



**HAL**  
open science

**Contribution à la conception, la réalisation et  
l'utilisation du système de bases de données Somine :  
gestion des mémoires, enseignement assisté par  
ordinateur**

Marcel Gaillard

► **To cite this version:**

Marcel Gaillard. Contribution à la conception, la réalisation et l'utilisation du système de bases de données Somine : gestion des mémoires, enseignement assisté par ordinateur. Multimédia [cs.MM]. Ecole Nationale Supérieure des Mines de Saint-Etienne; Institut National Polytechnique de Grenoble - INPG, 1976. Français. NNT: . tel-00828716

**HAL Id: tel-00828716**

**<https://theses.hal.science/tel-00828716>**

Submitted on 31 May 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre : 21S

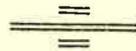
# THESE

*présentée par*

**Marcel GAILLARD**

*pour obtenir*

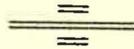
LE TITRE DE DOCTEUR DE 3° CYCLE  
SPECIALITE SYSTEMES ET RESEAUX INFORMATIQUES



CONTRIBUTION A LA CONCEPTION, LA REALISATION  
ET L'UTILISATION DU SYSTEME DE BASES DE DONNEES  
S O M I N E

GESTION DES MEMOIRES

ENSEIGNEMENT ASSISTE PAR ORDINATEUR



*Soutenu à Saint-Etienne, le 26 Janvier 1976, devant la Commission d'Examen*

MM. L. BOLLIET

**Président**

C. BELLISSANT

R. BOUCHE

S. GUIBOUD-RIBAUD

**Examineurs**

R. JACOUD

R. MAHL

THE UNIVERSITY OF CHICAGO  
LIBRARY  
540 EAST 57TH STREET  
CHICAGO, ILL. 60637  
TEL: 773-936-3200  
WWW.CHICAGO.EDU

N° d'ordre : 21S



# THESE

*présentée par*

**Marcel GAILLARD**

*pour obtenir*

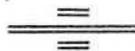
LE TITRE DE DOCTEUR DE 3° CYCLE  
SPECIALITE SYSTEMES ET RESEAUX INFORMATIQUES



CONTRIBUTION A LA CONCEPTION, LA REALISATION  
ET L'UTILISATION DU SYSTEME DE BASES DE DONNEES  
S O M M E

GESTION DES MEMOIRES

ENSEIGNEMENT ASSISTE PAR ORDINATEUR



*Soutenue à Saint-Etienne, le 26 Janvier 1976, devant la Commission d'Examen*

MM. L. BOLLIET

Président

C. BELLISSANT

R. BOUCHE

S. GUIBOUD-RIBAUD

Examineurs

R. JACOUD

R. MAHL



ECOLE NATIONALE SUPERIEURE DES MINES DE SAINT.ETIENNE.

Directeur : M. Lucien VIELLEDENT

Sous-Directeur : M. Jacques BOISSE

PROFESSEURS DE 1ère CATEGORIE

MM. COINDE Alexandre	Gestion
GOUX Claude	Métallurgie
LEVY Jacques	Métallurgie
RIEU Jean	Mécanique-Résistance des Matériaux
SOUSTELLE Michel	Chimie

PROFESSEUR ASSOCIE DE 1ère CATEGORIE

M. FORMERY Philippe	Mathématiques Appliquées
---------------------	--------------------------

PROFESSEURS DE 2ème CATEGORIE

MM. GUIBOUD-RIBAUD Serge	Informatique
LOWYS Jean Pierre	Physique
TOUCHARD Bernard	Physique Industrielle

PROFESSEURS ASSOCIES DE 2ème CATEGORIE

MM. FONTEILLES Michel	Géologie
SEROR Denis	Informatique

DIRECTEUR DE RECHERCHE

M. LESBATS Pierre	Métallurgie
-------------------	-------------

MAITRES DE RECHERCHE

MM. BISCONDI Michel	Métallurgie
BOOS Jean-Yves	Métallurgie
Mlle.FOURDEUX Angéline	Métallurgie
MM. LALAUZE René	Chimie
LANCELOT Francis	Chimie
LE COZE Jean	Métallurgie
THEVENOT François	Chimie
TRAN MINH Canh	Chimie



INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Président : M Louis NEEL

Vice-Présidents : MM. Jean BENOIT  
Lucien BONNETAIN

PROFESSEURS TITULAIRES

MM. BENOIT Jean	Radioélectricité
BESSON Jean	Electrochimie
BLOCH Daniel	Physique du solide
BONNETAIN Lucien	Chimie Minérale
BONNIER Etienne	Electrochimie et Electrometallurgie
BRISSONNEAU Pierre	Physique du solide
BUYLE-BODIN Maurice	Electronique
COUMES André	Radioélectricité
FELICI Noël	Electrostatique
LESPINARD Georges	Mécanique
MOREAU René	Mécanique
PARIAUD Jean-Charles	Chimie-Physique
PAUTHENET René	Physique du solide
PERRET René	Servomécanisme
POLOUJADOFF Michel	Electrotechnique
SILBER Robert	Mécanique des Fluides

PROFESSEURS ASSOCIES

MM. RABINS Michaël	Automatique
ROUXEL Roland	Automatique

PROFESSEURS SANS CHAIRE

MM. BLIMAN Samuel	Electronique
COHEN Joseph	Electrotechnique
DURAND Francis	Métallurgie
FOULARD Claude	Automatique
LANCIA Roland	Electronique
VEILLON Gérard	Informatique fondamentale et appliquée
ZADWORNY François	Electronique

MAITRES DE CONFERENCES

MM. BOUDOURIS Georges	Radioélectricité
BOUVARD Maurice	Génie mécanique
CHARTIER Germain	Electronique
GUYOT Pierre	Chimie Minérale
IVANES Marcel	Electrotechnique
JOUBERT Jean-Claude	Physique du solide
LACOUME Jean Louis	Géophysique
MORET Roger	Electrotechnique Nucléaire
ROBERT François	Analyse numérique
SABONNADIÈRE Jean-Claude	Informatique fondamentale et appliquée
Mme. SAUCIER Gabrièle	Informatique fondamentale et appliquée

CHARGE DE FONCTIONS DE MAITRE DE CONFERENCES

MM. ANCEAU François	Mathématiques Appliquées
PIERRARD Jean-Marie	Hydraulique.

CHERCHEURS DU C.N.R.S.

MM. FRUCHARD Robert	Directeur de recherche
ANSARA Ibrahim	Maître de recherche
DRIOLE Jean	Maître de recherche
MATHIEU Jean-Claude	Maître de recherche
MUNIER Jacques	Maître de recherche



A MES PARENTS



Je remercie,

Monsieur L. VIELLEDENT, Directeur de l'Ecole Nationale Supérieure des Mines de Saint-Etienne, qui a bien voulu m'accueillir dans son établissement pour effectuer le travail exposé dans ce mémoire.

Monsieur L. BOLLINET, Professeur à l'Université Scientifique et Médicale de Grenoble qui m'a fait l'honneur de présider ce jury.

Monsieur R. MAHL, qui a guidé le démarrage de cette étude, alors qu'il était chef du département Informatique de l'E.M.S.E.

Monsieur S. GUIBOUD-RIBAUD, Professeur, Chef du Département Informatique de l'Ecole, qui a permis à ce projet d'aboutir.

Monsieur R. BOUCHE, Professeur à l'Université de LYON I, qui a dirigé l'ensemble de ce travail et auprès de qui j'ai toujours trouvé aide efficace et encouragements.

Monsieur R. JACQUOT, professeur à l'Université de PARIS VII pour les idées et conseils qu'il a bien voulu me donner pour le travail de recherche sur l'Enseignement Assisté par Ordinateur.

Monsieur C. BELLISSANT, Maître-Assistant à l'Université Scientifique et Médicale de Grenoble pour l'intérêt qu'il a porté à ce travail.

Monsieur R. THOMAS, Chef du Centre de Calcul de l'E.M.S.E., qui m'a toujours apporté une aide spontanée.

Mes collègues et amis, J.P. SUDOUR et J. FOURNEL qui ont grandement contribué à la conduite des recherches sur l'analyse et la résolution des problèmes de statique d'une part, et sur l'essai de rééducation d'enfants dyslexiques d'autre part.

Tous les chercheurs, techniciens et étudiants du Département Informatique et du Centre de Calcul de l'Ecole pour leur collaboration et leur accueil sympathique.

.../...



Mes collègues de l'IUT qui ont su m'aider et me supporter pendant la durée de ces travaux.

Tous ceux qui ont contribué à la réalisation matérielle de ce document, et principalement :

Mademoiselle J. DEFOUR de l'I.U.T.

Mesdames G. BONNEFOY, C. CHAMBON, M.C. MONTMARTIN, Messieurs LOUBET et BROSSARD de l'E.M.S.E.

Enfin mes deux complices, P. MARTY et C. SAYETTAT avec lesquels j'ai réalisé ce travail dans l'amitié et la bonne humeur.



## SOMMAIRE DE LA THESE

	Préface
Partie I	Sommaire et présentation générale du projet
Partie II	La gestion des mémoires du système de bases de données SOMINE
Partie III	SOMINE et l'enseignement assisté par ordinateur
Partie IV	Conclusion
	Fiches signalétiques

REMARQUE : chaque partie commence par une feuille de couleur .

A l'intérieur de chaque partie, les pages sont numérotées



## PREFACE

Ce rapport résume le travail effectué pendant les deux dernières années par une équipe qui mène une recherche commune depuis trois ans. Nous sommes trois enseignants du département Génie Mécanique de l'Institut Universitaire de Technologie de Saint-Etienne. Réunis par nos travaux en informatique, nous avons été influencés dans le choix de notre sujet par notre profession commune et nos formations très diversifiées.

La réunion de l'informatique et de l'enseignement devait nous attirer à priori. Mais, tant dans notre pratique quotidienne (algorithmisation des solutions de problèmes, des raisonnements ..... etc) que dans un travail de recherche de D.E.A. à l'Université de Lyon I (2), nous avons rencontré les limites de l'apport de l'Enseignement Assisté par Ordinateur de forme tutorielle. Nous devons chercher autre chose !

Les rencontres avec Monsieur VIELLEDENT, Directeur de l'Ecole Supérieure des Mines de Saint-Etienne, très intéressé par les problèmes pédagogiques et avec Monsieur MAHL, Directeur du département informatique de cette école, ont contribué à définir les buts de notre recherche : implémenter un système de banques de données accessibles à des utilisateurs "étudiants" (aide à l'enseignement) "ingénieurs et techniciens" (aide à la conception assistée par ordinateur) ou "gestionnaires" (aide à la gestion).

Cet objectif fixé, nous avons étudié les réalisations françaises et étrangères dans ces domaines. Ainsi, une importante bibliographie a été consultée. Son analyse critique nous a conduit à préciser les grandes lignes de notre recherche.

Ces éléments sont résumés dans la première partie (introduction générale) de ce mémoire qui montre comment notre travail s'est orienté suivant

deux axes principaux :

a) La conception et l'implémentation d'un système de gestion de bases de données ( SOMINE )

b) Les recherches montrant comment ce système peut être appliqué à des domaines aussi divers que l'E.A.O., la C.A.O. ou l'optimisation de la structuration des informations.

Un tel travail ne pouvait être réalisé que par une équipe et c'est bien dans ce cadre que les choix principaux ont été fixés et les diverses solutions discutées. Mais, dès le début de notre travail, nous avons eu soin de donner à chacun la responsabilité d'une partie du travail. Grâce à cela, nous présentons aujourd'hui trois thèses de DOCTORAT du TROISIEME CYCLE qui, issue d'un travail commun, sont cependant indépendantes.

Elles portent sur les sujets suivants :

\* Interface entre SOMINE et ses utilisateurs et aide à la Conception Assistée par ordinateur (présentée par Cl. SAYETTAT )

\* Structuration des informations dans SOMINE et recherche des structures optimales (présentée par P. MARTY)

\* Gestion des mémoires de SOMINE et Enseignement Assisté par Ordinateur (présentée par M. GAILLARD)

Conscients d'avoir rencontré plus de questions que nous n'en avons résolues, nous espérons apporter cependant, par ce travail, une certaine contribution à la solution des problèmes concernant les banques de données et leurs interfaces avec des applications diverses.

**PREMIERE PARTIE**

**PRESENTATION GENERALE DU PROJET**



**SOMMAIRE DE LA PREMIERE PARTIE**  
**(INTRODUCTION GENERALE)**

		PAGE
1 :	<u>LES BUTS DE NOTRE PROJET</u>	I. 3
2 :	<u>DE LA "RECHERCHE DOCUMENTAIRE" aux "QUALITES SOUHAITABLES"</u>	7
	2.1 - LES BANQUES DE DONNEES ET L'ENSEIGNEMENT	7
	2.2 - LES BANQUES DE DONNEES ET LA CONCEPTION DES PROJETS SCIENTIFIQUES ET INDUSTRIELS	8
	2.3 - LES BANQUES DE DONNEES ET LA GESTION	10
	2.4 - LES QUALITES SOUHAITABLES	11
3 :	<u>DE SOCRATE A SOMINE</u>	13
	3.1 - QUELQUES RAPPELS SUR SOCRATE	13
	3.1.1. - La gestion de mémoire	14
	3.1.2. - La structuration des informations	14
	3.1.3. - Le langage de requête	15
	3.1.4. - Le langage de commande	16
	3.1.5. - La programmation	17
	3.2 - SOCRATE ET NOS EXIGENCES	17
	3.2.1. - Choix de certains éléments de SOCRATE	17
	3.2.2. - Les différences indispensables	18
	3.3 - SOMINE	20
4 :	<u>CARACTERISTIQUES GENERALES DU SYSTEME ET PLAN DU RAPPORT</u>	21
	4.1 - LA PROGRAMMATION	21
	4.2 - LA CONFIGURATION GENERALE DU SYSTEME	21
	4.3 - VUE D'ENSEMBLE SUR LE TRAVAIL REALISE	24
	4.4 - CONTENU DU PRESENT RAPPORT	27
ANNEXE :	<u>BIBLIOGRAPHIE DE LA PREMIERE PARTIE</u>	29



## 1. - LES BUTS DE NOTRE PROJET

De par sa composition et son travail antérieur, notre équipe était sensibilisée aux problèmes de :

- l'enseignement assisté par ordinateur,
- l'aide à la conception et la simulation des phénomènes scientifiques ou techniques.

Les ordinateurs ont été utilisés largement pour l'un ou l'autre de ces deux thèmes, car, dans chacun d'eux, on trouve une partie importante de recherche documentaire.

La documentation peut utiliser des supports bien différents. Le critère essentiel pour qualifier ces supports est le temps d'accès à l'information. D'où l'idée, aujourd'hui banale, d'utiliser des mémoires d'ordinateurs comme support d'information. Il y a trois avantages à cela :

- les mémoires, disques ou bandes, sont vastes
- les temps d'accès sont beaucoup plus courts que sur les supports traditionnels ( livres, etc...)
- on peut facilement actualiser les informations (effacement, création, mise à jour .... )

Documenter étant au centre de nos préoccupations, deux voies pouvaient être choisies :

- créer des systèmes spécialisés et distincts pour assurer l'enseignement, l'aide à la conception ou la simulation ( jeux d'entreprises..)
- créer un seul système composé de centres documentaires importants et de sous-systèmes assurant les interfaces de ces centres avec ses divers "clients", qu'ils soient des "étudiants" désirant un certain "cours", des "techniciens" voulant comparer un projet à l'ensemble des réalisations existantes, des "gestionnaires" ou des "scientifiques" voulant utiliser un grand nombre d'informations dynamiques.

Les systèmes spécialisés existent. Les avantages et les inconvénients des systèmes spécialisés dans l'enseignement assisté par ordinateur sont connus (1). Nous avons nous-même utilisé un tel système lors de notre travail de D.E.A. à Lyon (2). Notre cours de FORTRAN, opérationnel depuis deux ans, nous a permis de connaître les opinions des élèves ; ceux-ci sont généralement satisfaits mais se plaignent parfois d'un certain manque de souplesse. Les systèmes spécialisés dans l'aide au projet (voir notamment le remarquable travail de M. LATOMBE (3) ) donnent plus de responsabilité (donc de créativité) aux étudiants mais demandent en parallèle, un cours, sur la spécialité de projet auxquels ils apportent leur aide. Quant aux jeux d'entreprises et autres simulations de phénomènes, ils font toujours appel à beaucoup d'informations que leurs programmeurs ont bien du mal à stocker dans les fichiers traditionnels.

D'où l'idée de construire un système unique, comprenant notamment :

- une banque de données qui pourrait contenir toutes les informations nécessaires à chacune des applications,
- un langage de cours permettant de gérer les relations entre les étudiants et la banque,
- des programmes utilisateurs faisant appel à la base et permettant de résoudre en calcul ou simulation des problèmes scientifiques ou techniques.

A partir de ces objectifs qui devaient conduire à :

- des réalisations en E.A.O. non directif,
- des applications nécessitant l'interrogation et la mise à jour de grandes quantités d'informations en temps réel (gestion, aide au projet, aide aux calculs scientifiques et techniques, etc...) il nous appartenait de :

- rechercher et étudier la documentation nécessaire,
- faire des choix et en déduire les différentes parties du projet global,
- écrire des programmes,
- trouver les applications concrètes et les mettre en oeuvre,
- juger les qualités de notre travail à partir de son efficacité dans ces applications.

A partir de la recherche documentaire qui précède toutes les réalisations, nous avons dégagé les idées générales et les caractéristiques essentielles de notre projet. Nous verrons de quelle manière dans le prochain chapitre.



## 2. - DE LA "RECHERCHE DOCUMENTAIRE" AUX ...

### .. "QUALITES SOUHAITABLES"

L'utilisation des banques de données n'est pas nouvelle. Les constructeurs, eux-mêmes, ont mis au point, pour leurs machines, des logiciels spécifiques adaptés aux problèmes documentaires.

Cependant, leurs systèmes sont surtout adaptés aux problèmes de gestion (3). Ainsi, pour rechercher les qualités nécessaires à la satisfaction de nos différents objectifs, il est important de comprendre les questions qui se posent dans l'utilisation des banques de données pour :

- l'enseignement assisté par ordinateur,
- la mise au point des projets industriels ou scientifiques,
- la gestion..

#### 2. 1 - LES BANQUES DE DONNEES ET L'ENSEIGNEMENT :

La place des banques de données dans l'enseignement est encore restreinte. Dans leur rapport; "l'enseignement assisté par l'ordinateur en Amérique du Nord"(1), KAISER et COULON, en distinguent deux utilisations principales :

a) - banques de données utilisées comme instrument de recherche de faits (FACT RETRIEVAL). On désigne par là toutes les méthodes qui se bornent à retrouver, dans la banque de données, les informations à communiquer aux étudiants. Il n'y a aucun travail sur ces informations. Cette méthode a notamment été utilisée par l'équipe de Jaime CARBONNEL à Cambridge (10). Son principal inconvénient est la rigidité : toutes les réponses, tous les textes sont "prévus par le maître" et fixés une fois pour toutes dans la banque. L'avantage par rapport à un cours d'E.A.O. tutoriel classique n'est pas évident.

b) - banques de données utilisées comme support d'instrument déductif. Il s'agit d'utiliser les données stockées dans la banque, non pas d'une manière "brute" comme ci-dessus (a), mais comme éléments de l'information à fournir à l'étudiant. Une réponse à donner, par exemple, sera déduite de la question de l'étudiant et des informations contenues dans la banque. On connaît les travaux réalisés sur cette question par R.C. SCHANK (6), et son équipe (7). On sait aussi que, malgré des résultats remarquables le problème posé par la déduction n'est pas résolu de manière satisfaisante. Ou bien la déduction est complètement programmée et on retombe dans la fixité reprochée aux systèmes, vue en (a), ou bien elle s'effectue aveuglément ce qui demande des temps de réponse trop longs pour utilisation en mode conversationnel, c'est à dire en enseignement.

Un compromis semble pouvoir être recherché. C'est dans ce sens que vont les travaux de WINOGRAD (8) : il veut guider la recherche des informations dans la banque par une stratégie tenant compte, notamment, de ce qui "est possible dans le contexte donné". Appliquée à la pédagogie, il nous semble que cette voie peut être féconde, surtout si on la rapproche de l'idée émise par CHANDRASEKARAN (et rapportée par KAISER et COULON).

Il propose de "construire une banque de données conversationnelle qui demanderait à l'étudiant si la voie qu'elle emprunte pour accéder à sa requête lui paraît bonne". Qu'importe, en effet, la rigidité des "atomes" à enseigner si on peut les parcourir dans un ordre choisi, avec possibilité de retour en arrière, de tests, de construction de problèmes à partir de données choisies par l'étudiant ....

## 2. 2 - LES BANQUES DE DONNEES ET LA CONCEPTION DES PROJETS SCIENTIFIQUES ET INDUSTRIELS :

Dans ce domaine également, il semble que les bases de données n'occupent pas la place qui pourrait être la leur.

Faire un "projet", c'est, presque toujours, réaliser un compromis entre l'abstrait et le réel, entre la rigueur des théories scientifiques et les exigences de la réalité. Un bon "projeteur" doit connaître les réalisations existant dans son domaine de travail et maîtriser les théories scientifiques nécessaires. Tout cela constitue d'importantes masses de documentations, de connaissances qu'il faut actualiser sans cesse. Il est certain que l'utilisation de bases de données adaptées à ces problèmes faciliterait

ce travail. L'exemple des "projets d'électrotechniques" étudiés par M.LATOMBE (3) est significatif. Il faut stocker :

- des matériaux dont les caractéristiques et le nombre sont sans cesse variables,
- des coefficients modifiés chaque jour par de nouvelles découvertes techniques,
- des algorithmes de calculs qu'il faut adapter à de nouvelles formes de machines ou de composants
- etc.....

Et tout cela doit être rapidement accessible et mis à jour, puis utilisable pour des calculs sur ordinateurs.

A travers ce qui précède se dessine le rôle des banques de données comme "puissants dictionnaires du passé technique et scientifique". Mais les banques de données peuvent intervenir dans la conception et la recherche de bien d'autres façons. Par exemple, on peut se servir des informations contenues dans une banque pour simuler le fonctionnement d'un prototype. Ont été simulés ainsi des mécanismes simples (11), les machines électrotechniques dans le programme "ESPACE" (3), les caractéristiques principales d'un moteur thermique à essence fonctionnant à différents régimes (10). Ce dernier exemple est particulièrement intéressant. On peut schématiquement le décrire de la manière suivante :

Dans la banque de données, on stocke les informations :

- type du moteur
- nombre de tours par minute du vilebrequin
- cylindrée
- consommation d'essence
- couple moteur
- puissance mécanique
- rendement
- etc..

La banque peut alors être utilisée de différentes manières :

a) - Un étudiant peut "manipuler" un moteur comme s'il était tout à fait réel. Par exemple, il peut tracer, en fonction de la vitesse de rotation du vilebrequin, les courbes classiques de consommation, de puissance et de rendement...

b) - Un constructeur peut consulter la banque pour connaître les principales caractéristiques de tous les moteurs existants qui donnent une puissance de 60 kw à 4000 tr/mn ....

c) - Un projeteur, ayant dessiné un nouveau moteur pourra, en fonction des dimensions et autres caractéristiques qu'il a imposées à son projet, connaître "où se trouvera approximativement son moteur dans la gamme des existants" .... Il pourra ainsi répondre partiellement à la question : "faut-il continuer l'étude ? " avant la construction d'un prototype...

d) - etc .....

Enfin, parlons de l'intervention des banques de données dans la création elle-même. Toute recherche commence généralement par une recherche d'idées. Les techniques de la créativité sont connues ( brainstorming, listes de questions, méthodes analogiques, logiques ..... etc). Mais, toutes les idées étant trouvées la question qui se pose est la suivante : comment en dégager la solution optimale ? En général, c'est un "exploit" que l'on réserve aux personnes "d'expérience" ! Or, une approche possible est la comparaison des différentes solutions à l'aide de critères quantitatifs ou qualitatifs soigneusement hiérarchisés ( MAUFIT ( 13), GEMINARD (14), SARAZIN (12)). Dans cette recherche l'analyse combinatoire joue un rôle important et on comprend la place que pourrait y tenir une banque de données dont les informations seraient accessibles directement par des programmes classiques de calcul.

## 2. 3 - LES BANQUES DE DONNEES ET LA GESTION :

Depuis 1960, le développement des S.G.B.D. est étroitement lié au développement de l'informatique de gestion.

L'apparition des premiers systèmes de gestion de base de données IDS ( Integrated Date Store), TDMS ( Time-Share Data Management System) (9), IMS ( Information Management System) puis de SOCRATE et DBTG-CODASYL a permis de répondre en grande partie aux besoins des utilisateurs qui désiraient éviter les lacunes des méthodes traditionnelles de l'informatique de gestion ( redondance des informations, rigidité des fichiers, difficultés d'accès aux informations...)

Actuellement, nous pouvons constater un intérêt grandissant pour les systèmes de gestion de bases de données comme en témoignent d'une

part la multiplication des rencontres organisées sur ce sujet (niveaux recherche ou utilisateur), d'autre part la rapide augmentation du nombre des logiciels disponibles et des installations existantes.

A titre d'exemple citons :

- TOTAL (développé par CINCOM système) qui compte près de 800 installations dans le monde entier dont une dizaine en France.
- I.M.S. qui est utilisé en France par de nombreux groupes bancaires, des assurances, des grandes entreprises nationalisées ou privées (4)
- I.D.S. qui compte plus de 500 installations dans le monde.
- SOCRATE plus récent dont le nombre d'installations est voisin de 40 ( 5 ).
- ADABAS, FORTE, D.M.S., GIM, SYSTEME 2000 , MISFIIT ....

Notons que la taille trop grande des systèmes disponibles ne permet pas encore un développement dans les petites et moyennes entreprises et que certains problèmes de performance ( temps d'accès ... ) seront résolus lorsque le " hardware" aura rattrapé son retard sur la technologie de conception des bases de données.

#### 2. 4 - LES QUALITES SOUHAITABLES :

A travers la "recherche documentaire" concernant les trois types d'applications envisagées, il apparait clairement qu'un système de bases de données peut apporter des éléments nouveaux dans les recherches concernant ces trois domaines.

Mais pour cela, il est indispensable que le système de base de données utilisé possède un certain nombre de qualités principales qu'il paraît possible de résumer ainsi :

- a) - Indépendance du système par rapport au matériel informatique utilisé : cette qualité assurera la "transportabilité" du système de bases de données et permettra de comparer éventuellement des matériels.
- b) - Fiabilité et garantie du secret : ces qualités sont indispensables à tout système de bases de données.

c) - Adaptabilité du système : cette qualité est particulièrement importante. Il ne faut pas présenter un système "rigide", "non modifiable" mais, au contraire, un système qui, tout en étant complet, peut être "adapté" à des applications particulières : la modularité de la programmation et du système lui-même devront autoriser l'utilisateur à compléter, supprimer ou ajouter certaines fonctions.

d) Mesurabilité des caractéristiques du système : Ceci est nécessaire pour évaluer ses performances mais aussi, par exemple, pour mesurer les effets d'un changement de structuration ou pour procéder à certaines expériences à partir de la variation des valeurs des caractéristiques du système en vue de son optimisation future.

e) - Traitement efficace des informations : cela comporte notamment :

- \* la rapidité des opérations d'interrogation et de mise à jour
- \* la souplesse dans la structuration et la recherche des informations (plusieurs types de structures doivent être possibles, la recherche des informations doit être rapide et pouvoir utiliser plusieurs moyens : accès direct, critères de filtrages, etc...)
- \* la variété dans le choix des moyens de mise à jour et d'accès aux informations situées dans la base : les modes conversationnel et batch sont indispensables mais il faut surtout que des programmes puissent accéder aux informations, éventuellement les transformer puis les ranger à nouveau dans la base.
- \* la variété dans les centres d'informations possibles: listes de valeurs, mots, textes, numériques de tous types, chaînes de bits etc....

Cet ensemble de "qualités souhaitables" constitue une sorte de "CAHIER DES CHARGES" du système à réaliser. Il peut servir à comparer nos objectifs aux qualités présentées par les réalisations existantes.

### 3. - DE SOCRATE A SOMINE

A partir du "cahier des charges" défini à la fin du chapitre précédent, il convenait d'examiner les systèmes de bases de données existants pour les comparer et ..... pour éviter la reproduction de l'un ou de l'autre de ces systèmes.

Dans le paragraphe 2.3, nous avons donné la liste des systèmes de base de données dont nous avons consulté la documentation et, dans la conclusion de ce mémoire, nous reviendrons sur la comparaison de notre "produit" avec certains de ces systèmes.

Nous voulons cependant, dès cette introduction, traiter d'une manière spéciale le SYSTEME SOCRATE (5) car le groupe dans lequel nous avons travaillé au début de notre projet ( MM. MAHL et THOMAS notamment) était en relation avec l'équipe SOCRATE de Grenoble. Nous avons donc pu :

- entrer en contact avec cette équipe,
- obtenir toute la documentation sur le système SOCRATE.

De ce point de vue, on peut dire que SOMINE est un des fils de SOCRATE..... ce qui ne veut pas dire que le fils est une "copie" du père !. Nous verrons au contraire, dès ce chapitre et dans toute la suite de ce mémoire, que ces deux produits sont très différents.

Remercions cependant les concepteurs et les réalisateurs de SOCRATE qui nous ont reçus plusieurs fois à Grenoble. Il est certain que leur réalisation nous a fortement influencés dans le début de nos travaux. Examinons leur produit.

#### 3. 1 - QUELQUES RAPPELS SUR SOCRATE :

Ce paragraphe se contente de rappeler sommairement quelques éléments importants sur le système SOCRATE. Il suppose connues les nombreuses

brochures dans lesquelles ce système a été décrit et, principalement la publication citée en (5).

### 3. 1. 1 - La gestion de mémoire :

Une base de données SOCRATE utilise deux espaces mémoires secondaires : un espace "tambour" pour la structure et un espace "disque" pour les informations structurées ( données ).

A chacun de ces espaces secondaires correspond un espace "virtuel" de grande taille ( $2^{31}$  mots) dans lequel toute information a une adresse virtuelle. L'unité de passage de l'espace virtuel à l'espace mémoire secondaire est appelée "sous-page" ( bloc de  $2^n - 1$  mots consécutifs.

L'unité de passage entre les mémoires secondaires et la mémoire centrale de l'ordinateur est appelée "page" : bloc de m sous-pages consécutives.

Une information subit donc les transferts suivants :

Information  $\longrightarrow$  espace virtuel  $\longrightarrow$  espace disque  $\longrightarrow$  espace tore  
 (programmeur) (adresse virtuelle) (adresse disque) (adresse en mém. centrale)

Les applications permettant les passages entre les divers espaces mémoires sont réalisées par des fonctions de réallocations programmées, utilisant les méthodes de l'adressage dispersé, de l'adressage calculé et des tables ( dictionnaire de gestion de mémoire). Elles sont entièrement transparentes à l'utilisateur de la banque.

### 3. 1. 2 - La structuration des informations :

La structure SOCRATE de base est arborescente. Elle ordonne des caractéristiques de divers types : valeurs numériques entières, listes de valeurs, mots, textes, entités. Ces caractéristiques peuvent être "simples" ou "complexes", regroupées en "blocs" si on le désire. Les "entités" sont des "blocs à caractère répétitif".

exemple :      ENTITE 100 PERSONNES  
                   DEBUT  
                   NOM    MOT 6  
                   FIN

Cette entité permettra le stockage des noms de 100 personnes (au plus).

Le COMPACTAGE DES DONNEES est une qualité essentielle de SOCRATE car :

- les valeurs numériques, définies par leur plage de variations sont codées sur des frontières de bit,
- les mots sont codés sur des frontières d'octet,
- les textes sont codés sur des frontières de mot.

On peut ainsi gagner de la place sur les fichiers traditionnels de trois façons :

- non redondance des informations (propriété de toutes les banques d'informations)
- compactage,
- choix de la meilleure structure à partir du critère : "meilleur compactage".

Les structures SOCRATE admettent un certain nombre de "caractéristiques particulières" qui accroissent de façon importante la souplesse des mises à jour et des interrogations et diminuent les temps nécessaires à ces opérations. Citons par exemple :

- la caractéristique "REFERENCE" : elle transforme la structure arborescente classique en une structure de GRAPHE
- les caractéristiques "discriminante rapide" et "inverse" qui diminuent les temps d'accès à certaines informations.
- certains filtres : "alors", "pour" .... qui permettent des sélections particulières d'information.

### 3. 1. 3 - Le langage de requête :

C'est le langage du dialogue entre la banque et ses utilisateurs. Dans ce langage, on utilise une technique de type COBOL ( ou PL1) consistant à désigner tous les niveaux de blocs pour arriver à l'objet auquel on s'intéresse. Mais la présence des entités impose la création de "FILTRÉS" servant à caractériser plus précisément l'objet recherché. Les principaux filtres sont :

- AYANT : filtre de base généralement utilisé dans les requêtes. Il définit une réalisation possédant une ou plusieurs caractéristiques de valeur ( de type) particulière .

- les filtres utilisés à la suite du filtre "AYANT" : " UN" "DERNIER" ( positionnement dans l'entité), " LUI-MEME" et "EN COURS" (positionnement sur la caractéristique dont on vient de s'occuper ou sur laquelle on travaille actuellement), UN et TOUT ( nombre d'éléments auxquels on veut s'adresser) ..... etc.

- les filtres combinant des expressions " BOOLEENNES" : "ET", "OU".

- les démonstratifs : ils permettent de caractériser l'élément par son numéro d'ordre dans l'entité ; exemple : NOM D'UNE PERSONNE X (I) fournira le nom de la 50<sup>ième</sup> personne si  $X(I) = 50$ .

- filtre " TEL QUE" ...

Ce langage, assez complexe peut être allégé par l'utilisation de "MACROS". Un programme " MACROGENERA TEUR " transforme alors le texte allégé en un texte normal, (rédigé en langage de citation ordinaire) qui est transmis au compilateur puis à l'interpréteur de SOCRATE.

### 3. 1. 4. - Le langage de commande :

Il complète le langage de requête en précisant le "mode" du travail à réaliser ou certaines particularités de ce travail. Il autorise les opérations de :

MISE A JOUR : création, effacement et mise à jour

ADDITION : on peut ajouter un nouveau représentant à une classe d'entité

INTERROGATION et DENOMBREMENT : permet de connaître le nombre des réalisations effectives d'une entité

SURVEILLANCE : en traitant une certaine partie de la structure de manière "préférentielle", cette commande autorise l'allègement du langage de citation.

POUR : permet l'utilisation de "groupements de commandes" : par exemple, plusieurs interrogations seront réalisées par une seule requête (interrogation itérative).

### 3. 1. 5. - La programmation :

Socrate a été programmé en autocode I.B.M. à Grenoble puis une version a été redéfinie pour les ordinateurs produits par la compagnie internationale d'informatique (série IRIS).

Dans les deux cas, le système n'est pas indépendant de la machine.

### 3. 2 - SOCRATE ET NOS EXIGENCES :

Comment le système SOCRATE se situe - t - il par rapport aux "qualités souhaitables" résumées dans notre cahier des charges ?

Répondre à cette question c'est à la fois :

- choisir certains éléments de Socrate qui pourront être utiles pour la conception et l'implémentation de SOMINE
- Montrer en quoi SOMINE doit différer de SOCRATE de manière fondamentale.

#### 3. 2. 1. - Choix de certains éléments de Socrate utilisables dans SOMINE :

Certaines idées originales du système SOCRATE peuvent être conservées : il en est ainsi dans la GESTION DE MEMOIRE pour :

- \* l'utilisation d'un fichier-structure et un fichier-données
- \* la virtualisation de la mémoire et le partage de la mémoire réelle en espace disque et espace tore.

Dans la structuration des informations on peut conserver :

- la structuration arborescente dans le cas général,
- la caractéristique "référence" dans les cas où des structures de "graphes" sont indispensables.
- l'idée des "filtres" mais en transformant considérablement leur conception et leur utilisation car les filtres SOCRATE nous paraissent trop complexes, parfois redondant et pas tous utiles.

Au niveau du langage de requête, il faut retenir :

- la désignation successive de tous les niveaux de la structure qui permettent d'arriver jusqu'à l'objet (ou le groupe d'objets-entité) auquel on s'intéresse.

- les démonstratifs qui achèvent, de façon claire, cette détermination.

- certains éléments du langage (UN, TOUT, etc) mais il est absolument indispensable de simplifier et d'élaguer ce langage de requête si on veut le rendre facilement utilisable par des programmes.

Enfin, les modes "création", " interrogation", " mise à jour" "suppression", sont indispensables tandis que le "dénombrement" des réalisations effectivement créées est seulement utile.

La simplification du langage de requête devrait rendre sans objet les modes " surveillance" et "pour".

Au fil de cette analyse, on voit qu'il faut surtout transformer les parties qui sont plus ou moins directement en relation avec les utilisateurs. La comparaison des qualités de Socrate avec les qualités que nous avons estimées importantes confirme ce point de vue.

### 3. 2. 2. - Les différences indispensables :

Reprenons nos critères un à un en insistant maintenant sur les aspects nouveaux que SOMINE doit présenter par rapport à SOCRATE.

#### 3.2.2.I - Indépendance du système par rapport au matériel informatique :

Les réalisations actuelles du système SOCRATE ne sont pas transportables. Ainsi, une adaptation à un nouveau matériel impose une nouvelle programmation des algorithmes. SOMINE, au contraire, DOIT ETRE TRANSPORTABLE. Pour cela, nous devons programmer dans un langage algorithmique.

#### 3.2.2.II - Fiabilité et garantie du secret :

SOCRATE est fiable et le secret des informations y est correctement assuré. SOMINE n'apportera pas de nouveautés sur ce point.

#### 3.2.2.III - Adaptabilité du système :

SOCRATE est un système assez vaste et rapide. Ces qualités ont pour revers des difficultés d'adaptation en vue d'une application particulière. Sa programmation en langage autocode ne permet pas à l'utilisateur moyen de comprendre son fonctionnement.

La programmation de SOMINE devra être la plus COMPREHENSIBLE possible (choix du FORTRAN, modularité). Sur ce point SOMINE et SOCRATE seront donc très différents.

### 3.2.2.IV - Mesurabilité des caractéristiques du système :

Toutes les caractéristiques de SOCRATE (du nombre de pages en mémoire centrale au temps de réponse dans telle condition de fonctionnement) ont été mesurées. Il reste cependant un certain nombre d'expériences à réaliser. Nous pensons notamment à des expérimentations sur la taille des pages, sur l'effet de la présence ou de l'absence du cadre de manoeuvre, sur le nombre optimum de pages à garder simultanément en mémoire centrale.... etc. Les transformations de programmes nécessaires pour ces expériences ne sont pas faciles sur le système SOCRATE. Il faudra que SOMINE les permettent. D'autre part, il faudrait pouvoir compter les nombres d'interrogations et de mises à jour afin d'effectuer des recherches statistiques sur le fonctionnement de la base.

### 3.2.2.V - Traitement efficace des informations :

- a) Le temps de réponse de SOCRATE est satisfaisant. SOMINE, dans sa version originale essaiera seulement de... faire aussi bien.
- b) Les divers modes de structuration des informations ainsi que la variété des moyens d'accès sont un des atouts de SOCRATE. Par contre, la complexité du langage de requête est un défaut; il est souhaitable que l'utilisateur de la banque dispose d'un langage plus clair, plus proche du langage naturel. SOMINE s'efforcera de joindre cette qualité à une rapidité d'accès convenable.
- c) SOCRATE peut être utilisé à partir de télétypes en mode conversationnel. Mais, les programmes ne peuvent pas utiliser directement la banque et la banque ne peut pas lancer l'exécution d'un programme. Au contraire, pour atteindre ces objectifs (notamment pour le calcul scientifique) il est INDISPENSABLE QUE LE SYSTEME SOMINE PUISSE A LA FOIS :
  - être accessible par tout programme en langage évolué
  - provoquer l'exécution de programmes.

A partir de ce critère apparaît une des différences essentielles entre les deux systèmes.

En effet, l'exécution de programmes à partir de la base elle-même est une propriété très importante car, à partir d'une seule requête on pourra déclencher l'exécution d'un nombre quelconque de requêtes et de programmes. Cela ouvre la voie à l'utilisation directe d'une banque de données dans les domaines de la conception assistée par ordinateur, du calcul scientifique et technique et résoud le problème des actions automatiques sur les informations.

d) Tous les types d'informations retenus dans SOCRATE le seront également ans SOMINE ( entités, listes de valeurs, textes, mots, numériques entiers). Mais les calculs souhaités (aide à la conception, par exemple) imposent d'autres types d'informations : les valeurs numériques REELLES et, surtout, les PROGRAMMES.

### 3.3 - SOMINE :

SOMINE est donc fils de SOCRATE.

Mais SOMINE n'est pas SOCRATE car la discussion précédente fait apparaître trois sortes de différences :

-au niveau du mode et du langage de programmation

-au niveau des types d'informations et, surtout des types d'UTILISATEURS puisque SOMINE acceptera comme utilisateurs des PROGRAMMES EN LANGAGE EVOLUE.

-au niveau des langages de définition de structure et de requête : ces langages seront simplifiés et clarifiés afin de limiter au minimum le besoin de formation des utilisateurs de SOMINE.

Ayant défini ce que "nous souhaitions réaliser" dans les deux chapitres dont la lecture s'achève ici, nous devons présenter les caractéristiques générales de SOMINE et le plan du mémoire. Ce sera l'objet du chapitre suivant.

#### 4. - CARACTERISTIQUES GENERALES DU SYSTEME ET PLAN DU RAPPORT

Le "cahier des charges" défini à la fin du précédent chapitre nous a guidé au cours de la conception et de l'implémentation de la base de données SOMINE et de ses applications. Il nous a imposé un certain nombre de choix essentiels au niveau de :

- la programmation,
- la configuration générale du système global d'aide à l'enseignement, à la conception et à la gestion,
- la conception et l'implémentation de SOMINE,
- la mise en oeuvre des applications.

Ce sont ces choix généraux que nous allons présenter en renvoyant chaque fois au rapport dans lequel ils sont développés.

##### 4. 1 - LA PROGRAMMATION :

Nous avons voulu réaliser un système indépendant du matériel utilisé. Pour cela, nous avons écrit tous les programmes ( sauf deux sous-programmes de petite dimension) en FORTRAN.

Par souci de souplesse, la programmation a été réalisée de façon modulaire ce qui permet d'ajouter, d'enlever ou de modifier les fonctions de base du système.

##### 4. 2 - LA CONFIGURATION GENERALE DU SYSTEME :

Le système global d'aide à l'enseignement, à la conception et à la gestion est construit autour d'un système de base de données (SOMINE) dont les "caractéristiques" très diversifiées ( numérique entier et décimal ),

#### 4. 3 - VUE D'ENSEMBLE SUR LE TRAVAIL REALISE :

L'implémentation d'un système de gestion de bases de données pose des problèmes dans les domaines de :

- la gestion de la mémoire
- la structuration des informations
- l'interface entre le système et ses utilisateurs.

##### a) gestion des mémoires :

Une banque de données est un ensemble d'objets (informations et relations) reliés entre eux par une structure.

Ces objets ont des caractères dynamiques : leurs places et leurs tailles peuvent changer au cours d'un traitement.

La place physique nécessaire pour "loger" ces objets demande un espace important (de l'ordre de cent millions d'octets dans la plupart des applications de gestion ). Il est donc impossible, dans l'état actuel de la technologie, d'implémenter un système de bases de données sans assurer la répartition des informations sur plusieurs niveaux de mémoire physique. Assurer cette répartition et les passages entre les divers espaces en donnant à l'utilisateur de la banque l'impression qu'il travaille dans un espace UNIQUE, de taille pratiquement ILLIMITEE, constituent les fonctions principales de la gestion de mémoire. La conception et la réalisation de la gestion des mémoires de SOMINE sont exposées dans ce mémoire.

##### b) Structuration des informations :

Les performances d'un système de base de données dépendent largement des divers choix opérés lors de la définition de l'implémentation de la structure formelle des informations.

Les diverses structurations permises par SOMINE et le langage au demeurant extrêmement simple, qui permet leur implémentation seront définis et commentés dans la thèse de P. MARTY.

##### c) SOMINE et ses utilisateurs :

La structure des informations étant définie et implémentée, il faut maintenant "remplir" la banque, puis l'interroger. Cela concerne le

traitement des informations c'est-à-dire, finalement, les relations entre SOMINE et les utilisateurs. L'accès à la banque se fait à travers un langage de requête qui doit être simple et adapté à tous les types d'applications.

La conception et l'implémentation du langage de requête ainsi que les diverses possibilités de traitement des informations que SOMINE peut offrir sont exposées dans la thèse de Cl. SAYETTAT.

D'autre part, puisque nous étions partis d'un "cahier des charges" il était important de vérifier que le système construit possédait les propriétés qui avaient motivées sa conception. Aussi, nous avons développé trois recherches dans des domaines divers afin de mettre en évidence les qualités et les insuffisances de notre système et de montrer comment il était possible de le compléter pour pallier à ces insuffisances.

