MINISTERE DE L' EDUCATION NATIONALE

SECRETARIAT D' ETAT A LA RECHERCHE

THESE

Présentée au

CENTRE DE DEVELOPPEMENT DES TECHNOLOGIES AVANCEES

Pour l'obtention du grade de

MAGISTER EN CYBERNETIQUE

(Option : Architecture des systèmes)

OPTIMISATION DU STOCKAGE ET DE L' ACCES DANS LES BASES DE DONNEES TRES VOLUMINEUSES

Par

Mme Fatma-Zohra BESSAI
née MECHMACHE

Soutenue le 15 Novanbre 1992 Devant le jury composé de

Mr H. Bessalah	.....	Président
Mr A. Abdellaoui	.....	Examinateur
Mme Z. Ali-Mazighi	.....	Examinateur
Mr M. Benhamadi	.....	Examinateur
Mme F. Cherief	.....	Examinateur
Melle F. Azrou	.....	Rapporteur

Remerciements

Je tiens a remercier,

Monsieur H. BESSALAH, Directeur du centre de developpement des technologies avancees (CDTA), pour avoir bien voulu me faire l'honneur de presider le jury.

Monsieur M. BENHAMADI, Directeur du CERIST, pour avoir mis a ma disposition les moyens necessaires *8* la bonne marche de mon travail, pour la confiance qu'il m'a temoignee, et pour sa participation au jury.

Je suis tres honoree de la participation a ce jury de Madame Z. ALI-MAZIGHI, chargee de cours à l'USTHB, Madame F. CHERIEF, maître de conference (USTHB) et de Monsieur A. ABDELLAOUI, maitre de conference (USTHB).

Je tiens egalement a remercier Mademoiselle F. AZROU, chargee de recherche au CERIST, pour avoir propose le sujet de recherche et dirigé mon travail.

Je ne peux pas oublier ici H. LABIOD, Ingénieur au CERIST, pour son aide et sa collaboration pendant l'intégration de mon travail au systeme SIBADOC.

Je ne saurais clore ces pages sans adresser mon expression de sympathie la plus sincere a tous mes collegues et amis du Laboratoire de Recherche et Developpement en Informatique (LRDI), ainsi qu'à l'ensemble du personnel du CERIST. Que toutes celles et ceux qui m'ont témoigné leur disponibilité et apporte leur aide trouvent ici une expression de reconnaissance.

Enfin je tiens a remercier tous mes proches pour leur soutien et patience tout au long de l'élaboration de ce travail. Qu'ils trouvent ici une recompense a leurs efforts.

RESUME

La masse d'information mise a la disposition de la communaute scientifique s'accroît de façon continuelle, etant donnée l'évolution et la diversification des disciplines scientifiques et techniques. Cette grande masse d'information nécessite l'optimisation des moyens de stockage et le developpement de nouvelles methodes d'accès car les methodes classiques se revelent insuffisantes.

L'objectif de notre travail est de contribuer a l'amelioration des techniques d'accès aux bases de donnees volumineuses. Or une methode d'accès est etroitement liée a la methode de stockage utilisée. Aussi, une partie de notre travail a consiste a etudier le compactage de donnees, utilisé au CERIST pour resoudre le probleme de stockage de grands volumes de donnees.

Apres l'étude des diverses techniques de compression, notre travail a essentiellement consiste a concevoir et à realiser un systeme d'accès aux donnees volumineuses. Ce systeme utilise une technique d'indexation basee sur l'inversion totale, ce qui permet aux utilisateurs d'accéder a la base de donnees par n'importe quel mot contenu dans cette dernière et offre un acces rapide a l'information grâce a l'utilisation du Q-arbre qui est un arbre équilibré de profondeur 1. En plus de ces caracteristiques, le systeme a une structure modulaire lui permettant de s'adapter a tous les systemes gerant de grands volumes de donnees textuelles.

MOTS CLES

Base de donnees documentaire, systeme de recherche documentaire, systeme de gestion d'objets complexes, stockage des donnees, acces aux bases de donnees, compression de donnees, indexation, Q-arbre.

ABSTRACT

The amount of scientific information is growing steadily. This huge amount of information requires efficient storage management and new access methods.

Our goal is to improve the current access techniques to very large data bases. Since an access technique is closely related to the storage method used, in the first part of our work we study the data compression techniques already used by the CERIST in order to store a large amount of information.

After an analysis of the current techniques of data compression, we concentrate on the design and implementation of an access system to very large data bases. Our system uses an indexing scheme based on full inversion which allows users to query the data base using any word contained in the data base.
This system is based on the Q-tree (B-tree of depth one) that indeed allow fast access to the sought information. In addition, the system modular structure makes it adaptable to all system managing very large textual data.

KEY WORDS

Bibliographic data base, document-retrieval system, complex object management system, data storage, data base access, data compression, indexing, Q-tree.

S O M M A I R E

INTRODUCTION

Depuis l'apparition de la notion de bases de donnees vers le debut des annees 60, on assiste à une proliferation de systemes de gestion de donnees operant sur de grandes quantités d'information. Les possibilités limitees de stockage et d'accès rendent difficile la manipulation de cette importante masse de donnees. Par consequent, a defaut de disposer de toutes les capacites de stockage nécessaires, seules les donnees les plus recentes sont accessibles aux utilisateurs. Ce probleme est pose de façon aigue, au Centre de Recherche sur l'Information Scientifique et Technique (CERIST), dans les deux contextes suivants:

- D'une part, lors de l'exploitation des bases documentaires internationales telles que INSPEC, INIS et AGRIS. En effet, a titre d'exemple, une base annuelle de INSPEC necessite environ 300 MegaOctets de mémoire, sans compter les informations de gestion utilisées par le SRD. Aussi actuellement, seule la base qui correspond a la derniere année sera accessible aux utilisateurs.

- D'autre part, lors du developpement du systeme de gestion d'objets complexes SYGOC [BOU 91]. En effet, le systeme SYGOC est conçu pour la gestion et le stockage des donnees volumineuses et a structure complexe, tel que les documents, d'où necessite de beaucoup d'espace et de methodes d'acces optimisees.

Ce probleme de stockage et de manipulation de grands volumes de donnees nous a incite a developper un nouveau systeme informatique, dont le but est de manipuler efficacement des bases de donnees tres volumineuses, et ceci en :

- Minimisant l'espace de stockage pour les bases de donnees.

- Permettant une recherche rapide.

- Effectuant les mises a jour des bases de donnees en un temps satisfaisant.

Pour reduire l'espace de stockage de la base, une solution a ete adoptee, c'est la compression de donnees. Ce travail a été déjà réalisé et a conduit au developpement d'un schema de compression de donnees [MEC 89], qui, intégré au systeme de recherche documentaire **SIBADOC** [BEN 88], developpe au CERIST, a donne des resultats satisfaisants. En effet, la base INSPEC est réduite a environ le tiers de sa taille initiale etant donne que le taux de compression est approximativement égal a 2/3.

Le probleme de l'optimisation dans l'espace de stockage des donnees ayant ete ainsi resolu, notre travail consistera a developper un systeme d'accès rapide qui pourrait accelerer la recherche de l'information, et par consequent, augmenter l'efficacité de notre systeme de stockage et d'accès dans des bases de donnees volumineuses.

En ce qui concerne la realisation, le systeme SYGOC etant en cours de developpement, nous nous sommes interesses a l'application au systeme de recherche documentaire **SIBADOC** qui lui est en phase d'exploitation. Par la suite tout ce qui a été developpe sur **SIBADOC,** pourrait être transpose sur le systeme **SYGOC,** moyennant quelques modifications, car les problemes **poses** par les deux systemes sont identiques.

Apres une etude de l'évolution des systemes de gestion de donnees, presentee dans le ler chapitre , le reste du document est organise de la maniere suivante:

Le 2eme chapitre presente une etude sur la compression de donnees, en donnant une double classification des techniques de compression de donnees (TCD), tout en presentant les TCD les plus couramment utilisées et les criteres de choix d'une TCD pour la compression de donnees bibliographiques, ainsi que l'intégration de cette derniere au systeme SIBADOC.

Dans le 3eme chapitre nous mettons en evidence les differentes methodes d'accès aux donnees ainsi que les principaux criteres qui permettent de comparer ces methodes entre elles. Nous presentons ensuite, les criteres de choix de la technique d'acces utilisée dans notre systeme, pour rechercher l'information.

L'efficacité des methodes d'acces aux donnees depend en general de l'indexation de ces donnees. C'est-à-dire du choix des termes permettant de les retrouver. Aussi, le 4eme chapitre est-il consacré a l'étude des differentes rnethodes d'indexation, tout en presentant les criteres de choix de la technique d'indexation retenue pour notre systeme.

Le 5eme chapitre est dedie aux realisations et experimentations des techniques d'accès et d'indexation que nous avons définies. Nous y presentons le systeme d'accès aux donnees volumineuses, ainsi que l'intégration de ce dernier au systeme SIBADOC.

CHAPITRE I

Evolution des systèmes de gestion de données

Afin de situer le contexte dans lequel notre travail a ete realise, nous abordons dans ce chapitre les theme suivants:

- Les Systemes de Recherche Documentaire (SRD)
- Les Systemes de Gestion de Base de Donnees (SGBD).

1.1- Les systemes de recherche documentaire

La recherche documentaire englobe l'ensemble des techniques et modalites permettant de selectionner l'information dans un fonds documentaire moyennant des criteres specifies par l'utilisateur [DEW 83] [DEW 89]. son automatisation a ete entreprise pour differentes raisons parmi lesquelles nous citons:

- Impossibilite de canaliser le flux d'information a partir d'un certain seuil.

- Impossibilite de maîtriser les volumes de donnees des fonds documentaires, s'accroissant lineairement avec le temps.

- Impossibilite d'assurer une certaine precision informative a la recherche.

- Impossibilite d'accéder rapidement a l'information des lors que le volume couvrant l'information recherchee devient important et que la question devient complexe.

Cette informatisation de la recherche documentaire a conduit au developpement de systemes d'information spécialisés dans le stockage et la manipulation de donnees bibliographiques: Ce sont les systèmes de recherche documentaire.

La fonction essentielle des systemes de recherche documentaire consiste à stocker les documents sous forme d'un ensemble d'information caractérisant ces derniers, et a en permettre l'accès lors des recherches bibliographiques effectuées par les utilisateurs du systeme.

Les informations manipulees par le systeme ne sont pas les documents eux-mêmes, mais une description de ces derniers. Generalement, les systemes de recherche documentaire representent un document a l'aide d'une liste bibliographique, apelée notice bibliographique, et qui regroupe des informations telles que: nom d'auteur, references du document, titre, resume, annee d'édition, lieu d'édition, etc.

Les **SRD** gerent de grands volumes de donnees, ayant une representation condensee, comme par exemple des resumes associes a des listes de mots-cles appartenant a des vocabulaires contrôlés. Dans un SRD les insertions de documents sont realises par **"lots"**, par un administrateur du systeme et les suppressions sont tres rares, pour ne pas dire inexistantes. Les mises a jour de documents ne se presentent jamais.

L'offre en **SRD** connaît un veritable essor. Chaque annee des dizaines de systemes sont developpes. Parmi ces systemes nous avons le systeme **STAIRS [IBM 79], DIAMG [BOU 791, SMART** [SAL 71] [SAL 86a], etc. Helas, malgre la croissance rapide du nombre de ces systemes, la plupart d'entre eux, pour ne pas dire tous, utilisent la technique d'inversion classique pour acceder aux donnees bibliographiques.

1.2- Les systemes de gestion de base de donnees

Un systeme de gestion de base de donnees est aujourd'hui un logiciel de base essentiel dans un systeme informatique de gestion. Intuitivement, il permet a des utilisateurs concurrents de manipuler (insérer, modifier et rechercher) efficacement des donnees contenues dans une base de donnees. Depuis l'apparition des premiers SGBD vers 1962, d'importants resultats theoriques et pratiques ont ponctue l'histoire de la recherche en base de donnees. La mise en oeuvre de ces resultats a permis de faciliter l'administration et la manipulation d'une base de donnees, et d'accroître ainsi la productivite des utilisateurs des bases de donnees.

Les SGBD sont generalement classes selon le modele de donnees sur lequel ils sont bâtis. Leur histoire peut etre résumée en distinguant trois generations.

La premiere generation des SGBD est basee sur les modeles RESEAU et HIERARCHIQUE où les langages de manipulation des donnees sont de type procedural (navigationnel), autrement dit, l'utilisateur doit specifier le chemin d'accès aux donnees afin d'obtenir l'information desiree.

L'idée de la deuxième generation de SGBD est nee vers 1970 avec l'apparition du modele de donnees relationnel, défini par E.F CODD et base sur des fondements theoriques solides [COD 70] [GAR 85] [GAR 91]. Ceci a ouvert le champ a de nombreuses recherches de par le monde en vue de construire un systeme repondant aux normes du modele relationnel. Parmi ces projets deux systemes peuvent être consideres comme des references, il s'agit de SYSTEM-R developpe dans les laboratoires d'IBM [AST 76] et de INGRES chez BELL laboratory [STO 76].

Cependant, les premiers systemes relationnels n'ont ete commercialises que vers le debut des annees 80, avec l'apparition sur le marché du systeme SQL/DS [GAR 85] issu de sYS**TEM-R.**

Le modele relationnel témoigne d'une evolution des SGBD en parvenant a une plus grande independance des donnees par rapport aux programmes. Cette independance est d'une part physique : les applications ne doivent pas dépendre du mode de stockage des donnees, et d'autre part logique: les utilisateurs doivent pouvoir conserver leur vision propre des donnees contenues dans la base.

Les SGBD relationnels sont tres bien adaptes pour les applications de gestion classique, comme , par exemple, la comptabilité, la gestion de stock ou le contrôle d'inventaire. Cependant une tendance recente souligne les limites de ces systemes. Cette tendance est la manifestation de besoins pressants en gestion de donnees de la part d'applications nouvelles, telles que la bureautique, la conception assistée par ordinateur **(CAO)**, l'exploitation des bases de donnees spécifiques ou le genie logiciel, dont le besoin commun est la manipulation d'objets complexes (textes, graphiques, cartes, images, donnees multidimensionnelles), c'est-a-dire, des objets dont les attributs ne sont pas necessairement atomiques mais peuvent être eux mêmes des objets. Autrement dit, des donnees de tres grande taille ayant une sémantique riche et une structure complexe.
Par exemple, soit l'objet complexe LIVRE constitue d'un ensemble de chapitres et chaque chapitre contenant un ensemble de paragraphes. De tels objets peuvent être aisement modelises avec le modele relationnel en regroupant les objets de même type dans une même relation et en associant les sous-objets d'un même objet par des attributs de jointure.

La modelisation de l'objet complexe LIVRE nécessite alors, de façon tres simplifiée, les relations suivantes :

LIVRE(cote,titre,auteur) ;
CHAPITRE(num-chap,cote,titre-chap);
PARAGRAPHE(num-parag,num-chap,...)

ou cote et num-chap, respectivement, dans les relations CHAPITRE et **PARAGRAPHE** sont des attributs de jointure.

Un premier probleme avec l'approche relationnelle est que les liens sémantiques entre sous-objets sont perdus et doivent être explicites par des contraintes d'intégrité dont le maintient reste difficile et exige des algorithmes sophistiques [VAL 87].
Nous devons specifier dans le cas de notre exemple, que "cote" dans la relation **CHAPITRE** est une cle etrangere, ce qui garantit qu'un chapitre n'existe pas s'il n'est pas rattache a un livre particulier.

Un autre problème qui se pose, est que les performances de la manipulation des objets complexes sont faibles et souvent inacceptables. Par exemple, la reconstitution des informations concernant un livre exige, dans notre cas, deux jointures (operation coûteuse) des trois relations precitees. Generalement le nombre de jointures est tres important d'où un temps d'accès prohibitif.

Le support d'objets complexes par les **SGBD** relationnels actuels est difficile pour les raisons suivantes:

1- Les types de donnees supportes sont en general limités a quelques domaines numériques ou alphanumériques (par exemple, entier, reel et chaîne de caracteres).

2- Perte de semantique des objets complexes: Les structures de donnees fournies par le modele relationnel sont trop simples pour modeliser des objets a structure complexe comme une hiérarchie ou un graphe. Un objet complexe (par exemple, un objet hierarchique de **CAO)** peut dtre facilement decompose en relations. Cependant, cette solution pose un probleme sérieux. L'information semantique véhiculée par la structure complexe d'un objet est dispersee sous forme de valeurs dans differentes relations, imposant a l'utilisateur d'adapter sa vision a celle du SGBD.

3- Degradation des performances lors de la manipulation des objets complexes: la composition de donnees d'un même objet complexe peut exiger un grand nombre de jointures (une operation coûteuse en temps d'exécution).

Le modele relationnel etant incapable de prendre en compte la semantique des objets complexes, les chercheurs se sont tournes vers une troisième generation de SGBD, qualifies aussi d'avances, qui supportent bien ces applications nouvelles ou exploitent des environnements operationnels complexes. Cette nouvelle generation, initiée dans quelques laboratoires de recherche vers la fin des annees 70, fait maitenant l'objet de travaux de recherche et de developpement intenses.

Les SGBD avances ont pour ambition de lever les limites des SGBD relationnels actuels. Ils suscitent aujourd'hui de nombreux travaux de recherche et de developpement. Cependant, deux approches importantes peuvent être isolees: les systemes de gestion de bases de donnees orientees objets et les systemes de gestion de bases de donnees relationnelles etendues.

- Les SGBD orientes objets [BAN 88] [GAR 91] ont pour objectif essentiel le support d'objets complexes. Cette approche consiste a intégrer les techniques des langages de programmation orientes objets (comme C ++) et des bases de donnees afin d'offrir un langage unique et general pour programmer les applications traditionnelles et recentes des bases de donnees. De nombreux systemes orientes objets sont déjà operationnels. Parmi ces systemes nous avons les systemes **02** [LEC 88], **ORION** [KIM 87] et GBASE [GAR 91].

- Les SGBD relationnels etendus [GAR 89] [COD 79] cherchent a atteindre les objectifs des bases de donnees orientées objets tout en conservant la puissance et le support theorique du modele relationnel.

Dans le cadre des systemes gerant ces nouvelles bases de donnees, et des systemes de recherche documentaire, deux aspects paraissent essentiels: l'optimisation de l'espace de stockage et celle du temps d'accès a l'information qui font l'objet de notre travail.

CHAPITRE II

COMPRESSION DE DONNEES

11.1- Definition

On désigne par compactage de donnees toute technique qui reduit la taille de la representation physique des donnees tout en preservant un sous-ensemble d'informations. Par definition, une technique de compactage est dependante de la semantique des donnees.

