N°478

Doctyl

Sc.N.73/111 B (1)

THESE

Présentée à
l'UNIVERSITE DE NANCY I
pour l'obtention du grade de
DOCTEUR DE 3 eme CYCLE – SPECIALITE INFORMATIQUE

SYSTEME DE GESTION D'ENSEMBLES D'INFORMATIONS SEMI - VARIABLES

Soutenue publiquement le 8 novembre 1973
devant la commission d'examen
par **Nicole MIGNOT**

UNIVERSITÉ NANCY BIBLIOTHÈQUE

JURY

Monsieur	le	Professeur	**LEGRAS**	..... President
Monsieur	le	Professeur	**MARTIN**	..... Examinateur
Madame			**ROLLAND**	..... Examinateur
Monsieur			**SAVOYSKY**	..... Invité

N°478

Doctigl

Sc.N.73/111 B (1)

THESE

Présentée à
l'UNIVERSITE DE NANCY I
pour l'obtention du grade de
DOCTEUR DE 3ème CYCLE – SPECIALITE INFORMATIQUE

SYSTEME DE GESTION D'ENSEMBLES D'INFORMATIONS SEMI - VARIABLES

Soutenue publiquement le 8 novembre 1973
devant la commission d'examen
par **Nicole MIGNOT**

UNIVERSITE NANCY BIBLIOTHEQUE

JURY

Monsieur	le	Professeur	**LEGRAS**	..... President
Monsieur	le	Professeur	**MARTIN**	..... Examinateur
Madame			**ROLLAND**	..... Examinateur
Monsieur			**SAVOYSKY**	..... Invité

T E S E

présentée à

L'UNIVERSITE DE NANCY I
pour l'obtention du grade de

DOCTEUR DE 3e CYCLE
SPECIALITE INFORMATIQUE

SYSTEME DE GESTION D'ENSEMBLES
D'INFORMATIONS SEMI - VARIABLES

Ce sont des fichiers dont les structures sont définies par chaque utilisateur et dont les articles diffèrent les uns des autres non seulement par le contenu des informations mais par la taille et le nombre de ces informations.

Soutenue publiquement

le 8 NOVEMBRE 1[illegible]

devant la Commission d'Examen

par Nicole [illegible]

Jury :

Monsieur le Professeur [illegible]	: Président
Monsieur le Professeur MARTIN	: Examinateur
Madame ROLLAND	: Examinateur
Monsieur SAVOYSKY	: Invité

UNIVERSITE NANCY BIBLIOTHEQUE

Je remercie vivement Monsieur le Professeur LECRAS, de l'Université de NANCY I, de me faire l'honneur de présider ce jury.

Qu'il trouve dans ce travail l'expression de ma profonde reconnaissance pour l'attention, les encouragements et les nombreux conseils qu'il m'a apportés pour sa réalisation.

J'exprime également mes remerciements à Madame ROLLAND, de l'Université de NANCY I et à Monsieur le Professeur MARTIN, de la Faculté de médecine qui ont bien voulu accepter d'être Membres de mon jury.

Je remercie tout spécialement Monsieur SAVOYSKY, du laboratoire Central des Ponts & Chaussées de me faire l'honneur de participer à ce jury.

Je dois beaucoup à Monsieur HAVARD, du Laboratoire de l'Equipement de NANCY, il est à l'origine de ce travail. Je lui exprime ma gratitude pour le soutien constant qu'il a bien voulu m'apporter dans cette recherche et pour la confiance dont il m'a honorée.

Je ne saurais oublier le travail de tous ceux et celle qui ont participé à la réalisation technique de cette thèse : mademoiselle SCHURRECKER pour la frappe de mon manuscrit, monsieur MEY pour les dessins, monsieur LESCASSE et les membres de son équipe pour la réalisation des organigrammes et le tirage, monsieur REMY pour la reliure de cet ouvrage.

Je leur dis à tous un grand merci.

SOMMAIRE

Introduction

1 - Présentation des possibilités offertes par le programme

11 - Introduction

12 - Présentation d'un type de structure

121 - Nombre de niveaux

122 - Représentation et définition des identificateurs

1221 - Les noeuds

1222 - Les feuilles

13 - Information "associée" à un identificateur

131 - Information "associée" à une feuille

1311 - L'information élémentaire dépend d'un noeud de niveau 2.

1312 - L'information élémentaire dépend d'un noeud de niveau 3.

132 - Information "associée" à un noeud de niveau 3

133 - Information "associée" à un noeud de niveau 2

1331 - Ce noeud de niveau 2 n'est pas suivi de noeuds de niveau 3

1332 - Ce noeud de niveau 2 est suivi de plusieurs noeuds de niveau 3

134 - Information "associée" à un noeud de niveau 1.

14 - Description de l'article

141 - les noeuds

142 - les feuilles

15 - Manipulation des tables pour obtenir une information "associée" à un identificateur

151 - exemple

152 - les tables

1521 - table de structure

1522 - table des noeuds

153 - Manipulation des informations

1531 - premier exemple

1532 - deuxième exemple

154 - Conclusion.

16 - Contrôles

161 - contrôles généraux

1611 - erreurs grossières

1612 - thésaurus

162 - contrôles spécifiques

2 - Illustration à partir d'un exemple tiré de la géologie : le sondage carotté.

21 - sa structure
211 - description des noeuds
212 - description des feuilles
213 - représentation de la structure en mémoire

22 - Traitement d'un article
221 - exemple d'article
222 - représentation de l'enregistrement en mémoire

3 - Analyse organique

31 - Introduction
32 - fichier structure ou table structure
321 - Généralités
322 - représentation externe du fichier structure : les cartes
3221 - la carte 0
32211 - les éléments variables
32212 - les éléments fixes
3222 - une carte de niveau 1
3223 - une ou plusieurs cartes de niveau 2
32231 - noeud de niveau 2 "répétitif".
32232 - présence de noeuds de niveau 3
32233 - absence de noeuds de niveau 3
32234 - remarque
3224 - une ou plusieurs cartes de niveau 3
323 - représentation interne de la table structure en mémoire centrale
3231 - les trois premiers mots
3232 - les trois mots suivants
3233 - repère de thésaurus
32331 - présentation
32332 - organisation interne en mémoire centrale
32333 - remarque
3234 - taille du fichier structure

33 - L'enregistrement : table des noeuds + article
331 - généralités
332 - description externe
3321 - niveau 1
3322 - niveau 2
33221 - noeud de niveau 2 "répétitif" et présence de noeuds de niveau 3

33222 - noeud de niveau 2 "répétitif" et absence de noeuds de niveau 3
33223 - noeud de niveau 2 non "répétitif" et présence de noeuds de niveau 3
33224 - noeud de niveau 2 non "répétitif" et absence de noeuds de niveau 3
33225 - éléments de taille variable
3323 - niveau 3
333 - représentation interne de la table des noeuds
3331 - description
3332 - les deux premiers mots
3333 - le reste
33331 - premier mot
33332 - deuxième mot
33333 - troisième mot
3334 - résumé

34 - conclusion

4 - Dossier de programmation

41 - fichier structure
411 - présentation des éléments utilisés pour former le fichier structure
4111 - schéma de travail
4112 - zones de travail
4113 - quelques particularités du programme
412 - organigramme détaillé

42 - fichier maître
421 - présentation des éléments utilisés pour former le fichier maître
4211 - schéma de travail
4212 - zones de travail
4213 - quelques particularités du programme
422 - organigramme détaillé

43 - mise à jour des dictionnaires par addition d'informations
431 - présentation des éléments utilisés pour réaliser la mise à jour
4311 - schéma de travail
4312 - zones de travail
4313 - quelques particularités du programme
432 - organigramme détaillé

44 - édition d'un fichier structure
45 - édition du fichier maître
46 - édition du fichier dictionnaire
461 - particularités du programme
462 - analyse détaillée

CONCLUSION

A mon père
Sans lequel
Ceci ne serait pas
Et grâce auquel
Je suis.

Introduction

INTRODUCTION

Dans le cadre d'un D.E.A. nous avions établi pour le Laboratoire Régional de l'Equipement de NANCY un système de traitement d'informations géologiques et géotechniques destiné aux études de Génie Civil.

Le but de ce système était de pouvoir utiliser en permanence les données des sondages déjà faits soit afin d'éviter de refaire les mêmes sondages quelques années plus tard, soit afin d'en tirer certains résultats : traitements, tris, calculs statistiques et mathématiques.

Les géologues fournissent des documents qui contiennent les données relatives à un sondage. Ces éléments sont codés, certains en "numérique" d'autres en "alphanumérique mnémotechnique". Dans ce système plusieurs dictionnaires sont introduits pour traduire ces documents : l'un relatif aux noms de roches, l'autre à leur structure, l'autre aux adjectifs, etc .....

Toutes les données ayant un caractère de similitude donnent naissance à des "types de bordereaux" et une des particularités de ce système est la grande variété des bordereaux. Il y en a 15 types :

le premier comprend : le maître d'oeuvre, le projet,
le second : la situation géographique des travaux, ...
le cinquième : la description des différentes couches atteintes (il y a un bordereau 5 par couche décrite), ...
le huitième au quinzième : tous les essais réalisés sur ces couches (ces types de bordereaux peuvent se retrouver en plusieurs exemplaires).

Ces bordereaux sont évidemment reportés sur cartes perforées.

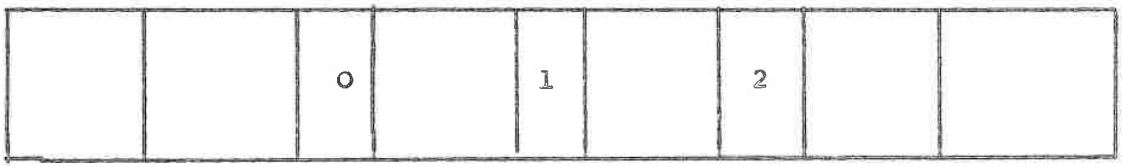
D'un sondage à l'autre, les structures des cartes varient sinon le programme de mise sur bande aurait été simplifié. Certaines cartes sont obligatoires, d'autresfacultatives, d'autres enfin peuvent se retrouver en plusieurs exemplaires.

.../...

Une particularité encore importante de ces cartes est la possibilité de contenir des informations de taille variable. Sauf les cartes 1 et 2 de longueur fixe, toutes les autres cartes ont une longueur variable.

Cette longueur variant considérablement entre deux sondages, il n'est pas question de choisir une longueur maximum que l'on remplit par des blancs si la zone n'est pas complète. Il s'agit donc de ne rentrer sur bande que ce qui est nécessaire et de faciliter une exploitation future du fichier.

Sur la bande magnétique les informations relatives à chaque sondage sont précédées d'une table. Cette table permet de repérer les informations portées sur chaque type de carte ainsi que d'autres indications complémentaires : par exemple, le nombre de couches du sondage qui permet de connaître aussitôt le nombre de cartes 5 .....



taille sondage | taille table | taille carte 0 (variable) | taille carte 1 (fixe)

A l'exception de quatre zones de cette table qui indiquent la taille du sondage, la taille de la table, le nombre de couches et la place occupée par tous les essais, chaque zone correspond à un type de carte et indique l'adresse du début des informations relatives à ce type de carte dans le sondage.

Durant l'exécution de ce programme de nombreux contrôles sont faits : tests de présence ou absence, tests de logique, plusieurs tests d'erreur. Il s'est avéré que ces contrôles sont importants.

.../...

Ils évitent les données fausses, qui une fois sur bande, pourraient falsifier les résultats ou en être omises.

Ce programme a tourné en semi-routine, l'expérience a été faite sur une centaine de sondages.

Tous les objectifs à atteindre n'étaient pas définis au départ. Nous nous sommes rendus compte que des modifications étaient nécessaires dans la présentation des cartes : certaines données manquaient, d'autres étaientredondantes, d'autres encore inutiles. Il fallait alors ou recommencer le programme adapté uniquement au premier système, ou concevoir un programme général. La deuxième solution a été adoptée.

Ce programme général est au départ réalisé pour le Laboratoire de l'Equipement. Il lui permet de recommencer son fichier une seconde fois et plus tard même unetroisième, si le besoin s'en fait ressentir, sans pour autant modifier le programme correspondant.

De plus, ce programme peut-être utile à d'autres utilisateurs ayant des fichiers de structure non classique (cf 1 : Présentation des possibilités offertes par le programme). En effet, COBOL et FORTRAN sont des langages mal adaptés à ce genre de problèmes et écrire un programme en METASYMBOL est long et fastidieux. Les utilisateurs pourraient toutefois utiliser des programmes tels que SOCRATE, MISFIIT, mais ce sont des programmes qui reviennent cher si on ne tire pas parti de toutes les possibilités offertes. Par exemple, SOCRATE est surtout orienté vers le conversationnel ce qui, dans notre cas, ne nous interesse pas.

Nous avons donc voulu avec notre programme nous placer dans une position intermédiaire : construire un programme assez souple sans aller jusqu'aux possibilités de SOCRATE ou MISFIIT. D'ailleurs SOCRATE, à l'époque où nous envisagions de généraliser notre premier programme venait d'être présenté au CNIT mais n'avait encore servi à aucun autre utilisateur que la C II.

.../...

Dans les chapitres suivants nous allons d'abord faire une présentation des possibilités offertes par notre programme puis nous en donnerons une application concrète et enfin nous aborderons la description détaillée des différentes parties du programme.

.../...

1 Présentation des possibilités offertes par le programme

I - PRESENTATION DES POSSIBILITES OFFERTES PAR LE PROGRAMME.

11 - Introduction :

Avant de décrire un type de structure conforme aux possibilites du programme, précisons quelques points :

- l'utilisateur peut, s'il le veut, former un fichier comprenant des articles dont l'organisation correspond à des types de structure différents.

La taille de la mémoire centrale de l'ordinateur étant insuffisante et les structures étant complexes, nous n'avons pas envisagé la possibilité d'avoir plusieurs types de structure présents en même temps en mémoire centrale. Si l'on a donc des articles correspondant à des types de structure différents, à chaque nouveau type de structure rencontré, on aura un échange mémoire périphérique (dans notre cas il s'agit de bande magnétique), mémoire centrale, ce qui ralentit l'exécution.

Nous verrons dans le chapitre 3 comment organiser les articles pour avoir le moins d'échanges possibles.

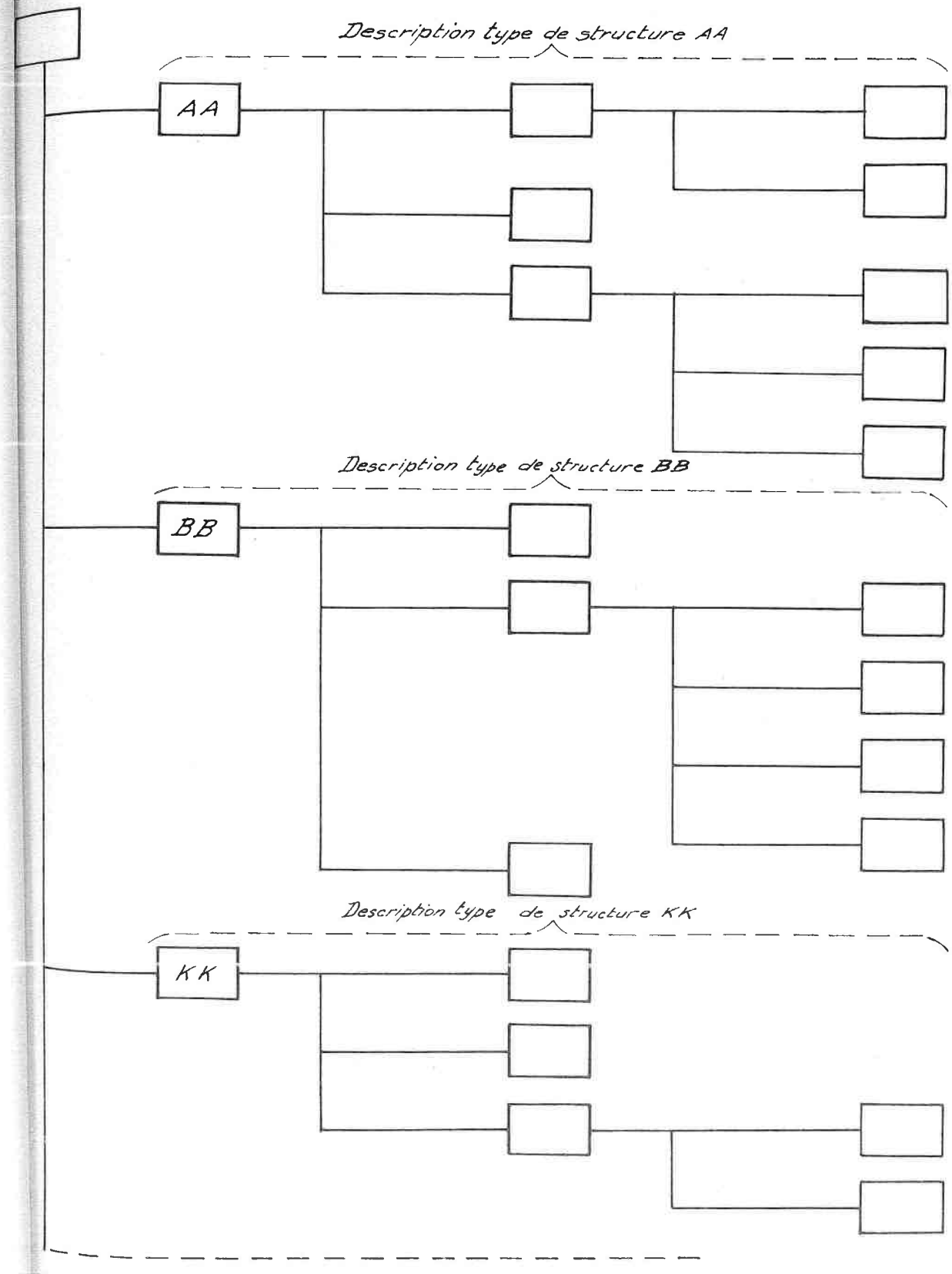
La figure 1 montre quelques exemples de types de structure possibles.

- chaque article a sa structure propre qui doit être compatible avec le type de structure correspondant, selon des règles que nous préciserons ultérieurement.

.../...

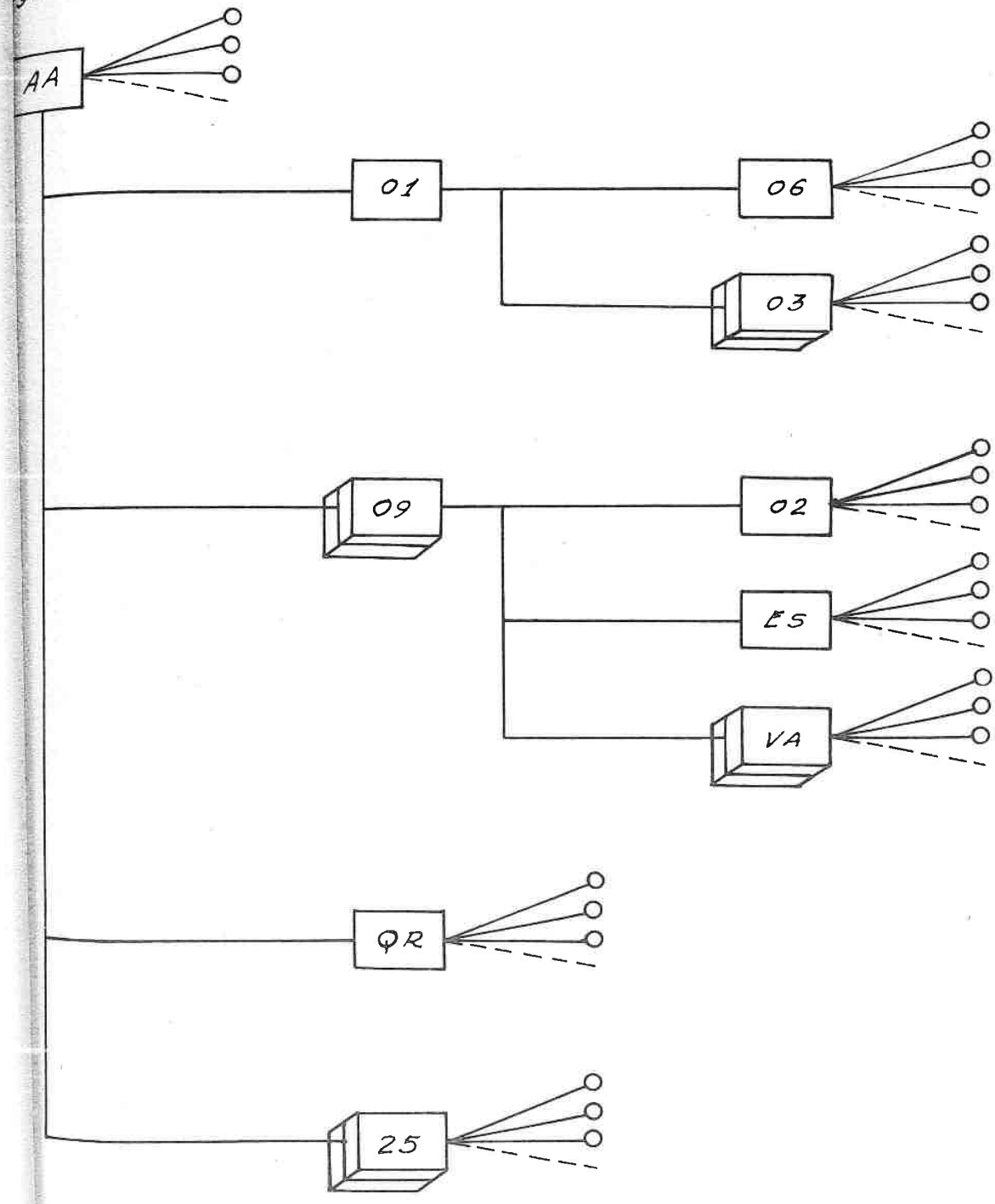
Figure 1 :

lection des Structures



12 - Présentation d'un type de structure :

igure 2.



Chaque type de structure est défini par une arborescence à 3 niveaux. Dans l'arborescence de la figure 2 nous avons présenté toutes les combinaisons possibles.

121 - Nombre de niveaux :

Pour chaque type de structure, il y a 3 niveaux possibles.

- dans la figure 2 l'identificateur de niveau 1 est AA c'est lui qui caractérise le type de structure décrit
- les identificateurs de niveau 2 sont 01 09 QR 25
- les identificateurs de niveau 3 sont 06 03 02 ES VA

122 - Représentation et définition des identificateurs :

Sur la figure 2, les noeuds de l'arborescence sont marqués par des "carrés" ou des "cubes" superposés, les feuilles par des "ronds".

1221 - Les noeuds :

- A un noeud correspond un identificateur de 2 à 4 caractères. Ce nombre de caractères doit être précisé pour chaque type de structure. Il ne doit pas varier pour tous les identificateurs d'une même structure. Ces identificateurs sont alphanumériques : on n'aura donc aucune relation d'ordre entre eux. Par exemple, dans la figure 2,

01 est suivi de 06 et 09 de 02

Deux caractères donnent déjà un grand nombre de possibilités, mais si l'on offre le choix entre 2, 3 et 4 caractères, c'est pour permettre aux utilisateurs de choisir des codes mnémoniques et dans ce cas 2 caractères peuvent être insuffisants.

.../...

Un noeud représente un ensemble de feuilles ex : [06] ou un ensemble de noeuds et de feuilles

ex : [01]

Nous verrons l'utilité de ces noeuds dans le paragraphe 13 : Information "associée" à un identificateur.

Les noeuds sont uniquement des repères.
La description des informations élémentaires ne se trouve jamais sur les noeuds, toujours sur les feuilles.

- Caractéristiques de ces noeuds :
  Ils peuvent être dans un article
  soit présents
  soit absents
  soit répétés.

Nous avons vu au début de ce chapitre que chaque article avait sa structure propre compatible avec celle du type de structure correspondant suivant certaines règles : absence ou répétition de certaines informations.

Dans la description de la structure, nous devons donc trouver tous les noeuds et toutes les feuilles susceptibles d'être rencontrés dans un article.

. Dans un article, au contraire, certaines informations pourront être absentes. Les noeuds qui regroupent ces informations en seront donc absents.

. Lenoeud est dit "unique" s'il est présent une fois au plus dans le noeud de niveau supérieur auquel il appartient. Dans la figure 2 il est représenté par un carré.
Les noeuds "uniques" représentés sont :

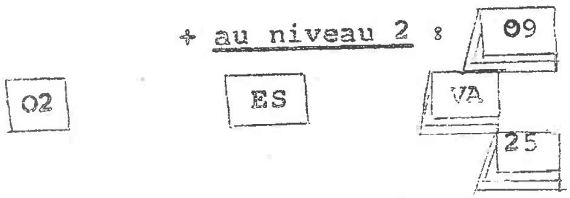
+ au niveau 2 : [01] qui a pour descendants [06] et [03] On ne parlera de descendants qu'au niveau des noeuds, les feuilles ne sont pas alors considérées comme descendants des noeuds.

[QR] qui n'a pas de descendants.

.../...

+ au niveau 3 : Le niveau 3 est le dernier niveau possible, les noeuds présents au niveau 3 n'auront donc pas de descendants possibles.

. Le noeud est dit "répétitif" s'il peut être présent une ou plusieurs fois dans le noeud de niveau supérieur auquel il appartient.
Dans la figure 2 il est représenté par des "cubes superposés".
Les noeuds "répétitifs" représentés sont :

+ au niveau 2 : 

qui a pour descendants

qui n'a pas de descendants.

+ au niveau 3 : Comme on l'a vu précédemment il ne peut pas y avoir de descendants.

Première remarque -
- Un noeud dit "unique" appartenant à un noeud dit "répétitif" pourra être présent plusieurs fois dans l'article.
Exemple d'article tiré de la figure 2 :

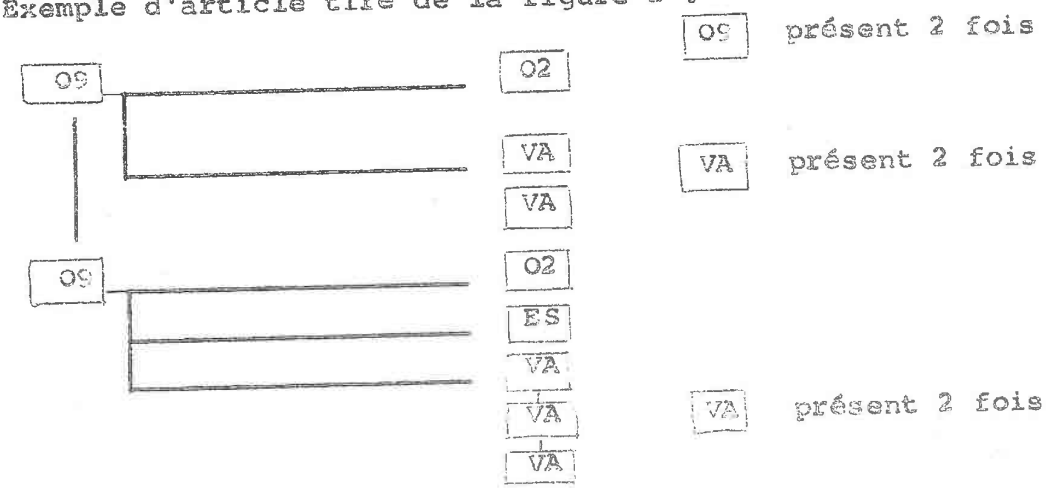


figure 3

.../...

02	est dit "unique". Il appartient à	09	dit "répétitif". Dans l'exemple, il est présent deux fois.

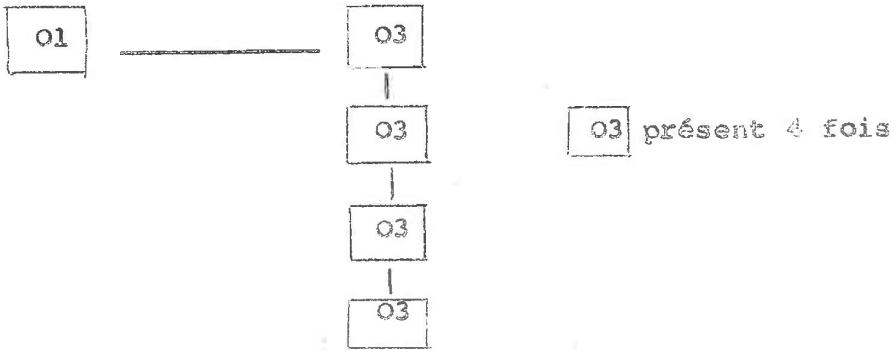
- un noeud dit "unique" appartenant à un noeud dit "unique" ne pourra être présent qu'une seule fois au plus dans l'article. Exemple tiré de la figure 2 :

06	est "unique". Il appartient à	01	"unique". Il est présent une seule fois au plus dans l'article.

- un noeud dit "répétitif" appartenant à un noeud dit "répétitif" pourra être présent plusieurs fois dans l'article. Exemple tiré de la figure 3 :

VA	est "répétitif". Il appartient à	09	"répétitif". Dans l'article il est présent 5 fois.

- un noeud dit "répétitif" appartenant à un noeud dit "unique" pourra être présent plusieurs fois dans l'article. Exemple tiré de la figure 2 :



03	est "répétitif". Il appartient à	01	"unique". Il est présent 4 fois dans l'article.

Deuxième remarque : Le noeud de niveau 1 est particulier. Il comporte obligatoirement des feuillesayant des informations

.../...

particulières dites "informations privilégiées". Ce sont des critères de sélection courants : par exemple, un numéro de classement, des coordonnées géographiques, etc ...
Chaque structure comporte obligatoirement un noeud de niveau 1 et peut ne comporter qu'un seul noeud de niveau 2. Au moins un noeud de niveau 2 suivi d'une ou plusieurs feuilles doit être présent dans l'article sinon on n'aurait aucune information élémentaire autre que "les informations privilégiées", ce qui n'aurait pas de sens.

1222 - <u>les feuilles</u> :

Elles portent la description des informations élémentaires. A chaque feuille correspond une information élémentaire.
L'information élémentaire c'est l'"atome" : c'est la plus petite unité que l'on peut atteindre dans l'article.
Chaque information élémentaire est décrite par un <u>identificateur</u> de quatre caractères alphanumériques et des caractéristiques.
- ces <u>caractéristiques</u> sont la taille et un repère du thésaurus.

. La <u>taille</u> est représentée par une valeur sur trois caractères (on peut donc aller jusqu'à 999) ou par un "V". Le "V" signifie taille variable. Cette taille n'est connue qu'à la lecture de l'article et peut varier d'un article à l'autre.

. Le <u>thésaurus</u>

+ le thésaurus a plusieurs rôles :
soit uniquement un rôle de contrôle
soit outre ce rôle de contrôle, un rôle de codage, décodage.
L'usage du thésaurus utilisé en tant qu'outil de contrôle est défini dans le paragraphe 16.
Le codage sert à remplacer une information élémentaire par un code, ce qui peut-être plus pratique.
Le décodage ne sert qu'à l'exécution ; il signifie que les informations ont été entrées codées.

+ le repère de thésaurus se trouve sur deux caractères et comporte un nombre :grâce à ce nombre, on pourra caractériser l'un des trois cas : contrôle uniquement
contrôle et décodage
contrôle et codage.

.../...

- chaque noeud comporte dans la structure une liste exhaustive d'informations élémentaires. Cet ensemble d'informations élémentaires forme une "unité d'informations".

. Un ensemble d'informations élémentaires (c'est-à-dire de feuilles) identifiées par un noeud dit "unique" représente une unité d'informations dite "unique".

. Un ensemble d'informations élémentaires identifiées par un noeud dit "répétitif" représente une unité d'informations dite "répétitive".
Dans ce cas, l'unité d'informations est décrite une seule fois dans la structure.

13 - Information "associée" à un identificateur :

Nous avons donné des identificateurs aux noeuds et aux feuilles en vue de faciliter l'extraction.
Quelles informations ou quels groupes d'informations peuvent-être extraits ?

131 - Information "associée" à une feuille :

Il suffit d'indiquer l'identificateur de l'information élémentaire que l'on veut obtenir.

1311 - L'information élémentaire dépend d'un noeud de niveau 2 :

Deux cas peuvent se présenter :

1 - Ce noeud de niveau 2 est "unique".
L'information élémentaire fait alors partie d'une unité d'informations "unique".
Dans ce cas l'on obtient l'information correspondant à l'identificateur demandé.

2 - Ce noeud de niveau 2 est "répétitif".
L'information élémentaire fait alors partie d'une unité d'informations "répétitive".
Dans ce cas l'on obtient l'information correspondant à l'identificateur demandé de la première unité ; puis celle de la deuxième unité ; de la troisième jusqu'à épuisement du nombre de répétitions.

.../...

1312 - L'information élémentaire dépend d'un noeud de niveau 3 :

Quatre cas peuvent se présenter :

- Ce noeud de niveau 3 est "unique" et le noeud de niveau 2 auquel il appartient est "unique".
Dans ce cas l'on obtient l'information correspondant à l'identificateur demandé comme dans le cas 1 du paragraphe 1311.

- Ce noeud de niveau 3 est "unique" et le noeud de niveau 2 auquel il appartient est "répétitif".
Donnons un indice à chacun des noeuds répétés.
L'information élémentaire fait partie d'une unité d'informations "unique".
Dans ce cas l'on obtient l'information correspondant à l'identificateur demandé, traitée comme dans le cas précédent, pour le noeud de niveau 3 appartenant au noeud de niveau 2 d'indice 1. Puis on obtient l'information correspondant au même identificateur pour le noeud de niveau 3 appartenant au noeud de niveau 2 d'indice 2 etainsi de suite jusqu'à ce que les indices du noeud de niveau 2 soient épuisés.

- Ce noeud de niveau 3 est "répétitif" et le noeud de niveau 2 auquel il appartient est "unique".
Dans ce cas l'on obtient l'information correspondant à l'identificateur demandé de la première unité, puis celle de la deuxième unité, de la troisième jusqu'à épuisement du nombre de répétitions comme dans le cas 2 du paragraphe 1311.

- Ce noeud de niveau 3 est "répétitif" et le noeud de niveau 2 auquel il appartient est "répétitif".
L'information élémentaire fait partie d'une unité d'informations "répétitive".
Dans ce cas l'on obtient l'information correspondant à l'identificateur demandé de la première unité, puis celle de la deuxième unité, de la troisième jusqu'à épuisement du nombre de répétitions du noeud de niveau 3 dans le noeud de niveau 2 d'indice 1.
Puis on obtient l'information correspondant au même identificateur de la première unité, puis celle de la deuxième unité, de la troisièmejusqu'à épuisement du nombre de répétitions du noeud de niveau 3 dans le noeud de niveau 2 d'indice 2.
On continue ainsi jusqu'à ce que tous les indices du noeud de de niveau 2 soient épuisés.

.../...

132 - Information "associée" à un noeud de niveau 3

Il suffit d'indiquer l'identificateur du noeud.
Quatre cas peuvent se présenter :

1 - Ce noeud de niveau 3 est "unique" et le noeud de niveau 2 auquel il appartient est "unique".
Les feuilles correspondant à ce noeud font alors partie d'une unité d'informations "unique"; sur la figure 2 : 06

Dans ce cas l'on obtient l'unité entière : c'est-à-dire une suite d'informations élémentaires (paragraphe 1222).

2 - Ce noeud de niveau 3 est "unique" et le noeud de niveau 2 auquel il appartient est "répétitif". Sur la figure 2 :

02 ES

Les feuilles correspondant à ce noeud font partie d'une unité d'informations "unique".
Dans ce cas l'on obtient les feuilles correspondant au noeud demandé et traitées comme précédemment pour le noeud de niveau 2 d'indice 1, puis celles correspondant au même noeud pour le noeud de niveau 2 d'indice 2 et ainsi de suite jusqu'à ce que les indices du noeud de niveau 2 soient épuisés.

3 - Ce noeud de niveau 3 est "répétitif" et le noeud de niveau 2 auquel il appartient est "unique". Sur la figure 2 : 03

Les feuilles correspondant à ce noeud font partie d'une unité d'informations "répétitive".
Dans ce cas l'on obtient la première unité, la deuxième unité, la troisième et ainsi de suite jusqu'à épuisement du nombre de répétitions.
Chaque unité est une suite d'informations élémentaires(cf paragraphe 1222)

4 - Ce noeud de niveau 3 est "répétitif" et le noeud de niveau 2 auquel il appartient est "répétitif". Sur la figure 2 : VA

Les feuilles correspondant à ce noeud font partie d'une unité d'informations "répétitive".
Dans ce cas l'on obtient la première unité, la deuxième unité, la troisième, traitées comme précédemment jusqu'à épuisement du nombre de répétitions du noeud de niveau 3 appartenant au noeud de niveau 2 d'indice 1. Puis on obtient la première unité, la deuxième unité, la troisième jusqu'à épuisement du nombre de répétitions du noeud de niveau 3 appartenant au noeud de niveau 2 d'indice 2.

.../...

On continue ainsi jusqu'à ce que les indices du noeud de niveau 2 soient épuisés.

133 - Information "associée" à un noeud de niveau 2 :

Il suffit d'indiquer l'identificateur de ce noeud.

1331 - Ce noeud de niveau 2 n'est pas suivi de noeuds de niveau 3 :

On se retrouve alors dans les cas 1 et 3 d'une information "associée" à un noeud de niveau 3. Sur la figure 2 : QR 25

1332 - Ce noeud de niveau 2 est suivi de plusieurs noeuds de niveau 3 :

Deux cas peuvent se présenter :

- Ce noeud de niveau 2 est "unique".

Sur la figure 2 : 01 suivi de 06 et 03

Dans ce cas l'on obtient les informations correspondant au premier noeud de niveau 3, puis celles correspondant au deuxième noeud de niveau 3, au troisième etainsi de suite jusqu'à épuisement de s noeuds de niveau 3 appartenant au noeud de niveau 2 demandé.
Les informations correspondant aux différents noeuds de niveau 3 sont traitées comme aux cas 1 et 3 du paragraphe 132 : Information "associée" à un noeud de niveau 3.

- Ce noeud de niveau 2 est "répétitif".

Sur la figure 2 : 09 suivi de 02 ES VA

Prenons un exemple d'article tiré de la structure décrite sur la figure 2 :

09	indice 1 suivi de	02	et VA	présent 2 fois
09	indice 2 suivi de	02	ES	
09	indice 3 ;..... jusqu'à épuisement du nombre de répétitions.			

.../...

Dans ce cas l'on obtient les informations correspondant à [09] (1)
Elles sont traitées comme s'il s'agissait d'un noeud de niveau 2 "unique". Puis on obtient les informations liées à [09] (2)

Elles sont présentées comme pour [09] (1)

On traite les noeuds [09] (3) [09] (4) .....

d'une façon identique jusqu'à ce que tous les indices de ce noeud de niveau 2 soient épuisés.

134 - Information "associée" à un noeud de niveau 1 :

Il suffit d'indiquer l'identificateur de ce noeud.
On obtient d'abord les informations"privilégiées" puis tous les noeuds de niveau 2 traités comme au paragraphe 133 : Information "associée" à un noeud de niveau 2.
Dans ce cas l'on extrait tout l'article.

14 - Description de l'article :

D'après la structure décrite on va retrouver dans l'article une succession de noeuds et de feuilles.

141 - Les noeuds :

Enonçons les règles de compatibilité entre la structure et l'article.

- Si un noeud a été déclaré "unique" ou "répétitif" dans la structure, il devra être "unique" ou"répétitif" dans l'article.

- L'ordre de classement des noeuds présents dans l'article doit être l'ordre décrit dans la structure.
Les noeuds dans l'article pourront donc être "uniques" "répétitifs" ou encore absents.

- Si les informations relatives à un noeud sont absentes, l'identificateur de cenoeud ne sera pas présent dans l'article.
Chaque fois que l'on rencontre un identificateur de noeud dans l'article on le compare avec l'identificateur de noeud, suivant celui qui vient d'être traité dans la structure. Si les deux identificateurs ne sont pas identiques, on passe au suivant dans la structure et ainsi de suite jusqu'à ce qu'il s'en trouve deux identiques.

.../...

La comparaison se fait toujours en progressant dans la structure, on ne peut revenir en arrière (deuxième règle de compatibilité). Donc si les identificateurs de noeuds sont mal placés, une erreur sera détectée.

- Lorsque les deux identificateurs de noeuds sont égaux c'est que le noeud est présent.
On vérifiera si les descendants de ce noeud, s'il en a, sont les mêmes dans la structure et dans l'article. Il est toutefois possible qu'un descendant soit absent.

- Si un noeud est "répétitif", on aura dans l'article une information supplèmentaire que l'on qualifiera d'"information de service". Elle indique le nombre de fois où ce noeud est présent, en effet, la répétition d'un même noeud varie d'un article à l'autre.

Remarque : On peut également considérer les identificateurs de noeuds comme "informations de service" puisque c'est eux qui permettent de faire le correspondance structure-article.

142 - Les feuilles :

- Les informations élémentaires correspondant aux noeuds présents dans l'article doivent se succéder dans l'ordre décrit par la structure.

- Une feuille ne peut-être absente que si le noeud auquel elle appartient est absent.
Si dans un noeud une des informations n'a pas de signification, on ne la supprime pas. On remplit la zone réservée à cette information de blancs. Il n'y a pas compactage. Cette éventualité doit être rare si la structure a été bien organisée.