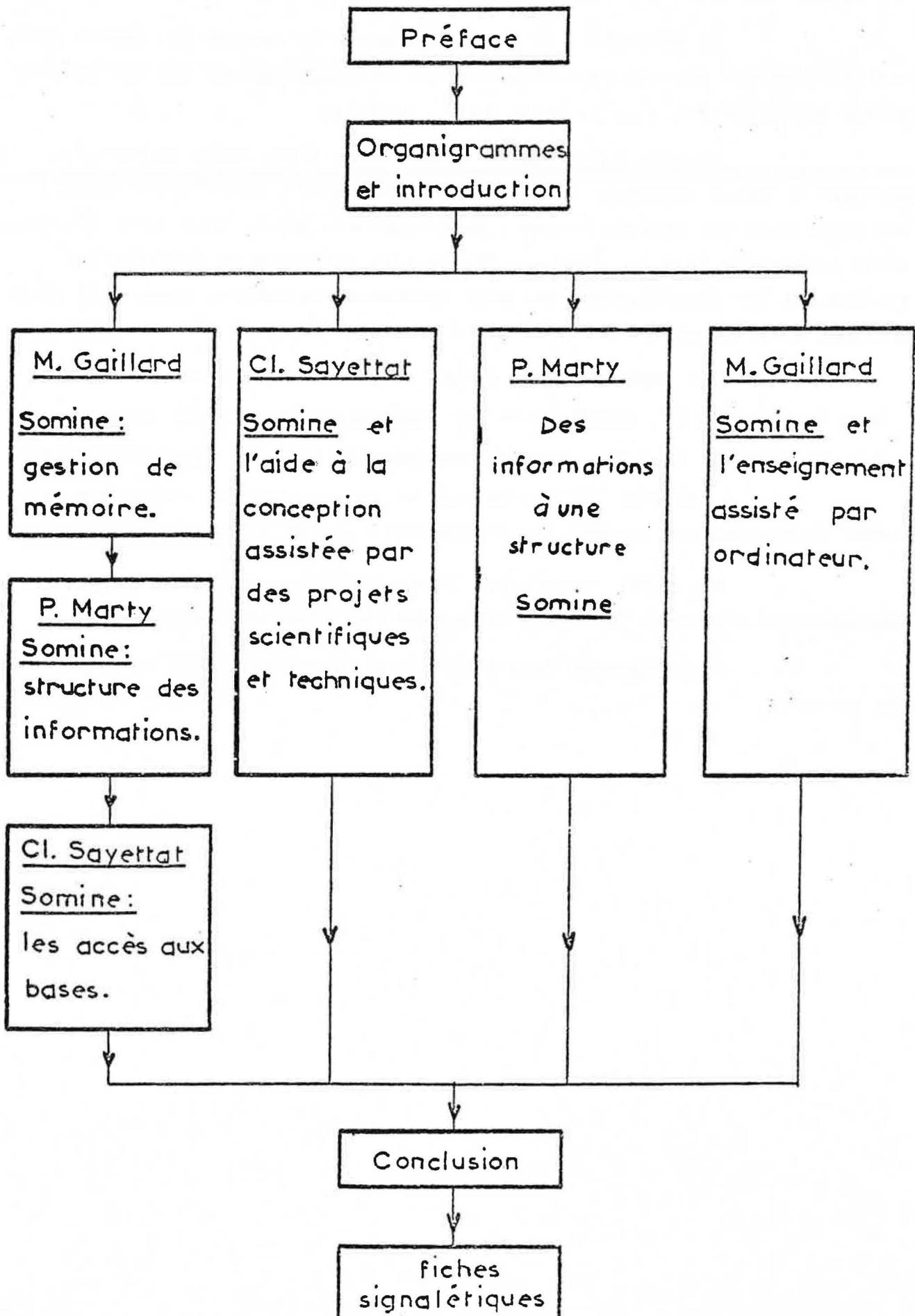
Les comptes-rendus de ces travaux sont exposés :

- dans la thèse de Cl. SAYETTAT pour la Conception Assistée par Ordinateur,
- dans la thèse de P. MARTY pour la Structuration Optimale des informations.

La partie (III) de ce rapport rend compte des recherches réalisées en Enseignement Assisté par Ordinateur à l'aide du système SOMINE.

Ces trois thèses sont le résultat d'un travail d'équipe cependant elles peuvent être lues indépendamment l'une de l'autre.

L'organigramme de l'état actuel du projet SOMINE se trouve à la page suivante.

ETAT ACTUEL DU PROJET SOMINE

4. 4 - CONTENU DU PRESENT RAPPORT :

Dans la suite de ce mémoire en trouvera :

- la gestion des mémoires du système de bases de données SOMINE ( partie II )
- SOMINE et l'enseignement assisté par ordinateur ( partie III )
- une conclusion générale ( partie IV ) dans laquelle nous donnons le bilan actuel des résultats obtenus par notre équipe.

nous espérons donner ainsi à la fois une vue exacte de notre travail et une vue d'ensemble sur le projet dans lequel notre travail est inclu.

Nous sommes à la disposition de toutes les personnes qui désireraient des compléments d'information.



**ANNEXE DE LA PREMIERE PARTIE**

**NOTES BIBLIOGRAPHIQUES**

- 1 - D. KAISER et D. COULON, Institut de programmation de l'Université de PARIS VI : " Contribution de l'informatique à l'évolution des méthodes d'éducation " . Contact C.R.I. numéro 72/27 ( Mai 1973)
- 2 - Cl. DUTRUC, Cl SAYETTAT, P. MARTY et M. GAILLARD, rapport de D.E.A., I.U.T. de Villeurbanne, Université de Lyon : " Cours de FORTRAN en Enseignement Assisté par Ordinateur " ( 1972)
- 3 - J. CL. LATOMBE, Ingénieur I.E.G. et I.M.A.G. , thèse de docteur ingénieur, Université de Grenoble : " Elaboration d'un système pédagogique d'assistance à la conception en électrotechnique " (novembre 1972)
- 4 - I.B.M. : Documentation sur les systèmes de bases de données.
- 5 - J. R. ABRIAL, Université de Grenoble, Institut de mathématiques appliquées : " Projet SOCRATE : spécifications générales " ( Août 70)
- 6 - R.C. SCHANK, Computer Science Dept. and Committee on linguistics -Stanford University - California 94305, USA : " Semantics in conceptual Analysis" Lingua (1972)
- 7 - R.C. SCHANK, N. GOLDMAN, C.J. RIEGER and CH. RIESBECK, Standford, University Standford, California, U.S.A. : " Margie, memory, analysis, response and inference on English " ( 1973)
- 8 - T. WINOGRAD : " Understanding Natural Langage" , Cognitive Psychology, vol. 3 n° 1 P P 1 - 191 (janvier 1972)
- 9 - S.D.C. Santa Monica S.P. 2747 : " T.D.M.S. a new approach to data management " (1967)
- 10 - J. CARBONELL : "Scholar - A new approach to computer Assisted Instruction " Naval Research review ( octobre 1971)
- 11 - J. HEBENSTREIT, Ecole supérieure d'Electricité de Paris : "l'enseignement et les ordinateurs ", 28 ièmes journées d'information de l'Association Suisse pour l'automatique " Genève (avril 1971)

- 12 - A. SARRAZIN, professeur à l'Université de Lyon : "créativité et innovation" (1972)
- 13 - S. AUFIT : "recherche de lois d'agencement et choix automatique des solutions techniques" - fiches du centre national de documentation pédagogique, PARIS ( 1973)
- 14 - L. GEMINARD, Inspecteur Général de l'Enseignement Technique : " Analyse technique et logique d'agencement", Seminaires pour professeur d'E.N.N.A. N° 66 ( novembre 1967)
- 15 - D. FENEUILLE, I.U.T. d'informatique de Villeurbanne, Université de Lyon : " L'enseignement assisté par ordinateur - bibliographie " ( février 1971)
- 16 - J.J. GIRARDOT, Ecole Nationale Supérieure des Mines de St.Etienne : " L'utilisation en Fortran des écrans alphanumériques sous SYSTDS" , information- informatique ( février, mars 1973).



**DEUXIEME PARTIE**

**LA GESTION DES MEMOIRES POUR LE  
SYSTEME DE BASES DE DONNEES SOMINE**



**SOMMAIRE DE LA DEUXIEME PARTIE**  
**(GESTION DES MEMOIRES)**

	<u>PAGE</u>
CHAPITRE 1 : <u>INTRODUCTION</u>	II - 7
1.1 - NECESSITE ET BUTS GENERAUX	II - 7
1.2 - MODE D'ALLOCATION DES ESPACES MEMOIRES	II - 8
1.2.1. - allocations dynamique et statique	
1.2.2. - les critères du choix	
1.2.3. - le choix : le mode statique	
1.3 - L'ENVIRONNEMENT DU PROJET	II - 9
1.3.1. - le matériel	
1.3.2. - un objectif fondamental	
1.3.3. - quelques conséquences	
1.4. - PROBLEMES A RESOUDRE ET PLAN DU RAPPORT	II -10
CHAPITRE 2 : <u>STRUCTURATION DES ESPACES MEMOIRES</u>	II -11
2.1 - DEFINITION ET HYPOTHESES	II -11
2.2 - LA SEGMENTATION	II -11
2.3 - LA PAGINATION	II -12
2.4 - CRITERES DE COMPARAISON	II -13
2.5 - COMPARAISON DE DEUX METHODES	II -13
2.5.1. - critère (a)	
2.5.2. - critère (b)	
2.5.3. - critère (c)	
2.6 - LE CHOIX : LA PAGINATION	II-14
CHAPITRE 3 : <u>L'ESPACE VIRTUEL ET L'ESPACE DISQUE</u>	II -17
3.1 - DEFINITION DE L'ESPACE VIRTUEL	II -17
3.2 - DEFINITION DE L'ESPACE DISQUE	II -17

3.3 - PROBLEME DE L'APPLICATION DE L'ESPACE VIRTUEL SUR L'ESPACE DISQUE	II - 18
3.4 - LES SOLUTIONS	II - 18
3.5 - LE CHOIX : FONCTION (g) PROGRAMMEE	II - 18
3.6 - APERCU SUR LE SYSTEME SOCRATE	II - 19
3.6.1. - Espace virtuel et espace disque	
3.6.2. - Application virtuel $\Rightarrow$ disque	
 CHAPITRE 4 : <u>LA GESTION DE MEMOIRE DANS L'ESPACE REEL</u>	 II - 25
4.1 - OPTIMISATION DES CARACTERISTIQUES	II - 25
4.1.1. - Optimisation de la taille de sous-page	
4.1.2. - Optimisation de la taille de page	
4.2 - ORGANISATION DE LA MEMOIRE CENTRALE	II - 28
4.2.1. - Nombre de pages en mémoire centrale	
4.2.2. - Cadre de manoeuvre	
4.2.3. - Configuration de la mémoire centrale	
4.3 - DICTIONNAIRE DE GESTION DE L'ESPACE REEL	II - 30
4.3.1. - L'espace réel	
4.3.2. - Nécessité d'un dictionnaire	
4.3.3. - Taille d'une information dictionnaire	
4.3.4. - Implémentation du dictionnaire	
4.4 - NULLITE DE L'INFORMATION : l'espace libre	II - 32
4.4.1 - Information nulle	
4.4.2 - Déclaration de sous-page libre	
4.4.3 - Gestion des sous-pages libres	
4.4.4 - Conclusions sur la gestion de l'espace libre	
4.5 - GESTION DE L'ESPACE OCCUPE	II - 35
4.5.1. - Découpage d'une information dictionnaire	

4.5.2. - Taille utilisable de l'espace virtuel	
4.5.3. - L'espace virtuel est généralement quasi vide	
4.6. - PROBLEMES POSES PAR LA GESTION DES SQUATTERS	II - 37
4.6.1. - L'anneau des squatters	
4.6.2. - Modifications dans des anneaux de squatters	
4.7. - LES RELATIONS ENTRE LES ESPACES DE LA MEMOIRE REELLE	II - 40
4.7.1. - Recherche d'une page	
4.7.2. - Copies des informations du cadre de manoeuvre	
4.7.3. - Recherche d'une sous-page libre	
4.8. CONCLUSION	II - 44
CHAPITRE 5 : <u>ASPECTS GENERAUX DU PROJET REALISE</u>	II - 47
5.1. - LE PROBLEME POSE	II - 47
5.2. - LES OPTIONS RETENUES	II - 47
5.2.1. - Les fichiers disques	
5.2.2. - Découpage de l'espace virtuel	
5.2.3. - Possibilité de plusieurs applications	
5.2.4. - Mémoire centrale	
5.2.5. - Relations mémoire centrale → disque	
5.2.6. - Langage de programmation	
5.3. - LES FONCTIONS A REALISER	II - 49
5.4. - CREATION ET PREFORMATAGE DES FICHIERS	II - 49
5.4.1. - Listing du DSKMT	
5.4.2. - Remarque sur les options prises	
5.5. - FORMATAGE ET INITIALISATION DES FICHIERS	II - 51
5.5.1. - Eléments nécessaires	
5.5.2. - Les fonctions de FORMFICH	
5.5.3. - Exemple	

5.6. - OUVERTURE DE FICHIERS	II - 51
5.6.1. - Forme de l'appel	
5.6.2. - Effets de l'appel	
5.7. - FERMETURE DES FICHIERS-SAUVEGARDE DES INFORMATIONS	II - 52
5.7.1. - Forme de l'appel	
5.7.2. - Effets de l'appel	
5.7.3. - Sauvegarde des informations	
5.8. - RELATION ENTRE MEMOIRE CENTRALE ET DISQUE	II - 53
5.9. - LECTURE D'UNE INFORMATION	II - 53
5.9.1. - Fonctions principales de lecture	
5.9.2. - Organigramme de ces fonctions	
5.10 - ECRITURE D'UNE INFORMATION	II - 54
5.10.1 - Fonctions principales de ECRIRE	
5.10.2 - Organigramme de ces fonctions	
5.11 - SCHEMA DE L'ORGANISATION DES SOUS-PROGRAMMES DE LA GESTION DE MEMOIRE	II - 54
CHAPITRE 6 : <u>PREMIERS RESULTATS ET.... NOUVELLES QUESTIONS</u>	II - 57
6.1. - PREMIERS RESULTATS	II - 57
6.1.1. - Un critère d'optimisation	
6.1.2. - Le système face à ce critère	
6.1.3. - Place occupée en mémoire centrale	
6.1.4. - Complexité des algorithmes	
6.1.5. - Risque de saturation de l'espace virtuel	
6.1.6. - Risque de saturation ponctuelle	
6.2. - LES PROBLEMES A RESOUDRE	II - 59
6.3. - UNE IDEE D'APPLICATION	II - 60

ANNEXE 1 : ORGANIGRAMME DES FONCTIONS DES SOUS-PROGRAMMES

II - 61

lecture ( IFI , IX , N , L , TABLE )

ECRIRE ( IFI , IX , N , L , TABLE )

ANNEXE 2 : BIBLIOGRAPHIE

II - 67



## CHAPITRE I : INTRODUCTION

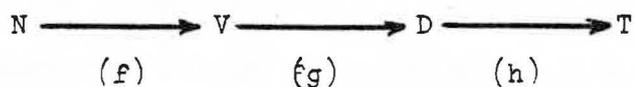
### 1. - NECESSITE ET BUTS GENERAUX :

Une information étant définie comme "un ensemble d'objets élémentaires liés par des fonctions d'accès" /1/ une banque de données peut être considérée comme un moyen de stockage et de recherche rapides de grandes quantités d'informations dynamiques (c'est à dire d'informations dont les lieux et les tailles peuvent changer au cours d'un traitement.)

Informations et relations occupent des places mémoires tellement importantes qu'il est actuellement inconcevable d'implémenter un système de bases de données sans assurer la répartition des informations sur plusieurs niveaux de mémoire physique.

D'autre part, l'utilisateur d'une banque "baptise" ses informations à l'aide de noms (généralement alphanumériques). Il leur affecte des valeurs sous la même forme. Ces noms et ces valeurs sont interprétés et, à chaque information est affectée une adresse unique, qui ne changera pas jusqu'à la mort de cette information /2/ , dans un espace virtuel des données. Cet espace virtuel homogène doit être "appliqué" sur un espace réel assez grand (généralement un espace disque). Enfin, l'exploitation des informations (lecture ou écriture) a lieu en mémoire centrale ( espace tore).

Il est donc possible de résumer les transformations subies par une information de la manière suivante /3/ :



- N est l'ensemble des noms et des valeurs affectés par l'utilisateur à ses informations en utilisant le langage de requête /4/

-V est l'ensemble des adresses virtuelles affectées aux informations par l'interpréteur du langage de requête et à partir de la définition de la structure de l'application concernée /5/.

-D est l'ensemble des adresses disques (espace des lieux de stockage physique des informations).

-T est l'ensemble des adresses tores (mémoire centrale) dans lesquelles sont traitées les informations.

Les définitions de l'ensemble des noms et de la fonction (f) font l'objet des travaux cités en /4/ et /5/ dans la bibliographie.

Les définitions des ensembles V (espace virtuel), D (espace disque) T (espace mémoire centrale) et des fonctions (g) et (h) sont l'objet de cette partie du mémoire.

La première question qui se pose au concepteur d'une gestion de mémoire concerne le mode d'allocation des divers espaces.

## 2. - MODE D'ALLOCATION DES ESPACES MEMOIRE :

### 2. 1. - Allocation dynamique et allocation statique :

Pour représenter les informations, leurs valeurs et leurs relations nous avons besoin de "blocs" de mémoire. Un "bloc" étant un "ensemble contigu d'unités adressables" (généralement des mots).

La naissance, la mort ou la modification d'une information demandent :

- \* soit l'attribution de nouveaux blocs de mémoire qu'il faut choisir parmi les blocs contenant l'information nulle.

- \* soit la restitution de blocs "libérés" à la partie libre de la mémoire.

L'existence d'une ou plusieurs zones de mémoire libre étant supposée, l'allocation de mémoire peut être effectuée de deux façons /1/, /6/ :

- \* d'une manière dynamique : cette méthode s'impose dans tous les cas où la prévision de la taille maximale de mémoire à allouer à une information (données ou programme) est IMPOSSIBLE avant l'exécution.

\* d'une manière statique : cette méthode est utilisable si, et seulement si, on peut prévoir avant l'exécution du programme la taille maximale des informations à traiter. Dans ce cas, le programmeur (ou un programme interne : compilateur, interpréteur, chargeur ...) pourra RESERVER à l'avance les zones mémoires nécessaires. Ces zones n'étant réellement occupées qu'au moment de l'exécution.

## 2. 2. - Les critères du choix :

Une première décision à prendre consiste à choisir un mode d'allocation : statique ou dynamique.

L'allocation statique est plus facile à implémenter que l'allocation dynamique (simplicité des algorithmes). Elle est donc préférable lorsqu'elle est possible. Elle peut conduire à des pertes de places puisque ce sont les tailles maximales des informations (et non les tailles réelles) qui sont prises en compte pour les réservations d'emplacements.

## 2. 3. - Notre choix : mode statique d'allocation :

Pour la base de données SOMINE, nous avons choisi le mode d'allocation STATIQUE. En effet, bien que les informations gérées dans les bases de données aient un caractère dynamique, on peut, au moment de la définition d'une application, prévoir la quantité maximale d'information en taille et en nombre. Il est donc possible de réserver les emplacements nécessaires dans les divers espaces mémoire.

## 3. - L'ENVIRONNEMENT DU PROJET :

### 3. 1. - Le matériel :

La réalisation du système de base de données SOMINE a été menée à bien dans le cadre du CENTRE DE CALCUL de l'ECOLE NATIONALE SUPERIEURE DES MINES DE SAINT-ETIENNE. Le matériel dont nous disposons a été décrit dans /7/. Rappelons brièvement qu'il comprend :

- un ordinateur PHILIPS, type P 1175 sur lequel on peut programmer en autocode, fortran, cobol ...
- cet ordinateur travaille en multiprogrammation. Il est équipé de disques à tête mobile (temps moyen d'accès: 12,5 ms). Son cycle de

base dure 1 micro-seconde.

3. 2. - Rappel d'un objectif fondamental :

Nous voulons réaliser un système indépendant du matériel (transportable) et, pour cela, nous avons choisi de rédiger tous les programmes en langage FORTRAN.

3. 3. - Quelques conséquences :

- a) Puisque nous programmons dans un langage évolué les problèmes de gestion de la mémoire pour NOS PROGRAMMES et pour les PROGRAMMES des utilisateurs du système de bases de données seront pris en charge par le logiciel de l'ordinateur. Seule la gestion de la mémoire pour les INFORMATIONS situées dans le système de banques de données est à notre charge.
- b) Si plusieurs algorithmes peuvent remplir une certaine fonction avec une efficacité comparable, nous nous efforcerons de choisir celui dont la programmation en FORTRAN sera la plus facile.

4. - LES PROBLEMES A RESOUDRE ET LE PLAN DU MEMOIRE :

D'après le schéma de la page 1, nous devons prendre en compte les problèmes suivants :

- a) Structuration des espaces mémoires : espaces virtuels, disque et mémoire centrale. Ce point sera développé dans le chapitre II.
- b) Choix et définition des fonctions de relocation ( g et h ).

\* la fonction (g) concerne l'application de l'espace virtuel sur l'espace disque. Elle fait l'objet du chapitre III où sont rappelés et analysés quelques éléments fondamentaux du système SOCRATE /2/, /8/, /9/ et /10/.

\* la fonction (h) concerne la gestion de mémoire dans l'espace réel (c'est-à-dire les relations entre mémoire centrale et disque). Nous en parlerons dans le chapitre IV dans lequel nous définirons les caractéristiques et les algorithmes principaux de notre projet.

- c) Programmation et implémentation de la gestion de mémoire : le chapitre V donnera un aperçu du travail réalisé.
- d) Expérimentation et résultats, idées de travaux d'amélioration et premières conclusions à tirer de ce travail feront l'objet du chapitre VI.

## CHAPITRE II : STRUCTURATION DES ESPACES MEMOIRES

### 2. 1. - DEFINITION ET HYPOTHESES :

Comme la plupart des auteurs, Peter J. DENNING /11/ distingue deux modes principaux de découpage des espaces mémoires : la segmentation et la pagination. Dans les deux cas, les espaces sont découpés en blocs (ensembles de mots contigus). Dans la segmentation, la taille des blocs est variable. Cette taille est constante dans la pagination.

Etudions maintenant plus en détail ces deux modes de découpage des espaces mémoires avant de les comparer et de choisir celui qui nous paraîtra le plus adapté à la résolution de notre problème.

### 2. 2. - LA SEGMENTATION :

L'espace mémoire est découpé en blocs de tailles variables et, à tout instant (t), il est formé :

- d'un nombre (s) de "segments" (blocs contenant des informations non nulles)
- d'un nombre (h) de "trous" (blocs contenant des informations nulles)

On peut adresser les informations aux moyens de couples (s,m) dans lesquels (s) représente un numéro de segment et (m) l'adresse du mot à l'intérieur du segment considéré. Sur les espaces segmentés, il faut pouvoir effectuer les opérations suivantes :

- Création ou déplacement d'un segment ou d'un trou de taille donnée
- Ajout ou retrait d'un certain nombre de mots à un segment ou à un trou de taille donnée.

La réalisation de ces fonctions conduit à structurer l'ensemble des segments et l'ensemble des trous selon diverses méthodes /1/ /3/ : contiguité, chaînage etc...

Au moment de la création d'un segment, par exemple, il faut rechercher un trou de taille suffisante. Que l'on utilise les méthodes du "BEST FIT" ou du "FIRST FIT", on est de toute façon conduit à déplacer des segments et à compacter des trous où, au contraire, à partager un trou de taille trop grande en deux parties : une pour le segment, l'autre restant dans la partie libre de la mémoire.

Ces problèmes ne sont pas simples : ils sont l'objet d'importantes parties de cours 'd'algorithmique et structures de données' /1/, /3/ ou d'ouvrages spécialisés /6/. Cette méthode, très logique, est donc assez difficile à mettre en oeuvre.

### 2. 3. - LA PAGINATION :

L'espace mémoire est découpé en blocs de taille uniforme (z mots) appelés "pages".

On peut adresser les informations au moyen de couples (p,m) dans lesquels p sont des numéros de page et m l'adresse de mot dans la page considérée.

La gestion de la mémoire est relativement simple car on gère des "blocs" de taille constante et on affecte toujours un nombre entier de pages à une information donnée. Par contre, on est conduit à perdre, en moyenne :

a) une demi-page par information si l'algorithme de placement des informations dans la mémoire aligne toujours le début de l'information et le début d'une page, (figure 1).

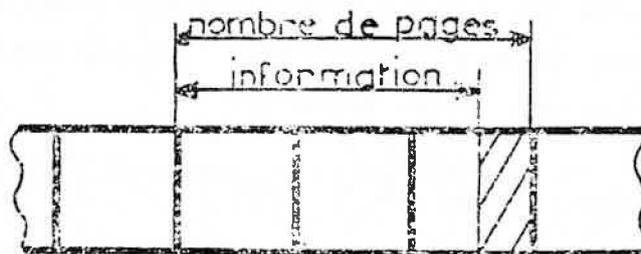


figure 1: une demi-page perdue en moyenne

b) une page par information (dont une demi-page au début et une demi-page à la fin) si l'algorithme de placement des informations ne remplit pas la condition précédente, (figure 2).

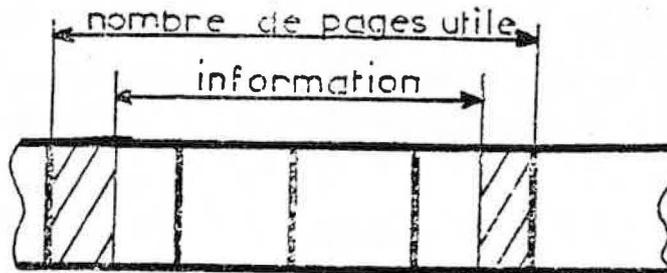


Figure 2: une page perdue  
en moyenne

Au moment de la création d'une information de taille (I) mots, on cherche une place libre dont la taille est suffisante. On attribue aux pages nécessaires des numéros qui serviront à l'adressage des informations qu'elles contiennent.

#### 2. 4. - CRITERES DE COMPARAISON DES DEUX METHODES :

Segmentation et pagination sont utilisées depuis de nombreuses années et ont fait l'objet de comparaisons systématiques à partir des critères suivants /11/ :

- a) Complexité des algorithmes nécessaires pour assurer :
  - \* la gestion des zones libres
  - \* la gestion des zones occupées
- b) Perte de place (inévitable dans les deux méthodes) pour les informations servant à la gestion de mémoire.
- c) Risques de perte de place par fragmentation (création de zones libres irrécupérables).

#### 2. 5. - COMPARAISON DES DEUX MEMOIRES :

##### 2. 5. 1 - Critère (a) :

Il est généralement plus simple de gérer des tableaux de taille fixe que des

éléments de longueurs variables. Ce critère est donc favorable à la pagination.

2. 5. 2 - Critère (b) :

La simplification précédente conduit à des pertes de place moins importantes pour la gestion de la mémoire dans la pagination que dans la segmentation. Ce critère est donc également favorable à la pagination.

2. 5. 3 - Critère (c) :

La fragmentation n'a pas la même origine dans les deux méthodes :

a) En segmentation, elle provient de la présence de "trous" dont les tailles individuelles sont trop faibles pour loger un segment de taille donnée, alors que la somme des tailles serait largement suffisante . On peut démontrer que /11/ :

- soit on essaie de compacter ces trous et le temps perdu dans cette opération est important,

- soit on tolère une certaine perte de place en ne compactant pas (ou en ne compactant que partiellement) les trous . On peut alors chiffrer la perte de place à dix fois la taille moyenne des segments.

b) En pagination, la fragmentation provient du fait que la taille d'une information est toujours "arrondie" à un nombre entier de pages. Nous avons vu (paragraphe III du présent chapitre) que cette perte est fonction de l'algorithme de placement de l'information selon que cet algorithme réalise ou non l'alignement du début de l'information avec un début de page. Ainsi, la "place perdue" sera -t- elle proportionnelle à la taille de la page.

Ce critère n'est donc pas, à priori, favorable à la pagination.

2. 6 - NOTRE CHOIX : LA PAGINATION :

Deux critères sur trois sont favorables à la pagination. Le premier (a) est important pour nous puisqu'il concerne la complexité des algorithmes à programmer.

Nous montrerons, d'autre part, que la perte de place due à la fragmentation peut être réduite sans trop de perte de temps en appliquant les

idées de RANDELL /12/ qui consistent à distinguer l'unité de pagination de l'espace mémoire (la page) et l'unité "de transport" des informations entre les "espaces mémoire" réels ( la grande page).

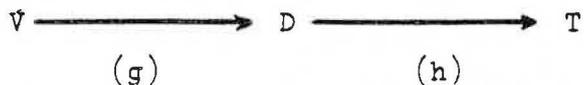
Cette discussion sera reprise plus loin (chapitre IV).  
En attendant, nous choisissons, comme la plupart des constructeurs d'ordinateur et des concepteurs de bases de données, de PAGINER nos "espaces mémoire".



### CHAPITRE III : L'ESPACE VIRTUEL ET L'ESPACE DISQUE

#### 3. 1. - DEFINITION DE L'ESPACE VIRTUEL :

Reprenons le schéma des transformations subies par une information depuis son "baptême" dans l'espace virtuel jusqu'à son exploitation en mémoire centrale :



L'espace virtuel d'une base de données doit avoir les qualités suivantes : /2/

- " 1°) l'adresse virtuelle d'une donnée y est unique pendant toute la vie de la donnée
- 2°) Il existe un moyen permettant de savoir si une adresse correspond à une donnée (c'est à dire, si une adresse est vide ou occupée)
- 3°) On doit toujours pouvoir choisir un nom pour une donnée."

Ces propriétés conduisent à rechercher un ESPACE VIRTUEL TRES GRAND afin que toutes les informations relatives à une application quelconque puissent y trouver une adresse.

#### 3. 2. - DEFINITION DE L'ESPACE DISQUE :

Il doit recevoir les informations créées dans l'espace virtuel dont les valeurs sont différentes de l'information nulle.

Sa taille, bien qu'assez grande, sera toujours limitée et généralement inférieure à la taille de l'espace virtuel.

### 3. 3. - LE PROBLEME DE L'APPLICATION DE L'ESPACE VIRTUEL SUR L'ESPACE DISQUE :

On peut le résumer ainsi : " Trouver une fonction (g) réalisant l'application d'un espace de taille très grande (l'espace virtuel) mais, dans la plupart des applications, QUASI VIDE, sur un espace disque de taille beaucoup plus petite mais qui ne doit recevoir que les informations de valeurs non nulles."

### 3. 4. - LES SOLUTIONS POSSIBLES :

Elles sont très nombreuses et font l'objet de développements importants dans les cours de structures de données /3/, /1/ ou dans des ouvrages spécialisés /6/. On peut les regrouper, le plus souvent, dans deux catégories :

a) Solutions dans lesquelles la LIAISON ENTRE LES DEUX ESPACES EST ENREGISTREE (tables des segments) : une série d'indexations permet alors de passer d'un espace à l'autre /13/, /14/.

b) Solutions dans lesquelles la liaison est PROGRAMMEE : un programme assure la détermination d'une adresse dans un des espaces quand on connaît l'adresse dans l'autre espace.

La première solution a notamment été utilisée dans la gestion de mémoire du système "MULTICS" ( Multiplexed Information and Computing Services) /13/ dans lequel le passage de l'espace virtuel (taille  $2^{32}$  mots de 36 bits) à l'espace disque utilise trois tables : la table des travaux en cours, la table des segments et la table des pages.

La deuxième solution est utilisée dans le système SOCRATE /8/ et sera décrite ci-dessous.

### 3. 5. - NOTRE CHOIX : FONCTION (g) ASSUREE PAR DES PROGRAMMES :

Les systèmes d'allocation dynamique de mémoire /13/, /14/ utilisent généralement des fonctions de relocation (g) faisant essentiellement appel à des gestions de tables.

Au contraire, dans les systèmes utilisant l'allocation statique, on se sert généralement de fonctions de relocation (g) programmées qui, par des

calculs, permettent le passage d'un espace à l'autre.

C'est cette deuxième méthode que nous choisissons. Nous nous inspirerons du mode d'application utilisé dans SOCRATE dont nous rappelons ci-après les aspects essentiels.

## 1. 6. - DESCRIPTION DE LA SOLUTION : APERCU SUR LE SYSTEME SOCRATE :

### 3. 6. 1. - L'espace virtuel et l'espace disque :

Les informations sont adressées STATIQUEMENT dans un espace virtuel dont la taille approximative est de  $2^{31}$  mots de 32 bits. Chaque information a ainsi une adresse unique, définie une fois pour toute en fonction de la structuration des informations adoptées par l'utilisateur pour une application donnée.

Cet espace virtuel a ainsi les qualités définies au paragraphe 3.1 du présent chapitre. Sa taille, notamment, est tellement grande qu'il sera généralement un espace quasi vide.

La taille de l'espace disque sera choisie par l'utilisateur en fonction du nombre et de la taille des informations qui devront y prendre place effectivement (informations non nulles). Cette taille sera donc différente d'une application à l'autre.

Dans les deux espaces :

- l'unité d'information adressable sera le BIT
- les BITS seront regroupés dans des mots de 32 bits (choix dicté par le matériel utilisé)
- les mots seront regroupés en "petites pages" ( que nous appellerons à partir de maintenant des SOUS-PAGES). La sous-page étant l'UNITE DE PASSAGE D'UN ESPACE A L'AUTRE et jouant donc le rôle de ce que nous appelions jusqu'ici des pages. On peut dire aussi que les espaces virtuels et disque sont "paginés" en SOUS-PAGES.

Selon la taille de la sous-page, le nombre de sous-pages est variable. Afin de simplifier les schémas, nous fixerons le nombre de sous-pages à  $2^{28}$  (sous-pages de taille égale à  $2^3$ ), par exemple. On peut alors schématiser les deux espaces comme sur la figure 3.

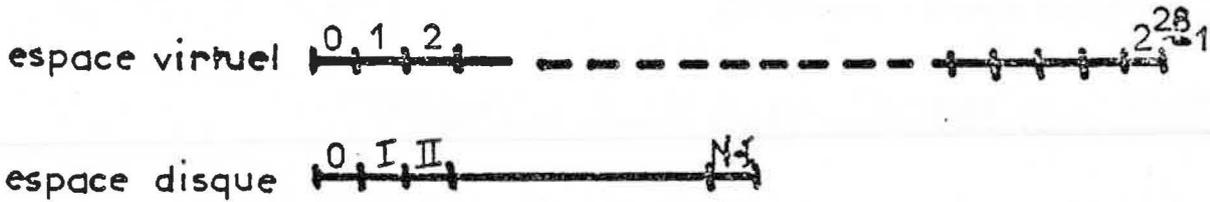


figure 3 : espaces mémoire

3.6. 2.--L'application de l'espace virtuel sur l'espace disque

a) On fractionne l'espace virtuel en plis : le nombre de sous-pages de chaque pli est égal au nombre ( $N$ ) des sous-pages de l'espace réel. Ces plis (figure 4) sont placés "les uns sur les autres" et "au-dessus de l'espace réel".

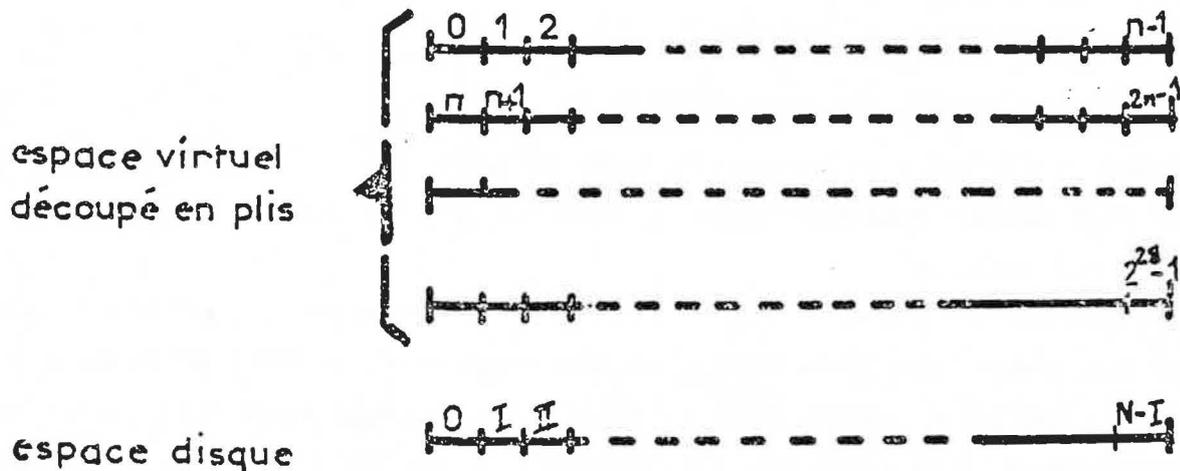


figure 4 : les plis de l'espace virtuel

b) Les sous-pages de l'espace virtuel CONTENANT DES INFORMATIONS sont projetées VERTICALEMENT sur l'espace disque. Par exemple, si la sous-page d'adresse

virtuelle  $n + 3$  contient de l'information, cette information sera écrite dans la sous-page III de l'espace réel.

L'informationn contenue dans la sous-page virtuelle  $2^{28} - 1$  sera écrite dans la sous-page réelle  $N - I \dots$  etc...

La correspondance entre les adresses virtuelles et réelles d'une sous-page est ainsi réalisée d'une manière très simple par le "H-code consistant à prendre l'adresse virtuelle MODULO la taille de l'espace réel".

Exemple :  $(n + 3) \text{ modulo } N = 3 \quad /10/ \quad (\text{figure } 5)$ .

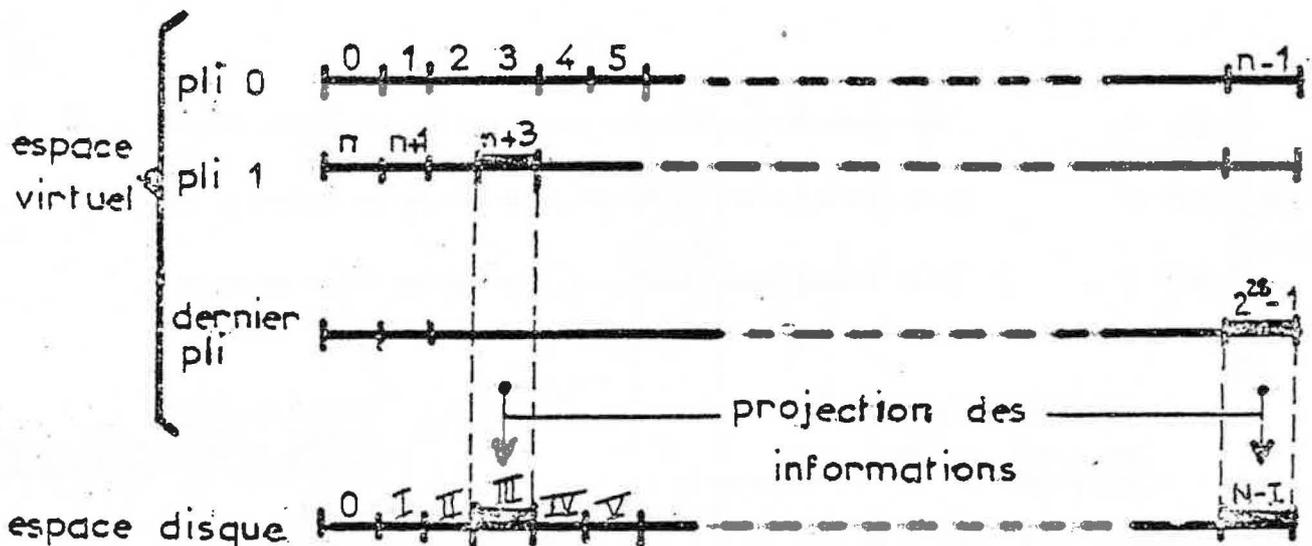


figure 5 : de l'espace virtuel à l'espace disque

c) La principale difficulté de la méthode vient du fait que des informations ayant des adresses virtuelles différentes auront la même adresse réelle. On dit qu'il y a "collision" entre des informations provenant de plis différents de l'espace virtuel et qui se projette "verticalement" sur une même sous-page de l'espace réel.

Pour éviter les altérations d'informations qui ne manqueraient pas de se produire en cas de collision, on crée des "EXCEPTIONS A LA REGLE DE PROJECTION VERTICALE" qui, dans la mémoire réelle, vont se traduire par la présence de "SOUS-PAGES DEPLACEES" appelées "SQUATTERS".

Prenons un exemple :

On vient de projeter la sous-page virtuelle  $(n + 3)$  dans la sous-page (III) de l'espace disque. On veut maintenant écrire les informations contenues dans la sous-page  $(2.n + 3)$  de l'espace virtuel. La sous-page (III) étant occupée, la projection verticale "normale" est impossible. On cherche une sous-page libre. On la trouve, par exemple, en (VII). On y écrit les informations venant de  $(2n + 3)$ . Elles occupent ainsi une place qui ne leur appartient pas. La sous-page située en (VII) est une sous-page SQUATTER . (figure 6) .

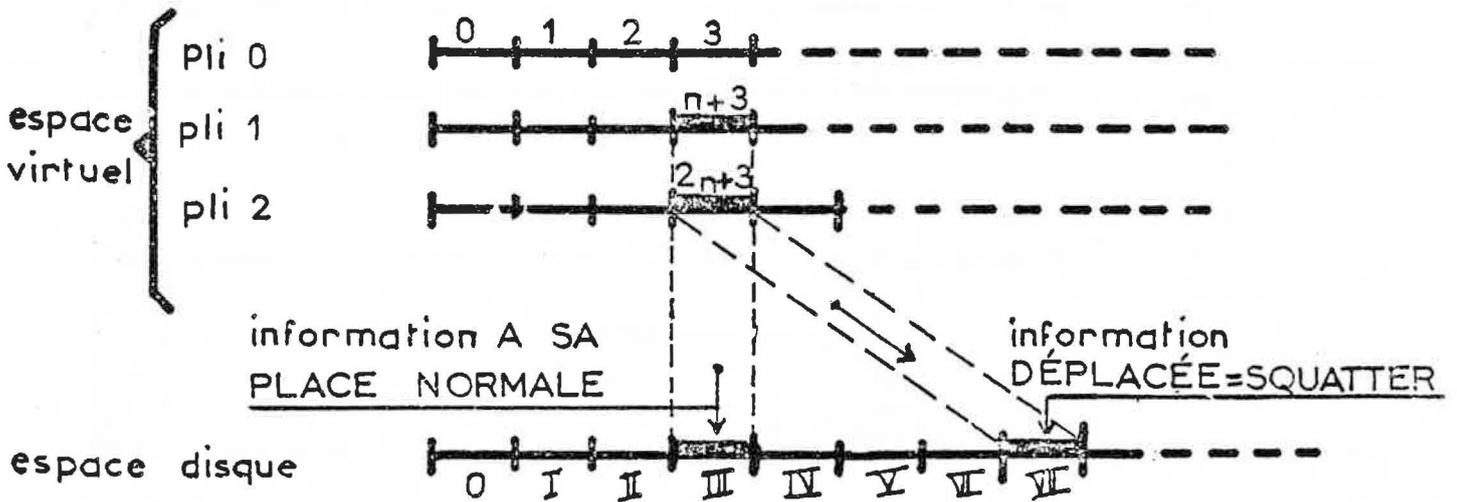


figure 6 : naissance d'un squatter

Le problème des sous-pages faisant collision est résolu en créant une table des sous-pages (taille de la table : un mot par sous-page de l'espace disque au moins) dans laquelle les fonctions suivantes sont réalisées :

- distinction entre sous-page "à sa place" et sous page squatter,
- chaînage en anneau des sous-pages faisant collision
- chaînage des sous-pages libres.

Remarquons que cette méthode d'application de l'espace virtuel sur l'espace disque ne permet pas l'alignement du début d'une information avec le début d'une sous-page. On est donc dans le cas le plus défavorable pour les risques de perte de place par fragmentation. Cela pose le problème de la taille de sous-page et, plus généralement, le problème du choix des caractéristiques principales du système. Nous traiterons ces questions dans le chapitre suivant.



## CHAPITRE IV : LA GESTION DE MEMOIRE DANS L'ESPACE REEL

L'application de l'espace virtuel sur l'espace disque a situé les informations dans un premier lieu physique. Il faut maintenant étudier le problème de leur passage dans l'espace tore (mémoire centrale) où les lectures et les écritures auront effectivement lieu.

Nous connaissons l'adresse dans l'espace disque d'une sous-page virtuelle contenant une information donnée. Cette adresse est obtenue à partir du nom de l'information (sa "clé") par des calculs et , éventuellement une recherche de sa position dans un anneau de squatters (utilisation d'une table). On peut donc amener la sous-page en mémoire centrale. Mais une question se pose : faut-il transférer UNIQUEMENT cette sous-page ? Cette question peut être reformulée ainsi : étant donné qu'il faut  $10^4$  fois plus de temps pour rechercher une information sur un disque à tête mobile que dans une mémoire tore, ne vaut-il pas mieux transférer, D'UN SEUL COUP, une grande quantité d'information ? La réponse à cette dernière question sera sans doute positive mais on retrouve ici l'éternel problème de "l'espace et du temps " : si on veut perdre moins de temps (en limitant le nombre des accès disques), il faut "perdre" de la place en mémoire centrale.

La solution de ce problème sera recherchée dans les paragraphes suivants à partir de :

- l'optimisation des tailles (sous-pages, pages, espace disque)
- l'organisation de la mémoire centrale
- la définition et l'implémentation d'un dictionnaire de gestion de mémoire
- la gestion de l'espace libre
- la gestion de l'espace occupé et la mise en place d'un mode de gestion des squatters

- l'étude et l'implémentation des algorithmes assurant le transfert des informations entre les deux espaces de la mémoire réelle.

Ainsi, au fur et à mesure, les éléments principaux de notre projet seront définis, discutés et choisis.

#### 4. 1. - OPTIMISATION DES CARACTERISTIQUES :

##### 4. 1. 1- Optimisation de la taille de la sous-page :

Nous reprenons ici le problème de la fragmentation de la mémoire provenant du principe même de la méthode de pagination des espaces. Nous savons que l'algorithme retenu est défavorable car il ne permet pas l'alignement des débuts des informations avec des débuts de sous-pages. La place perdue par fragmentation est alors directement proportionnelle à la taille de la sous-page. Cette taille doit donc être petite.

Mais d'autre part, la gestion des squatters conduit à gérer une table dont la taille ( en mots) est égale au nombre de sous-pages de l'espace disque : on perd un nombre de mots égal au nombre de sous-pages. Ce point de vue va évidemment dans le sens inverse du précédent. Une taille OPTIMALE semble possible.