La compression de donnees est une technique de compactage complètement reversible et qui a pour objectif la minimisation des quantités de donnees a stocker [GOT 75]. Elle est realisable par un changement de representation (codage) des donnees.

La compression de donnees offre de nombreux avantages parmi lesquels nous pouvons citer:

- La reduction des coûts de stockage et de transmission.
- L'amélioration des performances des systemes de gestion de l'information (reduction du nombre d'entrées/sorties physiques).

En contrepartie, l'utilisation de la compression de donnees implique certains inconvénients d'implémentation propres a des techniques particulieres telles que, la manipulation de bits, la representation a longueur variable des donnees, la maintenance du code, etc.

11.2- Concepts de base de la compression de donnees

La compression de donnees est une branche de la theorie de l'information qui a pour objectif la minimisation des quantites de donnees a transmettre. Celle-ci est realisee par un changement de representation (codage) des donnees.

Les elements de base de la theorie de l'information et de la codification sont presentes dans ce qui suit.

II.2.1- Le code et ses proprietes

Un code est une application d'un ensemble de messages sources, écrits dans un alphabet source, vers un ensemble de mots de code (mots-code) ecrits dans un alphabet code [HEA 72].

Exemvle: Le code **ASCII**

Alphabet source = {a..z, A..Z, 0..9, ...}
Alphabet code = {0,1}
Ensemble de mots de code = **(0000000 .. 1111111)**

Un code **cst** cnticrcmcnt dcfini par la donnée de trois elements;

- L'ensemble des messages sources a coder,
- L'ensemble des mots-code,
- La fonction de correspondance entre les elements de ces deux ensembles.

Les codes sont classes suivant la longueur des messages sources est des mots-code. On en distingue ainsi quatre categories:

- Bloc-bloc
- Bloc-variable
- Variable-bloc
- Variable-variable

- Bloc: Le message source (resp. mots-code) a une longueur fixe, la même pour tous les elements. Elle peut etre egale a 1 caractere, 2 caracteres ou plus (resp. un octet, 12 bits).

- Variable: La longueur des messages sources (resp. mots-code) est variable. Elle peut être une chaîne quelconque de caracteres (fragment, mot) (resp. une chaîne de bits, de quartets ou d'octets).

- **Proprietes d'un bon code**

1- Un code est distinct si tous les mots de code peuvent Btre distingues.

2- Un code est decodable d'une façon unique si dans toute sequence de mots de code le decodage d'un element ne donne lieu a aucune ambiguité.

3- Un code a la propriete prefixe si aucun mot-code n'est prefixe d'un autre. Sa decodabilite unique est assuree sans recourir a des informations supplementaires (autres que le mot-code en cours de decodage) pour lever les ambiguités dans le décodage d'une chafne.

4- Un code prefixe est minimal si pour tout **x**, si X est un prefixe d'un mot-code alors pour tout y appartenant a l'alphabet code, xy est un mot-code ou prefixe d'un mot-code. Cette propriete implique que le code a une longueur minimale.

Exemples :

1- Le code **C:** A ——> **B** (A={a,b,c}; **B={1,10000,100}**) est decodable de façon unique, mais le decodage d'une sous-chaîne commençant par 1 necessite la connaissance de ce qui suit le code pour lever l'ambiguité dans le decodage.

2- Le code *C:* **A** ——> **B** (A={a,b,c}; **B={1111,1010,00}**) verifie la propriete prefixe, aucune ambiguite ne peut apparaitre dans le decodage,

3- Le code C: **A** ——> B (A={a,b,c}; B={00,10,01}) vérifie la propriete prefixe mais n'est pas minimal car 1 est prefixe de 10 alors que 11 n'est ni mot-code ni prefixe d'un mot-code. Le code minimal serait **{00,01,1}.**

11.2.2- Quantite d'information et optimalité d'un code

La quantité d'information apportee par la réalisation d'un evenement a_i (dans notre cas, a_i est l'occurence d'un message dans le texte a comprimer) suivant une probabilité d'occurence $p(a_i)$, est mesuree par: $-Log_2 p(ai)$ [HEA 72].

Le contenu moyen en information d'un message d'une source (texte a comprimer) est donne par la sommation des quantités d'information apportees par chaque message ponderees par leurs probabilites d'occurence:

$$H = - SOM(\text{pour } i=1 \text{ a } N)\ (p(a_i) Log_2(p(ai)))$$ [HEA 721

ou N = nombre de messages distincts dans la source.

H est appelee l'entropie de la source.

L'optimalité d'un code est consideree dans le sens ou sa redondance est minimale. La redondance d'un code est definie par la difference entre sa longueur moyenne (L) et l'entropie de la source qu'il codifie [HEA 72].

$$R = L - H$$

où $L = \text{SOM}(\text{pour } i=1 \text{ a } N)\ (p(a_i) * l_i)$

l_i = longueur du code associé au message a_i.

L'efficacité d'un code est definie comme le complement de la redondance par rapport a L,

$$E = H/L$$

Un code est redondant si on a R>0, il est dit a redondance minimale si R est minimale (la longueur moyenne du code (L) proche de l'entropie (H))

Remarque :

Pour faire tendre L vers H (et donc minimiser R) il est preferable d'assigner des codes courts aux elements les plus frequents et de longs codes aux elements moins frequents. Quand les elements sont équiprobables la longueur des codes doit être la même pour tous les elements car il est plus avantageux de manipuler des codes a longueur fixe que des codes a longueur variable, etant donnée que l'efficacité des deux codes est presque la même.

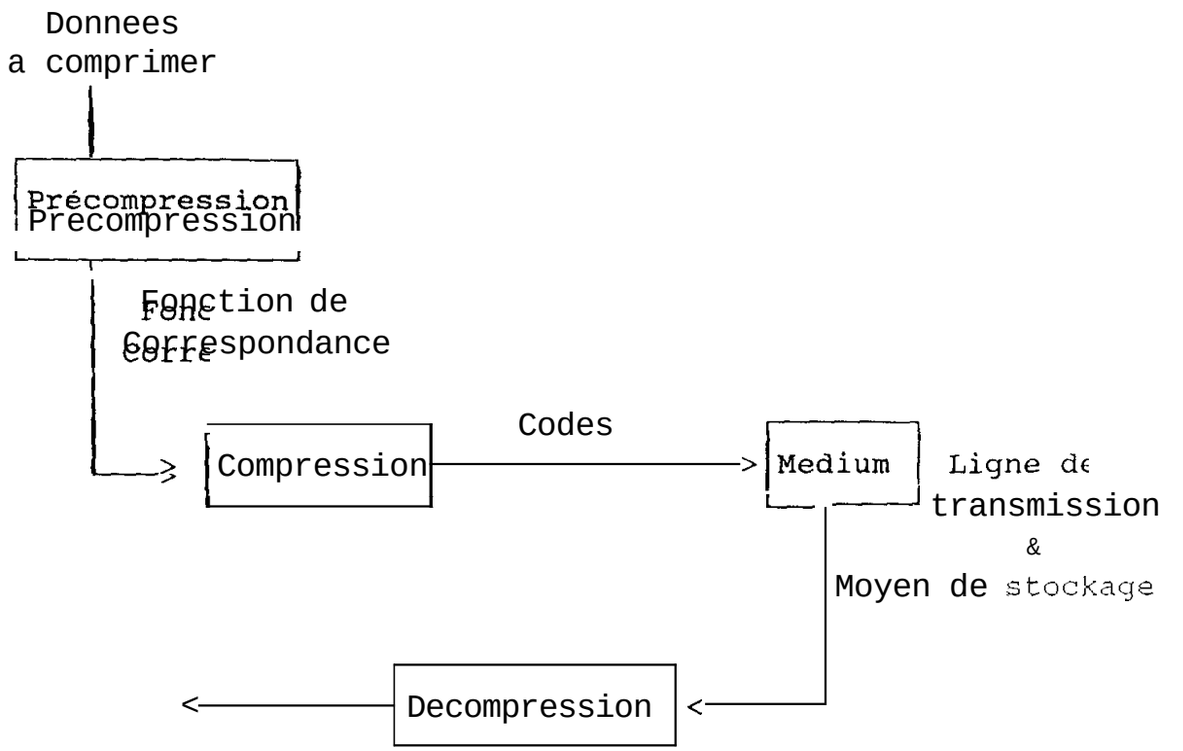
II.2.3~ Schema de compression de donnees

Ce schema réalise les fonctions de codage, de decodage et de construction du code. I1 doit être transparent à l'utilisateur, et est compose de trois principaux modules:

1- Le module de precompression, analyse le texte a comprimer (ou un echantillon representatif) et a partir de ses caracteristiques, il calcule la fonction de correspondance, c'est a dire définit le code.
Ce module peut être intégré au compresseur ou Btre independant de ce dernier.

2- Le module de compression (compresseur, encodeur) reconnait les messages delivres par la source et leur substitue dans la chaîne de sortie les codes equivalents.

3- Le module de decompression (decompresseur, decodeur) realise l'opération inverse en restituant les donnees initiales a partir des codes. Ceci peut Btre represente de la façon suivante:



11.2.4- Parametres d'évaluation des TCD

Différents parametres d'évaluation sont utilisés pour juger de l'efficacite des techniques de compression de donnees et pouvoir les comparer. Les deux parametres les plus significatifs sont :

a- Le taux de compression atteint : la formule la plus utilisée est celle donnant le gain realise par rapport a la taille originale du texte, exprimé en pourcentage.

$$T = (S - O) / S$$

ou: S est la taille du fichier original.
O est la taille du fichier comprime.

Le taux de compression differe d'une TCD a l'autre et varie generalement entre 40 et 90% [ALS 75].

b- Le temps de codage et de decodage

c- Les autres parametres sont: l'encombrement mémoire des programmes de compression/décompression et de la fonction de correspondance, la complexite de calcul des algorithmes (des trois modules), l'efficacité du code, etc.

II.3- Les redondances dans la representation des donnees

Toutes les TCD tentent d'améliorer la densité de stockage des donnees par l'élimination des redondances qui encombrent

la representation des donnees. Chaque TCD est conçue pour exploiter un certain type de redondance. Les TCD assez flexibles pour manipuler plusieurs types de redondances en même temps sont rares, et leurs resultats ne sont pas stables pour toutes les donnees [REG 81].
Les redondances les plus couramment rencontrees dans les fichiers et bases de donnees (donnees textuelles, numériques, alphanumeriques) sont:

- Une correlation significative existe entre des valeurs successives (dictionnaire de mots, thesaurus, ...).

- Longues sequences de caracteres repetees (blancs, zeros, ...).

- Certains caracteres se presentent avec une frequence bien supérieure a celle d'autres: non uniformite de la distribution statistique des caracteres.

11.4- Classification des techniques de compression de donnees

Les TCD peuvent être classees de différentes manieres suivant le critère de classification choisi.
Dans ce paragraphe, nous donnerons les deux classifications les plus usuelles. La premiere sera basee sur l'élément de base de la codification et la deuxième sera basee sur la fonction de correspondance (dependance de la fonction de correspondance des donnees, sa construction et son fonctionnement).

II.4.1- Classification par élément de base de la codification

Différents elements de base peuvent être pris comme unite de codification, le choix de ceux-ci conditionne la performance et la complexite de la TCD:

11.4.1.1- Les caracteres

Les TCD basees sur les caracteres sont nombreuses et ont l'avantage d'être simples et non encombrantes du point de vue espace (table de Codage/Décodage tres limitée). Cependant, l'utilisation du caractere comme unite de codage implique l'emploi de codes a longeur variable, ce qui est tres contraignant.

Pour des donnees textuelles les performances de telles TCD sont limitées a moins de *50%* [PIK 81].

11.4.1.2- Les bigrames

Le codage se fait au niveau des caracteres, mais l'alphabet est augmentee d'un certain nombre de bigrames (paire de caracteres les plus frequents).
Ces bigrames, seront codes par les combinaisons (ASCII, EBCDIC) non utilisees par l'alphabet initial.

Ce type de TCD est simple a implémenter et réalise un taux de compression de 43 a **48%** [WIS 87]. I1 est important de signaler que dans cette methode le rapport taux de compression/coût est intéressant. Par consequent ces TCD sont preferees aux precedentes dans le cas des donnees textuelles.

Cependant, le probleme reside dans le choix des bigrames a coder, ce qui necessite toute une analyse des textes a comprimer.

Notons que ce principe peut être etendu a des trigrames et des quadrigrames, mais le gain obtenu est trop faible par rapport au coût, au fur et a mesure que la taille des chaines de caracteres augmente [WIS 87].

11.4.1.3- Les fragments

Les TCD basees sur les fragments representent une extension de celles basees sur les bigrames. Dans celles-ci il y a creation d'un dictionnaire de fragments (chaînes quelconques de caracteres) les plus frequents, et le texte sera code comme une suite de fragments.

Ces techniques de compression donnent de bons resultats avec les donnees textuelles, mais leur coût peut être important. En effet, la taille du dictionnaire doit être assez grande pour permettre des taux de compression eleves, les algorithmes de precompression sont tres complexes (apres analyse du texte il faut définir l'ensemble optimal de fragments). Le même probleme se rencontre avec les algorithmes de compression car ils doivent determiner le decoupage optimal du texte en fragments [MEC 89].

11.4.1.4- Les mots

Pour les techniques basees sur les mots, le texte est code comme une suite de codes de mots. L'inconvénient de ces TCD est la taille du dictionnaire de mots qui est tres grande

(sauf si celui-ci ne contient que les mots ayant une frequence assez elevee, auquel cas une deuxième TCD sera utilisée pour le codage de mots les moins frequents [MEC 89] [HEA 72].

11.4.2- Classification suivant la fonction de correspondance

Suivant ce critère de classification, les TCD se repartissent en deux groupes [REG 81] [MEC 89], selon qu'elles dependent ou non des donnees a comprimer.

11.4.2.1- Les TCD semantiquement dependantes

Une TCD est semantiquement dependante si le codage qu'elle utilise depend du contexte et de la semantique des donnees a comprimer. Parmi ces TCD, nous avons la differenciation, le codage des repetitions [MEC 89], la compression des index [MEC 89] [WIS 86], etc.

- **La differenciation**

Une valeur est remplacee par sa difference avec une information de reference. La representation des differences est moins encombrante que celle des valeurs originales, surtout quand elle sont tres proches.

Avant compression	Apres compression
Compression Compilation Codage	Compression 4ilation 2dage

- **Codage des repetitions**

Une longue sequence du même caractere est remplacee par un triplet contenant un indicateur de repetition, un compteur et le caractere repete.

Exemple :

Avant compression	Apres compression
122222	**1*52**
76999999	76*69

11.4.2.2- Les TCD semantiquement independantes

Une TCD est dite semantiquement independante ou generale, si elle n'utilise aucune information liée aux donnees.

Cette generalite est limitée a une classe de donnees, ce qui assouplit la TCD pour plus d'efficacité dans l'implémentation de celle-ci.

Les TCD semantiquement independantes peuvent être utilisées avec n'importe quelles donnees, mais avec un degre d'efficacite variable. Parmi elles nous avons la TCD basee sur le code de HUFFMAN [MEC 89] [PEC 85], et celle basee sur le code de QOC (a Quasi-Optimum Code) [MEC 89] [FUM 86] [FUM 82].

11.5- Choix d'une TCD pour la compression des donnees bibliographiques

Les donnees contenues dans une base bibliographique se repartissent en trois types:

- **Donnees numériques:** Elles se trouvent dans les champs nombre de pages, numero de volume, issn,isbn, etc.

- **Donnees textuelles:** Elles sont contenues dans les champs titre, resume et mot-clé.

- **Donnees nontypees:** Cette classe regroupe toutes les rubriques qui ne contiennent ni du texte ni du numerique pur, tels que les champs auteurs, titre de conference, lieu de conference, etc.

Etant donnée la diversite des types de donnees constituant les elements des fichiers bibliographiques, une TCD generale considerant le fichier comme des donnees homogenes risque de ne pas être efficace [MEC 89]. La meilleure alternative serait donc un schema de compression dans lequel nous combinerons trois codages distincts , chacun adapte a un type de donnees (numerique, textuel et nontype) et pour lequel il pourrait donner les meilleurs resultats. Ce schema doit respecter le profil suivant [MEC 89]:

1- La fonction de codage (3 fonctions) est calculee a partir d'un echantillon representatif de la base documentaire.

2- Il doit réaliser le meilleur compromis entre un taux de compression très élevé et un temps de decompression tres bas

(pour ne pas augmenter sensiblement le temps de reponse du systeme). La rapidité de la fonction de codage est secondaire car cette operation n'est effectuee que rarement (lors de la mise a jour mensuelle de la base).

3- L'espace mémoire necessaire au stockage de la fonction de codage/décodage doit être negligeable devant le gain realise.

11.5.1- Compression des donnees numeriques

Le codage des donnees numeriques est base sur les caractères [MEC 89]. L'expérience réalisée sur la base INSPEC donne un taux de compression egal a 65%.

11.5.2- Compression des donnees nontypees

Cette classe de donnees est subdivisée elle-même en sept groupes:

1- Les dates
2- Les numeros de pages
3- Les numeros de volume
4- Les champs d'identification
5- Les champs de contrble
6- Les noms de personne
7- Les lieux et organisation

Par consequent, la technique préconisée est composee elle aussi de differents codages [MEC 89]. En effet, chaque groupe de rubriques prenant leurs valeurs dans le même domaine sera code par une technique adaptee au domaine. Les resultats obtenus pour ce type de donnees sont resumes dans le tableau suivant:

Groupes de donnees	Taux de compression
- Non typees	
. Dates	66,47%
. N° de pages	57,30%
. N° de volumes	71,87%
. Champs d'identification	42,88%
. Champs de contrôle	56,66%
. Noms de personnes	47,93%
. Lieux, organisations	45,13%

11.5.3- Compression de donnees textuelles

Les donnees textuelles sont contenues dans les champs: titre, resume et mots cles, et occupent la plus grande proportion du fichier bibliographique. Par consequent, si on veut avoir un taux de compression élevé pour le fichier bibliographique, il faut choisir un bon codage pour les donnees textuelles, c'est-a-dire un codage qui permet d'avoir un taux de compression eleve. Pour ce faire, deux techniques ont ete experimentees:

1- La premiere est basee sur les fragments, différentes variantes ont ete testees en jouant sur la taille du code, les valeurs *8*, 9, 10, 11, 12 bits ont ete retenues ce qui a fixe le nombre de fragments codes a 256, 512, 1024, 2048, 4096.

2- La seconde est basee sur les mots. Dans cette technique les mots a coder sont tries par ordre décroissant des frequences et chaque mot recevera comme code son rang dans l'ordre établi.

