/pluies acides et eau/pluies acides) ?*

Dans ce cas il suffit d'insérer deux relations père-fils entre le père commun et les deux fils.

- * *deux pères ont un fils commun qui se comporte différemment dans chaque contexte (traitement du fumier relié à fumier et gaz à effet de serre) ?*

Dans ce cas il suffit d'insérer deux relations fils-père entre le père commun et les deux fils.

- * *plusieurs fils peuvent-ils se rejoindre en un père autre que leur régulier ? (agriculture et gaz à effet de serre) ?*

Dans la structure de l'ontologie nous avons prévu qu'un thème peut avoir plusieurs pères.

- * *un thème peut-il renvoyer à une autre partie de l'arbre (problèmes de santé liés à la qualité de l'eau ou de l'air) ?*

Dans la structure de l'ontologie nous avons prévu la relation «*relié*» entre deux thèmes pour renvoyer un thème à un autre thème qui se trouve dans une autre partie de l'ontologie.

Au niveau de la structure de l'ontologie, les utilisateurs ont posé plusieurs questions dont nous mentionnons quelques unes dans ce qui suit:

- * *Est-ce que les concepts-expressions trouvés peuvent entrer en conflit les uns avec les autres parce qu'ils se recoupent?*

Les concepts-expressions ne peuvent pas entrer en conflit les uns avec les autres car ils ne se recoupent pas.

* *Est-ce qu'une ontologie construite par cette méthodologie dépend de l'ordre d'examen des textes considérés ou est-elle indépendante de cet ordre?*

Une ontologie construite par cette méthodologie ne dépend pas de l'ordre d'examen des textes considérés. Nous pouvons l'appliquer sur n'importe quel document que ce soit textuel ou multimédia, dans n'importe quel ordre.

• *Est-ce que l'ontologie est constamment cohérente?*

L'ontologie est constamment cohérente car elle se met à jour au fur et à mesure et il y a plusieurs règles utilisées qui assurent sa cohérence.

* *Quel est le coût associé au maintien de cette cohérence, par exemple si la fermeture transitive de l'ontologie est calculée de façon incrémentale?*

Le coût du maintien de cette cohérence est incrémental en fonction de la taille de l'ontologie utilisée, si l'ontologie est de plus en plus volumineuse, c'est évident que l'assurance de sa cohérence sera plus compliquée et prend plus de temps.

* *Quelles sont les conditions d'applicabilité de la méthode? S'applique-t-elle dans tous les cas ou seulement pour des domaines techniques? Peut-on l'adapter facilement à d'autres domaines? Est-elle indépendante de la nature des textes considérés?*

Cette méthode d'indexation et de recherche des documents basée sur une ontologie peut être appliquée dans n'importe quel domaine, il suffit d'utiliser l'ontologie spécifique au domaine. Chaque domaine est caractérisé par une ontologie bien déterminée. Cette méthode est aussi indépendante de la nature des documents utilisés qu'ils soient textuels ou non.

Le système est encore en phase d'essai par plusieurs utilisateurs, ceci veut dire que nous pouvons attendre d'autres questions que ce soit pour le module de gestion de l'ontologie ou pour les modules d'indexation et de recherche des documents multimédias.

7 – 2 - Conclusion

Dans ce chapitre nous avons présenté brièvement la phase d'expérimentation et d'évaluation du système par les utilisateurs. Cette phase a dégagé beaucoup de questions de la part de ces utilisateurs. Nous avons essayé de répondre à la plupart de ces questions et nous avons effectué les modifications nécessaires sur le système pour satisfaire au maximum ces utilisateurs. La phase d'essai du système n'est pas encore achevée et en attendant nous essayons d'améliorer le système et de l'optimiser.

Conclusion générale et perspectives

Dans le cadre de ce mémoire, et après un chapitre introductif, nous avons d'abord fait une revue de littérature sur l'indexation et la recherche documentaire dans un deuxième chapitre. Nous avons présenté les notions de base ainsi que les différents modèles d'indexation et de recherche documentaire dans des bases de données locales ainsi que sur le réseau Internet.

Ensuite, dans un troisième chapitre, nous avons présenté les différentes solutions existantes en matière d'indexation et de recherche de documents. Nous avons présenté des systèmes d'indexation et de recherche dans des bases de données locales et des moteurs de recherche sur le réseau Internet. Des tableaux comparatifs ont été élaborés pour bien comprendre les similitudes et les différences entre ces différentes solutions.