En effet, à partir de l'étude très complète d'ERIC WOLMAN /15/ ABRIAL et son équipe /2/ ont démontré que :

\* " des sous-pages de 16 mots sont optimales à 95 % pour des informations de tailles moyennes comprises entre 64 et 16 384 mots. "

\* " des sous-pages de 8 mots sont optimales à 95 % pour des informations dont la taille moyenne est comprise entre 16 et 1024 mots. "

Ces résultats ont deux conséquences :

- Il faut laisser à l'utilisateur le soin de fixer la taille de sous-page de son application,

- La sous-page, unité de passage des informations de l'espace virtuel dans l'espace disque ne peut pas être l'unité de transfert entre les espaces "disques" et "tores" car sa taille est trop faible : la multiplication des accès disques conduirait à des temps de réponse trop importants.



sous-pages de 16 mots (optimales) à des pages de 512 mots.

- Si on tolère un certain pourcentage d'exceptions à la règle précédente et si on admet simultanément plusieurs pages en mémoire centrale, la taille précédente peut être notablement réduite.

A partir de ces considérations, nous avons choisi une taille de page de un K octets environ (257 mots exactement car nous verrons qu'il faut un mot en-tête de page pour gérer les sous-pages libres).

#### 4. 2. - ORGANISATION DE LA MEMOIRE CENTRALE :

##### 4. 2. 1. - Nombre de pages en mémoire centrale :

L'appel à des informations sur le fichier structure est plus fréquent que l'appel des informations sur le fichier données. En effet, il faut souvent parcourir plusieurs branches de la structure arborescente pour déterminer tous les éléments nécessaires au calcul d'une adresse du fichier données. Heureusement la quantité d'information est faible dans le fichier structure (deux ou trois pages en général) /5/. On peut donc essayer de garder la majeure partie de ce fichier en mémoire centrale.

Enfin, pour assurer la rapidité de l'accès aux données, il serait souhaitable de conserver en mémoire centrale une ou deux pages du fichier données.

Ces conditions paraissent pouvoir être remplies si on admet simultanément QUATRE PAGES AU TOTAL ( appartenant à l'un ou l'autre des fichiers) en mémoire centrale.

Naturellement, le choix des algorithmes de gestion des relations entre mémoire centrale et disque aura un rôle également déterminant. Ces choix seront expliqués ultérieurement.

##### 4. 2. 2. - Cadre de manoeuvre :

On peut se poser la question de l'utilité d'un tableau, apte à recevoir une page et dans lequel aurait lieu toutes les opérations de lecture et d'écriture. Un tel tableau est souvent appelé "cadre de manoeuvre".

Les inconvénients du cadre de manoeuvre sont :

- utilisation d'une place supplémentaire (de un K octets) en mémoire centrale
- obligation de recopies fréquentes de pages en mémoire centrale car une écriture, par exemple, conduit aux opérations suivantes : copie d'une page sur le cadre de manoeuvre, écriture dans cette page, recopie de la page "à sa place en mémoire centrale" c'est - à - dire sur un des quatre "cadres" définis ci-dessus.

Les avantages du cadre de manoeuvre sont :

- simplification des algorithmes de lecture et d'écriture (ces opérations s'effectuent toujours à la même place en mémoire centrale)
- simplification des algorithmes d'appel de page (toutes les pages venant du disque doivent aller à la même place en mémoire centrale.

En raison de l'importance que présente pour nous le critère "simplicité des algorithmes", nous avons décidé d'utiliser un cadre de manoeuvre.

4. 2. 3. - Configuration de la mémoire centrale :

La configuration de la mémoire centrale réservée aux informations est alors conforme à la figure 7 :

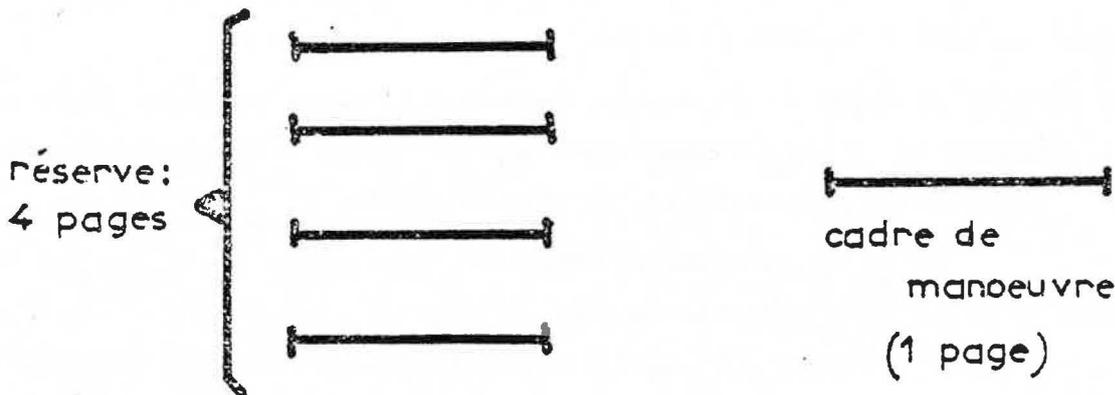


Figure 7 : pages en mémoire centrale

4. 3. - LE DICTIONNAIRE DE GESTION DE L'ESPACE REEL :

4. 3. 1. - L'espace réel :

Nous désignerons ainsi l'ensemble des espaces "mémoire centrale" et "disque". Ces deux espaces sont maintenant totalement définis : Ils sont découpés en page de 257 mots. Chaque PAGE est composée :

- d'un mot d'en-tête dont la nécessité sera démontrée ci-dessous (gestion de l'espace libre)
- d'un nombre entier de sous-pages, servant d'unité de transfert des informations entre l'espace virtuel et l'espace disque. Ce nombre étant choisi par l'utilisateur et généralement égal à 32 ( sous-pages de 8 mots) ou à 16 (sous-pages de 16 mots).

4. 3.2. - Nécessité d'un "dictionnaire"

Les paragraphes précédents, en présentant la solution adoptée dans la gestion de mémoire par méthode statique, ont mis en évidence trois sortes de sous-pages :

- les sous-pages libres
- les sous-pages contenant des informations et situées à la place normale obtenue par "projection verticale" de l'espace virtuel sur l'espace réel.
- les sous-pages contenant des informations et situées à des places différentes de celles que la projection verticale des informations aurait du leur attribuer (squatter).

Ces différents types de sous-pages imposent la création d'une table de gestion des sous-pages, le plus souvent appelée "dictionnaire de gestion de mémoire " /8/, /9/. Dans ce dictionnaire, il faudra prévoir :

- la distinction entre les sous-pages libres et les sous-pages occupées et la recherche des sous-pages libres
- Parmi les sous-pages occupées, il faudra être capable de :
  - \* faire la distinction entre les sous-pages situées "à la bonne place" et les sous-pages "squatters".
  - \* Entreprendre la recherche de l'information c'est-à-dire savoir rapidement, à partir d'une sous-page

" à sa place ", où sont ses squatters, si ils existent.

Ainsi apparait la nécessité d'une "TABLE DE GESTION DE MEMOIRE" ayant autant d'entrées qu'il y a de sous-pages dans l'espace disque. Cette table est souvent appelée "dictionnaire de gestion de mémoire". Essayons de définir ses caractéristiques principales.

#### 4. 3. 3. - Tailles d'une information du dictionnaire :

Pour la fixer de façon approximative, on peut dire qu'il faut :

- UN bit pour distinguer une sous-page libre d'une sous-page occupée
- UN bit pour distinguer une sous-page "à sa place" d'une sous-page "squatter".

Il reste à définir l'adresse de la sous-page.

Il est bien évident que nous avons intérêt à avoir un espace virtuel le plus grand possible car, notre allocation de mémoire étant statique, on risque de se trouver bloqué dès le départ si l'espace disponible n'est pas assez grand. Nous avons le choix entre deux solutions :

- a) Un élément du dictionnaire codé sur UN mot

Dans ce cas, 30 bits sont disponibles pour l'adressage de la sous-page. L'espace virtuel est alors de  $2^{30}$  sous-pages de 8 à 16 mots (tailles les plus généralement adoptées). Cet espace est déjà très confortable. Cependant, pour quelques applications particulièrement exigeantes en places, il risque d'être saturé.

- b) Un élément du dictionnaire codé sur DEUX mots.

L'espace précédent est maintenant multiplié par  $2^{32}$ . Il n'y a plus de problèmes de place. Toutefois, la gestion du dictionnaire est plus complexe et son volume est doublé (nous verrons que la place prise par le dictionnaire est loin d'être négligeable).

Dans un souci de simplicité et en prenant le risque de ne pas pouvoir apporter de solution à des cas extrêmes, nous avons choisi de CODER UN ELEMENT DU DICTIONNAIRE SUR UN MOT.

4. 3. 4. - Implémentation du dictionnaire :

La taille du dictionnaire que nous venons de définir est souvent importante. Par exemple, pour une application utilisant 1000 pages sur le disque, découpées chacune en 32 sous-pages de 8 mots, on utilisera un dictionnaire de 32 000 mots !

Donc, malgré l'importance du critère "temps d'accès aux informations" qui serait optimisé par la présence de tout le dictionnaire dans la mémoire centrale, l'adoption de cette solution est impossible. Nous avons choisi de REPARTIR LE DICTIONNAIRE dans l'espace disque en regroupant toutes les informations relatives à une page en-tête de cette page.

4. 4. - NULLITE DE L'INFORMATION : SOUS - PAGE LIBRE :4. 4. 1. - Information nulle :

Au sens propre du terme, l'information nulle n'existe pas. Il faut donc CHOISIR une information et décider qu'elle SERA l'information NULLE. Au départ, nous avons choisi pour information nulle la valeur ZERO aussi bien pour les mots du dictionnaire de gestion de mémoire que pour les mots "utilisateurs".

Ce choix a été conservé pour la partie dictionnaire mais, pour les mots "utilisateurs", il a semblé préférable d'adopter la valeur (-1) qui faciliterait la gestion des valeurs numériques entières et décimales /4/ et /5/. Ce changement n'a posé aucun problème et, s'il fallait un jour remettre en cause le choix de l'information nulle pour la partie utilisateur, cela conduirait à des modifications mineures.

En résumé, l'état actuel est le suivant :

- pour les mots du dictionnaire de gestion de mémoire totalement "transparent" aux utilisateurs, l'information nulle égale ZERO
- pour les mots "utilisateurs", l'information nulle est MOINS UN. Donc, tous les mots utilisables ont leurs bits positionnés à UN au début d'une application ( initialisation).

4. 4. 2. - Déclaration de sous-page LIBRE :

Il est parfaitement possible d'annuler l'information que contenait une sous-page par une ECRITURE puisqu'il suffit de placer la valeur "nulle" (-1) dans

chacun de ses mots. Après chaque ECRITURE, il faut donc tester le CONTENU de la sous-page. Ce test peut être effectué de deux manières :

a) Sur un compteur, placé dans le mot du dictionnaire correspondant à chaque sous-page. Cette méthode est décrite en /8/ et /9/. Pour chaque mot de valeur non nulle écrit dans la sous-page, ce compteur est incrémenté. Si un mot contient une information nulle, il n'est pas compté... Pour décider qu'une sous-page est "vide", un seul test de compteur est nécessaire. Mais la gestion du compteur prend du temps. En outre, un certain nombre de bits est utilisé par cette gestion ce qui diminue la taille de l'espace virtuel.

b) Sur l'ensemble de la sous-page : après une écriture, on teste la valeur de chaque mot. Si un mot est trouvé avec une valeur différente de moins un, la sous-page est déclarée occupée. Si tous les mots ont pour valeur moins un, elle est libre. Cela implique, en moyenne, l'examen de la moitié des mots de la sous-page.

c) Conclusion : La taille de l'espace virtuel devant être la plus grande possible, il nous a semblé préférable de choisir la deuxième méthode qui est favorable à ce critère.

#### 4. 4. 3. - Gestion des sous-pages libres :

Le but de cette gestion est le suivant : Indiquer le plus rapidement possible si une sous-page est libre et la position de cette sous-page dans la page. Pour cela, on peut imaginer de parcourir séquentiellement le dictionnaire d'une page et, testant chacun de ses mots, trouver les positions des sous-pages libres, comparer cette position à celle qui est demandée.... etc... Mais on peut trouver des méthodes plus rapides :

a) En tête de pages, en plus des mots constituant le dictionnaire de l'état des sous-pages, on réserve un mot pour indiquer la place de LA PREMIERE SOUS-PAGE VIDE. Pointant sur le mot du dictionnaire qui correspond à cette sous-page, on y trouvera soit l'adresse du mot d'en-tête de page (auquel cas, la sous-page libre trouvée est la seule de son espèce), soit l'adresse d'une autre sous-page libre ... etc. Si la page est pleine, le mot d'en-tête de page prend la valeur zéro. Ce mode de gestion par chaînage des sous-pages libres a été décrit en /8/, /9/.

b) Dans la recherche des informations, on utilise souvent des chaînes de bits de présence. Nous avons pensé que cette idée était intéressante

pour la gestion des sous-pages libres. Dans cette méthode, le mot d'en-tête de page existe aussi mais il est considéré comme la chaîne de bits d'existence des sous-pages libres dont il donne directement les positions. Si une sous-page est libre, le bit correspondant à son numéro d'ordre dans la page est positionné à zéro. IL est positionné à un si la sous-page est occupée.

exemple :

Si la page comprend 8 sous-pages ; la chaîne de bits 10110001 indique que les sous-pages 0, 2, 3 et 7 sont occupées tandis que les sous-pages 1, 4, 5 et 6 sont libres.

Dans la deuxième méthode, la recherche d'une sous-page libre est une suite de tests sur la valeur d'un bit. Si on veut savoir si une sous-page (dont la position est connue) est "occupée" ou "libre", un seul test suffit.

Dans la première méthode, la recherche d'une sous-page libre de position quelconque est rapide. Mais le test d'occupation oblige à parcourir l'anneau des sous-pages libres. Enfin, il est plus rapide de positionner un bit pour dire qu'une sous-page vient de recevoir de l'information que de refaire le chaînage des sous-pages libres.

Les raisons précédentes nous ont amené à choisir la seconde méthode qui se révèle fiable, rapide et dont l'implémentation sur le matériel dont nous disposions ( P 1175 - PHILIPS) /7/ était réalisable sans trop de difficulté.

4. 4. 4. - Conclusions sur la gestion de "L'ESPACE LIBRE" :

Au chaînage des sous-pages libres et à la gestion des compteurs d'occupation des mots utilisés dans le projet SOCRATE, nous avons substitué :

- des tests systématiques des mots de la sous-page après chaque écriture.
- une gestion par chaîne de bits des sous-pages libres qui utilise UNIQUEMENT le mot d'en-tête de page.

Outre, le fait qu'elles sont plus rapides pour la plupart des opérations de gestion de l'espace libre, ces méthodes offrent l'intérêt de séparer radicalement les gestions des espaces libres et occupés. Le bit qui permettait, dans une information dictionnaire, de distinguer entre les sous-pages des deux-espaces peut donc être récupéré et la taille de l'espace virtuel se trouve agrandie (multipliée par deux). La gestion du dictionnaire est plus simple puisqu'il ne contient plus aucune information relative à l'espace occupé.

intéressons-nous maintenant à cet espace.

4. 5. - GESTION DE L'ESPACE OCCUPE :

4. 5. 1. - Découpage d'un élément du dictionnaire :

Les informations relatives à une sous-page OCCUPEE sont codées sur un mot de 32 bits appelé " mot d'en-tête de sous-page". Ces mots sont regroupés au début de chaque page et sont utilisés ainsi :

- les sous-pages occupées situées "à la bonne place" sont distinguées des sous-pages "squatters" par la valeur du premier bit du mot d'en-tête de sous-page. Si la valeur de ce bit est zéro, la sous-page est à sa place. S'il est positionné à un, il s'agit d'un squatter.

- Il reste alors 31 bits pour :

- . Ecrire la partie de l'adresse virtuelle qui est conservée, c'est-à-dire le numéro du pli de l'espace virtuel d'où provient l'information.
- . Chaîner en anneau la sous-page "à sa place" et l'ensemble de ses squatters. Ce chaînage comprend deux parties : le numéro de la page dans laquelle se trouve l'élément suivant de l'anneau et le numéro de la sous-page qu'il occupe dans cette page.

Les mots d'en-tête de sous-page sont donc découpés conformément au schéma de la figure 8 :

SOUAT	Numéro du pli de l'espace virtuel auquel appartient la sous-page réelle	Numéro de la page où se trouve le squatter éventuel	Numéro de la sous-page où se trouve le squatter éventuel
1 bit	n1 bits	n2 bits	n3 bits

Figure 8 : mot d'en-tête de sous-page

Si une sous-page n'a pas de squatter, elle est chaînée sur elle-même. Si on admet qu'un squatter peut se trouver "sur toute page de la mémoire disque" (ce qui est peu probable d'après la discussion du paragraphe 4.1.2. mais permet d'éviter tous les risques de saturation "ponctuelle"), n2 n'est fonction que de la taille de l'espace disque :

$$\text{Taille de l'espace disque} \leq 2^{n2} \text{ pages.}$$

n3 doit permettre d'exprimer le nombre de sous-pages dans la page :

$$\text{nombre de sous-pages} = 2^{n3}$$

Les nombres de bits n1, n2 et n3 seront donc variables en fonction des choix de l'utilisateur (nombres de sous-pages par page et nombre de pages de la mémoire disque).

4. 5. 2. - Taille de la partie "utilisable" de l'espace virtuel :

La taille de l'espace virtuel est en définitive fonction de ces choix et, l'espace REELLEMENT à la disposition de l'utilisateur, compte tenu de la place prise par la gestion de mémoire (1 mot par sous-page) est donnée par le tableau de la figure 9 pour quelques exemples :

CHOIX DE L'ESPACE UTILISABLE		Nombre de bits nécessaires au chaînage des pages NPBIT = n2	Nombre de bits nécessaires au chaînage des sous-pages NSPBIT = n3	Nombre maxi de plis INDMA	Taille de la mémoire virtuelle
NP	NSP				
128	32 de 7 mots	7	5	2 <sup>19</sup> mots	2 <sup>31</sup> x 7
64	32 de 7 mots	6	5	2 <sup>20</sup> mots	2 <sup>31</sup> x 7
1024	16 de 15 mots	10	4	2 <sup>17</sup> mots	2 <sup>31</sup> x 15

Figure 9 : taille utilisable de l'espace virtuel

4. 5. 3. - Remarque : la mémoire virtuelle est bien un espace quasi-vide :

Les tailles énormes trouvées ci-dessus montrent bien que les risques de saturation de l'espace virtuel sont faibles. Naturellement, cette éventualité PEUT se produire puisque la taille de cet espace n'est pas INFINIE. Mais, pour prendre un cas concret d'étude d'une application importante, THOMAS/16/ a montré que pour assurer la gestion des résultats d'un questionnaire concernant dix mille entreprises françaises, il faut au maximum dix millions d'octets (ou 2,5 10<sup>6</sup> mots).... On est donc encore loin de la saturation !

Par contre, lors du choix de la TAILLE DE L'ESPACE DISQUE, il est important de prendre garde aux risques de résonance entre la taille des informations et la taille des plis. Il peut arriver que l'on crée de nombreux squatters ( si la taille du pli est exactement un multiple de la taille d'une entité ! ). La règle suivante doit être observée : la taille de l'espace utilisable de la mémoire disque ne doit pas être multiple de la taille d'une entité.

4. 6. - PROBLEMES POSES PAR LA GESTION DES SQUATTERS :

Terminons l'étude de la gestion de l'espace occupé en décrivant de manière plus précise les problèmes que nous a posé la présence des squatters et les solutions que nous leur avons apporté :

4. 6. 1. - L'anneau des squatters :

Nous le décrivons à travers les schémas suivants. Le cas où il n'y a pas de squatter est représenté à la figure 10. La figure 11 montre le cas d'un anneau contenant un seul squatter. Sur la figure 12 l'anneau contient plusieurs squatters.

	Numéro du pli virtuel	Emplacement réel		Mot d'en-tête de sous-page correspondant			
		page	sous-page				
Sous-page "à sa place"	1046	11	8	0	1046	11	8

Figure 10: sous-page "à sa place" chaînée sur elle-même

	Numéro du pli virtuel	Emplacement réel		Mot d'en-tête de sous-page correspondant			
		page	sous-page				
Sous-page "à sa place"	0	5	17	0	0	7	22
----- squatter	4	7	22	1	4	5	17

Figure 11: un seul squatter

	Numéro du pli virtuel	Emplacement réel		Mot d'en-tête de sous-page correspondant			
		page	sous-page				
Sous page "à sa place"	17	28	7	0	17	3	1
----- squatter	0	3	1	1	0	35	6
----- squatter	8	35	6	1	8	28	4
----- squatter	2147	28	4	1	2147	28	7

Figure 12: cas de plusieurs squatters

Sur ces exemples, le chaînage en anneau des squatters avec la sous-page tête de chaîne est parfaitement visible. Des problèmes de modification de l'anneau se posent en cas d'adjonction ou de suppression de squatters de suppression de la sous-page "tête de chaîne" (qui doit obligatoirement exister, etc...). Nous étudierons quelques-uns de ces problèmes ci-dessous.

#### 4.6. 2. - Modifications dans des anneaux de squatters :

##### 4. 6. 2. 1. - On ajoute un squatter à un anneau existant :

Soit l'anneau de la figure 12, on ajoute un nouveau squatter provenant du pli N°3 et qui trouve un "logement", par exemple, à la page 16, sous-page 2. Comment l'anneau est-il modifié ?

ON PLACE CE NOUVEAU SQUATTER A LA DERNIERE POSITION DE L'ANNEAU. C'est dire qu'il pointera sur la "tête de chaîne" et qu'on fera pointer sur le nouveau venu le squatter qui pointait sur cette tête de chaîne.

On obtient le schéma représenté sur la figure 13.

##### 4. 6. 2. 2. - Suppression d'un squatter :

Supposons que dans l'anneau obtenu ci-dessus (nouvel anneau) on ANNULE les informations contenues dans le squatter provenant du pli 8. La sous-page 6, de la page 35 devient libre. Comment l'anneau est-il modifié ? ON MODIFIE L'EN-TETE DU SQUATTER QUI POINTAIT SUR LA SOUS-PAGE LIBEREE DE FACON A CE QU'IL POINTE SUR LA SOUS-PAGE VERS LAQUELLE POINTAIT LE SQUATTER SUPPRIME. On obtient ainsi le schéma de la figure 14.

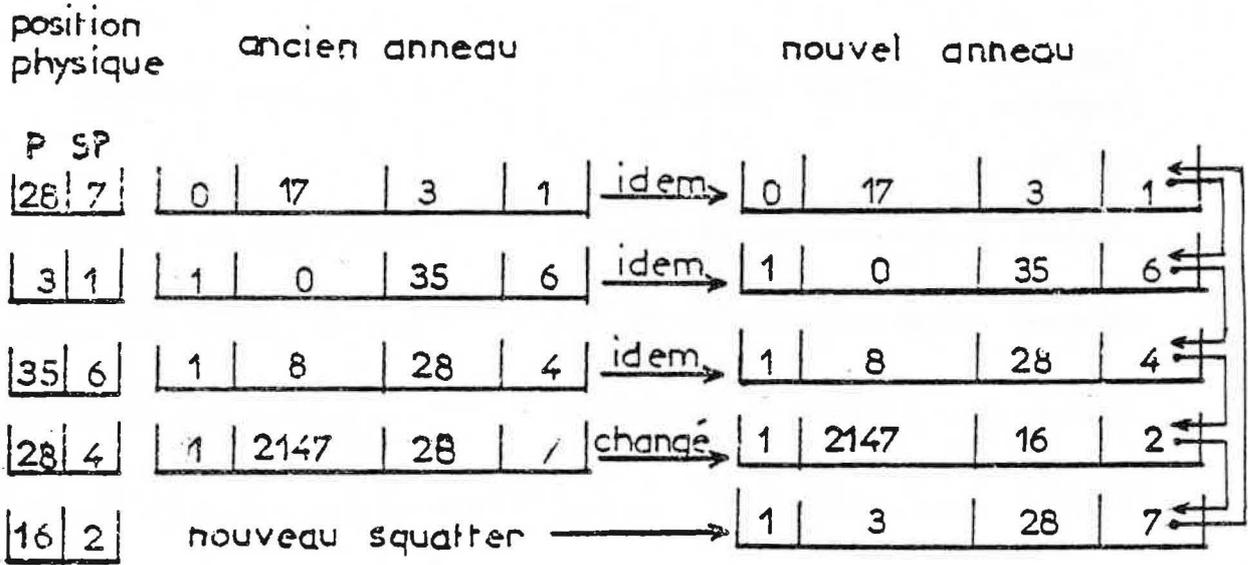


Figure 13 : un squatter est ajouté

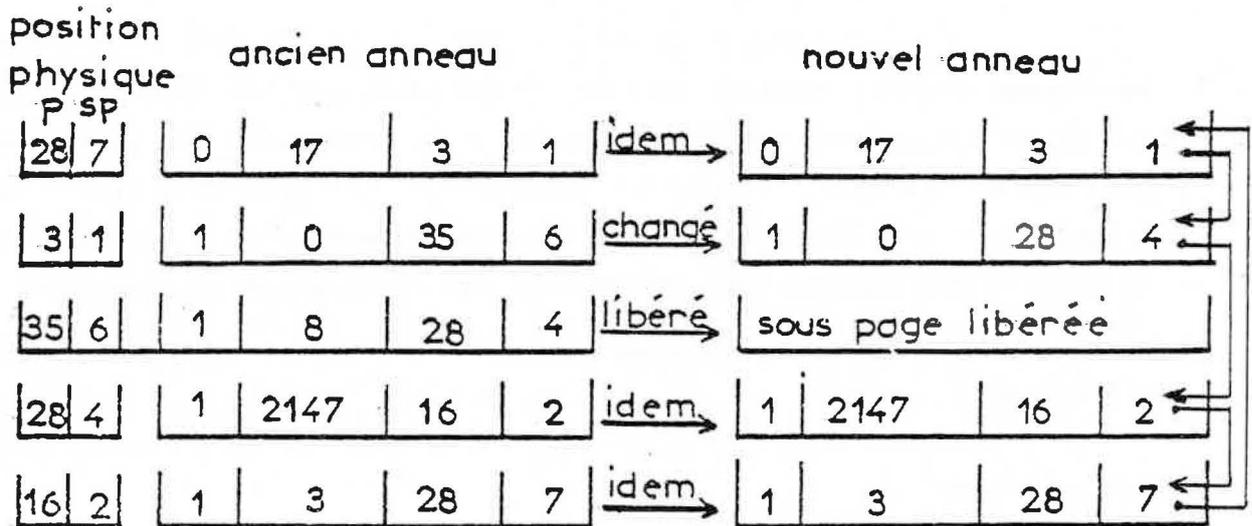


Figure 14 : un squatter est supprimé

4. 6. 2. 3. - La sous-page "tête de chaîne" d'un anneau est libérée :

Le squatter vers lequel elle pointait prend sa place et DEVIENT TETE DE CHAINE ( premier bit = 0 ) de l'anneau. La sous-page occupée par ce squatter est libérée ( figure 15).

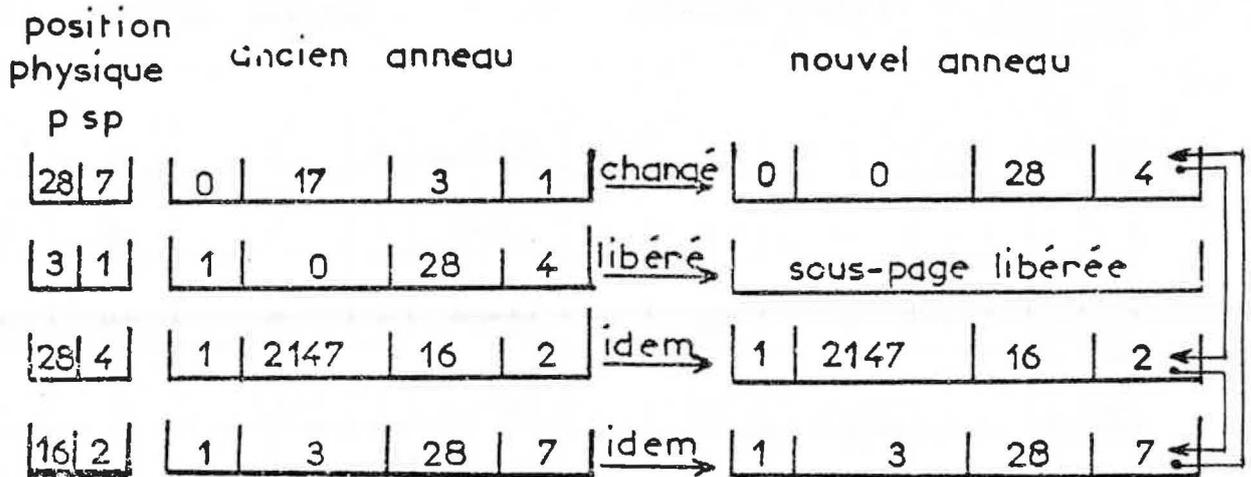


figure 15 : sous-page tête de chaîne libérée

4. 7. - LES RELATIONS ENTRE LES DEUX ESPACES DE LA MEMOIRE REELLE :

Les aspects essentiels de la réalisation du projet seront développés dans le chapitre suivant. C'est parce que les algorithmes gérant les relations entre la mémoire centrale et le disque ont une importance capitale sur le temps d'accès aux informations que nous en parlons ici. Les solutions de type matériel /17/ ou microprogrammées n'étant pas transportables et hors de nos moyens, nous retiendrons une réalisation par logiciel.

On peut regrouper sous trois rubriques les fonctions de ces algorithmes.

- recherche d'une page ( réalisée par le sous-programme APLPAG ( PAGE, IFI).)
- Copie d'informations situées dans le cadre de manoeuvre (fonction réalisée par le sous-programme MODIF ( page, IFI).)
- Recherche d'une sous-page libre ( fonction réalisée par le sous-programme LIBRE (PAGES, SOUS-PAGES, PLEIN, IFI).)

4. 7. 1. - Recherche d'une page :

Connaissant le numéro de la page recherchée (PAG) et le numéro du fichier auquel elle appartient (IFI), il s'agit de trouver cette page (soit en mémoire centrale, soit sur le disque) et de la placer sur le cadre de manoeuvre.

Pour chacun des quatre tableaux contenant les quatre pages situées en mémoire on connaît :

- le numéro du fichier auquel la page appartient
- le numéro de la page
- l'ancienneté d'utilisation

A partir de ces renseignements, le sous-programme APLPAG ( PAG, IFI) réalise les fonctions résumées dans l'algorithme, représenté figure 16 (voir la page suivante).

remarques importantes :

- a) Cette gestion des pages est très simple : Elle est possible grâce au choix que nous expliciterons dans les commentaires concernant "MODIF" (voir ci-après). TOUTE MODIFICATION DE LA PAGE SITUEE SUR LE CADRE DE MANOEUVRE ENTRAINE IMMEDIATEMENT SA COPIE DANS UNE DES PAGES EN MEMOIRE CENTRALE.
- b) Lorsque la page PAG est cherchée sur le disque, c'est l'enregistrement  $IV = PAG + 2$  qui est amené sur le cadre de manoeuvre P. Il faut se rappeler ici que les enregistrements 0 et 1 sont utilisés par le système. La page  $PAG = 0$  (espace virtuel) est donc projetée sur l'enregistrement N° 2 du disque.
- c) Les transferts d'informations dans le sens disque → mémoire centrale sont réalisés uniquement par ce sous-programme.

#### 4.7.2. - Copies d'information situées dans le cadre de manoeuvre :

Ce transfert d'information intervient chaque fois qu'une modification a lieu dans la page contenue sur le cadre de manoeuvre P. L'algorithme retenu (voir figure 17, ci-après) est réalisé par le sous-programme MODIF (PAG, IFI).

- \* IV est le numéro de la page réelle
- \* IFI le numéro du fichier
- \* PAGE le numéro de page virtuelle

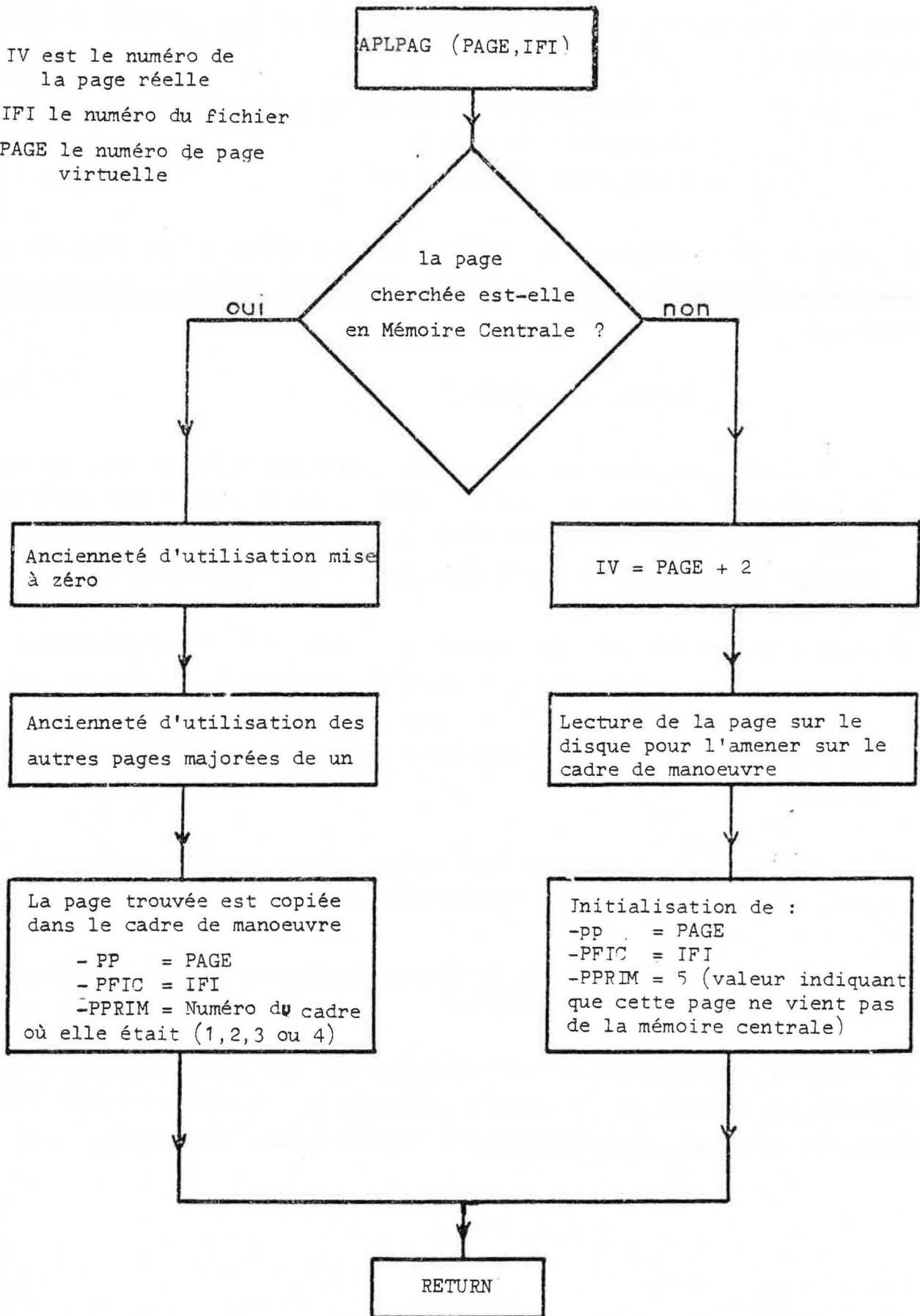


figure 16 : les fonctions réalisées par APLPAG (PAGE, IFI)

- \* IV est le numéro de la page réelle
- \* IFI est le numéro du fichier
- \* PAGE est le numéro de la page virtuelle

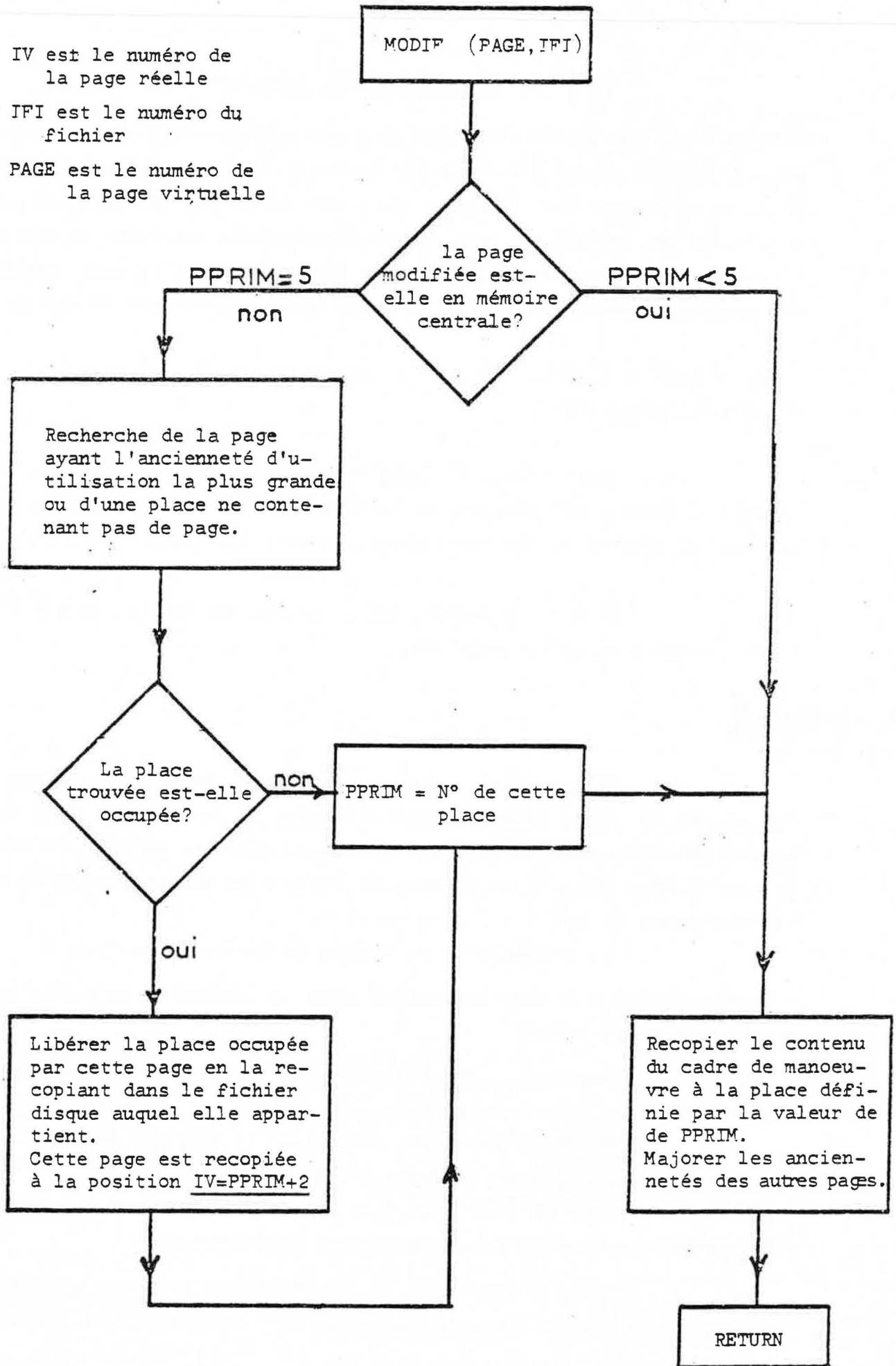


figure 17 : fonctions réalisées par MODIF (PAGE, IFI)

4. 7. 3. - Recherche d'une sous-page libre :

Désignons par NSP le nombre entier de sous-pages qui sont contenues dans une page. A chacune de ces sous-pages correspond un bit dans le mot d'en-tête de page. Si on cherche une sous-page libre on examine les NSP premiers bits de ce mot. Si un bit est nul, la sous-page correspondante est libre et son numéro est l'adresse de la sous-page. Si aucun des bits n'est nul, la page considérée est pleine. Un nouveau numéro de page est alors calculé par la relation :

$$\text{PAG} = (-1)^K \times K + \text{PAG} \quad (K \text{ prenant successivement les valeurs } 1, 2, 3 \text{ etc si cela est nécessaire}).$$

Cette relation permet d'explorer tout le fichier en "encadrant" la page de départ. Par exemple, si la page examinée au début est la page 2 et si elle est pleine, on examinera successivement les pages 3, 1 puis 4, etc...

Si tout le fichier est plein, un message est envoyé à l'utilisateur pour le lui signaler.

4. 8. - CONCLUSION :

Au cours de ce chapitre, un grand nombre de problèmes ont été abordés et, à son issue, il faut à nouveau faire le point. Tous les espaces mémoires seront découpés en sous-pages dont les tailles optimales seront de 8 ou 16 mots réels (7 ou 15 mots virtuels). Ces sous-pages seront regroupées dans des pages de (256 + 1) mots réels.

Le dictionnaire de gestion de mémoire sera formé :

- du mot d'en-tête de page interprété comme une chaîne de bits pour la gestion des sous-pages libres.
- d'un mot par sous-page. Les mots concernant les sous-pages situées dans une même page seront regroupés en-tête de la page.

Ce dictionnaire est imposé par la présence des squatters qui constituent un inconvénient important de la méthode d'allocation de mémoire adoptée : on ne peut plus dire que pour accéder à une information UN SEUL ACCES DISQUE est suffisant dans absolument tous les cas.

Le chaînage en anneau des squatters semble la méthode la plus efficace pour résoudre les problèmes posés par leur gestion. En effet, les

autres solutions possibles (réserve d'une place spéciale dans le fichier pour les squatters, gestion par chaîne de bits comme pour les sous-pages libres, etc...) se heurtent au faits suivants :

+ chaque sous-page "à sa place" peut avoir un nombre quelconque de squatters.

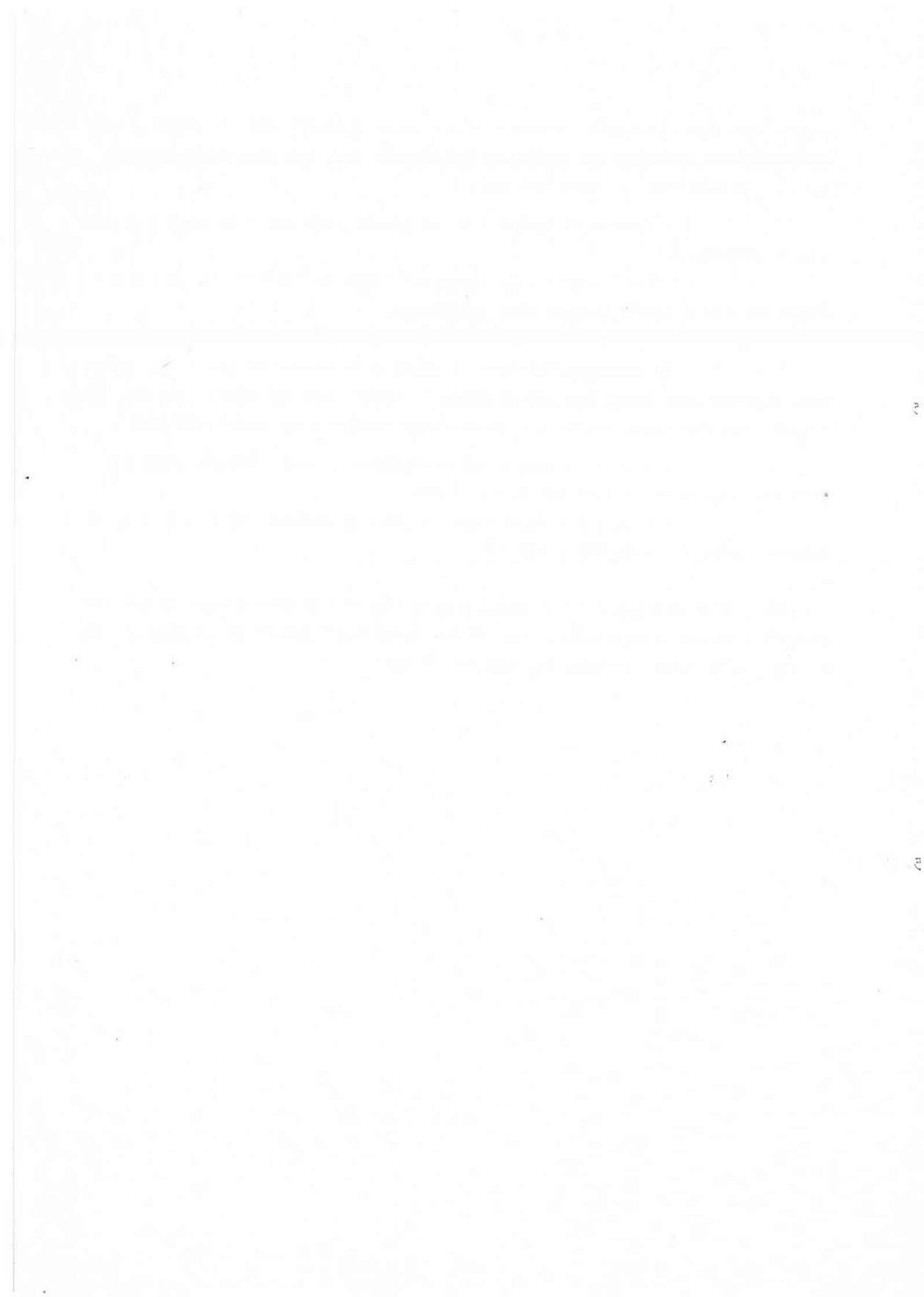
+ par rapport à la sous-page "tête de chaîne", la position d'une sous-page squatter peut être quelconque.