L'implémentation de ces TCD a donne les resultats resumes dans le tableau ci-dessous:

TCD	Taux de comp. (%)	Temps de codage ms/car	Temps de decodage ms/car	Encombrement mem Ko	
				codage	decodage
F256	45,93	0,053	0,025	3	2
F512	48/54	0,122	0,105		
F1024	53,34	0,120	0,092		
F2048	53,60	0,124	0,091	26	26
F4096	56,28	0,128	0,081	52,5	50
Mot	71,23	0,392	0,146	93	96

Remarques:

1- L'encombrement memoire est d'autant plus important que le taux de compression est grand. Aussi nous remarquons que le codage des fragments sur un octet **(F256)** est le moins encombrant du point de vue espace memoire.

2- Pour le temps de decompression la TCD **F256** est la plus efficace car elle ne necessite pas la manipulation de bits. Le temps de decompression de la TCD basee sur les mots est le plus important car elle travaille sur des codes a longueur variable, les autres TCD ont des temps de decodage tres proches.

3- Le temps de compression est tres important dans la TCD basee sur les mots car le codage implique une recherche dans une table. Ce temps est le même pour les autres TCD sauf la F256 qui reste la plus rapide.

4- Le taux de compression realise est nettement plus important pour la TCD basee sur les mots. En effet, ce type de TCD offre un taux de compression appreciable (premier critere de choix d'une TCD) par rapport a celles basees sur les fragments, et un taux de decodage acceptable (il peut être optimisé par la programmation du decodeur en assembleur).C'est pour toutes ces raisons que nous l'avons choisie, dans le cadre de notre systeme, pour la compression des donnees textuelles [MEC 89].

11.6- Integration de la technique de compression de donnees au systeme SIBADOC

Le Systeme d'Interrogation des BAses DOCumentaires SIBADOC a ete conçu et developpe au CERIST dans le but de prendre en charge la gestion et l'accès a l'information documentaire contenue dans des bases de donnees bibliographiques internationales.

SIBADOC a ete developpe sur vax-11/785 et est constitué de deux principaux modules [BEN 89]:

- le module de creation et de mise a jour des bases documentaires,

- le module d'interrogation pour la recherche

Une presentation detaillee de SIBADOC est donnée en annexe A.

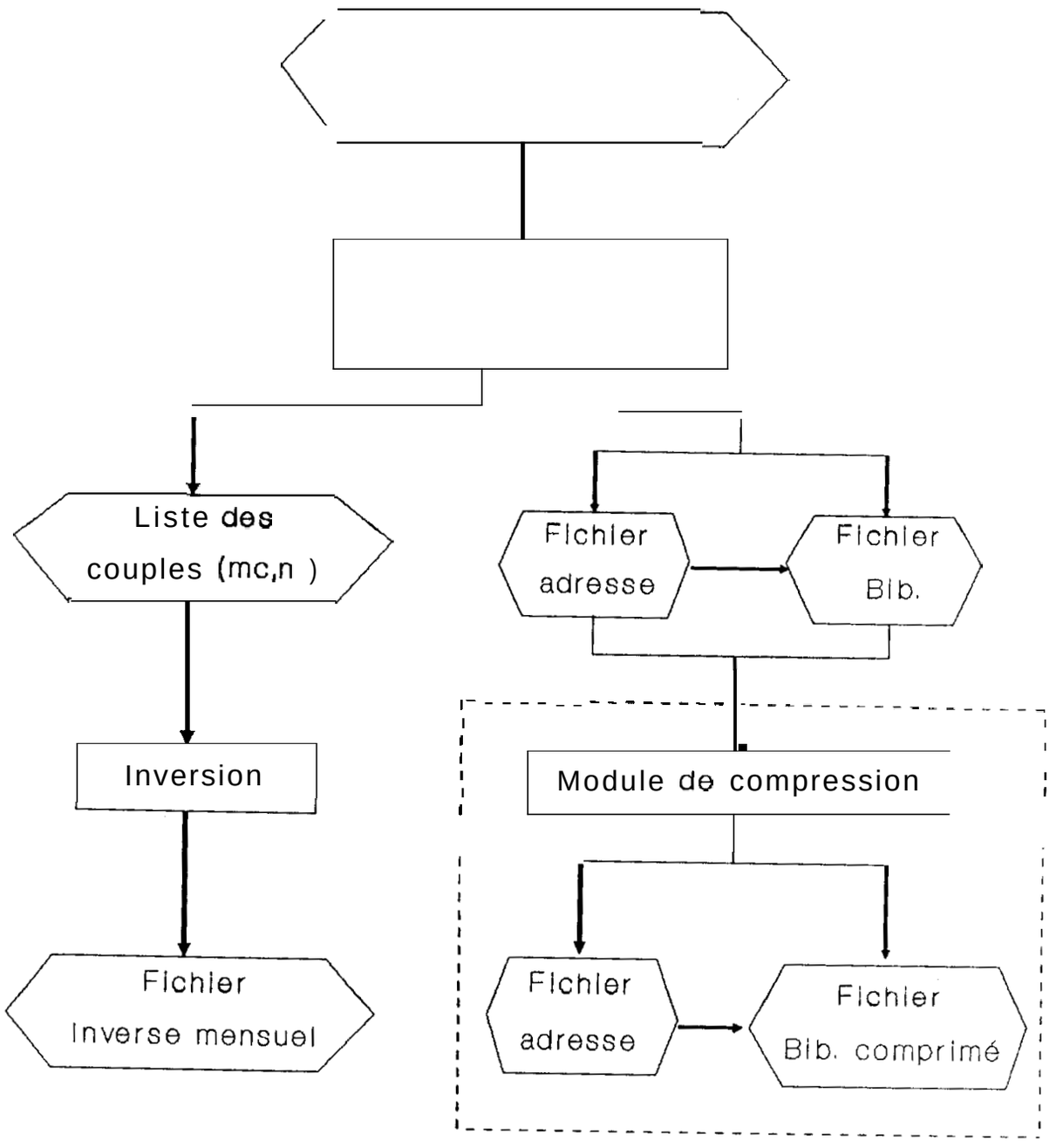
Apres avoir choisi la TCD appropriée aux donnees bibliographiques, cette derniere a ete integree a titre experimental au système SIBADOC dans le cadre d'un projet de fin d'études [MEC 89]. Le gain réalisé par le module de compression est satisfaisant, puisque ce dernier a permis un taux de compresion de 64% sur la base INSPEC.

Nous presentons dans ce qui suit la localisation dans le systeme SIBADOC des actions de compression et de decompression. Cette localisation est tres simplifiée etant donnée la structure modulaire du système SIBADOC.

II.6.1- La compression dans SIBADOC

Le module de compression est intégré au module de creation et de mise a jour et il n'intervient qu'à la fin de cette operation. Cet algorithme réduit considerablement la taille des notices et met à jour leurs adresses dans le fichier adresse correspondant.

Le schema d'exécution de ce module devient :

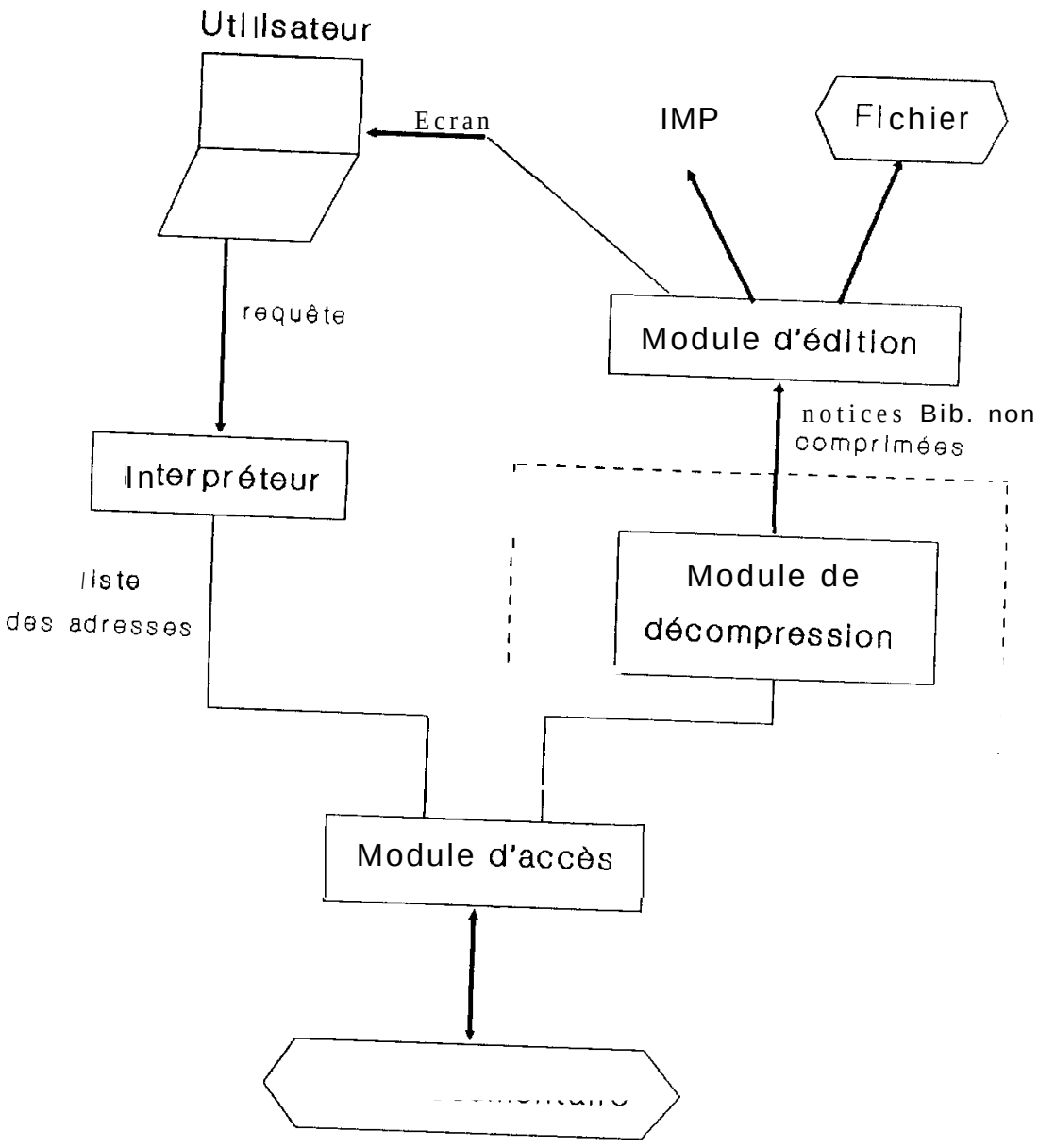


Module de creation et mise a jour de la base avec compression de donnees.

11.6.2- La decompression dans SIBADOC

Le module de decompression est integre au module d'interrogation de la base, il sera execute immediatement avant l'édition des notices demandees par l'utilisateur.

Le processus d'interrogation se schematisera alors comme suit:



Module d'interrogation de la base comprimée

Nous terminons la presentation de l'intégration de la technique de compression de donnees au systeme SIBADOC en donnant des resultats par rubriques obtenus dans la compression du fichier du mois de Janvier de l'année 1989, de la base INSPEC en exploitation sous SIBADOC, ainsi que les resultats de la compression de toute la base INSPEC (6 mois sont en exploitation).

Groupes de donnees	Taux de compression
- Numériques	65,57%
- Textuelles	71,21%
- Non typées	
. Dates	**66,479**
. N° de pages	**57,30%**
. N° de volumes	71,87%
. Champs d'identification	**42,88%**
. Champs de contrble	56,66%
. Noms de personnes	**47,93%**
. Lieux, organisations	**45,13%**
Moyenne de compression	64,10%

Resultats par rubrique de la compression de INSPEC0189

Fichier INSPEC	Taille du fichier avant compression	Taille du fichier apres compression	Taux de compress.
Janvier	**31066**	**10732**	65,45%
Février	**29512**	10254	65,26%
Mars	**20432**	**7136**	65,07%
Avril	**35640**	**1251%**	**64,88%**
Mai	**34334**	11982	65,10%
Juin	**42312**	**14734**	**65,18%**

Resultats de cornpression de la base INSPEC

Apres cette etude des methodes de compression de donnees existantes, nous allons passer a la determination d'une methode d'accès optimale et qui s'adapte aux donnees comprimees. Ceci fera l'objet du chapitre suivant.

CHAPITRE III

TECHNIQUES **D'ACCES**

Dans la littérature, diverses techniques d'accès aux donnees textuelles ont ete proposees. L'objectif de ce chapitre est d'effectuer une etude exhaustive des differentes techniques d'accès et de proposer une technique optimale d'accès aux donnees volumineuses.

111.1 **Description des differentes techniques d'accès**

Les techniques d'acces aux donnees se focalisent toutes autour des grandes classes suivantes :

III.1.1 - **Recherche** Séquentielle (Full **Text** Scanning)

Etant donne un échantillon de recherche, la base de donnees sera entierement consultée jusqu'à ce que les documents en question soient retrouvés. Cette méthode ne necessite pas d'espace overhead et ne demande pas beaucoup d'effort pour l'insertion, ni pour la mise a jour [FAL 84]. Cependant, le principal inconvenient est le temps clr recherche. Bien qu'il existe que1ques a1gorithmes de recherche d'une chaine de caracteres qui sont rapides [BOY 77] [KNU 77] [AHO 75] [MAL 71], la recherche dans une grande masse de donnees peut necessiter un temps considerable [HOL 79].

III.1.2- Inversion

Plusieurs systemes, tels que STAIRS [RAB 4] [IAE 82], LEXIS et SIBADOC ont adopte cette approche.

Cette methode utilise un index. Une entree de cet index consiste en un mot (ou theme ou concept), suivi d'une liste de pointeurs. Ces pointeurs donnent l'adresse des documents qui contiennent ce mot.

Le principal avantage de la methode est sa rapidité de recherche. Cependant, elle peut necessiter un large espace overhead de stockage pour l'index. Cet espace peut aller jusqu'à 3 fois la taille du fichier contenant les donnees si les informations de localisation des mots sont egalement pris en compte [FAL 84] [FAL 85].

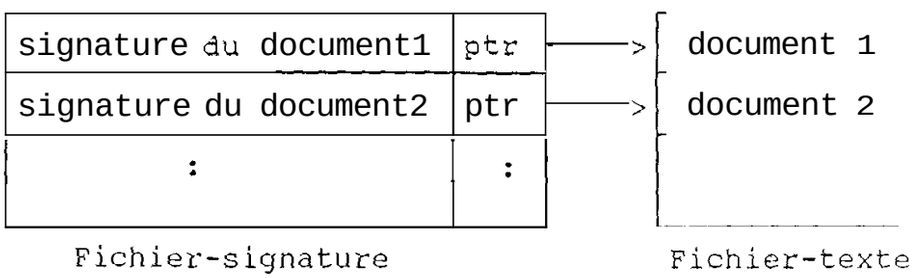
111.1.3 - Regroupement (Clustering)

Dans cette technique d'accès, les documents similaires sont regroupes ensemble pour former des groupes, appeles clusters, conformement au principe de regroupement qui stipule que les documents tendant a répondre aux mêmes questions sont étroitement lies, c'est-à-dire qu'ils appartiennent au même groupe [EAS 89] [BEL 87].

I1 semble difficile de comparer cette methode avec les precedentes, car le regroupement se fait selon le sens des requêtes. Par exemple, "donnez-moi les documents qui sont en rapport avec la recherche d'information (même si les documents ne contiennent pas les deux mots 'Recherche' et 'Information'). I1 semble que l'espace overhead est petit, de l'ordre de logn, ou n est le nombre total de documents. De même, le temps de recherche est de l'ordre de logn. Par contre, cette methode semble ne pas pouvoir gérer facilement les insertions [FAL 84].

111.1.4 - Fichier-signature (signature File)

De par le monde, de nombreux travaux concernant les signatures et l'accès utilisant les fichier-signatures ont été réalisés [RAB 84] [ROB 79] [SAC 87] [KUO 82] [TAV 92]. Dans cette technique d'accès, les documents sont stockes sequentiellement dans un fichier texte. A Chaque document est associé un mot-code appeli signature du document, obtenu en appliquant une fonction de hachagc à *ce* dernier. Les mots-code sont de même taille (F bits) et sont stockes sequentiellement dans un fichier **appele** fichier-signature.



Quand une requête arrive, au lieu de consulter chaque document, on consultera la forme condensee de ces documents, c'est-a-dire le fichier-signature. De ce fait,un grand nombre de documents qui ne satisfont pas a la **requête seront &cartes, et par consequent,** le texte d'un document ne sera vraiment consulte que s'il y a une forte probabilité d'y retrouver l'information desiree. Le fichier-signature peut être alors considere comme un resume ou des mots cles.

Cette methode est plus rapide que celle du 'Full text scanning', mais elle est supposee être plus lente que l'Inversion [FAL 84]. Cependant, la methode des signatures necessite moins d'espace overhead que celle de l'Inversion et peut gerer facilement les insertions [CHR 84] [JOH 69] [TSI 82].

Cette breve discussion sur les methodes de recherche textuelle laisse voir qu'aucune de ces methodes n'est clairement meilleure que les autres. Par consequent, nous allons proposer deux autres techniques d'accès qui s'intègrent tres bien aux donnees comprimees et dont l'objectif est d'accélérer la recherche d'information :

111.1.5 - Methode d'accès utilisant un Fichier-signature

Dans cette technique, des listes inversees sur des partitions comprimees de la base de donnees seront utilisées.

L'inversion se fait au niveau des symboles (mots ou fragments de mots), utilises pour la compression de donnees [GOY 85]. Cette methode s'intéresse donc aux symbolcs qui ont la particularité d'être tres frequents, etant donne qu'il existe une forte correlation entre les mots sélectionnés par les indexeurs et ceux qui apparaissent de manière frequente dans un texte [KEV 83]. La base de donnees consistera donc en des pages comprimees auxquelles sont associés des index (un index par page).