Puis, nous avons proposé trois techniques d'indexation et de recherche d'informations. La première se base sur la notion des mots clés, c'est une technique classique utilisée par presque tous les systèmes existant actuellement. La deuxième technique que nous avons envisagée est purement sémantique, elle permet à l'indexeur de construire des graphes conceptuels pour les index des documents et elle permet à l'utilisateur de construire des graphes conceptuels correspondants à ses requêtes. Dans ce cas, la recherche , n'est qu'un matching entre les différents graphes conceptuels des documents indexés et des requêtes. La dernière technique envisagée est une technique mixte qui se base sur la notion des concepts/expressions. Parmi ces trois techniques, nous avons retenu la troisième.

Après la présentation de la technique choisie, nous l'avons présentée en détail dans le cinquième chapitre. Nous avons présenté le système qui la met en œuvre ainsi que ses différents sous-systèmes.

Par la suite, nous avons présenté l'aspect pratique du système réalisé qui met en place la technique retenue dans un sixième chapitre. Dans ce chapitre, nous avons présenté l'enchaînement des écrans de l'application.

Ensuite, dans un dernier chapitre, nous avons clôturé notre mémoire par une conclusion générale.

De manière générale, la technique d'indexation et de recherche retenue, présente beaucoup d'avantages par rapport à la technique basée sur la notion des mots-clés et aussi par rapport à la technique purement sémantique en terme de simplicité d'interfaces d'indexation et de recherche.

Enfin, nous signalons que notre travail n'a pas été encore évalué par les utilisateurs potentiels. En attendant cette phase d'évaluation, nous allons améliorer les aspects ergonomiques des interfaces du système réalisé.

Bibliographie

[Belhassen 99] Amina Sayeb Belhassen. *Prise en compte de l'aspect utilisateur au niveau de la recherche documentaire sur Internet.* Article, Laboratoire PGL, ENSI, Tunis, 1999.

[Berinstein 98] Paula Berinstein. *The big picture : Image Search Engines on the Web.* Article, Mosco, Janvier 1998.

[CheinMugnier 92] M.Chein et M.L.Mugnier. *Conceptual graphs, fundamental notions.* In revue d'intelligence artificielle, 6(4) pp 365-406. 1992.

[DTI 97] Direction des technologies de l'information. *Les principaux problèmes de la recherche d'information sur Internet.* Article 1997

<http://www-dist.cea.fr/ext/neuf/moteur/tabledesmatieres.html>

[Herlin 97] Richard Herlin. *Indexation et recherche sur Internet.* Ecole Supérieur de Journalisme de Lile. Article, 1997.

[Kammoun 97] Hager Kammoun. *Classification automatique des textes dans un fond documentaire.* Mémoire de DEA, Faculté des sciences de Tunis, 1997.

[KraftBuell 83] Kraft, D. H. and Buell, D. A. *Fuzzy sets and generalized Boolean retrieval systems.* International Journal on Man-Machine Studies, 19: 49-56, 1983.

[Leloup 98] Catherine Leloup. *Moteurs d'indexation et de recherche: Environnement Client-Serveur Internet et Intranet.* Eyrolle, 1998.

[**Lamirel 97**] Jean-Charle Lamirel. *Application d'une approche symbolico connexioniste pour la conception d'un système documentaire hautement interactif, le prototype NOMAD.* Thèse de doctorat de l'université Henry-Point Carre, NancyI , 1997.

[**MartinTalon 97**] Philippe Martin et Bénédicte Talon. *Aide à l'acquisition de connaissances sémantiques: des cadres sémantiques au graphes conceptuels.* Articles, revue RIA, 1997.

[**MartinAlpay 96**] Philippe Martin et L.Alpay. *Conceptual structures ans structured documents.* In processing of ICCS'96. Article. Sydney, Australia, August 19-22, 1996.

<http://citeseer.nj.nec.com/martin96conceptual.html>

[**MartinEklund 97**] Philippe Martin et Peter Eklund. *Embedding knowledge in web documents.* Griffith university, School of Information Technology, Australia. 1997
<http://Meganesia.int.gu.edu.au/~phmartin/webKB/doc/papers/www8/www8.html>

[**Martin 96**] Philippe Martin. *Exploitation de graphes conceptuels et de documents structurés et hypertextes pour l'acquisition de connaissances et la recherche d'informations.* Thèse de Doctorat. Université Sphia Antipolis. 1996.

[**Nastar 99**] Chahab Nastar. *Indexation et recherche d'images : Enjeux, méthodes et perspectives.* Article (Congrès IDT'99), Paris, France , 10 Juin 1999 .
<http://www.rocq.inria.fr/~nastar/nastar-idt99.html>.

