

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Langage terminal d'édition d'une banque de données budgétaires

Colussi, Silvio

Award date:
1976

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX, NAMUR
Institut d'Informatique

Année académique 1975-1976

**LANGAGE TERMINAL D'EDITION
D'UNE BANQUE DE DONNEES
BUDGETAIRES**

Silvio COLUSSI

Mémoire présenté en vue de l'obtention
du grade de licencié et Maître en
Informatique.

2

Nous tenons à remercier toutes les personnes qui, de quelque manière que ce soit, ont participé à la réalisation de ce mémoire. En particulier, toute notre reconnaissance va à M. Bodart qui, par ses conseils judicieux, nous a permis de mener à bien ce travail. Nous remercions également M. Raway et M. Dehenneffe, dont la collaboration fructueuse nous fut précieuse, particulièrement lors de la détermination de la syntaxe du langage. Nos remerciements vont également à l'équipe du simulateur pour ses éclaircissements sur les aspects techniques du problème. Nous témoignons également notre gratitude aux personnes qui ont permis la réalisation matérielle de ce mémoire et en particulier à Melle Bianchi.

TABLE DES MATIERES.

	<u>Pages.</u>
INTRODUCTION	i
<u>PREMIERE PARTIE : SPECIFICATIONS FONCTIONELLES D'UN</u> LANGAGE TERMINAL D'EDITION D'UNE BANQUE DE DONNEES BUDGETAIRES	
CHAP.1. EXPOSE GENERAL DU SIMULATEUR	
1.1 Objectifs et architecture du simulateur	I-1
1.1.1 Objectifs du simulateur	I-1
1.1.2 Architecture du simulateur	I-4
1.2 Description de la banque de données	I-4
1.2.1 Modèle sémantique de perception	I-4
1.2.2 Graphe d'activité	I-5
1.2.3 Modèle sémantique du graphe d'activité	I-7
1.2.4 Modèle I.D.S	I-14
CHAP.2. DEMARCHE SUIVIE POUR LA DETERMINATION DE LA STRUCTURE ET DU CONTENU D'UNE REQUETE	
2.1 Liste de questions susceptibles d'être posées par un utilisateur	I-16
2.1.1 Questions relatives aux consommations des différents postes	I-16
2.1.2 Questions relatives aux prestations des différents postes	I-18
2.1.3 Situation des stocks	I-19
2.1.4 Situation sur les marchés	I-19
2.1.5 Analyse des prix de revient	I-20
2.2 Analyse des questions	I-21
2.2.1 Description de quelques questions	I-21
2.2.2 Conclusions	I-25
2.2.2.1 Liste des opérateurs à définir	I-25
2.2.2.2 Paramètres à faire figurer dans la requête	I-25
2.2.2.3 Types de questions pouvant être posées	I-26
<u>DEUXIEME PARTIE : DEFINITIONS ET PROPRIETES DU LANGAGE</u>	
CHAP.3. SYNTAXE DU LANGAGE	
3.1 Aperçu général d'une requête	II-1

3.2 Définitions préliminaires	II-4
3.2.1 Définitions	II-4
3.2.1.1 Générique de codification	II-4
3.2.1.2 Identifiant	II-5
3.2.1.3 Entité et entité relation	II-6
3.2.2 Objectifs du langage	II-7
3.3 Syntaxe	II-8
3.3.1 Partie localisation	II-8
3.3.1.1 Localisation d'une entité par son nom	II-8
3.3.1.2 Critères de sélection	II-10
3.3.1.3 Associations reconnues dans le système	II-18
3.3.1.4 Conclusions	II-25
3.3.2 Partie calcul	II-26
3.3.2.1 Opérateur d'addition	II-26
3.3.2.2 Opérateur de soustraction	II-26
3.3.2.3 Opérateur de multiplication	II-27
3.3.2.4 Opérateur de division	II-28
3.3.2.5 Opérateur de calcul d'un pourcentage	II-28
3.3.2.6 Opérateur de cumul	II-29
3.3.2.7 Remarques	II-30
3.3.3 Partie édition	II-31
3.3.3.1 Détermination de l'en-tête	II-31
3.3.3.2 Détermination de la séquence de rupture	II-32
3.3.3.3 Détermination des données à éditer	II-34
3.3.4 Remarque	II-35
CHAP.4. ALGORITHME D'ANALYSE DU LANGAGE	
4.1 Grammaire formelle du langage	II-36
4.2 Algorithme d'analyse	II-37
4.2.1 Représentation graphique de productions	II-38
4.2.2 Automate fini non-déterministe	II-38
4.2.3 Automate fini déterministe	II-38
4.2.4 Algorithme d'analyse	II-39
<u>TROISIEME PARTIE : ANALYSE DE L'IMPLEMENTATION</u>	
CHAP.5. DICTIONNAIRES DU SYSTEME	
5.0 Introduction	III-3
5.1 Définitions préliminaires	III-3
5.2 Dictionnaire des associations	III-4

5.3 Dictionnaire des caractéristiques	III-5
5.4 Dictionnaire des données	III-6
CHAP.6. MODULE I.D.S	
6.1 Objectifs du module	III-11
6.2 Fonctions utilisées	III-12
6.3 Autres fonctions du module	III-13
CHAP.7. SORTIES DE L'INTERPRETEUR	
7.0 Introduction	III-14
7.1 Génération d'informations sur base de la partie localisation	III-14
7.2 Génération d'informations sur base de la partie calcul	III-16
7.3 Génération d'informations sur base de la partie édition	III-18
7.4 Stockage des descripteurs	III-19
CHAP.8. SELECTION DES OCCURENCES ET CONSTITUTION DES LIGNES A EDITER	
8.0 Introduction	III-21
8.1 Sélection des occurrences désirées	III-21
8.1.1 Sélection sur base des valeurs des identifiants	III-22
8.1.2 Sélection par comparaison des valeurs des données avec une valeur fixée par l'utilisateur	III-28
8.2 Calcul de données	III-30
8.2.1 Organisation générale du calcul	III-30
8.2.2 Logique interne de chaque opérateur	III-32
8.3 Constitution des fichiers utilisés lors de l'édition	III-35
8.3.1 Constitution du fichier des lignes à éditer	III-35
8.3.2 Constitution du fichier des libellés	III-37
CHAP.9. STRUCTURE GENERALE DE L'EDITION	
9.0 Introduction	III-39
9.1 Définition des fichiers utilisés	III-39
9.2 Définition de l'état imprimé	III-39
9.2.1 Définition de séquence de rupture	III-39
9.2.2 Définition des groupes d'édition et des traitements associés	III-44
ANNEXES	
Description des équations du simulateur	An 1
Algorithme d'analyse d'une requête	An B
Logique des opérateurs de calcul	An C
Algorithme de sélection	An D
CONCLUSIONS	c
BIBLIOGRAPHIE	

I N T R O D U C T I O N .

Lorsqu'une entreprise a établi ses budgets, elle dispose d'une masse importante d'informations qu'elle doit mettre à la disposition des différents responsables. Ceux-ci ne désirent pas nécessairement disposer d'un seul coup des informations auxquelles ils peuvent accéder. L'objectif de ce travail est de proposer un outil d'interrogation d'une banque de données budgétaires devant permettre :

- de gérer par exception c'est-à-dire permettre à l'utilisateur de choisir parmi un ensemble de données, dont il a le droit de prendre connaissance, celles qu'il juge utile à un instant donné du temps. Par ce procédé, on évite de noyer l'utilisateur sous un tas d'informations non nécessaires;
- d'effectuer des regroupements de données selon une multiplicité de critères, définis par l'utilisateur, en vue d'éditer des documents de synthèse et d'analyse du programme prévisionnel d'activités déterminé par un modèle de simulation. Ceci nécessite la mise à la disposition de l'utilisateur d'un certain nombre d'opérateurs plus ou moins sophistiqués permettant l'élaboration d'informations complexes sur base des informations élémentaires figurant dans la banque de données.

Pour réaliser cet outil, nous avons, dans un premier temps, essayé d'imaginer les manipulations qu'un utilisateur souhaiterait réaliser sur le contenu de la banque de données. Nous en avons déduit un ensemble de questions que celui-ci jugerait utile de poser. Nous les avons reprises une à une pour en dégager les données demandées ainsi que la manière de les utiliser. Cela nous a permis de déterminer :

- les opérateurs à mettre à la disposition de l'utilisateur qui seront dans le cadre de ce travail des opérateurs arithmétiques uniquement;
- la structure d'une requête qui se compose de trois parties :
 - localisation : l'utilisateur y indiquera les entités dont il voudra utiliser les données;
 - calcul : il y indiquera les éventuelles manipulations de données;
 - édition : il y spécifiera la structure de l'état imprimé ainsi que les données qui le constitueront.

Après avoir défini la structure syntaxique de chaque partie

de la requête, en spécifiant les restrictions apportées, nous avons procédé à la définition formelle de cette syntaxe. De cette définition, nous dégagerons les propriétés du langage qui seront utilisées pour la construction d'un interpréteur qui devra fournir en sortie :

- l'ensemble des informations nécessaires à la sélection des valeurs des données désirées ainsi qu'à la composition de l'ensemble des lignes à éditer : ce seront un ensemble de descripteurs utilisés par un module de recherche dans la banque de données. Cette phase est réalisée en tenant compte de la structure hiérarchique de la banque de données qui apparaît déjà dans la codification des entités;
- l'ensemble des informations nécessaires à l'élaboration de nouvelles données : qui figureront dans ce que nous appellerons la "table des données calculées";
- l'ensemble des informations nécessaires à la constitution de l'état imprimé : celui-ci sera réalisé sur base des données, fournies par l'utilisateur, définissant l'en-tête d'état ainsi que les niveaux hiérarchiques auxquels il désire voir apparaître des sous-totaux. Ces informations permettront de définir une séquence de rupture de laquelle on pourra déduire l'ensemble des groupes d'édition constituant l'état. Cette description sera obtenue en générant le texte source de la "report section" (section définissant l'état imprimé dans le langage COBOL) qui sera intégré dans la partie "données" d'un module d'édition.

Remarque : Nous travaillons sur une banque de données dont nous ne pouvons modifier la valeur des données. Nous ne faisons que de la consultation, avec possibilité d'effectuer des traitements, n'ayant aucune influence sur le contenu de la banque de données.

PREMIERE PARTIE

SPECIFICATIONS FONCTIONNELLES D'UN LANGAGE TERMINAL D'EDITION D'UNE BANQUE DE DONNEES BUDGETAIRES.

Nous présentons, tout d'abord, le cadre dans lequel le travail s'inscrit à savoir la structure générale du simulateur. Nous travaillons plus précisément sur son réservoir de données constitué d'une banque de données dont le développement est fondé sur un modèle comptable. La structure de cette banque de données est basée sur le graphe d'activité qui est calqué sur la structure de la comptabilité analytique et générale et qui met en évidence les interrelations logiques existant entre les différents éléments de l'entreprise. Le modèle I.D.S de la banque de données, qui est celui sur lequel nous travaillons, reprend toutes ces interrelations logiques. Ensuite, nous formulons un ensemble de questions susceptibles d'être posées par un utilisateur concernant le contenu de la banque de données. Après une analyse de chacune d'elles, nous dégagerons les caractéristiques que le langage devra vérifier pour répondre aux besoins d'un utilisateur.

CHAPITRE I.

Exposé général du simulateur.

11. Objectifs et architecture du simulateur.

1.1.1. Objectifs du simulateur.

Classiquement, la procédure d'établissement du budget commençait relativement tôt dans l'année à savoir en avril ou juin. C'est en effet vers cette époque que débutait la collecte des données qui portait sur :

- les prévisions de vente et des prix de vente sur les marchés;
- les prévisions des capacités d'activité;
- les prix des natures de dépenses.