- Précisons brièvement comment sont manipulées les informations élémentaires.
Appelons article la suite de cartes perforées qui vont nous permettre de former l'enregistrement destiné à être rangé sur bande et à être exploité ultérieurement.
En mémoire centrale, on a une zone MEM dans laquelle on construit l'enregistrement à l'aide de la structure et des différents éléments portés par les cartes qui composent l'article.

. Dans la structure on prend la taille des informations élémentaires.
Trois cas peuvent se présenter :

+ soit la taille est connue et on peut ranger l'information élémentaire dans la zone MEM

+ soit dans la structure on a mentionné un repère de thésaurus. Si ce repère indique un codage, on rangera l'information une fois codée dans la zone MEM

.../...

+ soit la taille est "V". On calcule alors la place occupée par l'information élémentaire et on la met devant cette information avant de la ranger dans la zone MEM.

. Dans l'article, outre les informations élémentaires on trouve plusieurs renseignements :

+ soit une ou plusieurs "informations de service" qui donnent le nombre de répétitions quand un noeud est "répétitif",

+ soit des éléments qui ne servent qu'au contrôle. On ne les retrouve plus dans l'enregistrement : par exemple : le type de carte, le numéro de classement, le caractère $ etc....

Conclusion : La présentation qui vient d'être faite est assez générale. On a voulu montrer les possibilités offertes à l'utilisateur.
Nous allons maintenant à partir d'un exemple simple examiner la manipulation des informations.
En effet, pendant le rangement des informations que l'on vient de voir sont construites en mémoire des tables (on a gardé le même principe que pendant notre première expérience) permettant de retrouver n'importe quelle information ou n'importe quel groupe d'informations et de les traiter. On entrera progressivement dans le détail de ces tables en prenant un exemple complet (chapitre 2) puis en faisant une analyse organique complète (chapitre 3).

15 - Manipulation des tables pour obtenir une information "associée" à un identificateur.

151 - Exemple :

Prenons un exemple simple comportant trois cas typiques :
- noeud de niveau 2 "unique" sans noeuds de niveau 3
- noeud de niveau 2 "unique" avec des noeuds de niveau 3 "uniques" et "répétitifs"
- noeud de niveau 2 "répétitif" sans noeuds de niveau 3.

L'exemple est représenté sur la figure 4, à côté de chaque feuille sont décrits l'identificateur et la taille.

152 - Les tables :

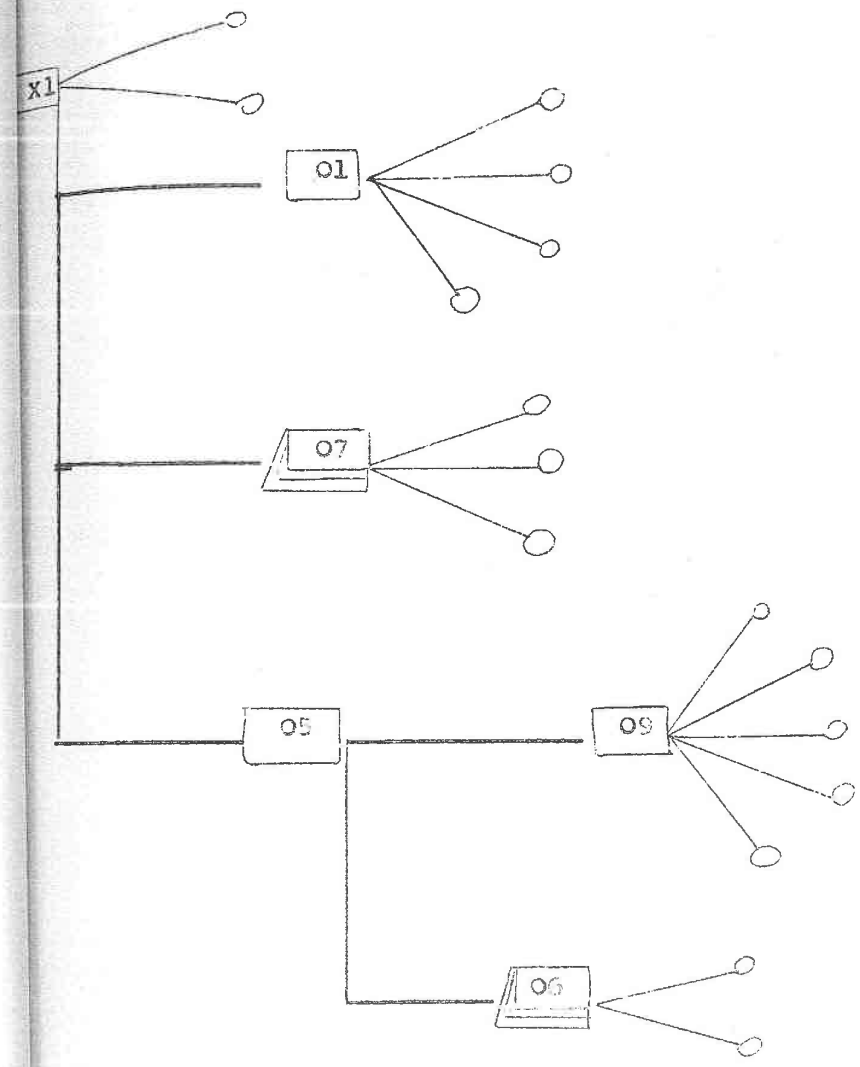
Pour construire un enregistrement on abesoin de deux programmes dont nous reparlerons. Le premier établit une table structure, le deuxième une table des noeuds.

.../...

Le deuxième programme a besoin de la table structure pour former la table des noeuds.
Nous ne nous intéresserons pas à la construction des tables maintenant mais seulement à leur utilisation pour obtenir une information "associée" à un identificateur.
Nous omettrons donc dans la première table certaines informations inutiles à l'extraction. Nous allons simplement donner une description de ces tables. Ces tables seront reprises dans les chapitres suivants (dans certains cas où cela facilite les explications, elles ne seront pas présentées d'une façon identique à celle que nous allons montrer).

.../...

Figure 4 :



identificateur	taille
NCLA	6
PRIX	3
ARM1	1
SØRA	5
PRIA	7
HELE	3
RØND	9
PRIS	2
CRAS	7
PLAI	2
JGEØ	3
PCRZ	10
PRAC	7
CØLI	4
PRØF	5
VALE	3

.../...

1521 - Table structure :

identificateur de noeud	identificateur de feuille	fixe ou repetitif	taille en caractères	position /origine de l'unité d'informations	longueur unité d'informations si répétitive	thésaurus
X1	NCLA	F	6	0		
	PRIX	F	3	6		
01	FIXE					
	ARM1	F	1	0		
	SØRA	F	5	1		
	PRIA	F	7	6		
	HELE	F	3	13		
07	FIXE					
	RØND	R	9	0	18	
	PRIS	R	2	9	18	
	CRAS	R	7	11	18	
05	FIXE					
09	ELAI	F	2	0		
	IGEØ	F	3	2		
	PCRZ	F	10	5		
	PRAC	F	7	15		
	CØLI	F	4	22		
06	PRØF	R	2	0	8	
	VALE	R	3	5	8	

Ne nous intéressons pas dans le cas étudié

.../...

Précisons certaines anomalies qui seront expliquées dans les chapitres suivants.

- A côté d'un noeud de niveau 2 dans l'emplacement "identificateur de feuille" on trouve :
FIXE si le noeud est "unique"
REPE si le noeud est "répétitif".

- "F", ou "R", signifie que l'unité d'informations à laquelle appartient la feuille est "unique" ou "répétitive". On répéte "F" ou "R" derrière chaque identificateur de feuille.
On répéte également derrière chaque identificateur de feuille la longueur de l'unité d'informations si elle est répétitive.

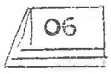
1522 - Table desnoeuds :

taille de l'enregistrement	Taille de la table
	80

identificateur de noeud	position/origine de l'enregistrement	multiplicité
X 1	80	
01	89	
07	105	3
05	159	
09	159	
06	185	2

.../...

- La multiplicité ne nous intéresse que dans le cas où un noeud est "répétitif". En effet elle indique le nombre de fois où ce noeud "répétitif" est présent dans l'article.
Nous savons qu'un noeud est "répétitif" si dans la table structure nous trouvons "R" à côté de l'identificateur de feuille pour un noeud de niveau 3 ou "REPE" à la place de l'identificateur de feuille pour un noeud de niveau 2. Dans la figure 4 :

et


sont "répétitifs".

Nous connaissons le nombre de fois où ce noeud "répétitif" est présent dans le noeud de niveau supérieur grâce à l'"information de service" (cf paragraphe 141).
Comme nous l'avions dit dans le paragraphe 152, nous avons donc besoin de la table structure pour construire la table des noeuds.

- La position par rapport à l'origine de l'enregistrement est calculée en faisant la somme de toutes les tailles des informations élémentaires et en ajoutant à la fin la taille de la table (cf paragraphe 3333 remarque).

- Les deux premiers mots de la table : taille de l'enregistrement et taille de la table ne nous interessent pas dans le cas étudié.

- Seuls sont présents dans la table les noeuds effectivement présents dans l'article. Si par exemple les noeuds 07 et 09 étaient absents de l'article, ils seraient absents de la table (cf paragraphe 33333 - pour le niveau 2).

153 - Manipulation des informations :

Prenons le cas d'une information associée à une feuille.

1531 - Premier exemple :

Il représente le cas 1 du paragraphe 131.
On indique l'identificateur de l'information à extraire : PRIA.
On a besoin de consulter les deux tables en même temps : il faut donc qu'elles soient présentes toutes les deux en mémoire centrale.

- On cherche PRIA dans la table structure, on en déduit le noeud auquel elle appartient 01

Dans la table des noeuds 01 commence à l'adresse 89.

.../...

- PRIA commence à l'adresse : position par rapport à l'origine de l'enregistrement du noeud [01] + position par rapport à l'origine de ce noeud (table structure).
C'est-à-dire à l'adresse 89 + 6 = 95

- Sa taille est de 7 caractères (table structure). On sort donc 7 caractères à partir du 95ème caractère et l'on obtient l'information.

- PRIA appartient à une unité d'informations "unique" : "F" (table structure). L'information obtenue est donc la seule d'identificateur PRIA.

1532 - Deuxième exemple :

Il représente le cas 2 du paragraphe 131.
On indique l'identificateur de l'information à extraire : PRIS.
- Dans la table structure pRIS appartient au noeud [07]
- Dans la table des noeuds [07] commence à l'adresse 105
- PRIS commence à l'adresse 105 + 9 = 114.

1 - Sa taille est de 2 caractères.
On sort donc 2 caractères à partir du 114ème caractère et l'on obtient l'information.

- PRIS appartient à une unité d'informations "répétitive" :"R" (table structure). Il y a donc d'autres informations d'identificateur PRIS.

- On connait le nombre de fois où cette unité est "répétitive" grâce à "multiplicité" de la table des noeuds. Il y en a 3.

2 - On ajoute donc à l'adresse d'origine de PRIS "longueur d'unité d'informations si répétitive" (table structure).
C'est-à-dire 114 + 18 = 132
On reprend 1 et l'on obtient la deuxième information.

- Il reste une information puisque la multiplicité est de 3.
- On reprend 2 en utilisant la dernière adresse d'origine de PRIS, puis 1 et l'on obtient la troisième information.

154 - Conclusion :

Nous avons étudié deux exemples pour montrer l'utilité des tables et comment elles étaient utilisées.
Ce sont les exemples les plus simples : nous avons vu dans le

.../...

paragraphe 13 : Information "associée" à un identificateur tous les cas d'extraction possibles. De plus nous n'avons pas parlé dans ce paragraphe des informations de taille variable. Le procédé que nous venons de montrer sera celui qui sera suivi lors de l'exploitation du système. Il sera plus ou moins compliqué suivant les cas étudiés au paragraphe 13.
En effet avant de procéder à n'importe quelle sélection, il faut d'abord extraire les informations élémentaires devant satisfaire à la condition de sélection.

16 - Contrôles :

Pendant l'exécution du programme,des contrôles ont été prévus. Certains contrôles sont généraux, d'autres spécifiques.

161 - Contrôles généraux :

1611 - Erreurs grossières :

Ces contrôles se trouvent dans le programme général et permettent de détecter des erreurs telles que : intervertion de cartes dans un article, ou d'un article à l'autre ; répétition de cartes.
Sur les cartes composant l'article, il y a un certain nombre d'éléments qui ne servent qu'au contrôle et qui disparaissent à la mise sur bande.
Citons quelques exemples :

- Tous les noeuds se trouvent sur des types de cartes définis (on verra ceci dans le chapitre suivant). Ces noeuds doivent se trouver dans l'ordre décrit par la table structure. S'il y a absence, on passe au noeud suivant et cette absence pourra être contrôlée dans les contrôles spécifiques.

- Un numéro de classement permet de retrouver toutes les cartes appartenant à un même article.

- Si dans la succession des informations élémentaires des cartes "suite" sont nécessaires, elles sont numérotées.

- L'"information de service" précisant le nombre de fois où un noeud est présent sert aussi de vérification. A chaque répétition on procède à un contrôle.

.../...

1612 - Thésaurus :

- On a vu que le thésaurus pouvait servir uniquement de contrôle. Il comportealors toute une liste de mots et on vérifie si l'information élémentaire est un mot de cette liste.
Dans le cas d'un décalage sur la carte perforée, c'est un contrôle supplémentaire ; en effet l'information à contrôler sera automatiquement fausse.

- On a vu que le thésaurus codage-décodage ou décodage seulement pouvait aussi servir de contrôle. Dans ce cas on a une table avec une liste des mots codés et une liste des codes. Si on ne trouve pas l'identificateur de l'information à coder ou à décoder dans la liste des mots codés, une erreur est détectée.

162 - Contrôles spécifiques :

C'est un programme que l'on adjoint au programme général. Il sert à faire des tests de pertinence. Est-ce que telle information est supérieure à telle valeur ou comprise entre telle et telle valeur ? Tel noeud est-il présent ?
Ce programme sera donc différent pour chaque structure et devra être écrit par l'utilisateur.

Conclusion : On a donc essayé de prévoir un grand nombre de contrôles pour éviter des erreurs irréparables.
Il s'est avéré après l'expérience sur le programme particulier réalisé pour le Laboratoire de l'Equipement que ces contrôles étaient indispensables.
Très peu d'articles ne comportaient pas d'erreurs mais après ces contrôles et un contrôle manuel, il semblait n'en rester pratiquement plus.

.../...

2 - Illustration à partir d'un exemple tiré de la géologie : le sondage carotté :

Il s'agit d'un exemple pris pour tester le programme général. A l'aide de cet exemple, on va pénétrer d'une façon plus détaillée dans le fonctionnement de ce programme.

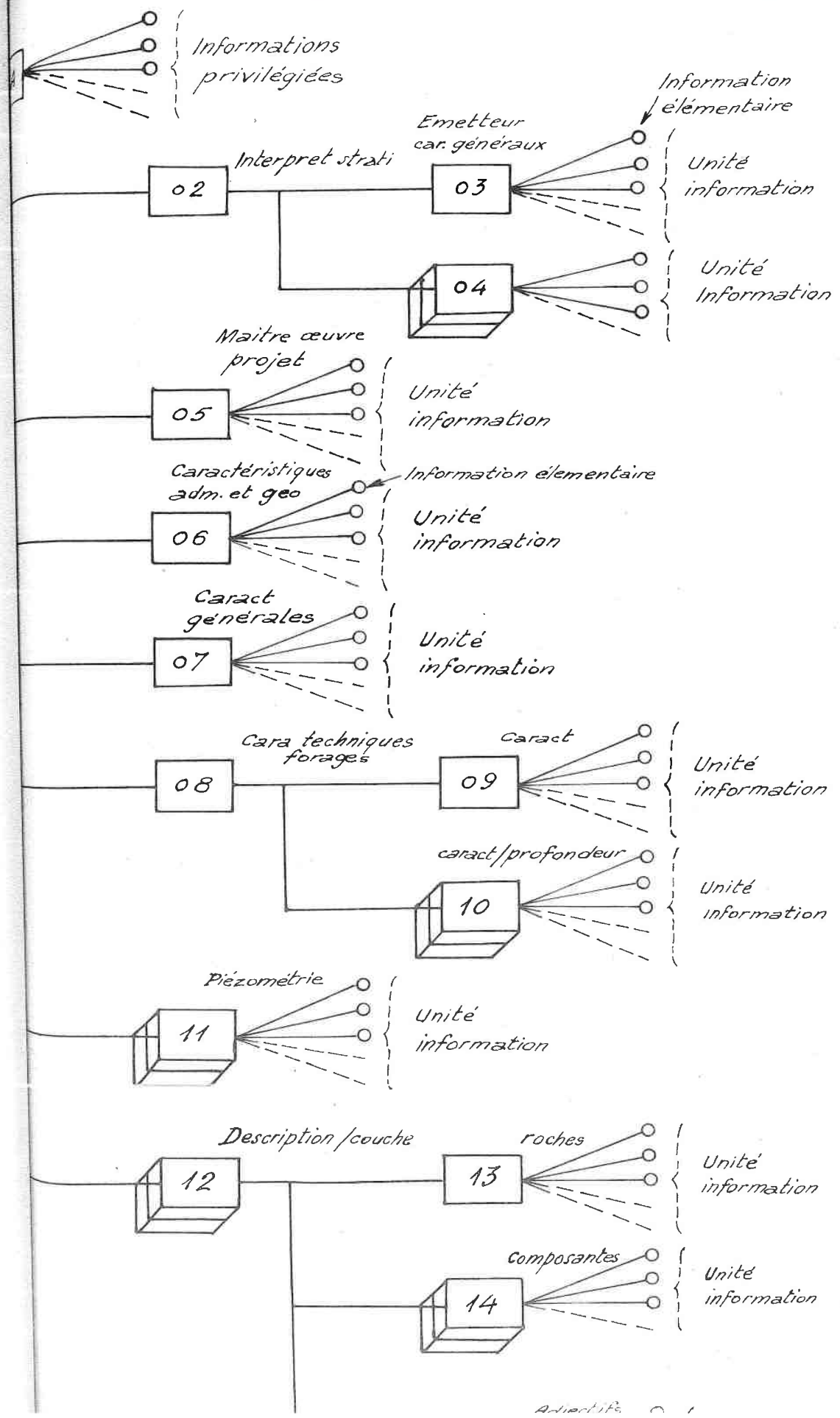
21 - Sa structure :

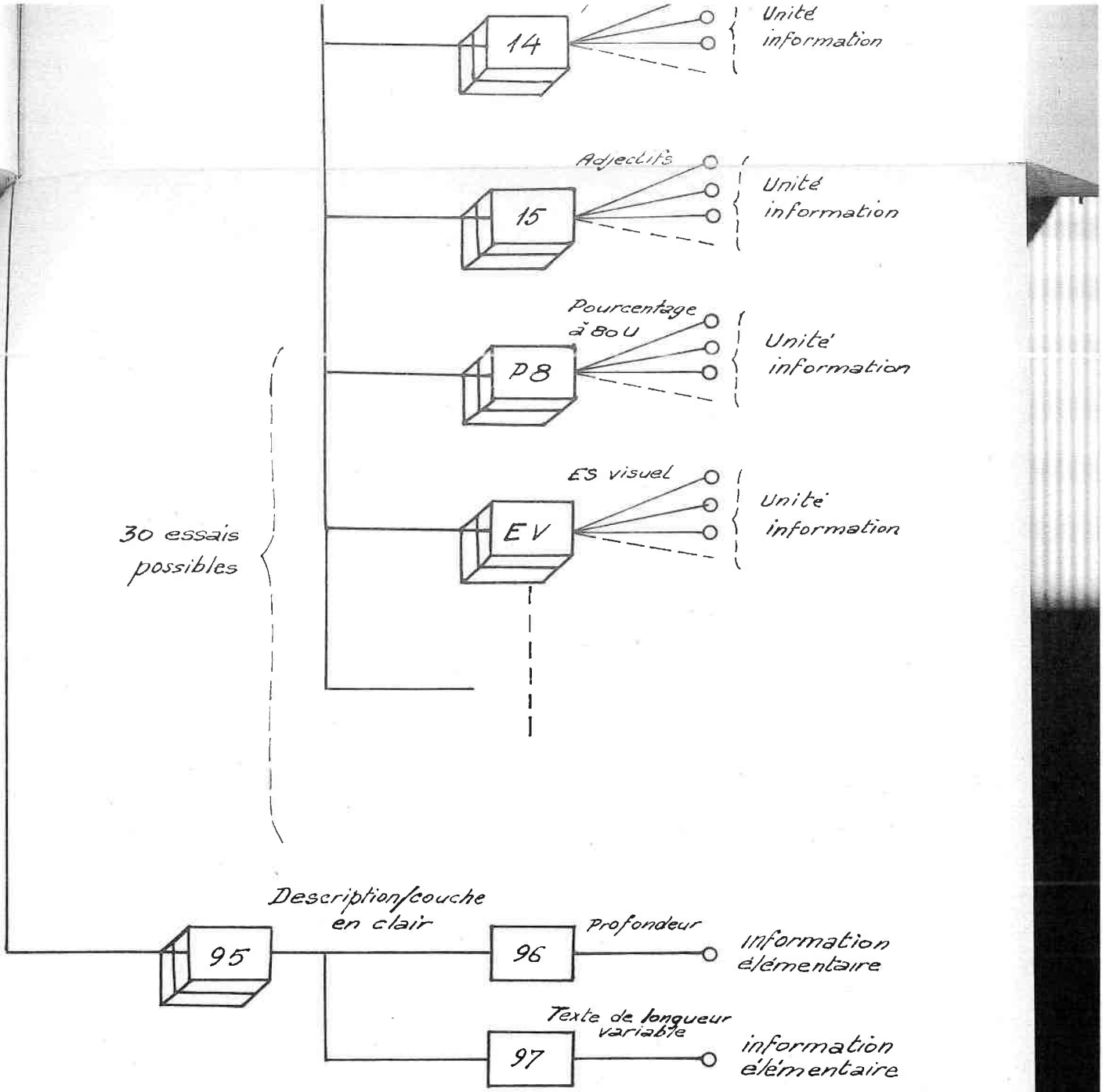
Pour la mise au point de notre programme nous avons utilisé plusieurs types de structure mais toujours des structures tirées de la géologie.
Entre la structure du sondage carotté que nous allons étudier et celle de la sismique réfraction, certains noeuds sont communs, nous avons donc choisi les mêmes identificateurs de noeuds et de feuilles.
Lorsqu'un article est en cours de traitement seule la stucture à laquelle il appartient est présente en mémoire (cf paragraphe 11). Il est donc obligatoire de décrire dans chaque structure tous les noeuds possibles même si certains d'entre eux sont communs avec d'autres structures.
Dans la structure de la figure 5, les trois niveaux sont représentés.

211 - Description des noeuds :

- Le noeud de niveau 1 est "unique". Nous avons vu qu'il portait des "informations privilégiées".
L'identificateur de ce noeud a un rôle supplémentaire : il représente l'étiquette ou label du fichier structure ainsi constitué . En effet chaque type de structure donne naissance à un fichier structure

. . . / . . .


Informations privilégiées
Information élémentaire
Emetteur car. généraux
Interpret strati
02
03
Unité information
04
Unité Information
Maitre œuvre projet
05
Unité information
Caractéristiques adm. et geo
Information élémentaire
06
Unité information
Caract générales
07
Unité information
Cara techniques forages
Caract
08
09
Unité information
caract/profondeur
10
Unité information
Piézométrie
11
Unité information
Description /couche
roches
12
13
Unité information
Composantes
14
Unité information



Il y a 30 essais possibles : nous ne pouvons tous les représenter sur ce schéma mais voici les identificateurs de nœuds : P8, EV, ES, DW, DD, CA, SC, LU, LG, LØ, DS, DH, FD, PØ, PI, PP, PL, PU, PE, WN, RC, VS, 2D, 2E, LG, LA, RG, VG, GR, SE.

Nous avons formé un fichier par structure au lieu de regrouper toutes les structures dans un même fichier pour la raison suivante. Le positionnement devant un fichier sur une bande se fait automatiquement, tandis que rechercher un enregistrement particulier dans un fichier provoque le transfert par l'utilisateur de cet enregistrement en mémoire centrale et un traitement particulier pour reconnaître l'enregistrement lu.
Former un fichier pour chaque structure est donc la solution la plus rapide et la plus économique. L'identificateur de niveau 1 sur la figure 5 est [AA]

- les noeuds de niveau 2 peuvent être présents, absents ou répétés dans l'enregistrement (cf chapitre 1).

. Si un noeud de niveau 2 est suivi de noeuds de niveau 3 et s'il est "répétitif", l'"information de service" (cf chapitre 1) est un nombre précédant l'identificateur de noeud et indiquant le nombre de répétitions.

. Si un noeud de niveau 2 n'est pas suivi de noeuds de niveau 3 (les feuilles se trouvent alors au niveau 2) et s'il est "répétitif", l'"information de service" est un nombre suivant l'identificateur de noeud et indiquant le nombre de répétitions (ceci par analogie au niveau 3 pour que le traitement des feuilles soit identique).
Exemples tirés de la figure 5 :

. Noeuds de niveau 2 contenant des noeuds de niveau 3 :

. Noeuds de niveau 2 ne contenant pas de noeuds de niveau 3 :

. Noeuds de niveau 2 "répétitifs" :

. Noeuds de niveau 2 "uniques" :

Précisons de nouveau qu'aucune relation d'ordre ne peut être établie entre les identificateurs de noeuds, ceux-ci étant alphanumériques. Dans l'exemple choisi la plupart des identificateurs sont numériques et placés dans un ordre croissant, ce qui pourrait provoquer une confusion mais c'est pur hasard.

.../...

- les noeuds de niveau 3 peuvent aussi être présents, absents ou répétés dans l'enregistrement lui-même.
Dans le cas de noeuds "répétitifs", l'"information de service" est un nombre suivant l'identificateur de noeud et indiquant le nombre de répétitions.
Les noeuds de niveau 3 quand ils existent contiennent les informations élémentaires, c'est-à-dire les feuilles.
Nous avons vu que s'il n'y avait pas de noeuds de niveau 3, les feuilles se trouvaient au niveau 2. Qu'elles soient derrière un noeud de niveau 2 ou de niveau 3, le traitement de ces feuilles est identique.
Dans le chapitre 1 nous avons parlé d'unité d'informations "unique" et d'unité d'informations "répétitive".
Dans la figure 5 nous retrouvons :

. des unités d'informations "uniques".
Elles sont représentées par un ensemble de feuilles dont les identificateurs de noeuds sont :

+ au niveau 2 : 05 06 07

+ au niveau 3 : 03 09 13 96 97

. Des unités d'informations "répétitives".
Elles sont représentées par un ensemble de feuilles dont les identificateurs de noeuds sont :

+ au niveau 2 : 11

+ au niveau 3 : 04 10 14 15 P8

EV... (tous les essais).

212 - Description des feuilles :

Nous avons vu que les feuilles comprenaient les informations élémentaires repérées par :

- un identificateur
- une taille
- un repère de thésaurus.

Voici la représentation des informations élémentaires dans la table structure. Nous allons donner une description des feuilles en les faisant précéder de l'identificateur correspondant et en les faisant suivre de la taille et du repère de thésaurus.

.../...

identificateur de feuille	Description "en clair" de l'information	taille	repère de thésaurus.
CLAS	numéro de classement	9	
ZLAM	Z Lambert	1	
XLAM	X Lambert	7	
YLAM	Y Lambert	6	
PRXY	précision sur X et Y	1	
ELAI	essai labo ou in situ	1	
IGRA	test granulat	2	
IGEØ	test géotechnique	1	
TMEC	test mécanique des sols	1	
THYD	test hydrogéologie	1	
TDIA	test diagraphie	1	
TGFI	test géophysique	1	
ØRGA	organisme	6	
NØMI	nom	10	
IFØB	information fictive observée	4	
ANNI	année	4	
MØII	mois	2	
PLUØ	plus ou moins	1	
GØTØ	côte origine NGF	6	
PREZ	précision sur Z	1	
PRCØ	précision des coupures	1	
PRAP	précision sur azimut et pendage	1	
FIAB	fiabilité	1	
FØR1	formation 1	2	3
FØR2	formation 2	2	3
AZIM	azimut	3	
PENT	pente	2	
PRFI	profondeur fin de formation	6	
MØEU	maître d'oeuvre	31	
PRØJ	désignation du projet	36	
ESØW	entreprise de travaux	6	
CENT	centralisation émettrice	6	
LESS	laboratoire essai sols	6	
SURV	origine de la coupe	6	
CCØØ	degré de confiance de la coupe	1	
NDØS	numéro de dossier	10	
NSDØ	numéro de sondage dans le dossier	6	
ANNS	année	4	
MØIS	mois	2	
DEPT	département	2	
CØMM	commune	3	
PLMØ	plus ou moins	1	
CNGF	cote NGF	6	
PRCZ	précision sur Z	1	

.../...

PFAT	profondeur atteinte	6
BUTS	but	2
PRPF	précision profondeur	1
FSUB	c. substratum	1
PSUB	profondeur substratum	5
ARTE	plus ou moins	1
PEAU	profondeur eau	6
DICA	diamètre minimum de la carotte	4
AZI1	azimut.	3
INCL	inclinaison	3
PAZI	précision sur azimut	1
PINC	précision sur inclinaison	1
NBTE	nombre de télescopages	1
PTUB	profondeur fond du tubage	5
DTUB	diamètre intérieur du tubage	4
PFCH	profondeur changement	5
TMAC	type de machine	8
MFØR	méthode de forage	3
DSØN	diamètre du sondage	4
ØUTI	outil	7
TEØU	tête d'outil	2
LUBR	lubrifiant	1
TPIE	type de piezo	2
HPSØ	hauteur du piezo sous-sol	3
LPIE	longueur totale du piezo	5
HCRE	hauteur crépinée	5
TCRE	type de crépine	1
FANN	file annulaire	1
PCCR	pourcentage de crépine	2
TNAP	type de nappe	1
1ANN	année)	4
1MØI	mois (date début de l'observation.	2
1JØU	jour)	2
2ANN	année (	4
2MOI	mois) date fin de l'observation.	2
2JOU	jour (	2
PBØU	profondeur base du bouchon	5
TBØU	type de bouchon	1
PMIN	profondeur minimum eau/sol	5
AMPL	amplitude	5
PMUR	profondeur du mur	6
PTCØ	perte totale chute d'outil	1
CPRE	caractéristique du prèlèvement	3
NBRØ	nombre de roches ≤ 5	1
TRØC	type de roche	3

.../...

CRE1	crédibilité (tout ceci est rap-	1	
RØC1	roche 1) pelé 5 fois pour	7	4
NBC1	nombre de composants - (décrire les	1	
NBA1	nombre d'adjectifs) 5 roches	1	
MUL1	multiplicité (possibles	1	
ST11	structure 1) les noms	1	
ST12	structure 2 (sont à cha-	1	
DIM1	dimension) que fois	1	
CE11	coordinateur d'ensemble 1 (modifiés	1	
CE12	coordinateur d'ensemble 2)	1	
C121	coordinateur relatif à roche 2 (	1	
C122	coordinateur relatif à roche 2)	1	
C131	coordinateur relatif à roche 3 (	1	
C132	coordinateur relatif à roche 3)	1	
C141	coordinateur relatif à roche 4 (	1	
C142	coordinateur relatif à roche 4)	1	
C151	coordinateur relatif à roche 5 (	1	
C152	coordinateur relatif à roche 5)	1	
SRA1	se rapporte à	1	
CØMP	composant 1	7	1
TYA1	type d'adjectif	2	
ADJ1	adjectif	7	2
SUPE	superlatif	1	
MULT	multiplicité	1	
STRU	structure	2	
DIME	dimension	1	
CECØ	coordinateur d'ensemble	2	
SRA2	se rapporte à	2	
TYA2	type d'adjectif	2	
ADJ2	adjectif	7	
STR2	structure	1	
PRP8	profondeur	6	
VAP8	valeur	3	
PREV	profondeur	6	
VAEV	valeur	4	
	On en a 22 de ces types. On met pour chacun d'eux PA .. (son nom), VA .. (son nom).		
PR2D	profondeur	6	
V12D	valeur 1	3	
V22D	valeur 2	3	
	on en a 6 de ce type.		

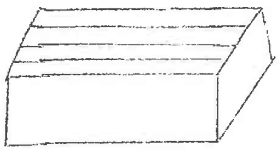
.../...

PRLA	profondeur	6	
V1LA	valeur 1	4	
V2LA	valeur 2	4	
	On en a 6 de ce type.		
PRGR	profondeur	5	
VAGR	valeur	56	
PRSE	profondeur	6	
V1SE	valeur 1	2	
V2SE	valeur 2	40	
PRØT	profondeur	5	
TEXT	texte en "clair" du contenu des couches de profondeur PRØT.	V	

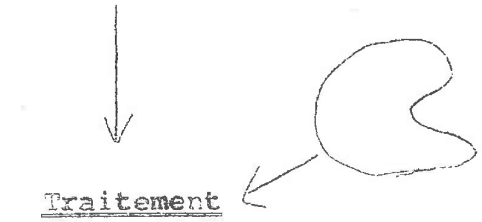
.../...

213 - Représentation de la structure en mémoire :

Nous avons vu toutes les données nécessaires pour construire une structure. Nous allons voir comment elles sont représentées en mémoire.



- cartes décrivant la structure servant de paramètres au programme général.
- cartes de contrôle permettant de faire des tests d'erreur.



← programme général sur bande

Traitement

fichier structure

Ce programme engendre un fichier structure. On aura autant de fichiers structures que de types de structure (cf paragraphe 2111)
En géologie on a 6 types de structure :
- AA sondage carotté
- BB sondage tarière
- CC affleurement
- DD ballastière
- EE carrière
- FF sismique réfraction

Chaque structure constitue un fichier. Par analogie avec la table des noeuds (cf paragraphe 222) au lieu de parler de fichier structure, nous parlerons de table structure.
Dans la table structure nous aurons les informations suivantes :
- les types de carte sur lesquels se retrouvent les identificateurs de noeuds et les feuilles s'il y a lieu.
Ces types de carte ont été introduits uniquement dans un but de contrôle. Sur un même type de carte, peut-être présent soit un identificateur de noeud de niveau 2 suivi ou non d'un identificateur de noeud de niveau 3 et de ses feuilles, soit d'un identificateur de noeud de niveau 3 suivi de ses feuilles.

.../...

Si dans l'article des cartes "suite" sont nécessaires l'identificateur de noeud n'est pas répété ; c'est le type de carte et le numéro de séquence qui servent de vérification.

- Les identificateurs de noeuds
- Les identificateurs de feuilles ou "FIXE" ou "REPE" suivant un identificateur de noeud de niveau 2 et signifiant que ce noeud est "unique" ou "répétitif". Ceci peut ne pas paraître homogène, mais a été conçu de cette façon pour faciliter l'exploitation.

- Une lettre "F" ou "R" précisant si l'unité d'informations à laquelle appartient un ensemble de feuilles est "unique" ou "répétitive".

- La taille des feuilles : cette taille ou bien est fixe et l'on connait alors sa valeur, ou bien est variable et l'on aura le caractère "V".
Si la taille de l'information élémentaire est fixe, grâce à cette taille on peut compléter le tableau représentant la structure en calculant la position des feuilles par rapport à l'origine de l'unité d'informations et la longueur de l'unité d'informations quand elle est "répétitive".
Ceci n'est évidemment pas possible quand une information élémentaire est de taille variable. Pour la position par rapport à l'origine de l'unité d'informations des feuilles qui la suivent, on aura "V". Pour la longueur de l'unité d'informations si elle est "répétitive" on aura à côté de chaque information "V". On ne connaitra la taille exacte de l'information qu'à l'exploitation.

- La position des feuilles par rapport à l'origine de l'unité d'informations à laquelle elles appartiennent.

- La longueur de l'unité d'informations si elle est "répétitive".

- Un repère de thésaurus.
Sur la figure 6 on a le schéma d'une structure en mémoire. Elle n'est pas identique à la représentation en mémoire, mais est ainsi présentée pour être plus compréhensible.
Tous les cas représentés ne sont pas tirés de l'exemple : sondage carotté. On a seulement voulu décrire toutes les possibilités.

22 - Traitement d'un article :

Nous venons d'établir la table structure correspondant au type de structure AA.

Nous allons traiter un article dont la structure sera conforme au type de structure AA (cf paragraphe 11).
Toutes les informations concernant l'article sont sur cartes perforées.

221 - Exemple d'article :

Etudions un article contenant les informations suivantes :

- le noeud [AA] et les informations privilégiées
- le noeud [02] et à l'intérieur de celui-ci le noeud [03]
et le noeud [04] présent 2 fois.

Sur un même type de carte, on peut avoir un noeud de niveau 2 et un noeud de niveau 3 (exemple : 02 porte les noeuds 02 et 03) mais jamais deux noeuds de niveau 3 (cf début du paragraphe). Figure 6.

- le noeud [05] - le noeud [11] présent 2 fois
- le noeud [06] - le noeud [12] présent 3 fois
- le noeud [07]

. Dans le noeud [12] d'indice 1 on aura les noeuds [13],
[14] présent 2 fois,
[15] présent 1 fois,
[P8] présent 2 fois,
[EV] présent 3 fois,

. Dans le noeud [12] d'indice 2 on aura les noeuds [13],
[14] présent 3 fois,
[LØ] présent 2 fois,
[GR] présent 3 fois.

. Dans le noeud [12] d'indice 3 on aura les noeuds [13],
[PI] présent 2 fois,
[RC] présent 3 fois.

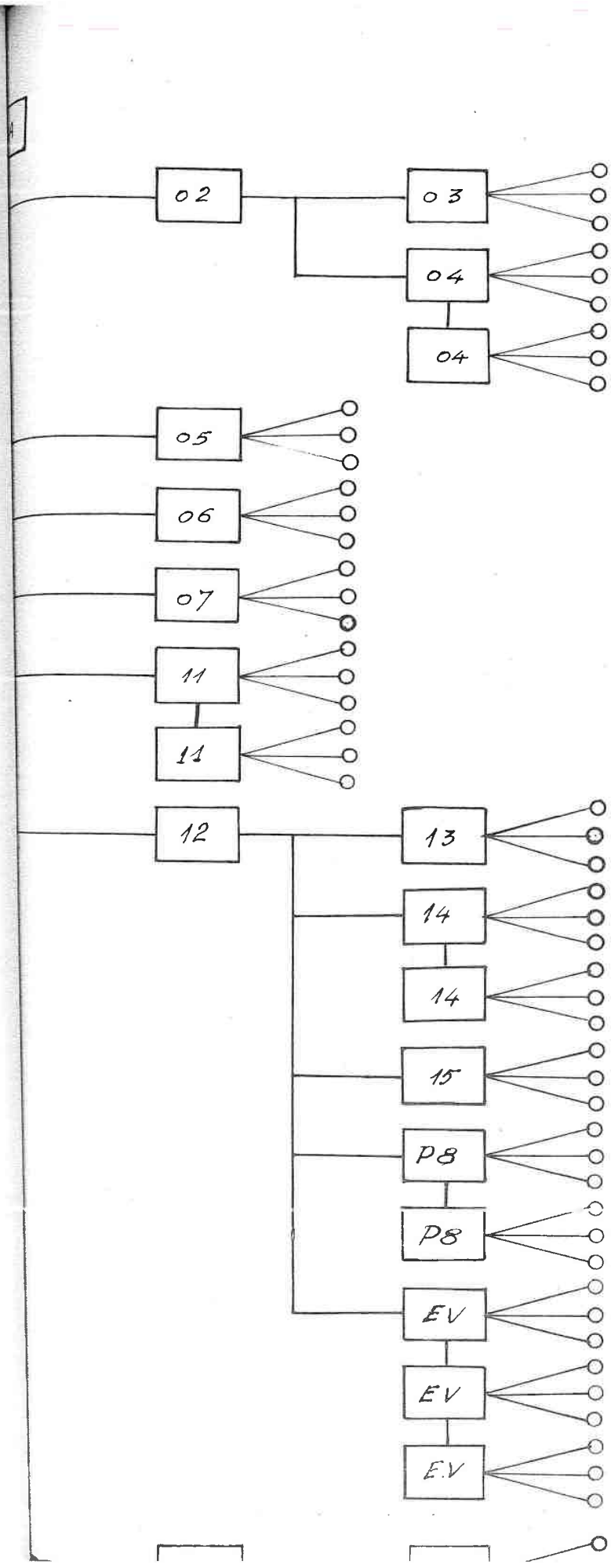
Sur la figure 7 on a la structure de l'article : sa structure propre est compatible avec le type de structure correspondant (voir descriptions paragraphe 21).
Les cartes décrites sur la figure 8 illustrent l'exemple choisi Le type de carte et le numéro de classement placé au début de chaque carte servent à faire des contrôles.