[**NastarBoujema 99**] Chahab Nastar, Nadia Boujema. *Indexation et recherche d'images par le contenu.* Atelier Traitement et Analyse d'Images : Méthodes et Applications (TAIMA'99). INRIA/CFDI/CERT., Hammamet (Tunisie), Mars 1999.
<http://www.rocq.inria.fr/~boujema/Recherche.html>.

[**OIN 2000**] Organisation Internationale de Normalisation. *MPEG-7 Applications Documents.* Mars 2000.

<http://www.meta-labs.com/mpeg-7/W2329.pdf>

[Proux 97] Deny Proux. *Extraction automatisée d'informations à partir de notices bibliographiques rédigées en langages naturels*. Mémoire de stage. Université de Bourgogne. 1996-1997.

[Radecki 79] Radecki, T. *Fuzzy set theoretical approach to document retrieval*. Information Processing & Management, 15: 247-259, 1979.

[RejmanFaltungs 97] Martin Rajman et Boi Faltings. *A la poursuite de l'information : Techniques de recherche et d'analyse pour données textuelles*. Article, EPFL-DI, Laboratoire d'intelligence artificielle, France . 2 Septembre 1997
<http://sawww.epfl.ch/SIC/SA/publications/FI97/FI-SP-97/sp-97-page34.html>.

[SaltonFoxAH 83] Salton, G., Fox, E. A. and Wu, H. *Extended Boolean information retrieval*. Communications of the ACM, 26(12): 1022-1036, 1983

[Schweighofer 99] Erich Schweighofer. *The revolution in legal information retrieval or : The empire strikes back*. Institute of Public International Law Research Center for Computers and Law. University of Vienna, 26 February 1999.

[Sowa 84] Sowa J. *Conceptual structures : Information processing in man and machine*. Addison-wesley, 1984.

[Stix 97] Gary Stix. *La recherche d'images sur le Web*. Journal Scientific American, Article numéro 235, mai 1997.

<http://www.pourlascience.com/numeros/pls-235/internet/lynchbox1.htm>

[Trigano 94] Philippe Trigano. *Indexation automatique et sauvegarde des connaissances de l'entreprise*. Article : Projet ISMICK'94, Université de Compiègne, France, 1994.

<http://magi.com/~godbout/Kbase/trigano.htm#2ind>

[Zadeh 65] Zadeh, L. A. *Fuzzy sets*. *Information and Control*, 8: 338-353, 1965.

[Vézina 99] Kumiko Vézina. Survol du monde de l'indexation des images. Cursus (périodique électronique étudiant de l'école de bibliothéconomie et des sciences de l'information (EBSI) de l'université de Montréal). Université de Montréal, 1999.
<http://www.fas.umontreal.ca/EBSI/cursus/vol4n01/vezina.html>.

[WallerKraftH 79] Waller, W. G. and Kraft, D. H. *A mathematical model for a weighted Boolean retrieval system.* Information Processing & Management, 15: 235-245, 1979.

Annexe A: Liste des principaux algorithmes

Dans cet annexe, nous présentons les différents algorithmes des programmes des différents modules de l'application.

L'algorithme d'extraction des mots de la requête :

Cet algorithme permet d'extraire les mots d'une chaîne de caractères qui contient plusieurs mots. Les mots extraits sont mis dans un tableau des mots. L'algorithme a comme paramètres «la requête» et «un tableau» qui va contenir les mots de la requête après leur extraction.

```
Algorithme ExtractMotReq : Requete , TabMotsReq
PosFin <- -1
Mot <- ""
Compt <- 0

Si (Requete <> "")
Alors
    Tant_que (PosFin <> 0)

        PosFin <- 0
        PosFin -> Requete.Position ("")
        Si (PosFin <>0)
        Alors
            Mot <- Requete.SousChaine (1, PosFin -1)
            Requete <- Requete.SousChaine (PosFin, Requete.Taille () +1)
            Requete <- Requete.Trim ()
```

```

        TabMotsRequ [Compt] <- Mot
        Compt <- Compt + 1
    Sinon
        Requete <- Requete.Trim ()
        TabMotsRequ [Compt] <- Requete
        Compt <- Compt + 1
    Fin Si

    Fin Tant_que
Sinon
    Afficher ("La requete est vide.")
Fin Si
Fin Algorithme

```

L'algorithme d'extraction des mots non vides de la requête :

Cet algorithme permet d'extraire les mots non vides de la requête. Il a comme paramètre le tableau des mots de la requête, un tableau des mots non vides et un tableau dans lequel nous allons extraire les mots non vides de la requête.

