



PROJET SOCRATE

(1) Spécifications générales



par

J.R. ABRIAL - J. BAS - G. BEAUME

G. HENNERON - R. MORIN - G. VIGLIANO

TABLE DES MATIERES

CHAPITRE 1

1. INTRODUCTION -----	1
1.1. Historique -----	1
1.2. Exemples d'utilisation pratique d'une banque de données ---	4
1.2.1. Hôpital -----	4
1.2.2. Entreprise -----	5
1.2.3. Documentation -----	6
1.2.4. Projet -----	7
1.3. Organisation générale d'une banque de données -----	8
1.3.1. Les modes de base -----	8
1.3.2. Schéma général minimum -----	8
1.3.3. Fonctionnement en accès multiple -----	9
1.3.4. Indépendance par rapport au hardware et au software de base -----	11
1.3.5. Modes de fonctionnement spéciaux -----	12
1.3.6. Problème de sécurité -----	13

CHAPITRE 2

2. LANGUAGE -----	15
2.1. Langage de définition -----	15
2.1.1. Spécifications générales -----	15
2.1.1.1. Définitions élémentaires -----	16
2.1.1.2. Combinaison de données élémentaires -----	16
2.1.1.2.1. Bloc -----	17
2.1.1.2.2. Entité -----	17
2.1.1.2.3. Conditions -----	18
2.1.1.2.4. Références -----	19
2.1.1.3. Conclusion -----	20
2.1.2. Etude détaillée du langage - extension du langage --	20
2.1.2.1. Programme et bloc -----	20
2.1.2.2. Caractéristiques -----	21
2.1.2.3. Caractéristique simple -----	22
2.1.2.4. Caractéristique conditionnelle -----	26

2.2. Le langage de citation -----	32
2.2.1. Principes du langage -----	32
2.2.2. Fonctionnement du langage de citation -----	34
2.2.2.1. Langage de citation simplifié -----	35
2.2.2.1.1. Citation -----	35
2.2.2.1.1.1. Filtre introduit par AYANT -----	36
2.2.2.1.1.2. Ordinaux : UN, DERNIER -----	37
2.2.2.1.1.3. Démonstratifs : ENCOURS, LUIEME --	39
2.2.2.1.2. Conditions - Filtres -----	41
2.2.2.1.2.1. Combinaison d'expres- sions booléennes : Forme "ET de OU" priorité du OU -----	42
2.2.2.1.2.2. Condition "existen- tielle" : EXISTE et AUCUN -----	44
2.2.2.2. Langage de citation étendu -----	46
2.2.2.2.1. Nécessité de l'extension -----	46
2.2.2.2.2. Opérations sur des ensembles - quantificateurs -----	47
2.2.2.2.2.1. Logique des prédicats	48
2.2.2.2.2.2. Quantificateurs : UN et TOUT -----	49
2.2.2.2.2.3. Les démonstratifs (ou lettres muettes)	50
2.2.2.2.2.4. Filtre introduit par TELQUE -----	51
2.2.2.2.2.5. Conditions -----	52
2.3. Langage de commande -----	53
2.3.1. Généralités -----	53
2.3.2. Commandes élémentaires -----	53
2.3.2.1. C : CREATION -----	55
2.3.2.2. A : ADDITION -----	55
2.3.2.3. M : MISE A JOUR -----	55
2.3.2.4. E : EFFACEMENT -----	55
2.3.2.5. I : INTERROGATION -----	56
2.3.2.6. N : NUMERATION -----	57
2.3.2.7. S : SURVEILLANCE -----	58

2.3.3. Commandes groupées P : POUR -----	60
2.3.3.1. Généralités -----	60
2.3.3.2. Fonctionnement -----	61
2.3.3.2.1. Groupe d'actions - Groupe simple et Groupe itératif -----	62
2.3.3.2.2. Liste d'actions -----	64
2.3.3.2.3. Actions élémentaires -----	64
2.3.3.3. Opérations sur des ensembles -----	68
2.3.4. Commandes utilitaires -----	69
2.4. Macro-générateur du langage -----	69
2.4.1. Introduction -----	69
2.4.2. Définition de macros -----	72
2.4.3. Appel de macro -----	74
2.4.3.1. Appel simple -----	74
2.4.3.2. Possibilités de concaténations -----	75
2.4.4. Exemples -----	76
2.5. Conclusion -----	78
2.6. Grammaire unifiée du langage -----	78

CHAPITRE 3

3. ORGANISATION DES FICHIERS -----	85
3.1. Introduction -----	85
3.1.1. Exemple -----	85
3.1.2. Forme interne des informations -----	87
3.1.2.1. Caractéristique liste de valeurs -----	87
3.1.2.2. Caractéristique nombre -----	87
3.1.2.3. Caractéristique mot -----	88
3.1.2.4. Caractéristique texte -----	89
3.2. Technique dynamique -----	90
3.2.1. Représentation physique d'une structure d'arbre -----	90
3.2.2. Les références -----	91
3.2.3. Organisation de la mémoire virtuelle -----	93
3.2.4. Opérations définies sur la mémoire virtuelle -----	95
3.2.5. Espace réel -----	96
3.2.6. Relation entre l'espace réel et l'espace virtuel -----	97
3.3. Technique statique -----	98