On comprend donc que, à moins d'envisager le cas d'une application précise dans laquelle, par exemple, le nombre des squatters pourrait être estimé, les solutions autres que le chaînage conduiraient inévitablement :

- soit à la saturation ponctuelle (place spéciale réservée pour les squatters se révélant insuffisante).

- soit à une impossibilité dans la gestion (chaîne de bits de longueur insuffisante, par exemple).

Enfin, l'organisation de la mémoire centrale et les algorithmes principaux ont été choisis et présentés. Nous devons maintenant parler de la réalisation du projet. Ce sera le thème du chapitre suivant.



## CHAPITRE V : ASPECTS GENERAUX DU PROJET REALISE

### 5. 1. - LE PROBLEME POSE :

La gestion de mémoire est la base du projet global d'implantation d'une banque de données. La méthode et les moyens étant choisis (chapitres I à IV) nous avons dû tenir compte, également, de la manière générale dont les informations seraient organisées dans la banque. Ce choix, discuté par ailleurs, a conduit à l'utilisation d'une méthode dérivée du projet SOCRATE de Grenoble /2/, /8/, /9/ et /10/ : TOUTE APPLICATION UTILISERA DEUX FICHIERS, UN CONTENANT LES INFORMATIONS ET L'AUTRE LES RENSEIGNEMENTS CONCERNANT LEUR STRUCTURATION. Notre problème devient donc le suivant :

" PROJETER DEUX ESPACES VIRTUELS SUR DEUX ESPACES REELS CONSTITUES PAR DEUX FICHIERS DISQUES. Les nombres de pages et les tailles des sous-pages des deux fichiers étant indépendants et choisis PAR L'UTILISATEUR DE LA BANQUE ".

### 5. 2. - LES OPTIONS RETENUES :

#### 5. 2. 1. - Les fichiers disques :

- définis par DEFINE FILE ( programme) et créés par DKFMT, ils sont découpés en pages de 257 mots.
- le premier mot de chaque page est réservé pour la gestion des sous-pages libres (1 bit par sous-page) -Dictionnaire des sous-pages libres- .
- Les 256 mots restant sont divisés en unités de 32, 16, 8, 4, 2 ou 1 "sous-page".
- Le premier mot de chaque sous-page est réservé à la gestion des sous-pages (dictionnaire des sous-pages occupées).

5. 2. 4. - En mémoire centrale :

Le plan primaire est constitué de :

- une page (P) de 257 mots ( " CADRE DE MANOEUVRE" ) sur laquelle on écrit ou lit les informations. Pour une écriture, le "texte" est placé dans TABLE (320 mots) puis transféré dans le cadre de manoeuvre. Pour une lecture, c'est l'inverse qui se produit : Le "texte" passe du cadre de manoeuvre dans TABLE puis sur un périphérique de sortie.

- une zone de 256 mots, servant à des opérations internes à la gestion de mémoire (copie d'informations à déplacer, pendant la recherche d'une sous-page "libre", par exemple).

- quatre pages, appartenant à l'un ou l'autre fichier, résidant en permanence en mémoire centrale dans des tableaux de 257 mots chacun .

5. 2. 5. - Relations entre mémoire centrale et disque :

Un seul espace en mémoire centrale, décrit ci-dessus, doit être en relation avec DEUX espaces disques : un pour le fichier STRUCTURE DES INFORMATIONS et l'autre pour le FICHER CONTENANT les informations.

Pour chaque page en mémoire centrale (cinq au total en comptant celle qui se trouve sur le "cadre de manoeuvre") un tableau de trois mots est géré ( voir 4.7,1.)

\* un mot contient le N° du fichier auquel elle appartient.  
\* un deuxième indique le numéro de la page elle-même  
\* un troisième l'ancienneté d'utilisation de la page. Cette "ancienneté" est incrémentée toutes les fois qu'une AUTRE PAGE est utilisée pour une lecture ou une écriture. Elle est ramenée à la valeur zéro si la page considérée est ELLE-MEME utilisée.

Les opérations de lecture et d'écriture des informations ont effectivement lieu dans le CADRE DE MANOEUVRE. A partir de cette idée, la gestion des relations entre la mémoire centrale et les fichiers disques, est réalisée de la manière suivante :

\* s'il s'agit d'une lecture, la page n'est pas modifiée. Elle n'est donc pas recopiée.

\* s'il s'agit de l'arrivée d'une page en mémoire centrale ou d'une écriture, le contenu du cadre de manoeuvre est MODIFIE et la page est IMMEDIATEMENT RECOPIEE à l'un des quatre emplacements prévus en mémoire centrale. Cet emplacement est :

- soit celui dont la page provenait. Il n'y a alors aucune relation avec le fichier disque.

- soit celui d'UNE AUTRE PAGE. Alors, parmi les quatre pages en mémoire centrale, celle ayant "l'ancienneté d'utilisation" la plus grande est recopiée dans le fichier disque, puis la page modifiée est recopiée à sa place.

Cette méthode conduit à un nombre réduit de relations entre mémoire centrale et disque, donc à un gain de temps.

#### 5. 2. 6. - Langages de programmation :

Comme prévu, l'essentiel de la programmation a été écrit en FORTRAN. Seuls, trois sous-programmes ( sur vingt) sont écrits en autocode. Ainsi est réalisé un de nos objectifs : produire une base de données la plus indépendante possible de l'ordinateur.

#### 5. 3. - LES FONCTIONS A REALISER :

Le programme de gestion de mémoire doit accomplir les fonctions principales suivantes :

- création et préformatage des fichiers-disques,
- formatage et initialisation des fichiers-disques,
- ouverture des fichiers,
- sauvegarde des informations et fermeture des fichiers
- relations entre mémoire centrale et fichiers-disques,
- lecture des informations
- écriture des informations

Nous décrivons, ci-après, les moyens mis en oeuvre pour réaliser ces fonctions.

#### 5. 4. - CREATION ET PREFORMATAGE DES FICHIERS :

Dans le programme principal (voir ci-dessous "SOMINE"), un

fichier est défini par DEFINE FILE. Par exemple, la carte :

```
DEFINE FILE 50 (102, 257, U IV), 60 ( 402, 257, U IV).
```

définit DEUX fichiers dont les clés d'accès sont 50 et 60, les nombres de pages 102 et 402. U et IV sont des indices utilisables dans le programme. Les pages sont de 257 mots chacune (ce qui correspond aux besoins -déjà indiqués- de la mémoire virtuelle).

Les fichiers définis par DEFINE FILE doivent exister sur le disque. Leur création et leur préformatage, à ce niveau, sont réalisés par un utilitaire appelé DSKFMT. Le listing suivant correspond aux cartes nécessaires pour créer et préformater un fichier de 99 pages, sur un disque VOL - D00130. Le nom du fichier est DSNAME = SOC-DONNEES. 1028 est la conversion en octets du nombre de mots de chaque page ( $1028 = 257 \times 4$ ). 1029 s'en déduit aussi ( $1029 = 1028 + 1$ )

Au moment de l'exécution, une carte DDCB réalise la correspondance entre le fichier préalablement créé par DSKMT et le fichier utilisé dans le programme (défini par DEFINE FILE). Ces deux fichiers doivent naturellement être identiques.

#### 5. 4. 1. - Listing du DSKFMT :

```
#DSKFMT
#DCB DSKFMT, DV=D1, VOL=D00130:99, DSNAME=SOC-DONNEES, RL=1028, RE=NF, #
# BL=1029, ACC=DRQL, STATUS=NEW
```

#### 5. 4. 2. - Remarque sur les options prises à ce niveau :

L'utilitaire DSKMT utilise la page 0 du fichier qu'il crée.

La gestion de la mémoire utilise la page 1.

Lorsqu'on définit un fichier en vue d'une application pour laquelle on désire un nombre de page NP, il faut donc créer un fichier ayant un nombre de page NP + 2, au moins.

exemple : Si l'espace utilisable doit comporter 200 pages, il faudra créer et préformater par DSKFMT et DEFINE FILE un fichier de  $200 + 2 = 202$  pages AU MOINS. Mais si on prévoit une possibilité d'extension de la mémoire utilisable (jusqu'à NP = 1000 pages) dans un avenir proche, rien n'interdit de

définir un fichier de 1002 pages et de n'en utiliser que 200 pendant un certain temps .

## 5. 5. - FORMATAGE et INITIALISATION DES FICHIERS :

### 5. 5. 1. - Les éléments nécessaires pour formater et initialiser un fichier :

Dans son programme, l'utilisateur fait appel à la subroutine FORMFICH (IFI, NPAGE , NTAILLE). Les arguments d'appel sont :

IFI : numéro (1) ou (2) du fichier concerné  
 NPAGE : nombre de pages à formater et initialiser  
 (égal au nombre de pages de la mémoire réelle choisie)  
 NTAILLE : taille des sous-pages (exprimée en puissance de deux)  
 de la MEMOIRE REELLE : par exemple , si on veut des sous-pages virtuelles de 7 mots, on prendra NTAILLE = 3 car la sous-page réelle contient aussi un mot du dictionnaire.

### 5. 5. 2. - Les fonctions de FORMFICH :

A partir de IFI, NPAGE et NTAILLE, FORMFICH appelle une subroutine de calcul (appelée DECOD) et :

- range dans la page 1 du fichier disque les informations nécessaires à toute utilisation ultérieure de la mémoire virtuelle.
- dans chaque page du fichier, FORMFICH initialise à zéro tous les mots du dictionnaire et à - 1 tous les mots utilisables par la banque.

### 5. 5. 3. - Exemple : Formatage de deux fichiers :

Il sera réalisé par deux appels à FORMFICH :

```
CALL FORMFICH ( 1, NPAGE1, NTAILLE1 )
CALL FORMFICH ( 2, NPAGE2, NTAILLE2 )
```

## 5. 6. - OUVERTURE DES FICHIERS :

### 5. 6. 1. - Forme de l'appel :

Lorsqu'il veut disposer du programme de gestion de mémoire,

l'utilisateur place dans son programme FORTRAN la carte :

CALL INFICH (IFI)

Dans laquelle il donne à IFI les valeurs :

1 s'il veut travailler avec seulement le fichier 1  
 2 s'il veut travailler avec seulement le fichier 2  
 3 s'il veut disposer, à la fois, des fichiers 1 et 2. Ce dernier cas étant le plus fréquent... et n'interdisant pas l'utilisation effective d'un seul fichier.

5. 6. 2. - Effets de l'appel :

Le sous-programme INFICH réalise deux fonctions :

- recherche dans la (ou les) première (s) page (s) du (ou des) fichier (s) des informations nécessaires au fonctionnement de la gestion de mémoire (voir FORMFICH).

- initialisation d'un certain nombre de variables servant au début du travail de la gestion de mémoire.

5. 7. - FERMETURE DES FICHIERS ET SAUVEGARDE DES INFORMATIONS :

5. 7. 1. - Forme de l'appel :

Lorsqu'il ne veut plus utiliser la base de données, le programmeur insère dans son programme FORTRAN La carte :

CALL OUTFICH

5. 7. 2. - Effets de l'appel :

Le sous-programme OUTFICH a les fonctions suivantes :

- transférer sur le disque toutes les informations contenues en mémoire centrale, dans les quatre pages qui y résident.

- repasser le contrôle au programme de l'utilisateur qui peut ainsi continuer.

Après l'appel à OUTFICH, les fichiers de la banque de données ne sont plus accessibles.

5. 7. 3. - Sauvegarde des informations :

L'utilisateur doit prendre garde à la fermeture des fichiers : Elle doit être effectuée QUELQUE SOIT la "sortie" empruntée dans son programme pour atteindre l'instruction finale (STOP). Attention notamment aux sorties consécutives à des erreurs.

5. 8. - RELATIONS ENTRE MEMOIRE CENTRALE ET DISQUE :

En raison de l'importance de ces relations, l'algorithme qui les régit a été présenté dans le chapitre 4.

5. 9. - LECTURE D'UNE INFORMATION :5. 9. 1. - Fonctions principales du sous-programme "lecture" :

Appelé directement par SOMINE (sous-programme transparent à l'utilisateur de la banque) par :

CALL LECTURE ( IFI, IX, N, L, ITABLE), ce sous-programme a deux fonctions principales :

a) LIRE UN " TEXTE" :

- dans un fichier IFI = 1 ou IFI = 2
- à partir de l'octet  $N/8 = 0, 1, 2$  ou 3 dans le mot d'adresse virtuelle IX (de 0 à  $2^{33}$  environ).
- la longueur du "texte" étant L (plus grand que zéro) octets. Cette longueur est limitée pour l'instant à 1280 octets (320 mots).
- le texte est transféré dans ITABLE puis édité sur un périphérique en format approprié.

b) LIRE LA VALEUR D'UN BIT :

- La longueur L est alors obligatoirement nulle. Cette information sert de test entre les deux cas d'utilisation, lecture d'un "texte" ou d'un bit.
- la position du bit dans le mot d'adresse IX est définie par N qui peut prendre toutes les valeurs de 0 à 31.
- la valeur du bit est mise dans ITABLE (1) ou elle peut être utilisée d'une manière quelconque.

5. 9. 2. - Organigramme des fonctions du sous-programme  
"lecture" :

Voir l'annexe I.

5. 10 - ECRITURE D'UNE INFORMATION :

5. 10. 1. - Fonctions principales du sous-programme "ECRIRE" :

Appelé directement par SOMINE sous la forme :

CALL ECRIRE (IFI, IX, N, L, ITABLE), ce sous-programme a deux fonctions principales :

- a) ECRIRE UN "TEXTE" :

- dans un fichier IFI = 1 ou IFI = 2
- à partir de l'octet N/8 ( N/8 = 0, 1, 2, ou 3) du mot d'adresse virtuelle IX ( de 0 à environ 2<sup>33</sup>)
- la longueur du texte étant de L (plus grand que zéro) octets. Cette longueur est limitée pour le moment à 1280 octets.
- le texte a été placé dans ITABLE par un périphérique quelconque. Il est écrit dans le fichier sous-page après sous-page jusqu'à épuisement de la longueur L.

- b) POSITIONNER UN SEUL BIT :

- la longueur L est alors obligatoirement 0. Cette information sert de test entre les deux cas d'utilisation : écriture d'un nombre d'octets ou d'un bit.
- la position du bit dans le mot d'adresse IX est définie par N qui peut prendre toutes les valeurs de 0 à 31.
- la valeur à donner au bit (0 ou 1) est stockée dans ITABLE (1) ou elle a été amenée par un moyen quelconque.

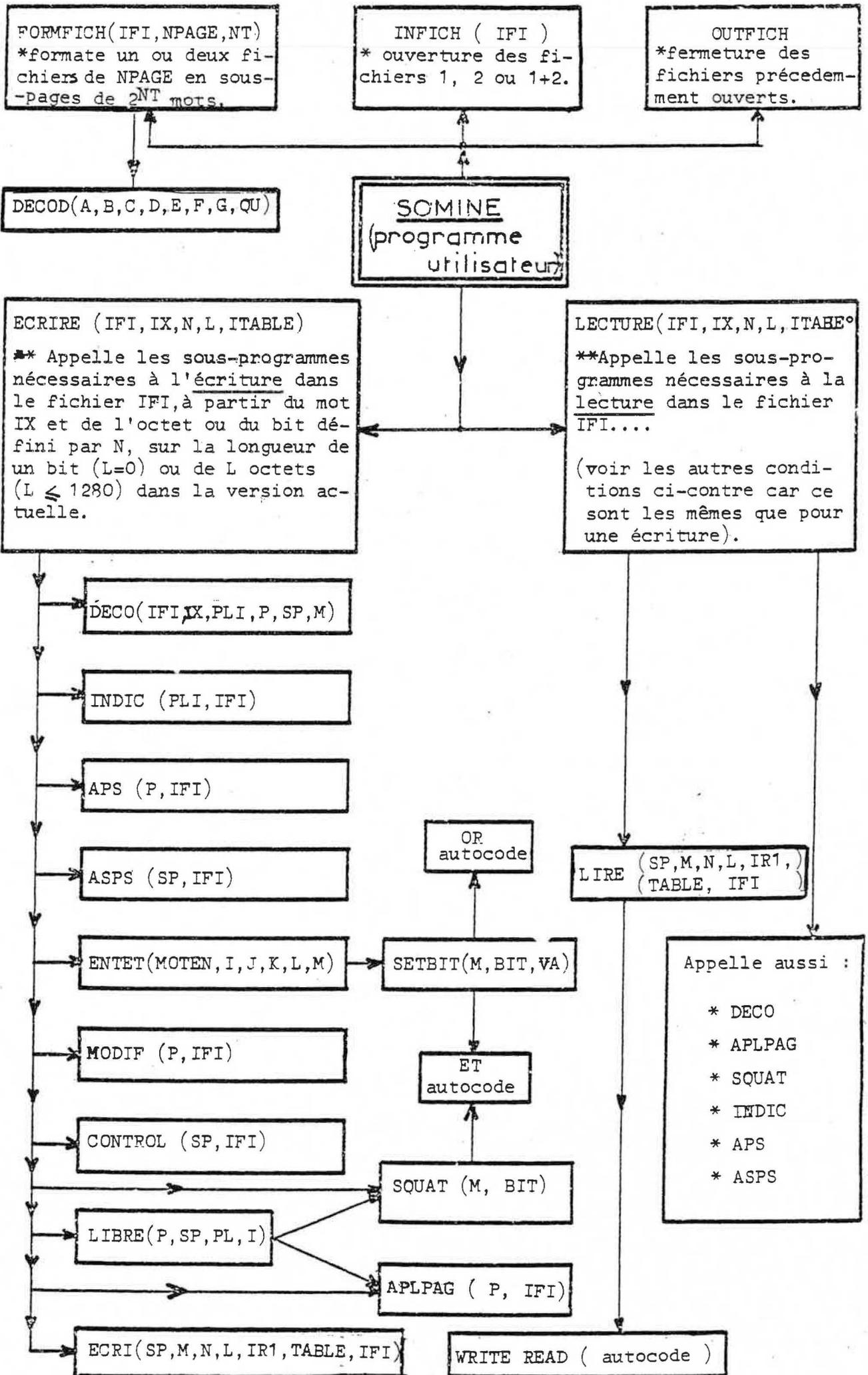
5. 10. 2. - Organigramme des fonctions du sous-programme  
"écrire" :

Voir l'annexe I

5. 11 - SCHEMA DE L'ORGANISATION DES SOUS-PROGRAMMES DE LA GESTION DE MEMOIRE :

(voir figure 18 à la page suivante)

.../...





## CHAPITRE VI : PREMIERS RESULTATS ET ...

### .. NOUVELLES QUESTIONS

#### 6. 1. - PREMIERS RESULTATS :

##### 6. 1. 1. - Un critère d'optimisation :

Dans l'article déjà cité /17/, les auteurs pensent qu'un bon critère pour mesurer la valeur d'une gestion de mémoire paginée est :

$$r = \frac{e}{e + T} \quad (I)$$

expression dans laquelle (e) est le temps moyen entre deux "accès pages" et (T) le temps moyen d'un accès page.

Si on pose  $F = \frac{1}{e}$  on peut écrire la relation (I) sous la forme

$$r = \frac{1}{1 + F \cdot T} \quad (II)$$

dans laquelle F est le "taux de défaut de page".

T est une caractéristique du système (12,5 ms sur l'ordinateur Philips P 1175 équipé de disque à têtes mobiles).

F doit être minimisé par la gestion de mémoire.

##### 6. 1. 2. - Notre système jugé par ce critère :

Bien des efforts ont été déployés pour minimiser la valeur de F. Nous rappellerons à ce sujet :

- la recherche d'une organisation de la mémoire centrale permettant d'admettre plusieurs pages simultanément dans l'espace "tore".

- le regroupement des sous-pages en pages qui permet de limiter le nombre des accès disques tout en limitant la place perdue par fragmentation.

- l'algorithme gérant les relations entre la mémoire centrale et les deux fichiers disques qui permet généralement de garder en mémoire centrale, pratiquement en permanence, l'ensemble de la structure ( une ou deux pages). Ainsi, les opérations sur ces pages ( les plus fréquentes) sont entièrement réalisées en mémoire centrale.

- la détermination de la taille de page de façon à pouvoir, presque dans tous les cas, appeler d'un seul coup une information du disque vers la mémoire centrale : anneau de squatters presque toujours contenu sur une seule page , possibilité de présence simultanée de deux pages du fichier données en mémoire centrale.

- utilisation d'un cadre de manoeuvre.

On peut donc espérer, que face à ce critère, notre système aura un comportement favorable. Naturellement, seules des mesures sérieuses pourront le prouver. Dès que la banque sera utilisée de façon opérationnelle, pour une application concrète, nous essayerons de mesurer les valeurs nécessaires. Mais il faut bien comprendre que cela ne peut être fait que dans le cadre d'une application complète car la manière dont les informations sont structurées le taux de remplissage des fichiers et le choix judicieux de la taille des fichiers disques par rapport aux taille des entités ont des influences importantes sur l'évaluation de ce critère.

#### 6. 1. 3. - Place occupée en mémoire centrale :

La taille du plan primaire proprement dit ( comprenant les quatre pages et le cadre de manoeuvre) est de CINQ K-OCTETS. Cette taille est nécessaire si on veut obtenir une vitesse de recherche des informations acceptables.

#### 6. 1. 4. - Complexité des algorithmes :

On peut discuter la présence du "cadre de manoeuvre" et il faut reconnaître qu'il n'est pas indispensable. Il occupe 1 K-octets en mémoire centrale et sa copie fréquente sur une des pages en mémoire fait perdre un peu de temps au cours des écritures d'informations. Par contre, sa présence a deux avantages :

- simplification des algorithmes de programmation
- important gain de temps au moment des lectures successives sur plusieurs pages : les pages nouvelles "écrasant" les anciennes dans le cadre de manœuvre.

Ces avantages nous ont parus déterminant. N'oublions pas que la programmation a été faite essentiellement en Fortran. Donc, la simplification des algorithmes à programmer était importante. Enfin, la consultation d'une banque, plus encore que la rentrée des informations, se doit d'être "rapide".

6. 1. 5. - Risque de saturation de l'espace virtuel :

Il est faible mais pas nul. Ses conséquences sur une application seraient catastrophiques puisqu'il serait impossible d'écrire une seule information sur un fichier des données entièrement vide ! Nous avons donc cherché à optimiser sa taille. Nous pensons avoir réalisé un progrès important par rapport à SOCRATE puisque de  $2^{31}$  mots dans ce système, nous sommes passés à près de  $2^{40}$  mots dans beaucoup de cas.

6. 1. 6. - Risque de saturation ponctuelle :

En logeant les squatters d'une sous-page donnée n'importe où dans l'espace réel, nous avons réduit ce risque à néant.

6. 2. - LES PROBLEMES A RESOUDRE :

A travers la réalisation de la gestion de mémoire de SOMINE, nous avons été amenés (voir bibliographie) à faire le point des connaissances sur les problèmes qui se posent aux concepteurs de mémoires virtuelles appliquées aux bases de données. Pour le type de gestion que nous avons choisi deux problèmes principaux restent à résoudre :

- Comment réduire au minimum le nombre de squatters ?
- Comment déterminer la valeur optimale de la taille de la page (unité de transfert des informations entre deux niveaux de mémoire réelle)?

En effet, la gestion des squatters constitue un inconvénient majeur de la fonction de relocation, adoptée pour passer de l'espace virtuel à l'espace disque. On pourrait sans doute trouver des méthodes plus simples (zone de débordement, par exemple) pour gérer les squatters dans le cadre d'une application particulière. Mais on ne connaît pas de méthode plus simple dans le cas général.

Quant au problème de la taille de l'unité de transfert entre deux niveaux de mémoire, il a été abordé par beaucoup d'auteurs /2/, /11/ /6/ etc.... mais sa solution reste encore à trouver.

### 6. 3. - UNE IDEE D'APPLICATION :

Les réponses aux questions précédentes passent sans doute par le développement de "l'analyse des systèmes informatiques" /2/. Notre système pourrait contribuer aux expérimentations nécessaires car la programmation en langage évolué permet de modifier facilement ses caractéristiques et son fonctionnement. On pourrait, pour des applications données, faire des mesures dans des cas de tailles de pages différentes, avec ou sans cadre de manoeuvre ainsi que des statistiques sur la répartition des squatters dans l'espace réel.

**DEUXIEME PARTIE : ANNEXE I**

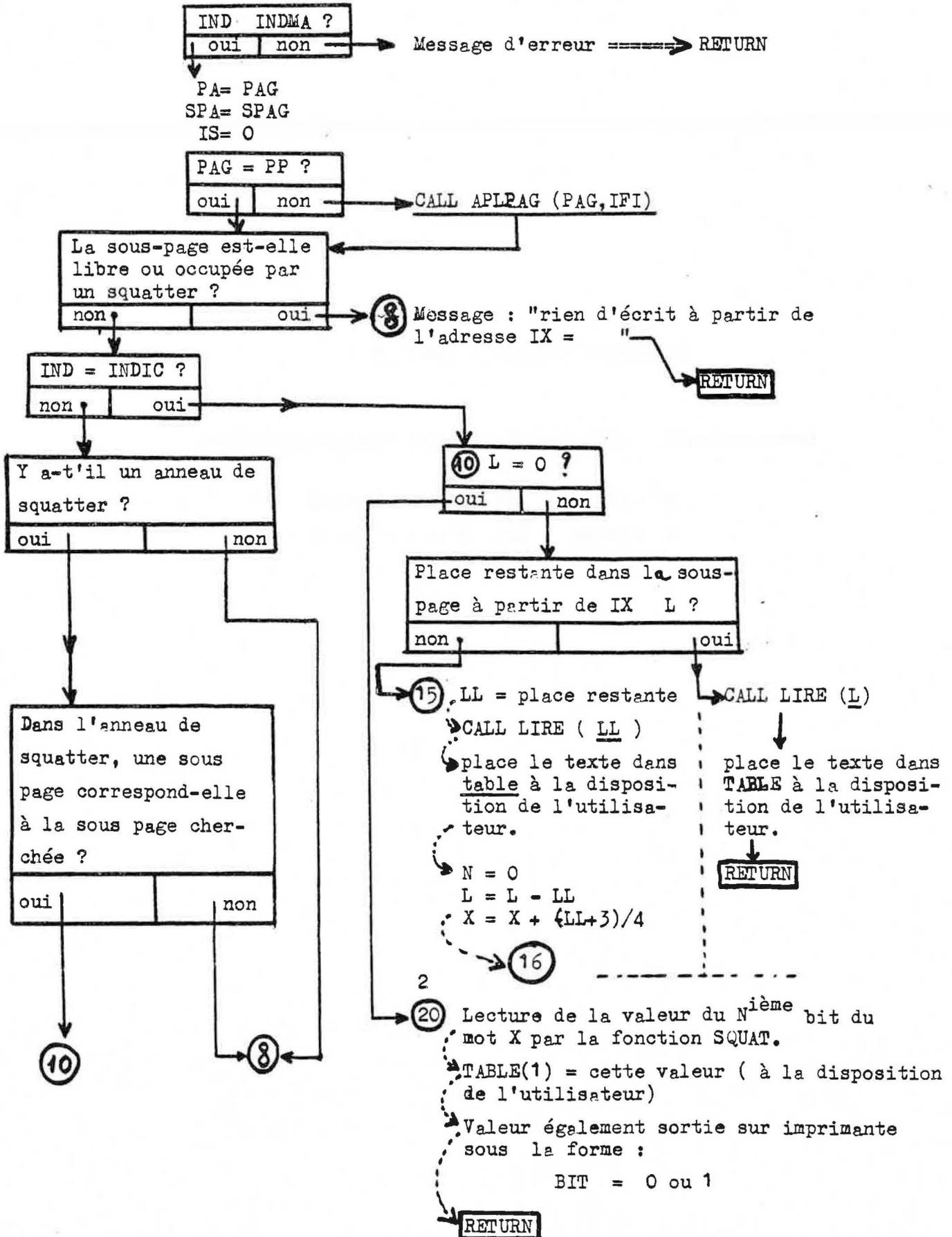
**ORGANIGRAMMES DES FONCTIONS DES SOUS-PROGRAMMES**

- LECTURE (IFI, IX, N, L, TABLE)
- ECRIRE (IFI, IX, N, L, TABLE)

ANNEXE I

ORGANIGRAMME DES FONCTIONS DU SOUS-PROGRAMME LECTURE (IFI,IX,N,L,TABLE)

16 Initialisation et décodage de l'adresse (donne IND, PAG, SPAG et MOT)



ORGANIGRAMME DES FONCTIONS DU SOUS-PROGRAMME ECRIRE (IFI,IX,N,L,TABLE)

(3 pages numérotées II - 63, II - 64 et II - 65)

Initialisation: PT = 0,...etc (voir listing)  
 Calcul de l'adresse X1 du DERNIER mot à écrire  
 et CALL DECO qui fournit IND.

IND    INDMA ?  
 oui | non → Message d'erreur : "adresse trop grande" **RETURN**

L = 0 ?  
 oui | non

La valeur à donner  
 au bit est placée  
 dans TABLE(1)

Le texte est placé dans TABLE(1), TABLE(2),  
 TABLE(3), etc...

⑨ CALL DECO (IFI,IX,...etc...)  
 LL = place restante dans la sous-page à partir  
 du mot IX et de l'octet défini par N.

LL L ?  
 oui | non

⑬ LL = L  
 REC = 0

⑫ L = L - LL  
 REC = 1

LA PAGE DANS LAQUELLE ON VEUT  
 ECRIRE EST-ELLE SUR LE CADRE  
 DE MANOEUVRE P ?  
 non | oui

CALL APLPAG (PAG,IFI)

LA SOUS-PAGE CHERCHEE (SPAG)  
 EST-ELLE LIBRE ?  
 (CALL LIBRE (SPAG,IFI))  
 oui | non

On écrit (CALL ECRI (...))  
 Après cette écriture, la sous-page  
 contient-elle une information  
 nulle ? (test réalisé par CONTROL)  
 oui | non

On va à ⑰ (VOIR PAGE SUIVANTE) →

⑯ REC = 0 ?  
 oui | non

**RETURN**

X = X + (LL+N/8)/4  
 N = 0  
 GO TO ⑨

+ CALL ENTET (chaînage de la sous-page  
 sur elle-même)  
 + CALL SETBIT (déclare la sous-page  
 occupée).  
 + CALL MODIF (recopie le cadre de  
 manoeuvre dans une des places en M. C.)  
 mémoire centrale)



**DEUXIEME PARTIE : ANNEXE II**

**NOTES BIBLIOGRAPHIQUES**

- /1/ - PAIR, Professeur à l'Ecole Nationale Supérieure de la Métallurgie et de l'Industrie des Mines de NANCY, "cours de structure de données et algorithmes fondamentaux (1974)
  
- /2/ - J.R. ABRIAL, J. BAS, J.P. CAHEN, J.C. FAVRE, G. MAZARE et R. MORIN, Université de Grenoble, Institut de Mathématiques appliquées, " Projet Socrate : Données, mémoires et utilisateurs d'une banque de données" (Juin 1971)
  
- /3/- R. MAHL, Professeur à l'Ecole Nationale Supérieure des Mines de Saint-Etienne, " Cours d'algorithmique et de structure de données" ( 1973-1974)
  
- /4/ - Cl. SAYETTAT, Thèse de Doctorat de 3ème cycle, Ecole Nationale Supérieure des Mines de Saint-Etienne, "Contribution à la conception, l'élaboration et l'utilisation du système de bases de données SOMINE : les accès aux bases - Aide à la conception assistée par ordinateur" - janvier 76 -
  
- /5/ - P. MARTY, Thèse de Doctorat de 3ème cycle, Ecole Nationale Supérieure des Mines de Saint-Etienne, " Contribution à la conception, l'élaboration et l'utilisation du système de bases de données SOMINE : structuration des informations " - janvier 76 -
  
- /6/ - R. EHRMANN Monographie de l'AFCEP, " L'allocation dynamique de la mémoire des ordinateurs " DUNOD 1972
  
- /7/ - C. CORVAISIER, Maître-assistant à l'Ecole Nationale Supérieure des Mines de St.Etienne , " Le système PHILIPS P 1000 " (1973).
  
- /8/ - J.R. ABRIAL, J. BAS, G. BEAUME, G. HENERON, R.MORIN et G. VIGLIANO, "Projet Socrate : spécifications générales (I, II, III) communication I.M.A.G. , Grenoble (Août 70)
  
- /9/ - G. BEAUME, thèse de doctorat de 3ème cycle de l'Université de Grenoble " Projet Socrate : gestion des mémoires " (décembre 70)

- /10/ - J.R. ABRIAL, J.P. CAHEN, J.C. FAVRE, D. PORTAL, G. MAZARE, R. MORIN :  
 "Projet Socrate : nouvelles spécifications (version 3) - communications  
 de l'I.M.A.G. - Grenoble ( septembre 1972)
  
- /11/ - P.J. DENNING : " Virtual Memory " - computer Surveys, vol. 2 N° 3  
 septembre 1970.
  
- /12/ - B. RANDELL + "A note on storage fragmentation and programm segmentation"  
 Communication of the A.C.M. Vol 12, n°7 - July 1969
  
- /13/ - R. DALEY et DENNIS : " Virtual Memory and sharing in MULTICS " -  
 communication of the A.C.M. Vol 11 , n° 7 - May 1968
  
- /14/ - J.J. GIRARDOT et MIRAOUX chercheurs à l'Ecole Nationale Supérieure des  
 Mines de St. Etienne : " Un compilateur pour le langage A.P.L. "  
 ( à paraître )
  
- /15/ - E. WOLMAN : " A fixed optimum cell size for records of various lengths"  
 Communication of A.C.M. VOL 12, N° 1 , janvier 1965
  
- /16/ - R. THOMAS, Chef du Centre de Calcul de l'Ecole Nationale Supérieure  
 des Mines de St Etienne : " Evaluation du système Socrate en vue de son  
 utilisation éventuelle comme support d'une banque de données sur les  
 enquêtes faites par le Ministère du Développement Scientifique et  
 Industriel " décembre 1972
  
- /17/ - E. GLENBE, A. BRANDWAJN, D. POTIER et J. LENFANT : "Algorithme de gestion  
 simultanée mémoire centrale-disque de programmation " - Communication  
 aux journées d'études sur les recherches en Mesures, Simulation et  
 Modélisation des systèmes - Toulouse ( juin 1973).



**TROISIEME PARTIE**

**SOMINE ET L'ENSEIGNEMENT ASSISTE PAR ORDINATEUR**



**SOMMAIRE DE LA TROISIEME PARTIE**  
**(SOMINE ET L'E.A.O.)**

	<u>PAGE</u>
CHAPITRE 1 : <u>INTRODUCTION</u>	III - 5
1.1. - LE POINT SUR LES RECHERCHES EN E.A.O.	III - 5
1.2. - NOS OBJECTIFS DE RECHERCHE	III - 7
1.3. - CONCLUSION	III - 8
 CHAPITRE 2 : <u>DE L'ANALYSE DE LA MATIERE A ENSEIGNER A LA STRUCTURATION</u> <u>DES INFORMATIONS DANS SOMINE :</u>	 III -11
2.1. - INTRODUCTION : ANALYSE DE LA MATIERE A ENSEIGNER	III -11
2.2. - LES CRITERES D'ORGANISATION D'UN DOMAINE D'ENSEIGNEMENT	III -12
2.2.1. - Liste des éléments et exploration d'un élément	
2.2.2. - Comment apprend - on ?	
2.2.3. - Critères d'organisation et fonctions d'enseignement	
2.3. - STRUCTURATION DES INFORMATIONS D'UN DOMAINE D'ENSEIGNEMENT	III -15
2.4. - STOCKAGE DES INFORMATIONS DANS SOMINE	III -18
2.4.1. - Rappel sur SOMINE	
2.4.2. - Conséquences sur l'application à l'enseignement	
2.5. - CONCLUSION	III -21
 CHAPITRE 3 : <u>SOMINE CENTRE D'UN SYSTEME D'ENSEIGNEMENT :</u>	 III -23
3.1. - NECESSITE D'UN LANGAGE DE COURS	III -23
3.2. - PRESENTATION DU LANGAGE DE COURS 'EPMINE'	III -23

3.2.1. - Exemple de séquence de cours	
3.2.2. - Quelques remarques	
3.3. - MISE A JOUR ET MODIFICATION DES INFORMATIONS	III - 27
3.4. - INTERROGATION DES INFORMATIONS	III - 27
3.5. - RECHERCHE DES RELATIONS ENTRE ELEMENTS	III - 29
3.5.1. - Recherche des différences entre deux éléments	
3.5.2. - Recherche de similitude	
3.6. - CONCLUSION	III - 34
 CHAPITRE 4 : <u>ENSEIGNEMENT DES BASES DU LANGAGE FORTRAN :</u>	 III - 39
4.1. - LES GRANDES OPTIONS	III - 39
4.2. - COURS STOCKE DANS UNE BANQUE SOMINE	III - 41
4.2.1. - Analyse de la matière	
4.2.2. - Choix de la décomposition d'un élément	
4.2.3. - Structure formelle " SOMINE "	
4.2.4. - Exemples de "pages" de ce cours FORTRAN	
4.3. - COURS " TUTORIEL " FORTRAN	III - 44
4.4. - PREMIERS RESULTATS	III - 45
 CHAPITRE 5 - <u>ENSEIGNEMENT DE LA STATIQUE : EXEMPLE DE PROGRAMME</u> <u>D'AIDE PEDAGOGIQUE" :</u>	 III - 47
5.1. - PROBLEMES POSES PAR L'ENSEIGNEMENT DE LA MECANIQUE RATIONNELLE	III - 47

5.2. - LES PROGRAMMES "D'AIDE PEDAGOGIQUE"	III - 48
5.3. - UN EXEMPLE D'AIDE PEDAGOGIQUE : LE PROGRAMME "MECANIK"	III - 49
5.3.1. - Les étapes d'un problème de statique	
5.3.2. - Les fonctions principales du programme MECANIK	
5.3.3. - Exemple d'application du programme MECANIK	
5.4. - MECANIK : UN ANALYSEUR ET UN GENERATEUR DE PROBLEMES	III - 55

CHAPITRE 6 : <u>UN ESSAI D'APPLICATION DE L'E.A.O. A LA REEDUCATION</u> <u>D'ENFANTS DYSLEXIQUES</u> (travail réalisé en collaboration avec J. FOURNEL, instituteur spécialisé)	III - 57
--	----------

6.1. - BÜTS DE LA RECHERCHE SUR LA REEDUCATION DES DYSLEXIQUES	III - 57
6.2. - CHOIX DES ELEVES ET DES MOYENS	III - 57
6.2.1. - Les élèves	
6.2.2. - Les cours	
6.2.3. - Apprentissage du fonctionnement du système	
6.2.4. - Mesures quantitatives	
6.3. - LES RESULTATS OBTENUS	III - 60
6.3.1. - Résultats quantitatifs	
6.3.2. - Propos et comportements d'élèves	
6.4. - CONCLUSION	III - 62
6.5. - VERS DE NOUVEAUX ESSAIS	III - 63

CHAPITRE 7 : <u>BILAN ACTUEL DES RECHERCHES EN E.A.O. :</u>	III - 65
7.1. - APPORT DE SOMINE A UN SYSTEME D'E.A.O.	III - 65

7.2. - CONSTRUCTION D'UN SYSTEME D'E.A.O. AUTOUR D'UN SYSTEME DE BASES DE DONNEES	III - 66
7.3. - UTILISATION DE LA FORTE MOTIVATION DES ELEVES PAR L'E.A.O.	III - 67
7.4. - POUR AMELIORER LE SYSTEME D'ENSEIGNEMENT	III - 68
<u>ANNEXE 1</u> : EXEMPLE D'UTILISATION DU PROGRAMME "MECANIK"	III - 69
<u>ANNEXE 2</u> : EXEMPLE DE "LECON" PROGRAMMEE AVEC EPMINE (rééducation des dyslexiques)	III - 75
<u>ANNEXE 3</u> : BIBLIOGRAPHIE	III - 81

## CHAPITRE I : INTRODUCTION

### 1. 1. - LE POINT DES RECHERCHES EN ENSEIGNEMENT ASSISTE PAR ORDINATEUR :

Depuis 1958, date de la première expérience connue /1/, l'enseignement assisté par ordinateur s'est développé de façon spectaculaire. Des firmes comme R.C.A. et I.B.M. ont investi des millions de francs dans les recherches et les réalisations de ce secteur. Des milliers d'articles ont été publiés. Dans sa bibliographie FENEUILLE /2/ comptait en 1971.

- 10 centres de recherche en France
- 21 centres de recherche aux Etats-Unis

Il citait plus de 200 articles relatant des applications de l'ordinateur dans les domaines aussi divers que :

- l'enseignement proprement dit
- l'interrogation (exercices et tests)
- la simulation (études de cas, études scientifiques, jeux etc)
- la gestion pédagogique
- la documentation
- la technologie de l'éducation
- la recherche pédagogique

Depuis 1971, les rapports de KAISER et COULON /3/ et celui publié récemment par le MINISTERE DE L'EDUCATION DU QUEBEC /4/ ont fait le point des recherches mondiales. A partir de ces documents, il semble que la recherche s'oriente plus particulièrement vers :

- l'assouplissement de l'emploi de la machine par l'équipe d'enseignement

-- l'assouplissement du dialogue Homme-Machine avec, comme principales directions l'étude des communications en langage naturel et celle des communications orales

- l'extension des expériences existantes (problème des réseaux d'enseignement)

- l'utilisation de l'ordinateur comme instrument de recherches pédagogiques

Au point de vue "matériel", on peut dire que "tout est possible" :

- couplage de cours d' E.A.O. et de films ou de diapositives  
- écrans alphanumériques et télétypes de toutes sortes (par exemple .... silencieuses !)

- écrans graphiques permettant de visualiser certains phénomènes complexes.

Au point de vue logiciel, on est frappé par la diversité et la qualité des réalisations :

- langages de cours variés et nombreux : coursewriter /6/, Magister /7/, Lyon /8/, Planner /9/ etc....

- utilisation de banques de données: /10/, /II/, /12/, etc...

- méthodes de simulation /13/ et /14/, utilisation d'automates /15/ etc....

- .....

On connaît également l'immense projet C-BE ( Computer Based Science and Engineering Education) de l'Université du Texas /16/ dont le budget, de 1972 à 1976 sera de plus de deux millions de dollars. En France, le système O.P.E. / 37 / est opérationnel depuis 1968. Des centres importants existent aussi à Grenoble, Lyon, Toulouse, Lagnion, St-Cloud, etc....

Au milieu de tant de recherches, comment situer notre travail ? D'abord, comme nous l'avons souligné dans la première partie de ce rapport ( voir l'introduction générale du mémoire ), l'utilisation des bases de données dans l'enseignement assisté par ordinateur n'est pas très développée. En France, on peut dire que cette utilisation est pratiquement nulle malgré les travaux de PEUCHOT /5/, très connus à l'étranger. Les recherches dans ce domaine sont parties soit dans la direction de la simple "recherche de faits"

( fact retrieval ) apportant peu de nouveauté par rapport à l'E.A.O. tuteuriel classique, soit dans la direction inverse de la "déduction complète" qui pose les problèmes ardues de l'intelligence artificielle. Les progrès sont donc lents.

Notre travail, à travers la réalisation des objectifs décrits ci-dessous devrait apporter une contribution à ces recherches et aboutir à la mise en place d'un système d'enseignement assisté par ordinateur dans lequel l'étudiant aurait la possibilité de prendre beaucoup d'initiatives (tendance à la non directivité). Dans ce système, les informations à fournir à l'étudiant seraient déduites à la fois de ses questions et des informations, stockées et structurées par une équipe enseignante, dans une banque de données de type SOMINE.

#### 1. 2. - NOS OBJECTIFS DE RECHERCHE :

Confier à l'ordinateur une fonction d'enseignement exige dans tous les cas :

- l'analyse de la matière à enseigner : découpage de la matière en éléments, recherche des relations entre ces éléments et restructuration du "cours" à partir de cette analyse.
- le stockage des éléments et des relations dans des "fichiers" ou dans une "banque d'informations".
- l'application du "cours" (il vaudrait mieux écrire : " de la stratégie pédagogique") défini ci-dessus à des groupes d'étudiants pour évaluer les résultats obtenus et les comparer aux ... espérances initiales.

Notre recherche portera sur ces quatre points et se développera à partir des idées directrices suivantes :

a) POUR L'ANALYSE DE LA MATIERE A ENSEIGNER : nous montrerons ce que peut apporter à la pédagogie une méthode d'analyse de la matière à enseigner utilisant les structures d'arbres et de graphes employées dans les BASES DE DONNEES SOMINE. Cela nous permettra d'insister sur les RELATIONS entre les éléments d'un enseignement.

b) STOCKAGE DES INFORMATIONS : Les informations et leurs relations seront stockées dans une banque SOMINE. Ainsi, la mise à jour et la

recherche des informations s'effectueront en mode conversationnel sur écrans alphanumériques, en langage quasi-naturel. Ce mode original de stockage permettra notamment à l'étudiant :

- \* de recevoir des informations et des questions à partir des cours stockés dans la banque. Ses réponses aux questions seront alors analysées. Des conseils pédagogiques en seront déduits.

- \* d'écrire dans la banque, à des emplacements prévus, pour suggérer des améliorations du cours ou faire part de ses remarques.

- \* d'obtenir de la banque des comparaisons d'informations (similitude, différence, liens de causalité entre deux éléments du cours...)

- \* de poser lui-même, à partir de ses propres données, des exercices, de projeter au moment choisi des diapositives ou de provoquer une simulation du phénomène étudié grâce à des programmes "d'aide pédagogique".

- \* de diriger lui-même son cheminement dans l'exploration de la matière à enseigner en tenant compte ou non des conseils pédagogiques donnés par le système en fonction de ses résultats.

c) MESURE DE L'EFFICACITE DU SYSTEME : en dehors des mesures classiques concernant les temps de réponse, l'encombrement de la mémoire, nous nous efforcerons de nous placer du côté de l'utilisateur et d'évaluer l'efficacité de ce système à partir :

- \* d'applications classiques : cours de FORTRAN, de MECANIQUE RATIONNELLE ... etc.