**Algorithme ExtractMontNonVideReq : TabMotsReq, TabMotsVides,
TabMotsNonVidesReq**

```

NbMotsNonVidesReq <- 0
EstVide <- Faux
Compt <- 0
Compt2 <- 0
MotReq <- ""
MotVide <- ""

Tant_que (Compt < TabMotsReq.Taille)

    MotReq <- TabMotsReq [Compt]
    MotReq <- MotReq.Trim ()
    MotReq <- MotReq.Majuscule ()

    Compt2 <- 0
    Tant_que (Compt2 < TabMotsVides.Taille)
        MotVide <- TabMotsVides [Compt2]
        MotVide <- MotVide.Trim ()
        MotVide <- MotVide.Majuscule ()
        Si (MotVide = MotReq)
            Alors
                EstVide = Vrai
            Fin Si
            Compt1 <- Compt2 + 1
        Fin Tant_que
        Si (EstVide = faux)

```

```

Alors
  TabMotsNonVidesReq [NbMotsNonVidesReq] <- MotReq
  NbMotsNonVidesReq <- NbMotsNonVidesReq + 1
Fin Si
EstVide <- False
Compt <- Compt + 1

Fin Tant_que
Fin Algorithme

```

L'algorithme d'extraction des Concepts/Expressions qui contiennent une des mots non vides de la requête:

Cet algorithme permet de chercher les identifiants des concepts/expressions qui contiennent au moins un mots des mots non vides de la requête.

**Algorithme ExtractConExprRes : TabMotsNonVidesReq, TabConceptsExpr,
TabIdConceptExprRes**

```

MotNonVide <- ""
Ok <- 1
Compt <- 0
Compt2 <- 0
ConExpr <- ""
IdConExpr <- 0
NbIdConceptExprRes <- 0

Tant_que (Compt < TabConceptsExpr.Taille)
  ConExpr <- TabConceptsExpr.Taille [Compt].IdConceptExpression
  IdConExpr <- TabConceptsExpr.Taille [Compt].IdConceptExpression

  Tant_que (Compt2 < TabMotsNonVidesReq)
    MotNonVide <- TabMotsNonVidesReq [Compt2]
    MotNonVide <- MotNonVide.Trim ()
    MotNonVide <- MotNonVide.Majuscule ()

    Si (ConExpr.Position (MotNonVide) = 0)
      Alors
        Ok <- Ok * 0
      Fin Si

      Compt2 <- Compt2 + 1
    Fin Tant_que

    Si (Ok = 1)
      Alors
        TabIdConceptExprRes [NbIdConceptExprRes] <- IdConExpr

```

```

NbIdConceptExprRes <- NbIdConceptExprRes + 1
Fin Si
Compt <- Compt + 1
Fin Tant_que

Fin Algorithme

```

L'algorithme de mise en correspondance des concepts/expressions avec les documents :

Cet algorithme permet de mettre en correspondance les expressions qui répondent à la requête de l'utilisateur avec la base d'index pour donner en résultats les documents qui répondent à cette requête.

**Algorithme MatchDocument : TabIdConceptExprRes, TabBaseIndex,
TabIdDocRes**

```

IdConExprBI <- 0
DocIndBI <- ""
IdConcExprRes <- 0
Compt <- 0
Compt2 <- 0
NbDocRes <- 0

Tant_que (Compt < TabIdConceptExprRes.Taille)
    IdConcExprRes <- TabIdConceptExprRes [Compt]
    Tant_que (Compt2 < TabBaseIndex.Taille)
        IdConExprBI <- TabBaseIndex [Compt2].IdConceptExprBaseIndex
        DocIndBI -> TabBaseIndex [Compt2].DocIndBaseIndex
        Si (IdConcExprRes = IdConExprBI)
            Alors
                NbrTi <- 0
                PosFin <- -1
                RestChaine <- DocIndBI.Trim ()
                Tant_que (PosFin < 0)
                    PosFin <- 0
                    PosFin <- RestChaine.Position ("")
                    Si (PosFin < 0)
                        Alors
                            RestChaine <- RestChaine.SousChaine (PosFin+1 ,
                                RestChaine.Taille ()-1)
                            NbrTi <- NbrTi + 1
                        Fin Si
                    Fin Tant_que

        DocIndParExpr <- ""