.../...

de carte	de noeud		identificateur de feuilles	fixe ou répétitif	taille en caractères	position/origine de l'unité d'informations	taille unité d'informations répétitive	thésaurus
AA	AA		CLAS	F	9	0		
			ZLAM		1	9		
			XLAM		0	10		
02	02		FIXE					
02		03	ØRGA	F	6	0		
			NØMI		10	6		
			ANNI		4	16		
03		04	FIAB	R	1	0	22	
			FØR1		5	1	22	
			FØR2		5	6	22	
04	05		FIXE					
			MØEU	F	31	0		
			PRØJ		36	31		
05	06		PLAL	R	V	0	V	
			SILI		5	V	V	
			ARGØ		6	V	V	
99	95		REPE					
99		96	PRØT	F	5	0		
98		97	TEXT	F	V	0		

Sur un même type de carte, on peut avoir un noeud de niveau 2 et un noeud de niveau 3 (exemple : 02 porte les noeuds 02 et 03) mais jamais deux noeuds de niveau 3 (cf début du paragraphe) .../...

Figure 6.



Le nœud [04] est présent 2 fois

Les nœuds [08] [09] [10] (cf § 21 structure) sont absents

Le nœud [11] est présent 2 fois

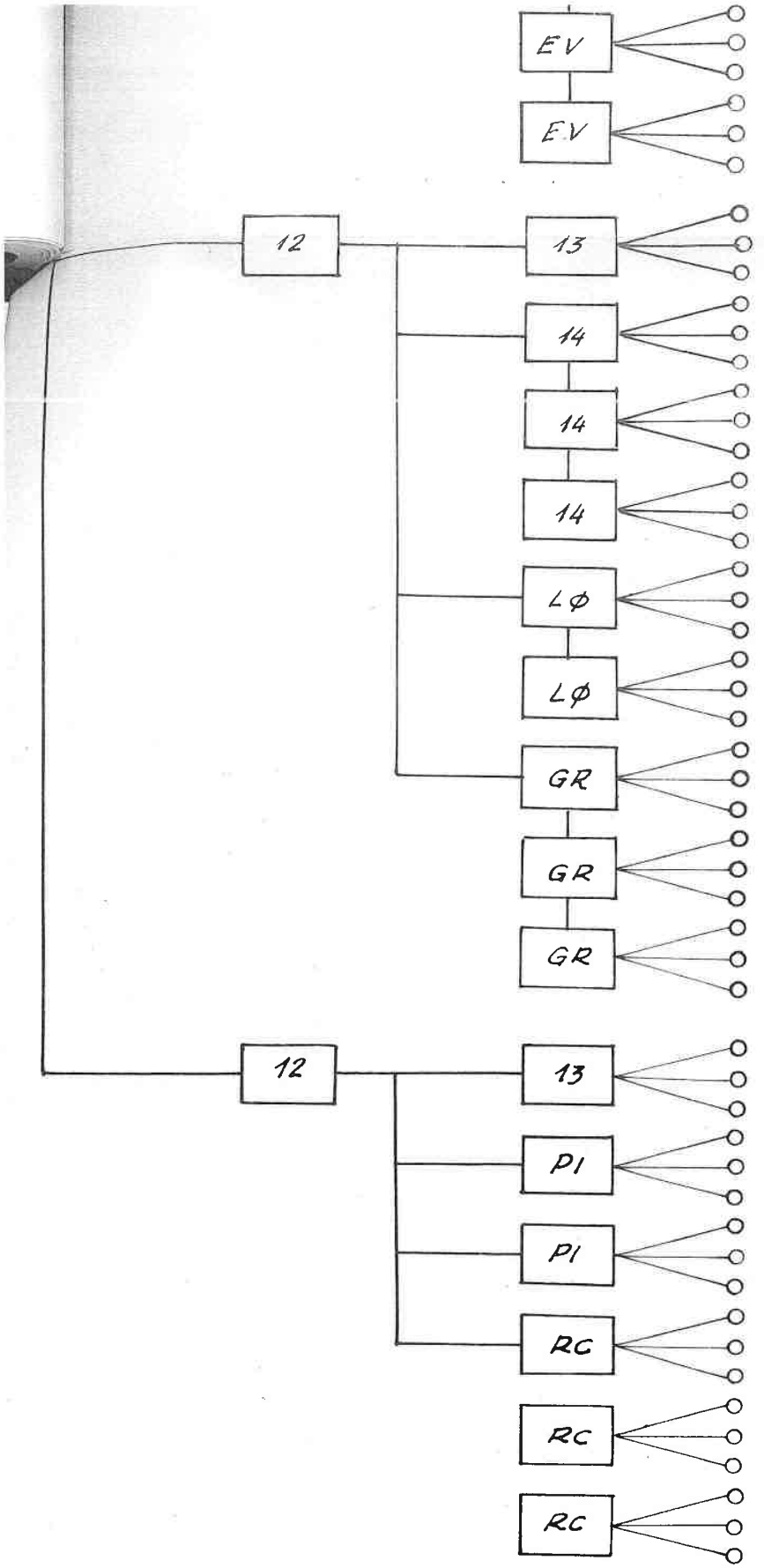
Le nœud [12] est présent 3 fois
Description du premier noeud 12

Le nœud [14] est présent 2 fois

Le noeud [P8] est présent 2 fois

Le noeud [EV] est présent 3 fois.
Les noeuds ES, DW, DD, LA, SC, LU, LG, LØ, DS, DH, FD, PØ, PI, PP, PL, PU, PE, WN, RC, VS, 2D, 2E, LG, LA, RG, VG, GR, SE sont absents

Description du deuxième



Les noeuds ES, DW, DD, LA, SC, LU, LG, LØ, DS, DH, FD, PØ, PI, PP, PL, PU, PE, WN, RC, VS, 2D, 2E, LG, LA, RG, VG, GR, SE sont absents

Description du deuxième noeud 12

Le noeud 14 est présent 3 fois

Le noeud LØ est présent 2 fois

Le noeud GR est présent 3 fois

Les noeuds 15, PB, EV, ES, DW, DD, LA, SC, LU, LG, DS, DH, FD, PØ, PI, PP, PL, PU, PE, WN, RC, VS, 2D, 2E, LG, LA, RG, VG, SE sont absents.

Description du troisième noeud 12.

Le noeud PI est présent 2 fois

Le noeud RC est présent 3 fois

Les noeuds 14, 15, PB, EV, ES, DW, DD, LA, SC, LU, LG, LØ, DS, DH, FD, PØ, PP, PL, PU, PE, WN, VS, 2D, 2E, LG, LA, RG, VG, GR, SE sont absents.
Les noeuds 95, 96 et 97 sont absents.

igure 8

N° de classement	Identificateur de nœud	N° de Class!	Z Lambert	X Lambert	Y Lambert	
	AA					— — — — — —

N° de Class!	Identificateur nœud de niveau 2	Nb. de nœuds de niveau 3	1er nœud de niveau 3	organisme	Nom	Inform fictive Observée	Année	
	02	2	03					— — — — —

N° de Class!	2e nœud de niveau 3	Nb repetitions de 04	Fiabilité	Form 1	Form. 2	
	04	2				— — — — — —

1er unité d'informations UI — 2e UI

N° de Class!	Nœud de niveau 2	Maitre d'œuvre	Projet
	05		

N° de Class!	Nœud de niveau 2	Entreprise travaux	Centralisateur emetteur	Labo essais sols	
	06				— — — — —

Figure 8 (suite)

carte	Nº de Classt	Noeud d niveau 2	But	Précision prof	C Subst	
5		07				— — — — — — — — — — — — — —

carte	Nº de Classt	Identificateur noeud niveau 2	Nb. répétitions de 11	Type piézo	Methode forage	
		11	2			— — — — — — — — — — — — — —

1ère UI

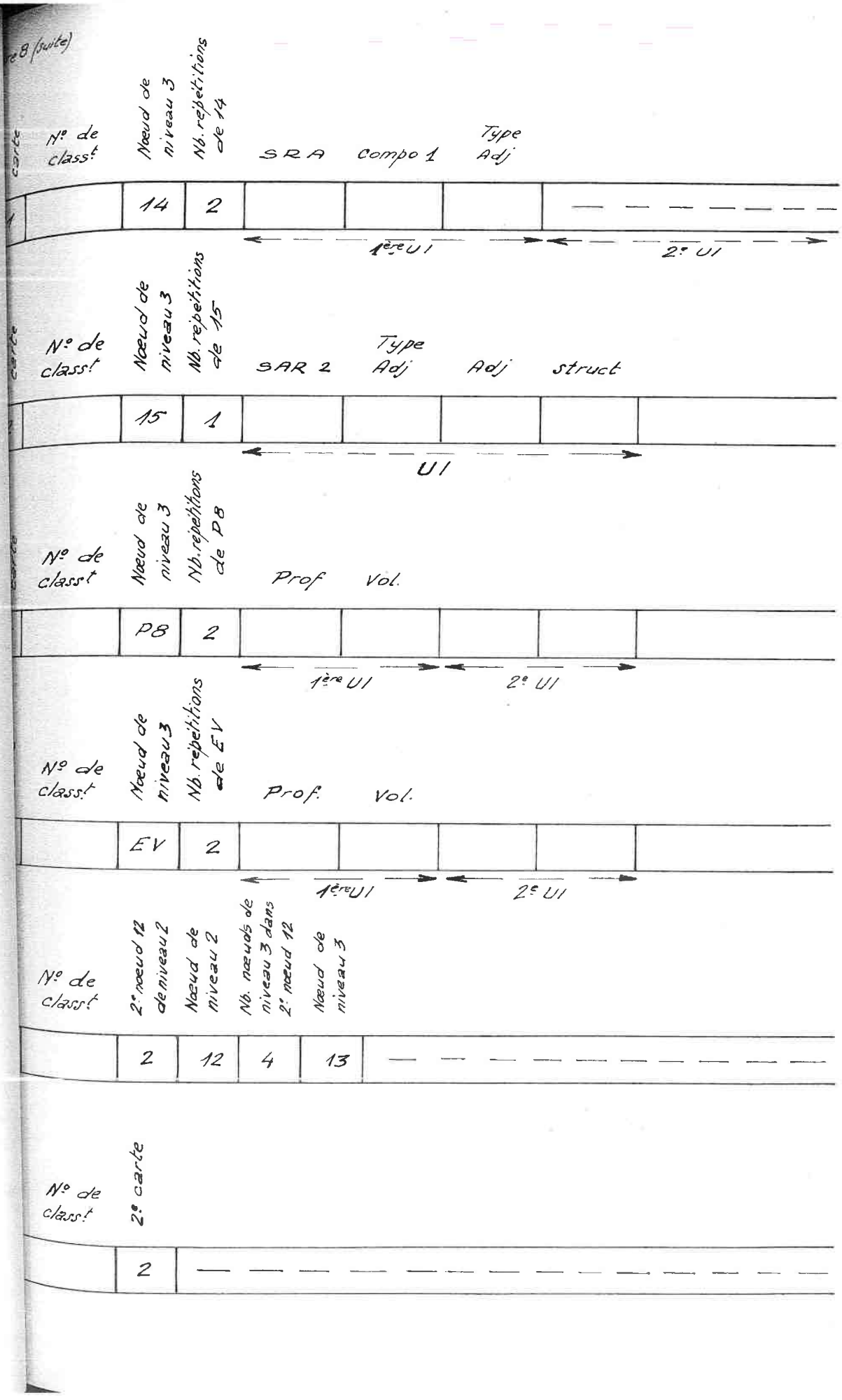
	Nº de Classt	nº suite	Type piézo	Methode forage	
		02			— — — — — — — — — — — — — —

2e UI

Nº de Classt	Nb. répétitions du noeud 12	Noeud de niveau 2	Nb noeuds de niveau 3 dans 1er noeud 12	Noeud de niveau 3	Profondeur	PTLØ	CPRE	
	3	12	5	13				— — — — —

Les feuilles appartenant au noeud 13 occupent 2 cartes

Nº de Classt	2e carte	C	Roche 3	
	2			— — — — — — — — — — — — — — — — — —


ce 8 (suite)
carte
N° de class!
Nœud de niveau 3
Nb. répétitions de 14
SRA
Compo 1
Type Adj
14
2
1ère UI
2e UI
carte
N° de class!
Nœud de niveau 3
Nb. répétitions de 15
SAR 2
Type Adj
Adj
struct
15
1
UI
carte
N° de class!
Nœud de niveau 3
Nb. répétitions de P8
Prof
Vol.
P8
2
1ère UI
2e UI
N° de class!
Nœud de niveau 3
Nb. répétitions de EV
Prof.
Vol.
EV
2
1ère UI
2e UI
N° de class!
2e nœud 12 de niveau 2
Nœud de niveau 2
Nb. nœuds de niveau 3 dans 2e nœud 12
Nœud de niveau 3
2
12
4
13
N° de class!
2e carte
2

ure 8 (suite)

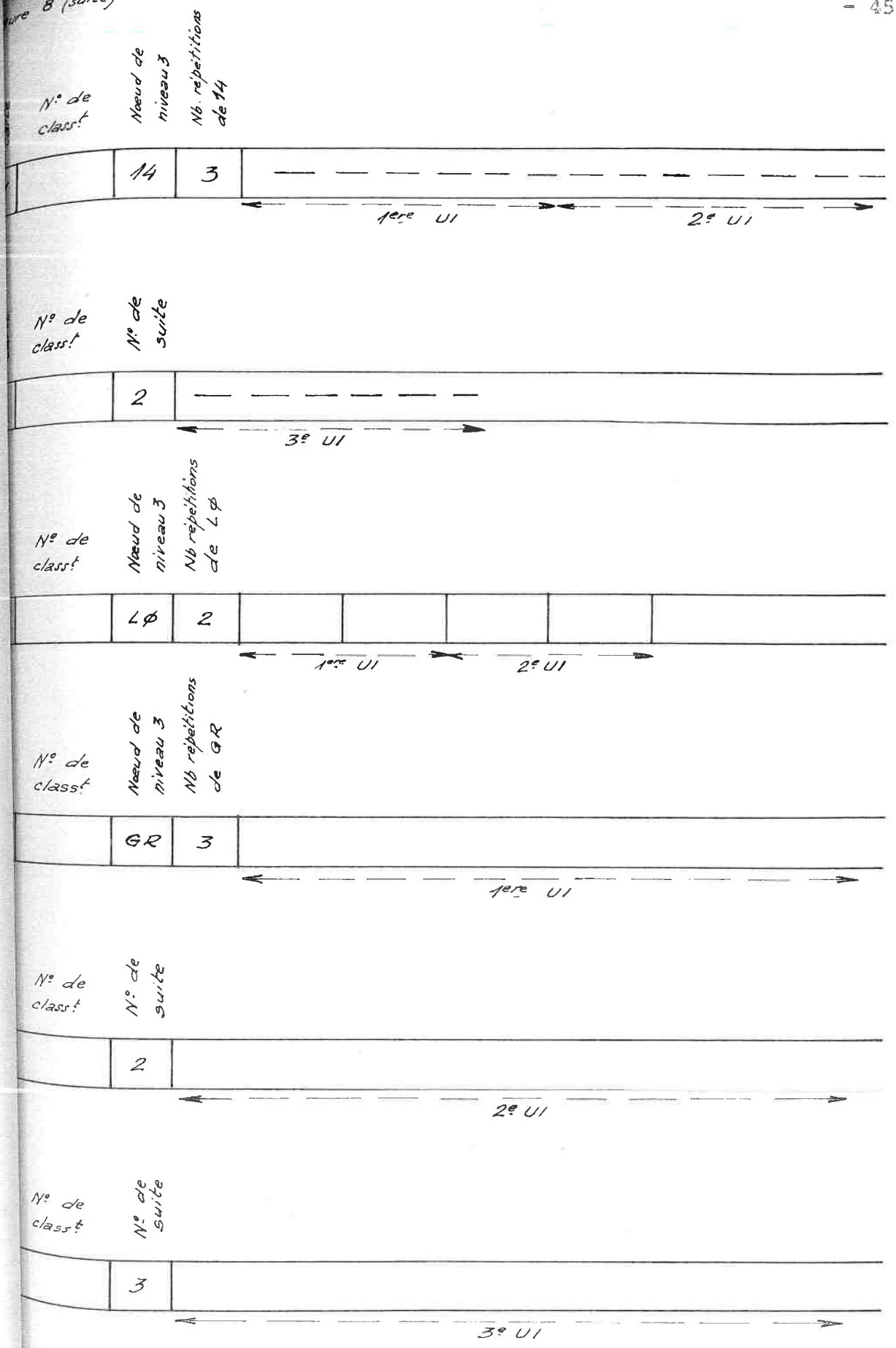
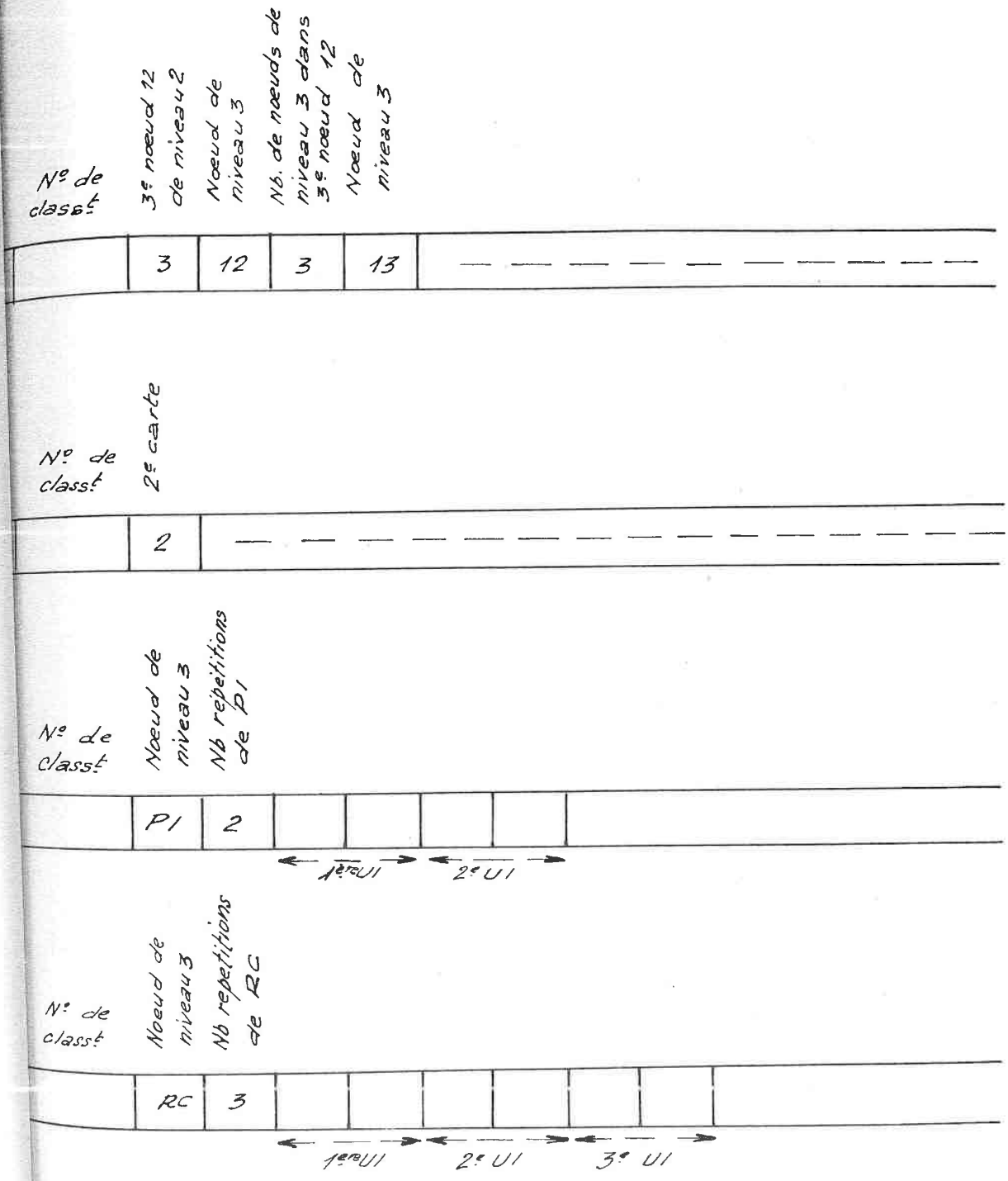

N.° de class.t
Nœud de niveau 3
Nb. répétitions de 14
14
3
1ere UI
2e UI
N° de class.t
N.° de suite
2
3e UI
N° de class.t
Nœud de niveau 3
Nb répétitions de LØ
LØ
2
1ere UI
2e UI
N° de class.t
Nœud de niveau 3
Nb répétitions de GR
GR
3
1ere UI
N° de class.t
N.° de suite
2
2e UI
N° de class.t
N.° de suite
3
3e UI

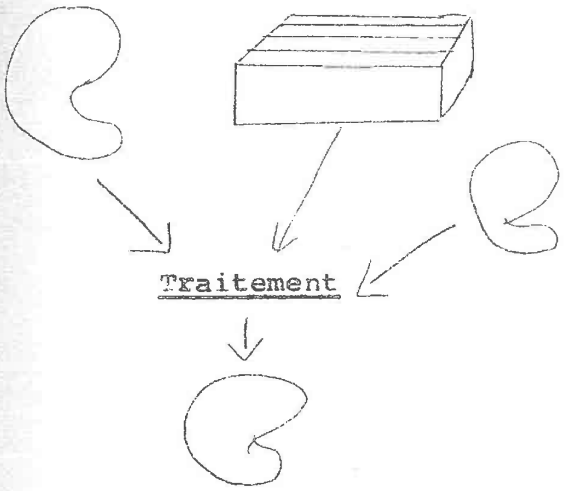
Figure 8 (suite)



Toutes ces cartes représentent donc les données

222 - Représentation de l'enregistrement en mémoire :

fichier structure



- Cartes décrivant l'article. Ce sont celles qui viennent d'être présentées (figure 8)

Programme général de traduction d'un article en fonction du type de structure et éventuellement programme spécifique de contrôle.

Ce programme engendre une table (description plus loin) qui forme avec l'article une fois traité un enregistrement.

l'enregistrement (table + article)
est rangé dans un fichier.

Pendant l'exécution du programme on range toutes les données dans une zone MEM, les identificateurs de noeuds ont alors tous disparu.
Pendant l'exécution de ce rangement, on construit dans une zone TAB une table des noeuds (cf paragraphe 1522) comprenant :

- la taille de l'enregistrement
- la taille de la table
Les deux zones ne sont remplies qu'en fin de programme.
- les identificateurs des noeuds :seuls ceux qui sont présents dans l'article.
- la position de chaque noeud par rapport à l'origine de l'enregistrement : c'est-à-dire son adresse de début.
- les différentes répétitions possibles.

Sur la figure 9 on a une description de la table correspondant à l'exemple du paragraphe 221.
Des explications plus détaillées seront données sur cette table dans le chapitre suivant en particulier sur la signification de multiplicité différente suivant les niveaux.

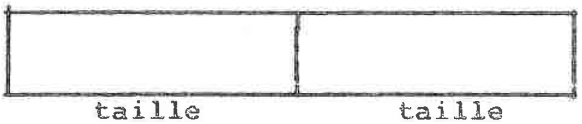
.../...

Avant de ranger MEM précédé de TAB sur mémoire secondaire, on pourra éventuellement faire des contrôles supplémentaires (cf paragraphe 162) : cohérence de certaines informations, présence ou absence de certains noeuds .....
En effet, par la suite, la présence ou l'absence d'un noeud ne pourra être testée que par sa présence ou son absence dans TAB.
On a donc maintenant avec la table structure et la table des noeuds tous les éléments nécessaires pour obtenir soit une information élémentaire soit un noeud de niveau 3, 2 ou 1 (cf paragraphe 15). Nous verrons progressivement pourquoi nous avons autant de renseignements dans la table structure.

Conclusion : Cet exemple nous a permis de bien définir les possibilités offertes par le programme général.
Nous appelons pour le moment programme général, l'ensemble de deux programmes :
le programme de construction de la table structure ou programme "fichier structure" et le programme de construction de l'enregistrement ou programme "fichier maître".
Ces programmes ont deux rôles bien distincts. Le programme "fichier maître" ne peut-être exécuté que si le programme "fichier structure" a déjà été exécuté. En effet, il a besoin d'une ou plusieurs tables structures (cas où il y a plusieurs articles de structures différentes) construites par le programme "fichier structure"; Une fois ces tables formées, le programme "fichier maître " pourra être exécuté plusieurs fois de suite. Le chapitre suivant v a nous montrer comment sont présentées les informations nécessaires à la construction des deux tables et de l'enregistrement puis nous verrons la représentation interne de ces deux tables. Les explications sont parfois assez complexes mais il est utile de bien les comprendre pour tirer parti de toutes les possibilités offertes.

.../...

figure 9



identificateur de noeud	position/origine d'enregistrement	multiplicité
AA	0	0
02	~	0
03	~	0
04	~	2
05	~	0
06	~	0
07	~	0
11	~	2
12	~	3
13	~	0
14	~	2
15	~	1
P8	~	2
EV	~	2
12	~	2
13	~	0
14	~	3
LØ	~	2
GR	~	3
12	~	1
13	~	0
PI	~	2
RC	~	3

.../...

3 Analyse organique

3 - Analyse organique :

31 - Introduction :

Nous venons de voir que notre programme général était un programme de traitement pour fichiers ayant des enregistrements de grande variété de structures : les fichiers se trouvent sur bande.

- le programme général n'est pas seulement un ensemble de deux programmes comme nous l'avons vu jusqu'à présent mais une bibliothèque de programmes.
Il y a d'abord les deux principaux programmes dont nous avons parlé puis des programmes de service.

. Les deux programmes principaux :

* le premier, à l'aide de cartes décrivant les structures, (cartes que nous allons étudier) forme une table structure (un fichier par structure) pour chaque structure.

* le second, à l'aide de cartes décrivant l'article (cartes que nous allons étudier) et de la table structure correspondant au type d'article, construit un enregistrement (table des noeuds + article traité).

Le premier est celui que nous avons appelé "fichier structure", le second celui que nous avons appelé "fichier maître".

. Les programmes de service :

Il y a un programme de mise à jour des dictionnaires (cf paragraphe 3233)
Il y a des programmes d'édition :
Edition d'une table structure,
Edition des enregistrements du fichier maître,
Edition des dictionnaires.

- Remarque : nous l'avons déjà mentionnée dans le paragraphe 11.
Sur la même bande nous pouvons avoir plusieurs fichiers structures que l'on distingue les uns des autres par un label.
Sur l'article on retrouve le label correspondant à sa structure.
Le programme "fichier maître" recherche donc grâce à ce label la structure de l'enregistrement sur la bande portant les fichier structures, la met en mémoire centrale et commence le traitement.
Les fichiers structures pouvant avoir une taille très importante, nous n'avons pas prévu d'en avoir plusieurs présents en même temps en mémoire centrale.

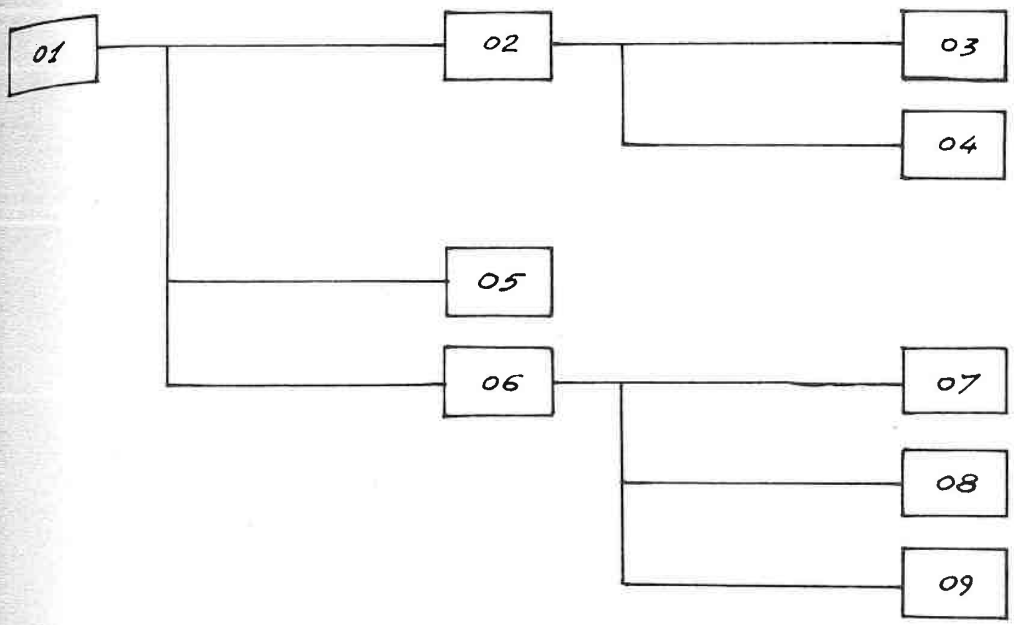
.../...

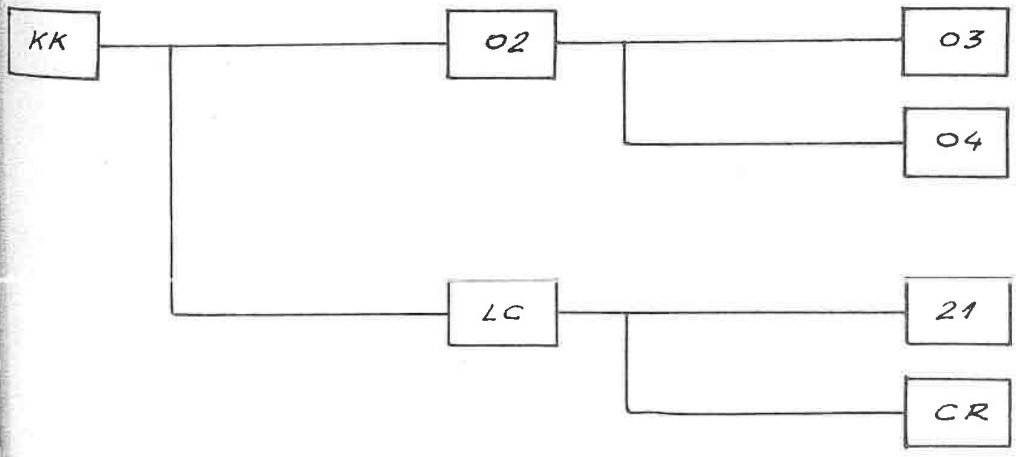
Si l'utilisateur veut former un fichier ayant des articles correspondant à différents types de structure, il devra les regrouper par type pour éviter de nombreux échanges bande-mémoire centrale.
Exemple : un géologue a différents types de structure :

AA : sondage carotté
BB : sondage tarière
CC : affleurement
DD : ballastière.

On pourrait former des fichiers avec des articles ayant tous la même structure mais si on s'intéresse par exemple à des fichiers géographiques, il peut-être préférable de former un fichier comportant toutes les recherches géologiques faites par exemple en Meurthe-et-Moselle.
On aura donc un fichier Meurthe-et-Moselle regroupant plusieurs types de structure.
Si on organise les articles en mettant d'abord tous les articles type sondage carotté puis tous les articles type ballastière et ainsi de suite, il y aura un seul échange bande-mémoire centrale pour chaque type de structure. S'ils sont mélangés, il y aura un échange pratiquement à chaque nouvel article traité.
Sur la bande fichier Meurthe-et-Moselle, on aura alors un nombre d'enregistrements de structure type AA puis un nombre d'enregistrements de structure type DD puis BB, CC .....
puis de nouveau AA si on fait une mise à jour.
En effet, initialement les articles sont regroupés par type, mais la mise à jour, du moins telle que nous la concevons actuellement fait apparaître des répétitions de structures.
Pour exploiter les enregistrements de cette bande, on aura besoin de plusieurs types de structure. Le problème de la présence de plusieurs structures en mémoire se pose de nouveau.
une solution envisagée mais non encore réalisée est de ne garder en mémoire que des parties de fichiers structures.
Prenons un exemple dont la structure est présentée sur la figure 10.
On a trois niveaux. Si la question porte sur une information élémentaire de niveau 3 appartenant au noeud [09] on ne gardera en mémoire que [09]
et les noeuds de niveau supérieur desquels il dépend c'est-à-dire la branche 01-06-09
Si la question est : rechercher tous les éléments ayant (caractéristique) ou (autre caractéristique), chaque caractéristique correspondant à un type de structure différent,

.../...


01
02
03
04
05
06
07
08
09

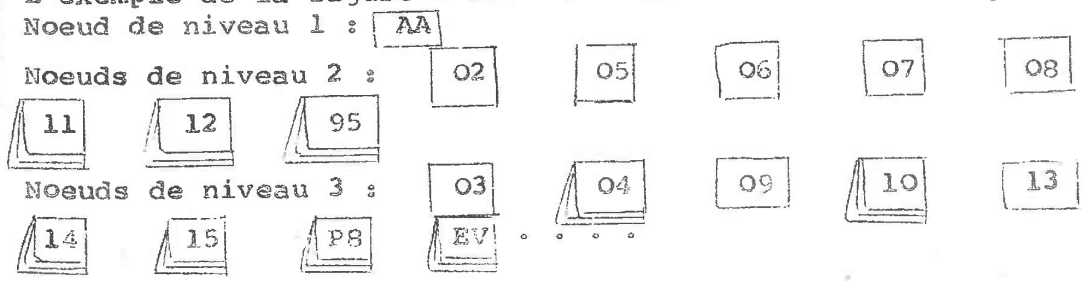

KK
02
03
04
LC
21
CR

On ne gardera que les deux branches nous intéressant dans chaque structure.
Exemple : figure 10
Première caractéristique porte sur 09 dans la structure 01
Deuxième caractéristique porte sur CR dans la structure KK
On gardera en mémoire les deux branches :
01 - 06 - 09 et KK - LC - CR.

32 - Fichier structure ou table structure :

321 - Généralités :

Nous avons vu que pour décrire une structure nous avions 3 niveaux possibles : M1, M2, M3 : On aura au moins le niveau 1 et un niveau 2 présents. Les autres sont facultatifs. (cf paragraphe 1221 : remarque).
Reprenons l'exemple de structure que nous avions vu dans le chapitre 2 : il s'agissait d'un sondage carotté : figure 5.
Les noeuds correspondant aux différents niveaux permettent de faire une correspondance entre la structure et l'article.
Si tous les noeuds et toutes les informations élémentaires appartenant à ces différents noeuds et décrits dans la table structure étaient obligatoirement présents et si toutes les informations étaient de longueur fixe, on n'aurait pas besoin de retrouver dans l'article une correspondance avec la table structure. Ce n'est pas notre cas et si un noeud est absent ou "répétitif", il faut avoirla possibilité de s'en rendre compte.
Nous allons dans certains cas pour rendre les explications plus compréhensibles reprendre certaines informations que nous avons déjà présentées dans les chapitres précédents.
L'exemple de la figure 5 met en évidence tous les cas possibles.



Le noeud de niveau 1 est obligatoirement présent.
Les nœuds de niveau 2 peuvent être présents, absents, répétés, au moins un doit être présent.

.../...

Les noeuds de niveau 3 peuvent être présents, absents, répétés. Nous avons vu que tous les noeuds pouvaient être absents ou répétitifs. Lorsqu'une structure a été établie il peut-être logique que certains noeuds soient absents ou "répétitifs" et pas d'autres. C'est dans cette optique que nous allons donner des exemples .

Au niveau 2 pourra être :

- absent : le noeud 11 piézomètre
- répétitif : le noeud 12 description par couche.

Au niveau 3 pourront être :

- absents : les noeuds P3 EV ... tous les essais
- répétitifs : les noeuds 04 désignation stratigraphique

10 caractéristiques / profondeur 15 : adjectifs

Le noeud 12 qui est "répétitif", à chaque itération ne comportera pas le même nombre de noeuds puisque comme on vient de le voir certains noeuds de niveau 3 qui lui appartiennent peuvent être absents ou "répétitifs". Ceci nous prouve (voir début du chapitre) l'utilité de retrouver dans l'article les différents noeuds décrits avec des identificateurs identiques à ceux de la structure.

Le niveau 3 lorsqu'il est présent, comporte les informations élémentaires sinon on les retrouve au niveau 2.

322 - Représentation externe du fichier structure : les cartes

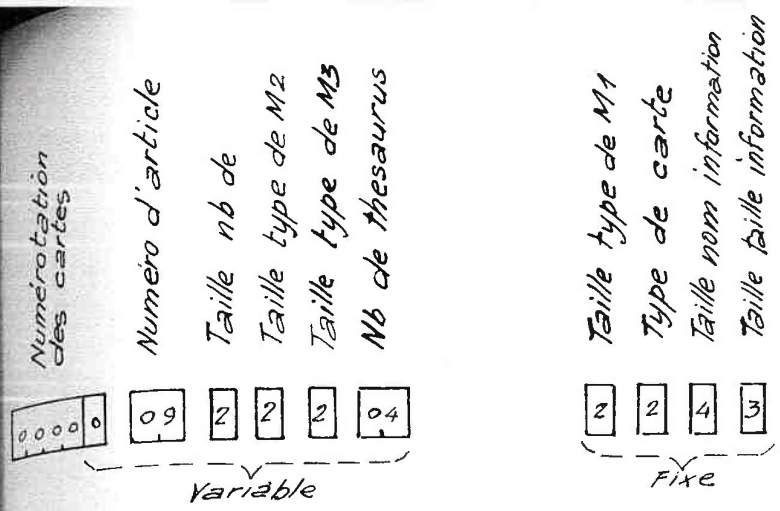
On aura un certain nombre de cartes.

3221 - La carte 0 :

Elle contient des paramètres. Elle donne des indications sur les tailles des éléments décrits : variables et non variables. La numérotation des cartes sert à vérifier s'il n'y a pas eu d'intervention.

0 : on pourrait assimiler cette carte à un niveau 0.

.../...



32211 - Les éléments variables :

Ce sont ceux qui d'une structure à l'autre seront différents.

- Numéro d'article : c'est la taille du numéro de classement de l'article traité. On retrouvera ce numéro sur toutes les cartes décrivant l'article. Cette répétition du numéro de classement sert de vérification : c'est un test d'erreur.

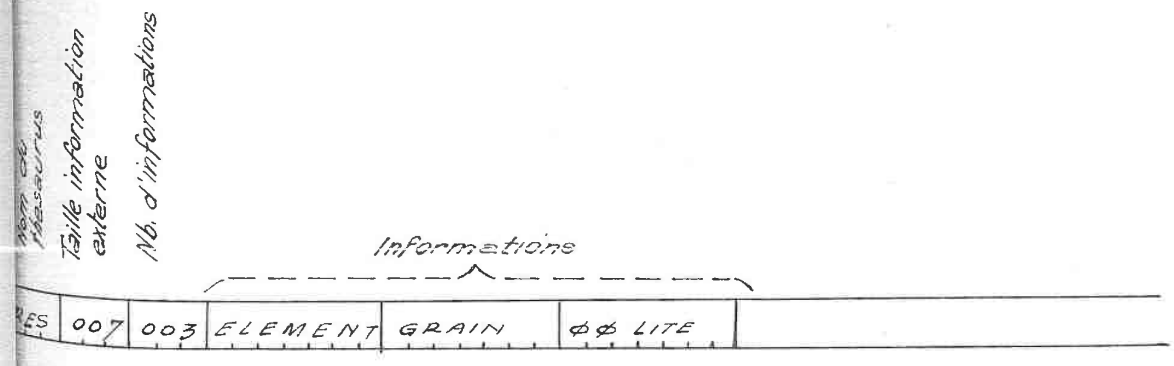
- Taille "nb de" : Lorsqu'une unité d'informations est "répétitive", il faut connaître le nombre d'itérations. Il est donné par un nombre : l'"information de service", "nb de " précise la taille de ce nombre, c'est-à-dire le nombre de caractères qu'il occupe sur les cartes décrivant l'article.
De même lorsqu'un noeud de niveau 2 est "répétitif", il faut connaître le nombre d'itérations : il s'agit toujours de l'"information de service"(cf paragraphe 141).

- Taille type de M2 et taille type de M3 :
Les différents noeuds de niveau 2 ou 3 sont représentés par des identificateurs.
"taille type de M2 ou type de M3" précise la place occupée par ces identificateurs : c'est-à-dire leur taille.
Si l'on a donné la possibilité de fixer la taille de "type de M2 ou de M3" au début de chaque structure, c'est afin de pouvoir choisir un identificateur mnémonique (cf paragraphe 1221). C'est le cas des essais où les identificateurs de noeuds sont :
P8 passing à 80 u
ES : ES piston
DS : deval sec .....

.../...

-"Nb de thésaurus" :
Etudions d'abord le cas où l'on n'a encore aucun fichier structure sur bande.
On a parlé d'un repère de thésaurus. C'est un nombre précisant le thésaurus que l'on veut utiliser.
Derrière les cartes décrivant la première structure que l'on met sur bande il faut avoir les cartes décrivant tous les thésaurus mentionnés dans la structure. Dans "nb de thésaurus" on mettra donc le nombre de thésaurus que l'on va trouver derrière la structure.
On veut ensuite mettre sur bande une deuxième structure dont certains numéros de thésaurus mentionnés dans les répères de thésaurus sont les mêmes que dans la première structure.
On ne va donc pas mettre derrière les cartes décrivant cette seconde structure les cartes décrivant les thésaurus existant déjà. On ne mettra que celles décrivant les thésaurus jamais mentionnés.
Dans ce cas "nb de thésaurus" désigne le nombre de nouveaux thésaurus et pas tous les thésaurus effectivement présents dans la structure.
Il y a trois sortes de thésaurus (cf paragraphe 1222).
La représentation que nous allons en donner est la représentation externe : sur cartes.
Nous verrons la représentation interne dans le paragraphe 32332.
C'est dans ce paragraphe que nous expliquerons l'utilité du numéro de thésaurus que l'on trouve au début des cartes.