Vers la fin septembre ou octobre, on pouvait obtenir une hypothèse de budget qui était alors examinée par les responsables attitrés. Ceux-ci la validaient ou la refusaient. Dans ce dernier cas, on recommençait pour disposer de la nouvelle hypothèse un mois plus tard. Cette procédure fait apparaître les problèmes suivants :

- lourdeur;
- temps de réponse beaucoup trop élevé;
- fraîcheur des données.

La source des informations nécessaires à cette procédure d'élaboration est l'ensemble des fichiers associés aux flux existant dans l'entreprise. Toute modification (changement de valeur, création ou suppression d'une donnée) devra automatiquement se répercuter dans le réservoir de données propre à cette procédure. Ceci permet de travailler avec des données fraîches (la répercussion ne se faisant pas nécessairement en temps réel, il se peut que celles-ci datent d'une semaine mais certainement plus de trois ou quatre mois) et d'éviter des contretemps fâcheux tel l'augmentation subite du prix d'une matière première consommée en grande quantité une quinzaine de jours avant la date de remise du budget et contre laquelle on ne peut réagir vu que le temps d'établissement d'une nouvelle hypothèse est d'un mois.

Un autre problème important dans une entreprise est celui de la détermination des prix de revient. On a remarqué que l'ensemble des informations nécessaires au calcul de ces derniers était exactement identique à celui utilisé pour l'élaboration du budget. Les deux procédures seront donc réalisées simultanément.

La possession de données fraîches serait inutile si le temps d'exécution de la procédure d'élaboration du budget et de calcul des prix de revient est trop important. On a donc développé un modèle qui travaille en un temps très court (48 heures maximum) et qui, par conséquent, peut s'insérer dans le processus normal de gestion des responsables de l'entreprise. Ceci rend la réestimation d'un budget beaucoup plus aisée. Mais la conséquence la plus importante de cette rapidité est la possibilité de tests de nouvelles politiques tant industrielles que commerciales. Traditionnellement, on arrivait difficilement à calculer trois hypothèses. L'augmentation de rapidité autorise l'évaluation d'un plus grand nombre de cas tout en leur accordant une plus grande crédibilité vu la fraîcheur des données. Ceci permet une plus grande adaptabilité du comportement de l'entreprise aux fluctuations brutales des marchés. Elle ne sera plus impuissante devant un changement soudain du prix d'une matière première fortement consommée, même si cela intervient durant l'exercice pour lequel les budgets ont été établis. L'entreprise pourra donc, même en cours d'exercice, revoir sa politique soit industrielle, soit commerciale.

Ces améliorations visent à répondre aux critères suivants :

- diminution du temps de réponse;
- l'analyse d'un nombre plus important de stratégies.

Le modèle est d'autant plus intéressant qu'il est intégré c'est-à-dire qu'il prend en compte les interactions existant entre les différents éléments de l'entreprise. Il devient possible de répondre à des questions du type : examiner la répercussion d'une hausse de $x\%$ du coût d'une nature de dépense sur le prix de revient à tel stade de transformation de tel produit. Ces interactions sont matérialisées au niveau des équations du modèle. Celles-ci se subdivisent en sept grandes classes :

- équation concernant les niveaux de vente par produit : ceux-ci sont compris entre deux bornes représentant les capacités minimales et maximales des marchés;
- équation concernant les niveaux de production des produits intermédiaires à chaque stade qui sont également compris entre une contrainte minimale et maximale de production. Les produits sont reliés aux stocks qu'ils alimentent, aux sections et aux stocks de fournitures primaires qu'ils consomment;
- équation concernant les stocks finaux de produits fabriqués;
- équation concernant les niveaux d'activité des sections de production qui sont reliées aux sections auxiliaires et aux stocks de fournitures primaires qu'elles consomment (la forme mathématique de cette équation figure annexe 1.1);
- les niveaux d'activité des sections auxiliaires reliés aux sections auxiliaires et stocks de fournitures primaires;
- les niveaux de consommation des fournitures primaires consommées soit par la production, soit par les sections de vente, ou directement incorporées aux produits finis;
- les niveaux de stock des fournitures primaires qui sont fonction du niveau d'achat et des stocks initiaux (formulation mathématique annexe 1.1)

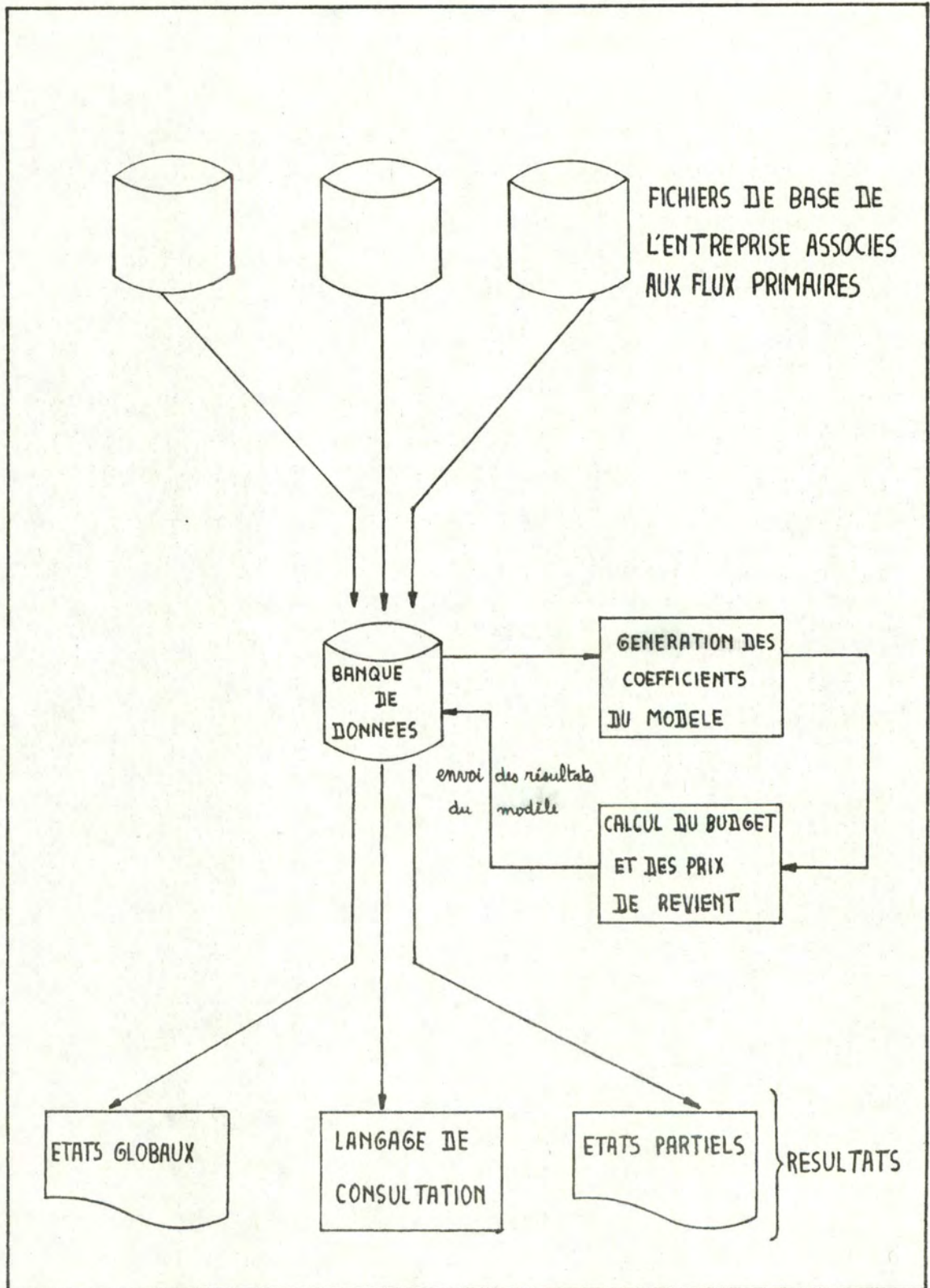


Figure 1.1

1.1.2. Architecture du simulateur.

Toutes les améliorations apportées jusqu'ici au processus d'élaboration du budget seraient sans grand effet si l'on injecte manuellement dans le modèle les données nécessaires à son fonctionnement. On est dès lors passé par une phase de génération automatique de la matrice des coefficients du programme linéaire d'activité. Un ensemble de programmes écrit les coefficients pour chaque type de contrainte du modèle d'optimisation, après les avoir calculés en fonction des données figurant dans la banque. Cette méthode présente les avantages suivants :

- rapidité supérieure à tout autre méthode;
- suppression de toute erreur de calcul ainsi que de toute intervention humaine. On évite non seulement toutes les erreurs de transcription mais on gagne surtout un temps précieux car injecter les coefficients d'un programme linéaire de 400 variables et 2 000 contraintes ne se fait pas très rapidement.

Ces avantages permettent d'obtenir une interaction entre le modèle et le simulateur puisque la génération des coefficients prend maintenant douze minutes. On obtient ainsi le système décrit figure 1.1.

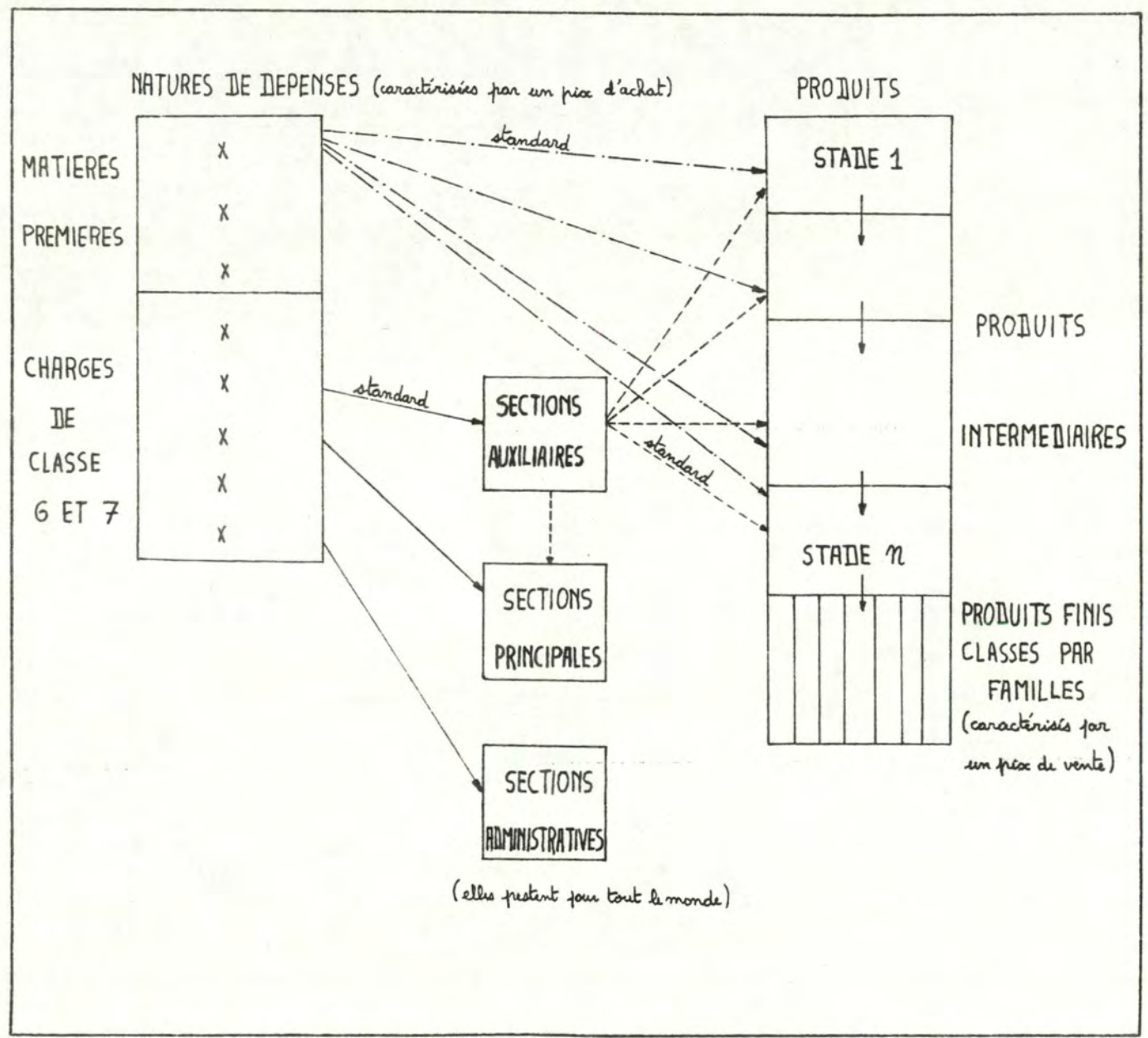
1.2. Description de la banque de données.