3.3.1. Définition d'une mémoire virtuelle -----	98
3.3.2. Organisation des informations dans la mémoire virtuelle ---	98
3.3.3. Les références -----	100
3.3.4. Relations entre l'espace réel et l'espace virtuel -----	101
3.3.4.1. Utilisation d'un dictionnaire en mémoire centrale	102
3.3.4.2. Utilisation d'un dictionnaire dispersé en mémoire virtuelle -----	104
3.3.5. Récupération de l'espace réel -----	111
3.3.6. Recherche rapide -----	113
3.3.6.1. Utilisation de l'organisation mixte -----	113
3.3.6.2. Détermination directe de l'adresse par hash-code -	114
3.3.7. Conclusions -----	114
3.4. Comparaison entre les techniques statiques et dynamiques -----	115
3.4.1. Utilisation de la mémoire secondaire -----	115
3.4.2. Nombre d'accès à la mémoire secondaire -----	115
3.4.3. Algorithmes -----	116
3.4.4. Conclusion -----	116
3.5. Exemple d'organisation en technique statique -----	116

CHAPITRE 4

4. COMPILATION - TRADUCTION -----	119
4.1. Définition -----	119
4.1.1. Traduction -----	119
4.1.2. Vérification -----	120
4.1.3. Utilisation -----	120
4.1.3.1. Au stade de la définition -----	120
4.1.3.2. Aux stades de l'interrogation et de la mise à jour	120
4.1.3.3. Schéma d'utilisation -----	122
4.2. Fonctionnement -----	122
4.2.1. Principes -----	122
4.2.2. Analyse lexicale -----	122
4.2.3. Analyse syntaxique - Grammaire hors contexte -----	124
4.2.3.0. Présentation de la Grammaire du langage de base sous forme de Backus -----	124
4.2.3.1. Texte de Définition de fichier -----	127
4.2.3.2. Phrase de citation -----	130

4.2.4. Analyse dépendant du contexte -----	134
4.2.4.1. Analyse dépendant du contexte dans le cas d'un texte de Définition -----	135
4.2.4.1.1. Portée des noms -----	135
4.2.4.1.2. IDEM -----	139
4.2.4.1.3. Références et conditions -----	141
4.2.4.2. Analyse dépendant du contexte dans le cas d'une phrase de citation -----	141
4.2.4.2.0. ENTITE PERSONNE -----	141
4.2.4.2.1. Vérification de définition des identi- ficateurs -----	142
4.2.4.2.2. Vérification portant sur le type des identificateurs -----	142
4.2.4.2.2.1. Blocs et entités -----	142
4.2.4.2.2.2. Référence -----	142
4.2.4.2.2.3. Hiérarchie des noms -----	143
4.2.4.2.2.4. UN, DERNIER -----	144
4.2.4.2.2.5. Filtres et conditions ----	144
4.2.4.2.2.6. Filtres -----	145
4.2.4.2.2.7. Dans les filtres : confor- mité de type de part et d' autre d'un opérateur -----	145
4.2.4.2.2.8. Les démonstratifs dans les citations -----	146
4.2.4.3. Conclusions -----	146
4.2.5. Liaison entre analyses lexicale et syntaxique -----	147
4.2.5.1. L'analyseur lexical (A.L.) est complètement disso- cié de l'analyseur syntaxique (A.S.) -----	147
4.2.5.2. Analyseur syntaxique guidant l'analyseur lexical -	148
4.2.5.3. Conclusion -----	149
4.5. Construction du codage interne -----	149
4.3.1. Codage interne résultant d'une définition -----	149
4.3.1.1. Souplesse ou rigidité du code interne de Base ----	150
4.3.1.1.1. Codage strictement figé -----	150
4.3.1.1.2. Codage souple -----	152
4.3.1.2. Souplesse ou rigidité des données réelles du fichier -----	158
4.3.1.2.1. Souplesse ... apparente -----	158
4.3.1.2.2. Rigidité ... dans un espace virtuel ---	158
4.3.1.3. Cas des caractéristiques conditionnelles et des Références -----	162

4.3.1.3.1. Caractéristiques conditions -----	163
4.3.1.3.2. Caractéristiques références -----	167
4.3.2. Codage interne des citations -----	167
4.3.2.1. Généralités -----	167
4.3.2.2. Terminologie des citations -----	168
4.3.2.3. Fonctionnement du compilateur pour le codage d'une citation -----	168
4.3.2.3.1. Premier passage - sens direct - analyse -----	169
4.3.2.3.2. Second passage - sens structurel logique - préinterprétation -----	169
4.3.2.3.3. Principaux codes utilisés -----	170
4.3.2.4. Exemples -----	171
CHAPITRE 5	
5. MANIPULATION DES FICHIERS -----	175
5.1. Création -----	175
5.1.1. Entrée des données -----	176
5.1.1.1. Mode conversationnel -----	176
5.1.1.2. Mode batch -----	177
5.1.1.3. Mode programme -----	178
5.1.2. Contrôle sémantique -----	178
5.1.3. Traitement des enregistrements -----	178
5.1.3.1. Cas des fichiers dynamiques -----	178
5.1.3.2. Cas des fichiers statiques -----	181
5.1.4. Utilisation d'un fichier type "Cobol" -----	182
5.2. Interrogation -----	182
5.2.1. Fonctionnement de l'interpréteur de base -----	183
5.2.1.1. Rappels sur le langage -----	183
5.2.1.2. Technique dynamique -----	184
5.2.1.2.1. Principe -----	187
5.2.1.2.2. Marche sur un exemple -----	189
5.2.1.2.3. Références -----	190
5.2.1.2.4. Traitement du "ENCOURS" -----	195
5.2.1.3. Technique statique -----	196
5.2.1.3.1. Principe -----	196
5.2.1.3.2. Organisation dispersée -----	198
5.2.1.3.3. Organisation verticale -----	199
5.2.1.3.4. Remarques -----	199