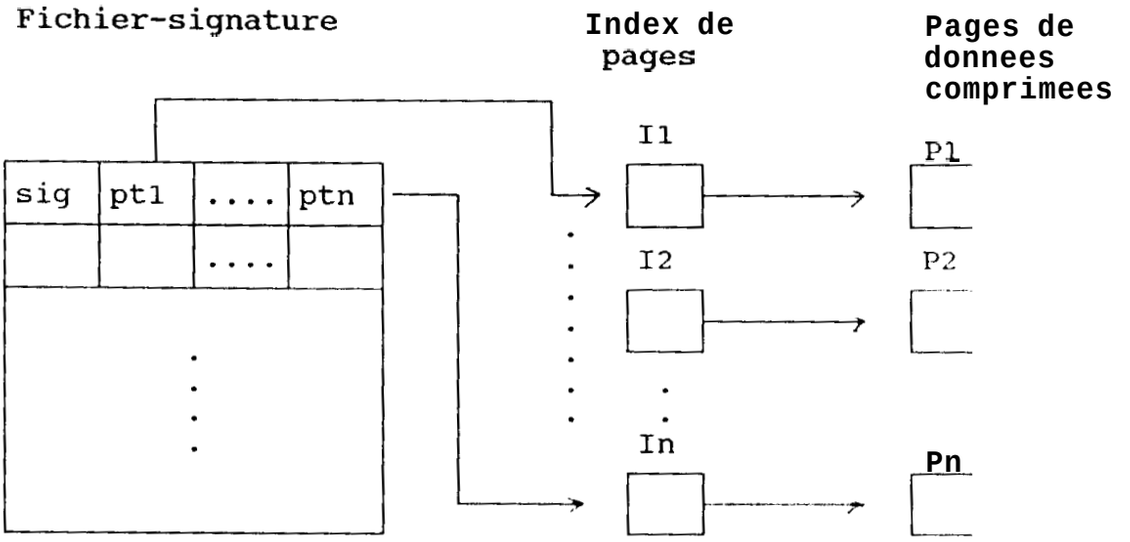
Pour rechercher une chaîne 'S', celle-ci est eventuellement divisée en plusieurs symboles 'S1''S2'...'Sm'

tel que chaque 'Si' appartienne a l'ensemble des symboles utilises pour la compression. Ces 'Si' sont utilises pour determiner les index en question.

Pour augmenter les performances de la recherche, en plus des index de pages, un fichier-signature contenant non pas des symboles, mais les mots-code (ou signatures) de ces symboles, sera utilisé [GOY 85] (voir fig 111.1). Ces signatures sont de même taille (f bits) et sont obtenues en appliquant une fonction de hachage a l'ensemble des symboles contenus dans les differents index de pages.

Le fichier-signature contiendra donc les signatures de tous les symboles contenus dans les differents index de pages (une signature par symbole), ainsi que des pointeurs vers les differents index qui contiennent ces symboles.

Ainsi, au lieu de parcourir tous les index de pages pour determiner ceux qui contiennent un symbole donne, on consultera le fichier-signature, et a l'aide des pointeurs, on selectionnera un sous-ensemble d'index de pages pour la recherche, ce qui augmente les performances de la technique d'acces.



signature d'un symbole	pointeur vers index1	...	pointeur vers indexn

Une entree d'un indexi

symbole j	adresse de la notice contenant le symbole j dans la page i

Fig 111.1: Acces en utilisant un fichier-signature

111.1.6 - Méthode d'accès utilisant un Q-arbre

III.1.6.1 - Definition d'un Q-arbre

Un Q-arbre est un arbre équilibré comme les B-arbres [BAY 72]. En effet, tous les chemins de la racine aux feuilles sont de même longueur. La difference entre un Q-arbre et un B-arbre est que la profondeur d'un Q-arbre est fixée a 1, et la taille de ses pages est variable (la taille de la racine est differente de celle des feuilles) [FUM 86].

La technique du Q-arbre est applicable pour n'importe quel index qui satisfait les conditions suivantes:

a) A la construction de l'index, il existe déjà un grand ensemble de mots. Ces mots sont appeles mots primaires.

b) Les mots qui vont être ajoutés a l'index seront appeles mots secondaires.

c) Aucun mot primaire ne sera supprime, car la suppression d'un mot necessite la reorganisation des feuilles du Q-arbre. De plus, les Q-arbres prennent en compte le fait que les mots cles sont rarement supprimes dans la plupart des systemes de recherche d'information.

d) Le nombre de tous les mots (primaires et secondaires) est superieur a plusieurs millions.

111.1.6.2 - Description du Q-arbre

Un element de l'index est un couple (w,q).
ou w : est le mot qui constitue la cle de recherche.
q : est un pointeur vers les informations qui sont associees a w.

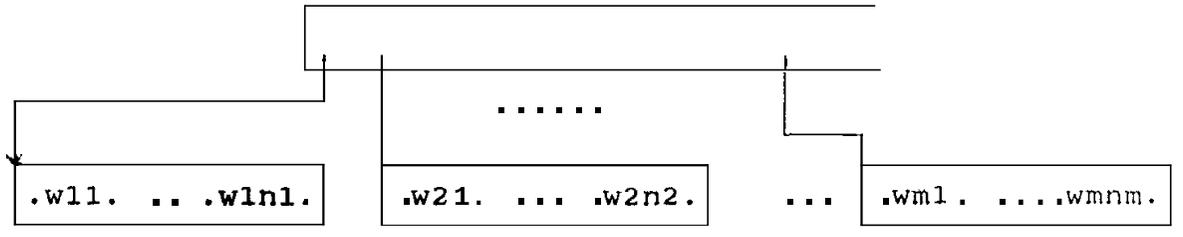
Toutes les feuilles du Q-arbre ont la même taille et sont stockees en memoire secondaire.

La racine reside en memoire centrale et peut ne pas avoir la même taille qu'une feuille. En fait, la taille de la racine depend de l'espace memoire dont le systeme peut disposer.

Les mots dans la racine et les feuilles sont tries dans l'ordre alphabetique.

Tous les mots secondaires sont stockes dans les feuilles du Q-arbre.

Contrairement aux feuilles, la racine n'a pas d'espace pour les ajouts. De plus, la racine ne contient que des mots primaires et vu les conditions d'application de la technique du Q-arbre (cond. c), ces mots ne seront pas supprimes.



Arbre équilibré de profondeur 1

111.2 - Critères de choix de la technique d'accès

C'est l'organisation du fichier inverse de mots cles qui determine l'efficacité d'un systeme de recherche d'information.

Concernant la technique d'accès utilisant un fichier-signature a la place de ce dernier, nous aurions pu utiliser la technique d'indexation traditionnelle; . c'est-à-dire utiliser un autre niveau d'indexation et qui sera un index global contenant tous les symboles au lieu de leurs signatures, mais l'apport de la methode des signatures en gain d'espace mémoire est tres important en comparaison avec celui des techniques d'indexation usuelles. Neanmoins, cette technique presente toujours des inconvenients du fait qu'elle necessite un grand espace overhead dû a l'utilisation *de* deux niveaux d'indexation: le fichier-signature et les index de pages. De plus, l'utilisation d'une fonction de hachage pour l'obtention des signatures peut engendrer des problèmes de collision, ce qui augmente le nombre de reponses non pertinentes par rapport aux questions et par consequent, influe sur l'efficacité du systeme.

D'autre part, le facteur principal qui determine l'efficacité d'un arbre équilibré est sa profondeur et elle est minimale dans un Q-arbre. La structure du Q-arbre semble donc être l'organisation la plus appropriee qui permet d'avoir un index rapide [MAT 81].

Comme l'index de mots cles satisfait les conditions d'application de la technique du Q-arbre, nlors, c'est cctte structure qui a été adoptée dans notrc système pour l'index de mots cles.

La solution retenue en utilisant un Q-arbre, est donc de placer les mots les plus frequents dans la racine du Q-arbre, ce qui diminue le nombre d'accès disque pour la recherche, et de placer le reste des mots dans les feuilles du Q-arbre.

L'ensemble des mots de depart, qu'ils soient stockes dans la racine (mots les plus frequents) ou dans les feuilles, seront appeles alors mots primaires.

La determination de l'ensemble des mots utilisés par le Q-arbre se fait par une methode d'indexation dont l'étude et le choix feront l'objet du chapitre suivant.

CHAPITRE IV

INDEXATION OU SELECTION DE MOTS CLES

L'indexation consiste a trouver ou a selectionner le mot vedette, appele aussi mot clé, qui est censé traduire le mieux le sujet principal dont traite chaque document [LON 80].

L'indexation est un art qui reclame beaucoup de qualités, souvent contradictoires de la part de l'indexeur humain [CHA 90]: comprehension du domaine, comprehension de la langue, esprit analytico-synthétique, etc.

Les produits de l'indexation sont des index, c'est-à-dire des listes de termes significatifs, qui servent a retrouver les informations et a les selectionner afin de répondre aux besoins des utilisateurs.

Comme toutes les activites d'information, l'indexation est d'abord un instrument de travail. Elle doit permettre de retrouver les informations utiles pour des categories bien précises d'utilisateurs effectuant des travaux determines. l'indexation doit donc Qtre défini de telle sorte que ces besoins puissent etre satisfaits au moindre coût.

De plus, la qualification et les effectifs du personnel disponible, le volume d'informations a traiter et leur nature, le systeme de stockage et de recherche de l'information, la nature de la matérialité des produits et des services de diffusion de l'information, les moyens financiers disponibles imposent egalement des limites dans le choix d'une methode d'indexation.

IV.l- Les methodes d'indexation

Du point de vue de sa finalité, l'indexation est destinée a perrnettre une recherche retrospective efficace des informations contenues dans un fonds documentaire.

L'indexation est donc la tâche la plus importante dans un systeme de recherche documentaire. En effet, c'est elle qui conditionne l'efficacité de ce dernier. Une mauvaise indexation ou une indexation insuffisante represente 90% des causes essentielles de l'apparition, lors d'une recherche, de bruits ou de silences [CHA 80].

Le bruit est le taux de documents parasites qui ne repondent pas a la question posee et qui sont extraits lors des operations de selection. Si pour 100 documents obtenus en reponse, 20 sont jugés ne pas repondre a La question, l'on dit que le bruit est de 20%.

Le silence est le taux de documents pertinents existant dans la base documentaire qui devraient repondre a une question et qui n'ont pas ete selectionnes lors de l'interrogation. Le silence est dû a l'emploi de caractéristiques non concordantes dans la formulation de la question et dans l'indexation du document. Un systeme donnant un taux de silence d'environ 30% est generalement considere comme satisfaisant.

IV.l.l- Indexation sans thesaurus et indexation avec thesaurus

IV.l.l.l- Definition d'un langage documentaire

Un langage documentaire est un ensemble de termes utilisés pour representer le contenu des documents [CHA 73] [RIV 90].

Les elements constitutifs d'un langage documentaire sont les suivants [GUI 90]:

- Des mots servant a decrire les informations, les descripteurs. Ils sont tires du langage naturel, reduits a une

forme grammaticale unique et invariable (generalement le substantif singulier). Ils peuvent Qtre simples ou composes.

- Des mots du langage naturel ayant des relations avec les descripteurs et qui sont repertories avec un renvoi au descripteur correspondant. Ces mots ne peuvent Qtre utilisés pour decrire les informations. On les appelle des non-descripteurs.

- Des relations entre les descripteurs: relations hiérarchiques, d'équivalence ou de voisinage, etc. Elles permettent de regrouper les notions sous un seul terme, d'élargir ou, au contraire, de preciser une recherche.

IV.1.1.2- Definition d'un thesaurus

Le thesaurus represente un mode d'organisation d'un langage documentaire. C'est un repertoire de mots qui ont entre eux des relations (hierarchique, voisinage ou equivalence, synonymie, ...), dans un domaine particulier dc connaissance.

Le thesaurus est utilisé par les indexeurs au moment oh les concepts d'indexation doivent être transformes en descripteurs [GUI 89].

Les concepts font partie du langage naturcl, les descripteurs du langage contrôlé du thesaurus.

Selon qu'il dispose ou non d'un thesaurus, l'indexeur aura recours a des procédés differents:

IV.1.1.3- Indexation sans thesaurus

Cette methode est frequemment appelee methode d'indexation en "langage naturel" car le rôle du mot cle (ou descripteur) est devolu, non par comparaison a une liste de mots cles existants deja, mais a tout candidat mot cle.

Cette methode offre une grande souplesse et une simplicité de mise en oeuvre, mais presente les inconvenients suivants:

- Tres coûteuse (en espace mémoire a cause de la grande taille des fichiers).
- On peut aboutir rapidement a une dispersion catastrophique du lexique. Pour y remedier, il faut au moins:

* eliminer les mots non significatifs appelés aussi mots vides (articles, prepositions, ...).

* réduire a une seule réalisation morphologique tous les mots presentant des variantes paradigmatiques (nombre, genre, conjugaison, déclinaison).

IV.1.1.4- Indexation avec thesaurus

Dans cette methode, l'indexeur recherche si les mots cles qu'il a repéré existent dans le thesaurus, sinon il les traite comme des candidats descripteurs en exerçant les dcux contrôles suivants:

1- Examen de la liste des candidats descripteurs et refus de certains d'entre eux (soit par mise a jour de l'ensemble des

mots non significatifs, soit par rejet pur et simple, soit par insertion dans une liste d'attente), car l'apparition du candidat descripteur est estimee comme etant tout a fait accidentelle ou non significative dans le corpus donne.

2- Examen de la liste des candidats descripteurs et remplacement de certains d'entre eux par des descripteurs estimés avoir une signification identique ou une signification proche et ceci en etablissant les relations classiques (hiérarchie, synonymie, voisinage, association, ...) entre descripteurs.

L'indexation avec thesaurus nous conduit donc a une liste structuree de descripteurs. Plus cette liste est structuree moins il est exigé d'aptitudes de l'utilisateur, qui peut etre non spécialisé et plus le systeme de recherche documentaire sera performant en qualite [DEW 81] [DEW 85].

IV.1.2- Indexation manuelle et indexation automatique

Il existe deux principales methodes d'indexation liées d'une part a l'importance du fonds documentaire et d'autre part a la disponibilité d'un equipement informatique:

1- Indexation manuelle
2- Indexation automatique.

IV.1.2.1- Indexation manuelle

L'indexation manuelle est une indexation faite par des indexeurs humains, spécialistes des domaines abordes dans les documents.

L'indexation manuelle est une solution satisfaisante dans un contexte particulier [TIG 92]:

1- Nombre de documents peu élevé.
2- Textes ne se trouvant pas sur support informatique.

Cependant, cette methode d'indexation s'avère inadequate dans les grands centres de documentation où le volume de l'information est tres important. De plus, ce type d'indexation presente les inconvenients suivants [CHA 89] [TIG 92] :

1- Rigidité de l'analyse et des concepts qui y découlent.
En effet:

* Indexation figee

* Mise a jour impossible, surtout dans des domaines qui evoluent tres rapidement tels que l'intelligence artificielle ===> effet de décalage entre le thesaurus et le contenu des documents.

* Manque de souplesse face a l'évolution dynamique des besoins des utilisateurs.

2- Qualite de l'indexation réduite a cause:

* De l'effet de variabilité ou de non uniformité : deux indexeurs choisissent rarement exactement les mêmes termes pour indexer un même document. On observe, parfois, l'absence de certains descripteurs que l'indexeur aurait dû retenir pour rendre compte du contenu.

* Du bruit (informations inutiles) et du silence (perte d'informations) documentaire qui sont consequence de l'absence d'uniformité dans une indexation manuelle.

Les difficultés liées au processus d'indexation manuelle ainsi que les volumes tres importants d'informations a traiter obligent et justifient l'apparition de systemes d'indexation automatique de documents.

IV.1.2.2- Indexation automatique

L'indexation automatique [SAL 89] consiste a faire reconnaitre par l'ordinateur des mots figurant dans le titre, le résumé ou le texte lui même d'un document.

Les termes reconnus sont incorporés dans le fichier de recherche et servent a retrouver le document.

L'indexation automatique suppose que l'on introduise dans l'ordinateur sinon tout le texte, du moins le titre et un resume d'auteur. L'ordinateur utilise alors differentes approches pour identifier les termes significatifs.

IV.1.2.2.1- Approche statistique

Dans cette approche on procéde a des analyses statistiques sur un échantillon de textes pour determiner la frequence d'apparition des mots. Le petit nombre qui apparaît constamment est considere comme sans signification. Un second groupe comprend des mots apparaissant moins souvent et considérés comme significatifs.

Un troisième groupe comprend des mots qui apparaissent très rarement et sont consideres comme trop spécifiques. A partir de la, on peut indiquer a l'ordinateur quelle doit être la frequence des mots a retenir pour l'indexation.

La faiblesse de cette methode, qui n'est applicable que dans un domaine donné, tient a ce qu'elle considère chaque mot isolement et permet seulement de dire qu'un concept ou un objet figure dans le document, sans preciser son rôle. Cette methode ne tient pas comptc, par exemple, du probleme **pose** par la synonymie. Une même information peut apparaitre plusieurs fois dans un document, mais sous des formes differentes. Cela sera ignore dans le calcul, fait par l'ordinateur, de frequence d'apparition d'un mot. Des dictionnaires de synonymie peuvent être etablis mais ils alourdissent le systeme [GUI 90]. Ainsi, le calcul de la frequence s'effectue sur le concept et non sur le mot.

IV.1.2.2.2- Approche linguistique

Vue l'insuffisance de l'approche statistique et la nature des informations traitees (langue naturelle) dans les systemes de recherche documentaire, un modele linguistique pour determiner les concepts importants d'un document, s'impose naturellement.

Les methodes linguistiques introduisent des traitements morpho-syntaxiques pour arriver a une reconnaissance des structures les plus significatives. L'indexation s'effectue alors en trois grandes phases [LAL 87] [MET 88] **[COY** 67] **[ROU** 87]**:**

- La premiere consiste a determiner les unites lexicales du document a analyser. C'est l'analyse morphologique.

- La deuxième consiste dans la reconnaissance semantique des termes retenus. Cela revient a classer ces derniers en differents groupes, selon qu'ils soient significatifs ou non. C'est l'analyse semantique.

- La troisième et dernière phase consiste a établir des liens entre les descripteurs pris deux a deux issus des deux premieres etapes. C'est l'analyse syntaxique.

L'indexation automatique vise donc a determiner les concepts importants d'un document en examinant la suite de caracteres qui forment le texte. Pour ce faire, il faut:

1- Bien connaitre les mécanismes du langage qui permettent de passer de la forme au sens.

2- Etre capable de simuler ce processus sur ordinateur.

Cependant, ces deux conditions ne sont pas remplies car la question qui se pose est de savoir comment simuler un processus, qui permet de passer de la forme au sens, sur ordinateur, si on ignore les mécanismes fondamentaux de ce processus. Ceci a amené certains chercheurs a dire que la veritable indexation automatique n'est qu'à l'état de projet de recherche [MET 88].

IV.2- Critères de choix de la methode d'indexation

Les problemes essentiels de la documentation sont des problemes sémantiques. L'analyse et l'indexation des documents

en vue de leur mémoristion forment un goulot d'étranglement dans tous les systemes documentaires (la production documentaire augmente plus vite que le nombre d'indexeurs qualifies) et cet obstacle doit être surmonte pour que le transfert de l'information s'effectue dans de bonnes conditions. Ainsi differentes voies faisant appel aux ressources de l'informatique ont ete explorees pour tenter de suppleer a l'indexation humaine (manuelle) et la remplacer:

- d'une part avec les systèmes dits en texte intégral ou en "full text", systemes supprimant la phase d'indexation et reposant sur une stratégie de recherche complexe.