1.2.1. Modèle sémantique de perception.

Le développement de la banque de données est fondé sur un modèle économique et plus précisément sur un modèle comptable (figure 1.2.). La nature des relations entre deux postes de ce modèle est obtenu en comparant au niveau quantitatif l'évolution de la variation d'une quantité d'un produit par une quantité de nature de dépenses. Ce sont toutes des relations linéaires car elles sont caractérisées par un standard qui est une constante. Les inconnues de ce modèle sont les niveaux de vente par produit, les niveaux de production des produits intermédiaires à chaque stade, les niveaux des stocks finaux de ces produits, les niveaux d'activité des sections de production et auxiliaires par stade et section, les niveaux de consommation des natures de dépenses et de leurs stocks.

Légende :
 - - - - : consommations d'unités d'œuvre des sections auxiliaires
 ——— : nature de dépenses consommées par les sections
 - · - · - : nature de dépenses directement affectée aux produits

Figure 1.2



On associe à ce modèle exprimant le processus de production, un graphe qui décrit schématiquement le processus de production et de commercialisation des biens et services de l'entreprise. Il reprend tous les éléments à considérer, ainsi que leur niveau de détail retenu.

1.2.2. Graphe d'activité.

Il sera la base de la structure de la banque de données. Etant calqué, dans toute la mesure du possible, sur la structure de la comptabilité analytique et générale, il permet d'assurer l'homogénéité entre ces dernières et le simulateur ce qui :

- facilite la collecte des informations nécessaires à ce dernier;
- rapproche le simulateur des préoccupations des utilisateurs.

Le graphe d'activité (figure 1.3.) comprend :

- l'ensemble des "recettes de cuisine" correspondant aux biens et services produits, c'est-à-dire :
 - les nomenclatures = qualités et quantités de matières premières à mettre en oeuvre;
 - les gammes opératoires = séquence des opérations spécifiant les services à appliquer aux matières par différents centres d'activités pour obtenir un produit fini;
- la description de la capacité de production des cellules de l'entreprise;
- les cheminements possibles des flux de charges, de services et de produits dans l'entreprise tant en quantité qu'en valeur. Les minima et maxima éventuels sont spécifiés.

Il se compose de deux grands axes :

- axe des activités qui détermine la capacité de production de l'entreprise. Il est subdivisé en sections (ou centre de frais, ou centre d'activité) de 3 types :
 - principales (de production ou de commercialisation) dont les services sont directement appliqués au flux de produits;

SCHEMA GENERAL DU GRAPHE D'ACTIVITE

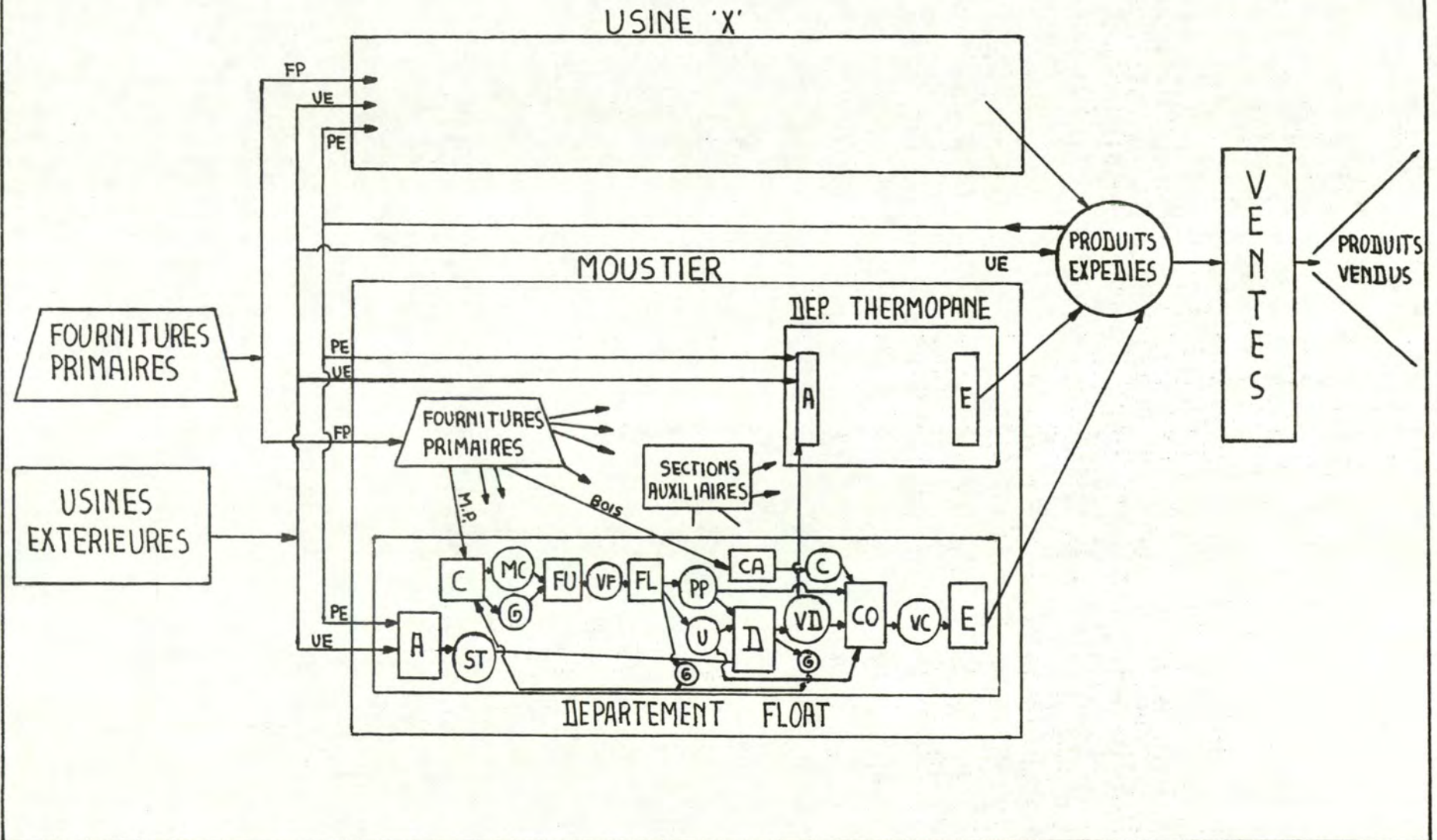
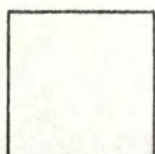


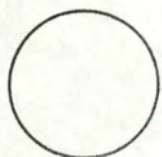
Fig. 1.3.

Légende du schéma général.



= bloc

A = Approvisionnements
C = Composition
FU = Fusion
FL = Flottage
CA = Caisserie
D = Découpe
CO = Conditionnement
E = Expéditions



= Centre de stock

St = Produits "sous-traités"
MC = Matières composées
VF = Verre fondu
PP = "Pré-Plateaux"
U = Uncuts
C = Caisses
VD = Verre découpé
VC = Verre conditionné
G = Groisil

MP = Matières premières

CA TP = Cessions à l'atelier Thermopane (= échanges inter
départements)

FP = Fournitures primaires

PE = Produits expédiés (= échanges inter-usines)

UE = Usines extérieures(= produits sous-traités)

- auxiliaires : dont les services sont prestés au bénéfice d'autres sections et dont on peut mesurer et, imputer les prestations;
- administratives : dont les services sont prestés au bénéfice de l'entreprise sans que l'on puisse les imputer.

Elles possèdent une unité de mesure appelée unité d'oeuvre qui n'est pas nécessairement unique. Elles consomment en entrée des natures de charges et des unités d'oeuvre prestées par les sections auxiliaires. Leurs sorties sont constituées par une activité exprimée en unités d'oeuvre. Chaque section est caractérisée par une capacité d'activité et la structure et le coût de l'unité d'oeuvre.

- axe des produits : il se compose de :
 - blocs ou stades de production : = groupe de sections principales dont les unités d'oeuvre sont appliquées simultanément aux matières ou produits. Les produits, qui y sont fabriqués, consomment des matières premières, des produits sortant des blocs précédents et des unités d'oeuvre de toutes ou de certaines sections qui le composent. En sortie, on obtient une certaine quantité de produit. Les blocs sont introduits pour matérialiser le fait que le cheminement des matières et produits n'est pas nécessairement continu et unique d'une section à l'autre. Il peut y avoir des interruptions et bifurcations;
 - centres de stocks : qui permettent de traduire les chemine-
ments possibles des produits entre blocs.
Ils comprennent :
 - centres de stocks de produits physiques qui donnent par produit les origines et destinations possibles ainsi que les minima et maxima éventuels;
 - centres de stocks de natures de charges qui par nature de charge définissent en quantité et en valeur celles qui sont disponibles et leurs destinations possibles;
 - les centres de stocks de matières premières qui présentent, en quantité et en valeur, celles qui sont disponibles, leurs destinations, les minima et maxima éventuels.

1.2.3. Modèle sémantique du graphe d'activité.

Le graphe d'activité reprend toutes les interrelations logiques existant entre les différents éléments de l'entreprise (Ex : un produit est lié à ses consommations en matières premières et en unités d'oeuvre, et au stock où il entre). On va reprendre ici toutes les relations existant dans le graphe et les détailler. Chaque relation sera représentée par un graphe possédant deux types de sommets : d'une part, ceux représentant les types d'objets ou entités, d'autre part, ceux représentant un type d'association. A chaque sommet seront associés les différents types de propriétés qui seront les données constituant l'article de la banque de données associé à l'objet cité.

- associations relatives à une section principale de production

(fig. 1.4)

types d'objets : - section de production

- produits
- bloc
- section auxiliaire
- centre de stock de natures de charges.

les types de propriétés constituent les articles correspondants de la banque de données.

types d'associations : - prestations
 - appartenance
 - consommations

les types de propriétés des associations, sauf pour appartenance, figurent dans les articles associés aux standards correspondants.

cardinalité de liaison : - 1 - n pour appartenance (bloc - section)
 - n n pour les autres.