- \* d'applications NOUVELLES dans lesquelles la forte motivation des élèves lorsqu'ils utilisent un ordinateur sera employée pour modifier des comportements dans des domaines tels que l'intensité de l'attention, l'esprit d'initiative, la mémoire. Cette dernière partie pourrait déboucher sur des méthodes nouvelles de traitement des dyslexies et des dysorthographies.

### 1. 3. - CONCLUSION :

Il semble /17/ que l'un des principaux avantages de l'E.A.O. est constitué par le potentiel "d'inventions possibles" qu'il a apporté à la pédagogie.

Ce potentiel provient de plusieurs facteurs :

- \* clarification des buts de l'enseignement
- \* individualisation de l'enseignement
- \* forte motivation des étudiants (renforcement immédiat, évaluation rationnelle du travail, etc..)
- \* démystification de l'ordinateur

Dans le système réalisé à l'ECOLE NATIONALE SUPERIEURE DES MINES de ST.ETIENNE, nous nous sommes efforcés d'utiliser au mieux ces qualités en offrant aux étudiants les possibilités suivantes :

a) obtenir des informations sur l'élément qu'il veut et au moment où il le veut par l'utilisation des fonctions d'interrogation en langage quasi naturel de SOMINE

b) rechercher les RELATIONS entre les éléments du cours en utilisant des fonctions d'interrogations DES RELATIONS entre deux éléments.

c) Améliorer le cours concernant chacun des éléments ou formuler des remarques en utilisant la fonction "AMELIORATION" qui lui permet d'écrire dans la banque. Cette propriété de notre système nous paraît particulièrement importante car on sait combien est fastidieux le dépouillement des compteurs relatant le cheminement d'un étudiant dans un cours. La possibilité de récupérer "sur le vif" les réactions des étudiants devrait être une source intéressante d'améliorations des cours.

d) contrôler son apprentissage : des tests sont prévus, que l'étudiant peut passer quand il le veut. Il peut ainsi obtenir des conseils pédagogiques lui permettant de rationaliser son cheminement dans le cours.

e) Enfin, l'étudiant a la possibilité d'utiliser des programmes "D'AIDE PEDAGOGIQUE" lui permettant d'obtenir, au moment où il le désire :

- de la documentation complémentaire au cours
- des moyens audio-visuels
- des simulations de phénomènes
- des générateurs de problèmes ETC....

Ces programmes "D'AIDE PEDAGOGIQUE" sont mis au point par l'équipe enseignante seule ou en collaboration avec les étudiants. Un exemple actuellement

opérationnel, est constitué par UN GENERATEUR ET ANALYSEUR DE PROBLEMES DE STATIQUE auquel tout étudiant peut poser des problèmes à partir de ses propres données. Le programme lui indiquera :

- soit que ses données sont suffisantes et correctes (et, il fournira les solutions demandées par l'étudiant)
- soit que ses données sont en nombre insuffisant ou incorrectes (et, dans ce cas, il invitera l'étudiant à repenser et à reformuler son problème).

Pour terminer cette introduction, disons notre espoir de pouvoir qualifier "d'intelligent" notre système à partir des critères définis à Leeds par HARTLEY et SLEEMAN /18/. En effet, nous aurons bien :

- \* une représentation de la matière à enseigner
- \* une certaine représentation des étudiants
- \* un ensemble de stratégies pédagogiques possibles
- \* un ensemble de moyens de "guidage" des étudiants dans le domaine d'enseignement (utilisation de SOMINE) ou dans le cours tutoriel.

## CHAPITRE II : DE L'ANALYSE DE LA MATIERE A ENSEIGNER...

### ... A LA STRUCTURATION DES INFORMATIONS DANS SOMINE

#### 2. 1. - INTRODUCTION : ANALYSE DE LA MATIERE A ENSEIGNER :

=====

Un des principaux apports de l'enseignement programmé est d'avoir attiré l'attention des enseignants sur l'organisation de la matière à enseigner. FRANK H. /19/ se propose de "rationnaliser l'enseignement dans tous les domaines" et distingue six "variables" : matière à enseigner, étudiant, milieu d'enseignement, organisation de la matière et moyen d'enseignement. Il estime principale la variable "ORGANISATION DE LA MATIERE" et cherche à construire un système automatique /20/ assurant cette fonction. Les travaux de DAVIES et de son équipe /21/ sont connus dans le monde entier ainsi que les recherches sur les "méthodes d'analyses comportementales" de LE XHUAN en France /22/ et de MECHNER aux Etats-Unis /23/.

Les essais de REGNIER et de MONTMOLLIN /24/ pour tenter de rationaliser l'analyse de la matière sont moins connus. Il nous semble qu'ils ont cependant mis l'accent sur les buts principaux des recherches portant sur l'analyse de la matière à enseigner lorsqu'ils proposent :

- \* de dresser la liste des éléments à enseigner (concepts ou règles)
- \* de rechercher entre ces éléments des relations (qu'ils limitent malheureusement à des relations d'ordre ou de pré-ordre)
- \* d'organiser ces éléments en tenant compte des relations trouvées et de l'expérience de l'équipe enseignante.

On ne peut manquer de rapprocher ce plan des consignes qui guident une personne au moment de la structuration des informations à stocker dans une base de données de type SOMINE /28/. Cette personne doit en effet,

appliquer les trois principes de REGNIER et MONTMOLLIN. Mais, au moment, de l'organisation des éléments en fonction de leurs relations (qui est une des étapes les plus difficiles) elle est GUIDÉE par la structure d'arbre ou de graphe, IMPOSÉE par la banque.

On peut alors se poser la question suivante : une structuration en arbre (ou en graphe) des éléments d'un enseignement serait-elle efficace ? Il semble bien que la réponse soit positive car la construction d'une arborescence montrera à la fois la place de chaque élément et ses relations avec les autres éléments.

2. 2. - LES CRITERES D'ORGANISATION D'UN DOMAINE D'ENSEIGNEMENT :

2. 2. 1. - Liste des éléments et exploration d'un élément :

On s'efforce de décomposer la matière à enseigner en "éléments" aussi "simples" que possible. Chaque élément est une parcelle de l'ensemble des connaissances à explorer, à comprendre, finalement à retenir, et qui forment le "domaine d'enseignement".

Pour nous, "l'exploration d'un élément" peut s'effectuer de deux manières différentes :

\* COMPLETE ET DETAILLEE : Il s'agit de comprendre à fond tout ce qui concerne l'élément en question. La présentation en enseignement programmé est alors faite suivant une succession d'items comprenant par exemple la définition, les propriétés, les applications possibles des connaissances contenues dans l'élément ainsi que les relations de cet élément avec les autres parties du domaine d'enseignement. Chaque item peut contenir un ou plusieurs exercices de contrôle destinés à la fois à vérifier la bonne marche de l'apprentissage et à soutenir l'attention de l'étudiant. Cette méthode d'exploration est, disons, la méthode classique utilisée aussi bien dans des cours de type Skinnérien que dans les cours de type ramifié ( Crowder ).

\* PARTIELLE ET SYNTHETIQUE : Il s'agit de recevoir, au moyen d'un résumé par exemple, une information sur les connaissances essentielles qui sont contenues dans un élément donné, de rechercher les relations de cet élément avec les autres éléments du domaine ou de connaître immédiatement les applications de l'élément.

Des considérations précédentes, on peut déduire qu'il est intéressant dans un cours quelconque, et notamment dans un cours rédigé à

partir des techniques de l'enseignement programmé, de donner aux étudiants la possibilité d'explorer les connaissances contenues dans un élément de plusieurs façons. Par exemple, il pourra avoir accès :

\* soit à un "TEXTE" (ensemble d'items avec exercices) permettant la compréhension complète et détaillée des connaissances.

\* soit à un "RESUME" (sans exercices) pour obtenir rapidement les informations essentielles sur l'élément du cours, sous forme synthétique.

Cependant, une étude de l'apprentissage nous montre qu'il ne faut pas, en général, considérer l'élément de façon isolée.

### 2. 2. 2. - Comment apprend - on ?

La plupart des auteurs déjà cités en /19/, /20/, /21/ à /24/, auxquels il faut ajouter PERRIAULT /25/ et LANDA /26/ et /27/, indiquent quatre principales méthodes d'apprentissage :

a) Apprentissage ELEMENT par ELEMENT : Cette méthode utilise l'exploration "complète et détaillée" définie ci-dessus. On étudie une parcelle de connaissance, puis une autre, .... etc.



b) Apprentissage par COMPARAISON ENTRE ELEMENTS : cette comparaison est souvent :

\* soit une recherche de SIMILITUDE. On essaie de comprendre ce qu'il y a de COMMUN à deux ou plusieurs éléments.

\* soit une recherche de DIFFERENCE. On essaie de comprendre pourquoi deux éléments ont été DISTINGUES en comparant leurs propriétés, leurs applications.... etc.

c) Apprentissage par la recherche de LOIS. Entre les éléments d'un domaine d'enseignement, on peut rechercher les lois de CAUSALITE, d'ANALOGIE LOGIQUE, les liens de PARENTES , etc... qui relient les éléments.

d) Apprentissage par consultation de documents annexes, par visualisation de films ou de diapositives à partir de la résolution ou de

l'analyse de problèmes, de la simulation d'un phénomène , etc...

On peut dire que dans la plupart des domaines d'enseignement ces quatre modes d'apprentissage se retrouvent avec plus ou moins d'importance relative. D'où la nécessité de tenir compte de ces quatre modes lorsqu'on projette un système d'enseignement.

Enfin, à partir des modes d'apprentissage, on peut déduire les critères d'organisation des domaines d'enseignement. C'est l'objet du prochain paragraphe.

2. 2. 3. - Critères d'organisation et fonctions d'enseignement :

A partir de ce qui précède, il paraît possible de déduire un certain nombre de critères pour classer les éléments d'un domaine d'enseignement et de dégager un ensemble de fonctions d'enseignement que tout système d'apprentissage pourrait utiliser :

a) l'apprentissage ELEMENT PAR ELEMENT conduit au critère suivant : " Un élément A doit être enseigné avant un autre élément B si les connaissances contenues dans A sont nécessaires pour aborder les connaissances contenues dans B" ( J. REGNIER, M. de MONTMOLIN /24/).

b) l'apprentissage par COMPARAISON :

\* la recherche des SIMILITUDES conduit à la recherche de l'ASCENDANT (ancêtre commun) des deux éléments concernés, c'est à dire à l'élément qui les CONTIENT ou les UTILISE tous les deux. Par exemple, une recherche de similitude sur les variables et les constantes (au sens informatique) pourra conduire au résultat : " ces deux éléments sont utilisables dans les expressions arithmétiques".

\* la recherche des DIFFERENCES conduit à la nécessité de pouvoir comparer les PROPRIETES ESSENTIELLES de deux éléments quelconques du cours. Ce résultat pourra ( par exemple) être obtenu par la comparaison des RESUMES de deux éléments.

c) recherche de LOIS :

L'équipe enseignante prévoiera les relations de causalité (ou autres) entre

les éléments et les fonctions permettant aux étudiants de rechercher ces relations.

d) Aide à la documentation complémentaire, à l'information au moyen de systèmes audio-visuels, possibilité de simuler des phénomènes seront données aux étudiants par des programmes complémentaires " D'AIDE PEDAGOGIQUE".

EN RESUME : Il existe de nombreux modèles d'apprentissage. Pour mettre en place celui que nous avons retenu (et progressivement défini ci-dessus), il faut assurer les FONCTIONS SUIVANTES :

\* fonction "sommaire" : donne aux étudiants la liste des éléments

\* fonction "exploration complète" d'un élément. : permet à l'étudiant d'interroger un "texte" présentant tous les aspects de l'élément.

\* fonction "résumé" : donne le résumé des connaissances essentielles contenues dans un élément.

\* fonction "recherche de similitude" : doit mettre en évidence les points communs entre deux éléments quelconque du domaine d'enseignement.

\* fonction "recherche des différences" doit assurer la mise en évidence des différences entre deux éléments quelconque du domaine d'enseignement.

\* fonction "feed-back" : donnera à l'étudiant la possibilité d'inscrire ses remarques, ses idées d'amélioration et ses critiques au niveau de chacun des éléments.

\* fonction "aide à l'enseignement" élargira les possibilités ci-dessus en donnant aux étudiants d'autres informations sous des formes variées : diapositives, films, etc...

## 2. 3. - STRUCTURATION DES INFORMATIONS D'UN DOMAINE D'ENSEIGNEMENT :

Une structure arborescente permet une excellente visualisation des relations d'ordre (nécessaires pour l'apprentissage élément par élément). Elle montre aussi clairement les relations de PARENTE entre éléments (nécessaires pour les recherches de similitude). Les différences entre éléments y sont visibles et elle peut servir de support pour la mise en place des autres relations (causalité, etc....)

Pour toutes ces raisons, il nous semble qu'elle peut aider dans l'analyse des matières à enseigner.

Ainsi, le schéma ci-après ( figure 1) montre clairement :

- \* Qu'il est souhaitable d'explorer les éléments dans l'ordre de la structure arborescente, en la parcourant des feuilles vers la racine. Cela correspond, dans notre exemple, à un parcours dans l'ordre alphabétique.
- \* Que les propriétés communes ( relation de similitude) à A et B viennent du fait qu'ils sont des "descendants" de l'élément C.
- \* Que chacun des éléments est DIFFERENT d'un autre : les résumés de chaque élément feront apparaître ces différences.
- \* Que la recherche des relations de causalité entre tous les éléments du domaine d'enseignement obtiendra une réponse négative, sauf pour les éléments H et C liés par une telle relation.

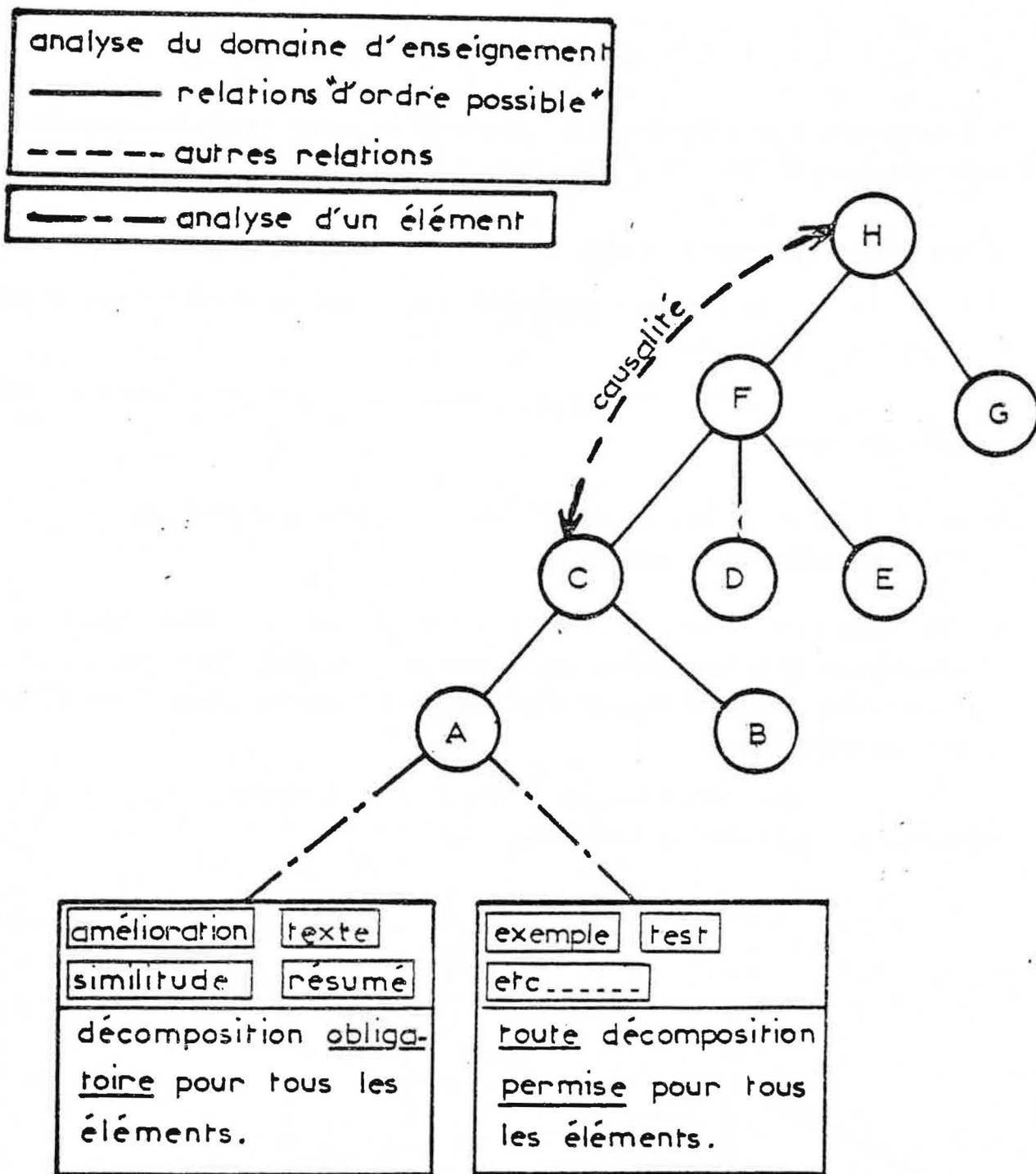


figure 1 : analyse d'un domaine d'enseignement

2. 4. - STOCKAGE DES INFORMATIONS DANS SOMINE :

2. 4. 1. - Rappel sur SOMINE :

On pourra lire à ce sujet les deux premières parties du présent mémoire ainsi que /28/ et /29/. Pour notre propos, les passages essentiels sont :

- a) Une base d'information somine est formée à partir de deux fichiers :
  - le fichier "structure" dans lequel on écrit la structure formelle des informations
  - le fichier "données" dans lequel sont effectivement contenues les informations.
- b) De la structure formelle sont déduits les ORDRES de MISE A JOUR et d'INTERROGATION des informations
- c) Ces ordres sont rédigés en langage SOMINE qui est un langage "quasi naturel" Ils peuvent être inclus dans des programmes (FORTRAN, COBOL etc...) ou directement écrit sur écrans alphanumériques lorsqu'on désire travailler en mode CONVERSATIONNEL

Pour illustrer les points a, b, c ci-dessus, la figure 2 reproduit une structure formelle très simple.

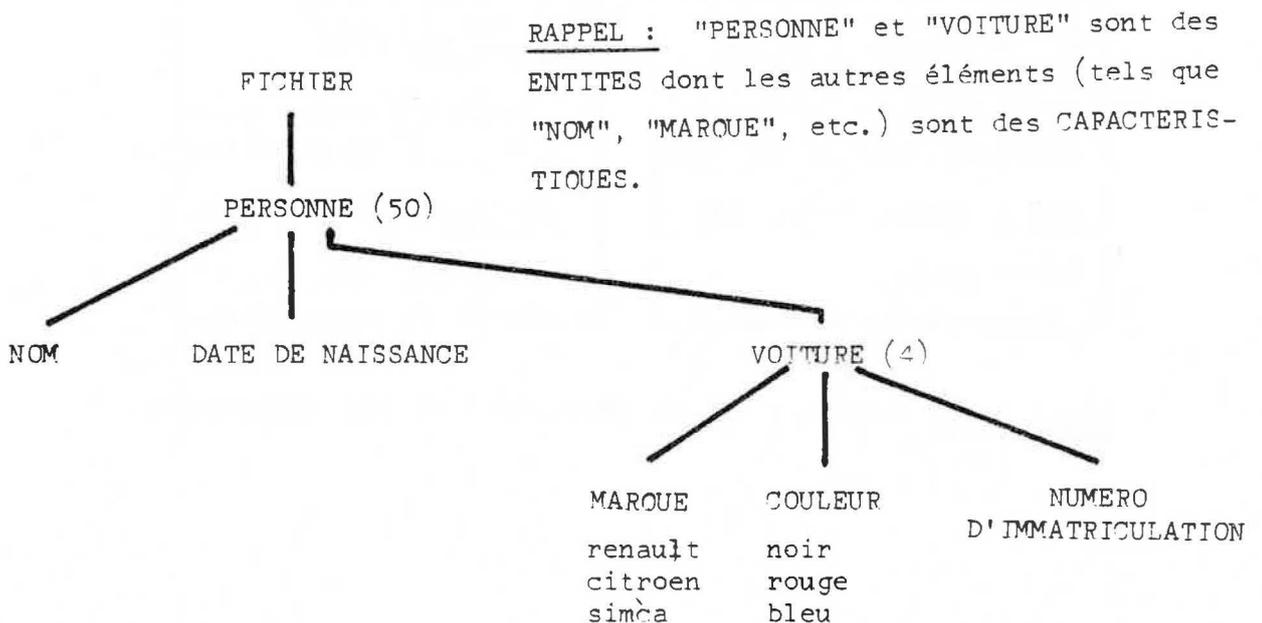


Figure 2 : exemple de structure formelle SOMINE

ne se servirait pas d'une des propriétés fondamentales de SOMINE : la possibilité de gérer des ENTITES.

Nous avons longuement réfléchi sur ce problème. On peut d'ailleurs le généraliser et se demander s'il est possible de DEDUIRE des questions souhaitées par l'utilisateur, la MEILLEURE structuration des informations. Cette question est discutée à propos de chaque application et notamment par MARTY /29/.

Pour ce qui nous concerne, nous avons finalement trouvé les critères suivants :

- les ordres de mises à jour et surtout d'interrogation doivent être simples et très condensés.

- les temps de réponses doivent être les plus courts possibles (travail en mode conversationnel)

- une numérotation des éléments serait intéressante, car on pourrait distribuer un sommaire à l'étudiant et il est plus court d'écrire et d'analyser des nombres que des mots.

A partir de là, et de la volonté de donner à l'équipe enseignante un outil souple, adaptable à chaque domaine d'enseignement, nous avons choisi une structure type représentée sur la figure 3.

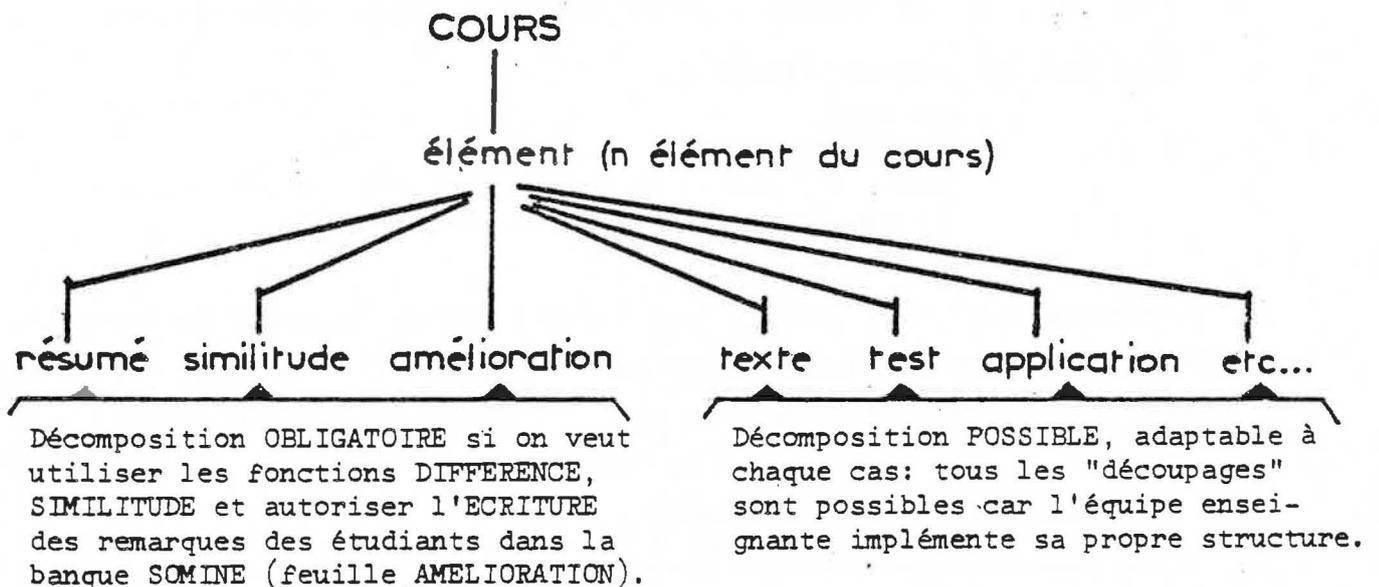


Figure 3 : structure formelle d'enseignement

Voici maintenant des ORDRES (de mise à jour -M- ou d'interrogations -I-) qu'il est possible de formuler à partir de cette structure :

M	NOM DE LA PERSONNE	48 = ARNAUD	#
M	N° d'immatriculation DE LA VOITURE 2 DE LA PERSONNE	37 = 1036FF42	#
I	DATE DE NAISSANCE DE LA PERSONNE	7	#
I	COULEUR DE LA VOITURE 1 DE LA PERSONNE	14	#

Naturellement, les mises à jour doivent précéder les interrogations.

#### 2. 4. 2. - Conséquences sur l'application à l'enseignement :

a) La simplicité du langage de mise à jour et d'interrogation de SOMINE est un avantage primordial pour son adaptation à l'enseignement. On pourra en effet :

##### . du côté de l'équipe enseignante :

- \* écrire et modifier les cours en mode conversationnel ou en batch. Pour cela, de simples "mises à jour" seront suffisantes.
- \* Retrouver les remarques des étudiants dans les "feuilles amélioration" de chaque élément. Ce seront des interrogations simples.

##### . du côté étudiant :

- \* écrire dans les "feuilles amélioration" : simples mises à jour.
- \* interroger pour un élément :
  - son texte
  - son résumé
  - toutes les autres décompositions prévues par l'équipe enseignante ( test, application .... etc...)

b) Les informations étant situées dans les FEUILLES de la structure formelle, les questions et les mises à jour seront réalisées par des ordres d'autant plus LONGS que la structure aura plus de niveaux. Exemple : dans la structure vue sur la figure 1, on pourra demander :

I RESUME DE A DE C DE F DE H	#
------------------------------	---

(RESUME étant une des "feuilles" du "noeud" A)

Or, une arborescence d'analyse d'une matière d'enseignement peut avoir un nombre très grand de niveaux. On n'a donc pas intérêt à l'utiliser directement comme structure formelle dans SOMINE. D'autant plus qu'une telle structure

2. 5. - CONCLUSION :

Au cours de ce chapitre, portant essentiellement sur l'analyse de la matière à enseigner, nous avons montré l'apport à cette question des structures arborescentes (graphes et arbres). Puis, utilisant ce type de structure pour placer les informations relatives à un enseignement dans une base de données SOMINE, nous avons vu que, grâce notamment à la simplicité de son langage de mise à jour et d'interrogation, SOMINE pouvait être le centre d'un outil d'enseignement intéressant, permettant à l'étudiant d'interroger facilement et de points de vue très divers, les informations du cours.

Mais, la possibilité qu'offrent les banques de données d'être mises à jour et interrogées facilement ne suffisent pas pour en faire des outils d'enseignement valables. Car un système d'enseignement doit permettre, en plus :

- D'INTERROGER LES ETUDIANTS et D'ANALYSER leurs réponses afin de leur donner des CONSEILS PEDAGOGIQUES appropriés à leurs démarches.

- DE RECHERCHER les SIMILITUDES, les DIFFERENCES et les AUTRES RELATIONS (causalité, parenté, analogie... etc..) reliant les éléments d'un domaine d'enseignement.

La mise en oeuvre de ces fonctions fait appel à :

\* la création et l'implémentation d'un LANGAGE DE COURS que nous appellerons EPMINE dans la suite de cet exposé,

\* la programmation d'un interpréteur de requêtes des étudiants et de celles de l'équipe enseignante. Cet interpréteur sera appelé : ENSEIGN

Enfin, si on veut élargir le champ des investigations des étudiants, et donner aux enseignants qui s'intéressent à l'E.A.O. d'autres possibilités que "l'expression par le texte" (dans laquelle on se confine trop souvent), il ne faut pas construire un SYSTEME FERME. Ouvrons les portes au contraire ! DONNONS AUX enseignants et étudiants la possibilité de RATTACHER au système prévu TOUS LES PROGRAMMES "D'AIDE PEDAGOGIQUE" qui leur paraîtront utiles : générateurs ou analyseurs de problèmes, programmes de simulations, d'appel à des films, à des diapositives, à des graphiques, à des fichiers ETC.....

Les trois points ci-dessus feront l'objet des développements des chapitres suivants.



### CHAPITRE III : SOMINE CENTRE D'UN SYSTEME D'ENSEIGNEMENT

#### 3. 1. - NECESSITE D'UN LANGAGE DE COURS :

Dans la conclusion du chapitre précédent, nous avons vu qu'un système complet d'enseignement doit notamment :

- fournir des informations à la demande des étudiants et donner à ceux-ci la possibilité d'écrire leurs remarques.

- poser des questions à l'étudiant dans les "textes" de présentation des éléments afin de soutenir son attention ou dans des "tests de contrôle de l'acquisition des connaissances" afin de lui donner des conseils pédagogiques valables.

Poser des questions suppose évidemment ATTENDRE DES REPONSES et pouvoir les ANALYSER. On peut donc dire qu'il faut stocker dans SOMINE deux sortes d'informations :

- des informations simples (en langage naturel) d'abord : ainsi pourront être écrits, par exemple, les résumés, les commentaires sur les relations de similitude ou de causalité ..... etc.

- des informations plus complexes ensuite contenant des QUESTIONS aux étudiants, et, par exemple, les réponses exactes possibles ainsi que les aiguillages et commentaires à fournir après comparaison des réponses exactes et des réponses réelles des étudiants. Ces informations ne pourront pas être stockées directement en langage naturel. Elles seront rédigées à l'aide d'un Langage de cours (EPMINE) dont les fonctions et propriétés essentielles sont développées ci-après.

#### 3.2. - PRESENTATION DU LANGAGE DE COURS "EPMINE" :

Ce langage a été construit à partir de deux impératifs principaux :

- simplicité et souplesse d'utilisation
- possibilité de perfectionnement par adjonction de nouveaux ordres grâce à la programmation modulaire de l'interpréteur.

Partis des langages de cours utilisés à Lyon /8/ et à Grenoble /7/ que nous connaissions, nous avons cherché à créer un outil expérimental, facile à implémenter sur tous les ordinateurs et dont l'apprentissage soit rapide.

Tous les caractères utilisés devaient se trouver sur la console de l'écran alphanumérique de l'Ecole des Mines de St.Etienne (écrans alphanumériques PHILIPS). Les principaux éléments de ce langage et les fonctions qu'ils remplissent sont expliqués sur la figure 4 de la page suivante. A l'annexe II à cete partie du mémoire, on trouvera un exemple de "leçon" programmée à l'aide du langage EPMINE et qui est utilisée dans un essai de rééducation des dyslexiques (enfants de 10 à 14 ans).

3. 2. 1. - Exemple de séquence de cours :

```

EPMI<<<DEBUT>>
LES VARIABLES SONT DES "MOTS" COMMENCANT PAR UNE LETTRE ET FORMES DE 6 CA
RACTERES ALPHANUMERIQUES AU PLUS. CELLES DONT LA LETTRE DE DEBUT EST I,J,K,L,M,N
SONT EN MODE FIXE (ENTIER). LES AUTRES SONT EN MODE REFI.
LES VARIABLES SONT UTILISEES COMME VARIABLES D' AFFECTATION (A GAUCHE DE=
) OU DANS DES EXPRESSIONS ARITHMETIQUES (A DROITE DE =).
RECOPIEZ LES VARIABLES CORRECTEMENT ECRITES DANS L' ORDRE DE LA LISTE SUI
VANTE, EN LES SEPARANT PAR DES VIRGULES: EPHIN,5IT,RL AN,NORS,XMASSÉS. >>20<<EPM
IN,NORS>> E<<T.B.>>A<<P>>;IN<<COI=COI+01>> SI<<COI=04>> F<<EPMIN ET NORS SONT SE
ULES CORRECTES: 5IT COMMENCE PAR UN CHIFFRE, EL AN CONTIENT UN...BLANC ET XMASSÉS
A TROP DE CARACTERES.TRANSMIT S.V.P.>>? T<<5IT>> E<<NON,UNE VARIABLE DOIT COMME
NCER PAR UNE LETTRE.TRANSMIT S.V.P.>> ?A<<DEBUT>>;T<<PLAN>>F<<UN BLANC N' EST PA
S UN CARACTERE ALPHANUMERIQUE.>> A<<DEBUT>>;I<<XMAS>>E<<ATTENTION: 6 CARACTERES
MAXI. TRANMIT S.V.P.>> ? A<<DEBUT>>; E<< REPONNEZ VOTRE REPONSE APRES AVOIR RELU
LE QUESTION.>> A<<DEBUT>> <<95>>
    
```

3. 2. 2. - Quelques remarques :

- a) Après une comparaison positive concernant les ordres C, T ou S, on continue en SEQUENCE jusqu'au prochain A (ALLER A), suivi d'une étiquette. Si la comparaison est négative, on cherche le premier (;) et on repart en séquence à partir de ce symbole.
- b) Actuellement, il n'y a pas encore d'analyseur lexico-syntaxique de ce langage de cours. Le cours est placé soit dans un fichier séquentiel à accès direct si on veut l'utiliser sans la banque, soit dans la banque SOMINE. Pendant l'exécution, le cours est analysé par un INTERPRETEUR.

ORDRE	CODE	FONCTION PEDAGOGIQUE	EXEMPLE D'UTILISATION-COMMENTAIRE
ECRIRE	E	Ecrire un texte de 1280 caractères au plus sur un terminal étudiant.	E «VOILA CE QUE JE VEUX ECRIRE» E «TRES BIEN» (le texte situé entre « et » s'écrira sur le terminal).
LIRE ATTENDRE	?	"Donner la main " à l'étudiant afin qu'il réponde à une question	E «SIX MULTIPLIE PAR SIX EGALE ? » ? (remarquer que l'ordre ? est situé à l'extérieur d'une chaîne comprise entre « et » ).
Analyse de réponse : COMPARER et TROUVER	C	Ordre de comparaison STRICTE entre la réponse PREVUE et celle de l'élève	E «COMBIEN FONT 3 * 5 » ? C «15» E «TRES BIEN» A «6» ; E «NON» A «ERREUR15»
	T	Ordre de comparer la réponse de l'élève à un SQUELETTE DE LA REPONSE PREVUE	E «QUELLE EST LA CAPITALE DE LA FRANCE ? » ? T «PRS » E «OUI» A « ALLEMAGNE» ; T «LONDRES» E « NON LONDRES EST LA CAPITALE DE L'ANGLETERRE.» T «LYON» ..... etc.
Branchements: ALLER A et SI	A	BRANCHEMENT inconditionnel (analogue à GO TO de FORTRAN)	A «10» conduira à l'étiquette «#10» A «FRANCE» conduira à l'étiquette «#FRANCE»
	S	BRANCHEMENT conditionnel (suivant la valeur d'un compteur)	S «CO4 = 07» A « AVERTIR» ; A «SUITE» (si le compteur CO4 a la valeur 07, on ira à l'étiquette «#AVERTIR» sinon, on se branchera à l'étiquette «#SUITE»

Figure 4 : présentation du langage EPMINE

ORDRE	CODE	FONCTION PEDAGOGIQUE	EXEMPLE D'UTILISATION-COMMENTAIRE
SINON	;	CARACTERE à partir duquel il faut reprendre le cours après une comparaison NEGATIVE	S «CO3 = 02» A «2»;A «3» C «JEUDI» A «JOUR»; A «NUIT» T «RPS» E «T.B.» A «4» ; A «3»
DEBUT D'ETIQUETTE	#	POINT DE BRANCHEMENT DANS UNE LECON	Une étiquette contient DIX caractères au plus. EXEMPLE :  «#PARIS» «#2» «#DEBUT» (étiquette de début de cours doit être précédée de EPMI si le cours est dans une base SOMINE)  «# \$» (étiquette de fin de cours)  Nota : ces deux étiquettes sont obligatoires.
INCREMENTER UN COMPTEUR	I	SERT A MODIFIER LA VALEUR D'UN COMPTEUR.	Il y a 99 COMPTEURS qui sont initialisés à ZERO au début d'une leçon. <u>exemples:</u> I «C04=C04-12» (la valeur de C04 est diminuée de 12) I «C87=C87+01» (la valeur de C87 est augmentée de 1)
COMMENTAIRE BANALISE	K	SERT A ECRIRE SUR LE TERMINAL DES COMMENTAIRES BANALISES (remplace donc un ordre ECRIRE)	On peut utiliser 99 commentaires banalisés dans une leçon. <u>exemples :</u> K «10» enverra sur le terminal le commentaire banalisé numéro 10 qui pourra être : "TRES BIEN, CONTINUEZ AINSI".

figure 4: suite de la présentation de EPMINE

- c) A tout moment l'étudiant peut, en répondant "CHOIX" à une question quelconque, choisir à nouveau entre la BANQUE, le COURS de type TUTORIEL, les programmes d'AIDE A L'ENSEIGNEMENT ( s'il y en a ) ou la FIN DU TRAVAIL.

### 3. 3. - MISE A JOUR ET MODIFICATION DES INFORMATIONS :

Le cours est entré soit :

- par CARTES s'il doit être stocké dans un fichier (cours tutoriel)
- par CARTES OU A PARTIR DES ECRANS; et en suivant la structure choisie par l'équipe enseignante, lorsqu'il s'agit d'un stockage dans la banque. Dans ce cas, des corrections sont possibles moyennant certaines précautions au moment de la première "mise à jour" du cours.

Dans tous les cas, le cours est écrit en format A4, les blancs non significatifs, c'est à dire situés en dehors des chaines (étiquettes, chaines compteur et commentaires ou chaines à écrire et à comparer) sont compactés. Ainsi, la mise en page est plus aisée.

### 3. 4. - INTERROGATION DES INFORMATIONS :

- a) Si le cours est sur fichier séquentiel, on peut choisir l'étiquette à partir de laquelle on veut travailler. C'est la seule "question" que l'on peut poser.

- b) Si le cours est dans la banque, il faut distinguer :

- l'interrogation d'une caractéristique d'un élément : elle est faite directement en langage SOMINE ; par exemple, on trouvera :

```
I RESUME DE L'ELEMENT 17 ##
```

le résumé de l'élément 17 apparaîtra sur l'écran.

- l'interrogation porte sur une FONCTION faisant intervenir plusieurs éléments. Par exemple :

```
I SIMILITUDE ENTRE 121 ET 124#
```

Cette requête sera analysée par un programme spécial car elle n'est pas en langage SOMINE. Le résultat apparaîtra également sur l'écran.

- c) Langage de cours et langage naturel :

Que l'interrogation porte sur un seul élément ou fasse intervenir une

fonction de deux éléments, la recherche dans la banque a pour résultat une caractéristique de type "texte" qui peut être :

- soit en langage naturel : il suffit alors d'envoyer l'information sur l'écran
- soit en langage EPMINE : il faut alors faire appel à l'interpréteur de ce langage.

Ces deux cas sont distingués à l'aide du premier mot du texte stocké dans la banque : s'il contient "EPMI", le texte est en langage de cours. Il est en langage naturel dans tous les autres cas.

On comprend alors qu'il appartient à l'équipe enseignante ET A ELLE SEULE ( sans aucune contrainte du système) de choisir les parties du cours dans lesquelles elle veut :

- pouvoir interroger les étudiants : elles seront alors en langage de cours,
- donner seulement des informations aux étudiants : rédaction en langage naturel.

Une même caractéristique de deux éléments différents peut très bien, par exemple, être en langage naturel pour un des éléments et en langage de cours pour l'autre. On peut aussi, comme dans l'exemple suivant (figure 5) adopter que toutes les caractéristiques "matière" et "tests" seront en langage de cours et toutes les autres en langage naturel :

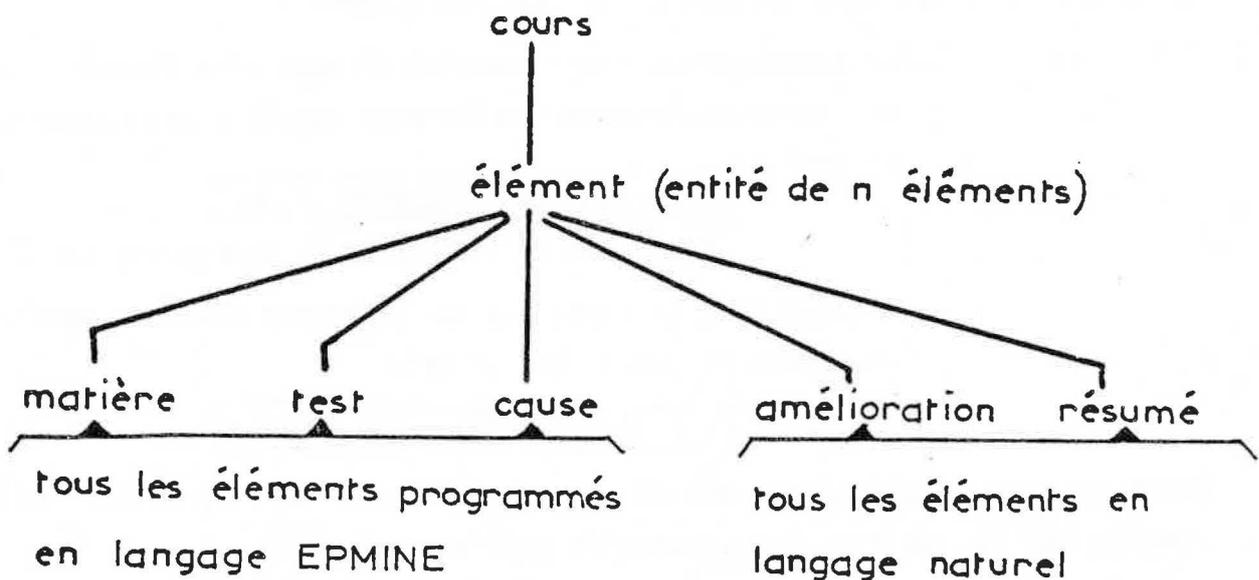


Figure 5 : exemple de structure formelle de cours

Pour cet exemple , une interrogation : I MATIERE DE L'ELEMENT 27 #

donnera des informations, des questions, des conseils pédagogiques, etc..

Au contraire, une interrogation : I RESUME DE L'ELEMENT 56 #

aura pour effet la publication d'un simple texte.

### 3. 5. - RECHERCHE DES RELATIONS ENTRE ELEMENTS :

Toutes les interrogations portant sur UN SEUL élément ne posent pas de problèmes. Par exemple, si on veut rendre possible la recherche de la CAUSE d'un élément, il suffit de prévoir la caractéristique CAUSE dans la structure SOMINE et d'interroger :

I CAUSE DE L'ELEMENT 112 #

- Si l'élément 112 n'est lié à aucun autre par une relation de causalité, on sortira le texte situé dans "cause" et expliquant pourquoi la relation demandée n'existe pas (car, s'il a écrit cette requête, c'est sans doute que l'étudiant croit qu'elle existe).

- Si l'élément 112 est lié à d'autres éléments par cette relation, le texte donnera les informations permettant de comprendre ces liens.

Par contre, toutes les requêtes portant sur DEUX éléments ne sont pas résolues par le langage SOMINE et demandent plus d'explications. Il en est ainsi pour les fonctions DIFFERENCE et SIMILITUDE expliquées dans les paragraphes suivants.

#### 3. 5. 1. - Recherches des différences entre deux éléments :

L'étudiant écrit sur l'écran :

I DIFFERENCE ENTRE L'ELEMENT 187 et L'ELEMENT 197 #

Le programme EAO interprète cette requête et recherche :

- le résumé de l'élément 187 qu'il publie en haut de l'écran
- le résumé de l'élément 197 qu'il publie dans la deuxième moitié de l'écran.

Cela présente l'inconvénient de limiter à 160 mots maxi la longueur des résumés mais a pour avantage de permettre une comparaison efficace entre les propriétés essentielles de chacun des deux éléments.