- d'autre part avec les methodes d'indexation automatique en amont de systemes documentaires devenus classiques.

IV.2.1- Les systemes en texte integral ou systemes en "full text"

Ces systemes dits en "full text" ou encore "free text" ne font appel a aucune indexation, le texte integral du document etant memorise et la recherche portant sur l'ensemble du texte et non plus sur les descripteurs [CHA 73] [CHI 86] [SAL 86b].

Generalement les systemes en texte integral portent non pas sur le document lui-même, sauf s'il est court (dépêche de presse, coupures de presse, articles de loi, jurisprudence, etc), mais sur le resume plus ou moins developpe du document.

Dans de tels systemes utilisant le langage naturcl, la recherche doit porter sur l'ensemble du texte memorise, aucune selection de termes ni aucune indexation n'ayant etc operees.

Un element important qui rend ces systemes tres interessants est le fait qu'ils ne comportent aucune indexation; ce qui permet d'éviter d'avoir recours a des indexeurs humains dont le travail n'est jamais entierement satisfaisant (& IV.1.2.1). Cependant, une evaluation détaillée de ce type de systemes, faite dans [BLA 85] et [MAR 90], montre que ces systemes ne sont performants que pour des bases de donnees non volumineuses. De plus, ces systemes ne reposant sur aucune indexation, utilisent des strategies de recherche tres complexes telles que [CHA 73]:

- Utilisation de masques sur les mots du texte.
Par exemple, en écrivant dans 1'equation de recherche DOCUMENTS, la recherche portera sur tous les mots ayant cette racine. I1 sera alors possible de selectionner les documents ayant 1'un des termes:

* DOCUMENTS
* DOCUMENTATION
* DOCUMENTAIRE
* DOCUMENTALISTE

- Utilisation d'une logique de recherche tres developpee: possibilités d'utiliser de nombreux extenseurs ou restricteurs (plus petit que, plus grand que, compris entre, adjacent, avec, etc).

Exemple:

La formulation TRAITEMENT AVEC DOCUMENTATION permettra de selectionner les documents ayant une phrase telle que: Le TRAITEMENT par des methodes informatiques de la DOCUMENTATION scientifique.

- Possibilité d'interrogation sur des zones determinees, et ceci en utilisant des indicateurs de zones (zone auteur, zone bibliographie, etc).

Exemple:

En écrivant une question:
CONTROLE ZONE **BIBLIOGRAPHIE** SALTON
On fera apparaître tous lea documents citant SALTON dans lours references bibliographiques.

IV.2.2- Méthodes **d'indexation automatique**

Comme nous l'avons vu au chapitre IV, l'indexation automatique se resume en deux principales methodes [SAL 70]:

- methode statistique
- methode linguistique

IV.2.2.1- Methode statistique

Les methodes d'indexation automatique les plus anciennes sont les methodes statistiques [ROU 87] [CHA 73]. Avec ces methodes, les mots les plus representatifs sont extraits par un calcul de frequence.

L'un des inconvénients majeurs de ces methodes reside dans la non prise en compte, pour le calcul de la frequence, des synonymes. Plusieurs methodes sont utilisées pour tenter de remédier a cet inconvenient. La plus simple consiste a dresser un dictionnaire de termes synonymes, de maniere à ce que le calcul de la frequence soit effectue sur le concept et non sur le mot.

Pour affiner ces methodes statistiques, envers lesquelles de nombreuses critiques ont ete formulees, d'autres elements sont parfois pris en compte, en affectant des poids aux termes [LAM 82] [COO 78] ou en utilisant des operateurs linguistiques [CHA 73]. Ainsi il a ete propose de retenir en priorité comme termes significatifs ceux qui sont precedes d'expressions comme: en resume, en conclusion, etc. Toutefois, le probleme crucial et qui restera toujours pose, est la determination du seuil **de frequence a** partir duquel nous déciderons si un mot est significatif ou non [ROU 87].

IV.2.2.2- Methode linguistique

Devant les insuffisances des methodes statistiques, d'autres methodes ont ete experimentees, faisant appel aux ressources de la linguistique. Ces methodes, necessitent au depart un long travail de preparation d'outils d'analyses, constitution de dictionnaires, construction de reseaux scmantiques, elaboration de grammaires, etc. Or ceci fera l'objet de tout un projet de recherche, ce qui nous ai impossible de traiter en detail dans le cadre de cette these.

IV.3- Conclusion

Apres cette longue discussion sur les methodes d'indexation qui laisse voir que la meilleure approche pour indexer des documents est l'approche linguistique, mais qui n'est pas realisable dans le cadre de cette these, la methode qui a ete choisie pour indexer notre corpus documentaire, se resume dans un premier temps a extraire les termes du fonds documentaire et a éliminer ceux qui sont non significatifs par consultation d'un dictionnaire de mots vides.

Les mots vides representent l'ensemble des mots qui ne sont pas significatifs pour une recherche retrospective et ne figurent donc pas dans le fichier inverse d'une referotheque [DEW 81] [DEW 89] [GET 87] [MER 84].

Le choix des mots vides est d'une importance directe avec la taille de l'index ainsi qu'avec la qualité du systeme.

Examinons quelques ensembles de mots vides :

1- Ensemble des 'n' plus frequents mots.

'n' etant un parametre du systeme qui doit être choisi de sorte que la taille du fichier inverse soit réduite et que l'efficacité de recherche du systeme ne soit pas deterioree.

2- Ensemble des articles, prepositions, conjonctions, pronoms et adverbes.

Théoriquement, en utilisant cet ensemble de mots vides, la taille du fichier inverse peut être réduite de 60% (approximativement) avec 2% de perte en efficacite de temps de recherche [FUM 86].

Pour une telle methode de selection de mots vides, une reduction de 40% est au minimum garantie [FUM 86].

CHAPITRE V

SYSTEMES D'ACCES AUX DONNEES VOLUMINEUSES

Dans ce chapitre nous presentons le systeme d'accès aux donnees volumineuses que nous avons conçu, son integration au systeme SIBADOC, ainsi qu'une evaluation des techniques d'indexation et d'accès utilisées.

V.1- Description du systeme d'accès aux donnees volumineuses

Les traitements du systeme d'accès aux donnees volumineuses ont ete envisages comme un ensemble de modules effectuant, chacun une tâche spécifique.

Les principaux modules developpes sont les suivants:

1- Indexation
2- Acces aux donnees volumineuses
3- Mise a jour de la Base de Donnees Documentaires (BDD)

V.l.l- Module d'indexation

Une base de donnees documentaire est une collection de notices bibliographiques, chacune etant associee à un document (livre, these, article,...) . Une notice contient toutes les informations qui permettent l'identification du document et de son contenu. Ces informations se presentent sous forme de texte de taille indéfinie et sont contenues dans differents champs, tels que le champ titre, auteur, resume, etc. Le nombre de champs descriptifs dans une notice n'est pas fixe, il varie d'un type de document a un autre.

Apres avoir choisi la technique d'indexation appropriée, nous allons voir quelles sont les options a indexer. En effet, il existe **deux** approches pour indexer les informations

contenues dans une base de donnees documentaires ou referotheque:

- La premiere consiste a indexer chaque mot qui apparait dans chaque champ d'une notice bibliographique et à constituer un lexique general unique. C'est le cas du systeme d'inversion totale présenté dans [GET 87].

- La seconde approche n'indexe que certains mots qui apparaissent dans certains champs et construit ainsi des lexiques specialises.

La premiere approche assure une recherche de haute precision, car elle permet de retrouver chayue mot paru dans la referoteque. Toutefois, cette methode génère des index de mots de tres grande taille [BLU 87].

La seconde approche limite l'indexation a certains types de mots de certains champs et permet une recherche sur la base des critères qui correspondent aux champs des mots qui sont indexes.

Comme le systeme d'accès aux donnees volumineuses est oriente vers une recherche de haute precision et comme la technique du Q-arbre est applicable pour des index de tres grande taille, alors c'est la premiere approche qui a été retenue pour la conception et la réalisation du module d'indexation. Cependant, dans le cadre de notre projet et pour une premiere version du systeme, nous allons nous limiter aux champs titre, resume et mot cle. En general, les resumes ont plus de mots cles que les titres. Par consequent, si les mots cles sont sélectionnés a partir des resumes, la taille du fichier inverse devient importante et depasse celle du fichier

de donnees lui-même. Et si les mots cles sont selectionnes uniquement a partir des titres, alors, beaucoup de documents pertinents par rapport a la question et qui existent dans le fonds documentaire ne peuvent être retrouves, ce qui donne un taux de silence élevé lors de la recherche [CHA 80] [DAL 89]. Par consequent, la methode la plus appropriee et realisable qui permet d'extraire les mots cles a partir d'une qrande masse de donnees, est celle de la suppression de mots vides.

Ce module se resume donc a:

1- Extraire tous les mots des champs titre, resume et mot cle, et ceci, pour l'ensemble des notices du fichier bibliographique.

Dans cette etape nous indiquons aussi pour chaque mot:

- sa frequence d'apparition dans la referotheque
- les informations de localisation du mot: le nombre et les adresses des notices qui le contiennent.

2- Filtrer les mots. Pour ce faire, nous appliquerons la notion de mots vides a la liste des mots obtenue a la premiere etape, et ceci en se réfèrant a une liste de mots vides prédéfinie, établie a partir de l'ensemble des articles, prepositions, conjonctions, etc.

A la sortie de ce module, nous aurons une liste complete et ordonnée de tous les mots contenus dans les trois champs titre, resume et mot cle et qui vont être reellement indexes, ainsi que les listes des notices qui les contiennent (une liste par mot). I1 s'agit d'une **inversion totale.**

V.1.2- Module d'accès aux donnees

L'objectif de ce module est d'acceder aux donnees d'une referotheque en utilisant la technique du Q-arbre. Pour ce faire, il se divise en deux principaux sous-modules:

- Le premier sous-module consiste en l'implémentation du Q-arbre a partir de la liste des mots fournie par le module d'indexation. Pour cela, nous allons prendre parmi la liste des mots a indexer, ceux qui sont les plus frequents et nous les mettons dans la racine du Q-arbre; ils seront donc en memoire centrale et le reste des mots sera place dans les feuilles du Q-arbre, c'est-à-dire en memoire secondaire.

Le nombre de mots a mettre en memoire centralc sera un parametre a determiner en faisant un compromis entre l'espace memoire occupe et le temps de reponse du systeme. En fait, cela depend essentiellement de la taille de la memoire qui sera mise a la disposition du systeme. Toutefois, cette taille pourrait être augmentee en utilisant une methode de compression au Q-arbre.

- Le deuxieme sous-module établit une procedure de recherche dans le Q-arbre qui permet, pour un mot clé donne, de selectionner les references bibliographiques contenant ce dernier.

Le schema suivant est une illustration du chemin suivi dans le Q-arbre, pour l'exécution d'une commande de recherche:

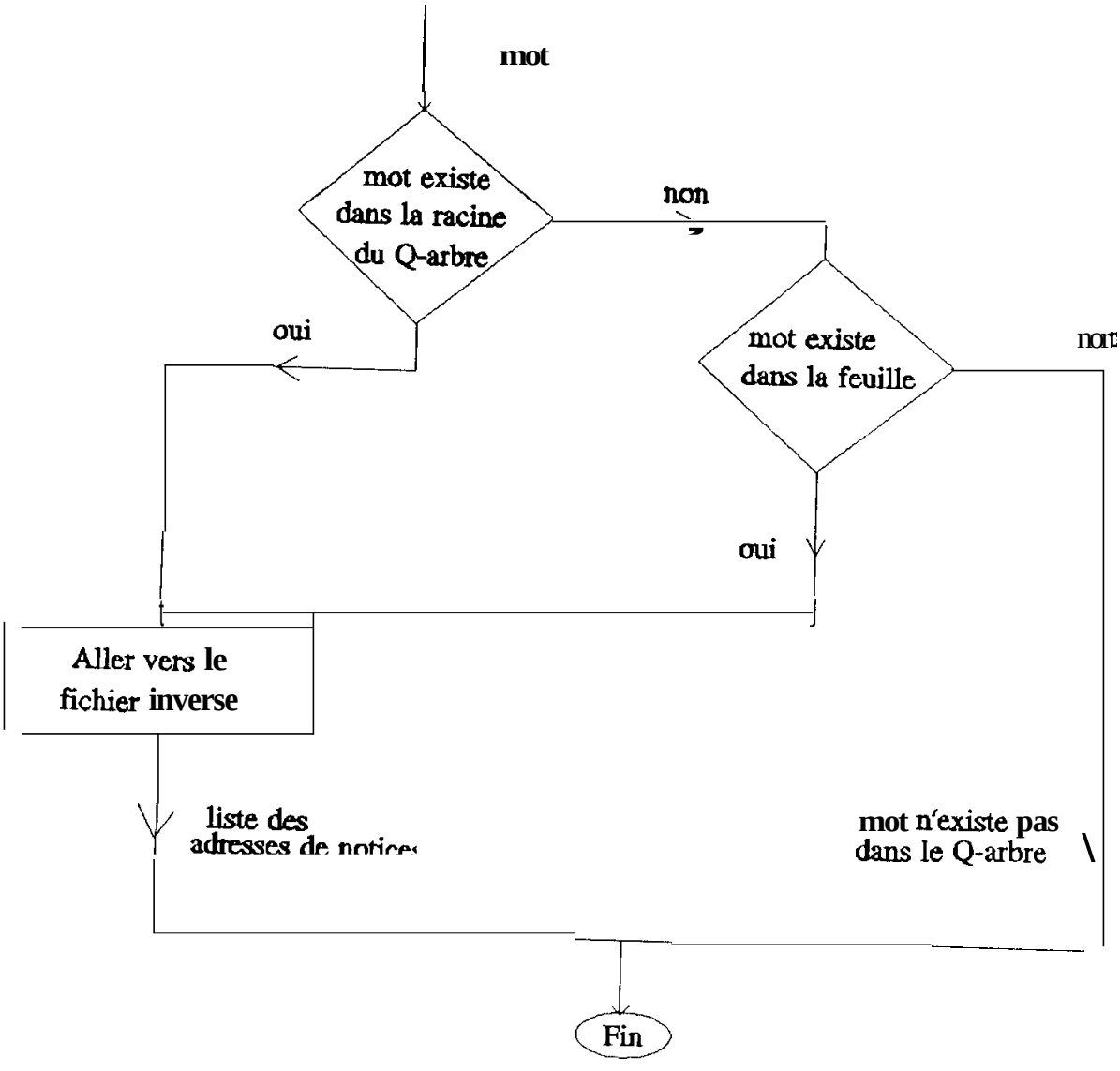


Schéma d'exécution d'une commande de recherche

V.1.3- Module de mise a jour de la BDD

La base annuelle d'informations est completee périodiquement par l'adjonction de la base documentaire mensuelle. Par consequent, le but de ce module est de mettre a jour mensuellement les divers fichiers rassemblant les informations sur lesquelles porteront les interrogations, a savoir, les fichiers inverse et lexique associes a la base annuelle, ainsi que les structures de donnees associees a la technique d'accès, c'est-à-dire au Q-arbre.

V.2- Integration du systeme d'accès au systeme SIBADOC

Etant donnée la structure modulaire du systeme documentaire SIBADOC (voir annexe a), l'intégration du systeme d'accès a ce dernier n'a pose aucun probleme d'implémentation.

Nous presentons dans ce qui suit la localisation dans le systeme SIBADOC des modules du systeme d'accès aux donnees volumineuses.

V.2.1- Module d'indexation

Le module d'indexation est intégré au module de creation et mise a jour et il intervient au debut de cette operation. L'indexation agit sur le fichier bibliographique et permet d'extraire pour chaque notice l'ensemble des mots apparaissant dans les champs titre, resume et mot cle. L'indexation permet, ainsi, de creer une liste de couples, chacun etant constitué d'un mot cle et de l'adresse de la notice contenant ce mot cle.

Ce module consiste donc a créer pour chaque fichier bibliographique mensuel un fichier lexique contenant l'ensemble des mots extraits a partir des notices bibliographiques et un fichier inverse correspondant, contenant les adresses de ces notices.

V.2.1.1- Structure du fichier lexique

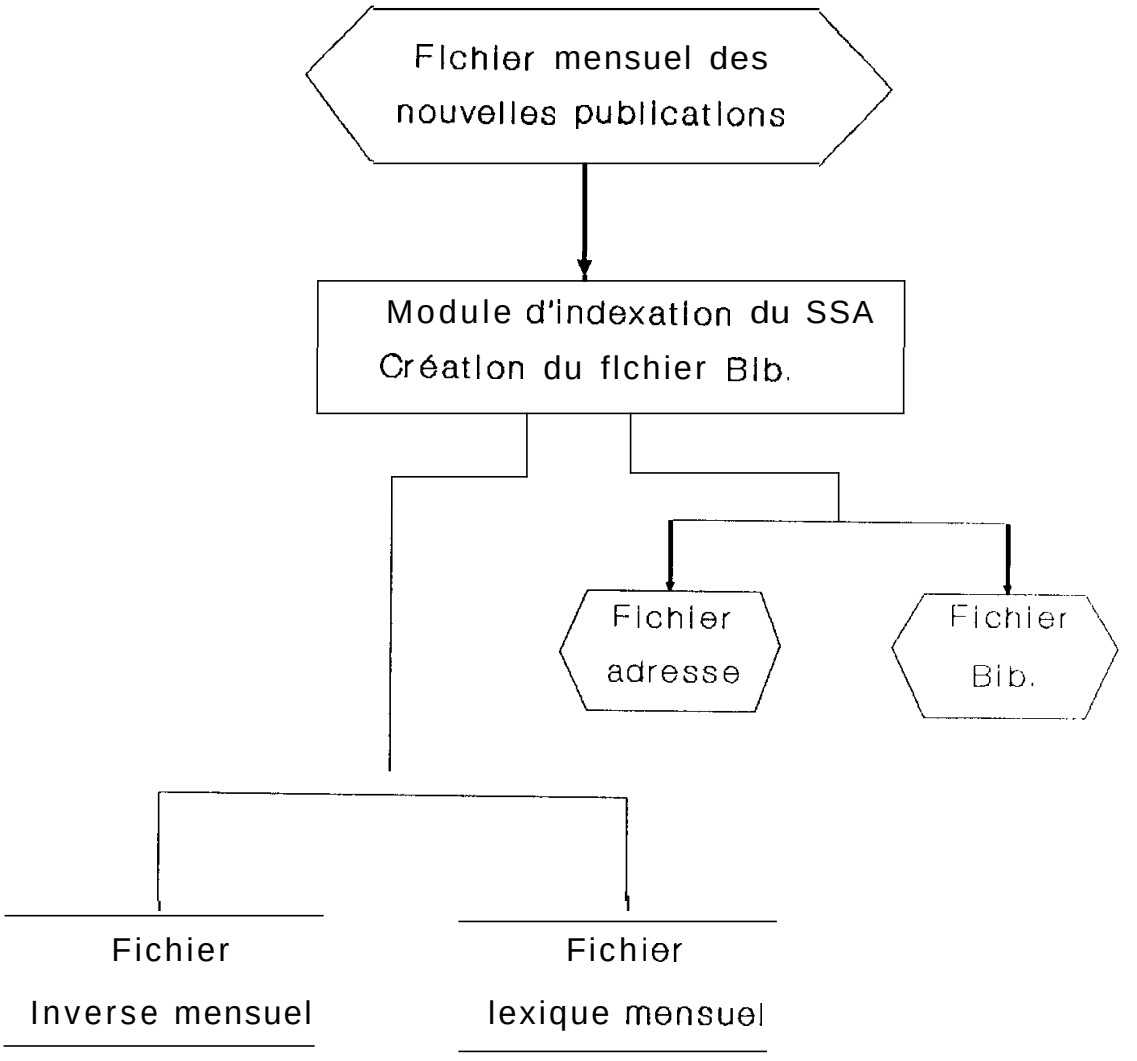
Le fichier lexique est un ensemble d'enregistrements presentes en ordre alphabétique et contenant chacun un mot cle et d'autres informations concernant ce dernier telles yue sa longueur, sa frequence d'apparition dans le corpus documentaire, un pointeur vers le fichier inverse, etc.