```

```

DocIndParExpr <- DocIndBI.Trim ()
ResteDocIndParExpr <- ""
IdDoc <- 0

ResteDocIndParExpr = DocIndParExpr.SousChaine (2 ,
                                                DocIndParExpr.Taille () - 2)
PosFinInd = -1
Tant_que (PosFinInd < 0)
    PosFinInd <- 0
    PosFinInd <- ResteDocIndParExpr.Pos (" - ")
    Si (PosFinInd < 0)
        Alors
            IdDoc <- ResteDocIndParExpr.SousChaine (1 , PosFinInd-1)
            ResteDocIndParExpr <- ResteDocIndParExpr.SousChaine
                (PosFinInd+1 , ResteDocIndParExpr.Taille () -2)
            ResteDocIndParExpr <- ResteDocIndParExpr.Trim ()
            TabIdDocRes [NbDocRes] <- IdDoc
            NbDocRes <- NbDocRes + 1
        Sinon
            TabIdDocRes [NbDocRes] <- IdDoc
            NbDocRes <- NbDocRes + 1
        Fin Si
    Fin Tant_que
Fin Si
Compt2 <- Compt2 + 1
Fin Tant_que
Compt <- Compt + 1
Fin Tant_que

```

Fin Algorithme

L'algorithme de filtrage :

Cet algorithme permet le filtrage de la liste des documents trouvés en résultat de la recherche.

Algorithme FiltrageDocument : TabIdDocRes, TabIdDocResFinauxFiltre

```

ComptRes <- 0
ComptResFinaux <- 0
Compt3 <- 0
ComptFiltre <- 0

Appart <- faux
Tant_que (ComptRes < TabIdDocRes.Taille)
    Eleml1 <- 0
    Eleml1 <- TabIdDocRes [ComptRes]

```

```

Tant_que (ComptResFinaux < TabIdDocResFinaux.Taille)
    Eleml2 <- 0
    Eleml2 <- TabIdDocResFinaux [ComptResFinaux]
    Si (Eleml1 = Eleml2)
        Alors
            Appart <- vrai
        Fin Si
    Fin Tant_que
    Si (Appart = false)
        Alors
            TabIdDocResFinaux [NbIdDocResFinaux] <- TabIdDocRes [ComptRes]
        Fin Si
        Appart <- faux
    Fin Tant_que

```

```

Tant_que (Compt3 < TabIdResFinaux.Taille)
    TabIdDocResFiltre [Compt3] <- TabIdDocResFinaux [Compt3]
Fin Tant_que

```

```

Si (FiltreTexte = 1)
Alors
    IdD <- 0
    Tant_que (ComptFiltre < TabIdResFiltre.Taille)
        IdD <- TabIdDocResFiltre [ComptFiltre]
        IDDocFiltre <- 0
        IDDocFiltre <- Chercher (TabDocument, IdD)
        Si (IDDocFiltre = 1)
            Alors
                TabIdDocResFiltre [ComptFiltre] <- IDDocFiltre
            Fin Si
        Fin Tant_que
    Fin Si

```

```

Si (FiltreImage = 1)
Alors
    IdD <- 0
    Tant_que (ComptFiltre < TabIdResFiltre.Taille)
        IdD <- TabIdDocResFiltre [ComptFiltre]
        IDDocFiltre <- 0
        IDDocFiltre <- Chercher (TabDocument, IdD)
        Si (IDDocFiltre = 2)
            Alors
                TabIdDocResFiltre [ComptFiltre] <- IDDocFiltre
            Fin Si
        Fin Tant_que
    Fin Si

```

Si (FiltreAudio = 1)
Alors
 IdD <- 0
 Tant_que (ComptFiltre < TabIdResFiltre.Taille)
 IdD <- TabIdDocResFiltre [ComptFiltre]
 IDDocFiltre <- 0
 IDDocFiltre <- Chercher (TabDocument, IdD)
 Si (IDDocFiltre = 3)
 Alors
 TabIdDocResFiltre [ComptFiltre] <- IDDocFiltre
 Fin Si
 Fin Tant_que
 Fin Si

Si (FiltreVideo = 1)
Alors
 IdD <- 0
 Tant_que (ComptFiltre < TabIdResFiltre.Taille)
 IdD <- TabIdDocResFiltre [ComptFiltre]
 IDDocFiltre <- 0
 IDDocFiltre <- Chercher (TabDocument, IdD)
 Si (IDDocFiltre = 4)
 Alors
 TabIdDocResFiltre [ComptFiltre] <- IDDocFiltre
 Fin Si
 Fin Tant_que
 Fin Si

Si (FiltreBD = 1)
Alors
 IdD <- 0
 Tant_que (ComptFiltre < TabIdResFiltre.Taille)
 IdD <- TabIdDocResFiltre [ComptFiltre]
 IDDocFiltre <- 0
 IDDocFiltre <- Chercher (TabDocument, IdD)
 Si (IDDocFiltre = 5)
 Alors
 TabIdDocResFiltre [ComptFiltre] <- IDDocFiltre
 Fin Si
 Fin Tant_que
 Fin Si