L'algorithme de recherche des différences est représenté sur la figure 6

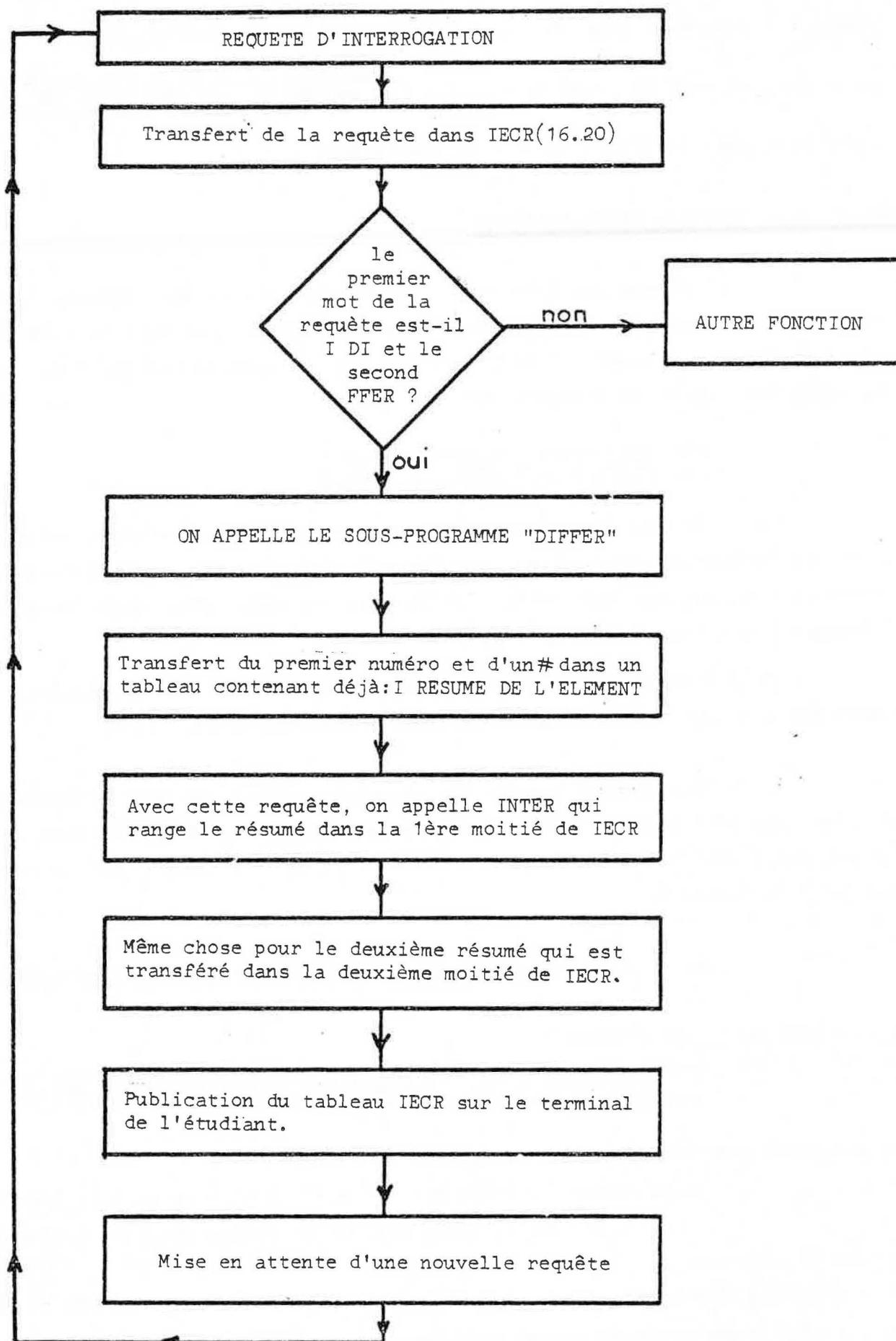


Figure 6 : algorithme de recherche des différences

3. 5. 2. - Recherche de similitude :3.5.2.1. - Place de la similitude dans la structure formelle :

Alors que les "pages" contenant les informations relatives à un élément sont TOUTES identifiables à partir du numéro de cet élément et du titre de l'information (matière, résumé, etc...) alors que la recherche des DIFFERENCES fait appel à deux "pages" de numéro et de titre connus, il en va tout autrement pour les recherches de similitudes. Prenons l'exemple d'une partie d'arbre d'analyse de la matière d'un cours de FORTRAN (figure 7).

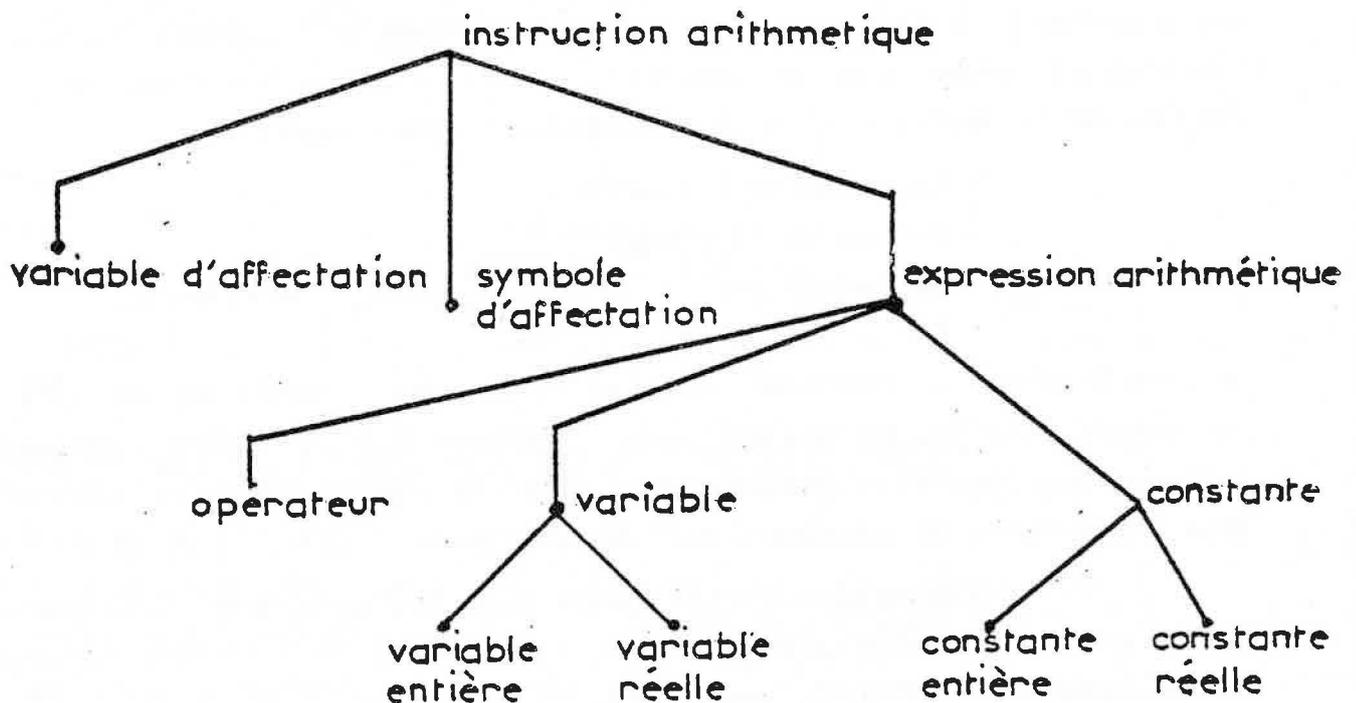


figure 7 : structure d'une séquence de cours

Dans cette séquence de cours, si on demande la similitude entre une constante réelle et une constante entière, on comprend que la réponse fera appel au fait que ce sont TOUTES LES DEUX DES CONSTANTES. Cette réponse sera donc située au niveau de l'ANCETRE COMMUN AU DEUX ELEMENTS concernés. Si on pose maintenant la question :

I SIMILITUDE ENTRE CONSTANTE ET OPERATEUR #

On comprend qu'il n'y a pas d'autres ressemblances que celles venant de leur utilisation commune dans les "EXPRESSIONS ARITHMETIQUES", ancêtre commun des deux éléments concernés.

I SIMILITUDE ENTRE 131 ET 1331 #

sera facilement transformée par un sous-programme en :

I SIMILITUDE DE L'ELEMENT 13 #

et, dans la page correspondante de la banque, on pourra, par exemple, trouver le texte suivant (cours de fortran, sommaire ci-dessus) :

SIMILITUDE

Les OPERATEURS, les VARIABLES et les CONSTANTES (en mode fixe ou réel) ont pour point commun d'être utilisés ensemble dans les EXPRESSIONS ARITHMETIQUES dont voici un exemple :

3.14 \*RAYON\*\*2 \* HAUT / 3.

remarques :

Un texte de "similitude" doit évidemment parler de tous les descendants de "l'ancêtre commun" auprès duquel il est placé.

Il faut penser, par exemple, qu'une similitude placée au niveau de l'élément 13 pourra être retrouvée à partir d'une requête portant sur les éléments 131 et 132, mais aussi à partir des éléments 1331 et 132507 s'il existe !

Un texte de similitude est d'autant plus intéressant qu'il compare des éléments VOISINS dans la structure d'analyse de la matière. Il est d'autant plus flou qu'il concerne des éléments éloignés dans cette structure.

3. 6. - CONCLUSION :

Assisté par un interpréteur de langage de cours (EPMINE) et placé sous les ordres d'un programme (ENSEIGN) qui gère l'interface avec les étudiants utilisateurs, SOMINE apparaît bien comme le "centre" d'un système d'enseignement dont l'organigramme général est conforme à la figure 9 (page suivante).

Les éléments du cours dans l'entité "élément" de la structure SEMINE sont maintenant bien définis. On a vu également sous quelle forme ils seront remis aux étudiants en même temps que la structure servant aux interrogations et aux mises à jour de la caractéristique "AMELIORATION".

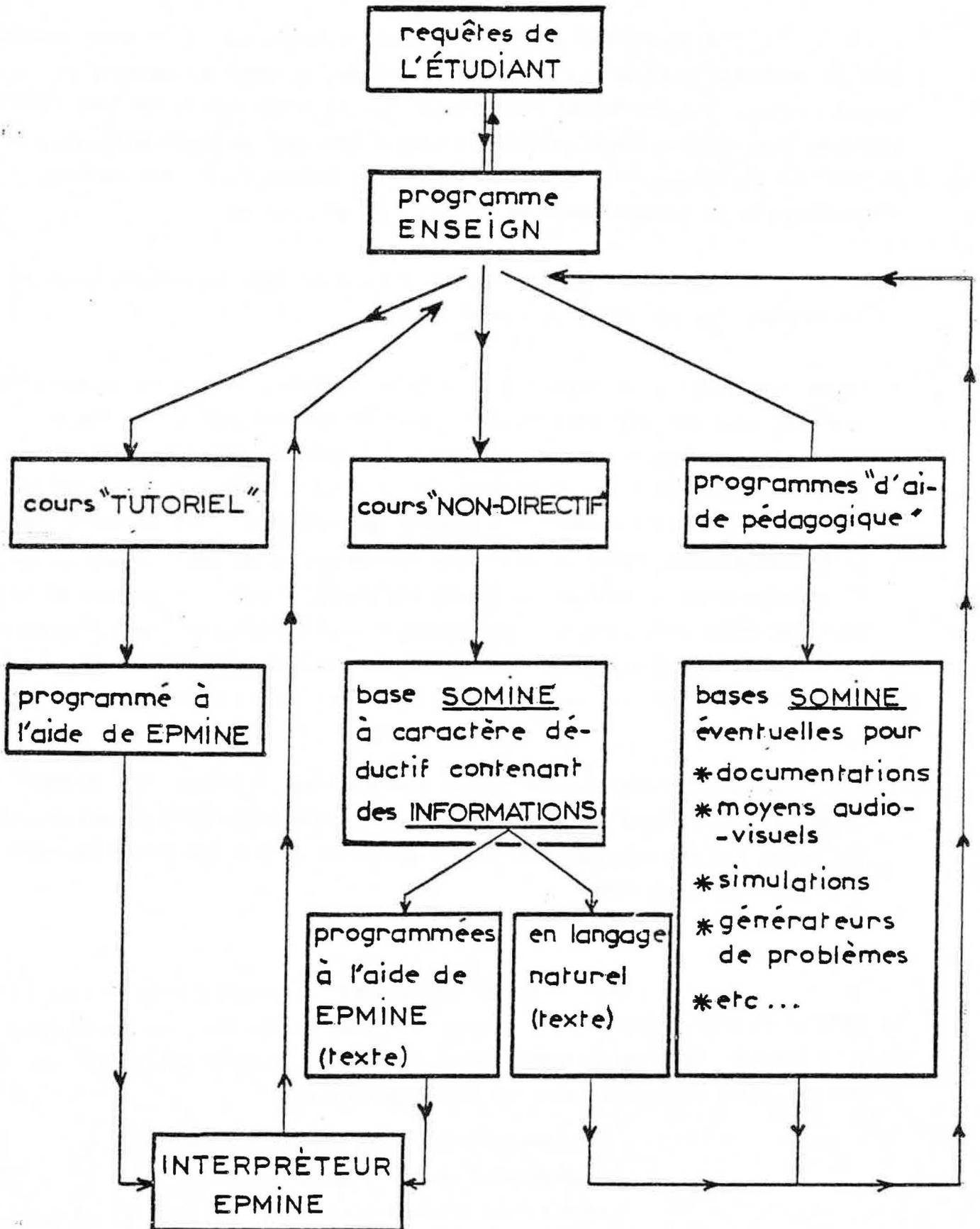


figure 9 : SOMINE et le système d'enseignement

A propos des numéros, on peut craindre que la méthode exposée utilise beaucoup de place sur les disques puisque de nombreux numéros ne seront pas utilisés. Ce défaut n'existe pas car ce n'est que lorsqu'une caractéristique d'un élément d'une entité est MISE A JOUR que sa place physique est retenue sur le disque du fichier donné. Or, les éléments dont les numéros n'existent pas ne seront jamais mis à jour, ni même créés.

Enfin, il nous semble que deux avantages importants viennent de la construction modulaire de notre système :

- a) On peut "prévoir " et commencer à réaliser morceau par morceau ce que l'on a prévu. Cela est vrai au niveau du cours où on peut prévoir un cours tutoriel classique et un cours non directif puis ne réaliser, dans un premier temps, que l'une ou l'autre de ces formes. Cela est vrai pour le cours non directif lui-même : on prévoit une structure pour un vaste domaine d'enseignement, formé de centaines d'éléments et on peut commencer par l'implémentation de quelques éléments seulement, formant une partie de cours. Cela est enfin vrai pour les "programmes d'aide pédagogique" que l'on peut ajouter facilement au moment où leur besoin est ressenti. Leur programmation peut être réalisée par les étudiants eux-mêmes dans certains cas.
- b) Plusieurs bases d'information SOMINE peuvent être utilisées SIMULTANEMENT à partir du programme "CHOIX". L'une d'elle contiendra les informations sur les éléments, une ou plusieurs autres seront utilisées par les programmes d'aide à l'enseignement.

Afin de tester rapidement les possibilités de tout le système et pour commencer les recherches pédagogiques dont nous avons parlé dans le premier chapitre de cette partie du mémoire, nous avons décidé de mettre en oeuvre trois applications très différentes :

- enseignement des bases du langage FORTRAN
- enseignement de la mécanique rationnelle
- rééducation d'enfants ayant des difficultés en lecture ou en écriture (dyslexie).

FORTRAN a été choisi à cause de l'environnement favorable que constitue toujours un centre informatique pour l'enseignement d'un langage de programmation. L'analyse de la matière et les premiers résultats obtenus seront développés dans le chapitre suivant.

La mécanique rationnelle a été choisie parce que son enseignement est inconcevable sans PROGRAMMES D'AIDE PEDAGOGIQUES. Aussi, dans le chapitre V, traitant de cette application, nous insisterons particulièrement sur les idées et les réalisations dans cette direction, essayant de résumer les questions qu'elle pose et les réponses déjà connues. Nous présenterons ainsi un analyseur de problèmes et commenterons brièvement son fonctionnement et ses résultats.

Enfin, la rééducation des dyslexiques nous a semblé intéressante pour savoir si, véritablement, l'E.A.O. est capable de modifier de façon profonde et durable certains comportements. Les élèves, dont on aurait mobiliser l'attention de manière intense pendant des séances de rééducation au moyen de terminaux d'ordinateur seraient-ils capable de plus d'attention dans la vie courante (classe ou famille) ? Cette question et quelques autres sont à l'origine du travail entrepris avec J. FOURNEL, instituteur spécialisé dans la rééducation des dyslexiques et dont nous parlerons dans le chapitre VI.



## CHAPITRE IV : ENSEIGNEMENT DES BASES DU LANGAGE FORTRAN

Apprendre un langage de programmation à partir d'un système d'enseignement assisté par ordinateur présente un avantage majeur : on utilise au cours de l'apprentissage le matériel (télétype, écran) que l'on utilisera après l'apprentissage. On gagne ainsi du temps et de l'assurance. L'environnement aussi est important : souvent, l'ordinateur n'est pas loin, on connaît le matériel et c'est plus important qu'il n'y paraît.

Apprendre un LANGAGE, c'est feuilleter une sorte de dictionnaire : les relations entre les éléments sont peu importantes et il faut plus APPRENDRE que COMPRENDRE. Mais la MATIERE est vaste. Il ne sera donc pas indifférent d'utiliser notre système en prévoyant une vaste structure d'analyse de la matière de façon à couvrir l'ensemble des informations nécessaires, quitte à ne stocker effectivement ces informations qu'au fur et à mesure des possibilités. N'oublions pas qu'il faut une centaine d'heures de travail en équipe pour obtenir une heure de travail valable d'étudiant en enseignement assisté par ordinateur !

Enfin, dans l'état actuel de notre réflexion, une "AIDE-PEDAGOGIQUE" par des programmes de documentation, de simulation etc... n'apparaît pas nécessaire. Dès qu'ils commenceront à programmer, les étudiants soumettront leurs travaux aux critiques du ... compilateur FORTRAN. Celui-ci signalera les erreurs lexico-syntaxiques.... Les autres ne concernent pas l'apprentissage du langage de programmation.

### 4. 1. - LES GRANDE OPTIONS :

Finalement, nous avons décidé :

- de stocker dans une base somine les informations nécessaires à la compréhension et à l'utilisation de tous les ordres FORTRAN de base. Nous les avons décomposés en :

.../...

- \* INSTRUCTIONS D'ENTREES/SORTIES
- \* ..... concernant les sous-programmes
- \* ..... arithmétiques
- \* ..... concernant les boucles
- \* ..... de branchement

- de rédiger un cours de forme "tutorielle" pour les étudiants qui seraient gênés par le trop grand nombre d'initiatives que leur laisse le cours non directif utilisant la base SOMINE.

Le passage d'une méthode à l'autre, d'apprentissage est possible à tout moment. Il suffit en effet, de répondre " CHOIX" lorsqu'on a "la main" (c'est à dire après n'importe quelle question de cours). On obtient alors la question suivante :

ECRIVEZ VOTRE REPONSE EN COMMENCANT PAR LE PREMIER OCTET DE L'ECRAN ET EN LA TERMINANT PAR # .  
 VOULEZ-VOUS "COURS" (1), " BANK" (2), " AIDES" (3) OU " FIN" (4) ?  
 REPONDEZ PAR UN DE CES CHIFFRES SUIVI DE # .

La réponse 4 entraîne la FIN DU TRAVAIL

La réponse 3 donnerait à l'étudiant la possibilité d'obtenir un des programmes d'aide pédagogique. Ici, il obtient seulement un message lui indiquant qu'il n'y a pas de programme d'aide à l'apprentissage du FORTRAN et l'invitant à rédiger lui-même des programmes.

La réponse 2 donne le cours stocké dans la banque. L'étudiant doit alors formuler une requête en fonction du sommaire qui lui a été remis.

La réponse 1 met l'étudiant en communication avec le cours de type "tutoriel classique". Le message suivant apparait :

A QUELLE ETIQUETTE VOULEZ -VOUS ALLER ?  
 REPONDEZ PAR LE NOM DE L'ETIQUETTE SUIVI DU CARACTERE # .  
 LA PREMIERE ETIQUETTE DU COURS EST : DEBUT

On répondra "DEBUT" si on suit le cours pour la première fois. Dans les autres cas, on pourra entrer dans le cours à une place choisie (par exemple, à la dernière ETIQUETTE que l'on aura vue pendant la séance précédente.

4. 2. - COURS STOCKE DANS LA BANQUE SOMINE :4. 2. 1. - Analyse de la matière :

Passons sur la décomposition en éléments et présentons tout de suite la structure d'analyse de la matière à enseigner (figure 18)

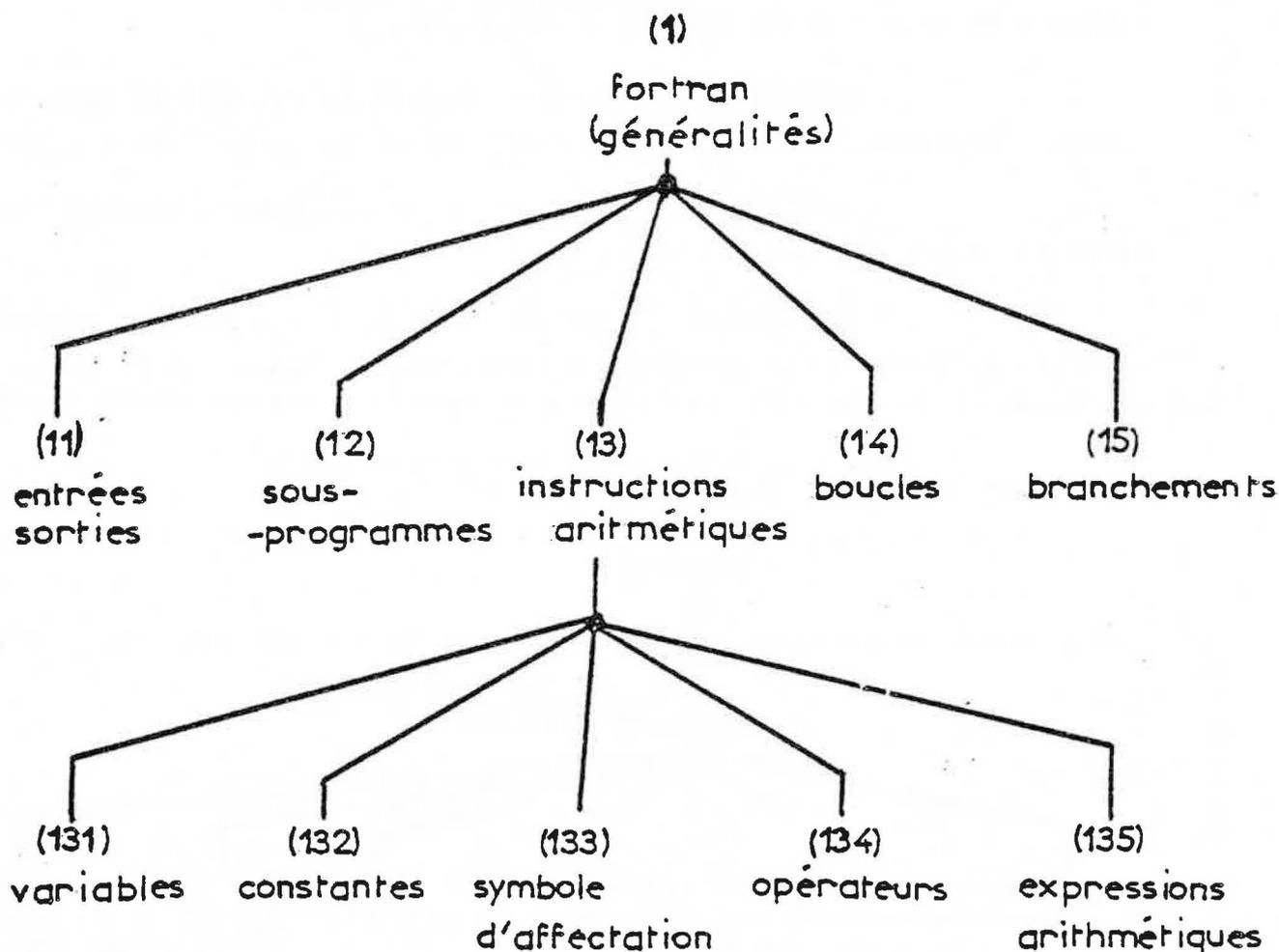


Figure 10: extrait de la structure d'analyse d'un cours de FORTRAN

4. 2. 2. - Choix de la décomposition d'un élément :

L'exploration d'un élément sera décomposée de la manière suivante :

- MATIERE (texte de 300 mots au plus) : donnera les définitions, les propriétés et quelques exercices pour soutenir l'attention. Cette caractéristique sera donc rédigée en LANGAGE EPMINE.

b) "RESUME DE L'ELEMENT I2" (instructions arithmétiques) sera rédigé en langage naturel :

RESUME SUR INSTRUCTIONS ARITHMETIQUES  
 DANS TOUS LES CAS ELLES SONT FORMEES PAR :  
 + UNE VARIABLE D'AFFECTION (A GAUCHE DE =)  
 + UN SYMBOLE D'AFFECTION (SIGNE =)  
 + UNE EXPRESSION ARITHMETIQUE (A DROITE DE =), CONTENANT DES OPERATEURS ( \*, \*\*, +, -, / ), DES PARENTHESSES, DES VARIABLES ET DES CONSTANTES EN MODE FIXE (ENTIERS) OU EN MODE REEL (NOMBRES REELS). LE MODE DE LA VARIABLE D'AFFECTION S'IMPOSE A LA VALEUR QU'ELLE RECOIT.

Lorsqu'un étudiant écrira la requête :

I RESUME DE L'ELEMENT I2 # ce texte s'écrira simplement sur l'écran et le programme attendra une nouvelle requête.

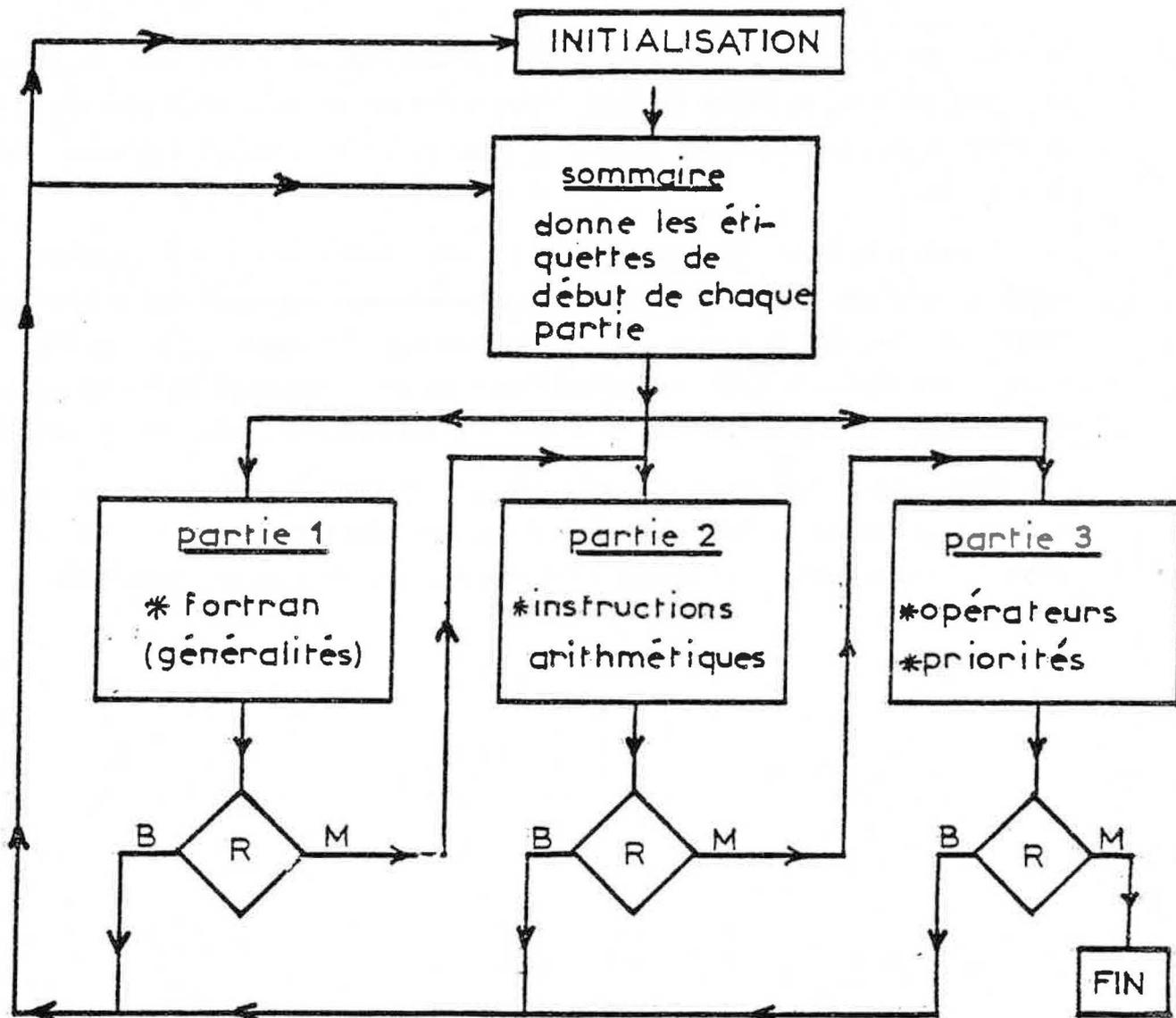
#### 4. 3. - COURS "TUTORIEL" FORTRAN :

Certains élèves étant gênés lorsqu'on leur donne la liberté de choisir leur démarche d'apprentissage, nous avons rédigé un cours de FORTRAN qui, à partir des éléments stockés dans la banque, impose un ordre et des "chemins" fonctions des résultats de l'étudiant. aux tests qui, dans ce processus sont intégrés au cours.

Sur la figure 12 (voir page suivante), on trouvera un schéma montrant une organisation d'un cours de FORTRAN de type "tutoriel".

On comprend que l'étudiant n'a plus qu'à choisir une partie à étudier puis à se conformer aux "ordres" du programme qui ne lui permet de passer à la partie suivante que si ses résultats sont satisfaisants, le renvoyant au "sommaire" dans tous les autres cas.

La comparaison entre les initiatives laissées aux étudiants par le cours utilisant SOMINE et la "directivité" imposée par le cours tutoriel schématisé ci-dessus montre bien les DIFFERENCES et aussi la COMPLEMENTARITE des deux méthodes d'apprentissage offertes par notre système. On devrait pouvoir mesurer l'efficacité de chaque méthode en fonction des "caractères des étudiants" et en tirer des enseignements profitables aux progrès en pédagogie.



R = résultat global à des tests

figure 12: organisation d'un cours "tutoriel" de FORTRAN

4. 4. - PREMIERS RESULTATS :

Tous ceux qui ont pratiqué l'expérimentation en pédagogie savent le temps qu'il faut pour obtenir des résultats ayant quelque signification. Le cours de fortran a été utilisé par trop peu de gens pour que nous puissions dire s'il est efficace ou non. Insistons davantage sur les caractéristiques du système :

- L'interpréteur du langage de cours a un temps de réponse de 2 secondes

environ. Cela veut dire qu'entre la fin d'une réponse d'étudiant et l'apparition sur l'écran du texte suivant, deux secondes se sont écoulées. Ce temps de réponse est satisfaisant puisqu'il correspond environ au temps de réponse de l'homme.

- Lorsque le cours est dans la banque, la recherche des informations fait appel à tous les programmes gérant la banque plus, quelquefois, à l'interpréteur de langage de cours. Dans l'état actuel des choses, le temps de réponse est alors de sept secondes. C'est un peu long mais la suppression de certaines écritures de contrôle devrait sensiblement réduire ce temps.

- Le stockage des cours est possible directement à partir de l'écran ou en batch, à partir de cartes. Cette deuxième solution est préférable car on garde la trace des informations stockées et les corrections sont plus faciles.

## CHAPITRE V : ENSEIGNEMENT DE LA STATIQUE : EXEMPLE DE PROGRAMME "D'AIDE PEDAGOGIQUE"

### 5. 1. - PROBLEMES POSES PAR L'ENSEIGNEMENT DE LA MECANIQUE RATIONNELLE :

En essayant de programmer un cours de STATIQUE (partie importante de la mécanique rationnelle), nous nous sommes heurtés à un certain nombre de difficultés venant des faits suivants :

- la mécanique est une science EXPERIMENTALE : son enseignement est difficilement concevable sans possibilités de MESURES sur des phénomènes réels ou simulés.

- Dans cette science (comme dans toute la physique d'ailleurs) il y a plus à COMPRENDRE qu'à APPRENDRE : les RELATIONS entre les éléments de connaissances sont aussi importantes que ces éléments eux-mêmes.

- Enfin, POSER correctement un problème est PLUS DIFFICILE que de le RESOUDRE : passer d'un mécanisme réel (une voiture, par exemple) à un ensemble de schémas sur lesquels il sera possible d'appliquer les théorèmes de la mécanique demande des capacités d'analyse et de synthèse infiniment plus importantes que celles qui devront être mises en oeuvre au moment de la recherche de la solution proprement dite.

L'enseignement de cette matière impose donc la "LIBRE EXPLORATION DU DOMAINE D'ENSEIGNEMENT" qui est la démarche la plus adaptée à la recherche des RELATIONS entre éléments. Il impose aussi les PROGRAMMES D'AIDE PEDAGOGIQUE car il serait presque impossible d'inclure DANS LE PROGRAMME D'ENSEIGNEMENT les simulations, les recherches documentaires, les analyses de problèmes indispensables.

La libre exploration du domaine d'enseignement est assurée par notre système lorsque les informations du cours sont placées dans une

banque SOMINE. Aussi, nous n'insisterons pas, dans les paragraphes suivants, sur cet aspect : l'analyse de la matière à enseigner, la recherche des relations entre éléments, le stockage et la structuration des informations ont été assez développés dans les chapitres précédents pour qu'il soit inutile d'y revenir. Par contre, après avoir rappelé quelques exemples de réalisations françaises et étrangères "d'aide pédagogique", nous centrerons notre exposé sur les buts et la réalisation d'un ANALYSEUR DE PROBLEME ORIGINAL : le programme MECANIK. Il permet d'analyser et de résoudre la plupart des problèmes de STATIQUE "dans l'espace et " le plan " et peut servir dans l'enseignement et dans l'aide à la conception des mécanismes.

#### 5. 2. - LES PROGRAMMES D'AIDE PEDAGOGIQUE :

Nous désignerons ainsi des programmes pouvant s'inclure dans un système d'enseignement quelconque, ne participant pas directement à la fonction d'enseignement mais pouvant être appelés par les étudiants lorsqu'ils désirent :

- faire appel à un moyen technologique autre que le cours : films, projections de diapositives, utilisation d'un écran graphique, etc...
- faire appel à un moyen pédagogique différent : simulations (expérience de physique, jeu d'entreprise, etc..), aide à la créativité, générateur de problèmes (posant des problèmes à un étudiant qui veut tester la valeur de ses connaissances) ou analyseur de problèmes (permettant à un étudiant de vérifier qu'il a compris comment les problèmes se posent dans tel domaine d'enseignement), etc..

Beaucoup de travaux ont été déjà publiés dans le domaine des "aides pédagogiques". Citons entre autres :

- la SIMULATION de phénomènes physiques, les travaux réalisés en France sous la direction de J. HEBEINSTREIT /14/, ceux réalisés aux Etats-Unis /10/, etc
- le PROGRAMME ESPACE est une aide pédagogique, complétant un cours d'électrotechnique et de construction des machines électriques. Il permet aux étudiants de construire un modèle de machine électrique et de vérifier si son fonctionnement satisfait ou non le cahier des charges initialement prévues /36/.

- Chacun connaît des exemples de jeux d'entreprises, de recherches documentaires qui, correctement utilisés, constituent d'importantes aides pédagogiques à des cours de gestion ou de conception de machines. Les travaux de M. Louis GAVET /31/, permettent de réduire dans de telles proportions les tailles mémoires nécessaires que ces programmes ne manqueront pas de se vulgariser.

- En France, l'INSTITUT NATIONAL DE RECHERCHE ET DE DOCUMENTATION PEDAGOGIQUES publie régulièrement des comptes rendus de travaux d'enseignants qui cherchent dans cette voie. Le numéro de 72-73 citent notamment les travaux de JOUADIN et MAUREL /34/ qui ont écrit, en collaboration avec leurs élèves, des programmes permettant l'ANALYSE ET LA RESOLUTION DE PROBLEMES DE PHYSIQUE.

- Enfin, nous citerons les travaux de WEXLER /32/ dans le domaine des GENERATEURS DE PROBLEMES : ces programmes ont pour avantages de limiter le nombre des exercices situés DANS le cours proprement dit et de pouvoir poser aux étudiants un grand nombre d'exercices d'une classe donnée, à partir d'un programme autonome et relativement petit.

5. 3. - UN EXEMPLE D'AIDE PEDAGOGIQUE : LE PROGRAMME "MECANIK" :

On peut dire très justement /34/ :

"l'ordinateur est capable d'effectuer les opérations logiques et les calculs nécessaires à la résolution des problèmes de STATIQUE. Pourtant, le temps consacré à l'apprentissage de ces calculs est beaucoup plus important que celui réservé à l'étude, à l'observation et à l'expérimentation des phénomènes physiques..." Pour économiser ce temps, JOUANDIN et MAUREL se sont donnés les objectifs prioritaires suivants :

- " - systématiser la résolution des problèmes de physique
- faciliter l'utilisation et la compréhension des lois de la physique par une recherche précise de leurs conditions d'application,
- faire utiliser l'informatique par les élèves au sujet de la résolution de ces problèmes. "

A partir de ces objectifs, ils ont réalisé un travail tout a fait remarquable avec des élèves d'une classe de lycée technique.

Voulant construire un cours de mécanique pour des élèves d'Institut Universitaire de Technologie, notre démarche ne pouvait être identique.

Nous sommes partis de l'idée qu'il est plus DIFFICILE DE POSER CORRECTEMENT UN PROBLEME DE MECANIQUE QUE DE LE RESOUDRE. Nous limitant à la STATIQUE, nous pouvons dire que, POSER UN PROBLEME, c'est DECRIRE UN MECANISME (réel ou représenté sur un dessin d'avant-projet) en SCHEMATISANT chacun des SOLIDES et TOUTES LES LIAISONS entre solides. Cette DESCRIPTION, comprend aussi les actions mécaniques CONNUES qui agissent en divers points de l'ensemble ISOLE. Elle doit être extrêmement claire et précise. Si on veut JUGER une telle description, l'ordinateur peut être d'un grand secours car, comme le remarque A. M. BORK /33/, il est doté d'une "INCOMPREHENSION FONDAMENTALE" : l'étudiant devra TOUT lui dire, dans des termes extrêmement PRECIS s'il veut que le schéma décrit soit "COMPRIS".

#### 5.3.1. - Les étapes d'un problème de statique :

Elles sont résumées sur la figure 13. On y remarquera principalement les quatre phases : analyse et mise en forme du problème, mise en équations à partir de l'application des lois de la statique, résolution des équations obtenues, interprétation des résultats et les deux tests permettant de juger la qualité de l'analyse.

A partir de l'ordinogramme de la figure 13, il est possible de trouver les fonctions principales du programme MECANIK (figure 14).

#### 5.3.2. - Les fonctions principales du programme mecanik :

A partir de l'ordinogramme de la figure 13 (page suivante), on peut déduire les fonctions principales du programme MECANIK qui sont résumées sur la figure 14.

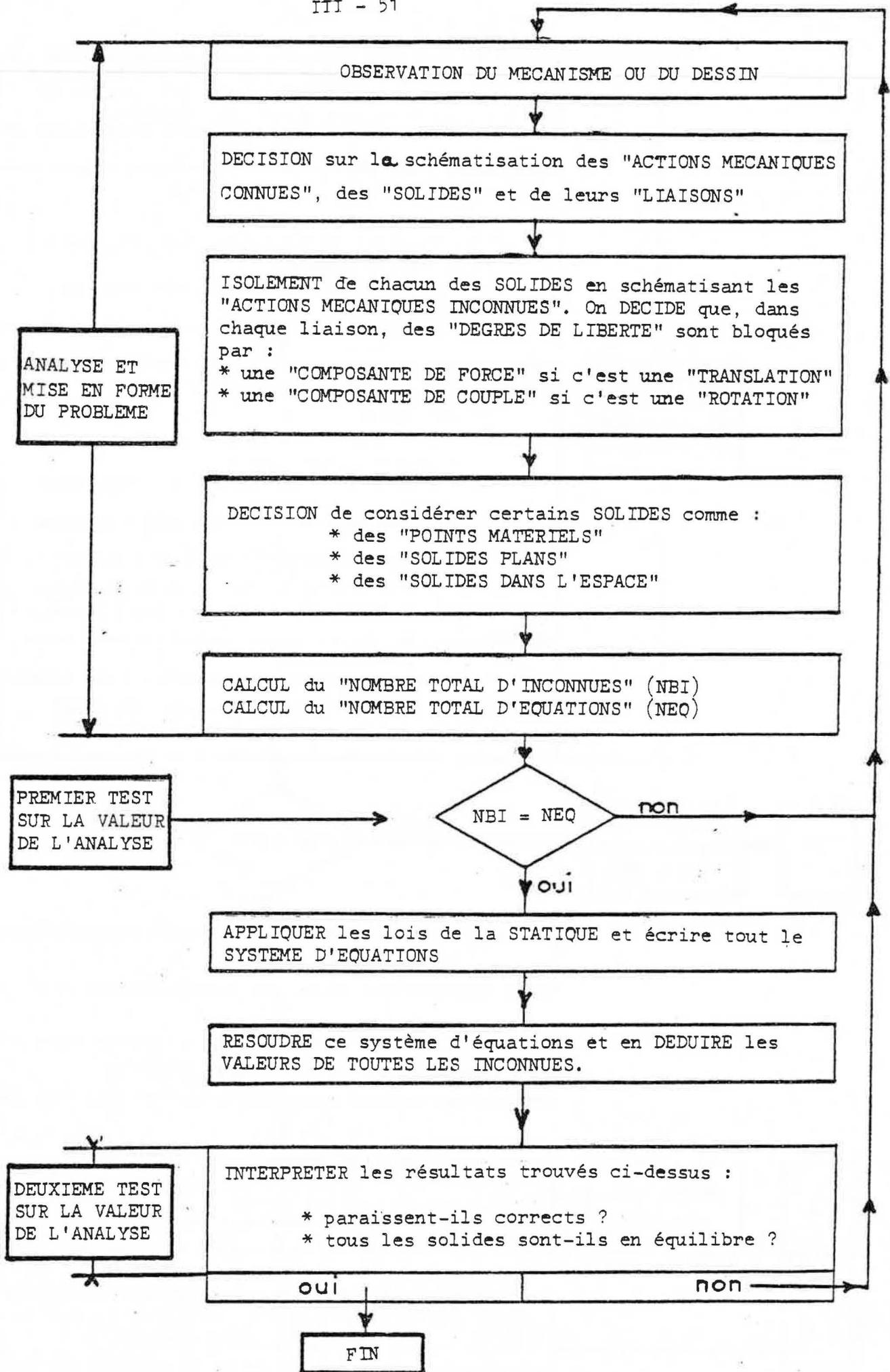


Figure 13 : les étapes d'un problème de statique

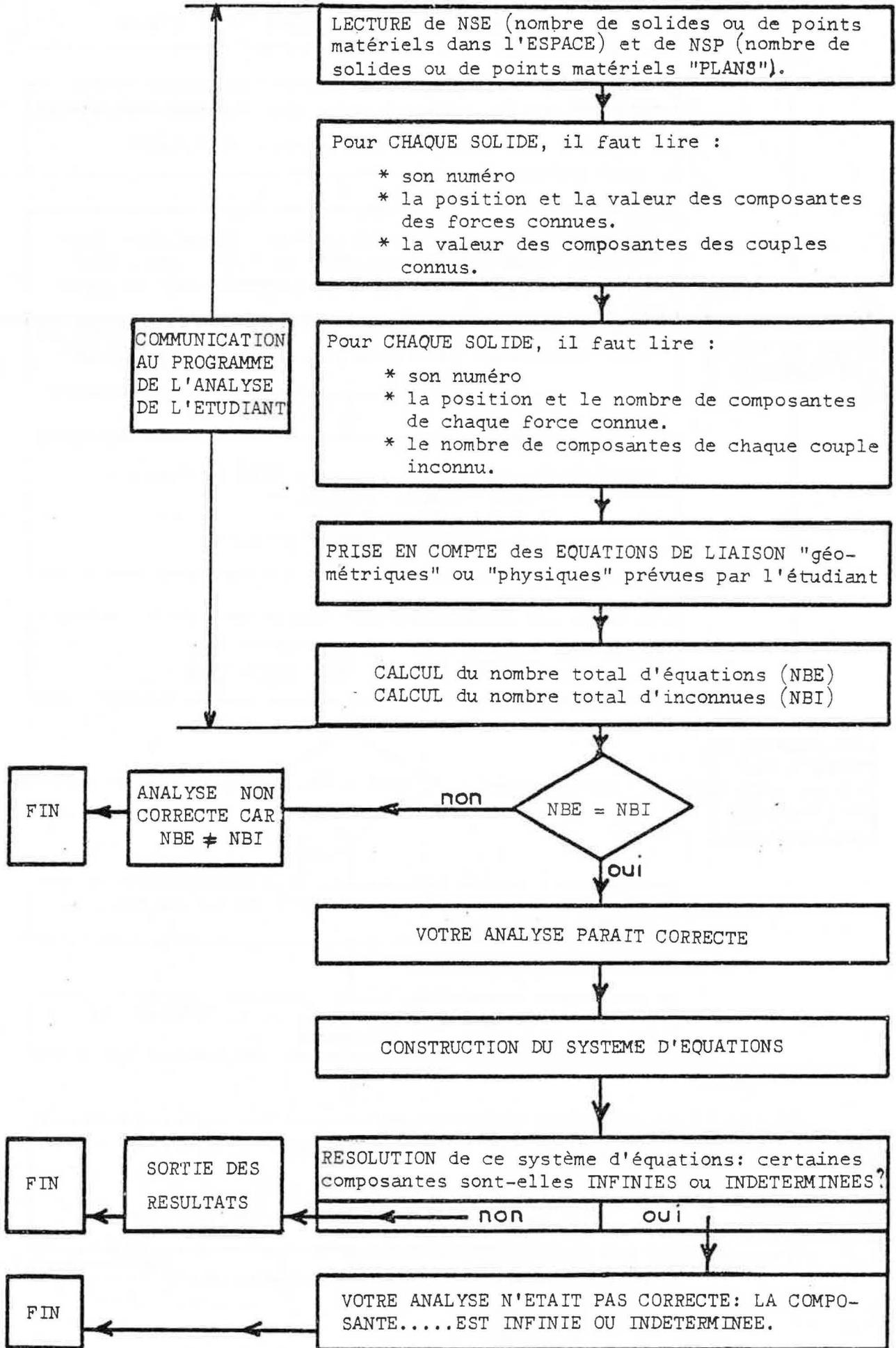


Figure 14 : Fonctions principales du programme MECANIK

## 5.3.3. - Exemple d'application du programme mécanique :

Voulant étudier l'ensemble d'un PEDALIER DE MOBYLETTE, on a dessiné le schéma qui représente les solides et les liaisons entre ces solides figure 15.