5.2.1.4. Exemples -----	201
5.2.2. Fonctionnement de l'interpréteur général -----	205
5.2.2.1. Rappel sur le langage -----	205
5.2.2.2. Principe -----	206
5.2.2.2.1. Principe de l'algorithme-----	208
5.2.2.2.2. Interprétation des lettres ruelles-----	214
5.2.2.2.3. Interprétation du TELQUE-----	214
5.2.2.2.4. Interprétation du PAR-----	215
5.2.2.2.5. Interprétation du POUR-----	216
5.2.2.3. Exemple-----	217
5.3. Mise à jour -----	220
5.3.1. Mise à jour standard-----	220
5.3.1.1. Caractéristiques simples-----	220
5.3.1.2. Références-----	221
5.3.1.3. Blocs-----	221
5.3.1.4. Entités-----	221
5.3.2. Mise à jour spontanée-----	222
5.4. Suppression-----	224
5.4.1. Entités-----	225
5.4.2. Blocs -----	227
5.5. Edition des résultats-----	227
5.5.1. Edition standard-----	227
5.5.2. Editions particulières-----	230
5.6. Sécurité des fichiers-----	233
5.6.1. Niveau physique-----	233
5.6.2. La sécurité au niveau de l'emploi -----	234
5.6.2.1. Correction d'erreurs - prévention d'erreurs-----	234
5.6.2.1.1. Erreurs syntaxiques-----	234
5.6.2.1.2. Erreurs sémantiques-----	234
5.6.2.1.3. Erreurs incontrôlables-----	235
5.6.2.1.4. Niveaux d'action et degrés de qualification-----	235
5.6.2.2. Discrétion - Fichier commun - Cloisonnement-----	236
5.6.2.2.1. Introduction-----	236
5.6.2.2.2. Méthode de discrétion utilisée-----	238
5.6.2.2.2.1. Choix-----	238
5.6.2.2.2.2. Exposé de la méthode-----	239
5.6.2.2.3. Exemple-----	239
5.6.2.2.4. Critique-----	241

CHAPITRE 6

6. MANIPULATION DES STRUCTURES ELLES-MEMES -----	243
6.1. Création de structures-types -----	244
6.1.1. Principe -----	244
6.1.2. Exemple -----	245
6.2. Interrogation de structures -----	247
6.2.1. Interrogation simple -----	248
6.2.1.1. Généralités -----	248
6.2.1.2. Fonctionnement -----	248
6.2.1.3. Exemple de fonctionnement -----	250
6.2.1.4. Techniques utilisées -----	251
6.2.1.5. Conclusion - Extension -----	252
6.2.2. Recherche d'identificateur par analogie morphologique	253
6.2.3. Langage libre -----	254
6.3. Mise à jour -----	255
6.3.1. Dispositif d'aide à la mise à jour de structure : es- pionnage -----	256
6.3.1.1. Généralités -----	256
6.3.1.2. Espionnage de caractéristiques élémentaires (mot, texte, valeur numérique, liste de va- leurs, référence) -----	256
6.3.1.3. Espionnage de blocs -----	258
6.3.1.4. Espionnage d'entité -----	259
6.3.1.5. Conclusion -----	260
6.3.2. Les différents niveaux de mise à jour de structure ---	260
6.3.3. Les différentes modifications -----	261
6.3.4. Modification de la structure formelle -----	262
6.3.5. Conséquences de la modification de la structure for- melle sur les objets physiques du fichier -----	266
6.3.5.1. Méthode du "fichier pivot et programme extérie- ne" -----	265
6.3.5.2. Méthode dite du "coup par coup" -----	268
6.3.5.3. Méthode par refonte complète du fichier -----	270
6.4. Suppression de structures -----	271