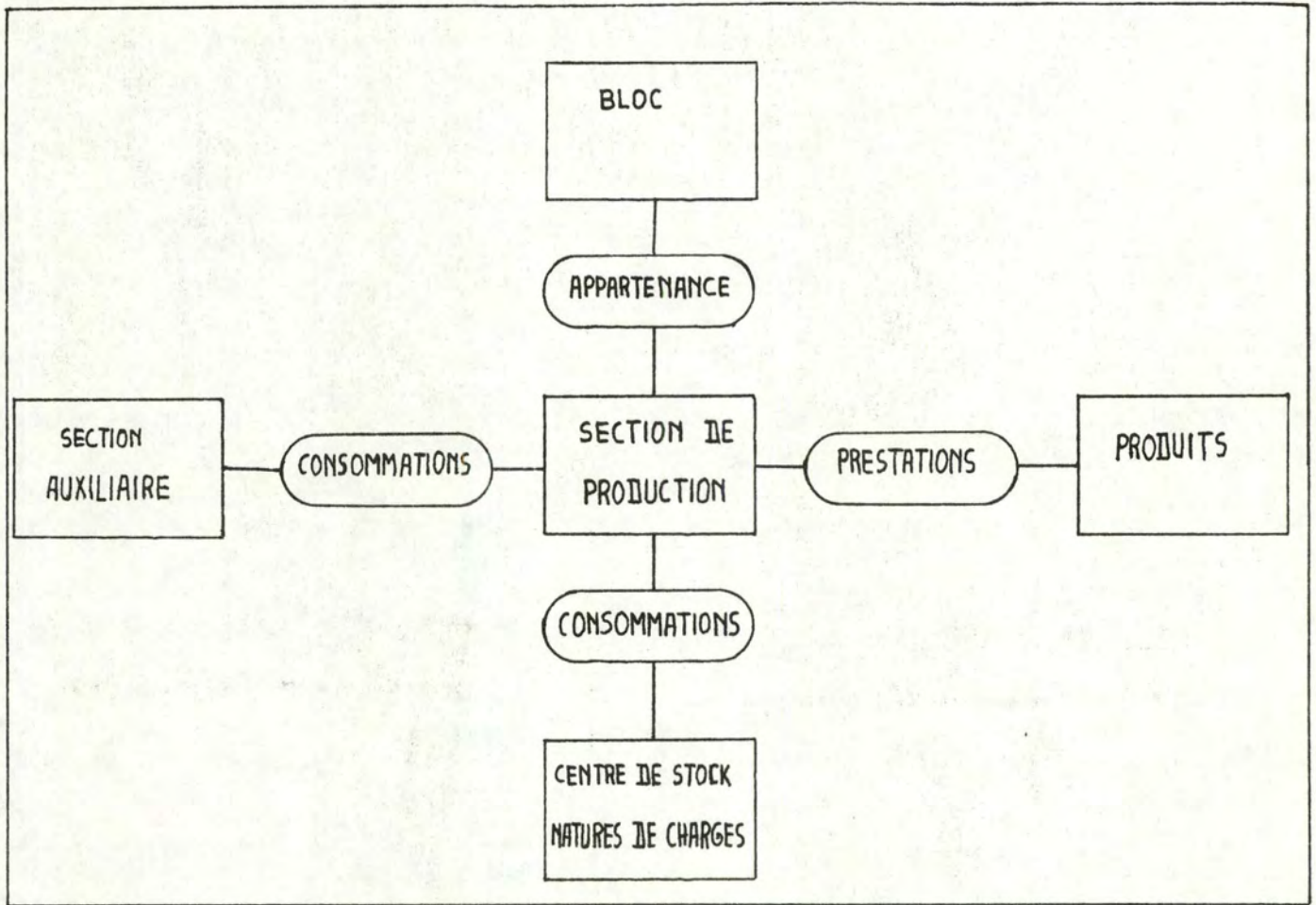


Fig. 1.4.

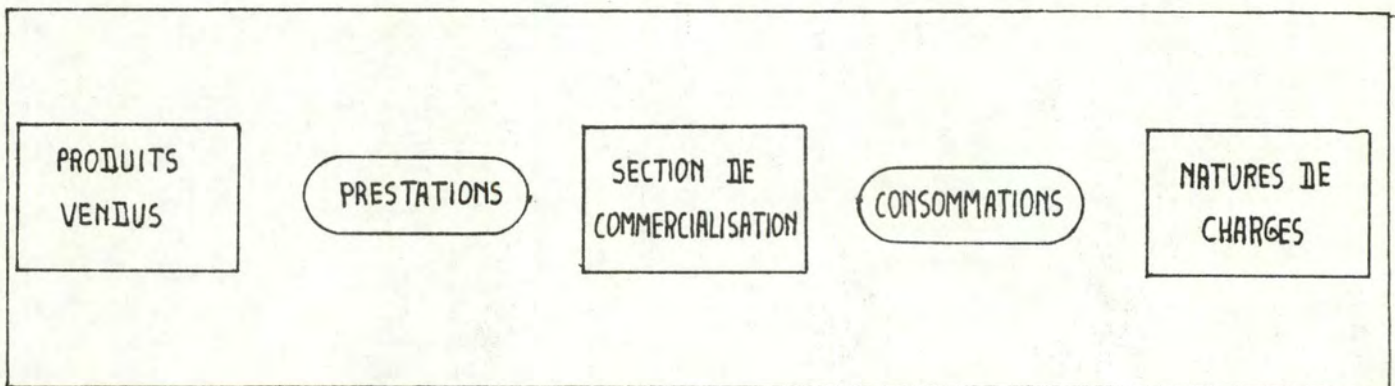


Fig. 1.5.

- associations relatives à une section principale de commercialisation (fig. 1.5)

types d'objets : - section de commercialisation (ou section de vente)
 - produits vendus
 - natures de charges

les types de propriétés de ces objets constituent les articles correspondants de la banque de données.

types d'associations : - prestations
 - consommations

les types de propriétés de celles-ci figurent dans les articles associés aux standards correspondants.

cardinalité de liaison : n - n pour les deux associations.

- associations relatives à une section auxiliaire : (fig. 1.6)

types d'objets : - section auxiliaire
 - natures de charges
 - section principale de production
 - usine
 - section administrative

les types de propriétés de ces objets constituent les articles correspondants de la banque de données.

types d'associations : - consommations
 - prestations
 - appartenance

excepté pour l'appartenance, les types de propriétés figurent dans les articles associés aux standards correspondants.

cardinalité de liaison : 1 - n pour appartenance (usine-section)
 n - n pour les autres.

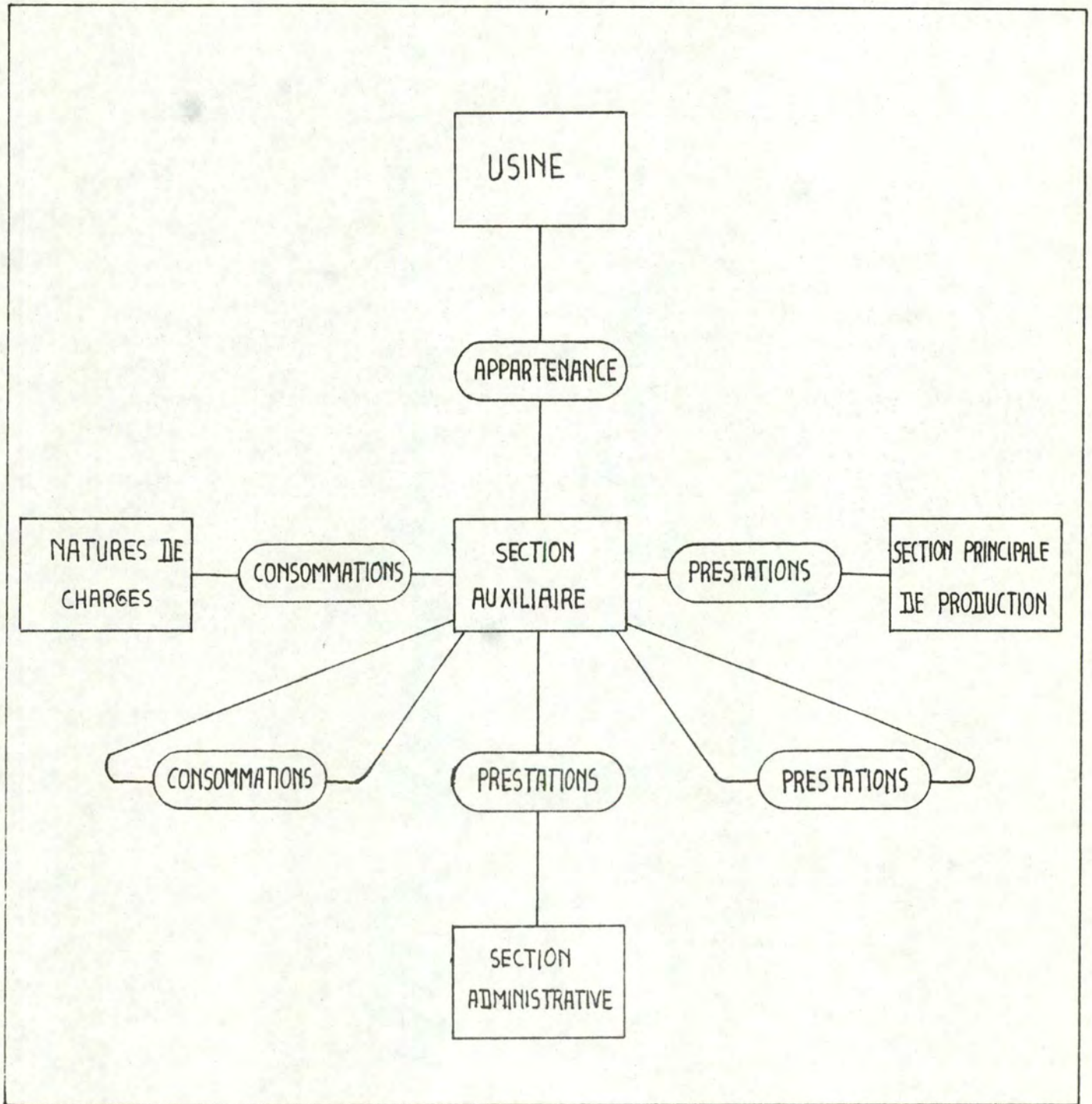


Fig. 1.6.

- associations relatives à une section administrative (fig. 1.7)

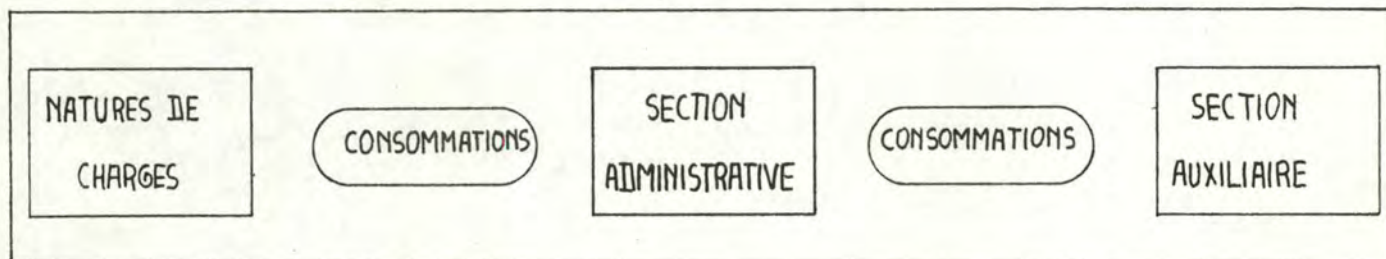


Fig. 1.7.

types d'objets : - section administrative
 - natures de charges
 - sections auxiliaires

les types de propriétés constituent les articles correspondants de la banque de données.

type d'association : consommation

les types de propriétés figurent dans les articles associés aux standards correspondants.

cardinalité de liaison : n - n

- associations relatives aux produits physiques (fig. 1.8)

types d'objets : - produit physique
 - centre de stock de produits expédiés
 - centre de stock
 - produits d'autres départements
 - centre de stock de produits des blocs amonts
 - section de production
 - matières premières

les types de propriétés constituent les articles correspondants de la banque de données.

types d'associations : - liaison
 - consommations
 - appartenance

excepté pour appartenance, les types de propriétés figurent dans les articles associés aux standards correspondants.

cardinalité de liaison : 1 - n pour appartenance (bloc-produit physique)

n - n pour les autres.