Si (FiltreAutres = 1)
Alors
 IdD <- 0

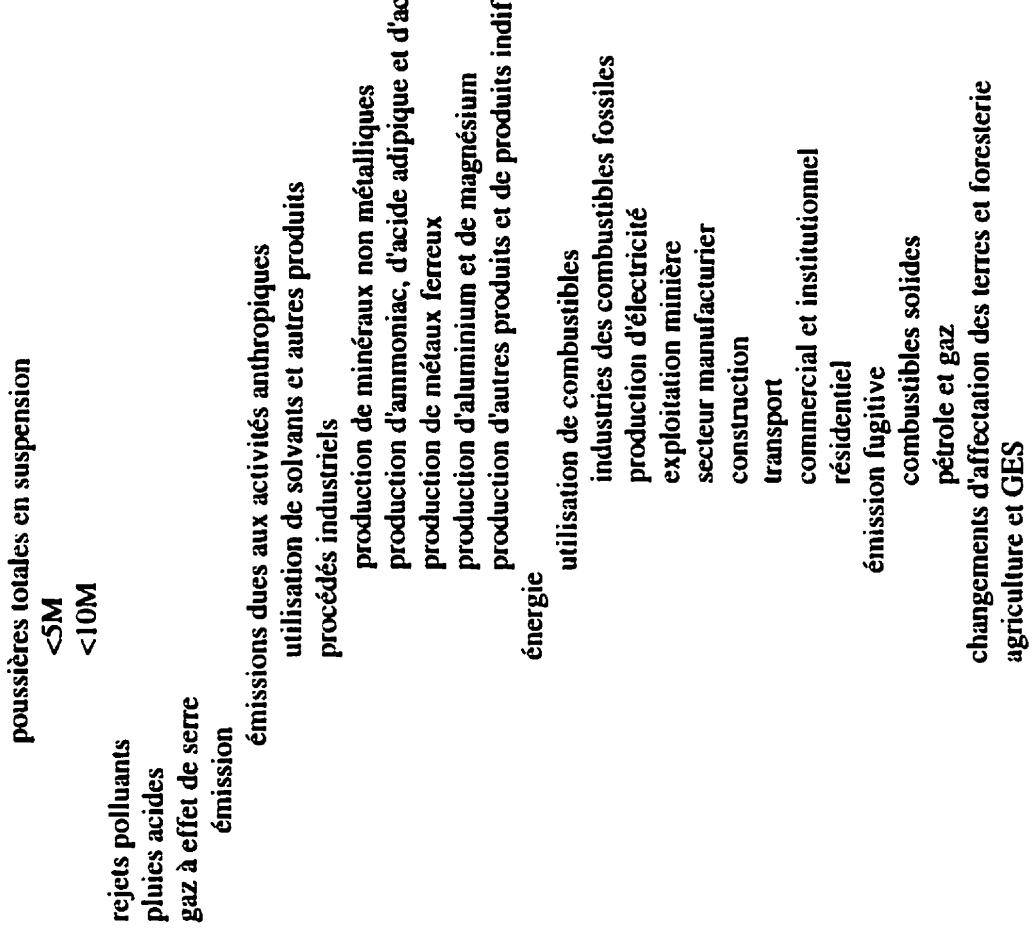
```
Tant_QUE (ComptFiltre < TabIdResFiltre.Taille)
    IdD <- TabIdDocResFiltre [ComptFiltre]
    IDDocFiltre <- 0
    IDDocFiltre <- Chercher (TabDocument, IdD)
    Si ( IDDocFiltre = 6)
        Alors
            TabIdDocResFiltre [ComptFiltre] <- IDDocFiltre
        Fin Si
    Fin Tant_QUE
Fin Si
```

Fin Algorithme

Annexe B : L'ontologie test du système

Dans cet annexe, nous présentons l'ontologie test du domaine santé environnementale que nous avons utilisé pour tester le système. Cette ontologie est introduite par l'utilisateur responsable de la gestion du système.