Une organisation en "séquentiel indexé" de ce fichier permet un acces rapide a l'information.

V.2.1.2- Structure du fichier inverse

I1 contient pour chaque mot cle du fichier lexique une liste d'adresses des notices bibliographiques contenant ce mot clé.

Le schema d'exécution du module de creation et de mise à jour du systeme SIBADOC devient:



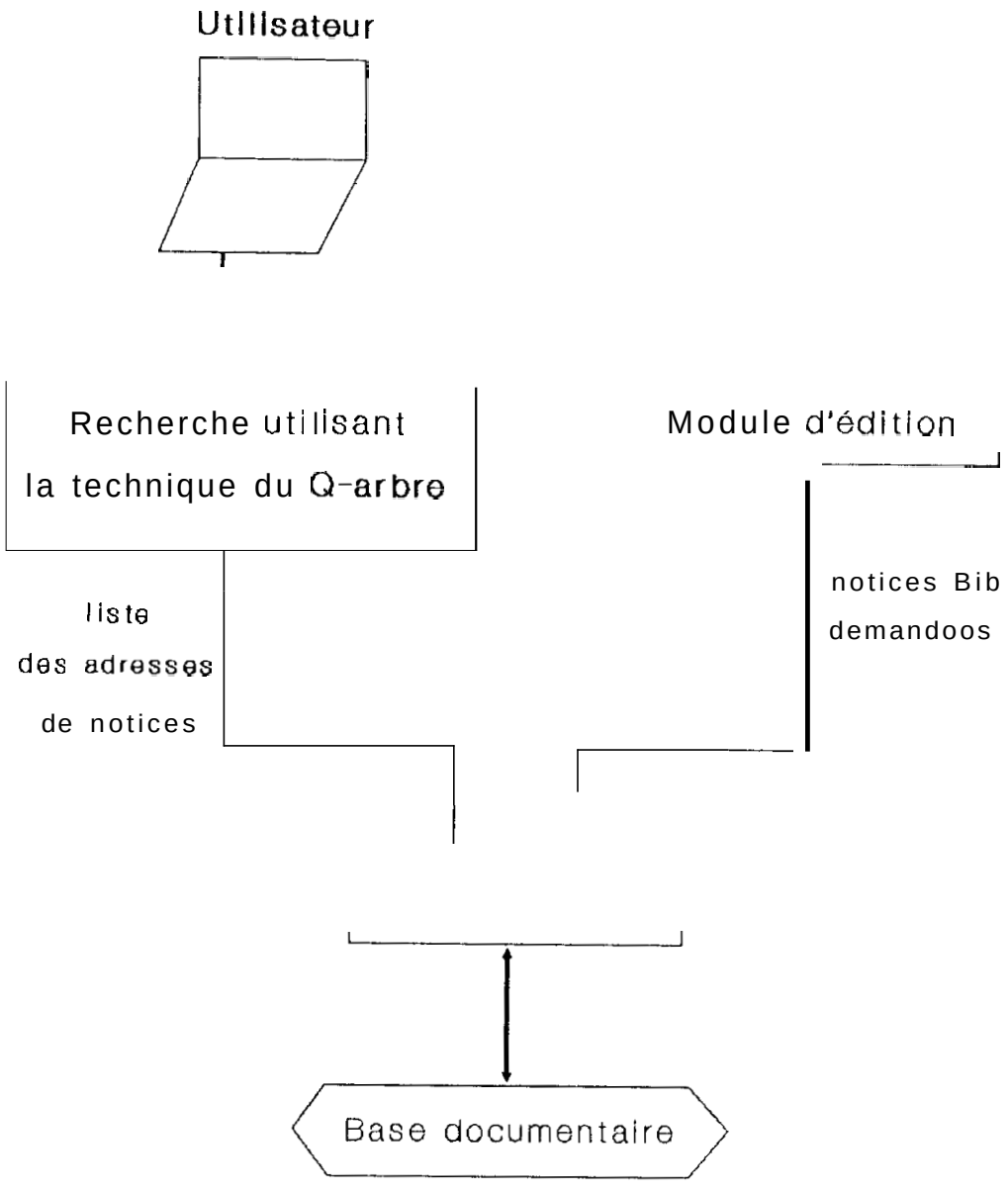
Module de creation et mise a jour

de la base avec indexation du sous systeme d'acces

V.2.2- Module d'accès aux donnees

Ce module qui utilise la technique du Q-arbre pour l'exécution d'une commande de recherche, est intégré au module d'interrogation de la base et plus precisemsnt au niveau de l'interpréteur de commandes.

Le processus d'interrogation n'ayant subi aucun changemcnt du paint de vue etapes de traitement, il se schematisera comme suit:



Module d'interrogation de la base utilisant la technique d'accès du Q-arbre

V.2.3- Module de mise a jour de la BDD

Ce module est integre au module de creation et mise a jour du systeme **SIBADOC**. I1 est exploitable en mode batch, ainsi que tous les autres modules du systemes d'accès aux donnees volumineuses, a l'exception du sous-module de recherche documentaire qui lui est executable en mode interactif.

V.3- Evaluation du systeme d'accès aux donnees volumineuses

Apres avoir integre le systeme d'accès aux donnees volumineuses au systeme **SIBADOC**, une evaluation de l'efficacité des techniques d'indexation et d'accès définies et utilisees, est presentee.

Cette evaluation se divise en deux grandes parties:

- Evaluation de l'indexation
- Evaluation de la technique d'accès

V.3.1- Evaluation de l'indexation

L'indexation est l'opération par laquelle on choisit les termes les plus appropries pour representer le contenu d'un document. C'est donc l'opération centrale de tout systeme documentaire pour le stockage et la recherche des informations. Par consequent, l'indexation doit offrir une description pertinente des documents. Cette pertinence peut se mesurer essentiellement par l'exhaustivité et la specificite [PRA 69] [KIN 87] [LES 68].

L'exhaustivité caracterise la capacite a signaler toutes les notions importantes d'un document.

La spécificité caracterise la capacite a garder une information precise et non generalisee.

Plusieurs caracteristiques principales se degagent de l'indexation utilisée par le systeme d'acces aux donnees volumineuses. Ces caracteristiques sont:

- **Evolutivité**

L'évolutivité permet de prendre en compte rapidement et retrospectivement des evolutions des documents scientifiques et techniques. Ce concept est facilité par notre systeme, du fait que ce dernier s'appuie non pas sur un vocabulaire pré-établi, mais sur le vocabulaire extrait directement des documents.

- **Spécificité**

La methode d'indexation utilisee par le systeme d'acces aux donnees volumineuses rend cornpte d'un niveau élevé de spécificité car elle rend compte de l'information telle qu'elle est citee dans les textes, sans effet de generalisation qui est lie a un effet de synthetisation de l'information. En effet, certains termes generaux telles que intelligence artificielle, reconnaissance des formes, etc, sont introduits de façon quasi systematique par les indexeurs dans le but de traduire le contenu des documents pour des demandeurs potentiels travaillant dans des disciplines variées, ce qui n'est pas le cas de notre systeme.

- **Exhaustivite**

L'indexation utilisee par le systeme d'accès aux donnees volumineuses presente une bonne exhaustivite, du fait qu'elle

prend en compte tous les mots du corpus documentaire (du moins important au **plus** important).

Outre ces caracteristiques, deux principales mesures sont utilisées pour evaluer l'efficacité d'une recherche documentaire, et qui sont le taux de rappel et le taux de precision.

- **Definition du taux de rappel**

Le taux de rappel est la proportion des documents pertinents retrouves par rapport a l'ensemble des documents pertinents presents dans le systeme. Par exemple, si, dans une base de donnees de 100 references, 20 repondent a une question, et que le systeme en retrouve 15 mais omet les 5 autres, le taux de rappel est 15/20 ou 75%. Les documents pertinents qui n'ont pas ete retrouves constituent ce que l'on appelle le silence.

$$\text{Taux de rappel} = \frac{\text{Nombre de documents pertinents retrouves}}{\text{Nombre de documents pertinents existants}}$$

- **Definition du taux de precision**

Le taux de precision ou de pertinence est la proportion des documents pertinents par rapport a l'ensemble des documents fournis par le systeme. Si une recherche donne 40 references en reponse et que 15 documents sont pertinents,

le taux de precision est 15/40, c'est-à-dire 37,5%. On appelle bruit les 25 documents non pertinents qui ont ete fournis en même temps.

$$\text{Taux de precision} = \frac{\text{Nombre de documents pertinents retrouves}}{\text{Nombre de documents extraits}}$$

Rappel et precision sont des qualités contradictoires. Plus la recherche est precise, plus le risque est grand qu'elle ne retienne pas les documents qui auront ete decrits en terme plus generaux mais qui peuvent concerner le sujet, donc le rappel diminue et le silence augmente. Inversement, plus la recherche est large, de façon a retrouver le plus grand nombre possible de documents ayant un rapport, si limité soit-il, avec le sujet, plus le risque est grand de trouver en même temps des documents non pertinents, donc d'augmenter l'imprecision et le bruit.

La solution est d'avoir une moyenne raisonnable de ces deux parametres. C'est-à-dire ni le taux de rappel, ni le taux de precision ne doivent être faible en même temps et donc avoir un vocabulaire d'indexation ni trop general ni trop spécifique. En pratique une indexation ayant 60% de taux de rappel et 50% de taux de precision est satisfaisante pour les utilisateurs.
Comme notre systeme est base sur une inversion totale, l'interrogation est tres souple et peut être menée de façon a privilégier soit le taux de rappel, soit le taux de precision [SWE 63]. En effet, l'équation de recherche conduit a un mauvais taux de rappel si elle ne réussit pas a

couvrir les differents aspects possibles de la question, utilise une formulation **trop spécifique ou** trop exhaustive (on ne trouvera pas assez de documents repondant a tous ces critères). Elle conduit a une mauvaise precision si elle n'est pas assez spécifique et exhaustive, si elle utilise des termes ou des combinaisons de termes inadequats et si sa logique est defectueuse.

V.3.2- Evaluation de la methode d'accès

Dans ce paragraphe une comparaison entre la technique d'accès utilisée par le systeme SIBADOC et celle du Q-arbre sera presentee. Cette comparaison prend en compte les parametres temps CPU et nombre d'entrée/sortie pour la recherche.

Pour la comparaison nous avons utilise un échantillon de *3000* notices de la base de donnees INSPEC de l'année 1990, soit une liste de 18996 mots cles.

Pour rechercher un mot cle, le systeme SIBADOC qui utilise une technique d'inversion classique doit acceder au fichier repertoire, au fichier lexique et au fichier inverse, ce qui necessite au minimum trois acces disque. Par contre pour la technique du Q-arbre, ou bien le mot cle existe dans la racine et donc il ne nécessitera aucun acces disque car la racine est en mémoire centrale, ou bien il est dans une des feuilles du Q-arbre et il necessite dans ce cas un seul acces disque pour le retrouver. Par consequent, la technique du Q-arbre necessite au plus un acces disque pour rechercher un mot clé donne.

Le temps de recherche est primordial, pour les utilisateurs, dans un systeme conversationel. Les resultats experimentaux montrent que la technique du Q-arbre est plus rapide que celle de l'inversion utilisée par le systeme **SIBADOC**, et ceci est une consequence du nombre d'accès disque utilisé par chacune d'elles lors de la recherche.

De plus, lorsque le nombre de mots dans la racine augmente le temps de recherche diminue (voir tableau ci-dessous des resultats). Aussi, pour augmenter les performances de la technique du Q-arbre il faut utiliser une technique de compression pour la racine, ce qui augmente le nombre de mots a mettre dans cette derniere.

Qelques resultats de la comparaison sont illustrés par le tableau suivant:

Mots clés	**SIBADOC** temps cpu (10^{-2} s)		
		500 **mots** dans la racine	1500 mots dans la racine
Robot	25	21	13
Cmos	22	18	**11**
Transient	25	21	**11**
Unix	23	21	11
Vapor	22	19	00
Vacuum	25	***22***	09
Vibration	25	23	12
Voltage	24	22	**11**
Workstation	25	23	11
Javelength	24	22	**10**

CONCLUSIONS ET PERSPECTIVES

L'optimisation du stockage et de l'accès aux donnees volumineuses telle qu'elle est decrite dans ce document est d'un grand secours pour l'exploitation d'un grand volume d'informations et pour le repérage rapide de documents, ce qui est particulierement important.

Le systeme d'accès aux donnees volumineuses, dont la conception et la realisation ont constitue l'essentiel de notre travail, est organise de façon modulaire. I1 est adaptable a tous les systemes gerant de grands volumes de donnees textuelles et est base sur des techniques d'indexation et d'acces qui ont les caracteristiques suivantes:

- Facilité de mise en oeuvre. En effet, la technique du Q-arbre et la methode d'indexation utilisees sont tres simples a mettre en oeuvre.

- Le vocabulaire d'indexation est structure à partir des documents, il evolue donc en même temps que le fonds documentaire. De plus, comme l'indexation est basee sur une inversion totale, prenant en compte tous les mots du corpus documentaire, l'interrogation est tres souple et peut être menée de façon a privilegier soit le taux de rappel, soit le taux de precision.

- La technique d'accès étant basee sur le Q-arbre, elle est tres rapide du fait qu'elle necessite, au plus, un acces disque pour rechercher le mot desire. De plus, elle est applicable a des index de tres grande taille, ce qui a permis l'utilisation de la technique d'inversion totale et donc l'augmentation de l'efficacité de la recherche d'information.

Toutefois, nous ne pouvons pas pretendre avoir traité tous les problemes lies a l'accès aux donnees volumineuses. En effet, plusieurs extensions a notre travail peuvent être envisagees:

- Prendre en compte les mots cles composes: un systeme qui n'utilise pas les mots composes est caracterise par une grande souplesse due a la simplicité de mise en oeuvre ct a la facilité des mises a jour. I1 est caracterise aussi par une indexation fine, mais presente une certaine perte dans la precision.
La prise en charge des mots composes peut Ctre realisee de deux façons:

- Soit en utilisant la notion de mots composes dans le lexique. Par exemple, considerer "Informatique documentaire" comme un seul mot.

- Soit en utilisant, en plus des operateurs classiques ET/OU, les operateurs de voisinage tel que l'opérateur ADJACENT qui est utilisé pour préciser la position relative des termes entre eux.

- Utiliser une methode d'indexation automatique se basant sur l'approche linguistique et ou tous les problemes lies a l'indexation seront pris en compte (synonymie, polysemie, normalisation des termes, pertinence des termes, relations entre termes d'indexation, ...).

- Appliquer une methode de compression au Q-arbre afin d'augmenter le nombre de mots a mettre en mémoire centrale, et par consequent augmenter les performances de la methode d'acces.

- Offrir la possibilité de rechercher un mot en fonction de sa position: par exemple, dans le titre, dans le resume, dans tel ou tel paragraphe, etc.

- Prendre en compte les donnees ecrites dans des langues autres que l'Anglais, par exemple, le Francais ou l'Arabe. Ceci est tres simple a realiser du fait de la modularité du systeme. En effet, il suffit d'avoir un dictionnnaire de mots vides de la langue desiree (Francais ou Arabe) est faire de légère modification au niveau du module d'indexation et plus precisement au niveau de l'extraction des mots cles.

- Integrer le systeme d'acces aux donnees volumineuses au systeme de gestion d'objets complexes SYGOC. En effet, les techniques definies et utilisées dans notre systeme d'accès sont facilement adaptables a un environnement autre que celui des bases de donnees documentaires: c'est, par exemple, le cas de l'environnement des objets complexes (formulaires, images, diagrammes, etc), qui fait a l'heure actuelle l'objet de nombreux travaux de recherche.

BIBLIOGRAPHIE

[AHO 75] : AHO A.V. and CORASICK M.J.
Efficient string matching.
Commun. ACM 18, p. 333-340, 6 (June 1975).

[ALS 75] : ALSBERG, A.PETER
Space and time savings through large database compression and dynamic restructuring.
Proc. of the IEEE, vol.83, N°8, Aug 1975, pp 114-1122.

[AST 76] : ASTRAHAN M.M., and CHAMBERLIN D.D.
SYSTEM R: Relational approach to database management
ACM TRANSACTIONS OF DATABASE SYSTEMS

[BAN 88] : BANCILHON F.
Object oriented database systems.
Int. Symp. on principles of database systems, Austin (Texas), 1988.

[BAY 72] : BAYER R., and McCREIGHT E.
Organization and maintenace of large ordered indexes.
ACTA informatica vol.1, n°.3, p. 173-189, 1972.

[BEL 87] : BELKIN N.J.
Retrieval techniques
Annual Review of Information Science and Technology (ARIST), vol.22, 1987.

[BEN 88] : BENDAKIR M., and YAHIA AISSA N.
Description du systeme documentaire SIBADOC.
CERIST, 1988.

[BEN 89] : BENDAKIR M., and YAHIA AISSA N.
Le systeme SIBADOC.
CERIST, 1989.

[BLA 85] : BLAIR D.C., and MARON M.E.
An evaluation of retrieval effectiveness for a full-text document-retrieval system.
Communication of the ACM, vol. 28, n°.3, p 289-299, March 1985.

[BLU 87] : BLUMER A., and HAUSSLER D.
Complete inverted files for efficient text retrieval and analysis.
ACM vol. 34, n°.3, p. 578-595, Jul. 1987.