CHAPITRE 7

7. ACCES MULTIPLES -----	273
7.1. Introduction -----	273
7.2. Multiprogrammation -----	274

7.2.1. Multiprogrammation au niveau des périphériques -----	275
7.2.1.1. Processus LIGNE -----	275
7.2.1.1.1. Principe -----	275
7.2.1.1.2. Synchronisation -----	276
7.2.1.2. Processus PAGE -----	278
7.2.1.2.1. Principe -----	278
7.2.1.2.2. Synchronisation -----	279
7.2.2. Multiprogrammation au niveau du système -----	283
7.2.2.1. Problème du dead-lock -----	284
7.2.2.2. Evolution de l'état d'une entité -----	285
7.2.2.3. Blocages -----	286
7.2.2.4. Déblocages -----	287
7.2.2.5. Stockage des états en mémoire -----	288
7.3. Scheduler -----	288
CHAPITRE 8	
8. EXEMPLES -----	291
8.1. Fonctionnement général du système -----	29
8.2. Carthographie -----	295
8.3. Fichier "MERDIC" -----	301
8.4. Fichier "QUARTO" -----	306
8.5. Dossier médical -----	313
8.6. Fichier de boursiers -----	324
8.7. Fichier MAISON-VOITURE -----	334
8.8. Bibliographie -----	340
ANNEXE -----	351

D I A L O G U E

SOCRATE - Nous sommes d'accord, n'est-ce pas, que pour se souvenir de quelque chose, il faut l'avoir su auparavant ?

SIMMIAS - *Tout à fait d'accord.*

SOCRATE - Le sommes-nous aussi sur ce point, que, lorsque la connaissance nous vient de cette façon-ci, c'est une réminiscence ? Voici ce que j'entends par "cette façon-ci" :

Si un homme qui a vu, entendu ou perçu quelque chose d'une autre manière, non seulement a pris connaissance de cette chose, mais encore a songé à une autre qui ne relève pas de la même science, mais d'une science différente, est-ce que nous n'avons pas le droit de dire qu'il s'est ressouvenu de la chose à laquelle il a songé ?

Comment cela ?

Prenons un exemple : autre chose est la connaissance d'un homme, et autre chose est la connaissance d'une lyre.

Sans doute.

Eh bien, ne sais-tu pas ce qui arrive aux amants à la vue d'une lyre, d'un manteau ou de quelque autre chose dont leurs amis ont l'habitude de se servir ? En même temps qu'ils reconnaissent la lyre, ils reçoivent dans leur esprit l'image de l'enfant à qui cette lyre appartient. Et cela, c'est une réminiscence, tout comme, quand on voit Simmias, on se souvient de Cèbes, et je pourrais citer des milliers d'exemples du même genre.

Des milliers, oui, par ZEUS !

N'y a-t-il pas dans un tel cas une sorte de réminiscence, surtout lorsqu'il s'agit de choses que le temps ou l'inattention a fait oublier ?

Assurément si.

Mais, en voyant un cheval ou une lyre, ne peut-on pas se souvenir d'un homme et, en voyant le portrait de Simmias se ressouvenir de Cèbes ?

Certainement.

Et en voyant le portrait de Simmias, se ressouvenir de Simmias lui-même ?

Certainement, on le peut Socrate.

PLATON

(PHEDON)

Ce document est le résultat d'une recherche appliquée. Il ne prétend pas décrire un Système de Banque de Données "on line" mais plutôt les techniques que nous avons peu à peu dégagées ou reprises pour la réalisation de plusieurs prototypes d'un tel système. Nous nous excusons donc auprès du lecteur du manque de précision ou de cohérence qu'il rencontrera sûrement dans ces pages.

Après un premier chapitre introductif qui donne une définition implicite d'une Banque de Données "on line" au moyen d'exemples, ainsi qu'un rapide tour d'horizon de ses principaux éléments, nous abordons avec le chapitre 2 une étude des différents langages (de définition, d'interrogation et de commande) mis à la disposition des utilisateurs.

Le chapitre 3 étudie en détail les structures de mémoires que l'on peut envisager pour stocker les fichiers dans un contexte de support à accès direct : Nous servant de l'abstraction d'une mémoire virtuelle, nous avons développé deux techniques (dites dynamique et statique) de gestion de ce milieu virtuel puis plusieurs méthodes de passage à la mémoire secondaire réelle.

Le chapitre 4 traite de la traduction des différents langages présentés au chapitre 2 dans le contexte des techniques introduites au chapitre 3.

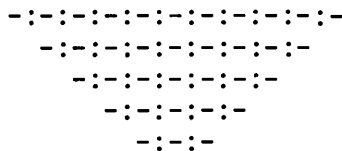
Le chapitre 5 étudie en détail le fonctionnement des algorithmes généraux de la Banque de Données. Les algorithmes correspondent aux fonctions de base de création, interrogation, mise à jour et suppression sur tout ou partie des fichiers. Ils "interprètent" les codes issus des traductions définies au chapitre 4.

Le chapitre 6 s'intéresse à la manipulation des structures de données elles-mêmes. Une part importante de ce chapitre est consacrée à la modification de structure et aux trois techniques que nous envisageons pour en assurer les conséquences au niveau des fichiers.

Le chapitre 7 est consacré à l'étude de l'accès multiple à la Banque de Données. Nous avons prêté une attention particulière aux problèmes délicats posés par l'accès multiple simultané en interrogation, mise à jour et suppression sur un même fichier.

Le chapitre 8 est constitué uniquement d'exemples de fonctionnement des prototypes du système dans des cas d'applications diverses.

Ce travail a été effectué dans le cadre du contrat DGRST
n° 68-01-450.