| | |
|-----------------|--|
| theme universel | |
| contamination | |
| vecteurs | |
| air | |
| | climat |
| | qualité de l'air extérieur |
| | pollution de l'air |
| | mesures de la qualité de l'air |
| | normes horaires et périodes de 8H, de 24H et annuelles |
| | fréquence des dépassements |
| | stations d'échantillonnage |
| | contaminants mesurés |
| | SO ₂ |
| | NO ₂ |
| | CO |
| | O ₃ |
| | H ₂ S |



| | |
|--|--|
| traitemet du fumier | |
| déchets et GES | |
| autres émissions | |
| interventions, enquêtes et avis de santé publique | |
| nombre de demandes reçues dans les DRSP | |
| pollen | herbe à poux |
| | parcs et espaces verts |
| | dépenses des entreprises |
| | dépenses gouvernementales |
| | exposition au soleil |
| | ozone stratosphérique |
| | lotions solaires |
| | évolution des ventes |
| | indice uv |
| | nombre de jours où il dépasse le niveau modéré |
| | bronzage |
| | coups de soleil |
| forêt | |
| | coupe et reboisement |
| | qualité de l'air intérieur |
| | humidité |
| | humidificateurs et déshumidificateurs |
| | moisissures et champignons |
| | maladie du légionnaire |
| | chauffage et générateurs d'air chaud |
| | ventilation et air climatisé |
| | échangeurs et purificateurs d'air |
| | interventions, enquêtes et avis de santé publique |
| | tabac |

règlements pour la protection des non-fumeurs

édifices publiques

L'ANNEE DES CO

卷之三

TAUX DE NOUVEAUX

problèmes de santé liés à la qualité de l'air

三

pollution de l'eau pluies acides

rejects noxious

විජය පෙරේරා

LIBRISQUES

allegans

culture

aericulture

Destinies

particular

Évolution des ventes de pesticides

utilisation de pesticides par type de culture

Catégories de pesticides

herbicides

coincides

rodenticides

Influence 1

bioRxiv preprint doi:

Insecticides

engrais

nature

sumier solide

Traitements du sumier

三

loss

chimique

évolution des ventes d'engrais

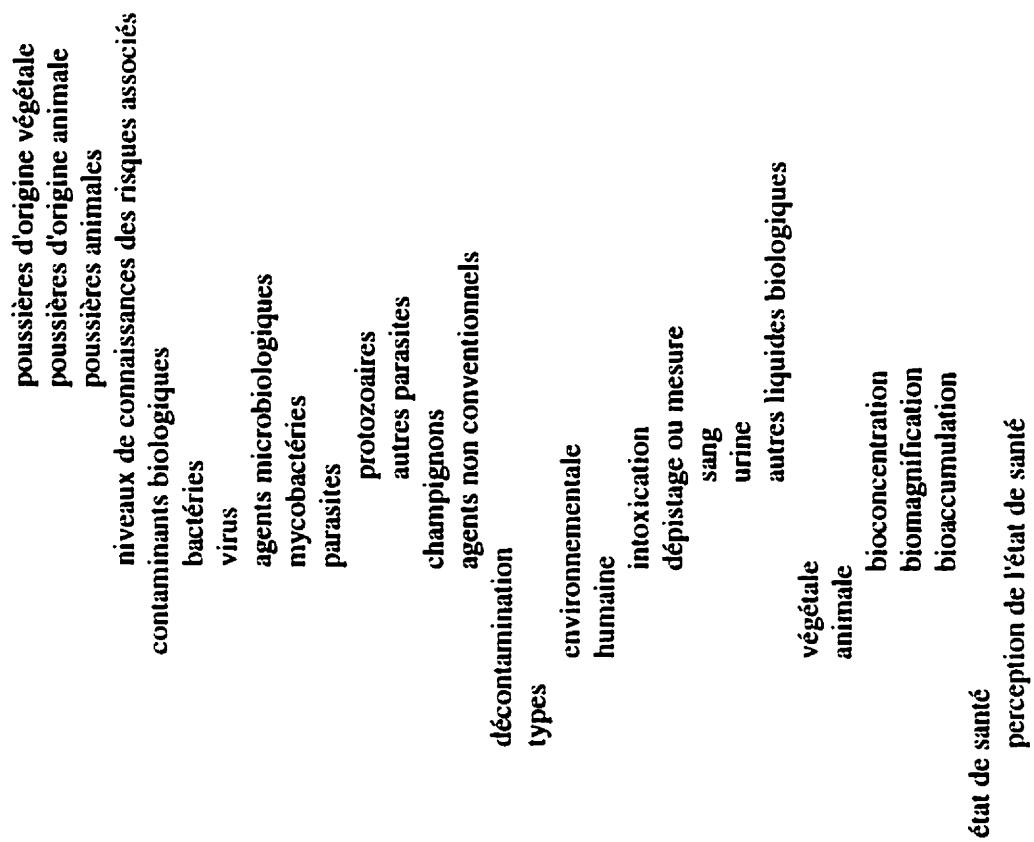
capacité de support des sols

| | |
|--------------------------------|--|
| régions agricoles | |
| irrigation | |
| exploitations agricoles | |
| aquaculture | |
| pisciculture | |
| horticulture | |
| eau potable | |
| | approvisionnement |
| | sources |
| | eaux de surface |
| | eaux souterraines |
| | puits individuels |
| | stations, postes ou usines |
| | réservoirs |
| | traitement ou assainissement |
| | appareil domestique de traitement de l'eau |
| | procédés |
| | filtration |
| | flocculation |
| | osmose inversée |
| | chloration |
| | ozonation |
| | charbon activé |
| | réseau de distribution |
| | nombre d'avis d'ébullition |
| | perception de la qualité |
| | canalisation |
| | consommation |
| | ingestion |
| | eau en vrac |

| | |
|--|--|
| eaux embouteillées | |
| hygiène | |
| chauffe-eau et douche | |
| mesures d'économie | |
| participation des entreprises et collectivités | |
| qualité de l'eau potable | |
| nombre de demandes reçues dans les DRSP | |
| interventions, enquêtes, avis de santé publique | |
| normes pour la qualité de l'eau | |
| normes pour les équipements et procédures | |
| eaux récréatives | |
| qualité des eaux récréatives | |
| nombre de demandes reçues dans les DRSP | |
| interventions, enquêtes, avis de santé publique | |
| normes pour la qualité de l'eau | |
| normes pour les équipements et procédures | |
| baignade | |
| plages | nombre de jours avec plages fermées |
| eaux usées | |
| traitement (assainissement, épuration) | |
| postes (ouvrages ou installations) | |
| égouts et fosses septiques | |
| qualité des eaux usées | |
| normes pour la qualité de l'eau | |
| normes pour les équipements et procédures | |
| régions hydriques | |

| | | |
|--------------------------------|---|--------------------------------------|
| lacs | épidémies d'origine hydrique | éléments chimiques |
| fleuves | problèmes de santé liés à la qualité de l'eau | substances inorganiques |
| mers | phénomènes physiques | éléments spécifiés et leurs composés |
| rivières | inondation | acides inorganiques |
| cours d'eau | érosion | chlorobenzènes |
| berges | sédiments | substances organiques halogénées |
| maraîches | pesticides | substances composées d'isomères |
| marécages | contaminants physico-chimiques | |
| bassins versants | contaminants chimiques | |
| sol | pesticides | |
| aliments | éléments chimiques | |
| animaux | substances inorganiques | |
| insectes | éléments spécifiés et leurs composés | |
| contaminants | acides inorganiques | |
| contaminants physico-chimiques | chlorobenzènes | |
| contaminants chimiques | substances organiques halogénées | |
| pesticides | substances composées d'isomères | |

| | |
|---|---------------------------------------|
| colorants | acides et bases faibles et leurs sels |
| sels | |
| hydrocarbures aromatiques | |
| autres hydrocarbures | |
| substances comportant de l'azote | |
| substances comportant de l'oxygène | |
| substances comportant du soufre | |
| contaminants physiques | |
| radiations | |
| ionisantes | |
| | cosmique |
| | gamma |
| | rayon x |
| non ionisantes | |
| visibles | |
| | violet |
| | bleu |
| | vert |
| | jaune |
| | orange |
| | rouge |
| non visibles | |
| | ultraviolet |
| | infrarouge |
| | micro-ondes |
| | radiofréquence |
| | bruit |
| | magnétique |
| poussières | |



| | |
|---|--|
| déterminants | |
| groupes | |
| individus | |
| habitudes de vie | |
| alcool | |
| tabac | |
| drogues | |
| indice de masse corporelle | |
| poids à la naissance | |
| âge | |
| alimentation | |
| susceptibilités | |
| héritéité | |
| allergies | |
| infertilité | |
| espérance de vie | |
| espérance de vie en bonne santé | |
| années potentielles de vie perdues | |
| comportements préventifs | |
| maladies | |
| classification | |
| maladies évitables par la vaccination | |
| maladies à déclaration obligatoire | |
| zoonoses | |
| pathologies diverses | |
| méthodes statistiques | |
| nombre de demandes d'investigations | |
| nombre d'investigations | |
| nombre de demandes courantes reçues dans les DRSP | |
| taux d'hospitalisation | |

taux de décès
naissances
durée de l'allaitement
retards de croissance
poids à la naissance
mortinassances
interventions
mesures
méthodologies
indicateurs
mesures de laboratoire