[BOU 91] : BOUMALHA H., BENDJENAHI F.
Vers la réalisation d'un systeme de gestion d'objets complexes: SYGOC
Dans ler Séminaire sur les Bases de Donnees, p 216-227, Juin 1991, Alger

[BOY 77] : BOYER R.S., and MOORE J.S.
A fast string searching algorithm.
Commun. ACM 20, P. 762-772, 10 (Oct. 1977).

[CHA 73] : **CHAUMIER J.**
Systemes informatiques de documentation.
Entreprise moderne d'édition, **1973.**

[CHA 80] : CHAUMIER J.
Travail et méthode du/de la documentaliste.
Collection formation permanente en sciences humaines. **1980.**

[CHA 90] :1 **CHAUMIER** J., and DEJEAN **M.**
L'indexation documentaire.
De l'analyse conceptuelle humaine a l'analyse automatique morphosyntaxique.
Documentaliste, vol.27, n°6, Nov.- Déc. 1990.

[CHR 84] : CHRISTODOULAKIS **S.** , and **FALOUTSOS C.**
Design considerations for a message file server.
IEEE Trans. Softw. Eng. SE- 10, p.201-210,
2 (Mar. **1984)**

[CHI 86] : **CHIARAMELLA Y.**
IOTA: A full text information retrieval system.
Organisation of the **1986-ACM** Conference on Research and Development in information. 1986.

[COD 70] : E. **F.** CODD
The relational model of data for large shared data banks.
Comm. of **ACM,** vol.13, n°6, Juin 1970.

[COD 79] : E.F.CODD
Extending the database relational model
to capture more meaning.
IBM research laboratory.

[COO 78] : COOPER W.S. and MARON M.E.
Foundations of probabilistic and utility
theoretic indexing.
Journal of the Association for Computing
Machinery, vol.25, n°1, p.67-80, Jan. 1978.

[COY 67] : COYAND Maurice, SIOT-DECAUVILLE Nelly
L'analyse automatique des documents.
Publié par le Centre National de la
Recherche Scientifique, Paris, 1967.

[DAL 89] : DALBIN S., and CHARTRON G.
Indexation manuelle et indexation automatique.
Documentaliste vol. 26, n°.4-5, p.181-187
(Jui. 1989).

[DEW 81] : DEWEZE, A.
Reseaux sémantiques : Essai de modelisation -
Application a l'indexation et a la recherche de
l'information documentaire.
These de doctorat, Lyon 1981.

[DEW 83] : DEWEZE, A.
L'accès en ligne aux bases documentaires.
Collection MASSON, 1983.

[DEW 85] : **DEWEZE,** A.
Informatique documentaire.
Collection MASSON, **1985.**

[DEW **89**] : DEWEZE A.
Information documentaire. Methode + Programme.
Collection MASSON, 1989,

[EAS 89] : EASTMAN C.M.
Approximate retrieval : **A** comparison of information retrieval and database management system in database engineering, vol.12, n°.2, p.41-45, **1989.**

[FAL **84**] : **CHRIS** FAMUTSOS and STAVROS CHRISTODOULAKIS
Signature files: an access method for documents and its analytical performance evaluation.
ACM Transactions on Office Information Systems, vol. 2, n°.4, p. **267-288,** Octo. **1984.**

[FAL 85] : CHRIS FALOUTSOS
Access methods for text
Computing Surveys, vol.17, n°.1, March 1985

[FUM 82] : FUMIRO.MATSMO and SHOUICHI.FUTAMURA
A Quasi-Optimum text compression code.
Ibid., vol.55, **N°2, 1982,** pp **103-106.**

[FUM 86] : FUMIHIRO MATSUO, SHOUICHI FUTAMURA, and TAKESHI SHINOHARA.
Efficient storage and retrieval of large document database.
computer, p. 456-463, jun. 1986.

[GAR 85] : GARDARIN G., and VALDURIEZ P.
Bases de donnees relationnelles
Analyse et comparaison des systemes. 1985

[GAR 89] : GARDARIN G., and VALDURIEZ P.
SGBD relationnels: Analyse et cornparaison de bases de donnees. 1989

[GAR 91] : GARDARIN G., and VALDURIEZ P.
SGBD avances:
Bases de donnees objets, déductives, réparties.
1991

[GET 87] : GETHIN P.
Text retrieval.
In : Database and Network Journal, vol.17, n 3, p. 2-7, 1987.

[GOT 75] : GOTTLIEB, DORON, HAGETH, A.STEVEN, LEHOT, G.H.PHILIPE and RABIBOWITZ, S.HENRY
A classification of compression method and their usefulness for large data processing center.
National Computer Conference 1975, pp 453-458.

[GOY 85] : GOYEL P., and KOTAMARTI V.
A strategy for compressed storage and retrieval of documents.
Compressed storage and retrieval.
p. 224-232, 1985.

[GUI 89] : GUINCHAT Clair, SKOURI Yolande
Guide pratique des techniques documentaires.
Volume 2. Traitement de l'information ".
Diffusion EDICEF, 1989.

[GUI 90] : GUINCHAT Clair, NENOV Michel
Sciences et techniques de l'information et de la documentation.
Publié par l'Organisation des Nations Unies pour l'Education, la Science et la Culture (UNESCO), 1990.

[HEA 72] : HEAPS. H.S
Storage analysis for a compression coding for a document databases.
INFOR, VOL.10, N°1, Feb 1972, pp 47-61.

[HOL 79] : HOLLAR L.A.
Text retrieval computers.
IEEE computer 12, p. 40-50, 3 (Mar. 1979).

[IAE 82] : IAEA-INIS-17
First step on STAIRS.
Document interne, 1982.

[IBM 79] : IBM WORLD **TRADE CORPORATION.**
IBM System/370 **(OS/VS).**
Storage and Information Retrieval
System/Vertical Storage.
Reference Manuel.

[JOH 69] : JOHN R., and *HARRY* D. HUSKEY
An information retrieval system based on superimposed coding.
In Fall Joint Computer Conference, **1969.**

[KEV 83] : KEVIN P.J.
How do we index?: A report of some ASLIB informatics group activity.
Journal of documentation vol. **39,** n°.1, Mar.**1983.**

[KIM 87] : KIM W., and **BANERJEE** J.
Data model issues for object oriented applications.
ACM TOIS, vol.5, n°1, 1987.

[KIN 87] : KINNUCAN M. and NELSON M.
Statistical methods in information science research.
Annuel Review of Information Science and Technology **(ARIST),** vol. **22, 1987.**

[KNU 77] : KNUTH D.E., MORRIS J.H. and PRATT V.R.
Fast pattern matching in strings.
SIAM J. Comput. 6, p. **323-350,** 2 (June **1977).**

[KUO 82] : KUO-CHUNG TAI, and ALAN L. THARP
The practicality of text signatures for accelerating string searching.
Software-Practice and Experience, vol. 12, p.35-44, 1982.

[LAL 87] : LALLICH-BOIDIN, ROUAULT, J.
Un modele linguistique de reconnaissance du français pour un systeme documentaire.
CRISS Université des Sciences Sociales de Grenoble, 1987.

[LAM 82] : LAM X. and SALTON G.
Term weighting in information retrieval using the term precision model.
Journal of the Association for Computing Machinery, vol.29, n°1, p.152-170, Jan. 1982.

[LEC 88]: LECLUSE C.
02 an object oriented data model.
Proceeding of the ACM SIGMOD conference, Chicago 1988,

[LES 68] : LESK S.
Computer evaluation of indexing and text processing.
Journal of the Association for Computing Machinery, vol.15, n'l, Jan. 1968.

[LON 80] : LONG B.
Linguistique et indexation
Documentaliste, vol. 17, n° 3, Mai-Juin 1980

[MAL 71] : **MALCOLM** C. **HARRISON**
Implementation of the substring test by hashing.
Communications of the **ACM** vol. 14, n°.12,
p. 777-779, Dec. **1971.**

[MAR 90] : **MARON** M.E., and **BLAIR** D.C.
Full-text information retrieval: further
analysis and clarification.
Information Processing & Management vol. 26,
n°.3, p. 437-447, **1990.**

[MAT 81] : **MATSUO** F., FUTAMURA S., TAKAGI T., **YOSHIDA** S.
Organization of word dictionnary indexes for
quick retrieval.
Ibid., vol. **54,** n°.3, **p. 183-187** 1981.

[McG 83] : McGILL M.J., and SALTON G.
Introduction to modern information retrieval.
McGRAW-HILL, **New York, 1983.**

[MEC 89] : **MECHKOUR M.**
Integration d'une technique de compression de
donnees a un système de recherche documentaire.
Mémoire de fin d'études, INI **1989.**

[MER 84] : MERRETT T.H.
First steps to algebraic processing of text.
New Applications of Databases.
ISBN 0-12- 275550-2, 1984.

[MET 88] : METZGER Jean-Paul
Syntagmes nominaux et information textuelle.
Reconaissance automatique et representation ".
These de docteur d'état ensciences.
Universite Claude Bernard, Lyonl, 1988.

[PEC 85] : PECHURA, A.MICHAEL and DAVID R.MCINTYRE
Data cornpression using static HUFFMAN
code-decode tables.
Communications of the ACM, vol.28, N°6,
June 1985, pp 612-616.

[PIK 81] : PIKE J.
Text compression using a 4-bit ending schema.
The computer journal, vol 24, N°4, 1981,
pp 324-330.

[PRA 69] : PRANAS ZUNDE and MARGARET E. DEXTER
Indexing consistency and quality
American Documentation. July 1969.

[RAB 84] : RABATTI F., and ZIZKA J.
Evolution of access methods to text documents in
office systems.
In Proceedings of the 3rd Joint ACM-BCS
Symposium on Reseach and Development in
Information Retrieval. 1984.

[REG 81] : REGHBATI, HASSEN K.
An overview of data compression techniques.
Computer Vo1.14, N°4, April 1981, pp 71-75.

[RIV 90] : RIVIER A.
Construction des langages d'indexation.
Aspects theoriques.
Documentaliste, vol. 27, n° 6, Nov-Dec 1990

[ROB 79] : CHARLES S. ROBERTS
Partial match retrieval via the method of superimposed codes.
In Proceedings of the IEEE, vol. 67, n°.12, Dec. 1979.

[ROU 87] : ROUAULT Jacque
Linguistique automatique. Application documentaires.
Edition Peter Lang SA, Berne, 1987.

[SAC 87] : SACKS-DAVIS R., and KENT A.
Multikey access methods based on superimposed coding techniques.
ACM Trans. on Database Sys., vol. 12, n°.4, p 655-696, Dec.1987.

[SAL 70] : SALTON G.
Automatic text analysis: Automatic document indexing and classification methods are examined and their effectiveness is assessed.
Science, vol.168, April 1970.

[SAL 71] : SALTON G.
The SMART Retrieval System-Experiment in automatic document processing.
Prentice-Hall, Englewood Cliffs, New Jersey.
1371

[SAL 86a] : SALTON G.
Recent trends in automatic information retrieval.
Proceeding of the 1986-ACM Conference on Research and Development in Information Retrieval. Pages 1-10. Pisa, Sept. 1986.

[SAL 86b] : SALTON G.
Another look at automatic text-retrieval systems.
Communication of the ACM, vol. 29, n°.7, p 648-656, July 1986.

[SAL 89] : SALTON G.
Automatic text processing.
The transformation, analysis, and retrieval of information by computer.
Addision-Wesley publishing company, 1989.

[STO 76] : STONEBRACKER.M, and WONG.E.
The design and implementation of INGRES
ACM Transactions on Database Systems, Sept.1976

[SWE 63]: SWETS John
Information Retrieval System. Statistical decision theory may provide a measure of effectiveness better than measures proposed to date.
Massachusetts Institute of Technology, Cambridge, 16 July 1963.

[TAV 92] : TAVAKOLI N., and RAY A.
A new signature approch for retrieval of documents from free-text databases.
Information Processing and Management vol.28, n°2, p.153-163, 1992.

[TIG 92] : TIGRINE B.
L'Indexation
Document interne au CERIST, 1992.

[TSI 82] : TSICHRITZIS D. CHRISTODOULAKIS
Message files., p. 110-112, ACM 1982.

[VAL 87] : VALDURIEZ P.
Objets complexes dans les systèmes de bases de donnees relationnels.
TSI vol.6, n°5, 1987.

[WIS 86] : WISNIEWSKI, JANUSZ L.
Compression of index dictionnary in an inverted-file-oriented-database : Some effective algorithms.
Info.Proceeding & management, vol.22, N°6, 1986.

[WIS 87] : **WISNIEWSKI, JANUSZ L.**
Effective text compression with simultaneous bigram and trigram encoding.
North Holland, Journal of information science 13, 1987.

ANNEXES

A N N E X E A

.

SIBADOC : Un systeme d'interrogation des bases de donnees documentaires

Une base de donnees documentaire est une collection de notices bibliographiques, chacune etant associée a un document (livre, these, article,...). Une notice contient toutes les informations qui permettent l'identification du document et de son contenu. Ces informations se presentent sous forme de texte de taille indéfinie et sont contenues dans differents champs, tels que le champ titre, auteur, resume!, etc. Le nombre de champs descriptifs dans une notice n'est pas fixe, il varie d'un type de document a un autre.

Les systemes documentaires ont pour objet, comme tous les systemes de gestion de donnees, de répondre a un besoin d'information. Leur objectif n'est pas de transmettre l'information primaire, mais plutbt des informations décrivant plus ou moins bien cette derniere, enregistrees sous forme de notices bibliographiques.

Les systemes documentaires sont donc des systemes d'information dont l'objectif est de mettre a la disposition des utilisateurs les documents secondaires; ce qui leur permet dans un premier temps de procéder a une selection sans devoir consulter les documents physiques, ensuite de localiser ces derniers.

Le systeme SIBADOC, developpe au CERIST!, s'inscrit dans cette optique et permet la gestion et l'accès a l'information documentaire des bases bibliographiques internationales acquises par le centre aupres d'agences specialisees. Actuellement, SIBADOC permet la gestion et l'accès aux bases bibliographiques suivantes:

INSPEC: pour la technologie et les sciences de l'ingénieur (INformation Services for the Physics and Engineering Communities).

INIS : pour le domaine du nucleaire et les techniques qui s'y rapportent (International Nuclear Information System)

AGRIS : pour le domaine de l'agriculture (AGRicultural Information System)

Ces bases sont commercialisees sous le format d'échange international ISO 2709.

Notons que toute base possedant une structure d'organisation similaire a celle des trois bases citees ci-dessus, est directement exploitable par le systeme SIBADOC.

1- Description du systeme SIBADOC

le systeme SIBADOC est .constitué de deux principaux modules: le module de creation et de mise a jour des bases et le modules d'interrogation [BEN 88].

1.1- Module de creation et de mise a jour

1.1.1- Structuration des bases bibliographiques

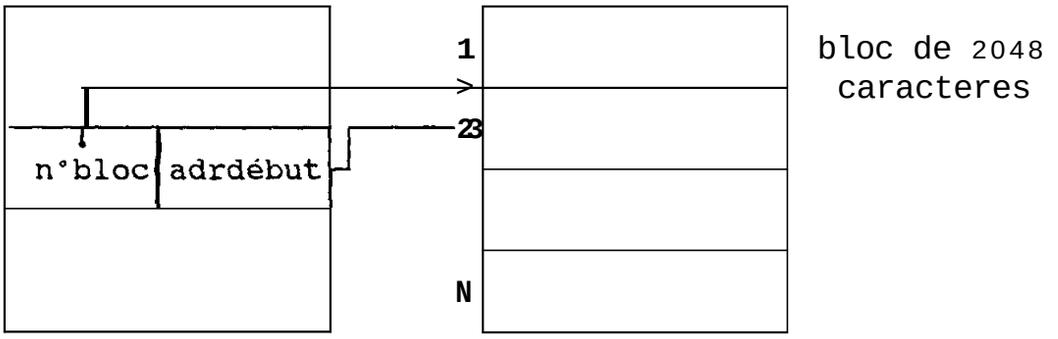
Cette premiere etape consiste a traiter les donnees émanant de differents producteurs, et donne en sortie un couple de fichiers par mois, en l'occurence un fichier d'adresses et le fichier bibliographique.

a- Structure du fichier d'adresses

L'enregistrement de ce fichier est de la forme (adr,depl) où **"adr"** désigne l'adresse du bloc et **"depl"** le deplacement ou l'adresse de la notice dans le bloc. L'objectif du fichier adresse est de localiser la notice dans le fichier bibliographique.

b- Structure du fichier bibliographique

C'est un ensemble d'enregistrements de longueur variable (appeles notices), chacun etant repéré par une adresse double (numero de bloc, deplacement dans le bloc) dans le fichier adresse. Cet ensemble de notices est organisé par périodes.
En effet, les notices du même mois sont regroupees dans le même fichier et les fichiers mensuels sont regroupes en bases annuelles dont la concatenation constitue la base generale.



Structure du fichier bibliographique mensuel

1.1.2- Creation du lexique

Pour les bases **INIS** et **AGRIS**, les thesauri existent et sont reps sur bandes magnetiques, par contre pour la base INSPEC, le CERIST dispose uniquement du thesaurus manuel.

Dans le cadre du projet **SIBADOC**, il a été procédé a la creation des lexiques pour les differentes bases.

1.1.2.1- Le fichier lexique

Le fichier lexique est constitué d'un répertoire et d'un dictionnaire de mots ou expressions de la langue naturelle representant des descripteurs, appeles aussi mots cles et tries par ordre alphabétique. Ces descripteurs sont extraits a partir du champ "mot **clé**" du fichier bibliographique et constituent un ensemble de partitions dont chacune rassemble les mots qui ont une même initiale.

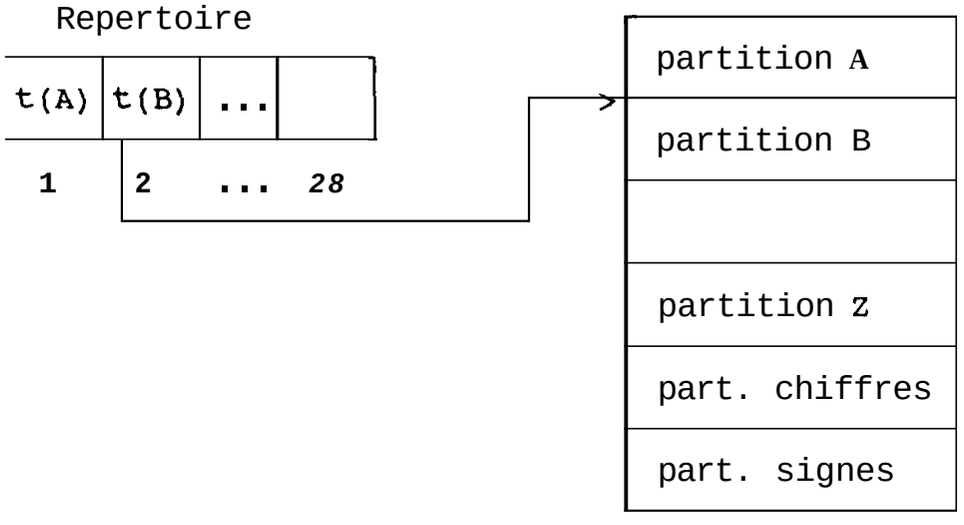
I1 existe ***28*** partitions:

- ***26*** pour l'alphabet (A..Z)
- 1 partition por les chiffres (0..9)
- 1 partition pour les signes speciaux (-,*, ...)