CHAPITRE 1

1. INTRODUCTION

Ce chapitre introductif a pour but de présenter l'organisation d'une banque de données générale (§ 1.3.). Il nous a semblé intéressant de la faire précéder par un bref rappel historique (§ 1.1.) suivi d'une série d'exemples d'utilisation (§ 1.2.) afin de la mieux placer dans ses contextes passé et présent.

1.1. Historique

Jusqu'à la fin de la première génération des ordinateurs, chaque programme était un tout indépendant : le programmeur devait décrire à chaque fois la totalité des opérations élémentaires qu'il voulait voir exécuter par la machine.

Avec la deuxième génération est apparue la nécessité de doter les machines d'un ensemble de programmes plus ou moins standards aptes à exécuter des tâches de routine comme les Entrées/Sorties, les calculs mathématiques les plus courants, la gestion des mémoires secondaires, le chargement relatif des programmes, etc...

./.

En même temps, le jeu de langages symboliques mis à la disposition des utilisateurs augmentait sans cesse, si bien qu'à la liste de programmes standards précédemment invoquée s'ajoutait la liste toujours croissante des divers compilateurs. Enfin, on s'est aperçu de l'intérêt que l'on pouvait tirer de la simplification du travail des opérateurs pour limiter les temps d'arrêt de la machine.

Ainsi, de la réunion de ces programmes généraux sont nés les "operating systems" et du travail simplifié des opérateurs, le fonctionnement en "batch".

La troisième génération, sur ce plan, n'a fait, en son début, que développer à outrance ce qu'avait fait la deuxième, tirant profit de la plus grande indépendance du hardware d'Entrée/Sortie par rapport au hardware de calcul pour introduire la multiprogrammation.

Sur le plan de l'utilisation, les petites et moyennes machines ont vu, depuis la fin de la première génération, leur nombre s'accroître de façon explosive pour un emploi presque unique : la gestion. Les grosses et très grosses machines étaient surtout utilisées pour le calcul scientifique, l'enseignement et les systèmes fonctionnant en temps réel.

Intéressons nous aux applications dans le domaine de la gestion. Elles se résument pratiquement toutes à une seule activité : manipuler des fichiers.

Avec la deuxième génération et la troisième jusqu'à une date récente, les utilisateurs-gestion disposaient en fait d'outils **de travail**, de la puissance des "operating-systems-batch" et du langage COBOL (puis de PL/1) ; manipuler des fichiers consistait donc essentiellement à écrire des programmes COBOL (ou PL/1) pour :

- définir des structures simples de fichier (structure arborescente)
 - entrer et contrôler les données constituant ces fichiers
 - assurer la mise à jour de ces fichiers en ajoutant, changeant ou supprimant des données (les "mouvements" du fichier)
 - enfin, exécuter des traitements plus spécifiques sur ceux-ci (classement, programme de paie sur un fichier de personnel, programme de gestion de stocks, voire programme d'ordonnancement, etc...)
- ./.

Plus précisément, chaque fois qu'une nouvelle opération, même simple, apparaissait sur un fichier existant (question élémentaire, statistique, exécution d'un modèle, etc...) on écrivait un nouveau programme COBOL pour cette nouvelle activité.

Les outils dont disposaient les utilisateurs (COBOL + operating-systems batch) étaient-ils adaptés et cohérents ?

On peut dire que, dans le cadre d'un fonctionnement en batch sur petite ou moyenne machine utilisant des bandes magnétiques comme mémoire secondaire, ces outils étaient relativement bien adaptés.

Deux faits sont venus sensiblement modifier les règles du jeu :

- sur le plan hardware, l'apparition des mémoires secondaires à accès aléatoire (disque, tambour) beaucoup plus performantes que les bandes magnétiques.
- sur le plan software, l'éclosion au niveau des grosses ou des moyennes machines des systèmes fonctionnant en accès multiple.

Ces faits ont eu pour conséquences, d'une part une beaucoup plus grande accessibilité de la machine pour ses usagers et d'autre part la mise à disposition des programmeurs d'un support mémoire se prêtant beaucoup mieux que les bandes magnétiques à un fonctionnement dynamique.

Or, de façon assez curieuse, alors que les applications de gestion ne cessent de croître et de vouloir se compliquer (gestion intégrée), il s'est trouvé que les outils (gestion de fichiers au niveau des systèmes plus COBOL) n'ont guère évolué pendant les quelques années où ces faits apparaissent de plus en plus nettement.

Par contre, dans le domaine des activités de "calcul", de recherche et d'enseignement il semble que les outils aient évolué compte-tenu de ces faits, dans un sens de plus en plus favorable pour les utilisateurs.

En bref, alors que les systèmes classiques actuels sont satisfaisants pour les activités "programmation" ils sont inadaptés pour les activités "manipulation de fichiers". Par ailleurs, le clivage entre ces deux activités se creuse chaque jour davantage.

Le besoin d'un outil spécialement adapté à ce genre d'activités se fait donc sentir. C'est pourquoi on assiste depuis quelques années à la naissance (souvent incohérente d'ailleurs) d'une classe nouvelle de systèmes que l'on a appelé : Data Management System, Management Information System, Data Bank, etc...

Il faut noter tout de suite que, pour ces divers systèmes, la spécificité ou la généralité d'une part et la dépendance ou l'indépendance par rapport aux operating systems d'autre part sont des plus variables.