. Le premier thésaurus sert uniquement à des contrôles et se présente de la façon suivante :



.../...

On a ainsi la taille de l'information et le nombre d'informations présentes dans le dictionnaire. On compare donc l'information à tester avec la liste des informations présentes dans le dictionnaire. Cette comparaison se fait lorsqu'on étudie l'article, c'est-à-dire lorsqu'on utilise le programme "fichier maître". Si l'information à tester n'est égale à aucune information de la liste, il y a erreur, sinon on poursuit l'exécution du programme.

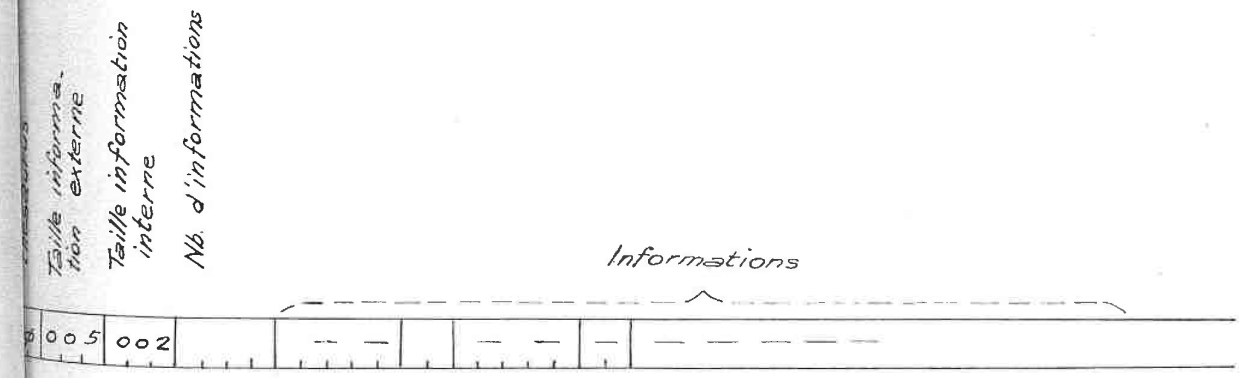
. Le deuxième thésaurus sert non seulement à coder et réciproquement à décoder mais aussi au contrôle.
L'intérêt du codage est de gagner de l'espace machine (mémoire centrale et bande magnétique) et d'augmenter la densité d'informations.
Par exemple une roche codée 123 :
1 représentera la famille de la roche,
2 sa structure,
3 la roche elle-même.
Le deuxième thésaurus se présente de la façon suivante :

Nom du thésaurus	Taille information externe	Taille information interne	Nb d'informations	Informations									
CODE	005	002	005	QUATI	01	TØARC	02	ALREC	03				

.../...

On a la taille de l'information externe, la taille de l'information codée et le nombre d'informations présentes dans le dictionnaire.
On compare l'information à coder avec chacune des informations de la liste des informations codées dans le dictionnaire. Si elle n'est égale à aucune, il y a erreur. Si on trouve l'information identique on remplace l'information de l'enregistrement par le code du dictionnaire et on poursuit le programme.
Nous avons vu dans le paragraphe 213, représentation de la structure en mémoire, la signification de nom information et taille information. Dans le cas d'une information à coder, la taille sera celle de l'information une fois codée.
Ainsi à l'exploitation, on n'aura pas besoin de repasser par le dictionnaire pour avoir la taille codée. Si la question est codée on aura immédiatement l'information ; sinon on passera par le dictionnaire pour connaître le code.
Si on veut imprimer "en clair" des informations codées, il suffit de faire le travail inverse : à partir du code rechercher l'information et l'imprimer.

. Le troisième thésaurus sert à <u>décoder</u>. Il ne sera donc utile qu'à l'exploitation. L'information élémentaire a été entrée codée en machine. Le troisième thésaurus se présente de la façon suivante :



Le travail est identique au cas précédent quand il s'agit de décodage.

32212 - <u>Les éléments fixes</u> :

Ils sont mis là, à titre de vérification.

.../...

- Taille type de M1 :
On a mis l'identificateur de niveau 1 fixe car il représente le label du fichier structure. On le retrouvera sur la description de l'article comme premier élément.
Tant qu'on ne connait pas la structure de l'article on ne peut commencer le traitement. Si "taille type de M1" était variable comment retrouverait-on la structure correspondant à l'article à traiter puisque c'est dans cette structure que se trouve "taille type de M1".
Il y aurait eu un autre moyen si le label n'avait pas été type de M1 mais un nom quelconque.
On aurait pu rajouter une carte avant le début des cartes décrivant l'article et qui aurait porté le label dont la fin serait exprimée par une étoile : exemple SØNDA *
Il aurait alors fallu une carte supplémentaire devant chaque article. On a préféré la solution économique.

- "type de carte" :
L'article est décrit par des cartes portant les différents noeuds
Aux mêmes types de carte doivent toujours correspondre les mêmes noeuds.
"type de carte" représente la place occupée par le type de carte : c'est-à-dire sa taille.
Sur un même type de carte on pourra décrire
un noeud de niveau 2 et un noeud de niveau 3
ou un noeud de niveau 2
ou un noeud de niveau 3
mais jamais deux noeuds de niveau 3 ensemble.
Exemple : deux noeuds de niveau 3 sont décrits sur une même carte. Le premier est absent dans un article, pour trouver le second il faudra tester toute une serie de caractères blancs, ce qui, si le premier noeud de niveau 3 est "répétitif" n'est pas possible.
Donc si derrière un noeud de niveau 2 qui est normalement suivi de noeuds de niveau 3 on a un caractère blanc, c'est que le premier noeud de niveau 3 est absent et on passe à la carte suivante.
Ceci est une précaution. En effet, si on avait autorisé la présence de deux noeuds de niveau 3 ensemble, il aurait fallu faire en sorte que le premier soit obligatoirement présent.

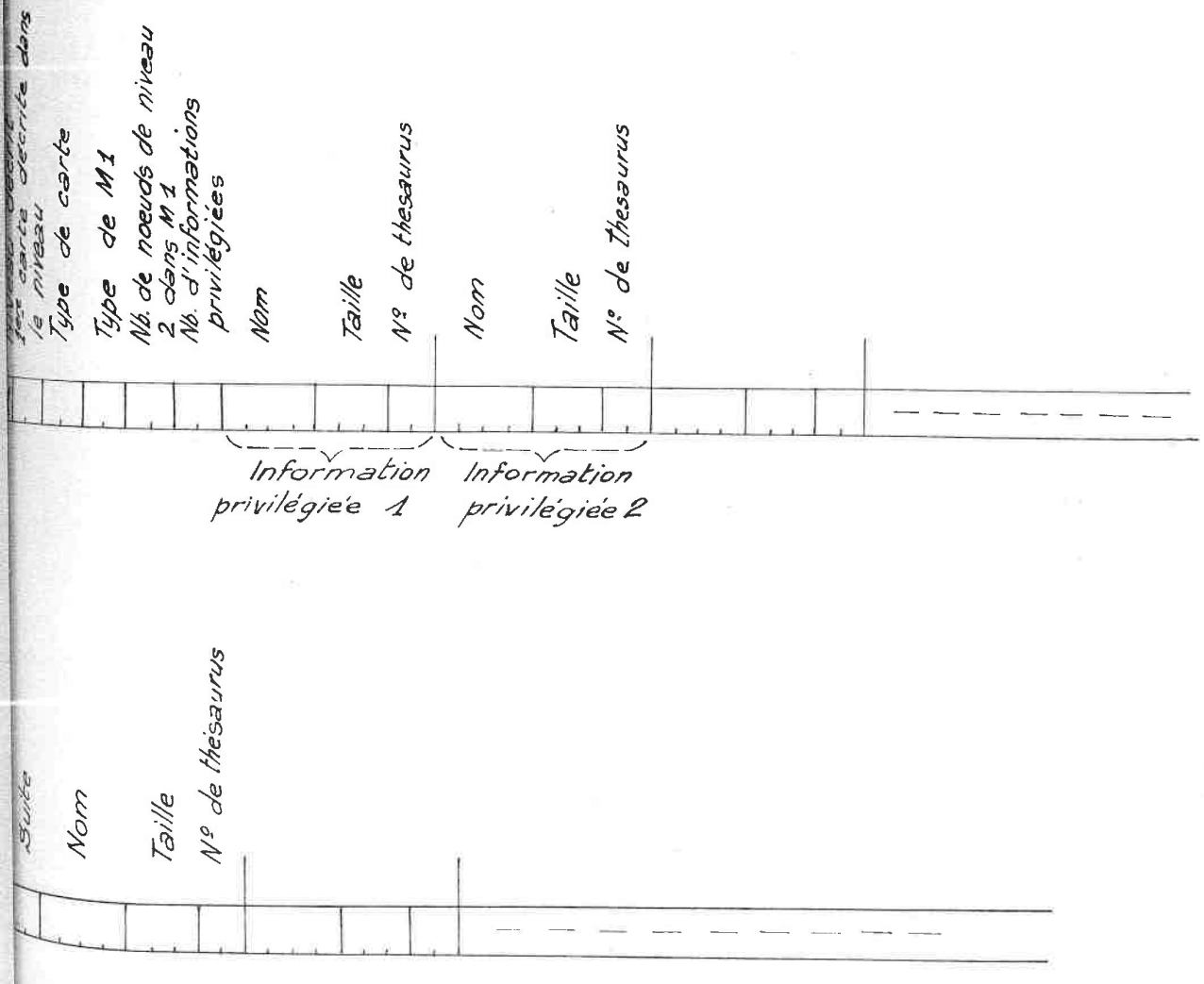
- "Taille nom information":
On l'a fixée à 4 caractères, ce qui donne déjà de nombreuses possibilités. On a 26 lettres et 9 chiffres à combiner entre

.../...

eux à raison de 4 caractères ensemble. Cela fait $C_4^{35} = 52360$ possibilités.
Ce "nom information" représente l'identificateur de chaque information élémentaire de l'enregistrement. On pourra donc, nous en avons déjà parlé, à l'exploitation extraire et traiter n'importe quelle information grâce à son identificateur.

- "taille de la taille de l'information" :
On l'a fixée à 3 caractères.
C'est la taille de la taille de chaque information élémentaire.
Elle se trouve après le nom.

3222 - Une carte de niveau 1 :



.../...

- On a une numérotation des cartes pour l'ensemble des cartes de structure.

- On a précisé le niveau décrit

- A l'intérieur du niveau un ordre des cartes est établi : sur la première on aura la première carte de niveau décrit sur les autres on aura la jème carte j=j+1.

- Type de carte : ce niveau dans l'article sera décrit sur ce type de carte, ce niveau lui-même se retrouvant sous le nom type M1.

- Nous avons vu précédemment que type M1 avait une longueur fixe parce qu'il correspondait au label du fichier structure. On doit le retrouver au début de l'article; à l'endroit où se situe normalement le type de carte : pour le niveau 1 type de carte sera alors égal à type de M1.

- On a le nombre de noeuds de niveau 2 dans M1.
On aurait pu ne pas le préciser et déterminer la fin des cartes de structure par une fin anormale, la non existence des cartes sur le lecteur de cartes. Ceci est possible si l'on ne met sur bande qu'une structure à la fois or on peut en mettre plusieurs. On risquerait alors de considérer la première carte de la nouvelle structure comme un autre niveau 2 pour la structure précédente.
On aurait pu aussi préciser le nombre de cartes présentes dans chaque structure mais on a trouvé que le nombre de noeuds de niveau 2 était plus élégant.

- Le niveau 1 comprend un certain nombre d'informations élémentaires : les "informations privilégiées". Nous avons vu dans le paragraphe 1221 : remarque, que ces informations étaient utilisée comme critères de sélection courants. On va donc chercher à les atteindre le plus rapidement possible et ainsi on les place à la tête de l'article sur bande. Grâce à la table structure on les obtiendra facilement : en effet ils sont de longueur fixe : on connait donc la position de chacun d'eux par rapport à l'origine de l'article.
Sur cette carte de structure on a le nombre d'"informations privilégiées" dont on précise le nom, la taille et éventuellement un repère de thésaurus.

- Sur cette carte sont de longueur variable "nb de niveaux^2 dans M1"; son nombre de positions occupées sur la carte est le même que celui "taille type de M2" de la carte 0.

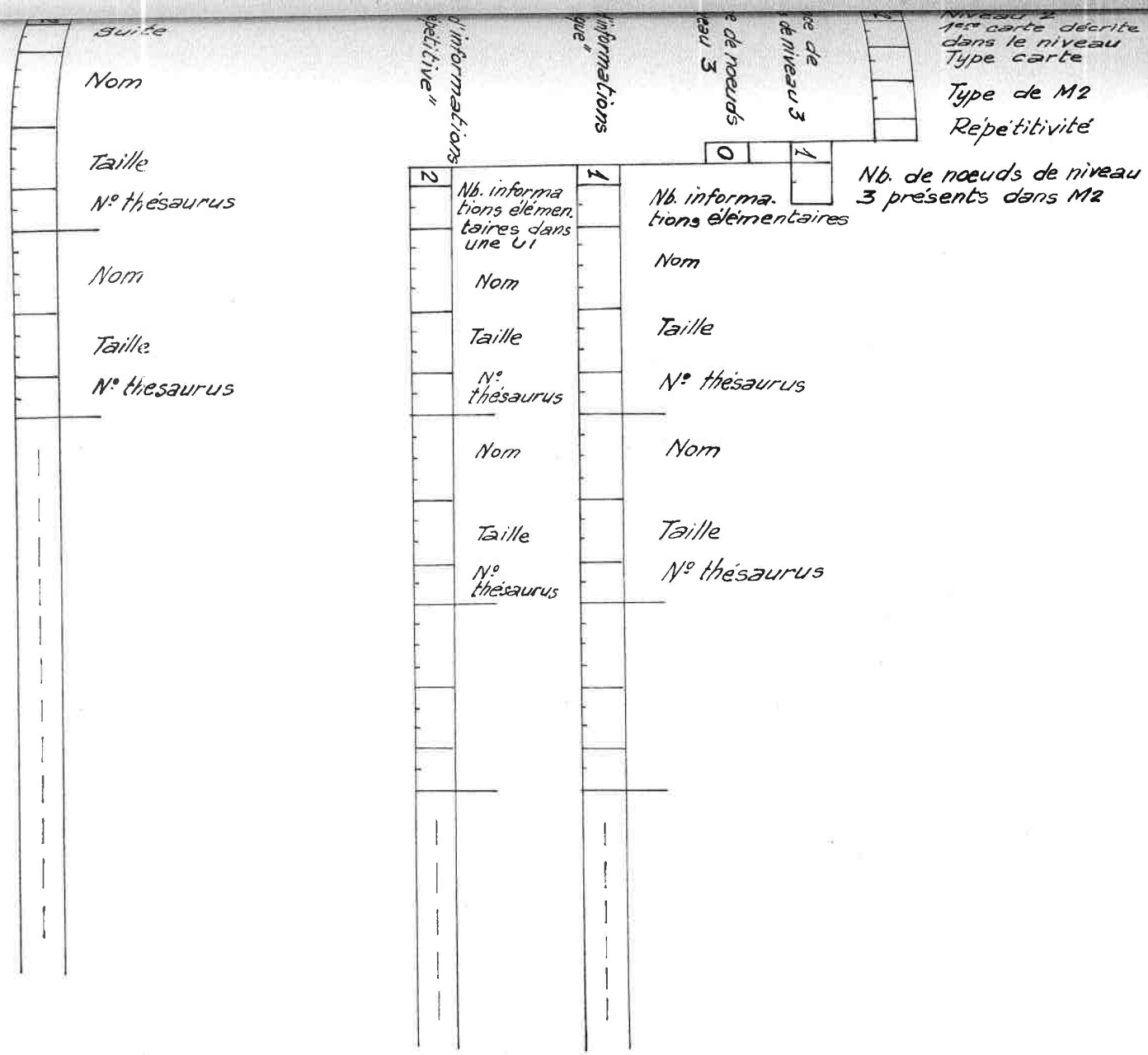
.../...

"Nb d'informations privilégiées" : sa taille est celle de "nb de " voici dans le fichier structure la représentation du noeud de niveau 1.

type N1	Nom	Fixe	Longueur propre	position / origine du noeud	x x x	Thésaurus
AA	CLAS	F	9	0	Ne concerne pas le niveau 1.	
	ZLAM	F	1	9		
	XLAM	F	7	10		
	YLAM	F	6	17		

3223 - Une ou plusieurs cartes de niveau 2 :

.../...


Suite
Nom
Taille
N° thésaurus
Nom
Taille
N° thesaurus
d'informations
répétitive"
2
Nb. informations élémentaires dans une U1
Nom
Taille
N° thésaurus
Nom
Taille
N° thésaurus
informations
que"
1
Nb. informations élémentaires
Nom
Taille
N° thésaurus
Nom
Taille
N° thésaurus
de noeuds
eau 3
0
de niveau 3
1
Nb. de noeuds de niveau 3 présents dans M2
Niveau 2
1ère carte décrite dans le niveau
Type carte
Type de M2
Répétitivité

Comme pour le niveau 1 on a d'abord une numérotation des cartes, le niveau décrit, la première carte de niveau décrit, le type de carte, le type de M2.
Le noeud de niveau 1 est unique mais on aura autant de cartes de niveau 2 que de noeuds de niveau 2 effectivement possibles.
Tous les noeuds de niveau 2 sont obligatoirement décrits dans la structure mais comme nous l'avons déjà précisé dans chaque enregistrement, ils pourront être ou présents,
ou absents,
ou répétés.

32231 - Noeud de niveau 2 "répétitif" :

La possibilité de répétition est décrite dans la carte de structure : 1 répétition
0 pas de répétition.
En effet, s'il y a répétition il faut en connaître le nombre : celui-ci pouvant être variable. Dans l'enregistrement, comme pour le niveau 1, on retrouve le type de carte et le type de M2.
S'il y a répétition juste avant le type de M2 on aura le nombre de répétitions (cf "information de service"). La place occupée par ce nombre est donnée par "nb de".
Le niveau 2 peut-être seul mais il peut aussi être suivi d'un ou plusieurs noeuds de niveau 3, ce qui est précisé dans la structure.

32232 - Présence de noeuds de niveau 3 :

On aura 1 suivi du nombre de noeuds de niveau 3.
Après la carte décrivant le noeud de niveau 2 on aura donc 1 et plusieurs cartes de niveau 3 pour décrire ces noeuds de niveau 3.

32233 - Absence de noeuds de niveau 3 :

- On aura 0.
Cela signifie que les informations élémentaires se trouvent au niveau 2.
Deux cas peuvent se présenter : l'unité d'informations qui regroupe les informations élémentaires peut-être "unique" ou "répétitive".

. Unité d'informations "unique" :
On aura 1 suivi du nombre d'informations élémentaires puis la description de ces informations élémentaires avec leur nom, taille et éventuellement repère de thésaurus.
Si une carte n'est pas suffisante pour leur description, on aura des cartes "suite" où on répétera le numéro du niveau décrit, le numéro de la carte suite : j=j +1 et nom, taille, éventuellement repère de thésaurus des informations élémentaires restantes.

.../...

. Unité d'informations "répétitive" :

On aura 2 suivi du nombre d'informations élémentaires dont l'ensemble forme l'unité d'informations. C'est cette unité d'informations qui sera "répétitive".

Quand un noeud de niveau 2 est "répétitif", ce que nous avons vu précédemment, il est forcément suivi de noeuds de niveau 3 sinon c'est le cas que nous étudions : à l'intérieur du noeud de niveau 2 l'unité d'informations se répète.

Comme précédemment si l'unité d'informations est "répétitive" il faut connaître le nombre de répétitions. Sur l'enregistrement on retrouve le type de carte et le type de M2. Derrière le type de M2 on aura le nombre de répétitions (cf "information de service"). La place occupée par ce nombre est donnée par "Nb de".

La description de l'unité d'informations sur la carte de structure se fait alors comme dans le cas de l'unité d'informations "unique". On a le nombre d'informations élémentaires, le nom de chaque information, sa taille, éventuellement un repère de thésaurus et des cartes "suite" s'il y a lieu.

- Sur cette carte de niveau 2 sont de longueur variable :

"type de M2" : sa longueur est donnée par "taille type de M2" de la carte O.

"nb de noeuds de niveau 3 dans M2" sa longueur est donnée par "taille nb de" de la carte O.

"Nb d'informations élémentaires" : sa longueur est donnée par "taille nb de".

32234 - Remarque :

- La taille d'une information élémentaire peut-être fixe ou variable (cf paragraphe 213).

. si elle est fixe sa taille effective est inscrite.

Dans le cas d'une information devant être codée c'est sa taille après codage qui est inscrite. L'inscription de la taille est assez libre. On pourra avoir 002 ou ØØ 2 (Ø = un blanc).

. si elle est variable on aura "V".

On ne connaitra donc sa taille qu'à la lecture de l'article. La fin de l'information élémentaire variable sera précisée par une * et avant de la ranger dans l'emplacement en mémoire qui sera écrit sur la bande, on mettra la taille juste avant l'information.

.../...

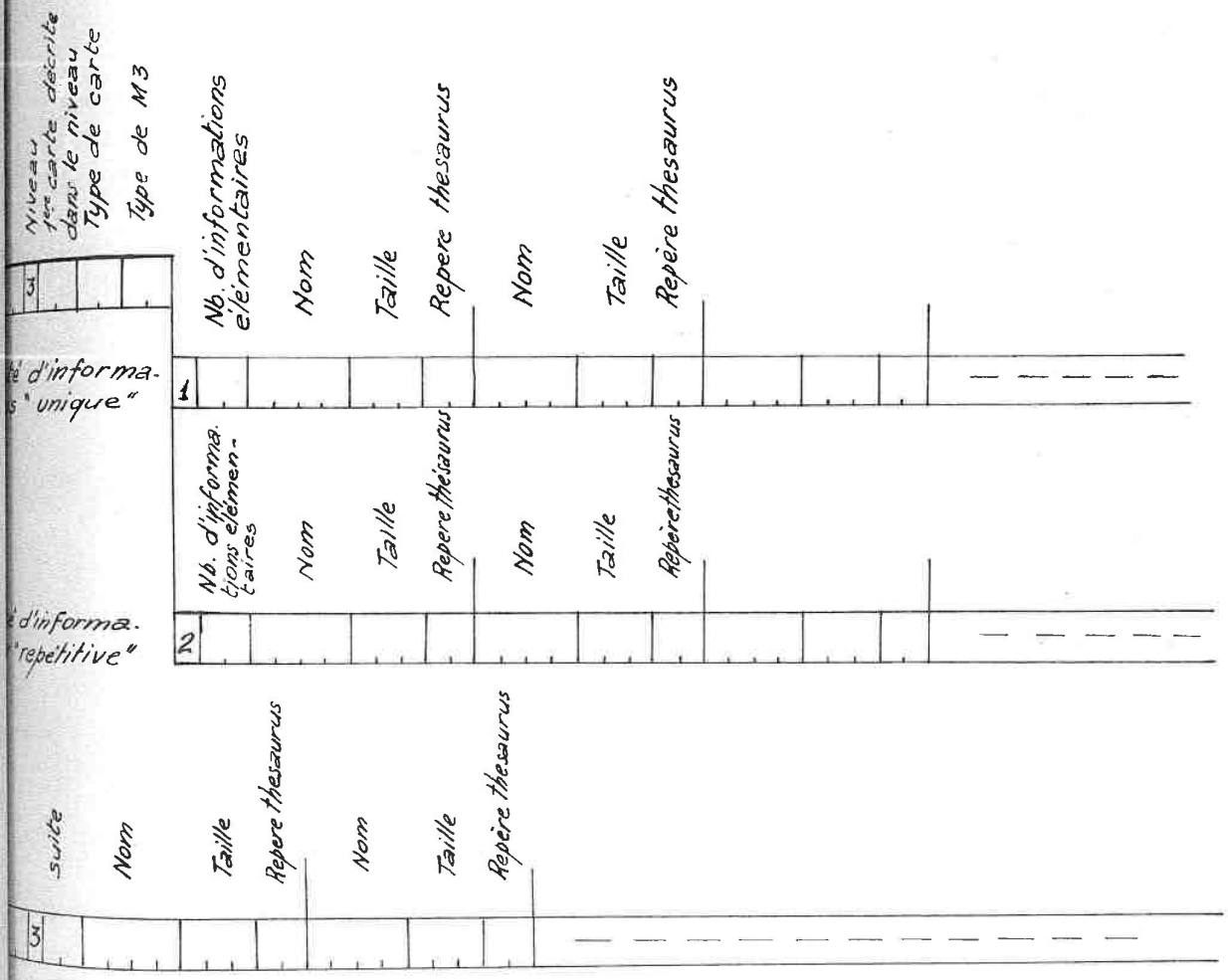
- Lorsque plusieurs cartes sont nécessaires pour décrire le niveau, on connaitra la fin de chaque carte lorsqu'on rencontrera le caractère ∅.
Dans le fichier structure les noeuds de niveau 2 seront ainsi décrits :

type M2	NOM	fixe ou répétitif	longueur propre	longueur / origine du noeud	longueur UI si répétitive	thésaurus
02	FIXE					
description des noeuds de niveau 3.						
05	FIXE					
	MØEU	F	31	0)	pas de noeuds de	
	PRØJ	F	36	31 (	niveau 3.	
11	FIXE					
	TPIE	R	2	0	49)	pas de
	NPSØ	R	3	2	49 (	noeuds
	LPIE	R	5	5	49)	de
12	FIXE				(	niveau
	ARIA	R	5	0	) V (	3
	TYIØ	R	V	5	V)	
95	REPE					
description des noeuds de niveau 3.						

On précisera l'organisation de cette structure en regroupant les trois niveaux plus loin.

.../...

3224 - Une ou plusieurs cartes de niveau 3 :



- Comme pour les deux niveaux précédents, on a d'abord une numérotation des cartes, le numéro du niveau, la première carte du niveau décrit, le type de carte, le type de M3.
On a vu que le noeud de niveau 1 et au moins un noeud de niveau 2 étaient obligatoirement présents. On pourrait avoir une structure qui se réduise à cela.
On pourrait également avoir une structure avec le noeud de niveau 1 et plusieurs noeuds de niveau 2. Il n'y aurait alors aucun noeud de niveau 3. Enfin, on peut avoir une structure avec le noeud de niveau 1 et des noeuds de niveau 2 et 3 entremêlés.
De même qu'au niveau 2, tous les noeuds de niveau 3 doivent être décrits dans la structure ; mais dans l'article ils pourront être présents,
absents,
ou répétés.

- La description des unités d'informations dans les noeuds de niveau 3 est la même qu'au niveau 2 lorsqu'il n'y a pas de noeuds de niveau 3.

. Unité d'informations "unique"

On aura 1 suivi du nombre d'informations élémentaires, suivi lui-même du nom, de la taille, éventuellement d'un repère de thésaurus pour chaque information élémentaire et de cartes "suite" s'il y a lieu.

. Unité d'informations "répétitive"

On aura 2 suivi du nombre d'informations élémentaires, suivi lui-même du nom, de la taille, éventuellement d'un repère de thésaurus pour chaque information élémentaire et de cartes "suite" s'il y a lieu.

La taille d'une information est fixe ou variable (codée "V"). Dans le cas d'une information fixe codée on a la taille après codage. La fin d'une carte qui doit être complétée par des cartes "suite" est notifiée par " ∅ " (cf niveau 2). Dans le cas d'une unité d'informations "répétitive", il faut préciser dans chaque article le nombre de répétitions correspondantes. Derrière le type de M3 on aura le nombre (cf "information de service"). La longueur de celui-ci est donnée par "nb de".

Dans le fichier structure les noeuds de niveau 3 sont ainsi représentés.

.../...

type de M3	Nom	fixe ou répé-titif	longueur propre	position / origine du noeud	longueur UI si répé-titive	thésaurus
03	ØRGA	F	6	0		
	NØMI	F	10	6		
	ANNI	F	4	16		
04	FIAB	R	1	0	22	
	FØR1	R	5	1	22	2
	FØR2	R	5	6	22	2
06	PLAL	R	V	0	V	
	SILI	R	5	V	V	
	ARGØ	R	6	V	V	

Nous allons étudier l'organisation du fichier structure.

323 - Représentation interne de la table structure : en mémoire centrale.

Nous l'avons déjà présentée dans le chapitre précédent. Nous allons la reprendre en donnant de plus amples détails.
Pour chaque niveau on a 7 mots :

- type de carte
- type de M1, M2, M3
- nom.
- fixe ou répétitif : 1 octet, longueur propre interne, 3 octets
- position par rapport à l'origine du noeud
- longueur de l'unité d'informations si elle est "répétitive"
- repère de thésaurus.

Cette représentation en tableau (figure 11) est une commodité graphique. En mémoire toutes les informations sont consécutives.

.../...

3231 - les trois premiers mots.

- type de carte : c'est le type de carte dans l'article sur lequel est décrit le niveau.

- type de M1, M2, M3 : à chaque changement de noeud on répète le type de carte même s'il ne change pas et à côté, on met le type de M1 ou de M2 ou de M3 correspondant.

- nom : c'est l'identificateur de l'information élémentaire : au niveau 2 s'il n'est pas suivi de noeuds de niveau 3 ou au niveau 3.
Une exception : sur la première ligne décrivant le noeud de niveau 2 on trouve "FIXE" ou "REPE". "FIXE" ou "REPE" sont une anomalie dans les identificateurs d'informations mais ils sont placés là uniquement pour faciliter le traitement.
"FIXE" signifie que le noeud de niveau 2 est "unique".
"REPE" qu'il est "répétitif". Dans ce dernier cas avant le type de M2, sur l'article, il faut mettre le nombre de répétitions.

3232 - les trois mots suivants :

- Ensemble répétitivité et taille :
  . Fixe ou répétitif sur un octet :
L'unicité ou la répétitivité se place au niveau de l'unité d'informations. Si celle-ci est "unique" sur l'octet, on aura "F" et cet "F" sera répété devant chaque information élémentaire. Si ell est "répétitive" sur l'octet on aura "R" et cet "R" sera répété devant chaque information élémentaire. On répéte "F" et "R" devant chaque information uniquement pour faciliter le traitement.
  . Longueur propre sur trois octets :
C'est la taille de l'information élémentaire.
Comme on l'a vu précedemment, on aura :
  ou la taille effective,
  ou la taille codée s'il y a codage,
  ou "V" si la taille est variable.
Cette taille sert dans la manipulation des informations (cf paragraphe 16) et elle sert à calculer les deux mots suivants.

- Position par rapport à l'origine du noeud :
C'est l'endroit où se trouve l'information élémentaire par rapport au début de l'unité d'informations.
Cette longueur peut-être inconnue dans le cas où avant l'information élémentaire étudiée, il y a eu information élémentaire de taille variable, on a alors "V". une

.../...

On a voulu préciser cette position pour faciliter l'exploitation si on connait le nom de l'information à traiter, on va le chercher dans la structure et ainsi sachant où se trouve l'adresse d'origine du noeud auquel appartient cette information, dans l'article (on verra dans le fichier-maître comment sont établies les adresses d'origine des noeuds) et où se trouve l'information (cas d'une unité d'informations "unique") par rapport au début de ce noeud, on l'atteindra très facilement (cf paragraphe 16).
Dans le cas d'une unité d'informations "répétitive" connaissant la longueur de l'unité, la position de l'information élémentaire par rapport au début de l'unité d'informations, le nombre d'unité d'informations et quelle est l'adresse d'origine du noeud dans l'article (pour ces deux derniers, on verra comment ils se présentent dans le fichier-maître) on atteindra assez rapidement l'information cherchée (cf paragraphe 16).

- Longueur unité d'informations si "répétitive".
On vient d'en voir l'utilité. Cela évite de refaire à chaque fois la somme de toutes les informations élémentaires pour passer à la suivante.
Cette longueur n'a d'intérêt que dans le cas où l'unité d'informations est "répétitive". En effet dans le cas où elle est "unique", on a directement la position de l'information élémentaire grâce à l'élément "position par rapport à l'origine du noeud".
Puis pour passer au noeud suivant, on connaît directement la position de ce dernier grâce à la table des noeuds.
Si dans l'unité d'informations "répétitive" on a une information élémentaire de taille variable la longueur de cette unité sera "V" et on sera obligé d'en faire le calcul à l'exploitation.

Remarque : comme on peut le constater il ne faut pas abuser des informations de taille variable. A l'exploitation elles ralentissent nettement le traitement puisqu'on est obligé de faire des calculs supplémentaires.
Dans la mesure du possible, il faudra en mettre assez peu à moins que la taille gagnée en ne prenant pas une taille fixe maximum ait plus d'intérêt que le temps perdu.

.../...

type de carte	type de M1 M2 M3	Nom	fixe ou répétitif	longueur propre interne	position / origine du noeud	longueur unité d'informations si répétitive	thésaurus
AA	AA	CLAS	F	9	0		
		ZLAM	F	1	9		
		XLAM	F	7	10		

02	02	FIXE					
02	03	ØRGA	F	6	0		
		NØMI	F	10	6		
		ANNI	F	4	16		

03	04	FIAB	R	1	0	22	
		FØR1	R	5	1	22	2
		FØR2	R	5	6	22	2

04	05	FIXE					
04	06	PLAL	R	V	0	V	
		SILI	R	5	V	V	
		ARGØ	R	6	V	V	

99	95	REPE					
99	96	PRØT	F	5	0		
98	97	TEXT	F	V	0		

Sur cette structure qui ne correspond pas tout à fait à celle du sondage carotte, on a tous les cas possibles. (figure 11)

3233 - Repère de thésaurus :

32331 - Présentation :

C'est le numéro du thésaurus qu'il faudra consulter pour cette information élémentaire.
Il y en a trois sortes : PRES qui sert uniquement à des vérifications d'erreur,
CØDE pour coder et éventuellement décoder l'information,
DECØ pour décoder l'information et qui sert uniquement à l'exploitation.
Ces trois sortes de thésaurus ont déjà été exposées plus longuement. Nous avons vu la représentation externe des thésaurus dans le paragraphe 32211.

32332 - Organisation interne des thésaurus : en mémoire centrale.

On a donné des numéros à ces thésaurus partant de 1 à 50 (c'est la limite qu'on s'est fixé) pour qu'ils soient atteints très rapidement non pas par une recherche séquentielle mais par le procédé suivant.
On a une zone appelée DIC. Dans le premier mot se trouve l'adresse du premier thésaurus, dans le deuxième mot celle du deuxième thésaurus et ainsi de suite. L'adresse est le nombre d'octets qu'il faut ajouter à DIC pour atteindre le thésaurus.

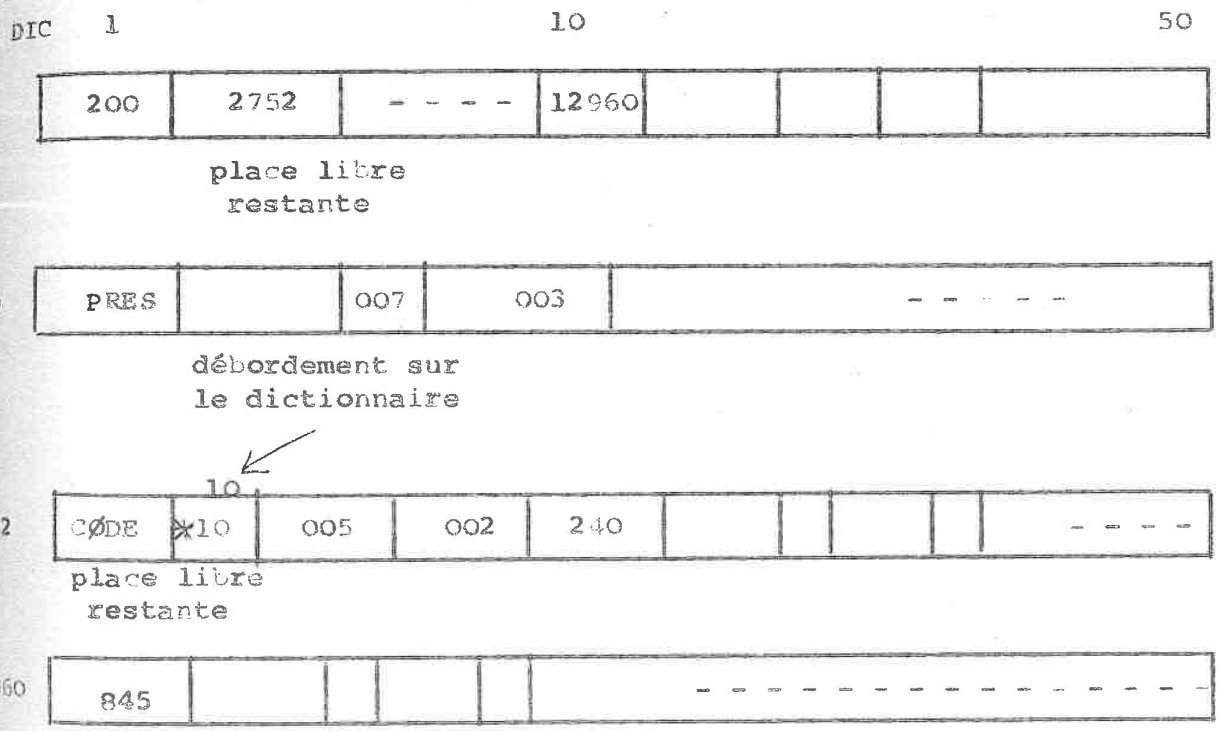
DIC

200	1476	2752		- - - - -

Aussi dès qu'on aura décodé le numéro du thésaurus on atteindra celui que l'on veut.
Sur bande on utilisera deux blocs pour ranger ces dictionnaires qui forment le fichier BIBLIØ. Dans la version actuelle on a fixé la longueur de ces deux blocs bien que la gestion des dictionnaires à l'intérieur de ces deux blocs soit très souple et permette de tasser les informations au maximum. Dans une version ultérieure on utilisera des paramètres pour réduire la taille de ces blocs ou même les supprimer s'il n'y a pas de dictionnaires.

.../...

En mémoire la zone où sont construits ces blocs s'appelle DIC. Elle est de 16000 mots.
50 dictionnaires sont possibles, chacun peut occuper 319 mots maximum.
Au début de la zone DIC, on réserve 50 mots pour mettre les adresses des différents dictionnaires. On a vu le procédé pour atteindre un dictionnaire. Il est possible que les utilisateurs aient de nombreux dictionnaires de taille moyenne ou peu de très grande taille. Si l'on avait réservé pour chacun une taille maximum cela aurait fait beaucoup de place perdue, on a donc réservé pour chacun une taille moyenne en prévoyant des débordements.
DIC se présente de la manière suivante :



Les 50 premiers mots représentent les adresses où se trouvent les 50 dictionnaires puis viennent ces derniers.
On a le nom du dictionnaire, la place libre restante puis
. si le nom est PRES : la taille externe de l'information, le nombre d'informations.
. si le nom est CØDE ou DECØ : la taille externe de l'information, sa taille interne et le nombre d'informations.
On connait la place réservée pour un dictionnaire, la taille occupée par les informations :

.../..

Pour PRES : (nb d'informations x taille information)
pour CØDE et DECØ : [nb d'informations x (taille externe +,taille interne de l'information)]

On peut donc calculer la place libre restante.
Si cette taille devient nulle puis négative, c'est qu'elle est insuffisante ; on va donc chercher dans les 50 premiers mots s'il y a un numéro de dictionnaire non encore attribué. Sinon il faudra entièrement réorganiser la zone DIC mais ce cas a très peu de chances de se produire. Si oui on mettra dans "place libre restante" sur le premier octet "*" puis le nouveau numéro de dictionnaire attribué. Ainsi le premier dictionnaire continuera sur l'emplacement libre de ce dernier et ainsi de suite. Sur le premier mot du nouvel emplacement on aura la "place libre restante" ou "*" et un nouveau numéro de dictionnaire.
Quand un nouveau dictionnaire est ainsi attribué, on aura un message sur l'imprimante donnant le ou les nouveaux numéros de dictionnaires attribués. Aucun dictionnaire devant être mis en mémoire ne pourra porter un de ces numéros.
Cette possibilité d'extension des dictionnaires est faite dans un programme de mise à jour des dictionnaires.

A l'entrée en mémoire d'un dictionnaire il faudra faire en sorte que sa taille ne dépasse pas la zone réservée pour un dictionnaire. En effet, si elle la dépasse, on lui réservera une nouvelle zone et cette nouvelle zone correspondra peut-être au numéro du dictionnaire suivant : celui-ci n'étant pas encore entré en mémoire. Il y aurait alors à l'entrée de ce dernier une erreur et il serait éjecté. C'est pour éviter ces retards que l'on utilise le procédé précédemment décrit.
Il n'est pas difficile si le contenu du dictionnaire doit être très important d'en porter une partie en mise à jour.

32333 - Remarque :

Pour ne pas mobiliser trop de dérouleurs de bandes, le fichier BIBLIØ se trouve sur la même bande que les fichiers structures. Si plusieurs fichiers structures sont entrés en mémoire les uns à la suite des autres, on ne rangera BIBLIØ qu'après le rangement du dernier fichier structure.
En effet, sur une bande on peut lire mais si on veut modifier un fichier, tous les fichiers qui sont derrière celui qu'on vient de modifier sont perdus.
Le seul fichier qui sera fréquemment mis à jour est le fichier BIBLIØ. Il doit donc toujours être le dernier fichier sur la bande.
Si l'on a de nouvelles structures à ranger, deux cas peuvent se présenter :

.../...