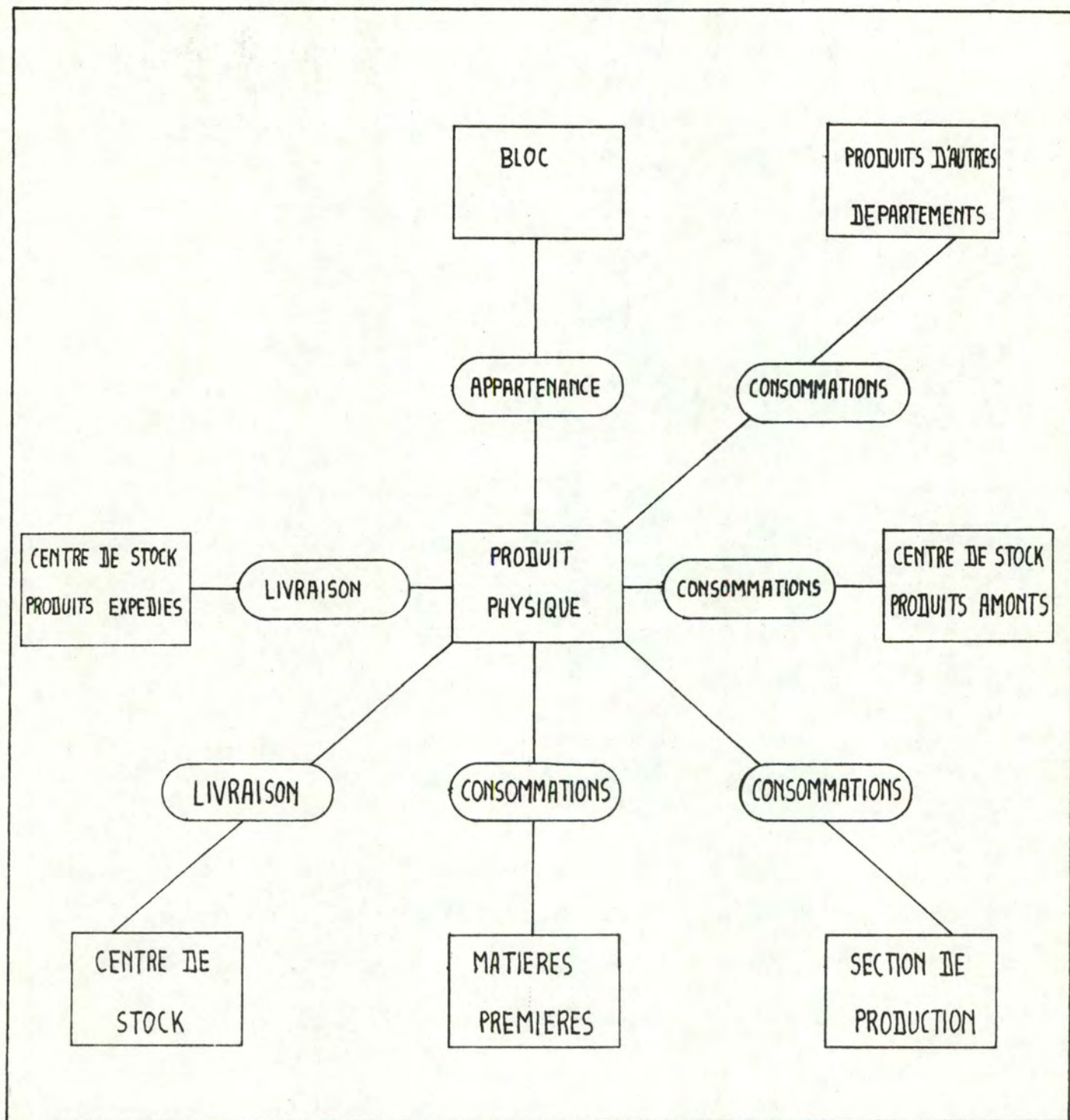


Fig. 1.8.

- associations relatives aux centres de stock de produits.
(fig.1.9)

types d'objets :

- centre de stock
- département
- produits
- bloc
- usines extérieures (ceci regroupe autres usines et autres entités)

les types de propriétés constituent les articles correspondants de la banque de données.

types d'associations :

- appartenance
- entrée
- cession
- livraison

excepté pour appartenance et cession, les types de propriétés figurent dans les articles associés aux standards correspondants.

cardinalité de liaison :

- 1 - n pour appartenance (bloc - centre de stock)
- 1 - n pour cession (centre de stock - département)
- n - n pour les autres.

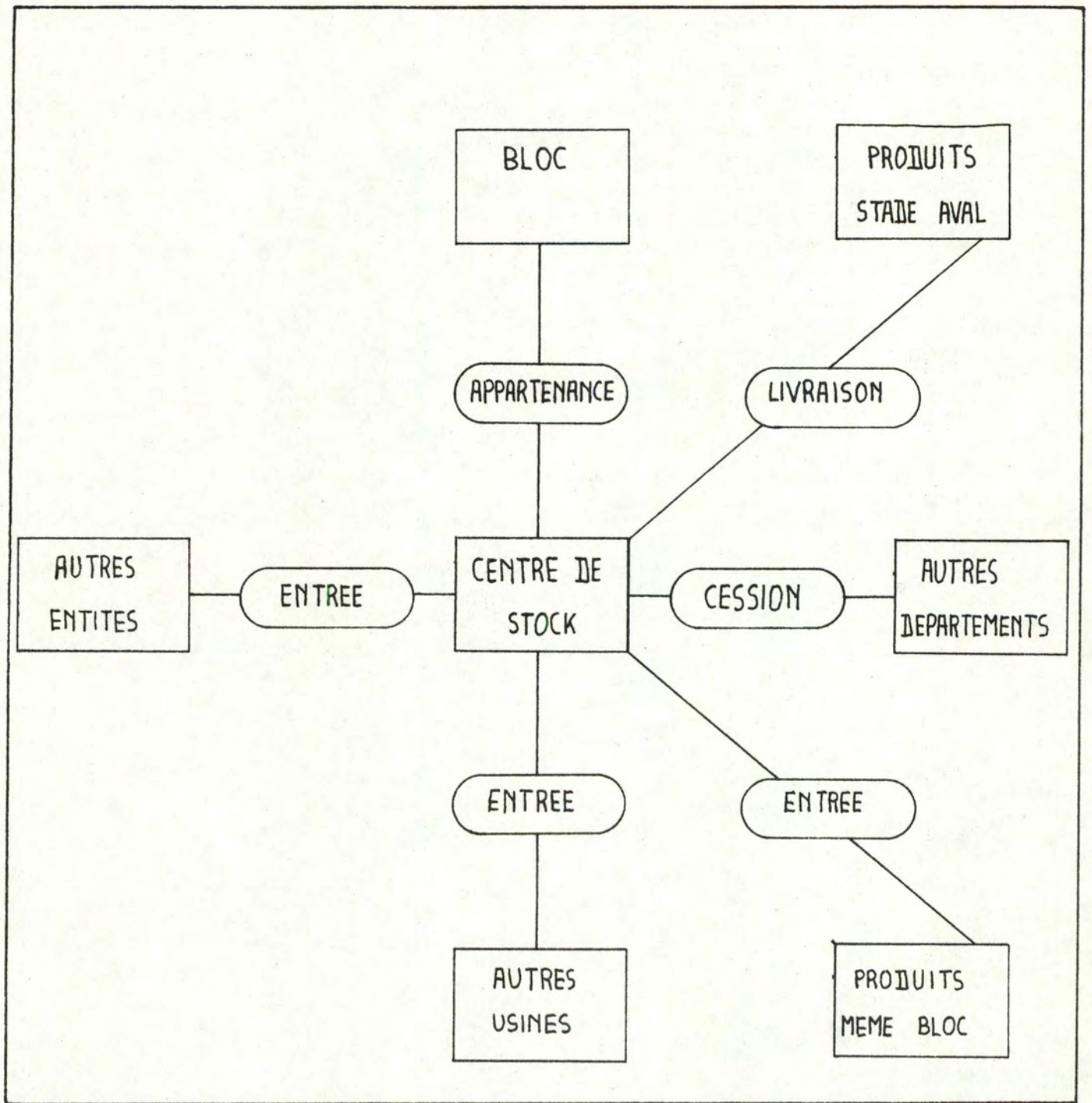


Fig. 1.9.

- associations relatives au centre de stock de produits expédiés.
(fig. 1.10)

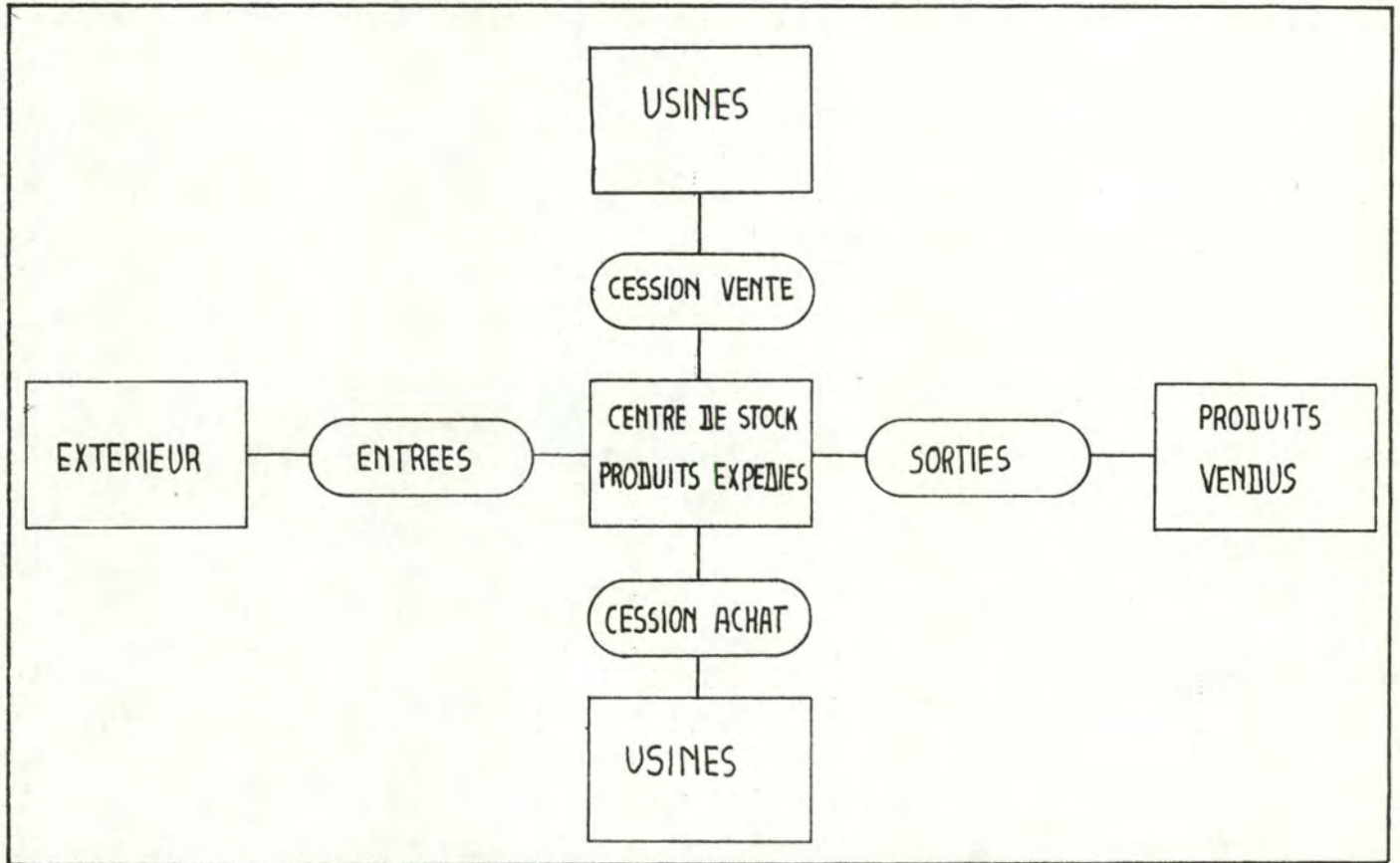


Fig. 1.10.

types d'objets : - centre de stock
- usine
- produit vendu
- échanges avec extérieur

les types de propriétés constituent les articles correspondants de la banque de données.

types d'associations : - sortie
- entrée
- cession achat
- cession vente

excepté entrée, les types de propriétés figurent dans les articles associés aux standards correspondants.

cardinalité de liaison : 1 - n pour entrée (centre de stock-
extérieur)

n - n pour les autres

- associations relatives aux stocks de fournitures primaires.