Le système que nous présentons ici est une Banque de Données Générale fonctionnant en accès multiple de façon indépendante ou non d'un operating system classique.

1.2. Exemples d'utilisation pratique d'une banque de données

Nous allons donner quelques exemples d'applications d'une Banque de Données Générale.

1.2.1. Hôpital

Un hôpital est divisé en services médicaux spécialisés (rhumatologie, gastro-entérologie, maladies infectieuses, chirurgie, etc...) et paramédicaux (laboratoires, radiologie, etc...). Supposons que dans chaque service soient implantés un ou plusieurs terminaux (machine à écrire, display à écran cathodique).

A partir de ces terminaux, on peut de façon conversationnelle gérer plusieurs types de fichiers concernant :

- les malades proprement dits
- les soins qu'on leur porte
- les examens qu'ils subissent
- les stocks (médicaments, nourriture, produits, lits)
- le personnel hospitalier
- etc...

./.

L'utilisation de tels fichiers peut être extrêmement variée :

- obtention immédiate de tout ou partie de l'information concernant un malade, soit s'il est à l'hôpital, soit lorsqu'il y revient en consultation ou en séjours successifs,
- conduite des soins au niveau d'un grand service,
- découloisonnement des services : le résultat d'une analyse pour un malade peut être connu dans un service médical dès qu'il a été enregistré au niveau du laboratoire ; un malade dont les soins requièrent les compétences de plusieurs services médicaux voit son dossier le "suivre" implicitement,
- au niveau de l'entrée du malade, détermination, à service égal, du meilleur endroit où l'envoyer (problème de charge),
- statistique et recherche médicale,
- gestion administrative : détermination du "prix de revient" du malade, d'un service, etc...,
- mouvement du personnel,
- etc...

La bonne marche d'une telle organisation suppose que les opérations de base sur les fichiers (création, mise à jour, suppression, interrogation) soient faciles, rapides et exécutables en permanence par un personnel non spécialisé.

1.2.2. Entreprise

L'utilisation d'une banque de données dans une entreprise ne diffère pas essentiellement de son utilisation dans un hôpital :

On peut considérer que chaque département de l'entreprise est à la fois créateur d'une certaine information et client de la même ou d'une autre information. La banque de données sert donc de lien entre les départements par ses aspects centralisés (ordinateur et moyen de mémorisation unique) et décentralisés (terminaux pour la prise d'information et la ventilation).

Comme pour l'hôpital, la vie informationnelle de l'entreprise est permanente ; le multiple accès favorise donc l'entrée de l'information dès sa naissance : ceci diminuera les risques de distorsion par rapport à un fonctionnement en différé (fonctionnement qui est toujours possible) et facilite le contrôle de l'information dès son entrée (possibilité de dialogue).

Exemple : Supposons que la Banque gère un fichier de personnel et que nous ayons un terminal à la Direction du Personnel. A l'embauche d'une personne on peut tout de suite créer l'enregistrement correspondant. Cet enregistrement sera sûrement très dépendant de facteurs comme le sexe, l'âge, les qualifications, etc... On peut, de façon générale, indiquer au niveau de la Banque certaines lois sémantiques propres à l'entreprise : tel poste requiert un titulaire de sexe masculin ayant plus de 30 ans et ayant fait des études secondaires ; la machine peut alors indiquer dès l'entrée la non cohérence éventuelle de l'enregistrement.

L'utilisation de la banque n'est pas limitée à un emploi conversationnel. On peut par exemple brancher un programme de paie, un programme de dépouillement d'enquête ou un programme d'ordonnancement. A ce moment, le résultat, au lieu de sortir sur un terminal, peut s'effectuer sur une imprimante rapide au niveau centralisé.

1.2.3. Documentation

Un grand centre de documentation peut tirer un avantage considérable de l'emploi d'une banque de données générale. Il s'agit essentiellement de gérer le fichier de milliers d'ouvrages, revues, articles pour assurer les services suivants :

- répondre à des questions documentaires,
- contrôler les mouvements des ouvrages (entrée, prêt, relance, etc...)
- alimenter le service de photocopies pour les articles à dupliquer.

Le fonctionnement d'un tel centre est organisé autour d'un central téléphonique qui assure le "dispatching" vers des opératrices placées devant des terminaux. Ayant connaissance de tout ou partie de la structure des fichiers, elles sont capables de transcrire des demandes et de donner dans la même communication les réponses de la banque puis, éventuellement, par dialogue (avec le demandeur et aussi avec la banque) de poursuivre le travail d'information et de commande.

Ce style de travail n'est pas du tout limité à un environnement documentaire. On peut très bien l'extrapoler à tout organisme qui doit fournir des renseignements à un grand nombre d'utilisateurs : agences, réservation, administration, etc...

1.2.4. Projet

Un projet est une activité limitée dans le temps et qui doit se concrétiser par un résultat.

Exemple : Le projet Concorde débute lors de la passation des accords gouvernementaux et se termine lorsque les prototypes ont franchi avec succès les essais prévus.

Le déroulement d'un projet nécessite une organisation humaine très dynamique. La charge en personnel y est en effet très variable.