. le fichier BIBLIØ n'existe pas sur la bande, on va donc ranger les nouveaux fichiers à la suite des précédents.

. le fichier BIBLIØ existe sur la bande, on va donc le faire venir en mémoire centrale.

S'il n'y a pas de nouveaux dictionnaires à mettre dans BIBLIØ on rangera les nouveaux fichiers à la suite des précédents, le premier se trouvant à la place de BIBLIØ, puis on réécrira BIBLIØ à la suite de ces fichiers.

S'il y a de nouveaux dictionnaires à mettre dans BIBLIØ, on mettra BIBLIØ à jour au fur et à mesure où les dictionnaires apparaitront. On rangera les nouveaux fichiers à la suite des précédents, le premier se trouvant à la place de BIBLIØ, puis on réécrira BIBLIØ modifié à la suite de ces fichiers.

Ainsi BIBLIØ sera toujours le dernier fichier sur la bande.

Dans l'avant dernier cas, BIBLIØ n'a pas été modifié, on a donc fait une manipulation inutile. On avait pensé à un autre système que voici :

Si on a de nouvelles structures à entrer, deux cas peuvent se présenter :

. les structures à entrer ne comportent pas de nouveaux dictionnaires : elles seront rangées derrière le fichier BIBLIØ

. les structures à entrer comportent de nouveaux dictionnaires. On met le fichier BIBLIØ en mémoire puis on prend une bande de travail pour ranger tous les fichiers structures les uns après les autres en supprimant BIBLIØ. On recopie cette bande sur l'ancienne et on met le fichier BIBLIØ, modifié en queue.

Le deuxième système nous est apparu beaucoup plus lourd que le premier.

Le travail de copie sur bande et de recopie était fastidieux, de plus il fallait un dérouleur de bandes supplémentaire.

Or pour les applications que nous avons en vue, les mises à jour de dictionnaires seront fréquentes, celles de fichiers structures très rares, la manipulation inutile du premier système sera par conséquent assez rare.

3234 - Taille du fichier structure :

On a un programme principal,

un sous-programme comportant toutes les réservations de zones, en particulier celle de STRUCT.

Si au cours du traitement la taille réservée pour STRUCT s'avère insuffisante, on remplace le sous-programme par un autre dans lequel la zone réservée pour STRUCT est plus grande. On peut pour la suite garder ce dernier sous-programme jusqu'à ce qu'il y ait un nouveau changement.

.../...

Conclusion : L'organisation d'un fichier structure peut paraître un peu lourde puisque pour chaque information élémentaire, on a besoin de sept mots mais elle a été conçue ainsi de façon à faciliter l'exploitation.
Ce travail en effet n'est réalisé qu'une fois et évite des calculs lourds qui risqueraient de se retrouver très souvent à l'exploitation. On peut dire que la complexité des structures rend nécessaire le contrôle, mais elle le facilite aussi puisqu'on a de nombreuses informations à contrôler.

33 - L'enregistrement :

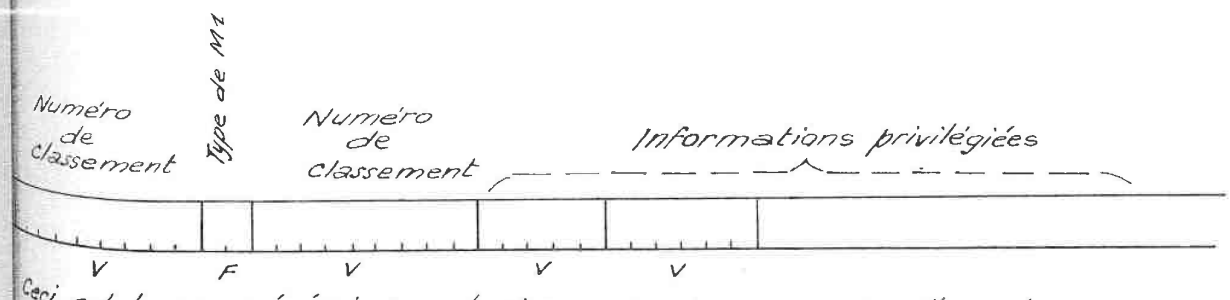
Il est formé de la table des noeuds et de l'article.

331 - Généralités :

On a une suite de cartes à examiner en fonction de la table structure pour former l'enregistrement. Certains éléments de ces cartes ne servent qu'à des vérifications, ils sont supprimés avant le rangement en mémoire.
Certaines informations élémentaires peuvent être modifiées : en cas de codage ou en cas de taille variable. Dans ce dernier cas on place devant l'information sa longueur et l'"*" indiquant la fin de l'information disparait.
Quand il s'agit d'unités d'informations "répétitives", le nombre de répétitions n'est plus devant la première unité, mais dans une table, dont nous avons déjà parlé mais que nous préciserons plus loin, à côté du noeud correspondant.
Sur la figure 12 nous donnons quelques exemples de cartes. Nous rappelerons ces exemples dans la suite par leur numéro : exemple 1 - 2 .....

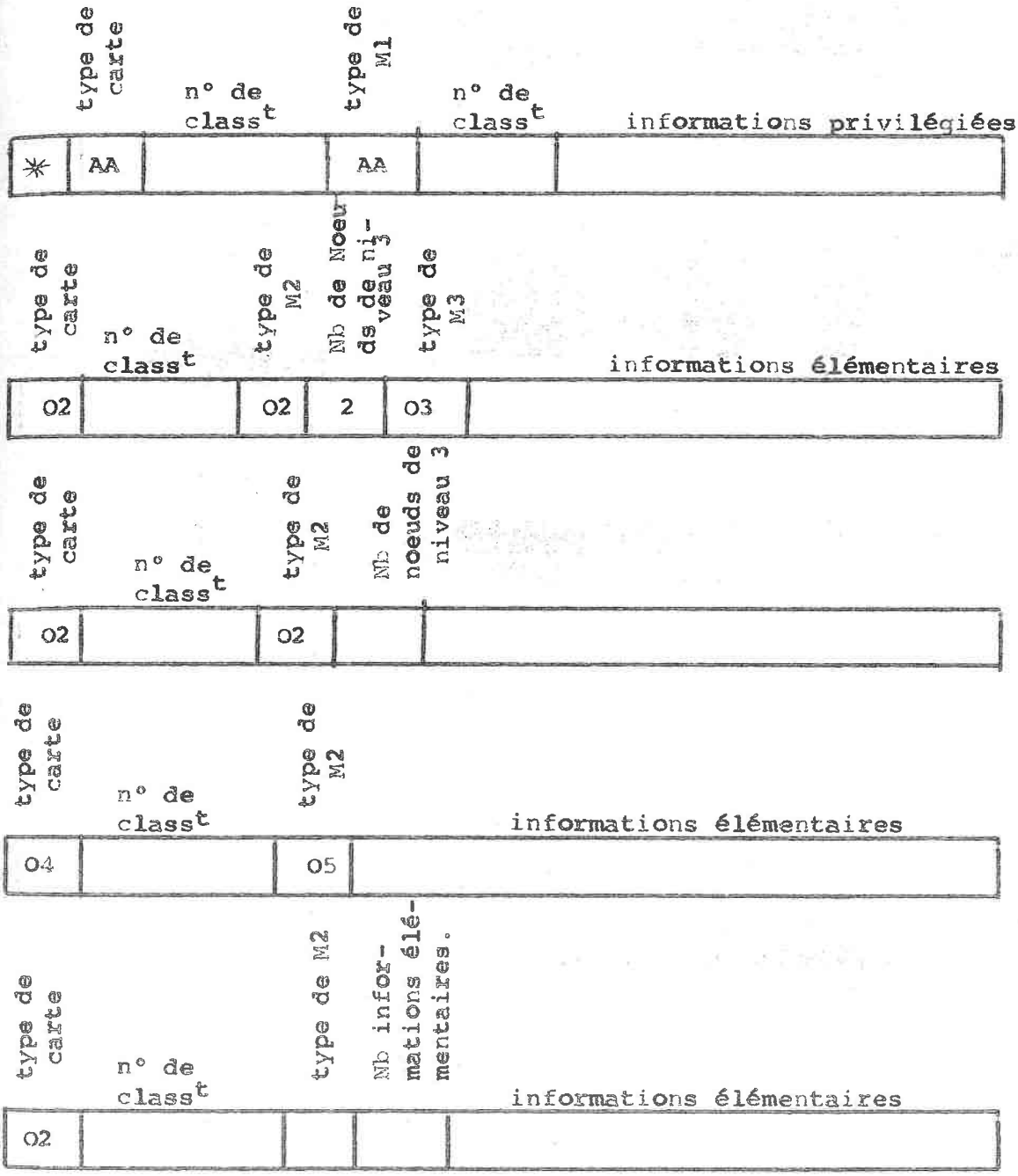
332 - Description :

3321 - Niveau 1 :



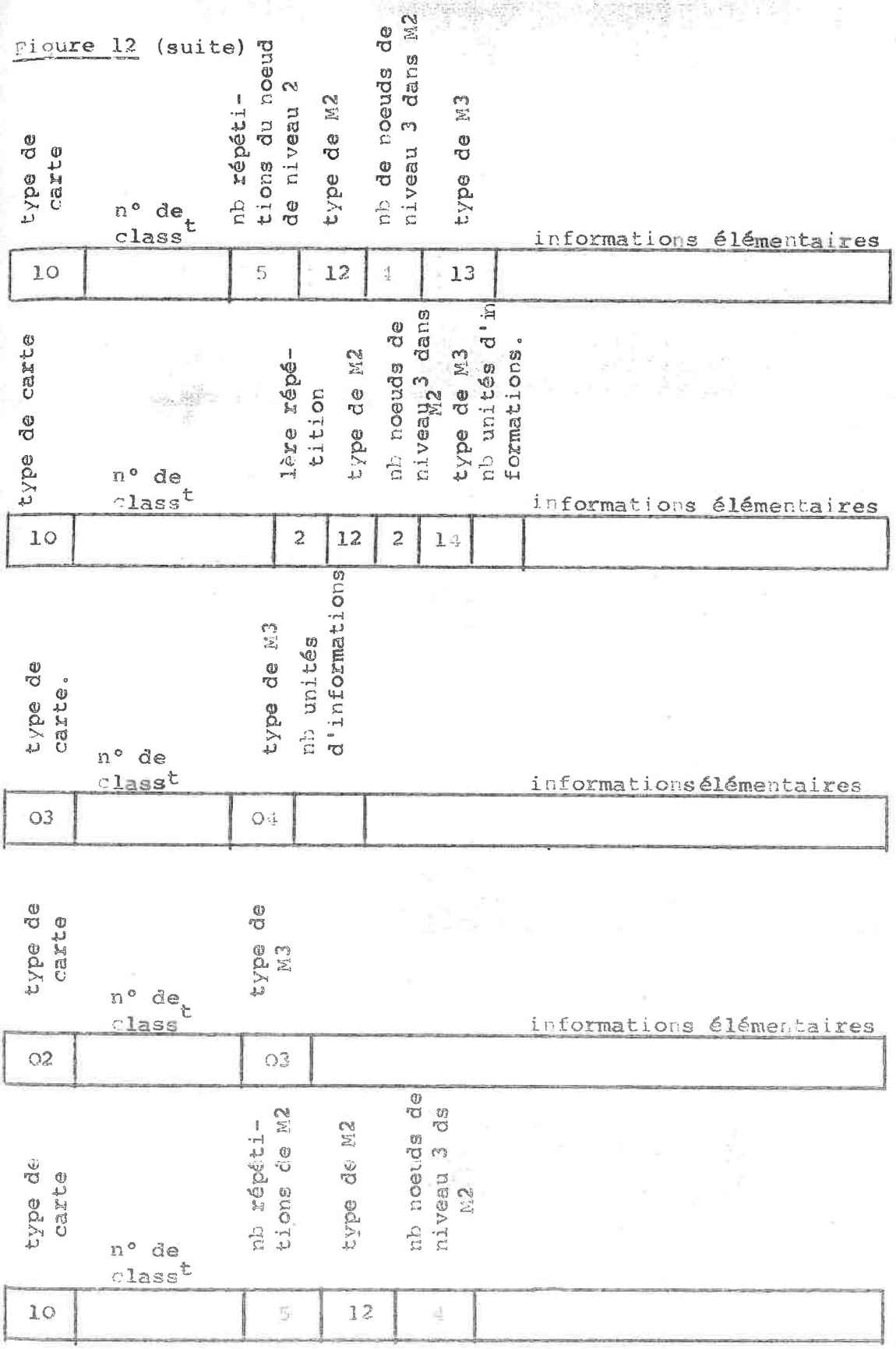
Ceci est le cas général pour le niveau 1. Il correspond à l'exemple 1 ...

Figure 12 -


type de carte
n° de classt
type de M1
n° de classt
informations privilégiées
1
*
AA
AA
type de carte
n° de classt
type de M2
Nb de Noeuds de niveau 3
type de M3
informations élémentaires
2
02
02
2
03
type de carte
n° de classt
type de M2
Nb de noeuds de niveau 3
3
02
02
type de carte
n° de classt
type de M2
informations élémentaires
4
04
05
type de carte
n° de classt
type de M2
Nb informations élémentaires.
informations élémentaires
5
02

.../...

figure 12 (suite)

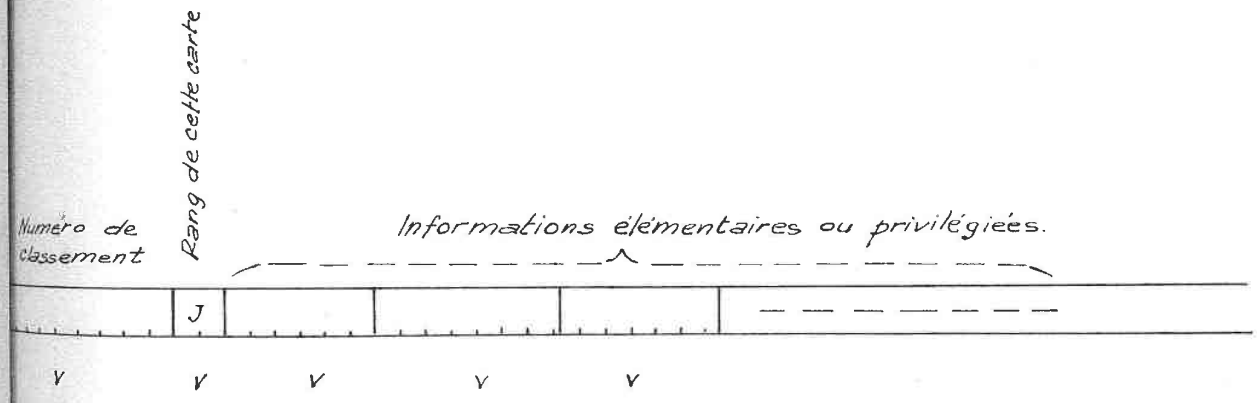


.../...

(suite)

	type de carte	n° de classt	type de M2	nb de noeuds de niveau 3	type de M3	nb de répétitions	informations élémentaires
11	02		02	2	06		

Observations : valables pour les trois niveaux M1-M2-M3.
M1-M2-M3 : signifient respectivement noeuds de niveau 1,2 ou 3.
F, V : signifient respectivement format imposé par le programme ou format défini dans le fichier structure.
Le signe "*" est obligatoire pour terminer une information élémentaire de taille variable.
Carte "suite" : elle est la même pour les trois niveaux.



La carte précédant une carte "suite" est obligatoirement terminée par le signe "$".

- La carte de niveau 1 commence par une "*".
Elle sert à séparer deux articles.
Pour repérer le passage d'une structure à l'autre, on connaissait le nombre de noeuds de niveau 2 de chacune. Pour les articles on ne peut connaître à l'avance le nombre de noeuds de niveau 2 : en effet celui-ci varie d'un article à l'autre.

.../...

Dans le cas du Laboratoire de l'Equipement de NANCY, les bordereaux sont remplis par des personnes ne connaissant pas le système, c'est-à-dire ignorant l'existence des différents niveaux. Les bordereaux sont pré-imprimés et ils ne savent pas quels noeuds correspondent au niveau 2 et lesquels au niveau 3. Il ne leur est donc pas possible d'indiquer le nombre de noeuds de niveau 2 présents dans chaque article. C'est pourquoi on a choisi la solution de faire commencer l'article par une "*".

- Comme nous l'avons vu dans les cartes de structure, le type de carte pour le niveau 1 est un peu particulier : il est le même que le type de M1. Il représente le label qui permet de retrouver le fichier structure correspondant à l'article traité.
Le fichier-structure une fois trouvé sur bande est mis en mémoire centrale. Il contient outre la description de la structure, quelques renseignements supplémentaires qui appartenaient à la carte 0 (cf la représentation externe du fichier structure : paragraphe 3221) et la taille de la structure.

- Comme pour le fichier structure on aura pour le fichier maître un programme principal et un sous-programme avec les réservations (celles qui nous intéressent sont alors la longueur de la table, la taille de l'enregistrement et la taille de la structure).
Si la taille de la structure (cas où elle était déjà insuffisante à la mise sur bande) ou la taille de la table ou la taille de l'enregistrement sont insuffisantes on remplace le sous-programme par un autre ayant des réservations plus grandes pour chacune de ces zones. On peut par la suite garder ce dernier sous-programme jusqu'à ce qu'il y ait un nouveau changement.

- Parmi les renseignements du fichier structure correspondant à l'ancienne carte 0, on a la taille du numéro d'article.
Ce numéro d'article ou numéro de classement est celui que l'on retrouve sur la carte 1 après le type de carte. Il sert uniquement à faire une vérification : c'est un test d'erreur. En effet à la lecture de chaque nouvelle carte, on vérifiera si ce numéro est toujours le même. Il caractérise l'article traité. Si ce n'est pas le même, il y a erreur : une carte appartenant à un autre article a pu être introduite par mégarde dans celui-ci.
Ce numéro de classement ne se retrouve qu'une fois dans l'article à la fin du traitement.

.../...

- type de M1 : nous avons vu pourquoi dans le fichier structure, type de M1 avait une taille fixe qui est celle du type de carte.
- puis on trouve les informations privilégiées qui, s'il le faut, se prolongent sur des cartes "suite".
Pour chaque carte "suite" on vérifie si le type de carte, le numéro de classement sont toujours les mêmes et si pour "suite" J = J + 1.
- Pour cette carte sont de taille variable :
numéro de classement : taille de l'ancienne carte O
les informations privilégiées : taille définie dans la structure.

3322 - Niveau 2 :

Un seul type de M2 et éventuellement un seul type de M3 (non nécessairement rempli même s'il a été prévu) sont présents par carte.
Nous avons représenté le cas général pour le niveau 2.
A côté de chaque cas nous avons mis le numéro de l'exemple correspondant (cf figure 12).
Après le niveau 1, vient le niveau 2 :

- type de carte est celui qui se trouve décrit dans le fichier structure à côté du n oeud en cours de traitement.
- Numéro de classement : même fonction que précédemment.

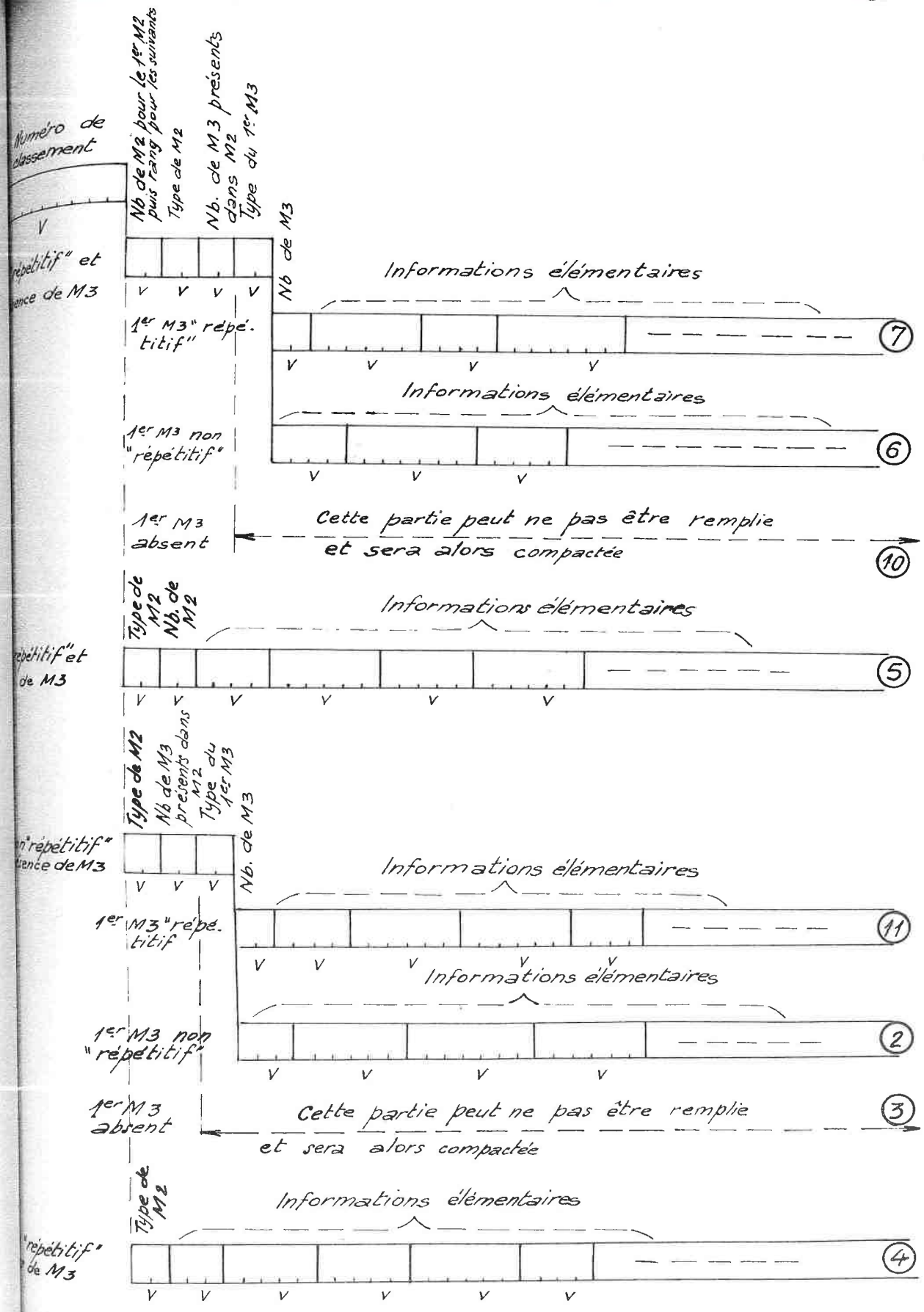
33221 - Noeud de niveau 2 "répétitif" et présence de noeuds de niveau 3 :

- Si le noeud de niveau 2 est "répétitif" avant type de M2, on trouve :
ou le nombre n de fois où ce noeud est présent dans l'article s'il s'agit du premier noeud,
ou le rang i de ces noeuds de niveau 2 pour les suivants. Pour le deuxième on aura i = 2 puis i = 3, 4 ... jusqu'à ce que i soit égal à n.

- Puis on a type de M2. C'est l'identificateur du noeud décrit dans la table structure à côté du type de carte.

- type de M2 est suivi du nombre de noeuds de niveau 3 présents dans ce noeud de niveau 2 et du premier noeud de niveau 3.
Ce dernier peut-être absent si pour la logique de la structure, sa présence n'est pas obligatoire (cas 10). On passe alors à la carte suivante pour chercher le noeud de niveau 3 suivant (cas 8 et 9 que nous étudierons plus loin).

.../.


Numéro de classement
Nb de M2 pour le 1er M2 puis rang pour les suivants
Type de M2
Nb. de M3 présents dans M2
Type du 1er M3
Nb de M3
répétitif" et
ence de M3
Informations élémentaires
1er M3" répé. titif"
7
Informations élémentaires
1er M3 non "répétitif"
6
1er M3 absent
Cette partie peut ne pas être remplie
et sera alors compactée
10
Type de M2
Nb. de M2
répétitif" et
de M3
Informations élémentaires
5
Type de M2
Nb de M3 présents dans M2
Type du 1er M3
Nb. de M3
n" répétitif"
ence de M3
Informations élémentaires
1er M3 "répé. titif
11
Informations élémentaires
1er M3 non "répétitif"
2
1er M3 absent
Cette partie peut ne pas être remplie
et sera alors compactée
3
Type de M2
Informations élémentaires
"répétitif"
de M3
4

- Si le premier noeud de niveau 3 est présent, deux cas peuvent se présenter :
  . type de M3 est "répétitif". On a alors le nombre d'unités d'informations suivi lui-même de ces unités (cas 7).
  . type de M3 n'est pas "répétitif". il est alors suivi de l'unité d'informations "unique" (cas 6).

33222 - Noeud de niveau 2 "répétitif" et absence de noeuds de niveau 3 :

Si le noeud de niveau 2 ne comprend pas de noeuds de niveau 3, c'est que les informations élémentaires sont au niveau 2.
On a alors type de M2 suivi du nombre d'unités d'informations suivi lui-même des unités d'informations (cas 5).

33223 - Noeud de niveau 2 non "répétitif" et présence de noeuds de niveau 3 :

On a :
- type de M2,
- puis le nombre de noeuds de niveau 3 présents dans ce noeud de niveau 2,
- puis le type du premier noeud de niveau 3.
Ce dernier comme dans le paragraphe 33221 peut-être absent (cas 3).
- si le premier noeud de niveau 3 est présent ; deux cas peuvent se présenter :
  . type de M3 est "répétitif" : on a alors le nombre d'unités d'informations suivi lui-même de ces unités (cas 11)
  . type de M3 n'est pas "répétitif": il est alors suivi de l'unité d'informations "unique" (cas 2).

33224 - Noeud de niveau 2 non "répétitif" et absence de noeuds de niveau 3 :

Si le noeud de niveau 2 ne comprend pas de noeuds de niveau 3 c'est que les informations élémentaires sont au niveau 2.
On a alors type de M2 suivi de l'unité d'informations "unique" (cas 4).
Si pour tous ces cas, des cartes "suite" sont nécessaires, elles auront le même format que celle qui a été décrite au niveau 1.

33225 - Eléments de taille variable :

Pour tous ces cas, sont de taille variable :
. numéro de classement : "taille numéro d'article" de l'ancienne carte O,
. type de M2 : "taille type de M2" de la carte O,
. type de M3 : "taille type de M3" de la carte O,

.../...

. nombre de noeuds de niveau 2, ou de niveau 3 taille "nb de" de la carte O,
. nombre d'unités d'informations : taille "nb de".
. informations élémentaires : taille définie dans la table structure.

Remarque : Tous les noeuds de niveau 2 possibles sont décrits dans la table structure. On ne les retrouve pas obligatoirement tous dans l'article. Si dansl'article certains noeuds sont absents, ils seront automatiquement absents dans la table des noeuds. Pour savoir si l'absence d'un noeud est logique ou non, on pourra à la fin du traitement faire une vérification particulière.

3323 - Noeud de niveau 3 :

Un seul type de N3 par carte. (voir croquis page suivante).

Après le noeud de niveau 2 viennent éventuellement des noeuds de niveau 3.
On en a déjà parlé dans le niveau 2 pour le premier noeud de niveau 3. Etudions les autres cas.
Comme le noeud de niveau 2, le noeud de niveau 3 peut-être absent et cela peut donner lieu à des vérifications.
S'il est présent on a :

- le type de carte,
- le numéro de classement,
- le type de N3 : c'est l'identificateur du noeud décrit dans la table structure à côté du type de carte.

Puis deux cas peuvent se présenter :

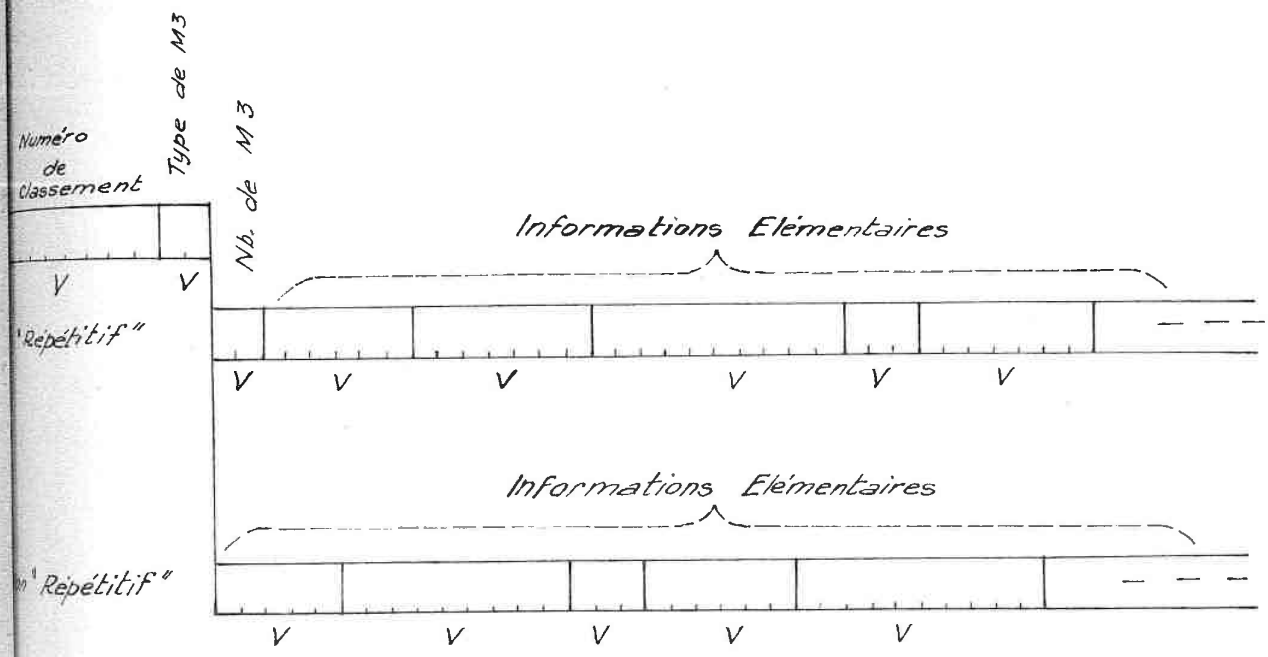
. l'unité d'informations est "unique" (cas 9). On a alors les informations élémentaires de cette unité.

. l'unité d'informations est "répétitive" (cas 8). On a alors le nombre d'unités d'informations suivi de ces unités d'informations.

-Si des cartes "suite" sont nécessaires, elles auront le même format que celle qui a été décrite au niveau 1.
-Pour les cartes correspondant au niveau 3, sont de taille variable :

. numéro de classement : "taille numéro d'article" de la carte O,

.../...


Numéro de Classement
Type de M3
Nb. de M3
Informations Elémentaires
V
V
"Répétitif"
V V V V V V
Informations Elémentaires
"Répétitif"
V V V V V

. type de M3 : "taille type de M3" de la carte O,
. nb d'unités d'informations : "taille nb de" de la carte O,
. informations élémentaires : taille définie dans la table structure.

On vient de voir comment sont établies les liaisons entre le fichier structure et les cartes formant l'article pour établir le fichier maftre.

On a parlé à certains moments d'une table TAB spéciale contenant les différents noeuds et qui permet de retrouver une information élémentaire ou un groupe d'informations élémentaires Comment se présente cette table ? Où se trouve-t-elle ?

333 - Représentation interne de la table des noeuds :

3331 - Description :

D'abord où se trouve-t-elle ?
Elle est construite dans une zone de la mémoire centrale et sera mise sur la bande contenant le fichier maftre juste avant l'article de façon à ce que cette table et l'article ne forment qu'un seul enregistrement.
Sur la figure 13 : nous avons une description de table. Dans cette table nous avons voulu présenter tous les cas possibles. Ils ne correspondent pas tous à la structure que l'on avait donnéed'un sondage carotté au début de cet exposé.

3332 - Les deux premiers mots :

La figure 13 représente la table telle qu'elle est construite pendant le traitement de l'enregistrement. Les deux premiers mots jouent un rôle particulier par rapport au reste de la table.
- Le premier comporte la taille de l'enregistrement qui peut permettre de passer directement d'un enregistrement à un autre en sautant cette taille.
Cette taille est celle de l'enregistrement proprement dit à laquelle on a ajouté la taille de la table.
- le deuxième comporte la taille de la table qui permet d'arriver directement au début de l'article.

.../...

3333 - Le reste :

Nous avons ensuite une succession de groupes de trois mots donnant chacun des renseignements sur le noeud qu'il représente.

33331 - Le premier mot :

On y trouve l'identificateur de noeud présent dans l'article, il doit être identique à l'identificateur correspondant de la table structure comme on a pu le constater.
Cet identificateur pour le niveau 1 est le label du fichier structure, mais pour les niveaux 2 ou 3, il est formé aussi bien de lettres que de chiffres.
On a vu que type de M1 était fixe et que type de M2 et type de M3 étaient de taille variable.
Cette taille ne doit pas dépasser 4 caractères ou octets. Quatre octets représentent un mot et dans cette table des noeuds on a prévu un mot pour ranger l'identificateur de noeud.
Peut-être peut-on trouver cela limité, mais cela représente, étant donné que l'on peut utiliser les lettres et les chiffres,
C_4^{35} : 52360 possibilités.

Ceci nous a paru suffisant pour les cas étudiés. En effet, si les types pouvaient avoir une taille indéfinie, il aurait fallu prévoir une table ayant une taille variable pour l'identificateur et pour le reste. Il aurait donc fallu de nombreux renseignements pour explorer cette table ce qui l'aurait alourdie et ce qui aurait prolongé énormément n'importe quelle recherche. Ce serait absurde et n'aurait aucun intérêt.
On a voulu donner le plus de renseignements possibles en utilisant le moins de place possible et en optimisant le temps de recherche d'une information élémentaire.

33332 - Le deuxième mot :

"Où commence-t-il?"
On a appelé le début de l'enregistrement : MEM.
Ce deuxième mot contient le nombre d'octets qu'il faut ajouter à MEM pour atteindre le noeud.

33333 - Le troisième mot :

"Multiplicité du niveau".
Suivant qu'il s'agit du niveau 1, 2 ou 3, la contenance de ce troisième mot n'a pas la même signification.

- pour le niveau 1 : on a toujours 0
- pour le niveau 2 : on a vu que si un noeud de niveau 2 était absent de l'article, on ne le retrouvait pas dans la table.

.../..

taille enregistrement	taille table

identificateur niveau 1,2 ou 3	où commence-t-il	multiplicité du niveau
AA	~	0
02	~	0
03	~	0
04	~	n
05	~	0
06	~	n
	~	0
12	~	n
13	~	0
14	~	0
15	~	0
ES	~	n
PI	~	0
12	~	n - 1
13	~	0
14	~	0
15	~	0
EV	~	0
GR	~	0

(figure 13)

.../...

Etudions donc le cas des noeuds présents :

. Si ce noeud de niveau 2 comprend des noeuds de niveau 3 et est "répétitif", ce troisième mot comprend le nombre de fois où ce noeud de niveau 2 est présent dans l'article, pour le noeud de niveau 2 d'indice 1, (ligne 40 de la table).
Pour ce noeud de niveau 2 d'indice 2 dans le troisième mot, on a le nombre de noeuds de niveau 2 du même type qui restent encore derrière lui - (ligne 57 de la table).
Cela est utile : en effet, si l'on recherche une information appartenant au noeud 12 et si l'on ne sait pas qu'il est "répétitif", la même information appartenant aux autres noeuds 12 ne sera pas exploitée.
Si l'on veut un certain noeud de niveau 3 appartenant au noeud 12 d'indice 1, on le recherchera par exploration séquentielle dans la table. On s'arrêtera au noeud 12 d'indice 2. Si on l'a trouvé, c'est qu'il est présent, sinon c'est qu'il est absent.
On aurait pu mettre dans la table tous les noeuds de niveau 3 possibles dans un noeud de niveau 2 et indiquer leur longueur. Si cette dernière était égale à 0 c'est que le noeud correspondant était absent. Ou encore on aurait pu comparer le deuxième mot "où commence-t-il" de chacun et en cas d'égalité, en conclure une absence. Seulement dans certains cas le nombre de noeuds de niveau 3 absents peut-être très important et la table serait alourdie par tous ces noeuds sans intérêt. on a décidé de choisir le test sur la présence ou l'absence.
Exemple : en géologie, ce noeud 12 correspond à une couche, on peut faire jusqu'à 44 essais. Sur ces essais aucun peut-n'être présent ou seulement un ou deux. on voit la place occupée par tous ces essais possibles et absents dans la table. On a oublié de préciser que chaque essai correspond à un noeud de niveau 3.

Conclusion : On aura donc dans la table pour un noeud de niveau 2, dans le troisième mot, le nombre de répétitions et derrière ce noeud de niveau 2 les noeuds de niveau 3 lui appartenant et effectivement présents dans l'article.

. Si ce noeud de niveau 2 ne comprend pas de noeuds de niveau 3 :

Deux cas peuvent se présenter :
ou l'unité d'informations élémentaires est "unique", dans ce cas on aura 0 dans le troisième mot (ligne 14)
ou cette unité est "répétitive", dans ce cas on aura le nombre de répétitions dans le troisième mot (ligne 17)

. Si ce noeud de niveau 2 comprend des noeuds de niveau 3 et n'est pas "répétitif" :

.../...

On aura 0 dans le troisième mot (ligne 5).
On a vu dans le paragraphe 15 comment étaient utilisés tous ces renseignements pour obtenir une information élémentaire ou un groupe d'informations élémentaires.

- Pour le niveau 3 :
On a vu que si un noeud de niveau 3 était absent de l'article on ne le retrouvait pas dans la table.
Etudions le cas des noeuds présents.
Deux cas peuvent se présenter :
ou l'unité d'informations élémentaires est "unique", on aura 0 dans le troisième mot (ligne 8)
ou elle est "répétitive", on aura alors le nombre de répétitions dans ce troisième mot (ligne 11).Cette représentation est celle d'un noeud de niveau 2 lorsqu'il ne contient pas de noeuds de niveau 3.

Remarque : Quand on a fini d'établir la table, on en connait sa longueur et on la place dans le deuxième mot (cf paragraphe 3332) "longueur de la table" puis on place cette table devant l'article lui-même. Pour accéder à la première information de l'article, il faudra donc sauter la longueur de la table. Pour passer à l'enregistrement suivant, il faudra sauter la longueur de la table et la longueur de l'article.
Il a donc paru plus commode au lieu de faire le calcul à chaque consultation de la table "où commence le noeud + longueur de la table" de l'établir une fois pour toutes. Avant de mettre la table suivie de l'article sur bande, on remplacera dans la table : taille de l'article par la somme taille de l'article + taille de la table, "où commence-t-il" pour chaque noeud par la somme "où commence-t-il" + taille de la table.
Ainsi, comme on a pu le voir, on aura effectivement dans le deuxième mot de la ligne, l'endroit exact où débute le noeud dans l'enregistrement.

3334 - Résumé :

On a essayé avec cette table de faciliter l'exploitation. On a essayé d'établir un juste milieu entre mettre le plus de renseignements possibles et économiser le plus de place possible.
Grâce à la table des noeuds et à la table structure, on a pu constater que l'extraction d'une information demandait peu de calculs, sauf dans le cas de tailles variables.

.../...

34 - Conclusion :

Il apparaît pendant la description de l'analyse organique que le format des cartes aussi bien pour construire le fichier structure que l'enregistrement est assez complexe.
L'utilisateur a donc deux possibilités :
ou écrire lui-même ses cartes
ou en donnant à l'informaticien un plan détaillé du fichier qu'il veut construire, le faire faire par ce dernier. L'informaticien pourra une fois le format des cartes fixé, établir des bordereaux pré-imprimés. Il sera alors possible de les faire remplir par des personnes ignorant l'informatique.
L'intérêt de confier le travail à un informaticien c'est qu'il est le mieux placé pour tirer parti de toutes les possibilités du programme.