(ceci regroupe matières premières et natures de charges)
(fig. 1.11)

- types d'objets :
- usine
 - stock FP usine
 - stock FP destinées à la vente
 - stock FP toutes usines
 - sections principales de commercialisation
 - sections principales de production
 - sections auxiliaires
 - sections administratives
 - produits

les types de propriétés constituent les articles correspondants de la banque de données.

- types d'associations :
- appartenance
 - entrée
 - consommations

excepté entrée et appartenance, les types de propriétés figurent dans les articles associés aux standards correspondants.

- cardinalité de la liaison :
- 1 - n pour appartenance (usine-
stock usine)
 - 1 - n pour entrée (stock toutes
usines - stock FP usine de vente)
 - n - n pour consommations

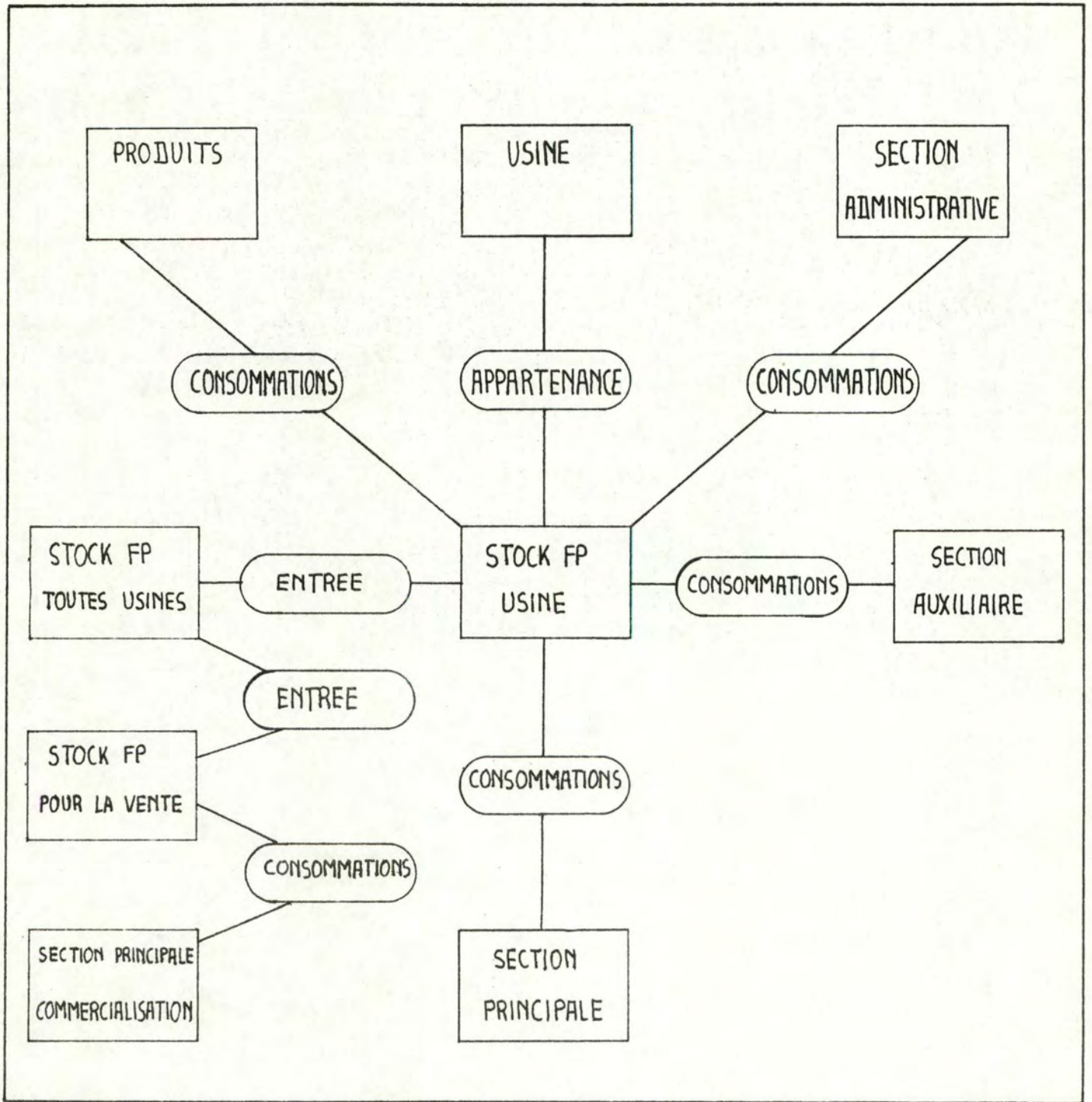


Fig. 1.11.

- associations relatives au bloc et au département. (fig. 1.12)

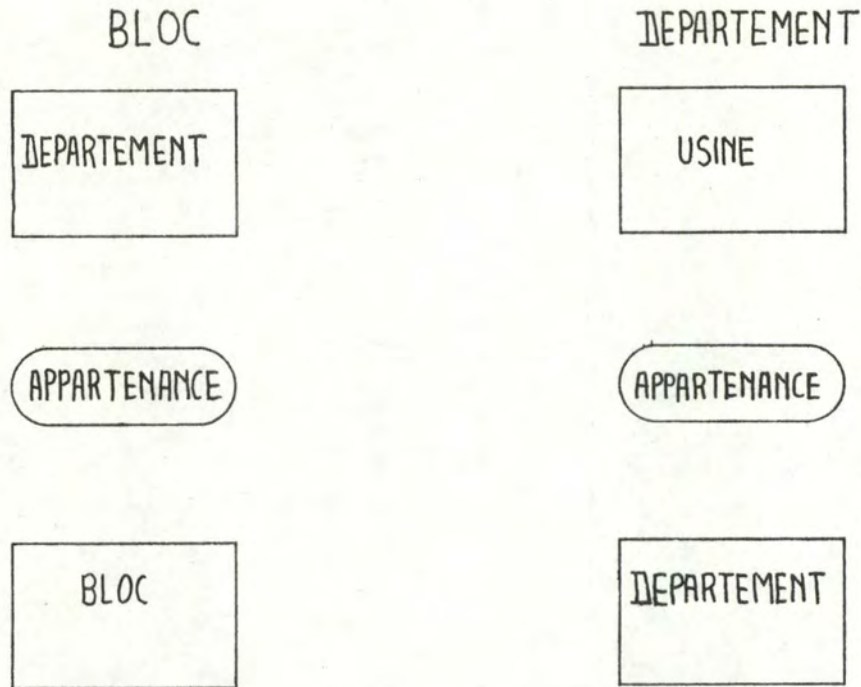


Fig. 1.12.

types d'objets : - bloc
 - département
 - usine

les types de propriétés constituent les articles correspondants de la banque de données.

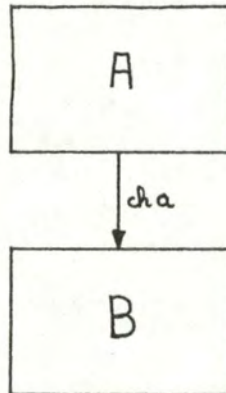
type d'association : appartenance

cardinalité de liaison : 1 - n (département - bloc et usine -
 département)

1.2.4. Modèle I.D.S.

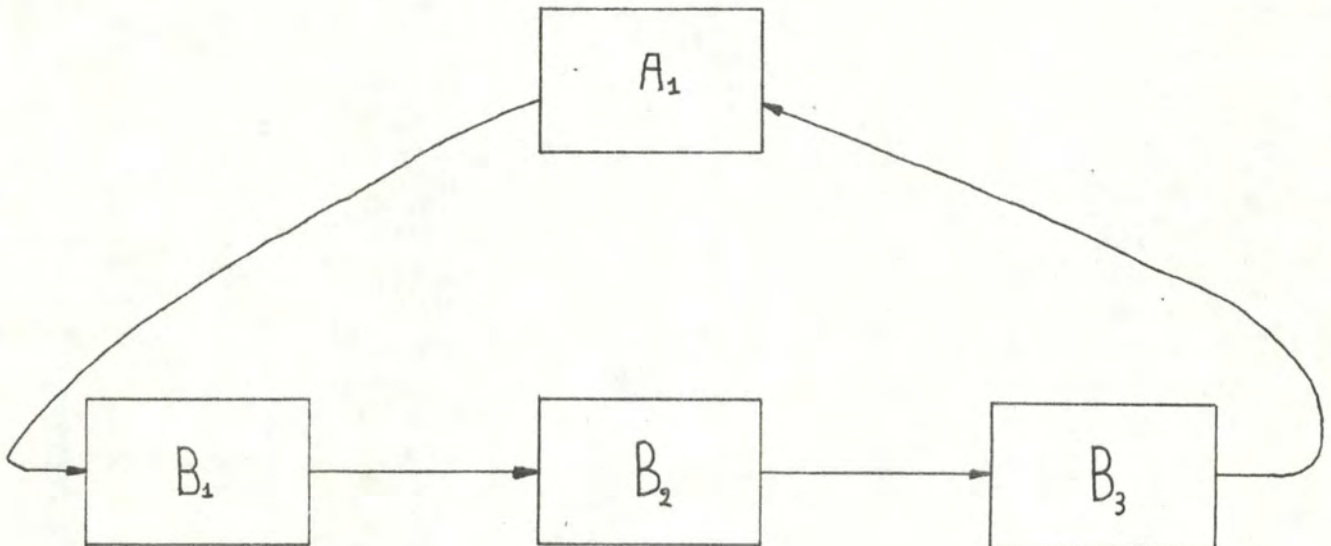
Ce modèle reprend toutes les associations décrites au paragraphe précédent où les associations :

- du type 1 - n : seront représentées par un schéma du type suivant:

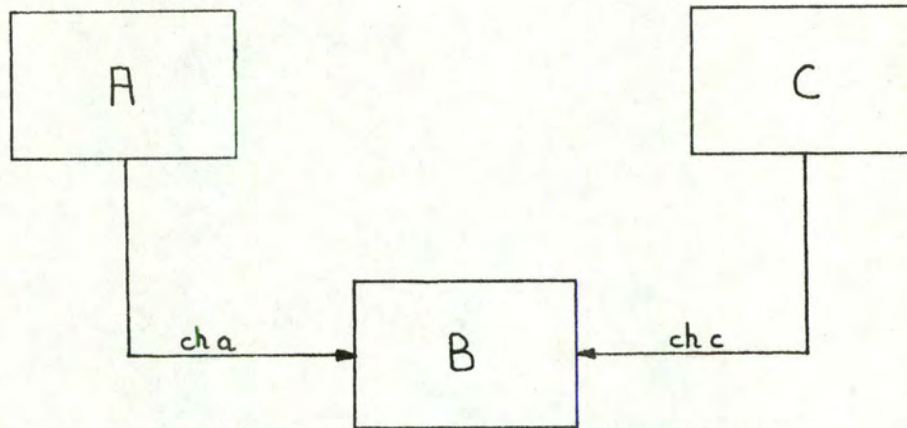


Toutes les valeurs des objets contenus (B) relatives à 1 valeur de l'objet contenant (A) lui sont rattachées au moyen de la chaîne a.