Cet ensemble humain est générateur d'une information que l'on peut schématiser comme suit :

- un système informel de relations humaines, réunions, conférences (internes ou externe), etc...
- un système formel beaucoup plus structuré, notes techniques, rapports d'avancement, plans, états des travaux, etc...

On assiste souvent à une inflation considérable de toute sorte de documents plus ou moins standardisés : il arrive que cette masse devienne inutilisable sans la formation de secrétariats spécialisés, de règles de dupli-

cation contraignantes (le problème de la saturation du service de copies), de personnel de liaison, etc... ; la dynamique du projet en souffre : il se sclérise dans l'administration.

Là encore, l'intérêt d'une Banque de Données "on line" est évident dès que le projet a une certaine importance.

1.3. Organisation générale d'une banque de données

1.3.1. Les modes de base

En programmation, un "operating system" permet au minimum à un utilisateur :

- de définir un algorithme
- d'assurer l'entrée des données (s'il y en a) de cet algorithme
- de sortir les résultats sur le support adéquat.

De la même façon, une banque de données générale procure à l'utilisateur un certain nombre de services élémentaires minima correspondant à 3 modes de fonctionnement :

- en mode "définition de données" on peut définir au moyen d'un langage de définition (voir § 2.1.) des modèles de structures de données complexes.
- en mode "manipulation de données" on peut créer, supprimer ou mettre à jour des données correspondant à un modèle précédemment introduit.
- en mode "récupération", le système répond à des questions concernant ces données et exprimées dans un langage d'interrogation (voir § 2.2.) que l'on peut améliorer grâce à un macrogénérateur (voir § 2.4.).

1.3.2. - Schéma général minimum

A chacun des modes que nous avons introduits au paragraphe précédent correspond un ensemble d'algorithmes généraux qui sont les suivants :

- Un traducteur (voir § 4) capable de transcrire une définition de structure sous une forme interne codée : le thésaurus.
- Des programmes généraux de création, mise à jour, suppression (voir § 5.1. et § 5.3.) capables de créer, mettre à jour ou supprimer tout ou partie des fichiers.
- Un Interpréteur de questions (voir § 5.2.) correspondant à l'activation du mode "récupération".

Enfin on dispose d'un quatrième groupe de programmes généraux d'accès aux fichiers (voir § 3).

Voir schéma page 10.

1.3.3. Fonctionnement en accès multiple (voir § 6)

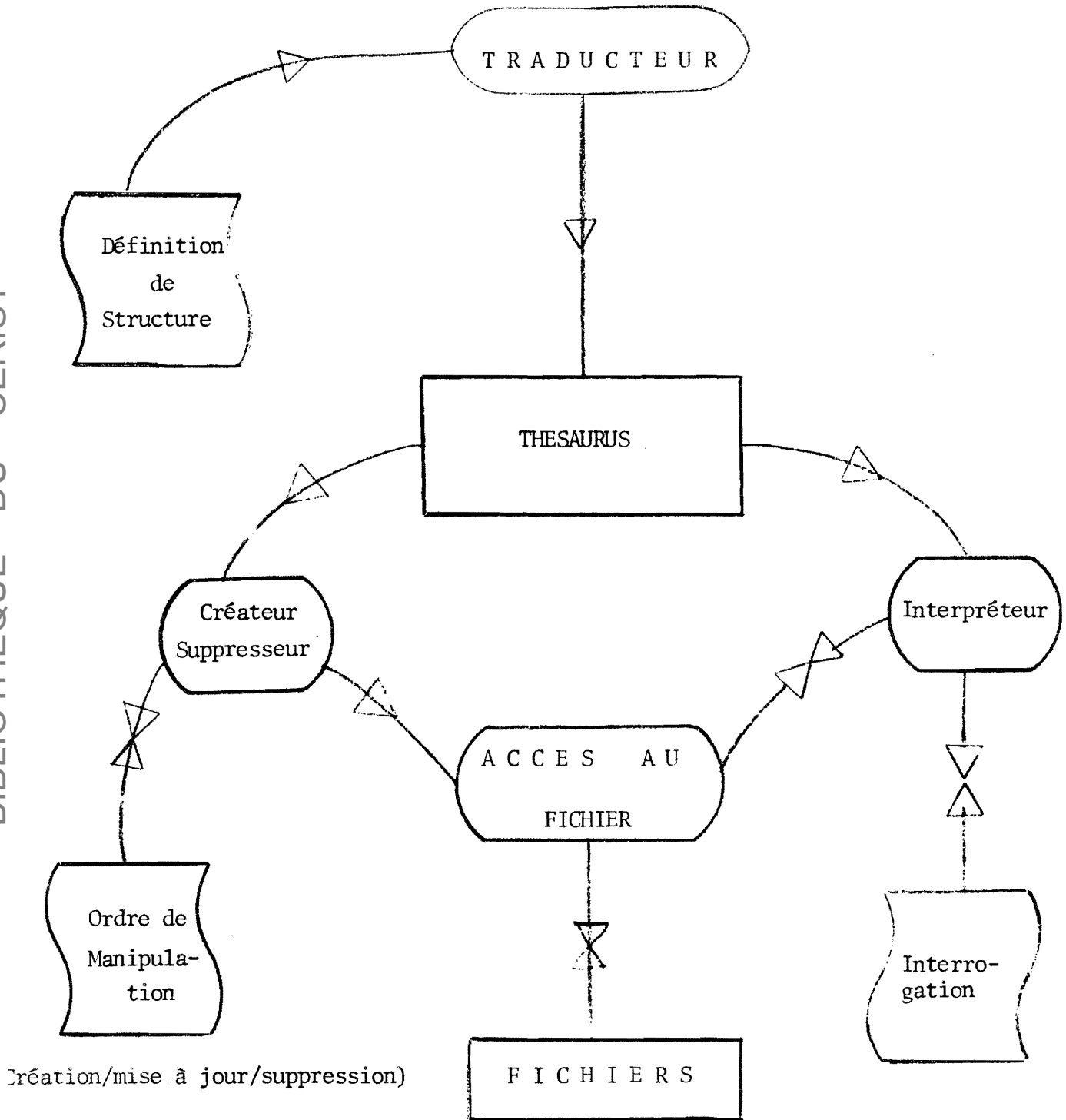
Les trois modes introduits au § 1.3.1 peuvent être activés simultanément par un grand nombre d'utilisateurs travaillant ou non sur la même base de données (la même structure formelle).