4[1] | Fichier sructure

4 - Dossier de programmation :

41 - Fichier structure :

411 - Présentation des éléments utiles pour former le fichier structure :

4111 - Schéma de travail :

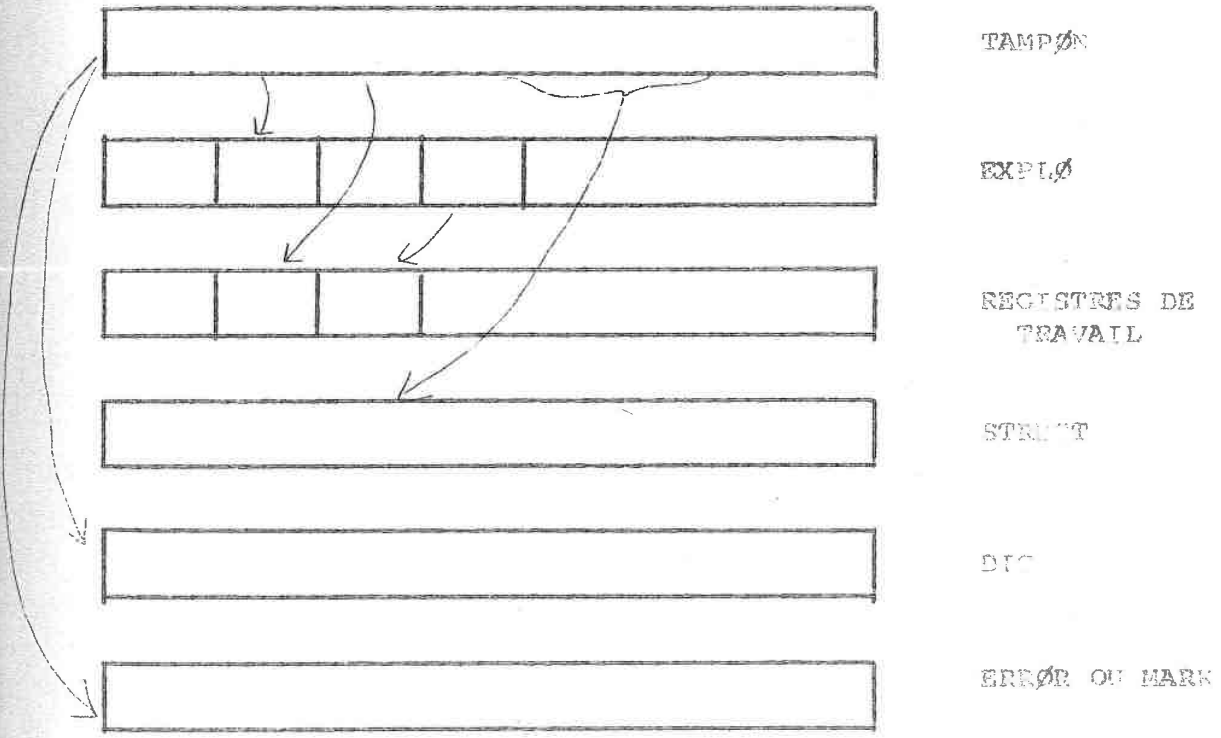


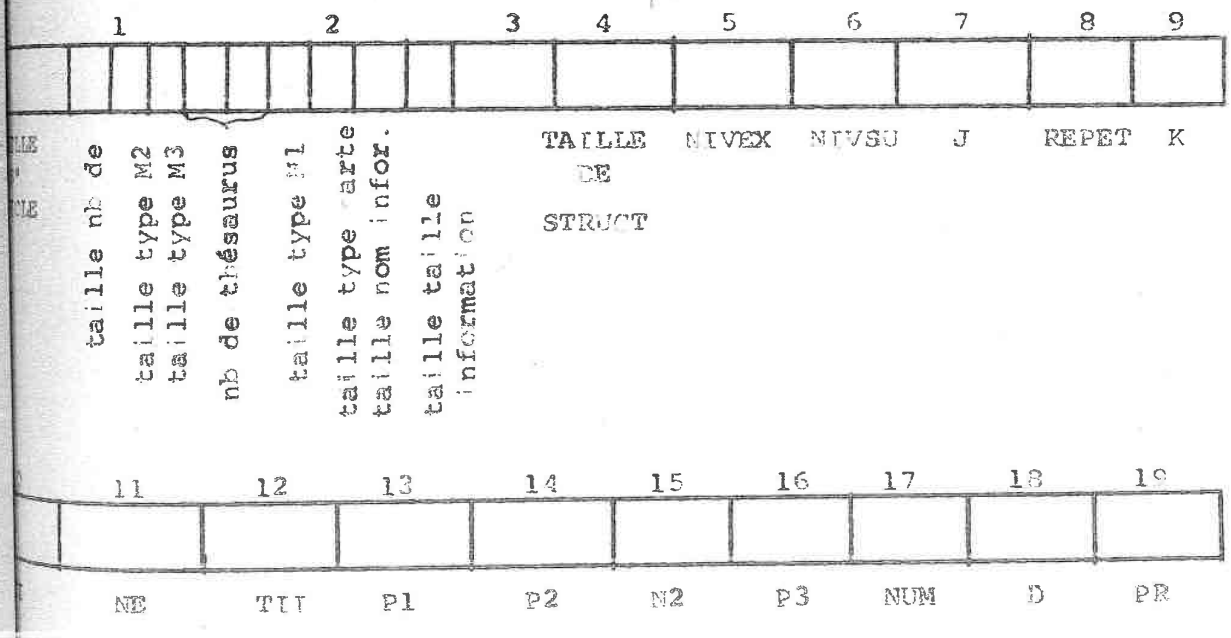
Figure 14.

On a en mémoire différentes zones qui nous permettent de stocker certaines informations ou d'en transformer d'autres en vue de former le fichier structure.

4112 - Zones de travail :

En mémoire pour le traitement on réserve plusieurs zones :
- une zone appelée TAMPØN de 20 mots où sera placé le contenu d'une carte lors de la lecture.
- une zone appelée EXPLØ de 20 mots dans laquelle sont rangés des renseignements pris sur certaines cartes et destinés au traitement de ces mêmes cartes ou d'autres cartes.

Par exemple toutes les informations de la carte 0 : prenons le cas de "taille nb de" : pour ranger les informations, il faut en connaître le nombre. Ce nombre se trouve avant la description de la première information élémentaire et la place occupée par ce nombre sur la carte est donnée par "taille nb de".
On a donnédes identificateurs aux informations présentes dans EXPLØ de façon à les retrouver facilement dans l'organigramme.
Voici la zone EXPLØ :



.../...

Les trois premiers mots de EXPLØ sont réservés pour mémoriser les informations de la carte 0. On va décrire les zones suivantes d'après leur position dans EXPLØ :

. EXPLØ + 4 : taille de STRUCT
Cette zone sera remplie dès que la zone STRUCT sera complète. Elle indique la taille de STRUCT en octets.

. EXPLØ + 5 : NIVEX
C'est le numéro du niveau qui est en cours de traitement.

. EXPLØ + 6 : NIVSU
C'est le nombre de noeuds de niveau 3 présents dans le noeud de niveau 2 en cours de traitement.

. EXPLØ + 7 : J
C'est le nombre d'informations élémentaires présentes dans le noeud en cours de traitement.

. EXPLØ + 8 : REPET
Suivant sa valeur indique si l'unité d'informations en cours de traitement est "unique" ou "répétitive".

. EXPLØ + 9 : K
K est égal à J au début du traitement de chaque noeud.

. EXPLØ + 10 : TI
Dans le cas de thésaurus PRES c'est la taille de l'information. Dans le cas de thésaurus CØDE c'est la somme taille de l'information interne + taille de l'information externe.

. EXPLØ + 11 : NE
C'est le nombre d'informations présentes dans un dictionnaire.

. EXPLØ + 12 : TIE
Dans le cas du thésaurus CØDE c'est la taille de l'information interne.

. EXPLØ + 13 : P1
Permet de savoir si la taille d'une information d'une unité d'informations est variable ou non.

. EXPLØ + 14 : P2
Permet de savoir si le fichier en cours de traitement est le premier ou non.

. EXPLØ + 15 : N2
C'est le nombre de noeuds de niveau 2 présents dans le noeud de niveau 1 en cours de traitement.

. EXPLØ + 16 : P3
Permet de savoir si la bande sur laquelle on veut écrire est vierge ou non.

. EXPLØ + 17 : NUM
C'est le numéro du dictionnaire en cours de traitement.

.../...

. EXPLØ +18 : D
Permet de savoir si le fichier BIBLIØ est présent ou non.

. EXPLØ + 19 : PR
Permet de savoir si le thésaurus en cours de traitement est du type PRES ou CØDE.
Nous verrons l'utilité de tous ces mots dans la description de l'organigramme.
- une zone appelée STRUCT de 3500 mots : c'est la zone dans laquelle on construit la table structure.
- une zone appelée ERRØR de 6 mots :
.....(1)..... ERREUR .....(2)
En cas d'erreur on inscrit dans (2) le numéro de l'erreur dont on retrouve la signification dans le dictionnaire des erreurs, et dans (1) le numéro de la carte fausse.
- une zone appelée MARK de 2 mots :
..... (1) ..... ERREUR ..... (2)
Dans le cas d'un dictionnaire s'il y a erreur on inscrit dans (2) le numéro de l'erreur dont on retrouve la signification dans le dictionnaire des erreurs et dans (1) le numéro du dictionnaire faux.
- une zone appelée IMP1 de 9 mots :
"TAILLE STRUCTURE INSUFFISANTE".
On sort ce message lorsque la zone réservée pour STRUCT est insuffisante.
- une zone appelée IMP2 de 6 mots :
" LA BANDE EST PLEINE ".
On sort ce message lorsque la bande sur laquelle on écrit les fichiers est pleine.
- une zone appelée DIC de 16.000 mots :
Elle sert à construire le fichier BIBLIØ.
- les registres de travail :
Ce sont les 16 registres du 10070.
Certains ont un rôle particulier :

.pour la construction du fichier structure

v Le registre 1 appelé CØMP sert de compteur pour vérifier la numérotation des cartes
v Le registre 2 appelé PØSI sert de pointeur dans la zone STRUCT
v Le registre 3 appelé tamp sert de pointeur dans la zone TAMP
v Le registre 9 appelé TAILLE sert à compter les octets occupés par les informations élémentaires et à compléter la zone "position par rapport à l'origine du noeud" et dans le cas d'une unité d'informations "répétitive", la zone "longueur de l'unité des informations".

.../...

v Le registre 13 appelé SUITE sert de compteur pour vérifier la succession des cartes lorsqu'il y a des cartes "suite".

.pour le traitement des dictionnaires :

v Le registre 1 appelé LØNGDIC sert de pointeur dans la zone DIC.
v Le registre 2 appelé dic permet de connaître le numéro du dictionnaire en cours de traitement.
v Le registre 4 appelé NBTH permet de connaître le nombre de thésaurus à ranger dans BIBLIØ.

- une zone appelée NØM de 5 mots :

Elle sert à écrire le LABEL du fichier structure sur bande. Ce label ne doit jamais commencer par un blanc sinon la zone serait ignorée.

4113 - Quelques particularités du programme :

- on a cinq zones en mémoire qui servent à faire des comparaisons.

. VAL qui contient la taille maximum en octets de la structure
. BLANC qui contient la valeur X'40404040'.
. ZERØ qui contient la valeur X'00000000'.
. PRES qui contient le mot PRES X'D7D9C5E2'.
. FI qui contient le mot "FIXE".
. RE qui contient le mot "REPE".

- on a une zone TRAD1 de 32 mots qui permet de passer de l'EBCDIC au binaire lorsqu'on a des comparaisons à faire en mémoire (instruction CVA) ou inversement du binaire à l'EBCDIC lorsqu'on veut imprimer un message (instruction CVS).

- les caractères à stocker de chaque carte, c'est-à-dire type de carte, type de noeud, nom de l'information, taille et thésaurus sont rangés dans STRUCT. Ce rangement se fait à l'aide d'une instruction métasymbol MBS qui utilise 2 registres :

.le registre 15 : dans lequel on range sur les trois derniers octets l'adresse du récepteur en octets :
ce sera BA (STRUCT) + PØSI (adresse de l'origine de la zone STRUCT en octets + pointeur en octets) et sur le premier octet le nombre d'octets à déplacer. Ce nombre peut-être connu à l'avance : 4 pour le nom de l'information ou connu seulement seulement à l'exécution : taille type de M2 ou type de M3 de EXPLØ pour le type de noeud.

.le registre 14 dans lequel on range l'adresse de l'émetteur en octets. Ce sera BA(TAMP) + tamp (adresse de l'origine de la zone TAMP en octets + pointeur en octets).

.../...

Dans l'organigramme on a écrit :
type de carte ——> STRUCT, PØSI
ou nom information ——> STRUCT, PØSI

- Certaines séquences d'instructions sont souvent répétées dans le programme en particulier celles qui utilisent MBS. Pour éviter de les reproduire plusieurs fois dans le programme on a utilisé des procédures.
On décrit au début du programme chaque procédure et quand on veut l'appeler dans le programme on écrit la ligne de référence de cette procédure.
Exemple : On veut ranger le nom de l'information dans STRUCT

```
procédure   RANGER      CNAME
                        PRØC
            LF          EQU          Ø
                        LI,CF (2)    BA (TAMP)
                        AW,CE (2)    AF (1)
                        LI,CF (3)    BA (STRUCT)
                        AW,CE (3)    AF (2)
                        LI,CF (4)    AF (3)
                        STB,CF(4)    CF (3)
                        MBS,CF(2)     0
                         PEND
```

Ligne de référence ou instruction d'appel :

```
ETI4      RANGER, 14, 15, 0      3, 2, 4
```

```
LF       est remplacé par  ETI4
CF(2)    est remplacé par   14
CF(3)    est remplacé par   15
CF(4)    est remplacé par    0
AF(1)    est remplacé par    3
AF(2)    est remplacé par    2
AF(3)    est remplacé par    4
```

Dans le programme on utilise 5 procédures :
NØMBRE
RANGER
PLACER
REME
DICØ

- Les instructions :
CØMP = CØMP + 1
tamp = tamp + 2
PØSI = PØSI + 4
NIVSU = NIVSU - 1

.../...

représentent l'incrémentation ou la décrémentation des différents registres ou mots de mémoire.
Les valeurs exprimées pour tamp et PØSI sont en octets, pour CØMP en mots.

412 - Organigramme détaillé :

Pour bien suivre cette analyse il est préférable de prendre l'organigramme.

DEB on ouvrele fichier lecteur de cartes et le fichier imprimante.
Certains indicateurs sont initialisés en DEB, d'autres en DEB1.
En effet si dans les données on a plusieurs fichiers structures on ne passera que la première fois en DEB et les autres fois en DEB1.
Seront donc initialisés en DEB les indicateurs ne devant être initialisés qu'une fois et en DEB1 ceux qui doivent être initialisés au début du traitement de chaque structure.
On met P2, P3, et D à O. On verra précisement à quoi servent ces indicateurs plus loin.
On initialise les 16000mots réservés pour DIC à O. On lit une carte. on a auparavant précisé dans la déclaration du DCB l'adresse de branchement en cas de fin de cartes : cette adresse est FINI.

DEB1 On initialisePØSI et CØMP à O, N2, NIVEX et REPET à 1.
On initialise les 3500 mots de STRUCT à O.
test : numéro de carte = CØMP? En effet, toutes les cartes sont numérotées en croissant. CØMP qui est un compteur interne est en binaire tandis que le numéro de carte est en EBCDIC. Avant de faire la comparaison il faut d'abord décoder le numéro de carte.
non : on range 1 dans un registre (c'est le numéro de l'erreur qu l'on retrouvera dans le dictionnaire des erreurs) et on se branche au traitement en cas d'erreur : FAUTE.
oui : test numéro de niveau = O? en effet on vient de lire la première carte.
non : erreur 2 et branchement en FAUTE
oui : on range dans les trois premiers mots de EXPLØ les informations de la carte O c'est-à-dire : la taille du numéro d'article, la taille nombre de, la taille de type de M2 et type de M3, le nombre de thésaurus, la taille de type de carte, la taille de type de M1, la taille du nom de l'information et la taille de la taille de l'information.

.../...

On se branche en NIVEAU.

FAUTE On range le registre contenant le numéro de l'erreur dans le sixième mot de la zone ERRØR. On transfère le numéro de la carte fausse qui se trouve alors dans la zone TAMPØN dans le deuxième mot de la zone ERRØR. On imprime la zone ERRØR et on se branche en M2.

NIVEAU
test : N2 = 0 ? sur la carte de structure contenant la description du niveau 1 on a le nombre de noeuds de niveau 2. Chaque fois qu'un noeud de niveau 2 aura été traité on fera N2= N2-1 et on se branchera en NIVEAU.
La première fois où l'on se branche en NIVEAU on ne connait pas encore le nombre de noeuds de niveau 2 mais on sait qu'il est au **moins** égal à 1. C'est pourquoi l'on a initialisé N2 à 1.
oui : Tous les noeuds de niveau 2 ont été traités, on va donc ranger le fichier sur bande. Pour cela on se branche en FIN.
non : On initialise P1 à 0. P1 est le mot qui nous permet de savoir si dans une unité d'informationsune information est de taille variable. En effet dès qu'une information de taille variable est rencontrée, pour le reste de l'unité d'informations la zone "position par rapport à l'origine du noeud" sera égale à "V" et pour toute l'unité d'informations la zone "longueur de l'unité d'informations si répétitive" sera égale à "V".
P1 devra donc rester positionné pendant tout le traitement d'une unité d'informations puis remis à 0 à l'arrivée d'un nouveau noeud. C'est pourquoi on l'initialise au début de NIVEAU puisque NIVEAU est l'endroit où l'on se branche après le traitement de chaque noeud. on lit une carte.
On initialise SUITE à 1. En effet SUITE permet de vérifier la séquence des cartes lorsque des cartes "suite" sont nécessaires pour la description d'un noeud. Donc lorsqu'on change de noeud on doit remettre SUITE à 1.
CØMP = CØMP + 1 : on incrémente CØMP de 1 puisqu'on vient de lire une nouvelle carte.
test : CØMP = numéro de carte ? après avoir décodé le numéro de carte.
non : erreur 3 et branchement en FAUTE
oui : tamp = 4. On vient de lire 4 caractères pour le numéro de carte. On initialise donc le pointeur de TAMP à 4 octets.

.../...

test : niveau = NIVEX ? dans NIVEX on a le numéro du niveau en cours de traitement.
La première fois où l'on passe par ce test le niveau doit-être égal à 1, c'est pourquoi en DEB1 on avait initialisé NIVEX à 1.
NIVEX est en binaire tandis que le numéro de niveau sur la carte est en EBCDIC. Il faut donc d'abord décoder le numéro de niveau avant de faire la comparaison.
non : erreur 4 et branchement en FAUTE
oui : niveau tient sur 1 octet on incrémente donc le pointeur de TAMP de 1.
tamp = tamp + 1.
test : numéro de carte dans le niveau = SUITE ? pour la première carte décrivant le niveau le numéro de carte est égal à 1, pour les cartes "suite" il sera égal à 2, 3 .....
Le numéro est de taille variable et sa taille nous est donnée par "taille Nb de" de la zone EXPLØ. On va donc chercher cette taille puis on décodera le numéro de carte qui est en EBCDIC et on fera la comparaison.
non : erreur 5 et branchement en FAUTE
oui : on vient de voir la taille de "Nb de"
on incrémente donc tamp de taille "Nb de"
tamp = tamp + taille Nb de;
On range le type de carte dans la zone STRUCT grâce au pointeur PØSI.
Le type de carte se trouve sur un mot dans le fichier structure, on incrémente donc PØSI de 4, et sur 2 octets sur la carte structure, on incrémente donc tamp de 2.
PØSI = PØSI + 4
tamp = tamp + 2
test : nivex = 1 ?
oui : si le niveau en cours de traitement est 1 on range le type de M1 dans la zone STRUCT grâce au pointeur PØSI.
Le type de M1 se trouve sur un mot dans STRUCT. on incrémente don PØSI de 4 et sur 2 octets sur la carte, on incrémente donc tamp de 2.
PØSI = PØSI + 4
tamp = tamp + 2
La place occupée par le nombre de noeuds de niveau 2 est de taille variable. Elle est donnée par "taille type de M2" de la zone EXPLØ. On range alors le nombre de noeuds de niveau 2 dans N2 puis on incrémente tamp de la place occupée par ce nombre

.../...

tamp = tamp + taille type de M2
On se branche en ETI 2.
non : test nivex = 0?
non : si le numéro de niveau n'est pas 2 il est forcément 3 puisque le nombre de niveaux possibles est 3 et que le niveau 1 a déjà été traité.
La place occupée par type de M3 est de taille variable. Elle est donnée par "taille type de M3" de la zone EXPLØ.
On range alors le type de M3 dans la zone STRUCT grâce au pointeur PØSI.
La place occupée par type de M3 dans STRUCT est de un mot. On incrémente donc PØSI de 4. On incrémente tamp de la place occupée par le type de M3.
PØSI = PØSI + 4
tamp = tamp + taille type de M3
On se branche en ETI3.
oui : la place occupée par type de M2 est de taille variable. Elle est donnée par "taille type de M2" de la zone EXPLØ.
On range alors le type de M2 dans la zone STRUCT grâce au pointeur PØSI.
La place occupée par type de M2 dans STRUCT est de un mot, on incrémente donc PØSI de 4.
On incrémente tamp de la place occupée par type de M2.
PØSI = PØSI + 4
tamp = tamp + taille type de M2.
test : répétitivité ≠ 0 ?
On a vu dans le chapitre précédent que dans le cas du niveau 2 à la place du nom de l'information dans STRUCT on avait FIXE ou REPE. Dans le cas du niveau 2 "répétitif" on devra trouver dans les cartes composant l'article le nombre de répétitions.
oui : on range FIXE dans STRUCT grâce à PØSI et on se branche en PL7.
non : test : répétitivité = 1 ?
non : erreur 6 et branchement en FAUTE. En effet le noeud ne peut-être autre chose qu'"unique" ou "répétitif". C'est une erreur au remplissage des bordereaux.
oui : on range REPE dans STRUCT grâce à PØSI et on se branche en PL7.

PL7 Dans le fichier structure en face de chaque noeud de niveau 2 on aura comme seule information FIXE ou REPE, on sautera les zones "taille information", "longueur par rapport à l'origine du noeud", "longueur de l'unité si "répétitive"" et "thésaurus" qui tiennent chacune sur un mot. On incrémente donc PØSI de 20.
La répétitivité 1 ou 0 tient sur un octet sur la carte. On incrémente donc tamp de 1.

.../...

PØSI = PØSI + 20
tamp = tamp + 1
test : présence de noeuds de niveau 3 = 1 ?
non : c'est que les informations élémentaires se trouvent au niveau 2.
On se place dans STRUCT devant "nom de l'information" et on va ranger toutes les informations élémentaires. Pour cela on saute "type de carte" et "type de M2" dans STRUCT, zones qui tiennent chacune sur un mot. On incrémente donc PØSI de 8
PØSI = PØSI + 8
Il n'y a pas de noeuds de niveau 3 donc NIVSU = 0
Présence de noeuds de niveau 3 tient sur un octet sur la carte, on incrémente donc tamp de 1.
tamp = tamp + 1.
On se branche en ETI3.
oui : le noeud de niveau 2 est suivi de noeuds de niveau 3. On veut alors connaître le nombre de noeuds de niveau 3 présents dans ce noeud de niveau 2. La place occupée par ce nombre est donnée par "taille Nb de" de EXPLØ. Le nombre est en EBCDIC, on le décode et on le range en NIVSU.
La carte de structure suivante doit-être une carte de niveau 3, on met 3 dans NIVEX et on se branche en NIVEAU.

ETI3 test : l'unité d'informations = 1 ?
oui : il s'agit d'une unité d'informations "unique", on range la valeur 1 dans REPET.
Unité d'informations tient sur un octet sur la carte, on incrémente donc tamp de 1.
tamp = tamp + 1.
On se branche en ETI2.
non : test : l'unité d'informations = 2 ?
oui : il s'agit d'une unité d'informations "répétitive".
On range la valeur 2 dans REPET. On incrémente tamp de 1 et on se branche en ETI2.
tamp = tamp + 1
Dans le cas d'une unité d'informations "répétitive" on aura dans l'article 1"information de service" (cf chapitres précédents).
non : l'unité d'informations ne peut-être qu'"unique" ou "répétitive". Il y a donc eu erreur au remplissage des bordereaux Erreur 7 et branchement en FAUTE.

ETI2 Chaque type de noeud comporte une liste exhaustive d'informations élémentaires qui est l'unité d'informations. le nombre d'informations présentes dans cette liste est donné par un nombre. La place occupée par ce nombre est connue grâce à "taille Nb de" de EXPLØ. Ce nombre est en EBCDIC, on le décode et on le range en J. Puis on range J dans K.

.../...

On va ranger toutes les informations élémentaires dans la structure en complétant les différentes zones de la table. Seule la zone "taille de l'unité d'informations si répétitive" ne peut-être remplie qu'à la fin. Cette zone est alors complétée pour chaque information élémentaire, c'est-à-dire que l'on va remonter la table. Il faut donc conserver le nombre d'informations présentes dans cette unité "répétitive".
Si les informations font partie d'une unité "unique", ce nombre K ne sera pas utilisé. On initialise TAILLE à 0.
TAILLE est la zone dans laquelle on va accumuler la taille des différentes informations élémentaires et qui nous permettra de remplir "position par rapport à l'origine du noeud" de la table structure. Après avoir traité la dernière information élémentaire TAILLE représentera la "taille de l'unité d'informations" et cela nous permettra de remplir la zone correspondante de la table structure. On vient de décoder le nombre d'informations élémentaires, on incrémente donc tamp de la place occupée par ce nombre.
tamp = tamp + taille Nb de

ETI4 On remplit la zone "nom de l'information" de la table structure. On se positionne toujours dans cette table grâce au pointeur PØSI.
Nom de l'information occupe un mot dans la table structure et sur la carte. On incrémente donc PØSI et tamp de 4.
PØSI = PØSI + 4
tamp = tamp +4
Pour faciliter l'exploitation on a dit qu'à côté de chaque information se trouvaient les caractères "F" ou "R" qui signifient respectivement que l'information fait partie d'une unité d'informations "unique" ou "répétitive".
test : REPET = 1 ?
oui : on range "F" dans la table structure et on se branche en PL10.
non : on range "R" dans la table structure et on se branche en PL10.

PL10 La place occupée par "F" et "R" est de un octet. On incrémente donc PØSI de 1.
PØSI = PØSI + 1
test : PL = 0 ?
oui : les informations élémentaires déjà rangées de l'unité d'informations en cours de traitement étaient de taille connue.

.../...

test : taille information = "V" ?
oui : on positionne l'indicateur P1 à 1. En effet, pour les informations élémentaires qui vont suivre on ne pourra connaître leur "position par rapport à l'origine du noeud", ni la "longueur de l'unité si elle est répétitive". Toutes ces zones auront un "V". On se branche en PL12.
non : on se branche en ETI1.
non : on se branche en PL11.

PL11 test : taille information = "V" ?
On a déjà eu une information de taille variable dans cette unité
non : on se branche en ETI1.
oui : on se branche en PL12.

PL12 La taille de l'information en cours de traitement est variable.
On va ranger "V" dans la table structure. Ce "V" occupe 3 octets sur la carte et dans la table structure. On incrémente donc PØSI de 3. Si on ne connaît pas la taille de l'information, il ne peut y avoir de référence à un thésaurus. La place occupée par le repère de thésaurus sur la carte est de 2 octets. On incrémente donc tamp de 5.
tamp = tamp + 5
On range taille dans la zone "position par rapport à l'origine du noeud". Si une des informations précédentes était déjà de taille variable : TAILLE est égaleà "V".
On va ranger l'information suivante, on se positionne donc dans la table derrière le thésaurus. La place occupée par les zones "position par rapport à l'origine du noeud", "taille de l'unité si répétitive" et "thésaurus" est de 3 mots (1 mot pour chacune). On incrémente donc PØSI de 12.
PØSI = PØSI + 12
Il suffit d'une information élémentaire de taille variable dans une unité pour que les zones "position par rapport à l'origine du noeud" et "taille unité si répétitive" soient inconnues. Pour le reste de l'unité TAILLE sera alors égale à "V".
On se branche en PL13.

ETI1 La taille de l'information est connue. On la range dans la table structure. Elle occupe 3 octets dans cette table et sur la carte. On incrémente donc PØSI et tamp de 3.
PØSI = PØSI + 3
tamp = tamp +3
On remplit la zone "position par rapport à l'origine du noeud" en rangeant TAILLE. On saute la zone "longueur de l'unité si répétitive" qui sera remplie à la fin s'il y a lieu.

.../...

On incrémente donc PØSI de 8.
PØSI = PØSI + 8
test : P1 = 1 ?
non : il n'y a pas eu d'information de taille variable dans l'unité ; on incrémente donc TAILLE de la taille de l'information qui vient d'être rangée.
oui : il y en a eu. TAILLE est alors égale à "V".
test : Thésaurus = blanc ?
non : un repère de thésaurus est mentionné pour cette information. Il faut donc le décoder et le reporter dans la table structure.
oui : il n'y a pas de repère de thésaurus. On se branche en PL14.

PL14 Le repère occupe 2 octets sur la carte et 4 octets dans la table. On incrémente donc tamp de 2 et PØSI de 4.
tamp = tamp + 2
PØSI = PØSI + 4

PL13 test : PØSI = VAL ?
non : branchement en PLVAL
oui : le pointeur a atteint la taille maximum que l'on avait réservéepour la table structure.
On sort un message : "TAILLE STRUCTURE INSUFFISANTE". On se branche en K2.

K2 test : P2 = 0 ?
oui : le fichier structure en cours de traitement est le premier. On arrête donc le programme.
non : ce n'est pas le premier. Il y en a donc déjà eu un écrit sur bande.
Avant d'arrêter le programme il faut d'abord fermer la bande.

Remarque : Cas où l'on a eu le message "TAILLE STRUCTURE INSUFFISANTE". Le programme est formé d'un programme principal et d'un sous-programme avec toutes les réservations. On a d'autres sous-programmes sur bande sous forme de modules objets et ayant des réservations plus grandes (dans ce cas-ci : pour STRUCT). On a également un exemplaire du programme principal sous forme de module objet.
On va donc créer un nouveau module de chargement avec le programme principal et un sous-programme ayant réservé pour STRUCT une taille plus grande. Ce nouveau module de chargement pourra remplacer l'ancien jusqu'à ce qu'il y ait un autre changement.

.../..

PLVAL J : représente le nombre d'informations élémentaires à ranger. On vient d'en ranger une, on décrémente donc J de 1.
test : J = 0 ?
oui : branchement en PL13.
non : on va ranger l'information suivante.
test : octet suivant sur carte = ∅ ?
non : test : tamp = 80 ?
non : on se place dans la table structure devant "nom de l'information". on incrémente donc PØSI de 8 puisqu'on saute : "type de carte" et "type de noeud".
PØSI = PØSI + 8
On se branche en ETI4.
oui : on a oublié de mettre le caractère ∅ en fin de carte pour indiquer qu'il y a une carte "suite". Erreur 11 et branchement en FAUTE.
oui : on a une carte "suite".
on lit cette carte et on incrémente CØMP de 1;
CØMP = CØMP + 1
test: numéro sur la carte = CØMP? après avoir décodé le numéro de carte.
non : erreur 8 et branchement en FAUTE.
oui : la place occupée par le numéro de carte est de 4 octets.
On initialise donc tamp à 4.
tamp = tamp + 4
test : niveau = NIVEX ? après avoir décodé niveau.
non : erreur 9 et branchement en FAUTE.
oui : le niveau mentionné sur la carte est bien celui qui est en cours de traitement.
La place occupée par niveau sur la carte est d'un octet. On incrémente donc tamp de 1.
tamp = tamp + 1
On incrémente aussi SUITE de 1.
SUITE = SUITE + 1
test : numéro de carte dans niveau = SUITE ?
après avoir décodé le numéro de carte et pris sa taille dans EXPLØ "taille Nb de". En effet, toutes les cartes "suite" sont numérotées.
non : erreur 10 et branchement en FAUTE.
oui : on incrémente tamp de la place occupée par le numéro de carte dans niveau.
tamp = tamp + taille Nb de.
On se place devant "nom information" de la table en sautant "type de carte" et "type de noeud".
PØSI = PØSI + 8
On se branche en ETI4.

.../...

PL15 test : REPET = 1 ?
oui : branchement en PL16.
non : on va remplir la zone "longueur de l'unité si répétitive".
On range le pointeur de STRUCT dans CØRR. CØRR est le pointeur qui va nous permettre de se placer devant la zone à remplir pour chaque information élémentaire faisant partie de cette unité.
On se place d'abord devant cette zone pour la dernière information qui vient d'être traitée.
CØRR = CØRR - 8

PL17 on range TAILLE dans la table structure. Une information élémentaire vient d'être complétée. On décrémente donc K de 1.
K = K - 1
test : K = 0 ?
non : on va se placer devant la zone suivante en sautant 7 mots.
CØRR = CØRR - 28
On se branche en PL17
oui : on se branche en PL16.

PL16 test : NIVEX = 1 ?
oui : derrière une carte de niveau 1 vient toujours une carte de niveau 2
On positionne donc NIVEX à 2 et on se branche en NIVEAU.
non : test : NIVEX = 2 ?
oui : les informations élémentaires qui viennent d'être rangées faisaient partie d'un noeud de niveau 2. Il n'y avait donc pas de noeuds de niveau 3 dans ce noeud de niveau 2.
On décrémente le nombre de noeuds de niveau 2 présents dans cette structure de 1.
N2 = N2 - 1
On se branche en NIVEAU pour traiter un nouveau noeud de niveau 2.
non : les informations élémentaires qui viennent d'être rangées faisaient partie d'un noeud de niveau 3.
On décrémente donc le nombre de noeuds de niveau 3 présents dans le noeud de niveau 2 de 1.
NIVSU = NIVSU - 1
test : NIVSU = 0 ?
non : on va traiter un nouveau noeud de niveau 3. On se branche en NIVEAU.
oui : on a traité tous les noeuds de niveau 3 faisant partie du noeud de niveau 2. On décrémente donc le nombre de noeuds de niveau 2 présents dans la structure de 1. On positionne le niveau à 2 puisque le prochain noeud va être un noeud de niveau 2.

.../...

N2 = N2 - 1
On se branche en NIVEAU.

FIN PØSI est le pointeur de la table structure. Lorsqu'on a rempli la table structure PØSI indique sa taille en octets.
On range donc PØSI dans EXPLØ et cette partie de EXPLØ sera mise sur bande avec la table structure, le tout formant le fichier structure.
test : P2 = O ?
non : branchement en PL91
oui : il s'agit du premier fichier structure à ranger sur bande au cours de cette manipulation.
test : premier label est vierge ?
oui : il n'y a encore eu aucun fichier mis sur bande. On se branche en FINVØL.
non : on positionne P3 à 1. Cela indique qu'il y a déjà un ou plusieurs fichiers sur bande.
On se place au début de la bande en la fermant avec rebobinage et en la réouvrant.
test : BIBLIØ existe ?
non : branchement en FINVØL
oui : on a vu dans les chapitres précédents que si BIBLIØ existai (c'est le fichier des thésaurus) il fallait que ce soit toujours le dernier sur la bande à cause des mises à jour. On va donc lire le fichier BIBLIØ en mémoire dans la zone DIC.
On positionne D à 1. D indique que lorsqu'on aura rangé tous les fichiers structures, il faudra réécrire DIC à la fin sur la bande. On rebobine la bande et on se place devant le fichier BIBLIØ de façon à l'écraser à la prochaine écriture.

PL91 On prépare une zone qui représentera le label du fichier structure que l'on va écrire.

PL19 test : P2 = O ? est-ce le premier fichier ?
non : branchement en ARM2
oui : test : P3 = O ? la bande est-elle vierge ?
oui : branchement en ARM2
non : test D = O ? BIBLIØ existe ?
Toute une série de tests sont faits pour savoir où l'on doit se positionner sur la bande.
oui : on se place derrière le dernier fichier écrit.
non : ARM2 on se place à l'endroit où a eu lieu la dernière fermeture.

.../...

ARM3 On écrit sur bande le fichier-structure qui est formé de deux enregistrements. Le premier comprend les cinq premiers mots de EXPLØ : renseignements de la carte 0 et la taille de la table structure. Ces renseignements serviront pour le programme fichier maître.
Le deuxième enregistrement est la table structure. On positionne P2 à 1 pour indiquer que le prochain fichier structure, s'il y en a un, n'est plus le premier. On range le nombre de thésaurus qui se trouvait sur la carte 0 dans R4.
test : Nb de thésaurus = 0 ?
oui : il n'y a pas de dictionnaires présents dans ce fichier structure. On passe donc au suivant. On lit une carte et on se branche en DEB1.
non : il y a des dictionnaires. On va donc compléter le fichier BIBLIØ s'il existe déjà, ou le créer dans le cas contraire. On positionne le pointeur de carte à 0 et celui des numéros "suite" à 1.
tamp = 0
SUITE = 1.

On lit le début du premier dictionnaire. On décode le numéro du dictionnaire présent sur la carte et on le range en NUM. Il servira à vérifier, si des cartes "suite" sont nécessaires, si l'on est toujours dans le même dictionnaire.
On décrémente dic de 1 ; dic est le pointeur de DIC pour atteindre les adresses des dictionnaires. Si le numéro du premier dictionnaire est 1, son adresse se trouve dans le premier mot de DIC. Le pointeur dic devra être égal à 0 (c'est une convention de langage)
Le numéro de dictionnaire occupe 2 caractères.
On incrémente donc tamp de 2.
tamp = tamp + 2
test : DIC, dic = 0 ?
non : le numéro de dictionnaire a déjà été employé ou par l'utilisateur ou par le programme de mise à jour. Il faut donc le changer -Erreur 1. On range le numéro de dictionnaire et le numéro de l'erreur dans MARK. On imprime cette zone MARK et on se branche en K2.
oui : l'emplacement pour mettre le dictionnaire est libre. On calcule dans longdic l'adresse où le dictionnaire se trouvera dans la zone DIC. Puis on range cette adresse dans l'un des 50 premiers mots de DIC, dans celui qui est indiqué par le pointeur dic (cf les dictionnaires dans les chapitres précédents)
test : nom de TAMP = PRES ?
non : branchement en PL20
oui : il s'agit d'un dictionnaire servant uniquement au contrôle.

.../..

On initialise PR à 0 pour reconnaitre ce type de dictionnaire.
On range le nom dans la zone DIC grâce au pointeur longdic. Ce nom occupe 4 caractères : on incrémente donc longdic de 4.
longdic = longdic + 4
On range l'adresse qui se trouve dans longdic dans X.
X servira plus tard à ranger dans un mot la place libre restante dans le dictionnaire. On a vu dans les chapitres précédents que les dictionnairès à ranger derrière les fichiers structures ne devaient jamais déborder. On les complétait ensuite grâce au programme de mise à jour.
On incrémente tamp de 4.
tamp = tamp + 4 car la place occupée par le nom de dictionnaire sur la carte est de 4 caractères.
longdic = longdic + 4 car la place réservée pour mettre la place libre restante est de 4 caractères.
On range la taille de l'information décodée dans DIC et dans TI. La place qu'elle occupe dans DIC et sur la carte est de 3 caractères.
longdic = longdic + 3
tamp = tamp + 3
On range le nombre d'éléments présents dans le dictionnaire dans DIC et dans NE, ce nombre ayant été décodé. La place qu'il occupe dans DIC et sur la carte est de 3 caractères.
longdic = longdic + 3
tamp = tamp + 3
on se branche en DC3.

PL20 Il s'agit d'un dictionnaire de codage ou décodage. Dans ces 2 cas le traitement est le même. On positionne PR à 1 pour se situer dans ces types de dictionnaires.
On range le nom dans DIC comme pour le cas PRES.
Longdic = longdic + 4
tamp = tamp + 4
On range longdic dans X pour les raisons exposées précedemment.
tamp = tamp + 4
longdic = longdic + 4
Comme il s'agit de codage ou décodage, on va avoir 2 tailles correspondant à l'information.
La taille externe qui une fois décodée est rangée dans TI et dans DIC. La place qu'elle occupe dans DIC et sur la carte est de 3 octets.
longdic = longdic + 3
tamp = tamp + 3
La taille interne qui une fois décodée est rangée dans TII et dans DIC. La place qu'elle occupe dans DIC et sur la carte est de 3 octets.
longdic = longdic + 3
tamp = tamp + 3

.../...

On range le nombre d'éléments décodé dans DIC et dans NE. La place qu'il occupe dans DIC et sur la carte est de 3 octets.
longdic = longdic + 3
tamp = tamp + 3
Dans TI on fait la somme TI + TII, ce qui va nous permettre d'avoir un traitement commun aux différents types de dictionnaires pour ranger les informations.

DC3 On calcule la place occupée par les informations dans DIC en multipliant dans Y le nombre des informations par leur taille
test : PR = 0 ?
oui : il s'agit du cas PRES. On ajoute à Y la place occupée par le nom du dictionnaire, la place libre restante, la taille de l'information et le nombre d'informations . Cela fait 14 caractères.
non : il s'agit des autrescas. On ajoute outre ce qui précéde la taille interne de l'information. Cela fait 17 caractères.
On calcule la place libre restante dans le dictionnaire, c'est Y;
test : Y<0 ?
oui : on a voulu ranger trop d'informations en une fois. Erreur 2 et branchement en FAUTE 2.
non : on range la place libre restante Y dans DIC grâce au pointeur rangé en X.

DC2 On va ranger les informations dans le dictionnaire. On range TI caractères de la carte dans DIC. On incrémente le pointeur de DIC et celui de la carte de TI caractères.
tamp = tamp + TI
longdic = longdic + TI
On vient de ranger une information. On décrémente donc le nombre d'éléments de 1.
test : NE = 0 ?
oui : on a rangé toutes les informations de ce dictionnaire. On positionne D à 1 au cas où le fichier BIBLIØ n'existait pas encore. On décrémente le nombre de thésaurus de 1 et on passe au suivant : branchement en DC5
non : test : octet suivant de TAMP = ø ?
non : test : tamp > 80 ?
non : branchement en DC2
oui : on a oublié de signaler qu'il y avait une carte "suite". Erreur 3 et branchement en FAUTE 2.
oui : il y a une carte "suite". On lit cette carte. On initialise tamp à 0 : tamp = 0.
On décode le numéro de dictionnaire.
test : numéro de dictionnaire = NUM ?
non : une carte d'un autre dictionnaire a peut-être été glissée.