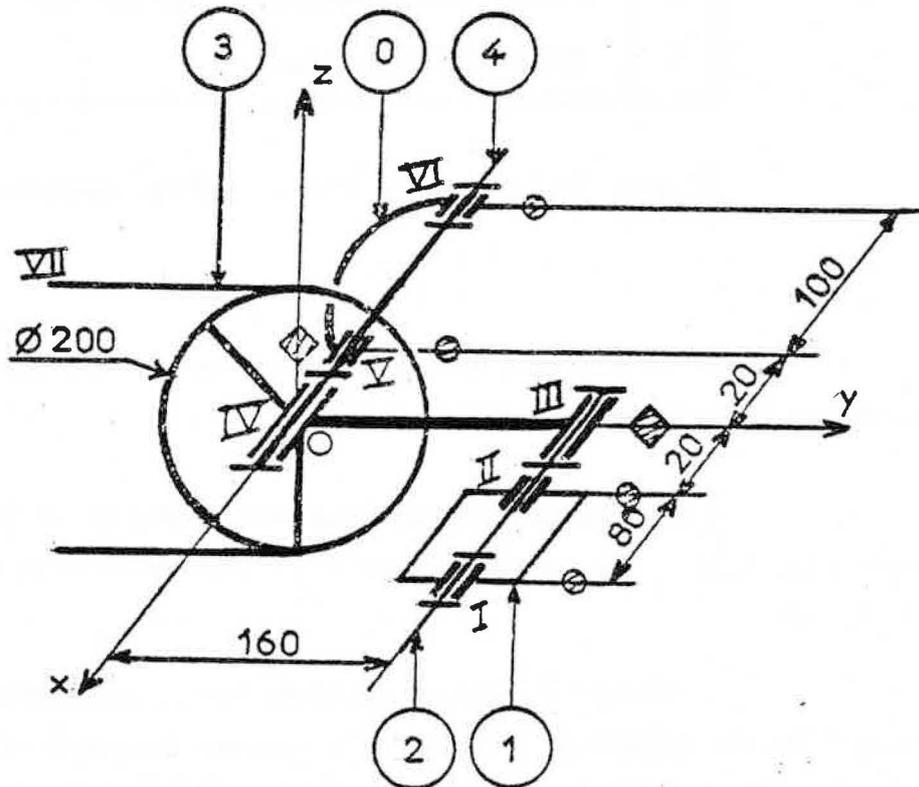


Figure 15: schéma d'un PEDALIER

On se place dans le cas où la manivelle est horizontale. L'action mécanique du PIED SUR LA PEDALE est considérée comme un torseur spécial de résultante :  $\vec{F}_{\text{pied/péda.}} = -500 \vec{z}$  (Newton). On néglige les poids des pièces et décide d'appliquer l'hypothèse "adhérence nulle" dans toutes les liaisons inter-solides.

Tous les systèmes, au nombre de quatre, (voir figure 16) seront isolés "dans l'espace" (NSE = 4 ; NSP = 0).

systèmes isolés	
N°	composant du système
1	pédale
2	axe de pédale
3	manivelle+pédalier+brins de chaîne
4	axe de pédalier

Figure 16 : les systèmes isolés successivement

On veut connaître les actions mécaniques dans TOUTES LES LIAISONS. Le listing de l'annexe (I) montre :

- la rentrée des "actions mécaniques CONNUES " agissant sur les solides 1, 2, 3, 4 . Les coordonnées et les composantes sont rentrées en format E 12.5

- Chaque liaison où agissent des actions mécaniques inconnues est numérotée. Le "point de réduction du torseur inconnu" est donné en format E 12.5. Les "composantes prévues" des forces et des couples sont indiquées par le numéro positif de la liaison s'il s'agit d'une ACTION et par ce numéro avec le signe moins s'il s'agit d'une REACTION (format I 4 ).

A ce moment, l'étudiant (ou le projeteur voulant calculer les actions mécaniques pour déterminer les formes à donner aux pièces) a pris TOUTES SES DECISIONS et les a indiquées sur cartes perforées (ou sur écran) comme données du programme "MECANIK". Ce programme s'exécute et donne :

- UN PREMIER AVIS SUR LA VALEUR DE L'ANALYSE :

ici : " Votre analyse paraît correcte car le nombre des équations est égal au nombre des inconnues (24)" .

- LES TABLEAUX DES COEFFICIENTS  $A(I,J)$  et  $B(I)$  du système d'équations linéaires qu'il a construit.

- Enfin on obtient :

\* soit un message indiquant que l'analyse est probablement incorrecte (ou que le système est instable), si ces tableaux conduisent à des solutions

indéterminées ou infinies.

\* Soit, (comme dans notre exemple), le tableau des solutions.

Les actions mécaniques dans les diverses liaisons sont ainsi calculées.

#### 5. 4. - MECANIK : UN ANALYSEUR ET UN GENERATEUR DE PROBLEMES :

Si les problèmes sont situés dans le cours lui-même, ils alourdisent considérablement la rédaction et la taille des fichiers nécessaires aux programmes d'enseignement. Certains chercheurs comme UTTAL /35/, cité par WEXLER /32/, ont écrit des programmes qui GENERENT des énoncés de problèmes et, parfois aident les étudiants dans la recherche de la solutions en leur donnant des conseils (s'ils les demandent) ou en effectuant certaines opérations de calculs à partir des données des élèves.

Nous pensons que le programme MECANIK peut être considéré comme un "générateur de problèmes" car, on peut facilement lui adjoindre un fichier contenant des énoncés. Mais il nous semble qu'il est plutôt un ANALYSEUR c'est à dire une sorte de JUGE, devant lequel les étudiants peuvent décrire, dans le code très précis que nous avons indiqué, des mécanismes REELS. Cette description a pour but d'obtenir des schémas sur lesquels les théorèmes de la statique des solides pourront être appliqués. MECANIK "dira" que ce but est atteint et publiera les résultats cherchés ou demandera à l'étudiant de revoir son analyse.

Actuellement, le programme MECANIK a été utilisé par soixante élèves de deuxième année de l'Institut Universitaire de Technologie de ST-ETIENNE qui lui ont "soumis" des analyses de mécanismes très divers : étau-limeur, scie alternative mécanique, petite grue d'atelier, cisaille, etc... Ils semblent que les buts qui ont présidé à l'écriture et la mise au point de ce programme sont atteints car :

- les étudiants sont fortement motivés
- ils comprennent la difficulté et la nécessité de beaucoup de rigueur dans la MODELISATION du réel
- ils établissent une liaison enrichissante entre l'informatique (comme mode de pensée), la programmation (à travers l'étude des formats et la rédaction des cartes de données) et la mécanique rationnelle.



## CHAPITRE VI : UN ESSAI D'APPLICATION DE L'E.A.O. A LA REEDUCATION D'ENFANTS DYSLEXIQUES

(Travail réalisé en collaboration avec J. FOURNEL, instituteur spécialisé)

### 6. 1 - BUTS DE LA RECHERCHE SUR LA REEDUCATION DES DYSLEXIQUES :

Comme nous l'avons écrit dans l'introduction de cette 3ème partie ( E.A.O.), il nous paraissait intéressant d'utiliser la FORTE MOTIVATION donnée par le travail en Enseignement Assisté par Ordinateur pour :

- accroître les capacités d'attention des élèves
- révaloriser à leurs yeux la lecture et l'écriture en leur donnant la possibilité d'accroître leurs performances.

En outre, en leur confiant un matériel cher et "captivant" nous espérions :

- leur montrer qu'ils sont "digne d'intérêt"
- augmenter leur confiance en eux.

Enfin, J. FOURNEL ayant accepté de rédiger les "leçons" et de les programmer au moyen du langage EPMINE, c'était, pour ce langage un test intéressant.

### 6. 2 - CHOIX DES ELEVES ET DES MOYENS :

#### 6. 2. 1 - Les élèves :

CHRISTIAN ( 11,3) en CM2 et PHILIPPE ( 12,4) en 6ème sont deux dyslexiques notoires, surtout PHILIPPE P. qui est presque un cas extrême.

FREDERIC (10,8) est un élève de CM2 qui n'est pas à proprement parler un dyslexique mais, il est faible en orthographe (note zéro en général) .

Il a, en particulier, de la difficulté à reproduire un dessin symétrique ou un dessin de mémoire.

ASTRID ( 14,6 ), élève de troisième a été rééduquée pendant sa scolarité primaire. Elle a encore des difficultés, principalement en ce qui concerne l'orthographe d'usage.

PHILIPPE B. ( 14 ans ) en 5ème qu'il redouble en grande partie à cause du français, ne peut pas non plus être qualifié de dyslexique mais il fait partie des élèves ayant de sérieuses difficultés.

Cet échantillon nous a semblé représentatif des divers troubles qui sont rencontrés le plus fréquemment et que nous voulions étudier. Nous avons parfois invité un élève "moyen en orthographe" à titre d'élément de comparaison.

6. 2. 2. - Les cours :

Pour un essai de quelques semaines, nous avons dû délimiter très nettement notre sujet d'étude. Nous avons décidé d'organiser une séance d'une heure environ par semaine et de laisser de côté toutes les difficultés possibles de socialisation, de communication orale, de répétition /38/, etc..... Nous avons voulu agir uniquement sur la mémoire visuelle, et même, il est intéressant de le préciser, sur un seul de ses aspects : celui qui concerne la mémorisation des lettres, des signes ou des groupes de mots.

A partir de ces contraintes et de ces objectifs, J. FOURNEL a donc rédigé six "leçons" comprenant chacune cinq parties principales :

\* 1) Lecture sur l'écran de groupes de lettres de plus en plus longs pendant un temps uniforme de quatre secondes. Effacement du groupe. Réécriture du groupe par l'élève sur l'écran. Analyse de la réponse par le programme d'E. A. O. Commentaire. Nouvelle question.

exemples de groupes de lettres : DINC  
 FRTVA  
 CFAVNO  
 .....

.....

CLUMTORL

CRIVLETPROL

Notre objectif a été d'obliger l'élève à voir toutes les composantes du groupe puisqu'il ne peut se référer au sens ; il s'agit donc ici davantage de rééducation de la dysorthographe que de la dyslexie. En effet, la lecture n'implique pas forcément (et heureusement) une analyse aussi minutieuse. Cependant, on peut se demander si la dysorthographe et la dyslexie ne sont pas intimement liées.

\* 2) Recherches de groupes de signes : on présente à l'élève, pendant quatre secondes le groupe de signes =?X. On projette ensuite les groupes =X? X?= =X? =?X et on lui demande les numéros des groupes identiques au premier (ici, il doit répondre : 4)

\* 3) Identification de mots : le principe est le même que lors de l'exercice précédent. Exemple : on présente le mot PUBLIC puis la suite de "mots" : PUBLIC PLUBLIC PULBIC PLUBIC PUBLIC et la réponse correcte est : 5. Naturellement, il peut y avoir plusieurs ou pas de réponses correctes.

\* 4) Exercices préparatoires aux règles d'accord : On travaille sur des groupes de signes. On présente une règle du type "COMPLETE PAR + TOUS LES GROUPES DE SIGNES SE TERMINANT PAR X". L'élève peut faire réapparaître la règle s'il en éprouve le besoin. On présente alors des séries de groupes de signes à compléter. Exemple :

3VA R'?NX+X ?/: AX

L'élève doit compléter les groupes 2 et 4.

\* 5) Exercices portant sur les accords d'adjectifs qualificatifs : On projette : " DES VIRAGES ... " (pendant quatre secondes) puis ce morceau de phrase disparaît pour être remplacé par la partie sur laquelle l'élève devra travailler : " ..... VERGLACE ET DANGEREUSE."

L'élève doit corriger les terminaisons des adjectifs.

|| On trouvera une de ces "leçons", programmée à l'aide du langage  
|| EPMINE, à l'ANNEXE II de cette partie du mémoire.

.../...

6. 2. 3. - Apprentissage du fonctionnement du système :

Lorsqu'une question apparaît sur l'écran, l'élève doit :

- attendre d'être dans l'état "réponse possible" (observation d'un signal)
- commencer sa réponse par un caractère particulier (flèche)
- écrire sa réponse
- terminer sa réponse par un caractère particulier (dièse)
- transmettre sa réponse en appuyant sur la touche "transmit"

L'ensemble de ces opérations constituent une "règle" complexe et nous pensions que son apprentissage serait long. Cette règle devant être appliquée cinquante fois par leçon, nous nous attendions à de nombreuses erreurs.

Au contraire, son apprentissage a été court (10 minutes au plus) et le nombre d'erreurs dans son application ultérieure négligeable (de l'ordre de 4%).

Nous reviendrons sur ce point au moment du bilan.

6. 2. 4. - Mesures quantitatives :

La deuxième leçon (voir annexe II) a servi de test. Les élèves l'ont abordée :

- à sa place normale (test de début de cours)
- lors de la septième séance (test de fin de cours)

6. 3 - LES RESULTATS OBTENUS :

Nous distinguerons :

- les résultats quantitatifs obtenus aux tests de début et de fin d'apprentissage.
- les résultats qualitatifs obtenus à partir de l'observation de certaines attitudes d'élèves réalisées par nous ou par leurs parents.

6. 3. 4. - Résultats quantitatifs :

Lors du passage de la deuxième leçon à sa place normale, chaque enfant a commis un certain nombre d'erreurs.

Lors du passage de la deuxième leçon COMME TEST DE FIN DE COURS, chaque enfant a commis un certain nombre d'erreurs qui dans la plupart des cas est moins important que lors du premier passage. Le tableau ci-dessous donne, pour chaque enfant, le pourcentage de diminution du nombre d'erreurs lors du deuxième passage.

NOMS	Pourcentage de diminution du nombre d'erreurs lors du deuxième passage du test
CHRISTIAN	0 %
PHILIPPE P	50 %
FREDERIC	20 %
PHILIPPE B	40 %
ASTRID	30 %

soit, en moyenne, une amélioration de 35 % qui peut être qualifiée "d'assez importante".

Un autre résultat quantitatif est constitué par le faible nombre d'erreurs enregistrées dans l'application complexe de la procédure de réponse aux questions ( voir paragraphe 4 - 3).

L'interprétation de ce résultat étonnant peut-être la suivante : Si les élèves COMPRENNENT L'IMPORTANCE D'UNE REGLE DANS LES RELATIONS QU'ILS ENTRETIENNENT AVEC LE "MILIEU EXTERIEUR", ils sont capables de l'appliquer. Si cette interprétation est valable, on pourrait en déduire que les élèves "mauvais en orthographe" ( ou dyslexiques ? ) n'ont pas, au moment où on LEUR APPREND CERTAINES REGLES compris leur IMPORTANCE. Cette constatation ne donne pas la solution du problème posé par la dysorthographe mais elle pourrait aider à trouver une voie de recherche.

#### 6. 3. 2. - Propos et comportements d'élèves :

PHILIPPE B, ASTRID et CHRISTIAN ont déclaré à plusieurs reprises " l'utilisation de la machine m'a vivement intéressé".

Autre propos : " réticent au début, j'ai trouvé cela captivant " (FREDERIC)

Au point de vue des comportements on peut noter :

- l'application générale des élèves qui s'est traduite par une attention soutenue
- la plupart des parents ont remarqué un "changement" dans l'attitude des enfants. Certains parents ont ainsi éprouvé le besoin d'assister à une "leçon".
- PHILIPPE P. a été "en progrès" au troisième trimestre à l'école.

Nous notons simplement ces réflexions et constatations en nous gardant bien d'en tirer, pour le moment, une conclusion quelconque.

6. 4. - CONCLUSION :

Dans ce premier essai d'applications de l'Enseignement Assisté par Ordinateur à la rééducation des dyslexiques, on peut noter un certain nombre de défauts importants :

- toutes les possibilités du système E.A.O. - Mines n'ont pas été utilisées. On s'est trop facilement et trop rapidement contenté d'employer la partie "tutorielle" de ce système.

- un seul type de leçon a été utilisé. On peut en imaginer bien d'autre.

- rien n'a été fait pour savoir COMMENT lisent les gens "normaux" et les "dyslexiques" (que voient-ils dans le mot ? dans la phrase ? dans un dessin ou une lettre ?)

- l'expérience a été courte et a porté sur un nombre insuffisant d'élèves : 7 leçons suivies par 5 ou 6 élèves.

Malgré cela, des résultats positifs ont été enregistrés :

- . Il semble que si un élève comprend vraiment l'IMPORTANT D'UNE REGLE, il apprend et applique cette règle avec plus de facilité.
- . L'ordinateur a été "motivante". L'expérience semble avoir apporté chez des élèves des changements de comportements dans d'autres domaines (école, milieu familial...)
- . Une amélioration sensible (35 %) a été obtenue entre les deux passages du

- même test, c'est-à-dire après un temps moyen d'apprentissage de l'ordre de 5 h.
- . Les élèves paraissaient contents. Il n'y a eu aucune absence.
  - . Au point de vue "système" le langage EPMINE et son interprétation ont été utilisés par J. FOURNEL et ont donné satisfaction.

Il serait illusoire de tenir ces quelques conclusions pour "définitives". On peut cependant les prendre comme nouvelles hypothèses de départ si on tente d'autres expériences

#### 6. 5. - VERS DE NOUVEAUX ESSAIS :

Au congrès sur l'Enseignement Assisté par Ordinateur qui vient de se tenir à MARSEILLE, il nous a semblé que notre direction de recherche était intéressante et assez originale. Elle porte en effet sur un sujet pratiquement inexploré. Cette constatation renforce notre désir de poursuivre nos travaux en adoptant les axes suivants :

- a) Faire suivre l'ensemble du cours qui est au point par d'autres élèves pour voir si les premiers résultats obtenus sont confirmés.
- b) Mettre au point un TEST permettant peut-être de distinguer les dyslexiques de ceux qui, POUR D'AUTRES RAISONS, ont des difficultés en lecture ou en orthographe. Il résulte en effet, de l'observation d'un certain nombre d'élèves passés en classe d'adaptation que certains ne sont pas à proprement parler des dyslexiques. Or, il se trouve que ces "pseudo-dyslexiques" obtiennent assez rapidement de bons résultats à des exercices de mémorisation de lettres ou de dessins. Un critère aidant au diagnostic semble donc pouvoir être avancé comme hypothèse de travail.
- c) Trouver d'autres types de "leçons" et mesurer leur impact sur les élèves :
  - \* lecture ou composition d'histoires
  - \* Utilisation de dessins à compléter
  - \* Cours de calculs
  - \* Apprentissage d'autres règles d'accord
  - \* Mémorisation de dessins
  - \* etc....

d) Mise au point d'une machine à enseigner ou de terminaux commandés par MICRO-ORDINATEUR afin d'accroître la diffusion des cours validés sur ordinateur. On pourrait ainsi faire bénéficier des résultats de recherche des élèves restant dans leur classe, pris en charge par les G.A.P.

Les travaux portant sur les MATERIELS et les LOGICIELS nécessaires pourraient être confiés à l'Ecole des Mines de Saint-Etienne qui entreprend une étude préliminaire dès maintenant.

e) Mise au point d'un terminal audio-visuel qui pourrait être :

- un écran ALPHANUMERIQUE actuel complété par un magnétophone dans un premier temps.
- un écran de type "graphique et alphanumérique" complété par un récepteur de télévision à plus long terme.

On pourrait alors comparer les méthodes et les résultats obtenus par des personnes "normales" et par des personnes atteintes de troubles de l'écriture, de la lecture ou de l'audition.

f) L'application des travaux de recherche en oculométrie /39/ serait sans doute d'une grande importance. Si on arrivait à filmer le déplacement de l'oeil sur le papier pendant la lecture (ou l'écriture), on pourrait certainement en tirer des conclusions intéressantes. Nous aimerions explorer cette voie interdisciplinaire.

o  
o o  
o

Voici, classés approximativement par ordre de complexité croissante, les axes de recherche qui compléteraient l'expérience réalisée pendant l'année scolaire ( 1974 - 1975 )

## CHAPITRE VII : BILAN ACTUEL DES RECHERCHES EN E.A.O.

A la fin de cette partie consacrée à " l'utilisation de SOMINE en E.A.O., " que peut - on retenir ?

D'abord, dans le contexte des recherches en Enseignement Assisté par Ordinateur dont un bilan a été fait récemment à MARSEILLE /40/ il est certain que nous devons insister sur la modestie de notre contribution ! Nous avons seulement posé quelques questions et, parfois, apporté des éléments de réponse dans les domaines suivants :

- \* Apport d'un système de bases de données de type SOMINE à l'E.A.O.
- \* Construction d'un système d'E.A.O. autour d'un système de bases de données.
- \* utilisation de la forte MOTIVATION des élèves par l'E.A.O. dans un essai de rééducation d'enfants dyslexiques.

Ci-après, nous développerons ces trois points avant de terminer cette conclusion par l'indication de quelques axes de recherches qui contribueraient à l'amélioration du système d'enseignement E.A.O.-MINE

### 7. 1. - APPORT DE SOMINE A UN SYSTEME D'E.A.O. :

L'utilisation du système de bases de données SOMINE dans l'Enseignement Assisté par Ordinateur a permis de proposer des améliorations de ce mode d'enseignement dans plusieurs directions principales :

- a) - Dans l'analyse de la matière à enseigner : les structures d'arbres ou de graphes de SOMINE sont utiles pour hiérarchiser les éléments de connaissance et mettre en évidence les relations (causalité, similitude, différence, etc....) qui relient éventuellement ces éléments.

b) - Dans la phase d'enseignement : Plusieurs voies sont à la disposition des étudiants pour explorer un domaine d'enseignement.

- exploration "élément par élément" dans l'ordre du "sommaire" ou dans un ordre totalement libre

- exploration "élément par élément" en utilisant toutes les "caractéristiques" prévues par l'équipe enseignante ou seulement certaines d'entre-elles.

- exploration à partir de la recherche des relations entre éléments.

On comprend que ces diverses méthodes enrichissent de façon appréciable le dialogue entre les étudiants et les "cours" qui leurs sont dispensés.

c) Dans le " SUIVI " des étudiants : SOMINE a permis d'ajouter aux "compteurs" la possibilité de saisir " sur le vif" les réactions des étudiants en langage naturel. A partir des compteurs les cheminements, les erreurs et les réponses exactes des étudiants peuvent être reconstitués et analysés. Des statistiques sur ces compteurs contribuent à l'amélioration des cours. A cela, SOMINE ajoute la possibilité d'enregistrer au niveau de chaque élément ( dans la caractéristique " AMELIORATION") les idées, suggestions ou critiques des élèves.

d) Dans le développement de la créativité des équipes enseignantes : SOMINE offre plusieurs bases de données. Une contient les éléments du cours. D'autres sont disponibles pour des programmes " d'aide pédagogiques". Nous espérons que cela incitera les enseignants à compléter les cours par de la documentation, des simulations, des générateurs -analyseurs de problèmes, des aides audio-visuelles etc.....

Enfin, le cours étant placé dans une base de données, il n'y a aucune difficulté pour le stocker ou le corriger en mode conversationnel à partir d'écrans alphanumériques ou graphiques.

7. 2. - CONSTRUCTION D'UN SYSTEME D'E.A.O. AUTOUR D'UN SYSTEME DE BASES DE DONNEES :

L'utilisation de SOMINE comme "centre" d'un système d'enseignement a mis en évidence le fait qu'un système de bases de données doit être complété si on veut l'utiliser en E.A.O..

Il est indispensable de lui adjoindre :

\* un langage de programmation de cours comprenant notamment un analyseur des questions et des réponses des étudiants.

\* des fonctions de recherche des informations dans la banque qui donnent à celle-ci un certain caractère déductif (fonctions, similitude, différence, causalité, etc.....)

Cela a été réalisé dans notre système au moyen du langage de cours " EPMINE" pour le premier point et de l'interpréteur de requête des étudiants "ENSEIGN" pour le second.

"EPMINE" est un langage simple mais facile à compléter par l'adjonction de nouvelles fonctions.

"ENSEIGN" coiffe l'interpréteur des requêtes du langage SOMINE pour lequel il "rédige" des requêtes "correctes" à partir des questions des élèves.

### 7. 3. - UTILISATION DE LA MOTIVATION DES ELEVES PAR L'E.A.O. :

Le matériel coûteux, le jeu à caractère un peu magique que propose l'enseignement au moyen d'un terminal relié à un ordinateur motivent fortement les élèves. Cela est confirmé aussi bien avec des étudiants d'I.U.T. ( 20 ans environ), utilisant l'analyseur de problèmes "MECANIK" que lorsque nous tentons la rééducation de jeunes dyslexiques (10 ans en moyenne) par ce moyen.

A l'I.U.T., les étudiants qui "posent" leurs problèmes de statique au programme "MECANIK" sont ainsi sensibilisés à l'informatique et le cours de mécanique générale devient alors " interdisciplinaire" !

Les jeunes dyslexiques, incapables parfois d'appliquer des règles simples d'écriture de mots, se révèlent aptes à utiliser le système de réponse à partir des écrans dont la procédure est assez complexe. Dans beaucoup de cas, une amélioration (sans doute passagère) de l'attention a été notée par les parents ou par les maîtres. Ces résultats sont intéressants et même s'ils portent sur un trop petit nombre d'essais pour qu'on puisse en tirer des conclusions définitives, ils peuvent au moins servir d'hypothèses de départ pour de nouvelles expériences.

7. 4. - POUR AMELIORER LE SYSTEME D'ENSEIGNEMENT :

Etant donné le contexte dans lequel se situe notre système et les défauts qu'ont fait apparaître la première année d'utilisation avec des élèves, il nous semble qu'il faudrait développer nos recherches suivant deux axes principaux :

a) au niveau du langage EPMINE : l'analyseur de réponses actuellement opérationnel est simple mais trop sommaire. Il faudrait le perfectionner et lui donner des caractéristiques comparables à celles de l'analyseur de réponse du système O.P.E. /41/. On pourrait aussi s'inspirer des travaux sur les " i - grammaires" de M. PEUCHOT /5/

b.) au niveau des terminaux : l'extension des essais sur la rééducation d'enfants dyslexiques exige la mise au point de terminaux de type "audio-visuel". Avec les écrans graphiques ou alpha numériques, on peut seulement déceler les troubles qui se produisent à la lecture, lors de l'interprétation ou lors de la production d'un message visuel.

Un terminal audio-visuel étendrait le champ des recherches car il permettrait d'écrire ou dicter un message et de contrôler sa compréhension. Ainsi, on pourrait déceler les troubles qui se produisent lors de l'audition, la lecture, l'interprétation et l'écriture d'un message.

Les travaux dans ce domaine s'inspireraient éventuellement du système SODOME /42/. Ils pourraient aussi conduire à la réalisation de "machines à enseigner" simples, réalisant les fonctions nécessaires à la diffusion dans les classes des cours mis au point sur terminal d'ordinateur. On peut imaginer, par exemple, l'installation d'un micro-ordinateur et de quelques terminaux dans une classe spécialisée pour le traitement de certains troubles.

**TROISIEME PARTIE : ANNEXE I**

**EXEMPLE D'UTILISATION DU PROGRAMME "MECANIK"**

ACTIONS INCONNUES AGISSANT SUR LE SOLIDE 1

=====

COORDONNEES DU CENTRE DE LA LIAISON

COMPOSANTES PREVUES INDIQUEES PAR LE NUMERO POSITIF DE LA LIAISON  
(UN NOMBRE NEGATIF INDIQUE UNE "REACTION")

	AXE(1)	AXE(2)	AXE(3)	AXE(1)	AXE(2)	AXE(3)
FORCE:	.10000E+03	.16000E+03	.00000E+00	1	1	1
FORCE:	.20000E+02	.16000E+03	.00000E+00	0	2	2
COUPLE:	COORDONNEES INUTILES OU FIGURANT CI DESSUS			1	0	0

III - 72

ACTIONS INCONNUES AGISSANT SUR LE SOLIDE 2

=====

COORDONNEES DU CENTRE DE LA LIAISON

COMPOSANTES PREVUES INDIQUEES PAR LE NUMERO POSITIF DE LA LIAISON  
(UN NOMBRE NEGATIF INDIQUE UNE "REACTION")

	AXE(1)	AXE(2)	AXE(3)	AXE(1)	AXE(2)	AXE(3)
FORCE:	.10000E+03	.16000E+03	.00000E+00	-1	-1	-1
FORCE:	.20000E+02	.16000E+03	.00000E+00	0	-2	-2
FORCE:	.00000E+00	.16000E+03	.00000E+00	3	3	3
COUPLE:	COORDONNEES INUTILES OU FIGURANT CI DESSUS			3	3	3
COUPLE:	COORDONNEES INUTILES OU FIGURANT CI DESSUS			-1	0	0

ACTIONS INCONNUES AGISSANT SUR LE SOLIDE 3

\*\*\*\*\*

COORDONNEES DU CENTRE DE LA LIAISON

COMPOSANTES PREVUES INDIQUEES PAR LE NUMERO POSITIF DE LA LIAISON  
(UN NOMBRE NEGATIF INDIQUE UNE "REACTION")

	AXE(1)	AXE(2)	AXE(3)	AXE(1)	AXE(2)	AXE(3)
ORCE:	.00000E+00	.16000E+03	.00000E+00	-3	-3	-3
ORCE:	.00000E+00	.00000E+00	.00000E+00	4	4	4
ORCE:	.00000E+00	.00000E+00	.10000E+03	0	7	0
OUPL:	COORDONNEES INUTILES OU FIGURANT CI DESSUS			-3	-3	-3
OUPL:	COORDONNEES INUTILES OU FIGURANT CI DESSUS			4	4	4

ACTIONS INCONNUES AGISSANT SUR LE SOLIDE 4

\*\*\*\*\*

COORDONNEES DU CENTRE DE LA LIAISON

COMPOSANTES PREVUES INDIQUEES PAR LE NUMERO POSITIF DE LA LIAISON  
(UN NOMBRE NEGATIF INDIQUE UNE "REACTION")

	AXE(1)	AXE(2)	AXE(3)	AXE(1)	AXE(2)	AXE(3)
ORCE:	.00000E+00	.00000E+00	.00000E+00	-4	-4	-4
ORCE:	-.20000E+02	.00000E+00	.00000E+00	0	5	5
ORCE:	-.12000E+03	.00000E+00	.00000E+00	6	6	6
OUPL:	COORDONNEES INUTILES OU FIGURANT CI DESSUS			-4	-4	-4

FIN D'ANALYSE

\*\*\*\*\*

L'ANALYSE PARAIT CORRECTE CAR LE NOMBRE D'EQUATIONS EST EGAL AU NOMBRE DES INCONNUES (24)

LIAISON 1 FORCE APPLIQUEE SUR LE SOLIDE 1  
=====

XF(1)= .00000E+00

XF(2)= .00000E+00

XF(3)= .25000E+03

LIAISON 1 COUPLE APPLIQUEE SUR LE SOLIDE 1  
=====

XC(1)= .00000E+00

LIAISON 2 FORCE APPLIQUEE SUR LE SOLIDE 1  
=====

XF(2)= .00000E+00

XF(3)= .25000E+03

LIAISON 3 FORCE APPLIQUEE SUR LE SOLIDE 2  
=====

XF(1)= .00000E+00

XF(2)= .00000E+00

XF(3)= .50000E+03

LIAISON 3 COUPLE APPLIQUEE SUR LE SOLIDE 2  
=====

XC(1)= .00000E+00

XC(2)= -.30000E+05

XC(3)= .00000E+00

LIAISON 4 FORCE APPLIQUEE SUR LE SOLIDE  
=====

XF(1)= .00000E+00

XF(2)= .80000E+03

XF(3)= .50000E+03

LIAISON 4 COUPLE APPLIQUEE SUR LE SOLIDE  
=====

XC(1)= -.24414E-01

XC(2)= -.30000E+05

XC(3)= .00000E+00

LIAISON 5 FORCE APPLIQUEE SUR LE SOLIDE  
=====

XF(2)= .06000E+03

XF(3)= .90000E+03

LIAISON 6 FORCE APPLIQUEE SUR LE SOLIDE  
=====

XF(1)= .00000E+00

XF(2)= -.16000E+03

XF(3)= -.40000E+03

LIAISON 7 FORCE APPLIQUEE SUR LE SOLIDE  
=====

XF(2)= -.80000E+03

**TROISIEME PARTIE : ANNEXE II**

**EXEMPLE DE "LEÇON" PROGRAMMEE AVEC "EPMINE"**

**(utilisée pour l'essai de rééducation des dyslexiques)**

```

<<#DERUT>> A<<1>>
<<#1>>E <<LIS ===> GLOC          >> K <<01>>? C<<GLOC>>          K<<07>>A<<2>>;
I<<C10=C10+01>> S<<C10=07>>K<<04>> A<<11>>;J<<C01=C01+01>> S<<C01=02>>K<<03>>A<<
2>>;K<<02>> A<<1>>
<<#2>>E <<LIS ===> TRPOR          >> K <<01>>? C<<TRPOR>>          K<<07>>A<<3>>;
I<<C10=C10+01>> S<<C10=07>>K<<04>> A<<11>>;J<<C02=C02+01>> S<<C02=02>>K<<03>>A<<
3>>;K<<02>> A<<2>>
<<#3>>E <<LIS ===> MERRAN         >> K <<01>>? C<<MERRAN>>          K<<07>>A<<4>>;
I<<C10=C10+01>> S<<C10=07>>K<<04>> A<<11>>;J<<C03=C03+01>> S<<C03=02>>K<<03>>A<<
4>>;K<<02>> A<<3>>
<<#4>>E <<LIS ===> CLATZNA        >> K <<01>>? C<<CLATZNA>>          K<<07>>A<<5>>;
I<<C10=C10+01>> S<<C10=07>>K<<04>> A<<11>>;J<<C04=C04+01>> S<<C04=02>>K<<03>>A<<
5>>;K<<02>> A<<4>>
<<#5>>E <<LIS ===> DIBTONLU       >> K <<01>>? C<<DIBTONLU>>          K<<07>>A<<6>>;
I<<C10=C10+01>> S<<C10=07>>K<<04>> A<<11>>;J<<C05=C05+01>> S<<C05=02>>K<<03>>A<<
6>>;K<<02>> A<<5>>
IV= 2
<<#6>>E <<LIS ===> JALXVCRAD       >> K <<01>>? C<<JALXVCRAD>>          K<<07>>A<<7>>;
I<<C10=C10+01>> S<<C10=07>>K<<04>> A<<11>>;J<<C06=C06+01>> S<<C06=02>>K<<03>>A<<
7>>;K<<02>> A<<6>>
<<#7>>E <<LIS ===> VEITROZUBR     >> K <<01>>? C<<VELTROZUBR>>          K<<07>>A<<11>>;
I<<C10=C10+01>> S<<C10=07>>K<<04>> A<<11>>;J<<C07=C07+01>> S<<C07=02>>K<<03>>A<<
11>>;K<<02>>A<<7>>
<<#11>>F <<LIS ===> RA B IR        >> K<<01>>?C<<RA B IR>>          K<<07>>A<<12>>;
I<<C20=C20+01>>S<<C20=07>>K<<04>>A<<21>>;J<<C11=C11+01>>S<<C11=02>>K<<03>>A<<12>>
>;K<<02>>A<<11>>
<<#12>>E <<LIS==>> BO VS T       >> K<<01>>?C<<BO VS T>>          K<<07>>A<<13>>;
I<<C20=C20+01>>S<<C20=07>> K<<04>>A<<21>>;J<<C12=C12+01>>S<<C12=02>>K<<03>>A<<13>
>>;K<<02>>A<<12>>
<<#13>>E <<LIS ===> C RLA DU      >>K<<01>>?C<<C RLA DU>>          K<<07>>A<<14>>;
I<<C20=C20+01>>S<<C20=07>>K<<04>>A<<21>>;J<<C13=C13+01>>S<<C13=02>>K<<03>>A<<14>
>>;K<<02>>A<<13>>
<<#14>>E <<LIS==>> COR DL PI     >>K<<01>>?C<<COR DL PI>>          K<<07>>A<<15>>;

```

IV= 3

I<<C20=C20+01>>S<<C20=07>>K<<04>>A<<21>>;I<<C14=C14+01>>S<<C14=02>>K<<03>>A<<15>>  
 >;K<<02>>A<<14>>  
 <<#15>>E<< LIS ===> PU NA CRO >>K<<01>>?C<<PU NA CRO>> K<<07>>A<<16>>;  
 I<<C20=C20+01>>S<<C20=07>>K<<04>>A<<21>>;I<<C15=C15+01>>S<<C15=02>>K<<03>>A<<16>>  
 >;K<<02>>A<<15>>  
 <<#16>>E<< LIS ===> VER Z0 BAL >>K<<01>>?C<<VER Z0 BAL>> K<<07>>A<<17>>;  
 I<<C20=C20+01>>S<<C20=07>>K<<04>>A<<21>>;I<<C16=C16+01>>S<<C16=02>>K<<03>>A<<17>>  
 >;K<<02>>A<<16>>  
 <<#17>>E<< LIS ===> FRA CL OURD >>K<<01>>?C<<FRA CL OURD>> K<<07>>A<<21>>;  
 I<<C20=C20+01>>S<<C20=07>>K<<04>>A<<21>>;I<<C17=C17+01>>S<<C17=02>>K<<03>>A<<21>>  
 >;K<<02>>A<<17>>  
 <<#21>>E<< LIS ===> FJ D AL >> K<<01>>?C<<FJ D AL>> K<<05>>A<<22>>;  
 I<<C30=C30+01>>S<<C30=07>>K<<04>>A<<30>>;I<<C21=C21+01>>S<<C21=02>>K<<03>>A<<22>>  
 >;K<<02>>A<<21>>  
 <<#22>>E<< LIS ===> N MR UN >> K<<01>>?C<<N MR UN>> K<<05>>A<<23>>;  
 I<<C30=C30+01>>S<<C30=07>>K<<04>>A<<30>>;I<<C22=C22+01>>S<<C22=02>>K<<03>>

IV= 4

A <<23>>;K<<02>> A<<22>>  
 <<#23>>E<< LIS ===> ART FL V >> K<<01>>?C<<ART FL V>> K<<05>>A<<24>>;  
 I<<C30=C30+01>>S<<C30=07>>K<<04>>A<<30>>;I<<C23=C23+01>>S<<C23=02>>K<<03>>A<<24>>  
 >;K<<02>>A<<23>>  
 <<#24>>E<< LIS ===> PRE DL AB >> K<<01>>?C<<PRE DL AB>> K<<05>>A<<25>>;  
 I<<C30=C30+01>>S<<C30=07>>K<<04>>A<<30>>;I<<C24=C24+01>>S<<C24=02>>K<<03>>A<<25>>  
 >;K<<02>>A<<24>>  
 <<#25>>E<< LIS ===> ZON PT AV >>K<<01>>?C<<ZON PT AV>> K<<05>>A<<26>>;  
 I<<C30=C30+01>>S<<C30=07>>K<<04>>A<<30>>;I<<C25=C25+01>>S<<C25=02>>K<<03>>A<<26>>  
 >;K<<02>>A<<25>>  
 <<#26>>E<< LIS ===> EUR HA REUR >> K<<01>>?C<<EUR HA REUR>>K<<05>>A<<27>>;  
 I<<C30=C30+01>>S<<C30=07>>K<<04>>A<<30>>;I<<C26=C26+01>>S<<C26=02>>K<<03>>A<<27>>  
 >;K<<02>>A<<26>>  
 <<#27>>E<< LIS ===> IN TRO FOIR >>K<<01>>?C<<IN TRO FOIR>>K<<05>>A<<30>>;  
 I<<C30=C30+01>>S<<C30=07>>K<<04>>A<<30>>;I<<C27=C27+01>>S<<C27=02>>K<<03>>A<<30>>  
 >;K<<02>>A<<27>>

IV= 5

<<#30>> E<< VOILA UN AUTRE EXERCICE: TU VERRAS UN GROUPE DE SIGNES ET TU DEVRAS LE RETENIR. PUIS TU VERRAS PLUSIEURS GROUPE DE SIGNES ET TU ECRIRAS LES NUMEROS DES GROUPE IDENTIQUES A CELUI QUE TU AURAS RETENU. PAR EXEMPLE, SI JE VOIS LE GROUPE: AL PUIS LES GROUPE AL LA AL LA LA AL, JE REPONDRAS: 1,3,6 SUIVI DU DIEZE DE FIN DE REPONSE. AS-TU COMPRIS (OUI OU NON)? >>? C<<OUI>> E<< A TOI DE JOUER.>> AL<<31>>;C<<NON>>K<<10>>A<<30>>;K<<11>>A<<30>>  
 <<#31>>E<< REGARDE ==>> ( > > K<<09>>E<< > ) ( > . ) ( > >>?C<<2,4>>K<<06>>  
 >A<<32>>;I<<C31=C31+01>>S<<C31=02>>K<<03>>A<<32>>;K<<02>>A<<31>>  
 <<#32>>E<< REGARDE ==>> =, >> K<<09>>E<< =X, X=, =X, =X >>?C<<1,3>>  
 K<<06>>A<<33>>;I<<C32=C32+01>>S<<C32=02>>K<<03>>A<<33>>;K<<02>>A<<32>>  
 <<#33>>E<< REGARDE ==>> <+? >> K<<09>>E<< +<? <+? <+? ?<+ >>?C<<2,3>>  
 K<<06>>A<<34>>;I<<C33=C33+01>>S<<C33=02>>K<<03>>A<<34>>;K<<02>>A<<33>>  
 <<#34>>E<< REGARDE ==>> )X\*( >>K<<09>>E << )X\*( (X\*) )\*( )X\*( >>?C<<1,4>>K<<06>>A<<35>>;I<<C34=C34+01>>S<<C34=02>>K<<03>>A<<35>>;K<<02>>A<<34>>  
 <<#35>>E<< REGARDE ==>> IA >>K<<09>>E<< AI IA AI IA AI >>?C<<2,4>>  
 K<<06>>A<<36>>;I<<C35=C35+01>>S<<C35=02>>K<<03>>A<<36>>;K<<02>>A<<35>>

IV= 6

<<#36>>E<< REGARDE ==>> CAR >>K<<09>>E<< CAR CRA CRA CAR CRA >>?C<<1,4>>K<<06>>A<<37>>;I<<C36=C36+01>>S<<C36=02>>K<<03>>A<<37>>;K<<02>>A<<36>>  
 <<#37>>E<< REGARDE ==>> LION >>K<<09>>E<< LOIN LION LION LOIN LION >>?C<<2,3,5>>K<<06>>A<<38>>;I<<C37=C37+01>>S<<C37=02>>K<<03>>A<<38>>;K<<02>>A<<37>>  
 >  
 <<#38>>E<< REGARDE ==>> FOIR>>K<<09>>E<< FROI FROI FOIR FROI FOIR>>?C<<3,5>>K<<06>>A<<40>>;I<<C38=C38+01>>S<<C38=02>>K<<03>>A<<40>>;K<<02>>A<<38>>  
 <<#40>> E<<

RECHERCHE DE MOTS

\*\*\*\*\*

TU VERRAS UN MOT...PUIS PLUSIEURS MOTS. TU DONNERAS LES NUMEROS DES MOTS IDENTIQUES AU MOT VU AU DEBUT.