Le repertoire facilite la recherche dans une partition puisqui'il indique le debut et la taille de chaque partition.

Remaraue:

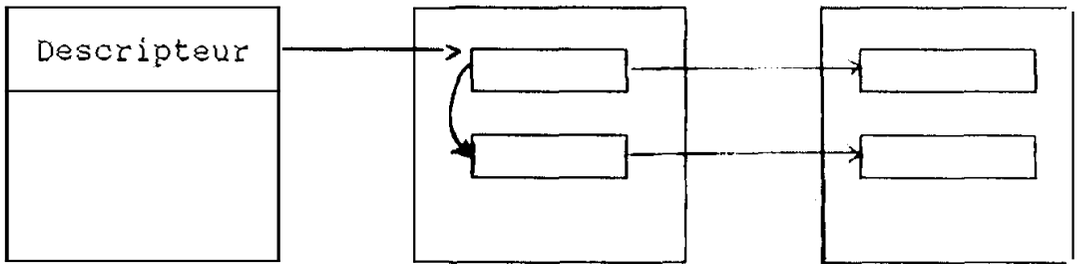
Nous avons un fichier lexique par an.



Structure du fichier lexique

1.1.3- Inversion

Cette phase consiste a creer le fichier inverse qui permet de faire le lien entre les descripteurs et les notices qu'ils indexent dans le fichier bibliographique.



Structure du fichier inverse

Remarque :

Nous avons un fichier inverse par an.

Le module de creation et de mise a jour permet donc l'évolution de la base par le traitement et l'insertion dans celle-ci du fichier mensuel des nouvelles publications. Pour cela, quatre operations sont effectuees :

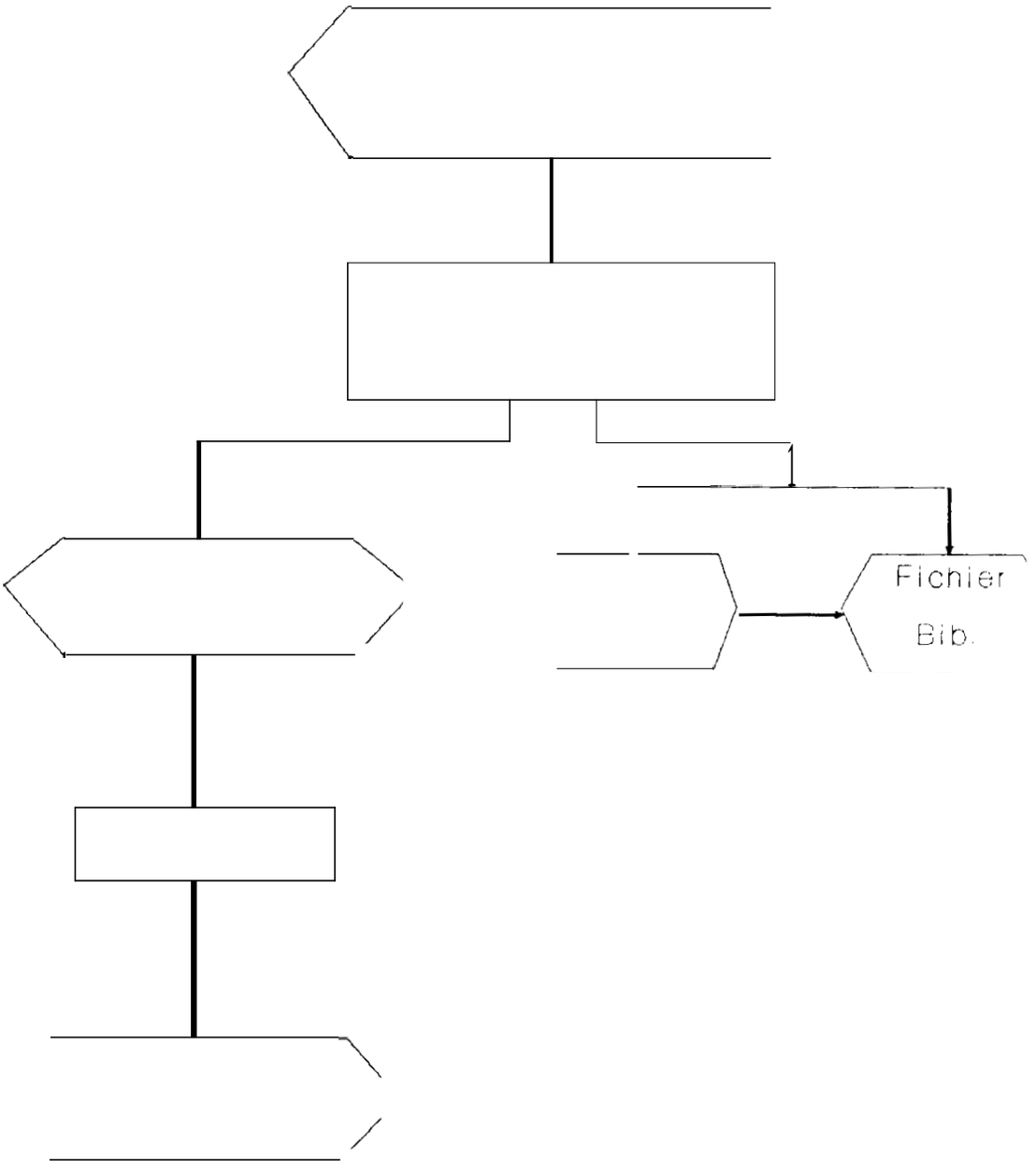
1- Creation du fichier bibliographique et de son fichier adressse pour le mois courant.

2- Indexation des documents par les mots cles figurant dans les notices par la creation d'une liste de couples (mot clé, numero de notice contenant ce mot cle).

3- Inversion : creation du fichier inverse mensuel

4- Mise a jour des fichiers inverse et lexique annuels (uniquement l'ajout est pris en compte).

Le module de creation et de mise a jour se schématise comme suit :



Module de creation **et mise a** jour **de** la **base**

1.2- Module d'interrogation des bases documentaires

Ce module permet la consultation du fonds documentaire, et comporte principalement :

- Un langage d'interrogation
- Un interpreteur de commandes.

1.2.1- Langage d'interrogation

Ce langage offre de riches possibilités de recherche aux utilisateurs desireux de communiquer avec n'importe quelle base documentaire disponible au CERIST.

I1 permet l'écriture d'une equation de recherche ou requête et de traiter tous les rapprochements informatiques nécessaires a sa satisfaction. I1 offre egalement des services d'assistance, d'édition et de navigation dans la base (changement de bases et d'années de la base).

Ces differentes fonctions sont les suivantes [BEN 89) :

1- Fonctions d'assistance

HELP : demande d'aide, fournit une documentation detaillee sur l'ensemble des commandes.

2- Fonctions de recherche

FIND : fonction de recherche par mot cle

INDEX : fonction d'extension de racine, recherche tous les mots cles contenant la racine specifiee.

SELECT: fonction de selection d'un mot cle, d'une liste de mots clés, apparaissant apres l'exécution de la fonction index.

3- Fonctions d'édition/impression
DISPLAY : affichage des notices bibliographiques
PRINT : impression des notices bibliographiques

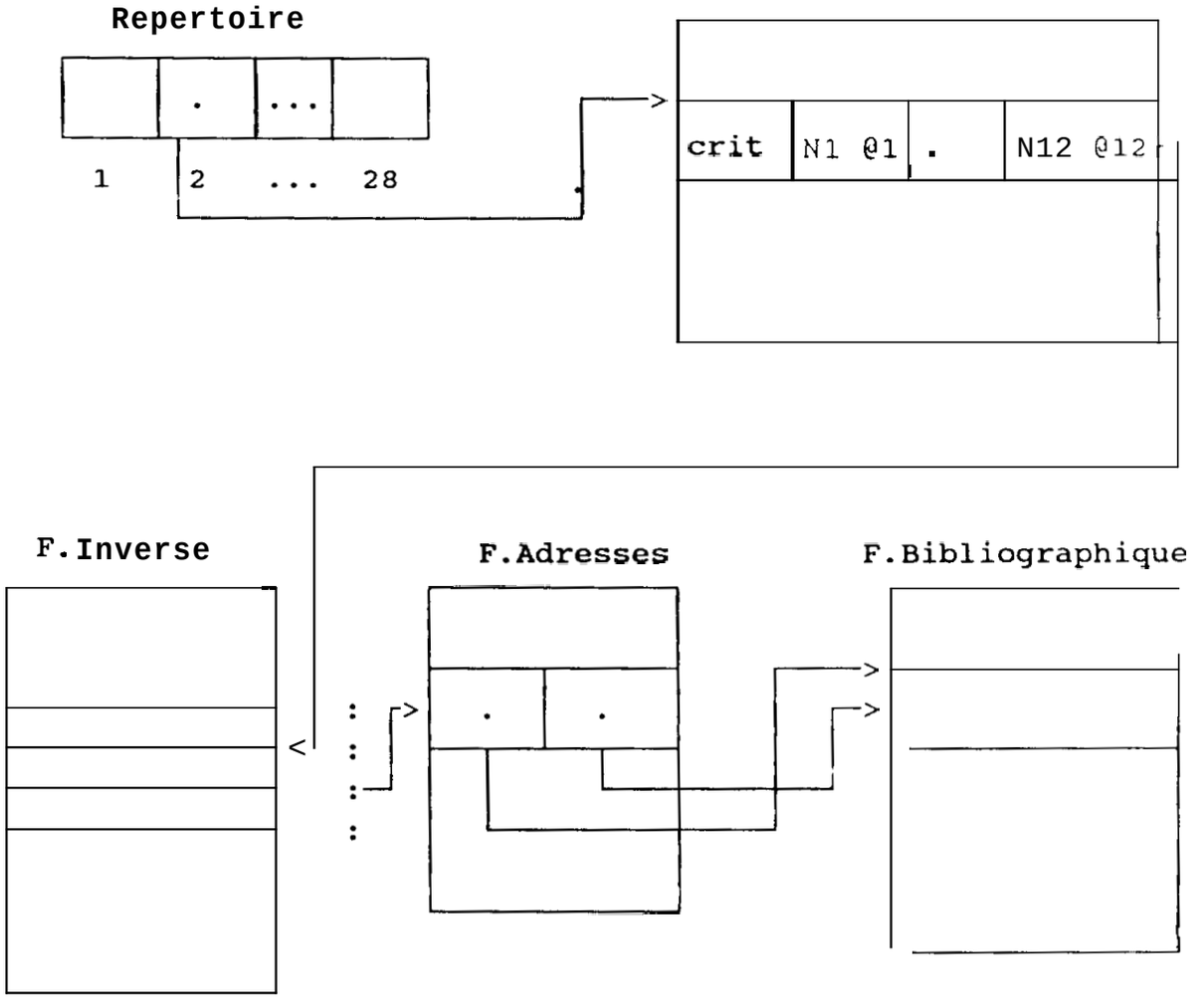
4- Fonctions de service
CHANGE : permet de changer de base ou d'année dans la base choisie et de charger les fichiers correspondants.
QUIT : terminaison de la session

6- Operateurs
OR, AND, NOT, $ (pour l'extension) et #(pour le masquage).

Une requkte est un ensemble de descripteurs lies entre eux par des relations logico-syntaxiques issues de la syntaxe du langage documentaire propre a SIBADOC. Par consequent, toute requkte etablie par un utilisateur doit Qtre soumise a une analyse syntaxico-sémantique a l'issue de laquelle celle-ci sera codee dans une notation interne et transmise à l'interpreteur de commandes.

1.2.2- L'interpréteur de commandes

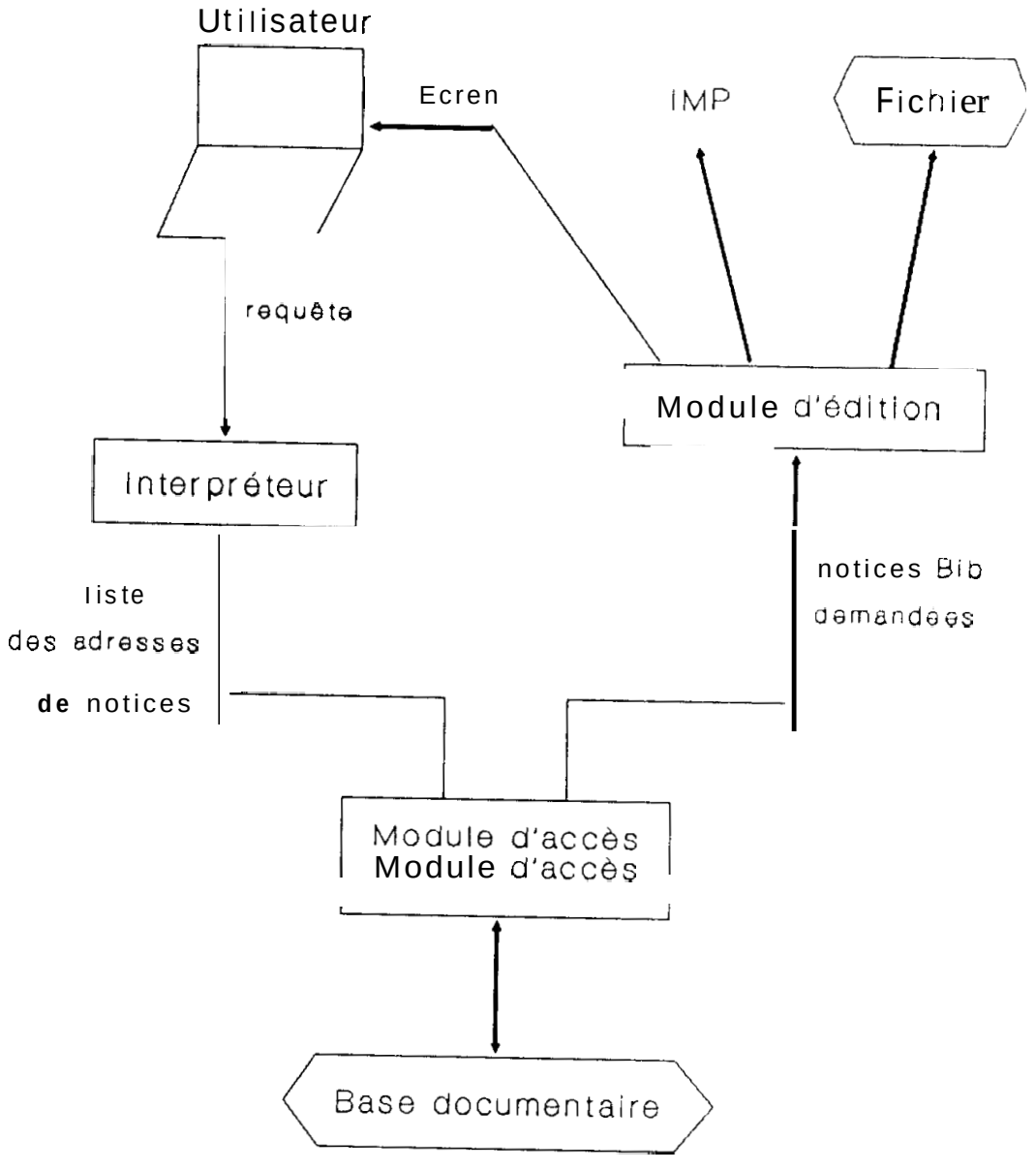
Une commande reconnue valide est executee par l'interpréteur. A ce niveau, le cheminement effectue par l'exécution d'une commande de recherche se schematise comme suit :



Ni : nombre de references du mois i (i=1 a 12)
@i : adresse de rangement des adresses dans le fichier INVERSE

Schema d'exécution d'une commande de recherche

Le processus d'interrogation se schematise comme suit :



Module d'interrogation de la **base**

Annexe B
Liste des mots vides utiliss par le module d'indexation du sous-systme d'accs aux donnes volumineuses

A
ABOUT
ABOVE
ACCORDING
AFTER
AGAINST
ALMOST
ALONE
ALONG
ALREADY
ALSO
ALTHOUGH
ALWAYS
AMONG
AND
ANOTHER
ANY
ANYONE
ANYPLACE
ANYTHING
ANYTIME
ANYWAY
ANYWHERE
ARE
AROUND
AS
AT
AWAY
BACK
BECAUSE
BEFORE
BELOW
BETTER
BETWEEN
BOTH
BRIEFLY
BUT
BY
CLEARLY
DOWN
EACH

EARLY
EASILY
EITHER
ENOUGH
ESPECIALLY
ETC
EVEN
EXCEPT
FOR
FROM
FURTHER
HARDLY
HE
HENCE
HER
HERE
HERS
HERSELF
HIM
HIMSELF
HIS
HOW
HOWEVER
I
IF
IN
INTO
IT
ITS
ITSELF
LATE
LEST
MAINLY
ME
MERELY
MINE
MOST
MUCH
MY
MYSELF
NEARLY
NECESSARY
NEVER
NEVERTHELESS
NO
OF
OFF
OFTEN
ON

ONCE
ONE
ONES
ONLY
OR
OTHER
OTHERS
OUR
OURS
OURSELVES
OUT
OUTSIDE
OVER
PER
PROVIDED
PROVIDING
QUICKLY
QUITE
RATHER
REGARDING
RELATIVELY
ROUND
SELDOM
SHE
SINCE
SLOWLY
SOME
SOMETIMES
SOON
STILL
THAN
THAT
THE
THEIR
THEIRS
THEM
THEMSELVES
THERE
THEREBY
THEREFORE
THEY
THOUGH
THROUGH
THROUGHOUT
THUS
TILL
TO
TODAY
TOGETHER
TOMORROW

TOO
TOWARD
TOWARDS
UNLESS
UNTIL
UP
UPSTAIRS
US
USUALLY
WE
WHATEVER
WHEN
WHENCE
WHENEVER
WHERE
WHEREAS
WHETHER
WHICH
WHILE
WHO
WHY
WILL
WITH
WITHIN
WITHOUT
YESTERDAY
YET
YOU
YOUR
YOURS
YOURSELF
YOURSELVES