Exemple :

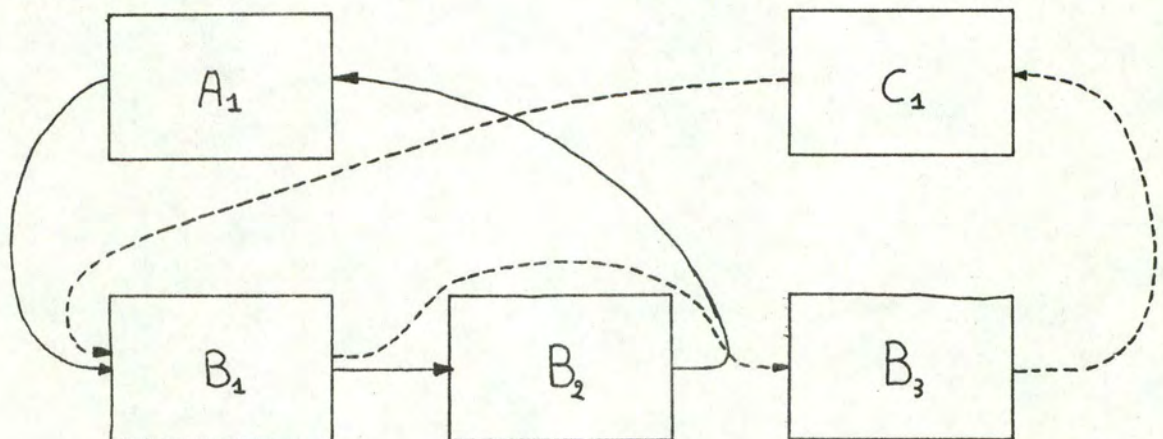


- du type n - n seront représentées par un schéma du type suivant:



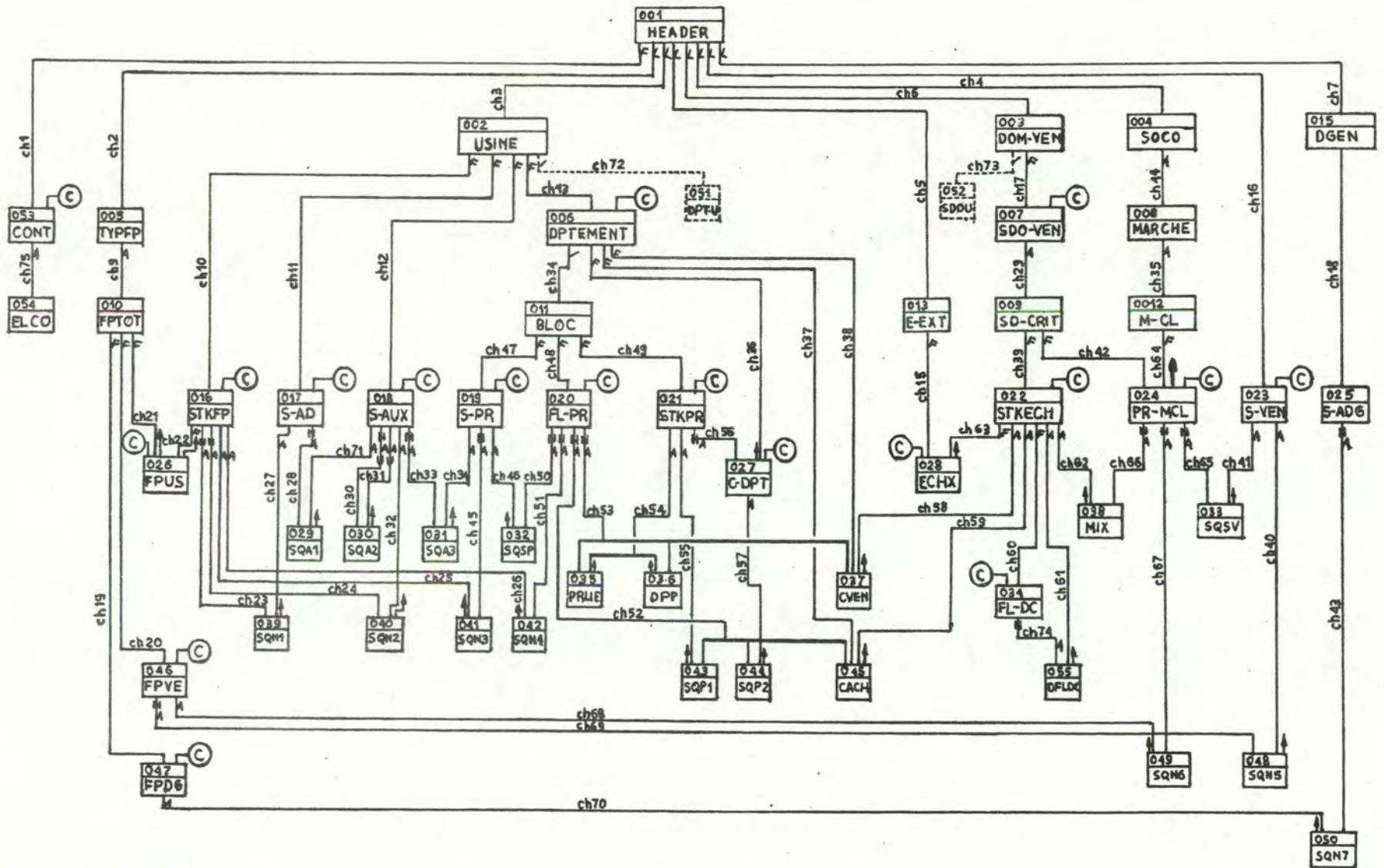
un ensemble de valeurs de l'objet B sera reliée à 1 valeur de l'objet A au moyen de la chaîne a et sera relié à 1 valeur de l'objet C par la chaîne c. Une valeur A_1 de l'objet A est associée à une valeur C_1 de l'objet C s'il existe une valeur B_1 de l'objet B qui figure à la fois dans la chaîne a et la chaîne c.

Exemple :



MODELE I.D.S.

MARS 76



CHAPITRE 2.

DEMARCHE SUIVIE POUR LA

DETERMINATION

DE LA STRUCTURE ET DU

CONTENU D'UNE REQUETE.

Examinons les questions qu'un utilisateur est susceptible de poser sur la banque de données. On n'envisagera pas les questions qu'on appellera "ponctuelles" et qui visent à obtenir une (ou plusieurs) donnée(s) appartenant à un ensemble d'occurrences d'une même entité sans demander une manipulation (calcul) de données. Ceci sous-entend que le langage doit être apte à répondre à une demande de ce type. On en dégagera la structure de la requête et les principaux opérateurs à définir.

2.1 Liste de questions susceptibles d'être posées par un utilisateur.

2.1.1. Questions relatives aux consommations des différents postes.

2.1.1.1. Sections administratives :

- quelles sont les consommations totales en fournitures primaires et unités d'oeuvre de sections auxiliaires de toutes les sections administratives de toutes les usines ou d'un sous-ensemble de sections administratives;
- quelles sont les consommations en fournitures primaires uniquement ou en unités d'oeuvre de sections auxiliaires par toutes ou une partie des sections administratives;
- quel est le niveau de consommation d'une fourniture primaire ou d'un groupe par rapport aux consommations totales de cette fourniture primaire ou ce groupe;
- pour chaque section administrative, quel est le niveau de consommation d'une fourniture primaire ou d'un groupe par rapport aux consommations totales de toutes les sections administratives en cette fourniture primaire ou ce groupe.

2.1.1.2. Sections auxiliaires :

- on peut reprendre le même ensemble de questions en remplaçant section administrative par section auxiliaire;
- pour toutes ces questions, on peut demander soit les consommations à caractère fixe, soit à caractère variable, soit les deux;
- quelles sont les dépenses de sous-traitance d'une ou plusieurs sections auxiliaires;
- pour chaque section, quel est le pourcentage d'utilisation de ses outils en comparant l'activité réelle avec la capacité minimale et maximale.

2.1.1.3. Sections de production :

- on peut demander les mêmes questions que pour les sections auxiliaires en excluant celle concernant la sous-traitance.

2.1.1.4. Produits :

- on peut demander les mêmes questions que pour les sections de production mais où une consommation d'unité d'oeuvre de section auxiliaire sera remplacée par une unité d'oeuvre de section de production;
- pour un stade de fabrication ou un département d'une usine, quelles sont les quantités de produits achetées à l'extérieur (autre usine ou autre entité);
- pour un produit, un stade ou un département d'une usine, quelles sont les quantités de produits achetées à un autre département de la même usine

2.1.1.5. Sections d'administration générale et sections de vente :

- même liste de questions que pour les sections administratives excepté celles relatives aux consommations d'unités d'oeuvre de sections auxiliaires qui n'existent pas ici.

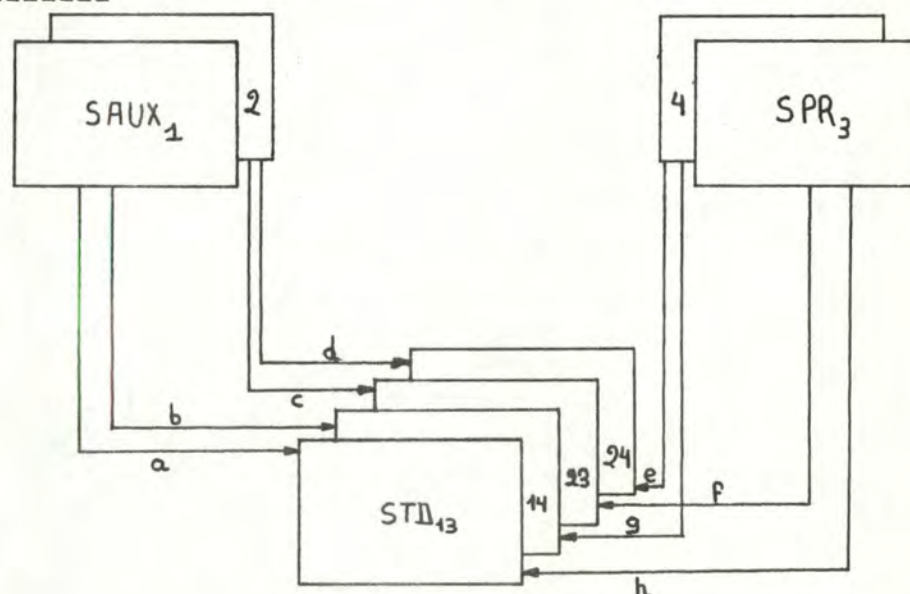
2.1.1.6. Imputation directe aux produits marché :

- même ensemble de questions que pour les sections administratives où les consommations d'unités d'oeuvre de sections auxiliaires sont remplacées par des consommations d'unités d'oeuvre de sections de vente.

2.1.2. Questions relatives aux prestations des différents postes.

Elles reprennent pour les sections auxiliaires, de production, produits et sections de vente les mêmes informations que celles utilisées dans les questions précédentes mais en les présentant différemment.