AS TU-COMPRIS (OUI OU NON)?>> ? C<<OUI>> K<<08>>A<<41>>;C<<NON>>K<<

10>>A<<40>>;K<<11>>A<<40>>

<<#41>>F<<RETIENS CE MOT ==>> VERTE>>K<<09>>E<< VRETE VERT VERTE VERTES>>?C<<3>>K<<05>>A<<42>>;I<<C41=C41+01>>S<<C41=02>>K<<03>>A<<42>>;K<<02>>A<<41>>

I' = 7

<<#42>>E<< REGARDE CE MOT ==> POINTE>>K<<09>>E<< POITE POINTE PIONTE PO  
 INTER>>?C<<2>>K<<05>>A<<43>>;I<<C42=C42+01>>S<<C42=02>>K<<03>>A<<43>>;K<<02>>A  
 <<42>>  
 <<#43>>E<< RETIENS CE MOT ==> COUSIN>>K<<09>>E<< COUSIN COUSSIN COUSSIN COUS  
 IN>>?C<<1,4>>K<<05>>A<<44>>;I<<C43=C43+01>>S<<C43=02>>K<<03>>A<<44>>;K<<02>> A  
 <<43>>  
 <<#44>>E<< REGARDE ==> BOUILLON>>K<<09>>E<< BOULLION BOUILLON BOUILLON BOULL  
 ON>>?C<<2,3>>K<<05>>A<<45>>;I<<C44=C44+01>>S<<C44=02>>K<<03>>A<<45>>;K<<02>>A  
 <<44>>  
 <<#45>>E<< RETIENS ==> DEVASTER>>K<<09>>E<< DEVASTE DEVASTERA DEVASTE DEVAST  
 ER>>?C<<4>>K<<05>>A<<50>>;I<<C45=C45+01>>S<<C45=02>>K<<03>>A<<50>>;K<<02>>A<<45>  
 >  
 <<#50>>

E<<

APPLICATION DE REGLES D'ACCORD

\*\*\*\*\* \*\* \*\*\*\*\* \*\*\*\*\*

REGLE: COMPLETE PAR + LES GROUPES DE SIGNES SE TERMINANT PAR <

I' = 8

\*\*\*\*\*  
 AS-TU COMPRIS (OUI OU NON) ? >>? C<<OUI>> K<<05>>A<<51>>;C<<NON>>K  
 <<10>>A<<50>>;K<<11>>A<<50>>  
 <<#51>>K<<12>>F<< )=+ <X> ?\*"< +(>?< >>?C<<)>=+ <X> ?\*"<+ +(>  
 ?<+>>K<<07>>A<<52>>;I<<C51=C51+01>>S<<C51=02>>K<<03>>A<<52>>;K<<02>>A<<51>>  
 <<#52>>K<<12>>F<< X<)< X<? =)+(< ?<X- >>?C<<X<)<+ X<? =)+(<+  
 <?<X>+ ?<X->>K<<05>>A<<53>>;I<<C52=C52+01>>S<<C52=02>>K<<03>>A<<53>>;K<<02>>A  
 <<52>>  
 <<#53>> E<<NOUVELLE REGLE: COMPLETE PAR X+ LES GROUPES DE SIGNES SE TERMINANT  
 PAR ) AS-TU COMPRIS (OUI OU NON) ?>>?C<<OUI>>K<<06>>A<<54>>;C<<NON>>K<<10>>  
 A<<53>>;K<<11>>A<<53>>  
 <<#54>>K<<12>>F<< )=+ ?X= <X\*( \*(> )>>?C<<)>=+ )X+ ?X= <X\*(  
 \*(> )X+>> K<<08>>A<<55>>;I<<C54=C54+01>>S<<C54=02>>K<<03>>A<<55>>;K<<02>> A  
 <<54>>  
 <<#55>>K<<12>>F<< ?X=( +\*(> +(>)\*< =+>)?>>?C<<?X=( +\*(> )X+ +(>  
 >\*< =+>)?>>X+>>K<<08>>A<<60>>;I<<C55=C55+01>>S<<C55=02>>K<<03>>A<<60>>;

IV= 9

K<<02>>A<<55>>  
 <<#60>>

E&lt;&lt;

ACCORD DES ADJECTIFS

=====

CORRIGE LES ADJECTIFS SI TU PENSES QUE C'EST NECESSAIRE.>>E<< ON ECR  
 IT UN VENT.....>>F<< VIOLENT ET CHAUDS>>?C<<VIOLENT ET CHAUD>>K<<07>>  
 A<<61>>;I<<C60=C60+01>>S<<C60=02>>K<<03>>A<<61>>;K<<02>>A<<60>>  
 <<#61>>E<< ON ECRIT DES ARBRES.....>>E<< ROUGES DE FRUITS>>?C<<ROUGES DE FRU  
 ITS>>K<<07>>A<<62>>;I<<C61=C61+01>>S<<C61=02>>K<<03>>A<<62>>;K<<02>>A<<61>>  
 <<#62>>E<< ON ECRIT...UN CLIMAT.....>>E<< HUMIDES ET MALSAINS>>?C<<HUMIDE ET  
 MALSAIN>>K<<07>>A<<63>>;I<<C62=C62+01>>S<<C62=02>>K<<03>>A<<63>>;K<<02>>A<<62>>  
 <<#63>>E<< ON ECRIT...DES PLUIES.....>>E<< TORRENTIELLE >>?C<<TORRENTIELLES>>  
 K<<07>>A<<64>>J<<C63=C63+01>>S<<C63=02>>K<<03>>A<<64>>;K<<02>>A<<63>>  
 <<#64>>E<< ON ECRIT...DEUX VOITURES.....>>E<< ACCIDENTES ET IRREPARABLE>>?  
 C<<ACCIDENTEES ET IRREPARABLES>>K<<07>>A<<65>>;I<<C64=C64+01>>S<<C64=02>>K  
 <<03>>A<<65>>;K<<02>>A<<64>>

IV=10

<<#65>>E<< ON ECRIT...LES ENFANTS FORMAIENT UN GROUPE.....>>E<< ANIMES ET BAVARD  
 >>?C<<ANIME ET BAVARD>>K<<07>>A<<65>>;I<<C65=C65+01>>S<<C65=02>>K<<03>>A<<65>>;  
 K<<02>> A<<65>> <<#65>>

**TROISIEME PARTIE : ANNEXE III**

**NOTES BIBLIOGRAPHIQUES**

- /1/ - Documents I.B.M. : " Research Reports" janvier 1966
- /2/ - D. FENEUILLE, Université de LYON I : " L'enseignement assisté par ordinateur - bibliographie - " ( février 1971 )
- /3/ - D. KAISER et D. COULON, Institut de programmation de l'Université de PARIS VI : " Contribution de l'informatique à l'évolution des méthodes d'éducation". Contrat C.R.I. numéro 72/27 ( Mai 1973)
- /4/ - Ministère de l'Education du Québec : " Rapport sur l'utilisation de l'ordinateur à des fins pédagogiques - tomes I,II,III et IV -" ( 1973 )
- /5/ - M. PEUCHOT, Advisory systems engineer, I.B.M. France : " Instruction module generator : user's guide " ( 1974 )
- /6/ - I.B.M. : " le langage COURSWRITER"
- /7/ - C. BELLISSANT, Thèse de 3ème cycle de la Faculté des Sciences de Grenoble " Conception et compilation d'un langage pour l'écriture de cours " ( mars 1970)
- /8/ - D. FENEUILLE, Université de LYON I : " Etude, conception et réalisation d'un système d'enseignement assisté par ordinateur sur petite machine ", thèse de Doctorat de 3ème cycle ( Décembre 1973)
- /9/ - C. HEWITT : " PLANNER - A langage for proving theorems in robots ". Proceedings of the International Joint Conference in Artificial Intelligence, pp 295 - 301 - ( mai 1969)
- /10/ - J. CARBONNEL : " SCHOLAR - A new approach to Computer - Assisted - Instruction " , Naval Research review ( octobre 1971)

- /11/ - S. C. SHAPIRO : " The MIND system : a data structure for semantic information processing " report R-837-PR, the Rand Corporation ( Aout 1971)
- /12/ - S. C. SHAPIRO : " A net structure for semantic information storage, deduction and retrieval". Proceeding of the international joint conference in artificial intelligence ( septembre 1971)
- /13/ - J. D. WEXLER : " Generative teaching system and data structures " - States University of New-York, report 9/71, Department of Computer Science (Juillet 1971)
- /14/ - J. HEBEINSTREIT, Ecole Supérieure d'Electricité de PARIS : " recherches sur l'introduction de l'informatique dans la pédagogie des sciences physiques dans l'enseignement secondaire ". Premier rapport d'activité (janvier 1973)
- /15/ - S. PAPERTE and C. SALOMON , Artificial Intelligence Laboratory : " twenty things to do with a computer", Artificial intelligence mémo 248, logo Mémo n° 3 ( juin 1971)
- /16/ - J.J. ALLAN III and J.J. LAGOWSKI, University of Texas at Austin : "Project C - BE : Computer Based Science and engineering education, général information " ( 1972)
- /17/ - WORKING' GROUP ON COMPUTERS IN INSTRUCTION, I.F.I.P. Technical Committee for Education : " Use of the computer in teaching and learning " ( Août 1974)
- /18/ - J. R. HARTLEY and D. H. SLEEMAN , University of Leeds : " Towards more intelligent teaching systems " , research councils computer based learning project ( 1972)
- /19/ - H. FRANK : " Aspects cybernétiques de l'apprentissage et de l'enseignement" Cybernetica vol. IX, n° 3, pages 196 et 197 ( .1966)
- /20/ - H. FRANK : " ALZUDI, exemple de didactique formelle" , Bulletin du Greti n° 5 ( novembre 1967)
- /21/ - I.K. DAVIE, L. THOMAS, D. OPENSAW, J.B. BIRD : " Programmed learning", in perspective, London City Publicity Service, L t d ( 1963)  
voir aussi GAVINI , "Manuel de formation aux techniques de l'enseignement programmé " - DUNOD éditeur ( 1959)

- /22/ - LE XHUAN : " Cours d'analyse comportementale " , U.E.R. de Didactique des disciplines scientifiques, Université de PARIS VII
- /23/ - F. MECHNER, University of Chicago III, "Behavioral Analysis and Instructional sequencing" in Programmed Instruction, the 66 th yearbook of the national society for the study education, Part II , chapitre IV, pages 81-103 ( 1967)
- /24/ - J. REGNIER et M. MONTMOLIN : " Reconnaissance de l'organisation, recherche de l'ordonnancement et choix du mode d'enseignement de la matière", Actes du colloque de l'O.T.A.N. de Nice ( mai 1968)
- /25/ - J. PERRIAULT : " problèmes d'organisation didactique d'une matière à enseigner " , séminaires de l'U.E.R. de didactique des disciplines scientifiques, Université de PARIS VII ( juin 1972)
- /26/ - L.N. LANDA : " diagnostic et enseignement programmé - théorie des algorithmes - ", 4ème conférence internationale sur l'enseignement programmé et les machines à enseigner , Dusseldorf ( 1966)
- /27/ - L. N. LANDA : " le problème des algorithmes et l'enseignement aux élèves des méthodes de pensée rationnelle" , in Voprossi Psichologii n° 1, pages 103 - 118 ( 1961)
- /28/ - Cl. SAYETTAT : " Contribution à la conception, l'élaboration et l'utilisation du système de bases de données SOMINE : accès au système - aide à la C.A.O. " - Thèse de Doctorat de 3ème cycle, Ecole Nationale Supérieure des Mines de Saint-Etienne (janvier 1 976)
- /29/ - P. MARTY : " Contribution à la conception , l'élaboration et l'utilisation du système de bases de données SOMINE : structuration des informations " Thèse de Doctorat de 3ème cycle, Ecole Nationale Supérieure des Mines de Saint-Etienne (janvier 1 976)
- /30/ - R. MAHL, Professeur à l'Ecole Nationale Supérieure des Mines de Saint-Etienne : " Algorithmique et structure de données " (1973)
- /31/ - G. LOUIS - GAVET : " Compactage de données structurées. Contribution à la conception d'un système d'informations composé de fichiers multiples et volumineux " , thèse de doctorat d'Etat de l'Université de LYON I ( Juin 1974)
- /32/ - J. D. WEXLER , State University of New York at Buffalo : " A design for describing (elementary ) programming problems generators in an automatic teaching system " ( décembre 1973)

- /33/ - A. M. BORK, Department of physics, University of California : "Computer - based mechanics ", Irvine - California ( 1972)
- /34/ - G. JOUANDIN et MAUREL, professeurs de physique : " système informatique de résolution des problèmes scientifiques et techniques ( exemples en statique) " , Bulletin de l' I.N.R.D.P., rue d'Ulm - Paris ( 1973)
- /35/ - W. R. UTTAL, T. PASICH, T. ROGERS and R. HERANGMUS : " Generative Computer Assisted Instruction" , Communication 243, Mental Health Research institute, University of Michigan, Ann Arbor ( janvier 1969)
- /36/ - J. C. LATOMBE, Ingénieur I. E. G. et I.M.A.G. : " Elaboration d'un système pédagogique d'assistance à la conception en électrotechnique", thèse de docteur ingénieur de l'Université de Grenoble ( 1972)
- /37/ - F. ADAM, R. JACOUD, M. JACQUES, J.J. KESSIS, J.J. PELLETREAU, J. PENNE, J.P. RAMBAULT et J.P. TOULOUSE, Laboratoire O.P.E. 2 de l'Université de PARIS VII : " A current C.A.I. project : the O.P.E. system " in information processing 74 - North Holland Publishing company (1974)
- /38/ - M. LOBROT : " Troubles de la langue écrite et remèdes ", collection des sciences de l'éducation, Editions sociales de France (1972)
- /39/ - M. W. PETRUK et E.N. SKAKUV ,université d'Alberta, U.S. : " l'oculomètre à infra-rouge : description et applications en Enseignement Assisté par Ordinateur", Annale de la Conférence sur " INFORMATIQUE ET ENSEIGNEMENT " organisée par l'I.F.I.P. et l'A.F.C.E.T. à Marseille - North Holland Publishing Company, AMSTERDAM (1975)
- /40/ - Annale de la Conférence sur "INFORMATIQUE et ENSEIGNEMENT" organisée par l'I.F.I.P. et l'A.F.C.E.T. à Marseille - North Holland Publishing Company, AMSTERDAM - Tomes I et II ,(1975)
- /41/ - M. JACQUE; Université de PARIS VII : " Le Système O.P.E. " dans Acta Electronica , pages 319-327 (Mars 1974)
- /42/ - Laboratoire Commun des Techniques Audio-visuelles et des Moyens Modernes d'Enseignement de l'Ecole Polytechnique et de l'Ecole Nationale Supérieure des Télécommunications : "le système SODOME"



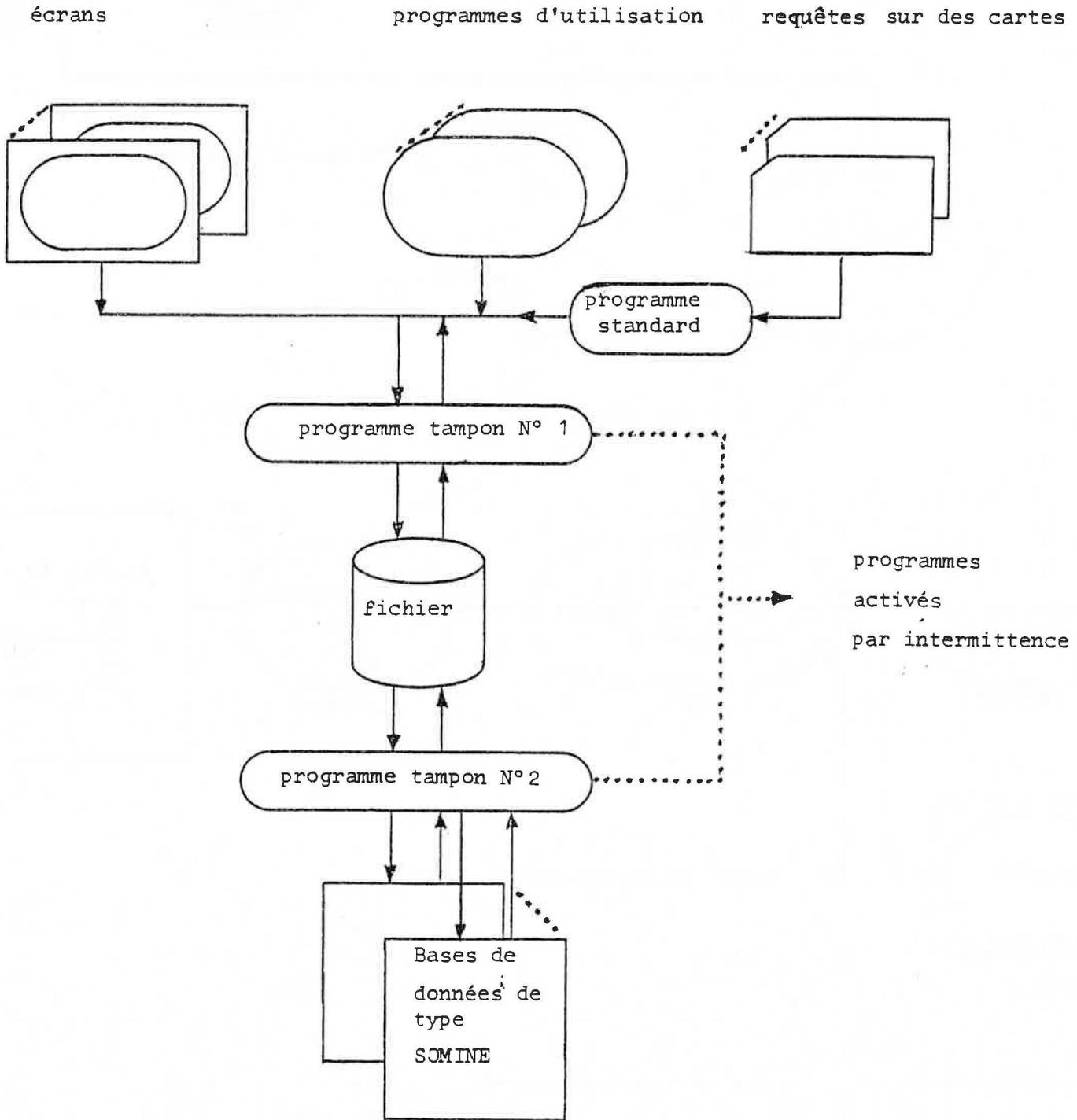
## QUATRIEME PARTIE

● CONCLUSION GENERALE

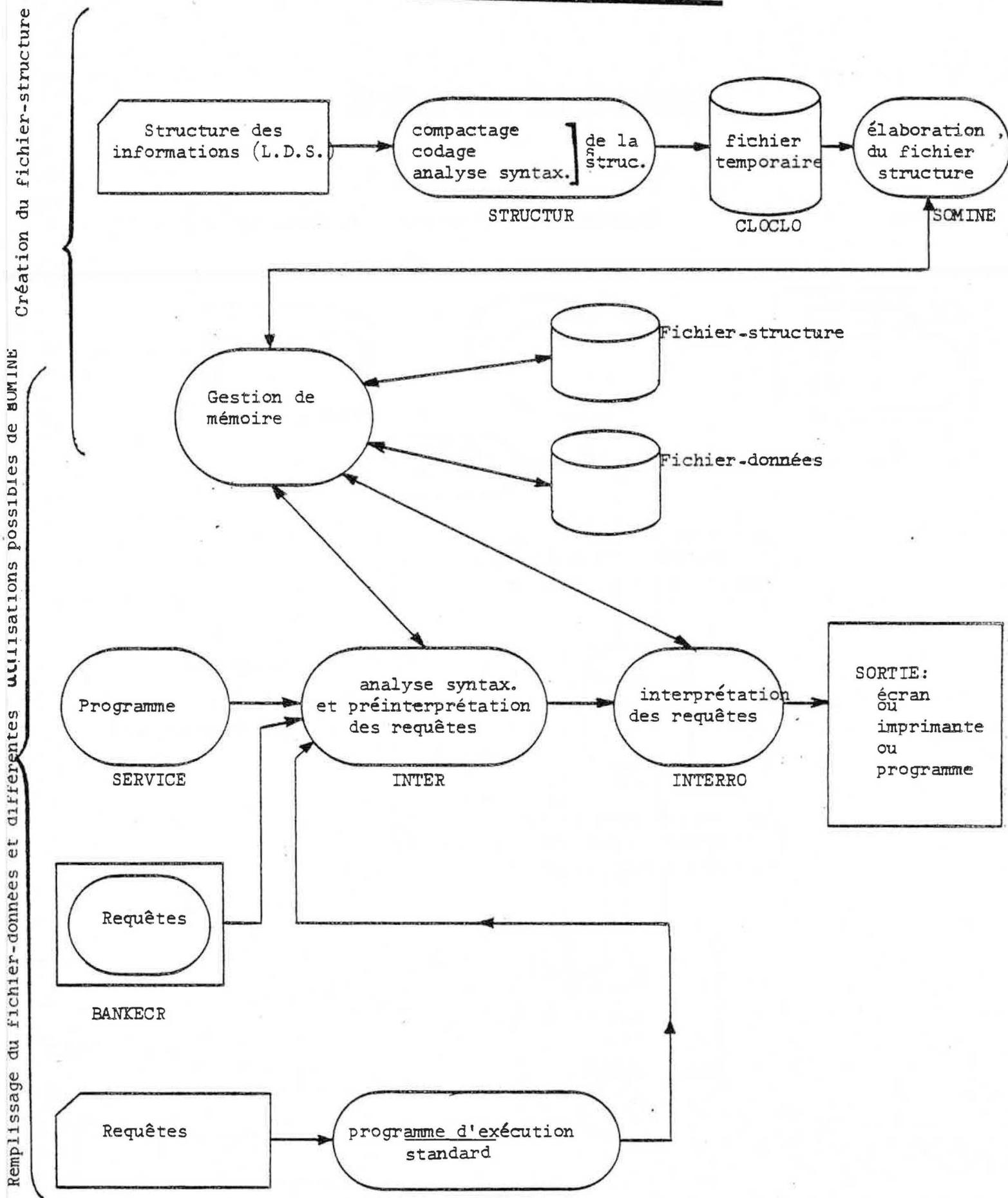
● ANNEXE : Fiche signalétique de SOMINE

Voici venu le moment de faire la synthèse et le bilan des travaux que nous avons effectués et dont une partie fait l'objet de cette thèse. C'est le système de bases de données SOMINE qui constitue l'essentiel de nos travaux, nous allons le schématiser et le concrétiser avec les figures des deux pages suivantes. La première représente l'utilisation d'un système de bases de données de type SOMINE, la deuxième illustre le fonctionnement d'une de ces bases depuis l'étape de création jusqu'aux différentes possibilités d'utilisation.

SYSTEME DE BASE DE DONNEES SOMINE



UNE BASE DE DONNEES SOMINE



En réalisant le système de bases de données SOMINE, nous avons essayé de résoudre le problème de l'optimisation de l'organisation et de la recherche des informations appartenant à une collection d'informations.

Par certains côtés ce souci de discipline dans l'organisation et la structuration des informations en vue d'approcher d'une connaissance n'est pas sans rappeler la structure même du cerveau humain. On peut aussi penser aux philosophies orientales qui par des postures physiques et mentales essayent d'atteindre la connaissance et le bonheur.

Nous nous sommes donc attachés à trouver une philosophie applicable aux bases de données et assurant la vie de la collection d'informations qu'elles gèrent. Cette vie devra être conforme aux objectifs que nous nous étions fixés dans l'introduction de ce rapport, la nécessité de quelques autres critères s'est aussi dégagée au cours de notre travail.

Le premier objectif retenu est essentiel puisque c'est de lui que découle la notion même de base de données. Il s'agit d'assurer un traitement varié et efficace des informations contenues dans la base. Nous pouvons développer un peu les principales actions qui devront être réalisables pour que le traitement des informations soit satisfaisant. La base devra assurer :

- le placement des informations dans les mémoires auxiliaires réservées à cet effet.

- la recherche et la découverte rapide d'une information sur ces mêmes supports.

- enfin, et c'est ce point qui permet de juger de l'efficacité d'une base, les informations doivent pouvoir être manipulées dans le but d'en créer de nouvelles. Ces créations peuvent être faites de plusieurs façons, nous pouvons citer les calculs sur les éléments de la base ainsi que les informations composées par coordination ou subordination des éléments de la base. Enfin des applications telles que la concaténation d'informations ou les filtres sur les informations permettent de juger de l'aptitude à donner naissance à de nouvelles informations obtenues à partir de celles qui sont dans la base.

Si le traitement des informations est assuré par une base, on peut se fixer d'autres objectifs permettant d'en faciliter l'utilisation ou d'en augmenter les qualités. Les critères suivants nous ont semblés intéressants à retenir :

- indépendance du système par rapport au matériel utilisé ceci afin d'en assurer la transportabilité.
- indépendance de la collection d'informations par rapport aux utilisateurs ( personnes ou programmes )
- adaptabilité à de nouvelles exigences .
- fiabilité et protection de la base .
- mesurabilité du système en vue d'améliorer ses performances ou de modifier la structure des informations.

Ces critères que nous avons retenus sont le fruit de notre réflexion et de notre expérimentation . Nous avons également consulté " l'étude sur les softwares de gestion et d'interrogation des fichiers " publication éditée par la délégation à l'informatique groupe " systèmes d'informations " (avril 1973).

Ce document constitue une très bonne étude comparative des différentes bases de données. Il se termine en donnant pour chacun des systèmes de bases de données étudié une fiche signalétique. Nous avons retenu ce modèle de fiche signalétique pour SOMINE, nous l'avons complété et placé en annexe de cette conclusion.

Le lecteur pourra s'y reporter pour connaître des détails techniques mais nous allons faire ici un bref rappel des originalités ou des qualités du produit SOMINE; dans cette énumération nous aurons présent à l'esprit les objectifs que nous nous étions fixés.

- Naturellement les bases SOMINE peuvent effectuer les actions de rangement et de recherche des informations qui leur sont confiées. Nous avons vu dans la partie III que ceci est possible grâce aux requêtes de base dont la forme est simple et logique et dont l'apprentissage est rapide.

- La mise en oeuvre des filtres (caractéristique inverse) et des requêtes booléennes assurent une bonne faculté de création d'informations non contenues explicitement dans la base. Toute requête contient une citation hiérarchique elle assure donc la naissance d'informations composées par subordination, quant aux informations composées par coordination elles peuvent être obtenues par les requêtes groupées ou celles contenant le séparateur "tout". Une assez grande possibilité de manipulation des informations est laissée à l'utilisateur qui pourra les mettre en oeuvre dans ses programmes d'utilisation.

- Le système peut être amélioré assez facilement car la programmation des différentes parties est modulaire. De plus les automates de définition sont simples et une ou plusieurs transitions entre deux états sont toujours possibles à programmer. En somme nous pouvons dire que SOMINE est un produit évolutif, c'est-à-dire que le rajout d'une nouvelle fonction de manipulation, d'un nouveau type de caractéristique ou d'un nouveau mode de recherche ne brise en rien l'unité du système. On en trouvera une illustration dans la partie IV du rapport de Cl. SAYETTAT.

- Puisque ce logiciel est adaptatif il est perfectible. Naturellement la programmation n'en est pas parfaite et une optimisation peut toujours être envisagée, mais le plus intéressant est la possibilité d'effectuer une programmation plus fine (en assembleur par exemple) de certaines parties du système. Ceci ne peut être envisagé nous semble-t-il que dans le cadre d'une application donnée et pour un matériel donné. En effet généraliser cette remarque irait à l'encontre de l'objectif suivant qui est :

- l'indépendance vis-à-vis du matériel utilisé. Pour Somine celle-ci est assurée par la programmation effectuée presque totalement en Fortran d'où la transportabilité du produit.

- rappelons que les programmes d'implémentation de la structure et les programmes d'utilisation et d'analyse des requêtes sont totalement indépendants, ceci évite d'avoir à recréer le fichier des informations si les traitements qu'elles subissent varient.

- SOMINE est un système mesurable sous deux aspects. D'une part l'évaluation du temps d'exécution de l'implémentation ou de l'utilisation d'une base peut être faite facilement. D'autre part des compteurs situés dans la structure permettent d'espionner et de compter le nombre d'accès à chaque caractéristique. Une simple lecture de ces compteurs permet des évaluations statistiques et peut-être une modification de la structure des informations s'en déduira ce qui pourra contribuer à l'optimiser.

- du point de vue de l'utilisateur nous remarquons que les modes d'accès aux bases de type SOMINE ont été diversifiés : batch, écran, programme. Ce dernier point, l'appel par programme, est dû au fait que les requêtes sont traitées comme des chaînes alphanumériques et peuvent donc faire partie d'un programme écrit dans un langage qui permet leur traitement. Ces chaînes alphanumériques sont des phrases syntaxiquement correcte du langage de requête dont l'apprentissage est particulièrement rapide et logique.

- la protection peut être assurée facilement par un mot (ou un code) de passe contenu dans la requête ou la précédant.

- l'utilisation simultanée d'une base par plusieurs utilisateurs est réalisée par deux programmes tampons activés à des intervalles de temps réguliers et qui ont accès à un fichier qui contient les requêtes ainsi que le nom de l'utilisateur, ce fichier sert aussi de véhicule aux informations qui transitent entre la base et les utilisateurs. Ces programmes et ce fichier assurent le partage du temps entre les différents utilisateurs, les accès à la base sont déclenchés dans l'ordre dans lequel ils ont été demandés.

Ces différents rappels ont permis de montrer que SOMINE a atteint les objectifs que nous avons retenus dans l'introduction.

On peut se demander si SOMINE n'est pas un langage. Nous pensons que oui et qu'il est complémentaire des langages algorithmiques comme Fortran, Algol, PL/1 puisque les requêtes de SOMINE peuvent être traitées par ces langages. Nous ne pensons pas que SOMINE fasse double emploi avec PL/1 car la définition des structures et les différentes utilisations possibles sont dynamiques dans SOMINE. Comme atout pour notre produit nous pouvons ajouter le fait que la gestion de mémoire est tout à fait invisible à l'utilisateur. SOMINE a aussi l'avantage d'être efficace dans la recherche des données manipulées.

En effet, la structuration des informations sous forme de graphe ou d'arbre est plus riche que la plupart des structurations trouvées dans les langages de programmations usuels. Rappelons que :

- Fortran travaille sur des tableaux de taille fixe.
- Algol travaille sur des tableaux de tailles variables (la mémoire libre est gérée par le système)
- Dans PL/1 il existe des structures imbriquées mais l'allocation et la désallocation de la mémoire doivent être demandées par l'utilisateur.
- Dans le langage Pascal il existe également des structures mais elles sont statiques.

Nous espérons que ces rappels et ces comparaisons ont mis en évidence les originalités de SOMINE. La souplesse et la simplicité des accès à la base favoriseront, nous l'espérons, son utilisation dans divers domaines.

Nous allons faire maintenant la synthèse de l'aspect utilisation de l'outil SOMINE. Nos recherches se sont orientées dans trois domaines qui nous ont semblés susceptibles de nécessiter l'utilisation d'une base de données et de mettre en évidence les qualités de notre produit.

\* Domaine de la C. A . O. ( Conception Assistée par Ordinateur ) voir thèse de Cl. SAYETAT.

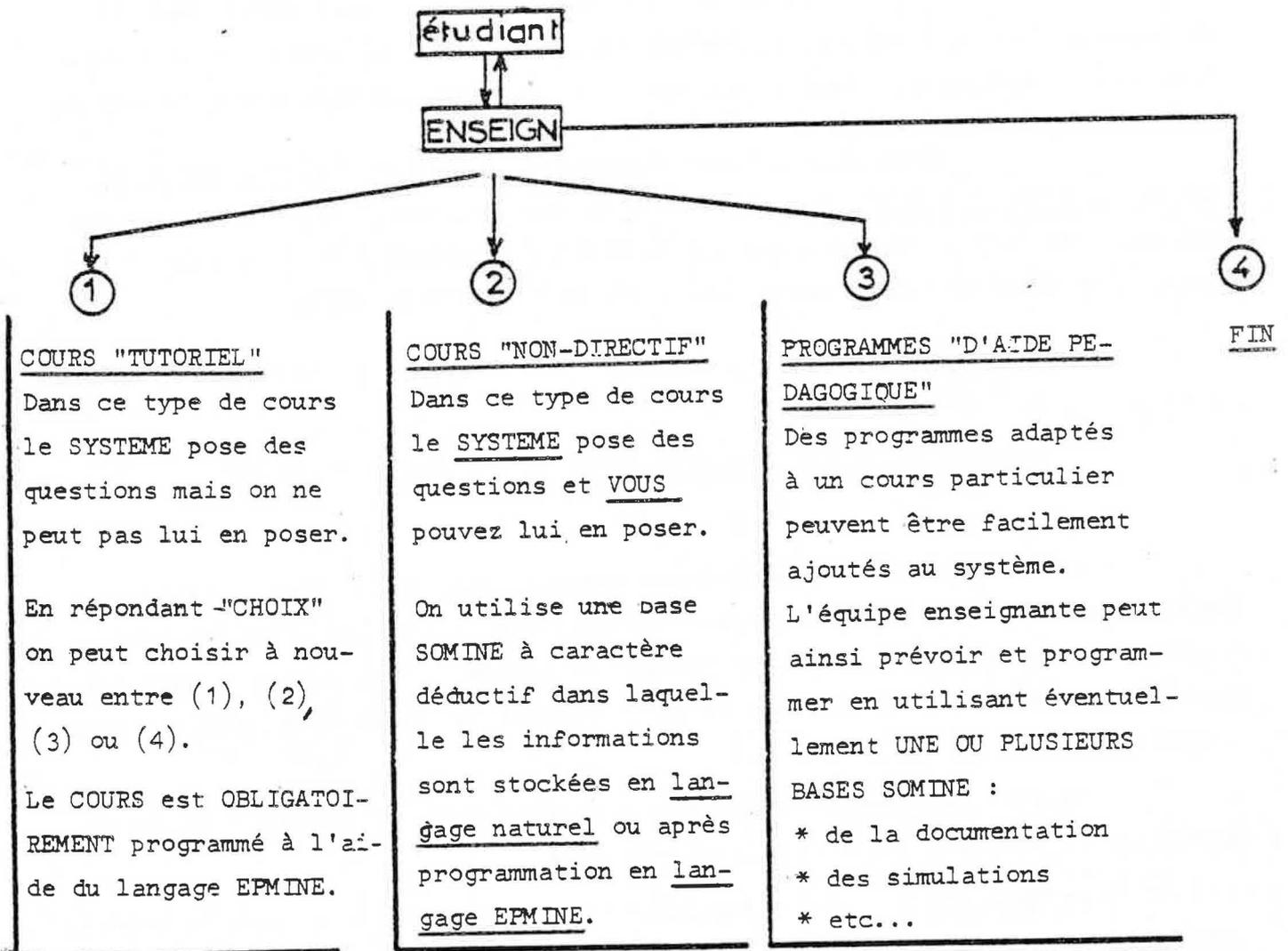
\* Domaine de l'étude des structures voir thèse de P. MARTY

Dans la troisième partie de ce mémoire, nous avons principalement relaté les conditions dans lesquelles SOMINE a été utilisé comme centre d'un système d'Enseignement Assisté par Ordinateur (E.A.O.). Nous allons rappeler les résultats obtenus dans ce domaine et les perspectives d'avenir qui en découlent actuellement.

Dans le chapitre 1, nous avons essayé de faire le point des recherches actuelles en E.A.O. et de définir nos objectifs.

Le chapitre 2, montre l'apport de l'utilisation des structures arborescentes ( graphes et arbres ) dans l'analyse des matières à enseigner. Ces structures, également utilisées dans SOMINE, ont en effet l'avantage de mettre en évidence les relations hiérarchiques, de similitude, de différence, de causalité,

etc.. susceptibles d'exister entre les éléments de matière d'un domaine d'enseignement. Un type de structure SOMINE, particulièrement simple, permet alors d'organiser ces informations avant leur stockage dans l'espace des données d'une base SOMINE. Le langage de requête, quasi-naturel, mis au point pour SOMINE permet alors la mise à jour et l'interrogation du domaine d'enseignement par les enseignants et les étudiants. Ces qualités sont cependant insuffisantes pour faire de SOMINE un outil d'enseignement complet. Pour cela, il faut encore mettre à la disposition des utilisateurs un langage de cours ( EPMINE), un interpréteur des questions des étudiants (ENSEIGN) et donner des possibilités d'extension du système par des programmes spécifiques " d'aide pédagogique" : simulation, analyse et résolution de problèmes, génération de problèmes, documentation ... etc. Ces trois compléments de SOMINE et leurs diverses fonctions apparaissent sur la figure ci-après qui montre également les choix offerts à un élève, à tout moment, au cours d'une séance de travail sur le système E.A.O.- MINE.



La définition du langage EPMINE, les problèmes posés par son implémentation et des exemples d'application sont donnés dans le chapitre 3. On y trouvera également des renseignements sur les diverses fonctions d'enseignement retenues et sur leur programmation.

Afin de tester les possibilités de tout le système E.A.O. - MINE et de commencer des recherches pédagogiques, trois applications ont été mises en oeuvre :

- enseignement des bases du langage FORTRAN au moyen de la partie du système qui utilise SOMINE pour le stockage et l'interrogation du cours. Voir à ce sujet le chapitre 4.

- enseignement de la STATIQUE : l'analyseur du problème MECANIK exemple de programme " d'aide pédagogique" (voir chapitre 5) a été utilisé par soixante élèves de deuxième année de l'I.U.T. de Saint-Etienne.

- Essai d'application de l'E.A.O. pour la rééducation d'enfants dyslexiques. Cet essai a donné des résultats encourageants et permis de dégager des axes de futures recherches interdisciplinaires. On consultera, sur ce sujet, le chapitre 6.

Cela constitue le point d'une partie du travail qu'a réalisé notre équipe depuis deux ans que l'idée du produit SOMINE est née. Tant que les travaux continuent le système évolue et nous ne manquerons pas de publier ses perfectionnements et de faire le point de l'avancement des travaux dans chacun des domaines explorés. Nous espérons que notre équipe a la souplesse et l'adaptabilité de notre produit mais nous souhaitons également posséder l'imagination et la créativité qui seules peuvent rendre notre système de bases de données utile et utilisé.

Nous pensons continuer de rechercher ou d'exploiter les voies d'utilisation ouvertes mais n'oublions pas qu'apprendre à créer et à utiliser une base SOMINE ne demande pas beaucoup de temps ni de peine. Nous espérons que cette facilité encouragera les utilisateurs éventuels que nous sommes prêts à renseigner et à former.

**ANNEXE DE LA QUATRIEME PARTIE**

**Fiche signalétique de SOMINE**

**(extraite de "l'étude sur les SOFTWARES  
de gestion et d'interrogation de Fichiers"**

**Délégation à l'informatique, groupe**

**"Systèmes d'information" -**

I - RENSEIGNEMENTS GENERAUX :

1.1. - Nom du système

SOMINE

1.2. - Réalisateurs :

Marcel GAILLARD

Pierre MARTY

Claudette SAYETTAT

Ecole Nationale Supérieure des Mines  
SAINT - ETIENNE

1.3. - Personnes responsables:

les réalisateurs

II - CARACTERISTIQUES HARDWARES ET SOFTWARES :

2. 1. - Unité centrale

- SOMINE est implémentable sur tout matériel.

2.2. - Taille mémoire centrale nécessaire

80 K octets

2.3. - Unités périphériques nécessaires :

- . lecteur de cartes
- . disques
- . imprimante
- . écrans alphanumériques ou graphiques

2.4. - Types de terminaux supportés

- . toute imprimante ou lecteur de cartes
- . tous écrans graphiques ou alphanumériques

2.5. - Nombre de terminaux supportés

limité par le temps de réponse seulement

2.6. - Taille mémoire centrale supplémentaire par terminal

non chiffrée actuellement

2.7. - Système d'exploitation utilisé

Système PHILIPS P 1000

2.8. - Langage de programmation du S.G.B.D.

SOMINE est programmé en FORTRAN ( + 50 instructions en assembleur PHILIPS)

2.9. - Taille du système

limitée par les matériels utilisés

2.10. - Processeurs nécessaires pour l'utilisation du système

- . Compilateur Fortran
- . éditeur de liens
- . système de gestion des terminaux

2.11. - Type du système

réseau

III - SOUPLESSE DU SYSTEME :

3.1. - Macrolangage de génération du système

inexistant

3.2. - Facilité d'utilisation du système

3.2.1. - Niveau requis par les utilisateurs du système

aucune compétence particulière

3.2.2. - Temps de formation de l'utilisateur

très rapide ( 2 ou 3 heures)

3.2.3. - Niveau requis par le responsable du système

niveau ingénieur

3.2.4. - Temps pour réaliser une application

Très variable suivant l'application. Il y a lieu de faire une analyse approfondie de la structuration des informations cela peut demander une dizaine de jours.

3.3. - Portabilité du système :

Peut être implémenté sur tout matériel ayant une taille mémoire suffisante.

3.4. - Comptabilité fichiers - langage de haut niveau

Accessibilité aux fichiers par langages de tous niveaux

IV - ORGANISATION DES FICHIERS :4.1. - Structure logique :

## 4.1.1. - Types de structures logiques admis

- structures arborescentes à plusieurs niveaux
- structures de graphes (caractéristiques références)

## 4.1.2. - Organisation des fichiers :

## 4.1.2.1. - Dictionnaire des fichiers :

Les fichiers n'ont plus de signification logique, seules les adresses de mots sont prises en compte. Elle sont calculées par la gestion de mémoire.

## 4.1.2.2. - Descriptif d'un fichier :

Le fichier qui contient la structure est formé d'une suite de vecteurs de 15 mots mémoire contenant les renseignements nécessaires au calcul des adresses virtuelles des caractéristiques.

## 4.1.2.3. - Organisation des données :

A cause de la gestion de mémoire (transparente à l'utilisateur) la notion de fichier disparaît dans SOMINE. Les données sont caractérisées par leurs adresses virtuelles (dans l'espace virtuel) et c'est la gestion de mémoire qui calcule les adresses réelles des données sur le disque.

## 4.1.3. - Corrélations

## 4.1.3.1 - Au niveau des fichiers :

Les adresses dans le fichier des données sont calculées à partir des

éléments contenus dans le fichier structure.

#### 4.1.3.2. - Au niveau enregistrement :

Dans SOMINE, la notion d'enregistrement n'a pas la signification habituelle. Il faudrait plutôt parler de réalisations d'entités définies par la structure des informations.

#### 4.1.4. - Langage de description

L.D.S. défini par des automates

### 4.2. - Structure physiques

#### 4.2.1. - Support physique des données :

Les données sont placées sur des disques magnétiques

#### 4.2.2. - Organisation physique des données :

Dans la mémoire virtuelle, pour chaque entité, on réserve une place correspondant au nombre maximum de réalisations. Seules les zones de mémoire virtuelle qui contiennent de l'information sont projetées sur l'espace disque. Cette projection se fait par blocs de longueur variable ( de  $2^3$  à  $2^8$  mots) appelés sous-pages.

#### 4.2.3. - Méthode d'accès :

- . Accès séquentiel (limité à une seule réalisation d'entité)
- . Accès par les caractéristiques inverses ou par les expressions booléennes de telles caractéristiques.
- . Accès direct par le numéro de réalisation.

#### 4.2.4. - Type d'adressage :

SOMINE calcule l'adresse réelle à partir de l'adresse virtuelle d'une caractéristique, il y a création de "squatters" (qui sont chainés entre eux) lorsque plusieurs adresses virtuelles donnent la même adresse réelle. La gestion de ces "squatters" est assurée par la gestion de mémoire.

## V - FONCTIONS REMPLIES PAR LE SYSTEME :

### 5.1. - Interface de création

#### 5.1.1. - Nature du processeur de création

La création se fait en "batch" il y a analyse de la structure des informations

écrite en L.D.S., cette chaîne alphanumérique est ensuite transformée en vecteurs contenant les renseignements relatifs à chaque caractéristique et qui permettront de calculer son adresse virtuelle dans l'espace virtuel du fichier-données.

#### 5.1.2. - Type de données admises :

- . chaînes alphanumériques de longueur inférieure à 1280 caractères
- . données numériques de type entier ou réel simple précision ou double précision.
- . liste de valeurs
- . chaîne d'inverse
- . pointeurs de référence
- . programmes, valeur + programme, procédure

#### 5.1.3. - Possibilité de codage et de décodage

Les données sont stockées dans leur état d'utilisation

### 5.2. - Consultation :

#### 5.2.1. - Nature du processeur de consultation

Programmes Fortran d'analyse et d'interprétation des chaînes alphanumériques que constituent les requêtes. L'action à entreprendre est déterminée par le mode de la requête : interrogation, mise à jour, création, suppression, fréquence.

#### 5.2.2. - Mode de consultation :

SOMINE permet de mode conversationnel ( écran, ) et le mode "batch".

#### 5.2.3. - Multi-accès

possibles

#### 5.2.4. - Consultation multi-fichiers

Possible

#### 5.2.5. - Bibliothèque de questions

Chaque utilisateur peut créer son propre recueil de questions dans chacun de ses programmes d'utilisation.

#### 5.2.6. - Extraction des fichiers

Très simple puisque se sont des fichiers FORTRAN à accès direct.

### 5. 3. - Mise à jour :

#### 5.3.1. - Nature du processeur de mise à jour :

Tout programme en conversationnel ou en "batch" écrit dans un langage algorithmique.

#### 5.3.2. - Mode de mise à jour :

Dans la version actuelle de SOMINE les mises à jour se font à l'aide de requêtes à l'écran ou en "batch"

#### 5.3.3. - Mise à jour au niveau des enregistrements

Cette mise à jour se fait sans problème mais le nombre maximum de réalisation d'une entité est limité à celui déclaré au moment de la définition de la structure.

#### 5.3.4. - Mise à jour au niveau des informations élémentaires :

Se fait à l'aide de requêtes ponctuelles. Des précautions doivent être prises lorsque la base contient des caractéristiques programme ( possibilité d'actions implicites)

#### 5.3.5. - Mises à jour spéciales :

Il est possible de mettre à jour des caractéristiques filtrées par une caractéristique inverse ou une expression booléenne sur des caractéristiques inverses. Signalons également la possibilité d'effectuer des mises à jour groupées ou des mises à jour à valeur variable.

#### 5.3.6. - Mise à jour de la structure logique des données

Semble nécessaire pour certains types d'applications des bases de données. Les travaux sont en cours.

### 5.4. - Possibilités de traitement :

SOMINE permet d'effectuer les requêtes de bases suivantes :

- interrogation
- mise à jour
- création
- suppression

- fréquence
- valeur

on peut facilement ajouter d'autres modes

#### 5.5. - Editions :

##### 5.5.1. - Nature du processeur d'édition

C'est l'utilisateur qui par une valeur donnée à une variable décide de la nature de l'édition :

- imprimante
- écran
- formatage ( spécification FORTRAN)

##### 5.5.2. - Support d'édition

- imprimante
- écran

##### 5.5.3. - Tri

Peut être effectué par des instructions contenues dans le programme d'utilisation écrit en langage évolué.

## VI - MAINTENANCE-SECURITE-PROTECTION-INTEGRITE

### 6.1. - Maintenance des fichiers :

#### 6.1.1. - Réorganisation, rechargement des fichiers

Sauvegarde du système utilisé

#### 6.1.2. - Statistiques d'occupation des fichiers

Non prévues dans le prototype

### 6.2. - Sécurité d'utilisation

#### 6.2.1. - Possibilités de redémarrage du système

En cas de panne, les modifications effectuées dans les quatre pages (de 257 mots) qui se trouvent dans la mémoire centrale (plan primaire) sont perdues.

#### 6.2.2 - Point de reprises

non

.../...

6.2.3. - Journal des manipulations effectuées sur les fichiers :  
pourra être mis en oeuvre si des utilisateurs en sentent la nécessité.

6.3. - Protection :

6.3.1. - Identification de l'utilisateur :

Un code de passe peut précéder ou être **contenu** dans les requêtes et filtrer ainsi les entrées dans la base.

6.3.2. - Réglementation de l'accès à l'information

Un système analogue à l'identification de l'utilisateur peut être mis en oeuvre pour réglementer l'accès à l'information. SOMINE étant encore un prototype sa réalisation dépendra de la demande.

6.4. - Intégrité des fichiers :

Les références sont chaînées automatiquement mais c'est l'utilisateur qui s'assure de la cohérence de ses-fichiers

VII - RENSEIGNEMENTS DIVERS :

7.1. - Encombrement des fichiers :

La notion de mémoire virtuelle entraîne un encombrement minimum des fichiers SOMINE

7.2. - Temps de réponse :

Variable suivant la nature du traitement. Dans le contexte actuel (multiprogrammation) il dépend de l'encombrement de la machine.

7.3. - Fiabilité :

Le produit est trop récent pour que nous donnions une réponse valable à cette question.

7.4. - Coût d'utilisation :

Par une bonne organisation du fichier-structure et une bonne formulation des requêtes, on doit **arriver** à une dépense de temps machine assez faible.