.../...

Erreur 3 et branchement en FAUTE 2.
oui : le numéro de dictionnaire occupe 2 caractères sur la carte. on incrémente donc tamp de 2.
tamp = tamp + 2
On incrémente SUITE de 1.
SUITE = SUITE + 1
test : numéro de suite = SUITE ?
On regarde si les cartes "suite" sont dans le bon ordre.
non : erreur 4 et branchement en FAUTE 2.
oui : la place occupée par le numéro de suite est de 3 caractères. On incrémente donc tamp de 3.
tamp = tamp + 3
On se branche en DC2.

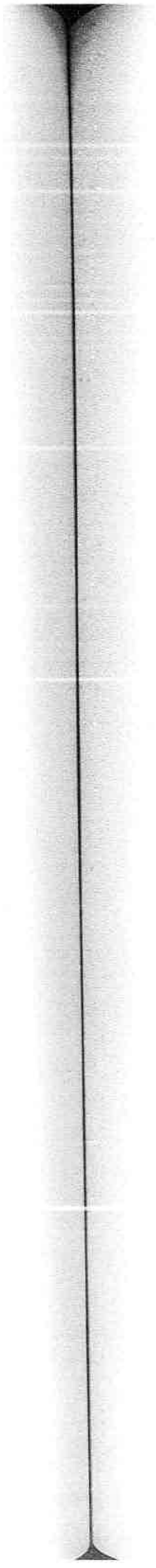
FINI C'est une séquence d'anomalie. On s'y branche quand il n'y a plus d'autres cartes à lire.
test : nb de thésaurus = 0 ?
non : il y a erreur. On avait mentionné la présence de dictionnaires derrière le fichier structure et il n'y en a pas. Erreur 12 et branchement en FAUTE2.
oui : test D = 1 ?
non : le fichier BIBLIØ n'existe pas. Branchement en DC4.
oui : le fichier BIBLIØ existe. On va donc le mettre sur bande derrière les fichiers structures. Branchement en DC4.

DC4 On ferme le fichier structure et on arrête l'exécution du programme.

FINVØL C'est une séquence d'anomalie pendant le traitement sur la bande.
test : 1er octet de R5 = '02'?
non : branchement en FIB1
oui : test : P3 = 1 ?
non : on est arrivé au bout de la bande; cela signifie qu'elle est vierge. On va donc écrire le fichier structure. Pour cela on se branche en PL91
oui : la bande n'est pas vierge et l'on est au bout de la bande. C'est une anomalie qui arrive lorsque l'on veut écrire. On imprime le message 'LA BANDE EST PLEINE' puis on ferme la bande et on arrête l'exécution.

FIB1 test : 1er octet de 5 = X '14'?
oui : le fichier que l'on cherche n'existe pas ; c'est le cas lorsque BIBLIØ n'est pas sur bande. On rebobine la bande et on se branche en PL91.
non : erreur 5 et branchement en FAUTE2.

.../...



42 - Fichier maftre :

421 - Présentation des éléments utilisés pour former le fichier maftre :

4211 - Schéma de travail :

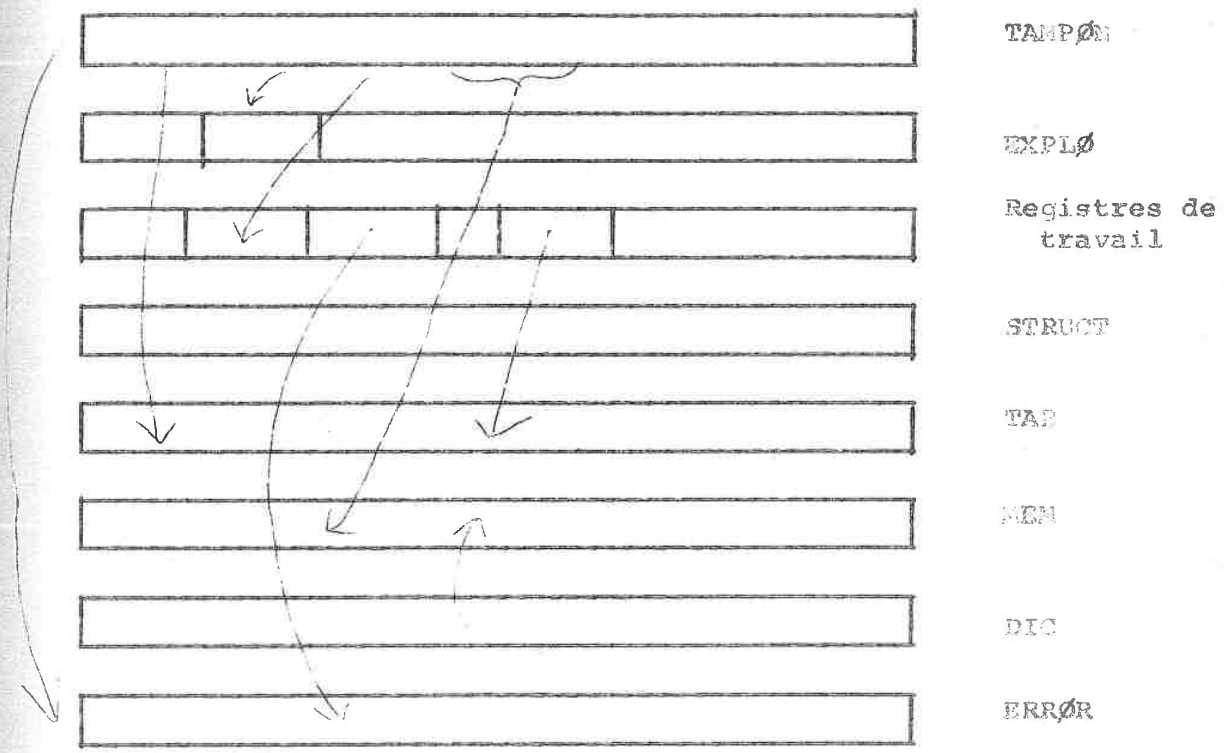


Figure 15.

Toutes ces zones vont nous aider à constmire le fichier maftre.

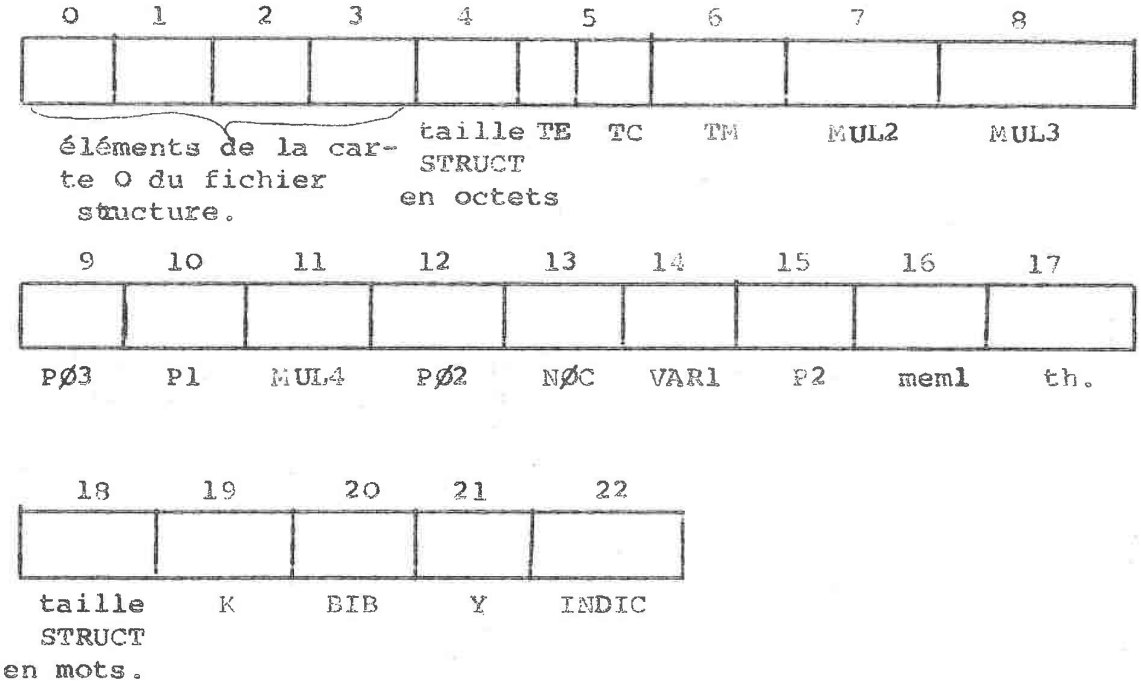
4212 - Zones de travail :

Décrivons les différentes zones utilisées pour constmire le fichier.

- une zone TAMPØN de 20 mots : même utilisation que pour le fichier structure.
- une zone EXPLØ de 23 mots : même principe que pour le fichier structure.

Voici la zone EXPLØ

.../...



Décrivons chaque zone en la repérant par son adresse par rapport à EXPLØ.

. Les trois premiers mots de EXPLØ sont destinés à mémoriser les éléments de l'ancienne carte 0, lorsqu'on a lu le fichier structure.

En effet, on en a encore besoin : par exemple la place occupée par les "informations de service" est donnée par "taille nb de", le numéro de classement à titre de vérification est répété sur chaque carte, sa taille est donnée par "taille n° d'article".

. EXPLØ + 4 : taille STRUCT

On lit également cette taille dans le fichier structure; elle est en octets.

. EXPLØ + 5

Premier demi-mot : TE. On mémorise le type de l'enregistrement que l'on est en train de traiter. Deuxième demi-mot : TC. On mémorise le type de la carte que l'on est en train de traiter.

. EXPLØ + 6 : TM

On mémorise le type de noeud que l'on est entrain de traiter.

. EXPLØ + 7 : MUL2.

Il représente le nombre de répétitions ("information de service") du noeud de niveau 2 en cours de traitement.

. EXPLØ + 8 : MUL3.

Il représente le nombre de noeuds de niveau 3 présents dans le meud de niveau 2 en cours de traitement.

.../...

. EXPLØ + 9 : PØ3.
On mémorise dans PØ3 l'adresse de l'origine dans le fichier structure du noeud de niveau 3 ou de niveau 2 (s'il ne contient pas de noeuds de niveau 3) en cours de traitement lorsqu'il est "répétitif".
. EXPLØ + 10 : P1
C'est un indicateur. Il permet de savoir si l'on a déjà écrit un enregistrement sur la bande.
. EXPLØ + 11 : MUL4.
Il représente le nombre de répétitions ("information de service") du noeud de niveau 3 en cours de traitement.
. EXPLØ + 12 : PØ2.
On mémorise dans PØ2 l'adresse de l'origine dans le fichier structure du noeud de niveau 2 en cours de traitement lorsqu'il est "répétitif".
. EXPLØ + 13 : NØC
Lorsqu'un noeud de niveau 2 est "répétitif", NØC sert à repérer l'indice du noeud en cours de traitement.
. EXPLØ + 14 : VAR1
C'est un indicateur. Lorsqu'on a lu une carte "suite" et fait toutes les vérifications possibles VAR1 permet de se brancher à l'une ou l'autre adresse.
. EXPLØ + 15 : P2.
C'est un indicateur. Lorsqu'on range une information de taille variable P2 permet de savoir s'il y a eu des cartes "suite" ou non.
. EXPLØ + 16 : mem1.
Lorsqu'on range une information de taille variable, s'il y a des cartes "suite", on mémorise le pointeur de MEM dans mem1.
. EXPLØ + 17 : th
c'est un indicateur. LOrsqu'on a utilisé un thésaurus, th permet de se brancher à l'une ou l'autre adresse.
. EXPLØ + 18 : taille de STRUCT.
C'est la taille de STRUCT trouvée dans EXPLØ + 4, mais exprimée en mots.
. EXPLØ + 19 : K.
Cést un indicateur. Il permet de savoir, lorsqu'on a rangé un enregistrement sur bande, si le programme est terminé ou s'il reste encore des cartes à lire.
. EXPLØ + 20 : BIB.
C'est un indicateur. Il permet de savoir si l'on a déjà lu le fichier dictionnaire ou non.
. EXPLØ + 21 : Y
Permet de mémoriser la taille interne de l'information lorsqu'on est dans un thésaurus.
. EXPLØ + 22 : INDIC.
Indicateur qui permet de tester si la bande est vierge ou pleine.

.../...

- Les registres de travail :

Certains ont un rôle particulier.

. Pour tout le programme.

*Le registre 1 appelé TAB sert de pointeur dans la zone TAB : c'est un pointeur de mots.

*Le registre 2 appelé PØSI sert de pointeur dans la zone STRUCT : c'est un pointeur de mots.

*Le registre 3 appelé tamp sert de pointeur dans la zone TAMP : c'est un pointeur d'octets.

*Le registre 4 appelé mem sert de pointeur dans la zone MEM : c'est un pointeur d'octets.

*Le registre 8 appelé tampl sert à mémoriser tamp lorsqu'il y a une information de taille variable.

*Le registre 9 appelé VAR sert à compter le nombre de caractères d'une information lorsqu'elle est de taille variable

*Le registre 13 appelé SUITE sert de compteur pour vérifier la succession des cartes lorsqu'il y a des cartes "suite".

. Pour le thésaurus.

*Le registre 5 appelé Y contient la taille interne de l'information lorsqu'il y a codage.

*Le registre 6 appelé I contient le nombre d'informations présentes dans le thésaurus.

*Le registre 10 appelé X contient la taille externe de l'information.

*Le registre 11 appelé V contient la "place libre restante".

*Le registre 12 appelé dic sert de pointeur dans la zone DIC. C'est un pointeur d'octets.

- Une zone STRUCT de 3500 mots :

Elle contient la table structure que l'on a luesur la bande.

- Une zone TAB de 300 mots :

Elle contient pour chaque noeud, le type de noeud, son adresse par rapport à l'origine de l'enregistrement, sa multiplicité. Les deux premiers mots contiennent la taille de l'enregistrement et la taille de la table.

- Une zone MEM de 6000 mots :

C'est la zone dans laquelle on construit l'article.

- Une zone DIC de 16000 mots :

Elle contient le fichier BIBLIØ que l'on a lu sur bande.

.../...

- Une zone ERRØR de 6 mots :

(1) ..... ERREUR ..... (2)
En cas d'erreur on inscrit dans (2) le numéro de l'erreur dont on retrouve la signification dans le dictionnaire des erreurs et dans (1) le type de carte faux.

- Une zone IMP1 de 9 mots :

"TAILLE STRUCTURE INSUFFISANTE".
On sort ce message lorsque la zone réservée pour STRUCT est insuffisante.

- Une zone IMP2 de 8 mots :

"TAILLE TABLE INSUFFISANTE".
On sort ce message lorsque la zone réservée pour TAB est insuffisante.

- Une zone IMP3 de 12 mots :

" TAILLE ZØNE DE RANGEMENT : MEM INSUFFISANTE".
On sort ce message lorsque la zone réservée pour MEM est insuffisante.

- Une zone IMP4 de 6 mots :

" LA BANDE EST PLEINE".
On sort ce message lorsqu'il n'y aplus de place sur la bande.

- Une zone IMP5 de 8 mots :

"TYPE D'ENREGISTREMENT FAUX".
On sort ce message lorsqu'on n'a pas trouvé de fichier structure correspondant à ce type d'enregistrement.

- Une zone NØ4 de 5 mots :

Elle contient le type d'enregistrement. Elle permet de lire le fichier structure correspondant à ce type.

- Une zone NØ1 de 7 mots :

Elle contient le numéro de classement de l'article en cours de traitement.

- Une zone Z de 1 mot :

Elle contient l'adresse de début du dictionnaire ou s'il y a débordement l'adresse de début du dictionnaire sur lequel on déborde.

.../...

4213 - Quelques Particularités du programme :

- On a 9 zones en mémoire qui servent à faire des comparaisons.
  . VAL qui contient la taille maximum de la zone STRUCT en octets
  . V1 qui contient la taille maximum de la zone VAR en mots
  . M1 qui contient la taille maximum de la zone MEM en octets
  . BLANC qui contient la valeur X '40404040'
  . ZERØ qui contient la valeur X '00000000'
  . PRES qui contient le mot 'PRES'
  . CØDE qui contient le mot 'CØDE'
  . FI qui contient le mot 'FIXE'
  . RE qui contient le mot 'REPE'

- On a une zone TRAD1 : même utilisation que pour le fichier structure.

- On utilise l'instruction MOS pour ranger les informations de la carte dans MEM. Cette instruction a été décrite dans le fichier structure.
Seules les adresses contenues dans l'émetteur et le récepteur différent.
Ce sera pour l'émetteur BA (TAMP) : adresse de début de la carte et pour le recepteur BA(MEM) adresse de début de MEM. On se positionne dans ces deux zones grâce aux pointeurs tamp et mem.

- Comme dans le fichier structure, on a utilisé des procédures :
CØMPAR
CLASSE
NØMBRE
RANGE
MEME

- Les instructions
tamp = tamp + 2
PØSI = PØSI + 1
MUL2 = MUL2 - 1
représentent l'incrémentation ou la décrémentation des différents registres ou mots de mémoire.

422 - Organigramme détaillé :

Pour comprendre cette analyse, il faut la suivre sur l'organigramme.

DEB On initialise DIF, P2, P1, TM, VAR1 et t1 à 0, MUL2 à 1.
On lit une carte dans TAMP.
Pour les mêmes raisons que dans le fichier structure, certains indicateurs sont initialisés en DEB, d'autres en DEB1.

.../...

DEB1 On initialise TAB à 0 et tamp à 1.
test : type de carte de TAMP = TE ?
On a initialisé TE à 0. La première fois le test ne pourra donc pas être positif.
oui : les autres fois, si le test est positif : l'article que l'on va traiter est du même type que le précédent, on ne lira donc pas la bande structure puisque le fichier structure est déjà en mémoire centrale.
On se branche en PL1.
non : on range le type de carte dans TE.
On range TE dans NØM. NØM est une zone de 5 mots qui va nous permettre de lire le fichier structure correspondant à TE. On doit obligatoirement remplir ces 5 mots, l'ordinateur s'arrêtera au premier blanc rencontré. C'est la raison pour laquelle le label ne doit pas commencer par un blanc.

PL2 On ouvre la bande structure.
test : fichier de label NØM existe ?
non : le type d'enregistrement doit-être faux. On imprime un message et on arrête l'exécution du programme.
oui : le fichier structure est composé de 2 enregistrements. On lit le premier dans EXPLØ . Dans la zone EXPLØ on a alors la taille de la table structure en octets.
test : Nb octets de STRUCT > VAL ?
VAL est la taille en octets de la zone que l'on a réservée pour STRUCT.
non : on se branche en PLVAL
oui : cette taille VAL est insuffisante
Dans ce cas on avait déjà eu une zone insuffisante lors de la mise sur bande de ce fichier structure. On imprime le message "TAILLE STRUCTURE INSUFFISANTE".
test : P1 = 0 ?
non : on a déjà écrit un enregistrement sur bande, il faut donc fermer la bande. On se branche en PL0.
oui : la bande n'a pas été ouverte puisqu'il s'agit du premier enregistrement. On n'a donc pas besoin de la fermer. On se branche en PL0.

PL0 On avait commencé à lire le fichier structure. Il faut donc fermer la bande contenant les structures. On arrête l'exécution du programme.

Remarque : Cas où il y a eu le message " TAILLE STRUCTURE INSUFFISANTE ".
On résout le problème de la même manière que pour le fichier structure (étiquette 2).

PLVAL

On lit la table structure dans la zone STRUCT
On ferme la bande structure
On calcule la taille de la structure en mots et on la range dans EXPLØ + 18
On se branche en PLL.

PLL On initialise tab à 2 (on saute la taille de l'enregistrement et celle de la table que l'on remplira à la fin).
On initialise mem à 0, SUITE à 1, et PØST à 0.
On range le type de carte lu sur la carte dans T1. En effet, on conserve le type de carte pour faire des vérifications lorsqu'il y a des cartes "suite".
La place occupée par type de carte sur la carte est de 2 octets.
On incrémente donc tamp de 2
tamp = tamp + 2
On prend dans EXPLØ la taille du numéro d'article et on range le numéro pris sur la carte dans NUM.
On conserve le numéro de classement dans NUM car sur toutes les cartes formant l'article, on répète cenuméro à titre de vérification.
La place occupée par le numéro d'article est donnée par "taille n° d'article"; on incrémente donc tamp de cette taille.
tamp = tamp + taille n° d'article.
test : type d'enregistrement de TAMP = T1 ?
Il s'agit de la première carte où le type d'enregistrement doit-être égal au type de carte (cf chapitres précédents).
non : erreur 1.

FAUX 1 On range le numéro d'erreur dans le 5ème mot de la zone ERRØR.
On range le type de carte dans les 2 premiers octets du 2ème mot de ERRØR.
On imprime ERRØR.
test : P1 = 0 ?
non : on a déjà écrit un enregistrement, on ferme la bande. On se branche en P1.
oui : on n'a pas besoin de fermer la bande qui n'a pas été ouverte. On se branche en P1.

P1 On arrête l'exécution du programme.

oui : la place occupée par le type d'enregistrement est de 2 octets. On incrémente donc tamp de 2.
tamp = tamp + 2
On prend la taille du n° d'article dans EXPLØ.

.../...

test : NUM = numéro de classement après type ?
non : erreur 2 et branchement en FAUTE1.
oui : sur la première carte le numéro de classement est présent 2 fois à titre de vérification. Seul le second sera présent dans l'enregistrement;
test : TC = STRUCT, PØSI ?
Le noeud de niveau 1 étant obligatoire, le type de noeud se trouvant sur la carte doit donc être le premier de la table structure.
non : erreur 3 et branchement en FAUTE1.
oui : on remplit la zone TAB.
On range TE dans le premier mot, on incrémente donc tab de 1.
tab = tab + 1
La deuxième zone de TAB contient la position du noeud par rapport à l'origine de l'enregistrement , cette position est donnée par le pointeur de MEM : mem.
On range donc mem dans le 2ème mot de TAB et on incrémente tab de 1.
tab = tab + 1.
Dans la table structure, on se positionne devant la taille de l'information ; on incrémente donc PØSI de 3 : (on saute type de carte, type de noeud, et nom information).
PØSI = PØSI + 3
On se branche en PL4.

PL4 test : 1er caractère de STRUCT = P ?
non : le noeud de niveau 1 ne peut pas être répétitif.
Erreur 5 et branchement en FAUTE1.
oui : test : caractère suivant de TAMP = ∅ ?
On regarde si l'on est arrivé à la fin de la carte.
oui : branchement en PL3
non : test tamp > 80 ?
oui : on a oublié de préciser qu'il y avait des cartes "suite". Erreur 6 et branchement en FAUTE1.
non : BTI22 On se place dans la table structure devant le thésaurus. Pour cela on saute taille information, position / origine du noeud, longueur U1 si répétitive.
PØSI = PØSI + 3
test : thésaurus = 0 ?
oui : on se branche en MV1
non : on positionne th à 1, cela nous permettra après avoir consulté le thésaurus de reprendre le programme là où on l'avait laissé.
On se branche en th.eus.

.../...

ETI On retourne devant la taille de l'information
PØSI = PØSI - 3

ETI23 On prend les 3 caractères représentant la longueur propre dans STRUCT. On range cette longueur propre prise dans TAMP dans MEM grâce aux pointeurs tamp et mem.
On incrémente tamp et mem de cette longueur propre.
tamp = tamp + longueur propre.
mem = mem + longueur propre
On se branche en ETI1.

ETI1 On se positionne devant le type de carte dans STRUCT on saute donc taille information, position / origine, longueur UI si répétitive et thésaurus.
PØSI = PØSI + 4
test : type de carte de STRUCT = 0 ?
oui : on n'a pas fini de ranger les informations concernant le noeud en cours de traitement. On va ranger la suivante. Pour cela, on se positionne devant la taille de l'information.
PØSI = PØSI + 3
on retourne en PL4.

PL3 On a une carte "suite".
On lit cette carte dans TAMP.
On initialise tamp à 0;
test : type de carte de TAMP = TC ?
non : on n'est plus dans le même noeud.
Erreur ? et branchement en FAUTE11.
oui : la place occupée par type de carte est de 2 octets, on incrémente donc tamp de 2.
tamp = tamp + 2
On prend la taille du numéro d'article dans EXPLØ.
test : Numéro de classement de TAMP = NUM ?
non : il y a peut-être eu une intervertion de cartes.
Erreur 8 et branchement en FAUTE1.
oui : on incrémente tamp de la place occupée par le numéro d'article.
tamp = tamp + taille n° d'article.
on incrémente SUITE de 1.
SUITE = SUITE + 1
On prend taille nb de dans EXPLØ.
test : suite = SUITE ?
non : il y a eu intervertion dans les cartes "suite".
Erreur 9 et branchement en FAUTE1.
oui : on incrémente tamp de la place occupée par suite sur la carte.
tamp = tamp + taille nb de.
On se branche en ETI22.

.../...

ETI6 On initialise SUITE à 1
On lit une carte dans TAMP.
test : 1er caractère = ✳ ?
oui : on a dit que pour délimiter 2 articles on mettait sur le premier caractère de la première carte une étoile "✳ ". L'article que l'on traitait est terminé. On va donc le ranger sur bande en se branchant en FIN.
non : on initialise tamp à 0 et on se branche en PL6.

PL6 test : type de carte de TAMP = type de carte de STRUCT ?
oui : on se branche en PL5.
non : le noeud sur lequel on est positionné dans la structure n'est pas présent dans l'article.
On passe donc au suivant puisqu'on a vu que certains pouvaient être absents.
Pour cela on saute tout ce qui concerne le noeud dans la structure en incrémentant PØSI de 7.
PØSI = PØSI + 7
test : PØSI ⩾ EXPLØ + 18 ?
oui : le pointeur de STRUCT devient supérieur à la taille de la table structure. Il y a donc une erreur : le type de carte est faux ou peut-être est-il mal placé (nous avons vu que l'on ne pouvait remonter dans la structure). Erreur 10 et branchement en FAUTE1.
non : on retourne en PL6.

PL5 On range le nouveau type de carte dans TC. La place qu'il occupe sur la carte est de 2 octets, on incrémente donc tamp de 2.
tamp = tamp + 2
On saute le type de carte dans la table structure en incrémentant PØSI de 1.
PØSI = PØSI + 1
On prend le type de noeud dans la structure et on le range en TN. La place qu'il occupe est de 1 mot.
PØSI = PØSI + 1
On prend la taille du numéro d'article dans EXPLØ.
test : numéro de classement = NUM ?
non : erreur 12 et branchement en FAUTE1.
oui : on incrémente tamp de la taille du numéro d'article.
tamp = tamp + taille n° article.
test : nom information = FIXE ?
Il s'agit d'un noeud de niveau 2 où le nom de l'information est remplacé par FIXE ou REPE.
oui : le noeud de niveau 2 est "unique", on met donc 0 dans NUL2 et on se branche en PL7.
non : test : nom information = REPE ?
non : erreur 11 et branchement en FAUTE1.
oui : le noeud de niveau 2 est "répétitif".

.../...

On prend taille nb de dans EXPLØ.
On va décoder l'"information de service" et la ranger en MUL2.
On incrémente tamp de la place occupée par cette information.
tamp = tamp + taille nb de.
Puisqu'il s'agit d'un noeud "répétitif", il faut garder l'adresse où débute ce noeud dans la structure. On se place donc devant le type de carte : PØSI = PØSI - 2.
On range l'adresse dans PØ2.
On se remet devant le nom de l'information.
PØSI = PØSI + 2
On initialise l'indice de ces noeuds NØ2 à 1.
On se branche en PL7.

PL7 On prend taille type de N2 dans EXPLØ.
test : type de N2 de TAMP = TN ?
non : au niveau 3 on peut avoir plusieurs types de noeuds qui correspondent au même type de carte mais pas au niveau 2.
Erreur 26 et branchement en FAUCE1.
oui : on incrémente tamp de la place occupée par type de N2.
tamp = tamp + taille type de N2.
On range le type de noeud dans TAB grâce au pointeur tab. On incrémente ce pointeur de 1 puisque chaque information dans TAB tient sur 1 mot.
tab = tab + 1
On range la position par rapport à l'origine de l'enregistrement: mem, dans tab et on incrémente tab de 1.
tab = tab + 1
Dans la table structure, on se positionne devant le type de carte suivant.
PØSI = PØSI + 5
test : MUL2 = 0 ?
non : si le noeud de niveau 2 est "répétitif" il est forcement suivi de noeuds de niveau 3 puisque pour remplir MUL2 nous avons pris l'"information de service" précédant le type de noeud (cf chapitres précédents). On se branche en ET13.
oui : test : type de carte de STRUC2 = 0 ?
non : le noeud de niveau 2 est suivi de noeuds de niveau 3. On se branche en ET13.
oui : les informations élémentaires se trouvent au niveau 2. Nous avons vu dans les chapitres précédents : lorsqu'un noeud de niveau 2 est "unique", par analogie avec le niveau 3, l'unité d'informations peut-être "unique" ou "répétitive".
On initialise NN3 à 0 puisqu'il n'y a pas de noeuds de niveau 3
On se place dans la structure devant la taille de l'information.

.../.

PØSI = PØSI + 3
On se branche en ETI7.

ETI7 test : 1er caractère de STRUCT = "F" ?
non : l'unité d'informations est "répétitive" et on se branche en ETI5.
oui : l'unité d'informations est "unique". On initialise MUL4 à 0.
On remplit la zone multiplicité de TAB de 0 et on incrémente tab de 1.
tab = tab + 1
test:tab = V1 ?
non : il reste encore de la place dans TAB . On se branche en ETI3.
oui : PL8 : On imprime le message "TAILLE TAB INSUFFISANTE". On résoud ce problème d'une façon identique à "TAILLE STRUCTURE INSUFFISANTE" (cf remarque après étiquette F10)

F20 test : P1 = 0 ?
non : on a déjà écrit un enregistrement sur bande. Il faut donc la fermer. On se branche en F11.
oui : F11 : on arrête l'exécution du programme.

ETI5 Le noeud de niveau 2 est alors traité comme un noeud de niveau 3. Il est "répétitif", on garde donc l'adresse de la taille de la première information du noeud dans PØ3.
En effet, en cas de répétition à ce niveau les unités d'informations se succèdent, on ne répète ni le type de carte, ni le type de noeud à chaque unité. L'"information de service" se trouve après le type de noeud. On prend la taille nb de dans EXPLØ. On décode le nombre de répétitions et on le range dans MUL4.
On incrémente tamp de taille nb de.
tamp = tamp + taille nb de.
On range MUL4 dans la zone multiplicité de TAB.
tab = tab + 1
test : tab = V1 ?
On fait ce test chaque fois que l'on a rempli toutes les zones de TAB concernant un noeud.
oui : on se branche en PL8.
non : on va ranger une unité d'informations. On décrémente MUL4 de 1 et on se branche en ETI3.

ETI3 Caractère suivant de TAMP = ∅ ?
oui : on se branche en ETI11 pour traiter une carte "suite"
non : test : tamp > 80 ?
oui : on a oublié de préciser qu'il y avait une carte "suite". Erreur 13 et branchement en FAUTE 1.

.../...

non : [ETI21] On va ranger une information. On se place devant le thésaurus dans la structure. PØSI = PØSI + 3
test : thésaurus = 0 ?
non : onva alors consulter un thésaurus avant de ranger l'information.
oui : on va ranger l'information. Pour cela on se replace devant la taille de l'information.
PØSI = PØSI - 3
On se branche en ETI2.

[ETI11] On lit une carte dans TAMP.
On initialise tamp à 0;
test : type de carte de TAMP = ?
non : la carte "suite" ne comporte pas le même type de carte que celui qui correspond au noeud en cours de traitement. Erreur 14 et branchement en FAUTE1.
oui : la place occupée par type de carte est de 2 octets.
tamp = tamp + 2
On prend la taille du numéro d'article dans EXPLØ.
test : numéro de classement = NUM ?
non : erreur 15 et branchement en FAUTE1.
oui : on incrémente tamp de la place occupée par le numéro de classement.
tamp = tamp + taille n° article.
On incrémente SUITE de 1.
SUITE = SUITE + 1.
On prend taille nb de dans EXPLØ.
test : suite de TAMP = SUITE ?
non : erreur 16 et branchement en FAUTE1.
oui : on incrémente tamp de la place occupée par suite sur la carte.
tamp = tamp + taille nb de.
test : VAR1 = 1 ?
Cet indicateur permet de revenir là où on avait laissé le programme.
oui : on se branche en ETI10
non : on se branche en ETI21.

[ETI2] On prend 3 caractères de TAMP : longueur propre.
test : longueur propre = "-"?
non : on range la longueur propre prise dans TAMP dans MEM.
On incrémente les 2 pointeurs tamp et mem de la taille de l'information.
tamp = tamp + longueur propre
mem = mem + longueur propre.
On se branche en ETI12.
oui : [PLS] On initialise VAR1 à 1. VAR1 est l'indicateur de retour en cas de carte "suite".

.../...

Il s'agit d'une information de taille variable.
On initialise VAR à 0. VAR sert à compter les caractères qu'occupe cette information sur chaque carte. On se branche en ETI10.

ETI10 On range tamp dans tampl.

PL12 On lit le caractère suivant de TAMP
test : caractère = "*" ?
oui : l'information se termine par une "*".
On connaît donc sa longueur et on se branche en PL10.
non : test : caractère = ß?.
oui : on se branche en PL11
non : test : tampl > 80 ?
oui : on n'a pas précisé qu'il y avait une carte "suite". Erreur 17 et branchement en FAUTE1.
non : on passe au caractère suivant.
On incrémente tamp et la taille de l'information de 1.
tamp = tamp + 1.
VAR = VAR + 1
On retourne en PL12.

PL11 test : P2 = 0 ?
non : on a déjà eu une carte "suite". On se branche en PL13.
oui : la longueur de l'information est rangée dans un mot qui précède l'information elle-même. On initialise donc cette zone dans MEM à 0.
On met P2 à 1 pour signaler qu'il y a eu une carte "suite". On mémorise le pointeur mem de MEM dans meml pour pouvoir remplir la zone : longueur de l'information. On saute cette longueur.
mem = mem + 4
on se branche en PL13.

PL13 On a terminé une carte. On va donc ranger la partie de l'information se trouvant sur cette carte.
On ajoute VAR à longueur dans MEM grâce au pointeur meml, puis on range le nombre de caractères VAR pris dans TAMP dans MEM.
On incrémente le pointeur de MEM de VAR.
mem = mem + VAR.
On initialise VAR à 0 et on va lire la suite de l'information sur une carte "suite". On se branche en ETI11.

PL10 test : P2 = 0 ?
oui : il n'y a pas eu de carte "suite" et on a lu toute l'information. On range donc VAR dans MEM et on incrémente mem de 4.
mem = mem + 4
On se branche en PL14.
non : PL14 On remet P2 à 0. Il y a déjà eu des cartes "suite", on a donc déjà rangé une partie de l'information. On ajoute VAR à MEM grâce au pointeur meml. On se branche en PL15.

.../...

PL15 On range ou toute l'information si elle tenait sur une carte, ou la fin s'il y a eu des cartes "suite". On prend le nombre de caractères VAR dans TAMP et on le met dans MEM.
On incrémente mem de VAR et tamp de VAR + 1, puisque le caractère "*" signifiant la fin de l'information ne se retrouve pas dans MEM.
mem = mem + VAR
tamp = tamp + VAR+ 1
On initialise VAR1 à 0. On se branche en ETI12.

ETI12 Test : mem > M1 ?
non : on se branche en ETI13.
oui : on imprime le message "TAILLE ZØNE DE RANGEMENT: MEM INSUFFISANTE". On se branche en F20. On résout le problème de la zone MEM insuffisante d'une façon identique à "TAILLE STRUCTURE INSUFFISANTE" (cf remarque après étiquette F10).

ETI13 On se place devant le type de carte suivant dans la structure.
PØSI = PØSI + 4.
test : PØSI > EXPLØ + 18 ?
oui : on est en train de traiter le dernier noeud de la structure et on a atteint la dernière information. on se branche en FN1.
non : test : type de carte de STRUCT = 0 ?

oui : il reste des informations appartenant au noeud traité à ranger. On se place devant la taille de l'information.
PØSI = PØSI + 3
On retourne en ETI3.
non : FN1 test : NUL4 = 0 ?
non : on a rangé toutes les informations d'une unité, on va ranger celles de la suivante. Pour cela on se place devant la taille dela première information du même noeud dans la table structure. On range l'adresse de cette information dans le pointeur PØSI de STRUCT.
On décrémente NUL4 de 1 et on retourne en ETI3.
oui : s'il s'agissait d'un noeud "répétitif", on a rangé toutes les unités. S'il s'agissait d'un noeud "unique," on a finide ranger l'unité. On regarde s'il y a d'autres noeuds de niveau 3 à ranger dans le noeud de niveau 2 traité.
test : NUL3 = 0 ?
non : il reste des noeuds de niveau 3. On décrémente NUL3 de 1 et on retourne en ETI1.
oui : on a rangé tous les noeuds de niveau 3 appartenant au noeud de niveau 2 traité. On regarde si ce noeud de niveau 2 est "répétitif".

.../...

test : NIV2 = 1 ou 0 ?
oui : soit le noeud de niveau 2 n'est pas "répétitif", soit s'il est "répétitif", on a fini de le traiter. On passe au noeud de niveau 2 suivant en se branchant en [illegible].
non : le noeud est "répétitif". On se place dans la table structure devant le type de carte correspondant à ce type de noeud. Pour cela on range l'adresse du type de carte dans le pointeur PØSI de la table structure. on décrémente NIV2 de 1.
On initialise SUITE à 1.
On lit une carte dans TAMP.
On initialise tamp à 0.
test : type de carte de TAMP = type de carte de STRUCT ?
non : erreur 18 et branchement en FAUTE1.
oui : on range le type de carte dans T.
On incrémente tamp de la place occupée par ce type de carte.
tamp = tamp + 2
On se place devant le type de noeud dans la structure.
PØSI = PØSI + 1
On prend la taille du n° d'article dans EXPLØ.
test : numéro de classement = NUM ?
non : erreur 19 et branchement en FAUTE1.
oui : on incrémente tamp de la place occupée par le numéro de classement.
tamp = tamp + taille n° article.
On range le type de noeud de STRUCT dans TM.
On prend la taille nb de dans EXPLØ.
On passe à l'indice suivant du noeud de niveau 2, en incrémentant NØC de 1.
test : numéro de couche de TAMP = NØC ?
non : il y a eu intervertion dans les indices. Erreur 20 et branchement en FAUTE1.
oui : on incrémente tamp de la place occupée par ce numéro de couche.
tamp = tamp + taille nb de.
On prend la taille de type de N2 dans EXPLØ.
test: type de N2 de TAMP = TM ?
non : erreur 21 et branchement en FAUTE1.
oui : on incrémente tamp de la place occupée par type de N2.
tamp = tamp + taille type de N2.
On range le type de noeud dans TAB et on incrémente tab de 1
tab = tab + 1
On range l'adresse d'origine du noeud: mem dans TAB et on incrémente tab de 1.

.../...

tab = tab + 1
On se place devant le type de carte suivant
PØSI = PØSI + 6
On se branche en ETI0.

ETI6 On prend taille nb de dans EXPLØ.
On décode le nombre de noeuds de niveau 3 présents dans le noeud de niveau 2 et on le range en NUL3. On incrémente tamp de la place occupée par ce nombre.
tamp = tamp + taille nb de.
On décrémente NUL3 de 1.
On range NUL2 dans la zone multiplicité de TAB et on incrémente tab de 1.
tab = tab + 1.
test : tab = V1 ?
oui : on se branche en PLS car il n'y a plus de place dans TAB.
non : on prend la taille type de N3 dans EXPLØ
test : type de N3 = 0 ?
non : on traite le noeud de niveau 3 en se branchant en ETI0.
oui : sur une même carte on peut mettre un noeud de niveau 3 et un noeud de niveau 2.
Le premier noeud de niveau 3 est absent, on passe alors au suivant en se branchant en ETI6.

ETI5 On initialise SUITE à 1.
On lit une carte dans TAMP.
On initialise tamp à 0, on se branche en PLL5.

PLL5 test : type de carte de TAMP = type de carte de STRUCT ?
oui : on se branche en PLL7.
non : certains noeuds de niveau 3 pouvant être absents de l'article, on cherche celui qui est présent en passant au type de carte suivant dans la structure.
PØSI = PØSI + 7
test : PØSI > EXPLØ + 12 ?
non : on retourne en PLL5
oui : on n'a pas trouvé le type de carte. Erreur 22 et branchement en FAUX 1.

PLL7 On range le type de carte dans T.
On incrémente tamp de la place occupée par ce type de carte.
tamp = tamp + 2
On prend la taille du numéro d'article dans EXPLØ.
test : numéro declassement = NUL ?
non : erreur 23 et branchement en FAUX 1.
oui : on incrémente tamp de la place occupée par le numéro d'article.
tamp = tamp + taille n° d'article.
On se branche en ETI8.

ETI8 On se place devant le type de noeud de la structure.
PØSI = PØSI + 1
On prend la taille de type de N3 dans EXPLØ.