Ainsi, par exemple, avec 30 terminaux connectés à la banque de données, on peut envisager la répartition suivante :

- 10 terminaux dans un hôpital
- 2 terminaux dans une entreprise X
- 5 terminaux dans une entreprise Y
- 3 terminaux dans une entreprise Z
- 1 terminal dans une bibliothèque
- 3 terminaux dans un supermarché
- 6 terminaux dans une grande administration.

On peut de même envisager une organisation où tous les terminaux dépendent de la même base de données : l'outil général n'a alors qu'une application spécifique.

S C H E M A



1.3.4. Indépendance par rapport au hardware et au software de base

Les seuls éléments qui a priori pourraient contrecarrer cette indépendance sont les suivants :

- nécessité d'utiliser un "driver" disque ou tambour pour les liaisons élémentaires avec la mémoire secondaire
- nécessité d'utiliser un ou plusieurs "drivers" pour les terminaux
- nécessité de placer le fonctionnement de la banque sous le contrôle d'un moniteur de multiprogrammation
- obligation d'écrire les programmes de la banque dans un langage d'assez bas niveau pour qu'ils soient assez optimisés.

Les éléments qui favorisent l'indépendance sont les suivants :

- programmation modulaire de façon à ce que les relations de la banque avec les drivers et le moniteur de multiprogrammation soient très simples et nettement définies,
- écriture des programmes de la banque en utilisant un langage aussi universel que possible tout en étant assez performant,
- possibilité d'incorporer le moniteur de multiprogrammation à la banque elle-même.

Cette dernière solution rend la banque complètement indépendante de l'operating-system de la machine sur laquelle elle est implantée mais elle permet aussi de la faire tourner comme un "job" sous cet operating-system. Si celui-ci accepte la multiprogrammation on aura alors 2 moniteurs de multiprogrammation l'un sous l'autre. La lourdeur de cette construction est évidente ; elle présente cependant l'avantage de pouvoir faire fonctionner la banque en même temps que n'importe quelle autre activité classique de la machine.

- possibilité de mettre dans la banque les informations de fichiers existants. Grâce à un programme externe de transfert (voir § 5.1.4.) (dans les deux sens), une organisation qui aurait déjà ses propres fichiers peut utiliser

le système. Ce programme traduit ces fichiers "classiques" de façon à les mettre sous une forme compatible avec le mode "création". Inversement, on peut sortir tout ou partie d'un fichier de la banque de données et ce programme le transforme en un fichier classique (COBOL par exemple).

1.3.5. Modes de fonctionnement spéciaux

STATISTIQUE

Les réponses à une question ne sont plus les éléments bruts du fichier mais les résultats de comptage ou de sommations effectués sur ces éléments.

Un opérateur "par" du langage d'interrogation et les programmes correspondants (voir § 5.2.2.) permettent de ne faire qu'un seul passage sur les fichiers lorsque plusieurs résultats sont demandés.

MISE A JOUR SPONTANEE

Cette extension (voir § 5.3.2.) du mode "mise à jour" consiste à assurer un certain fonctionnement dynamique autonome des fichiers.

MODIFICATION DE STRUCTURE

Ce mode (voir § 6.3.) permet de redéfinir la structure d'un système de données auquel correspondent déjà des fichiers réels dans la banque. Les modifications à apporter aux fichiers doivent être automatisées au maximum. Nous donnons trois possibilités de solution à ce problème.

DOCUMENTATION

De façon à ne pas obliger un utilisateur à connaître entièrement la structure pour poser une question, un mode "documentation" (voir § 7) lui donne des renseignements sur celle-ci.

1.3.6. Problème de sécurité

Ce problème se présente sous les aspects suivants :

- se garantir contre un mauvais fonctionnement du système qui pourrait perturber gravement les fichiers (voir § 5.6.1.)
- assurer que seul les "ayant-droit" puissent accéder à tout ou partie des fichiers (voir § 5.6.2.)

000000000000