.../...

test : type de K3 de tAMP = type de K3 de STRUCT?
oui : on se branche en AT1.
non : AT2 sur un même type de carte on peut avoir plusieurs types de K3.
On passe au type de carte suivant.
test : type de carte de STRUCT = 0 ?
oui : on est toujours dans le même type de noeud. On passe au suivant en incrémentant PØSI de 1 et en se branchant en AT2.
non : test : type de carte de STRUCT = K ?
oui : on va regarder si le type de noeud est celui que l'on cherche en se branchant en ETIQ.
non : on passe au type de carte suivant, le type de noeud doit donc être faux. Erreur 2 ! et branchement en FAUTE1.

AT1 On a trouvé le type de noeud.
On incrémente tamp de la place occupée par ce type de noeud.
tamp = tamp + taille type de K3.
On range le type de noeud dans T .
On se place devant la taille de la première information dans STRUCT.
PØSI = PØSI + 2
On range le type de noeud dans TAB et on incrémente tab de 1.
tab = tab + 1
On range l'adresse d'origine du noeud : mem dans TAB et on incrémente tab de 1.
tab = tab + 1
On se branche en ETIQ.

theus
test : BID = 0 ?
non : on se branche en TH1.
oui : on n'a pas encore consulté de thésaurus.
On va donc mettre le fichier BIBLIØ en mémoire. On ouvre le fichier structure.
test : BIBLIØ existe ?
non : on se branche en F1. On ne peut pas consulter de thésaurus s'il n'y a pas de fichier BIBLIØ.
oui : on lit les 2 blocs du fichier BIBLIØ dans DIC.
On ferme le fichier structure et on se branche en TH1.

TH1 On range le numéro de thésaurus dans N. On va chercher l'adresse de ce thésaurus grâce à N dans DIC et on la range dans dic.
On range dic dans Z.
test : nom de thésaurus = PRES ?
non : on se branche en TH2.
oui : on incrémente dic de la place occupée par PRES.
dic = dic + 4
On range la "place libre restante" dans Y et on incrémente dic de la place qu'elle ocupe.

.../...

dic = dic + 4
On range la taille de l'information dans X et on incrémente dic de la place occupée par cette taille.
dic = dic +3
On range le nombre d'informations présentes dans le thésaurus dans I et on incrémente dic de la place occupée par ce nombre.
dic = dic + 3
On se branche en ETI15.

ETI15 test : information de TAMP = information de DIC ?
oui : on se branche en ETI14.
non : on décrémente le nombre I de 1.
test : I = 0 ?
oui : on n'a pas trouvé l'information.
erreur 27 et branchement en FAUTE1.
non : on passe à l'information suivante dans DIC en incrémentant dic de la place occupée par une information.
dic = dic + X
test : 1276 - (dic - Z) < Y ?
la taille d'un dictionnaire est 1276 octets.
Z représente l'adresse d'origine du dictionnaire.
non : on n'a pas encore lu toutes les informations du dictionnaire. On retourne en ETI15.
oui : on alu toutes les informations d'un dictionnaire. On passe au suivant.
test : 1er caractère de V = "*" ?
non : il aurait dû y avoir un débordement.
Erreur 28 et branchement en FAUTE1.
oui : on prend les 2 derniers caractères de V et on a le numéro du dictionnaire sur lequel on a débordé. On va chercher l'adresse de ce dictionnaire que l'on range en dic puis en Z.
On range le premier mot du nouveau dictionnaire dans V et on incrémente dic de la taille de cette "place libre restante".
dic = dic + 4.
On retourne en ETI15.

ETI14 On a trouvé une information du dictionnaire identique à l'information de la carte. On se place dans la structure devant la taille de cette information pour la ranger dans MEM.
PØSI = PØSI - 3.
test : th = 1 ?
indicateur qui permet de se brancher là où on avait laissé le programme.
oui : on initialise NUM à 1 et th à 0.
On se branche en ETI23.
non : FN3 On initialise NUM à 1 et on se branche en ETI2.

.../...

TM2 test : nom = CØDE ?
non :il s'agit d'un thésaurus de décodage qui nous ne intéresse pas pour le moment.
On retourne au programme en se branchant en ETIL4.
oui : on incrémente dic de la place occupée par CØDE.
dic = dic + 4
On range la "place libre restante" dans Y et on incrémente dic de la place occupée par elle.
dic = dic + 4
On range la taille externe de l'information dans X et on incrémente dic de la place occupée par cette taille.
dic = dic + 3
On range la taille interne de l'information dans Y et on incrémente dic de la place occupée par cette taille
dic = dic + 3
On range le nombre d'informations présentes dans le thésaurus dans I et on incrémente dic de la place occupée par cette taille.
dic = dic + 3
On se branche en ETIL5.

ETIL5 test : information de TAMP = information de DIC ?
oui : on se branche en TM4.
non : on décrémente I de 1.
test : I = 0 ?
oui : on n'a pas trouvé l'information dans le dictionnaire. Erreur 29 et branchement en FAUTE1.
non : on passe à l'information suivante.
On incrémente dic de la place occupée par les tailles externe et interne de l'information.
dic = dic + X + Y
test : 1276-(dic-Z) < X + Y ?
On regarde s'il reste des informations dans le dictionnaire.
non : on n'a pas encore lu toutes les informations du dictionnaire
On retourne en ETIL5.
oui : test : 1er caractère de V = "*"?
non : il aurait dû y avoir débordement . Erreur 29 et branchement en FAUTE1.
oui : on prend les 2 derniers caractères de V et on a le numéro du dictionnaire sur lequel on a débordé. On va chercher l'adresse du dictionnaire que l'on range en dic et en Z.
On range le premier mot du nouveau dictionnaire dans V et on incrémente dic de la taille de cette "place libre restante".
dic = dic + 4
On retourne en ETIL5.

TM4 On a trouvé l'information du dictionnaire identique à celle de la carte.

.../...

On incrémente dic de la taille externe de l'information. On range le code del'information dans MEM.
On incrémente mem de la place occupée par le code et tamp de la place occupée par l'information.
mem = mem + Y
tamp = tamp + X
On se place devant la taille de l'information dans STRSCI.
PØSI = PØSI - 3
On retourne dans le programme.
test : th = 1 ?
oui : on initialise BIB à 1 et th à 0. On va en ETI1.
non : $\boxed{\text{TH5}}$ On initialise BIB à 1 et on va en ETI12.

$\boxed{\text{FIN}}$ On range la taille de l'enregistrement, c'est-à-dire la taille de l'article + la taille de la table dans le premier mot de TAB.
On range la taille de la table dans le deuxième mot de TAB.
La longueur qui se trouvait dans TAB dans la zone position par rapport à l'origine était la position par rapport à l'origine del'article. Pour avoir la position par rapport à l'origine de l'enregistrement, on ajoute à chaque niveau la taille de la table.
test : tab = VI ?
oui : on se branche en F2.
non : il reste de la place entre TAB et MEM. On va donc ranger MEM à la suite de TAB. On se branche en F2.

$\boxed{\text{F2}}$ test : P1 = 0 ?
non : on se branche en F3.
oui : on traite le premier enregistrement. on positionne INDIC à 0.
test : la bande est vierge ?
oui : on crée le label et on se positionne au début de la bande.
non : il y a déjà eu des enregistrements écrits mais lors d'une exécution précédente du programme.
On se positionne derrière le dernier enregistrement.
On se branche en F3.

$\boxed{\text{F3}}$ On positionne INDIC à 1.
On écrit l'enregistrement sur bande.
On positionne P1 à 1.
test : P = 1 ?
non : il reste des cartes à lire, on va donc créer un nouvel enregistrement et on se branche en DEB1.
oui : il n'y a plus de cartes. On ferme le fichier maître et on arrête l'exécution du programme.

.../...

FIN : C'est une séquence de déroutement lorsqu'il n'y a plus de cartes sur le lecteur de cartes.
On positionne [illegible] à 1 et on va écrire le dernier enregistrement sur bande en se branchant en FIN.

FINTRU : C'est une séquence de déroutement qui correspond à la bande structure.
test : premier octet de R[illegible] = X'1[illegible] ?
non : erreur 4 et branchement en FAUTE1.
oui : F5 On ferme le fichier structure. On n'a pas trouvé le fichier cherché. On imprime le message "TYPE D'ENREGISTREMENT FAUX". On arrête l'exécution du programme.

FINBAN : C'est une séquence de déroutement qui correspond à la bande fichier maître.
test : premier octet de R[illegible] = X'02' ?
non : erreur 25 et branchement en FAUTE1.
oui : F5 test : NUDIC = 0 ?
oui : la bande est vierge. On la ferme et on l'ouvre de nouveau en création. On se branche en F3.
non : on imprime le message : "LA BANDE EST PLEINE".
On ferme le fichier maître et on arrête l'exécution du programme.

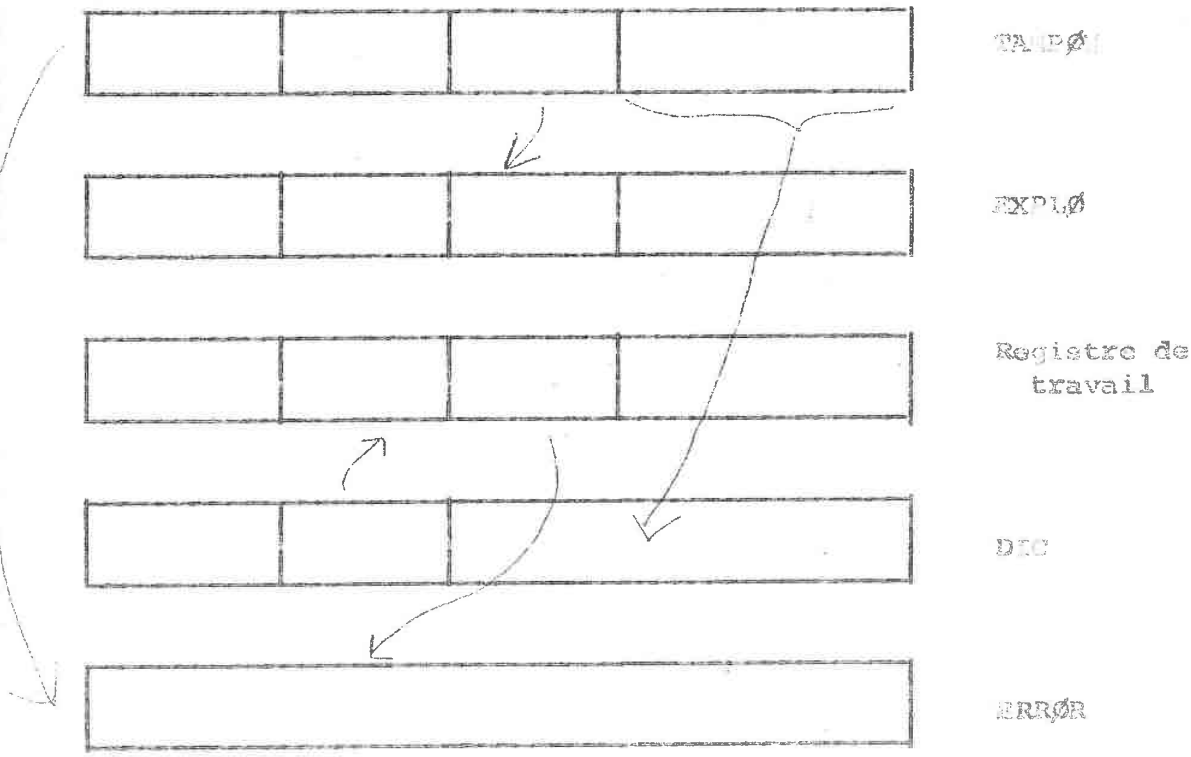
.../...

4^3 mise à jour des dictionnaires par addition d'informations

43 - Mise à jour des dictionnaires par addition d'informations :

431 - Présentation des éléments utilisés pour réaliser la mise à jour :

4311 - Schéma du travail :



.../...

Ces zones dont on a réservé la place en mémoire nous aident à mettre à jour le fichier des dictionnaires que nous avons appelé BIBLIØ.

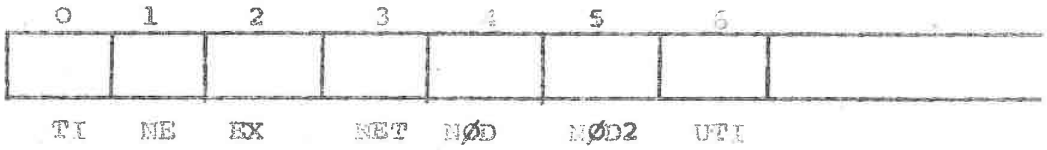
4312 - Zones de travail :

Précisons la taille et l'utilisation des zones précédemment décrites et décrivons les indicateurs ou zones supplémentaires pour cette mise à jour.

- la zone TAMPØN de 20 mots : même usage que dans le fichier structure.

- la zone EXPLØ de 7 mots ;

Voici cette zone.



Décrivons chaque élément de cette zone en les repérant par rapport à l'origine.

. EXPLØ : TI.

Pour le dictionnaire PRES c'est la taille de l'information.
Pour les dictionnaires CØDE et DECØ c'est la somme taille externe et taille interne de l'information.

. EXPLØ + 1 : ME

C'est le nombre d'informations présentes dans le dictionnaire à mettre à jour.

. EXPLØ + 2 : EX.

Dans l'organigramme nous avons appelé X "la place libre restante" dans le dictionnaire. EX représente l'adresse où se trouve X.

. EXPLØ + 3 : NET.

C'est le nombre d'informations à ajouter au dictionnaire en mise à jour.

. EXPLØ + 4 : NØD.

C'est le numéro du dictionnaire à mettre à jour.

. EXPLØ + 5 : NØD2.

C'est le dernier numéro de dictionnaire attribué au dictionnaire en cours de traitement lorsqu'il y a déjà eu des mises à jour.

. EXPLØ + 6 : UTI.

C'est un indicateur qui sert pour l'impression des numéros de dictionnaires attribués au cours d'une mise à jour.

- les registres de travail :

On a donné à certains registres un rôle particulier :

. le registre 3 appelé tamp sert de pointeur dans la zone TAMP : c'est un pointeur d'octets.

. le registre 5 appelé longdic sert de pointeur dans la zone DIC c'est un pointeur d'octets.

.../...

. le registre 7 appelé A donne le nombre d'informations que l'on peut encore ranger dans le dictionnaire en mise à jour sans être obligé d'attribuer un nouvel emplacement.
. le registre 12 appelé X contient la zone que nous avons intitulée "place libre restante".
. le registre 13 appelé SUITE sert de compteur pour vérifier la succession des cartes lorsqu'il y a des cartes "suite".
- la zone DIC de 12000 mots :
Elle contient le fichier DIBLIØ pour lequel on fait une mise à jour.
- la zone ERRØR de 6 mots :
.. (1) .... ERREUR .... (2)
Même utilisation que dans le fichier structure.
- la zone DICØ de 14 mots :
"IL N'Y A PLUS DE PLACE DANS LE FICHIER DICTIØNNAIRE".
On sort ce message lorsque tous les dictionnaires ont été utilisés.
- la zone IMP1 de 2 mots :
"ERREUR".
On sort ce message lorsqu'on n'a pas trouvé le fichier DIBLIØ. En effet, on ne peut faire une mise à jour sur un fichier qui n'existe pas.
- la zone IMP3 de 7 mots :
"DICTIØNNAIRES UTILISÉS".
Impression du titre.
- la zone IMP2 de 2 mots :
Sous le titre précédent, on imprime les nouveaux numéros de dictionnaires attribués lorsqu'il y a eu débordement.

313 - Quelques particularités du programme :

- On a une zone en mémoire qui sert à faire des comparaisons : PRES qui contient le mot "PRES".
- On a une zone TRAD1 : même utilisation que dans le fichier structure.
- Le principe du rangement de chaque information est le même que celui qui a été décrit dans le fichier structure.
Il s'agit de l'instruction MES. Les adresses contenues dans les registres 11 et 15 diffèrent: pour 11 on aura l'adresse du tampon et pour 15 l'adresse du dictionnaire.
- les incrémentations ou décrémentations des différents registres se présentent comme pour le fichier structure.
exemple : tamp = tamp + 3

32 - Organigramme détaillé :

Il faut se servir de l'organigramme pour suivre les explications.

.../...

DEB On ouvre les fichiers lecteur de cartes, imprimante et structure.
On cherche sur la bande contenant les fichiers structures le fichier BIBLIØ.
S'il est absent on se branche en FINVØL et on imprime IMP1 (cf paragraphe 4312).
S'il est présent on lit le fichier BIBLIØ dans la zone DIC puis on ferme la bande structure. On met l'indicateur TST à 0.

DEB1 On lit une carte dans la zone TAMP. On initialise les 3 premiers mots de la zone EXPLØ à 0. On initialise SUITE à 1 et tamp à 0.
On décode le numéro de dictionnaire que l'on a lu sur la carte et on le range en dic, en NØD et en NØD2. On décrémente dic de 1, nous avons vu pourquoi à l'étiquette AR13 du programme fichier structure.
dic = dic - 1
La place occupée par le numéro de dictionnaire sur la carte est de 2 caractères, on incrémente donc tamp de 2.
tamp = tamp + 2
On range dans longdic l'adresse du dictionnaire que l'on veut mettre à jour.
test : nom de TAMP : nom de DIC ?
non : le dictionnaire à mettre à jour et le dictionnaire existant ne portant pas le même nom PRES, CØDE ou DECØ il doit y avoir une erreur dans le numéro de dictionnaire sur la carte. Erreur 1 et branchement en FAUTE.

FAUTE On range le registre contenant le numéro de l'erreur dans le 3ème mot de la zone ERRØR. On range le numéro du dictionnaire faux qui se trouve alors dans TAMP dans le 2ème mot de la zone ERRØR.
On imprime la zone ERRØR et on arrête l'exécution du programme.
oui : la place occupée par le nom du dictionnaire dans DIC est de 4 octets. On incrémente donc longdic de 4.
longdic = longdic + 4
On range le contenu de la "place libre restante" de DIC dans X et l'adresse de X dans EX. En effet, puisque l'on fait une mise à jour il faudra modifier cette zone ou bien en mettant l'adresse d'un nouveau dictionnaire en cas de débordement, ou bien en calculant à nouveau la "place libre restante".
La place occupée par X est de 4 octets. On incrémente donc longdic de 4.
longdic = longdic + 4
On prend la taille de l'information située dans DIC et on la range en TI.
La place occupée par celle-ci est de 3 octets, on incrémente donc longdic de 3.
longdic = longdic + 3

.../...

test : nom de TAMP = PRES ?
oui : on se branche en ETI1.
non : il s'agit d'un dictionnaire CØDE ou DECØ il faut donc ranger outre l'information elle-même, son code. La taille de l'ensemble est donc TI plus la taille interne de l'information. La place occupée par la taille interne dans DIC est de 3 octets. On incrémente donc longdic de 3 et on se branche en ETI1.

ETI1 On incrémente tamp de la place occupée par le nom du dictionnaire sur la carte, c'est-à-dire 4 octets.
tamp = tamp + 4
On range le nombre d'informations présentes dans le dictionnaire dans NE.
On décode le nombre d'informations à ajouter au dictionnaire et qui se trouve sur la carte. On le range en NET. La place occupée par ce nombre sur la carte est de 3 octets, on incrémente donc tamp de 3.
tamp = tamp + 3
On met à jour le nombre d'informations existant dans le dictionnaire, c'est-à-dire NE en ajoutant le nombre d'informations de la carte, c'est-à-dire NET.
Puis on range NE dans le dictionnaire et on se branche en ETI3.

ETI3 test : premier octet de X = "*" ?
oui : le dictionnaire a déjà été mis à jour et ayant débordé il a fallu lui attribuer un autre emplacement. Le numéro de dictionnaire se trouvant à cet emplacement, occupe les 2 derniers octets de X. On le range dans NØD2 : on en verra l'utilité à l'étiquette ETI4. On incrémente NØD2 de 1.
On le range également dans dic, ce qui va nous permettre de trouver l'adresse de ce nouveau dictionnaire dans DIC. On place cette adresse dans longdic : pointeur de DIC.
Le premier mot de ce nouveau dictionnaire représente la "place libre restante", on garde son adresse en EX pour le mettre à jour s'il reste de la place dans le dictionnaire.
Le contenu de ce mot est mis en X et on se branche en ETI3.
En effet, on va de nouveau regarder s'il n'y avait pas eu au cours d'une ou plusieurs autres mises à jour, d'autres dictionnaires attribués. On s'arrête au dernier dictionnaire où il reste de la place pour ranger les informations.
non : il reste de la place dans le dictionnaire pour ranger les informations et on se branche en ETI2.

ETI2 X < TI ?
On regarde si la "place libre restante" est suffisante pour ranger une ou plusieurs informations.
oui : la place libre est inférieure à la taille de l'information. Il va donc falloir attribuer un nouveau dictionnaire. Pour cela on se branche en ETI4.
non : il reste de la place pour ranger les informations. On se place derrière la dernière information rangée.

.../...

Pour cela on met à jour longdic. On va rechercher l'adresse d'origine du dictionnaire et on lui ajoute la place occupée par les informations dans le dictionnaire. Pour calculer cette place on a la taille maximum du dictionnaire 12?? octets de laquelle on enlève la "place libre restante" c'est-à-dire X.
On se branche en ETI.

ETI A représente le nombre d'informations que l'on peut encore ranger dans le dictionnaire.
A se calcule en faisant le rapport : $\frac{\text{"place libre restante"}}{\text{taille d'une information}}$

On se branche en ETII.

ETI1 On range une information de la carte dans le dictionnaire en se servant du pointeur de carte tamp et du pointeur de dictionnaire longdic. On connait la taille de cette information par TI.
On incrémente alors tamp et longdic de TI
tamp = tamp + TI
longdic = longdic + TI
La "place libre restante" diminue alors de la taille d'une information : X = X - TI.
On décrémente le nombre d'informations à ranger NBT de 1.
test : NBT = 0 ?
oui : on replace X, qui a été mis à jour pendant que l'on rangeait les informations, à son adresse ?X dans le dictionnaire.
On retourne en DEB1 pour mettre à jour un autre dictionnaire s'il y a lieu.
non : on se branche en ETI6.

ETI6 On décrémente A de 1 puisque l'on vient de ranger une information.
test : A = 0 ?
oui : on se branche en ETI4 pour attribuer un nouveau dictionnaire.
non : test : caractère suivant de TAMP = ∅ ?
oui : on se branche en ETI7
non : test : tamp > 80 ?
oui : on a oublié de préciser qu'il y avait des cartes "suite". Erreur 2 et branchement en FAUX.
non : sur la carte lue il y a une autre information à ranger. On retourne donc en ETI5.

ETI7 Il y a une carte "suite". On lit cette carte et on initialise tamp à 0.
tamp = 0.
On décode le numéro de dictionnaire sur la carte et on le compare avec le numéro de dictionnaire en cours de mise à jour.
test : numéro de dictionnaire = NØ ?

.../...

non : erreur 3 et on se branche en FAUTE.
oui : on range le numéro de dictionnaire dans dic et on le décrémente de 1.
La place occupée par ce numéro sur la carte est de 2 octets, on incrémente donc tamp de 2.
tamp = tamp + 2
Il s'agit d'une carte "suite", on incrémente donc SUITE de 1, et on le compare avec le numéro de séquence sur la carte une fois décodé.
test : suite = SUITE ?
non : il y a sans doute eu intervertion de cartes. Erreur 4 et branchement en FAUTE.
oui : la place occupée par le numéro de suite est de 2 octets, on incrémente donc tamp de 2.
tamp = tamp + 2
Pour ranger la première information de cette carte on se branche en ETI5.

ETI4 Le numéro de dictionnaire suivant le dernier dictionnaire attribué est NØD2, on le range en R6 qui nous sert d'indice.

ETI9 test : DIC, R6 = 0 ?
oui : on regarde si le dictionnaire suivant NØD2 est libre ou non. Si le contenu de son adresse est 0, il est libre et on se branche en ETI8.
non : le dictionnaire est déjà pris. On passe au suivant en incrémentant R6 de 1.
test : R6 = 50 ?
oui : le nombre maximum de dictionnaires possibles est 50. Ils sont tous pris. On sort alors le message: "IL N'Y A PLUS DE PLACE DANS LE FICHIER DICTIONNAIRE".
Il faudra alors réorganiser le fichier. On arrête l'exécution du programme.
non : on regarde si le dictionnaire suivant est libre ou non en retournant en ETI9.

ETI8 On range le numéro de dictionnaire attribué dans longdic, dans NØ qui nous permettra de ranger son adresse et dans NØD2. Toujours pour des raisons de langage, l'indexation étant inférieure de 1 au numéro réel du dictionnaire, on incrémente NØD2 de 1.
test : UTI = 0 ?
oui : il s'agit de la première attribution. On va alors imprimer le titre "DICTIONNAIRES UTILISES" et on se branche en TR1.
NON : TR1 on imprime le numéro du dictionnaire attribué après l'avoir décodé pour que l'utilisateur ne se serve pas du numéro pour entrer dans le fichier un nouveau dictionnaire.

.../...

On positionne UTI à 1, le titre ne devant être imprimé qu'une fois.
Dans R6 on a le numéro du dictionnaire attribué, on place dans le premier octet de R6 le caractère "*" et on range R6 dans DIC à l'adresse EX. On a alors formé un maillon de la chaine.
A l'endroit de la place libre restante on a le numéro du dictionnaire sur lequel on a débordé. On calcule l'adresse du nouveau dictionnaire : dans longdic on a son numéro et la place occupée par chaque dictionnaire est de 1276 octets. Cela nous fait longdic x 1276. On ajoute la place occupée par les 50 premiers mots de DIC où on a les adresses des 50 dictionnaires, c'est-à-dire 200 octets.
L'adresse est donc : longdic = (longdic x 1276) + 200
On range cette adresse grâce à NØ dans DIC et dans EX puisque la place libre restante d'adresse EX est le premier mot du dictionnaire. On saute la place libre restante dans le dictionnaire : longdic = longdic + 4.
Cette place libre restante est alors égale à 1276 - (taille du dictionnaire) - 4 (place occupée par elle-même).
On va ranger les informations dans ce dictionnaire en se branchant en ETI.

FINI Séquence de déroutement du lecteur de cartes.
On ouvre la bande. On réécrit le fichier BIBLIØ mis à jour sur l'ancien.
On ferme la bande et on arrête l'exécution du programme.

FINVØL Séquence de déroutement de la bande structure.
On imprime le message "ERREUR". On essaie de lire le fichier BIBLIØ qui n'existe pas.
On ferme la bande et on arrête l'exécution du programme.

.../...

44 - Edition d'un fichier structure :

Le programme d'édition étant assez simple, nous n'avons pas fait d'organigramme. Nous préciserons seulement la façon dont le programme se présente.
Nous lisons dans une zone MEM de 3500 mots la première partie du fichier structure, ce qui correspondait à la zone EXPLØ.
Puis nous éditons cette zone.
TAILLE NUMERØ D'ARTICLE :
TAILLE NØMBRE DE :
TAILLE TYPE DE M2 :
TAILLE TYPE DE M3 :
TAILLE TYPE DE M1 :
TAILLE TYPE DE CARTE :
TAILLE NØM INFØRMATIØN :
TAILLE TAILLE INFØRMATIØN :
TAILLE STRUCTURE EN ØCTETS :
Puis nous lisons dans la même zone MEM la table structure.
Nous éditons les titres des 7 zones de la table structure.
TYPE DE CARTE
TYPE DE MØDULE
NØM DE L'INFØRMATIØN
FIXE ØU REPETITIF } une zone
LØNGUEUR PRØPRE }
PØSITIØN/ØRIGINE DU MØDULE
LØNGUEUR UNITE SI REPETITIVE
THESAURUS
Puis nous remplissons un tableau que nous éditons avec toutes les données de la table structure.

45 - Edition du fichier maître :

Pour cette édition nous n'avons pas fait d'organigramme.
Nous lisons dans une zone MEM de 6304 mots un enregistrement.
Nous éditons d'abord la table, ce qui donne
TAILLE ARTICLE + TABLE ;
TAILLE TABLE ;
Puis l'impression d'un tableau comportant 3 zones dont on imprime d'abord les titres.
TYPE DE MØDULE
ADRESSE DE DEBUT DU MØDULE
MULTIPLICITE DU MØDULE
Derrière cette table on imprime l'article brut à raison de 130 caractères par ligne.
On passe alors à l'enregistrement suivant et ceci jusqu'à épuisement des enregistrements contenus dans le fichier maître.

.../...

46 - Edition du fichier dictionnaire :

461 - Particularités du programme :

- On a réservé . une zone DIC de 10000 mots pour lire le fichier BIBLIØ.
. une zone P1 et une zone P2 qui sont des indicateurs.
. une zone Y pour garder l'adresse de l'adresse d'un dictionnaire.
. une zone Z pour garder l'adresse de l'origine d'un dictionnaire.

- On a deux zones de comparaison
PRES qui contient le mot "PRES"
CØDE qui contient le mot "CØDE".

- On a une zone ERREUR qui contient le mot ERREUR.

- On a des zones d'édition.
Une zone NOM de 5 mots : "1 DICTIØNNAIRE".
1 sur le premier octet provoque un saut de page à l'impression.
Une zone IMP1 de 3 mots qui contient "PRESENCE".
Une zone IMP2 de 3 mots qui contient "CØDAGE", "DECØDAGE".
Une zone IMP3 de 3 mots qui contient "RECØDAGE".
Une zone IMP4 de 13 mots qui contient
"NØ:" CØDE"
Une zone IMP5 de 14 mots que l'on remplira pendant l'exécution du programme.

- On a une zone TRAD1 : même usage que dans les autres programmes.

- On a 2 procédures
RANGE
PLACER
qui utilisent l'instruction MOS que nous avons déjà décrite.

- Les registres qui ont un rôle particulier sont :
le registre 1 appelé dic qui sert de pointeur pour retrouver les dictionnaires.
le registre 2 appelé longdic qui sert de pointeur dans DIC. C'est un pointeur d'octets.
le registre 3 appelé LØNG qui contient la taille de l'information pour PRES et la somme : taille externe et interne de l'information pour CØDE et DECØ.
le registre 4 appelé NØMB qui contient le nombre d'informations présentes dans le dictionnaire.

.../...

le registre 6 appelé X qui contient le numéro du dictionnaire que l'on va éditer.
le registre 9 appelé J qui contient la taille occupée par toutes les informations.
le registre 12 appelé LØNCEX qui contient la taille externe de l'information pour CØDE et DEJØ.
le registre 13 appelé LØNCIN qui contient la taille interne de l'information pour CØDE et DEJØ.

62 - Analyse détaillée :

prenons l'organigramme pour suivre cette analyse :

DEB On ouvre la bande structure : on lit les 2 blocs de BIBLIØ dans DIC.
On ferme la bande structure.
On initialise dic, P1 et P2 à 0.

ETI6 test : DIC, dic = 0 ?
oui : ETI1 le dictionnaire de numéro dic n'a pas été utilisé.
test : dic = 10 ?
oui : on a lu tous les dictionnaires.
On arrête l'exécution du programme.
non : on passe au dictionnaire suivant en incrémentant dic de 1 et en retournant en ETI6.
dic = dic + 1
non : on remplit la zone ETP5 de 0.
On met l'adresse du dictionnaire à éditer dans longdic.
On range longdic en Z.
On imprime le numéro du dictionnaire.
test : nom de dictionnaire = PRES ?
non : on se branche en DC1.
oui : on incrémente longdic de la place occupée par PRES.
longdic = longdic + 4
On imprime sous le numéro du dictionnaire le nom "PRESENCE" qui le caractérise.
test : premier caractère de DIC, longdic = "*" ?
oui : on se branche en ETI3.
non : on incrémente longdic de la place occupée par "place libre restante".
longdic = longdic + 1
On se branche en ETI1.

ETI3 Le dictionnaire a débordé.
On positionne P1 à 1.
On incrémente longdic de la place occupée par1"*" .
longdic = longdic + 1.
On range le numéro de dictionnaire sur lequel on a débordé dans X. On range dans X l'adresse où se trouve le dictionnaire de débordement.
On remplit le mot où se trouvait cette adresse de 0.

.../...

En effet, lorsqu'on passera en revue les 50 premiers mots de DIC (cf étiquette ETI6), si on trouve la zone remplie, on pourra croire qu'il s'agit d'un nouveau dictionnaire.
On incrémente longdic de la place occupée par ce numéro de dictionnaire.
longdic = longdic + 3
test : P2 = 1 ?
indicateur qui permet de se brancher à l'une ou l'autre adresse.
non : on se branche en RET1.
oui : on positionne P2 à 0 et on se branche en ETI2.

RET1 On range la taille de l'information dans LØNG.
On incrémente longdic de la place occupée par cette taille.
longdic = longdic + 3
On range le nombre d'informations NE dans NØMB.
On incrémente longdic de la place occupée par ce nombre.
longdic = longdic + 3
On calcule la place occupée par toutes les informations.
J = LØNG x NØMB.
On se branche en ETI2.

ETI2 On prend LØNG caractères de DIC grâce au pointeur longdic et on les range dans la zone d'impression IMP5.
On imprime IMP5.
On décrémente J de la taille d'une information.
J = J - LØNG
et on incrémente longdic de cette même taille.
longdic = longdic + LØNG
test : J = 0 ?
oui : on se branche en ETI7.
non : test : P1 = 1 ?
P1 est un indicateur. Il permet de savoir s'il y a un ou plusieurs dictionnaires de débordement.
non : on retourne en ETI2.
oui : on calcule si l'on a rangé toutes les informations du dictionnaire avant de se brancher sur un dictionnaire de débordement.
I = 1276 - (longdic - Z)
1276 est la taille maximum d'un dictionnaire en octets;
longdic est le pointeur dans le dictionnaire.
Z représente l'adresse d'origine du dictionnaire.
test : I ⩾ LØNG ?
oui : il reste des informations à ranger. On retourne en ETI2.
non : on avait gardé dans Y à l'étiquette ETI3,

.../...

l'adresse du dictionnaire sur lequel on avait débordé.
on range Y dans longdic.
test : DIC, longdic = "* " ?
On regarde si ce dictionnaire est le dernier ou s'il est aussi suivi d'un dictionnaire de débordement.
oui : on se branche en PLL.
non : il est le dernier.
On positionne P1 à 0, ainsi on n'aura plus besoin de calculer si l'on est au bout du dictionnaire puisqu'il n'est pas plein.
On incrémente longdic de la place occupée par "place libre restante".
longdic = longdic + 4
On retourne en ETI2.

PLL On range l'adresse d'origine du dictionnaire dans Z.
Puisqu'il va y avoir un nouveau dictionnaire de débordement, on devra faire le calcul I = 1270 - (longdic - Z).
On positionne P2 à 1.
On retourne en ETI2, pour chercher où se trouve le prochain dictionnaire de débordement.

ETI7 On a fini d'imprimer un dictionnaire. On passe au suivant en incrémentant dic de 1 et en retournant en ETI6.

DEL test : nom = CØDE ?
non : on se branche en ETI7.
oui : on incrémente longdic de la place occupée par CØDE.
longdic = longdic + 1
On imprime le titre "CØDAGE DECØDAGE".
On se branche en ETI8.

ETI8 Il s'agit d'un dictionnaire de décodage. Seul le titre change : on imprime "DECØDAGE". Le traitement est ensuite le même que pour CØDE.
On incrémente longdic de la place occupée par DEC Ø.
longdic = longdic + 4
On se branche en ETI8.

ETI9 On imprime les titres MØ. et CØDE.
test : premier caractère de DIC, longdic = "*" ?
oui : on se branche en ETI .
non : on incrémente longdic de la place occupée par "place libre restante".
longdic = longdic + 2
On se branche en ETI2.

ETI5 Le traitement est le même qu'en ETI3. Seules les adresses de retour changent.
test : P2 = 1 ?
oui : on positionne P2 à 0 et on va en ETI4.
non : On va en ETI2.

.../...

RET2 On range la taille externe de l'information dans LØNGEX.
On incrémente longdic de la place occupée par cette taille.
longdic = longdic + 3
On range la taille interne de l'information dans LØNGIN.
On incrémente longdic de la place occupée par cette taille.
longdic = longdic + 3
On range le nombre d'éléments dans NØMB.
On incrémente longdic de la place occupée par ce nombre.
longdic = longdic + 3
On calcule la somme des deux tailles dans LØNG et la place qu'occupent toutes les informations dans J. On se branche en ETI4.

ETI4 On prend LØNGEX caractères de DIC grâce au pointeur longdic et on les range à partir du quatrième octetde la zone IMPS.
On incrémente longdic de la place occupée par la taille externe.
longdic = longdic + LØNGEX
On prend LØNGIN caractères de DIC grâce au pointeur longdic et on les range à partir du 46ème octet de la zone IMPS.
On incrémente longdic de la place occupée par la taille interne.
longdic = longdic + LØNGIN
On imprime IMPS : la taille externe de l'information se trouve sous le titre "LØNG" et la taille interne sous le titre "LØNGI".
On décrémente J de ces deux tailles.
J = J - LØNG
test : J = 0 ?
oui : on se branche en ETI7.
non : test : P1 = 1 ?
même signification que pour le dictionnaire PRES.
non : on retourne en ETI4.
oui : on fait le calcul : I = 1276 - (longdic - Z)
test : I ⩾ LØNG ? comme pour PRES.
oui : on retourne en ETI4.
non : on range Y dans longdic.
test : DIC, longdic = "*" ? comme pour PRES.
non : on positionne P1 à 0. On incrémente longdic de 1.
longdic =longdic + 1
On retourne en ETI4.
oui : FIN2 On range longdic dans Z.
on positionne P2 à 1. On va en ETI1.

ERREUR séquence d'anomalie.
On n'a pas trouvé le fichier ENTLØ sur la bande, on imprime le message "ERREUR".
On ferme la bande structure et on arrête l'exécution du programme.

.../...

Conclusion

CONCLUSION :

Nous avons présenté dans cette thèse la mise sur bande de fichiers à enregistrements de longueur variable et décrits sous forme d'arborescence à trois niveaux.
Le système de traitement est constitué, par une bibliothèque de programmes :

- le programme "fichier structure" qui range sur bande un ou plusieurs fichiers structures et un fichier particulier dit BIBLIØ ne comprenant que des thésaurus.
- le programme "fichier maître" qui utilise un ou plusieurs fichiers structures rangés précédemment, pour mettre en forme les enregistrements qui seront dans un même fichier sur bande.
- le programme de mise à jour du fichier BIBLIØ
- le programme d'édition d'un fichier structure quelconque
- le programme d'édition de tout le fichier maître
- le programme d'édition de tous les thésaurus.

Après avoir réalisé cette mise sur bande, nous avons abordé l'extraction. Cette extraction, comme la mise sur bande, est générale : nous pourrons l'appliquer à tous les types de fichiers dont la structure est conforme à notre système . Nous avons expliqué au cours de ce travail le principe de l'extraction et nous décrirons dans une phase ultérieure chaque programme d'extraction, ceux-ci étant encore en cours d'écriture.
Précisons brièvement l'organisation de cette extraction. Nous avons quatre programmes :
- le premier extrait une information élémentaire ou un groupe d'informations élémentaires de même identificateur si l'information est présente plusieurs fois dans l'article (cf répétitivité des différents niveaux).
- le deuxième extrait un noeud de niveau 3 ou un ensemble de noeuds de niveau 3 correspondant au même identificateur si le noeud est présent plusieurs fois dans l'article (cf répétitivité des différents niveaux).
- le troisième extrait un noeud de niveau 2 ou un ensemble de noeuds de niveau 2 correspondant au même identificateur si ce noeud est "répétitif".
- le quatrième extrait le noeud de niveau 1 c'est-à-dire tout l'article.

Le traitement sera constitué d'une chaine de programmes comprenant des programmes d'extraction généraux et des programmes de traitement spécifiques à chaque utilisateur.
Prenons un exemple simple : sélection d'un article suivant ses coordonnées géographiques et construction d'un tableau contenant les épaisseurs des matériaux et le numéro de classement de l'article.

.../...

- extraction de X Lambert (1)
- appartenance de X Lambert à une fenêtre de valeurs (2)
- extraction de Y Lambert (3)
- appartenance de Y Lambert à une fenêtre de valeurs (4)
- sélection de la découverte (couche supérieure ou toit) à condition qu'elle contienne de la grave, du sable ou des alluvions (5)
- sélection du substratum (couche inférieure ou mur) à condition que celui-ci contienne de la grave, du sable ou des alluvions (6)
- calcul de l'épaisseur des matériaux (7)
- extraction du numéro de classement (8)
- construction du tableau des épaisseurs (9)

Cet ensemble de questions est géré par un programme moniteur. Les questions (2) et (4) semblables pourront correspondre à un même programme.
Certaines questions, telles que (2), (4), (7) et (9) peuvent se retrouver chez différents utilisateurs.
Actuellement ce sont des programmes spécifiques. Nous envisagerons de dépasser ce stade, c'est-à-dire de construire une bibliothèque commune de programmes de traitement, si le nombre d'utilisateurs devient suffisant.
Les questions (5) et (6) resteront des programmes spécifiques, leur complexité étant due à la répétitivité des couches.

UNIVERSITÉ DE NANCY BIBLIOTHÈQUE

NOM DE L'ETUDIANT : MIGNOT Nicole

NATURE DE LA THESE : Doctorat de Spécialité en Mathématiques Appliquées

VU, APPROUVE

& PERMIS D'IMPRIMER

NANCY, le 28.9.73.

LE PRESIDENT DE L'UNIVERSITE DE NANCY I

J.R. HELLUY