Exemple :



Supposons que les sections de production 3 et 4 consomment des unités d'oeuvre des sections auxiliaires 1 et 2. Pour connaître :

- les consommations d'unités d'oeuvre de sections auxiliaires par les sections de production, il faut parcourir les chaînes fc et ha (pour la section de production 3), de et bg, pour la 4;
- les prestations des sections auxiliaires 1 et 2 aux sections de production, il faut parcourir les chaînes ah et bg pour la 1, de et cf pour la 2.

2.1.2.1. Prestations des sections auxiliaires :

Quel est le montant des prestations de toutes ou d'un sous-ensemble de sections auxiliaires pour les sections administratives, auxiliaires ou de production.

2.1.2.2. Sections de production :

Quel est le montant des prestations de toutes ou d'un sous-ensemble de sections de production pour les produits.

2.1.2.3. Produits :

Quel est le montant des sorties d'un produit et vers quels stocks ces produits ont-ils été envoyés.

2.1.3. Situation des stocks :

- Que ce soit pour les stocks de produits fabriqués, les stocks d'échange (= artifice permettant le regroupement d'un même produit fabriqué dans différentes usines et destiné à la vente) et les stocks de fournitures primaires, on peut demander :
 - le montant des entrées en stock et cela par source d'approvisionnement;
 - le montant des sorties de stock;
- Pour les stocks de produits et de fournitures primaires uniquement, on peut demander l'état des stocks initiaux et finaux;
- Pour les stocks de fournitures primaires, quelle est la position de nos achats par rapport aux disponibilités.

2.1.4. Situation sur les marchés :**2.1.4.1. Examen du coût des produits :**

On peut donner par marché, par client, pour un produit ou plusieurs, les coûts totaux de ces produits si on leur ajoute :

- les coûts de distribution;
- les coûts de consommation d'unités d'oeuvre des sections de vente;
- la somme de ces deux coûts.

2.1.4.2. Examen du chiffre d'affaires :

- Quel est le montant des ventes (en quantité et/ou en valeur) d'un client sur un produit et comparer cela avec les ventes totales de ce produit;
- Quelle est la part du chiffre d'affaires d'un client sur un marché ou dans un domaine;
- Quelle est la dégradation du chiffre d'affaires pour un produit sur un marché, pour un client ou dans un domaine de vente;
- Dans quel(s) domaine(s) un client fait-il ses achats et leur montant.

2.1.4.3. Examen du résultat sur ventes et situation sur les marchés :

- Connaissant le prix de revient des produits, les frais liés à la vente et le prix de vente des produits, on peut connaître par client, par marché, par domaine de vente, le résultat sur ventes;
- On peut demander comment se situent les ventes par rapport aux contraintes de marché.

2.1.5. Analyse des prix de revient :

- on demande l'édition du prix de revient complet ou marginal d'un produit par type de fourniture primaire aux différents stades de production par lequel ce produit est passé;
- on demande l'édition du prix de revient complet ou marginal de l'unité d'oeuvre d'une section par type de fourniture primaire ou globalement.

2.2. Analyse des questions.

Pour chacune de ces questions, on peut effectuer une analyse qui permettra de mettre en évidence :

- la liste des données concernées;
- le chemin suivi pour y accéder;
- les opérateurs éventuellement utilisés dans la manipulation des données.

2.2.1. Description de quelques questions :

- donner les consommations en fournitures primaires (u,v) des sections auxiliaires (a,b) d'une usine x (figure 2.1)

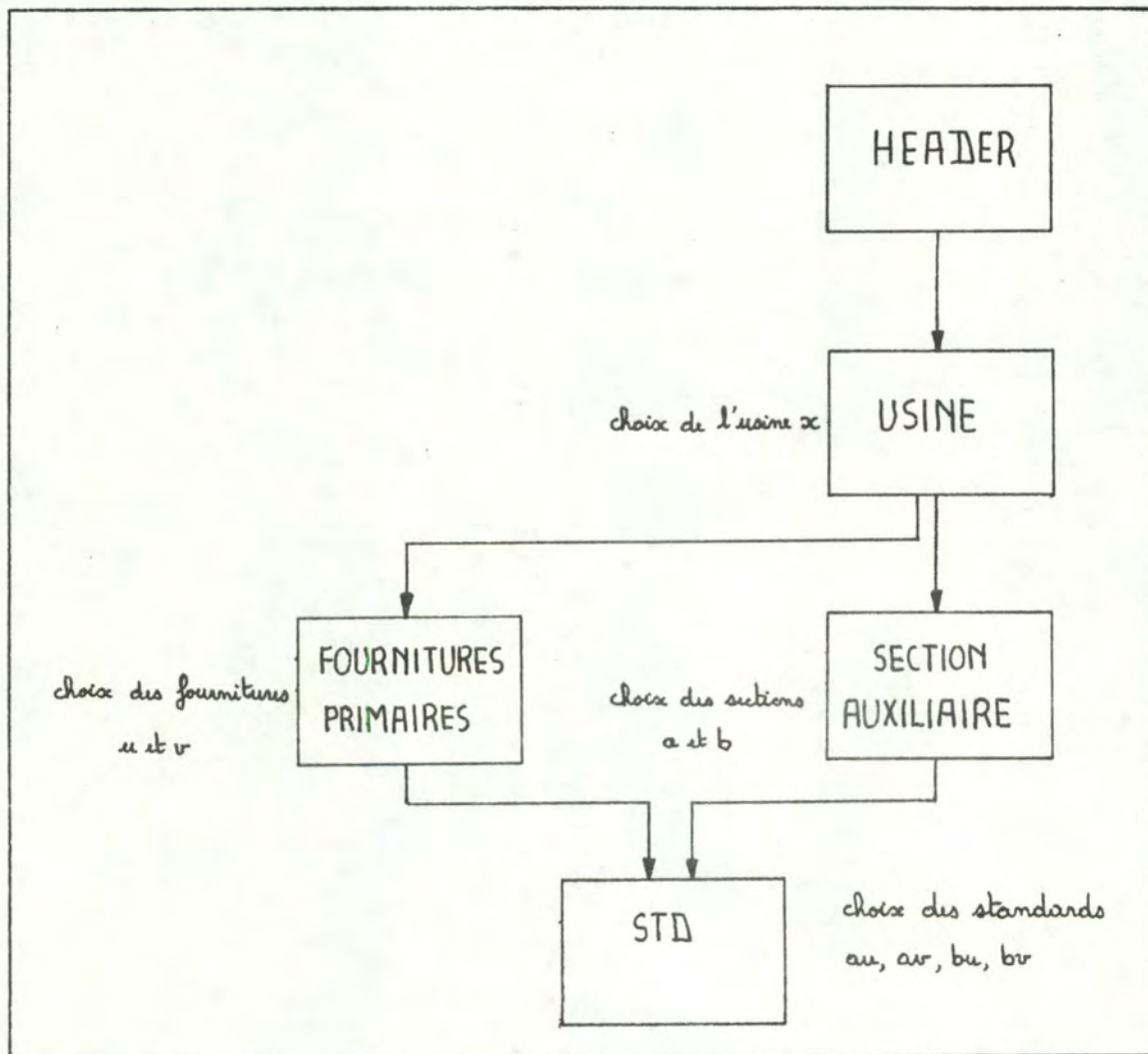


Fig. 2.1.

La démarche sera la suivante : pour une section auxiliaire donnée (par exemple a), on recherche quelles fournitures primaires, parmi celles citées, ont été consommées par la section choisie. Si a et b consomment les deux fournitures citées, la séquence d'accès aux standards sera celle énoncée plus haut. La quantité consommée sera obtenue sur base du niveau d'activité de la section et du standard, la valeur s'obtenant par la multiplication de la quantité par le prix unitaire de la fourniture. On a donc la séquence de calcul suivante :

- quantité = niveau d'activité * standard;
- valeur = quantité * prix unitaire.

On remarque la nécessité de mettre à la disposition de l'utilisateur un opérateur de multiplication.

- quel est le montant des prestations des sections auxiliaires a, b aux sections de production u,v du bloc x du département y de l'usine z, (figure 2.2.):

La démarche sera la suivante : choix d'une section auxiliaire et puis choix des sections de production pour lesquelles on suppose qu'elle a presté. Si la section a ne preste pas pour la section u, on aura la séquence suivante d'utilisation des standards : av, bu, bv.

Les calculs à effectuer seront les suivants :

- Quantité prestée = Niveau d'activité de la section de production * standard ;
- Valeur = Quantité * coût de l'unité d'oeuvre de la section auxiliaire.

- quel est le montant des consommations par le produit u du stade s en produits intermédiaires x,y du stade s' (=s-1) (figure 2.3.)

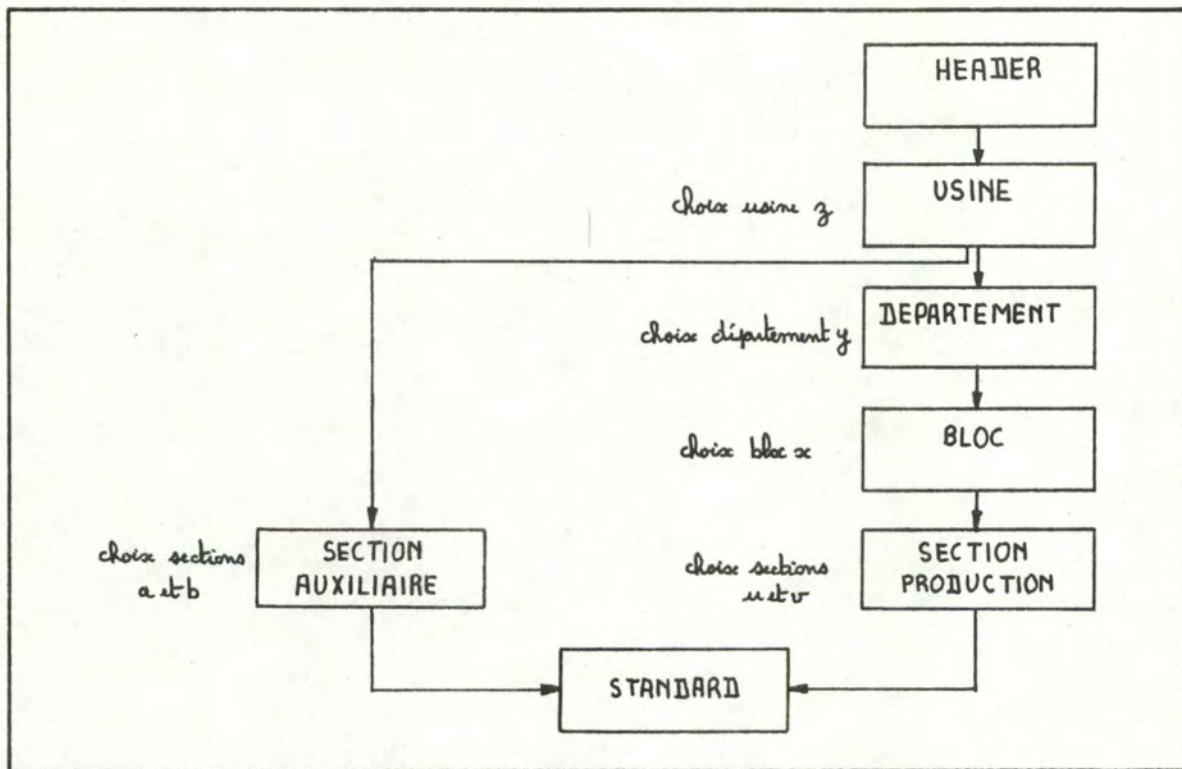


Fig. 2.2.

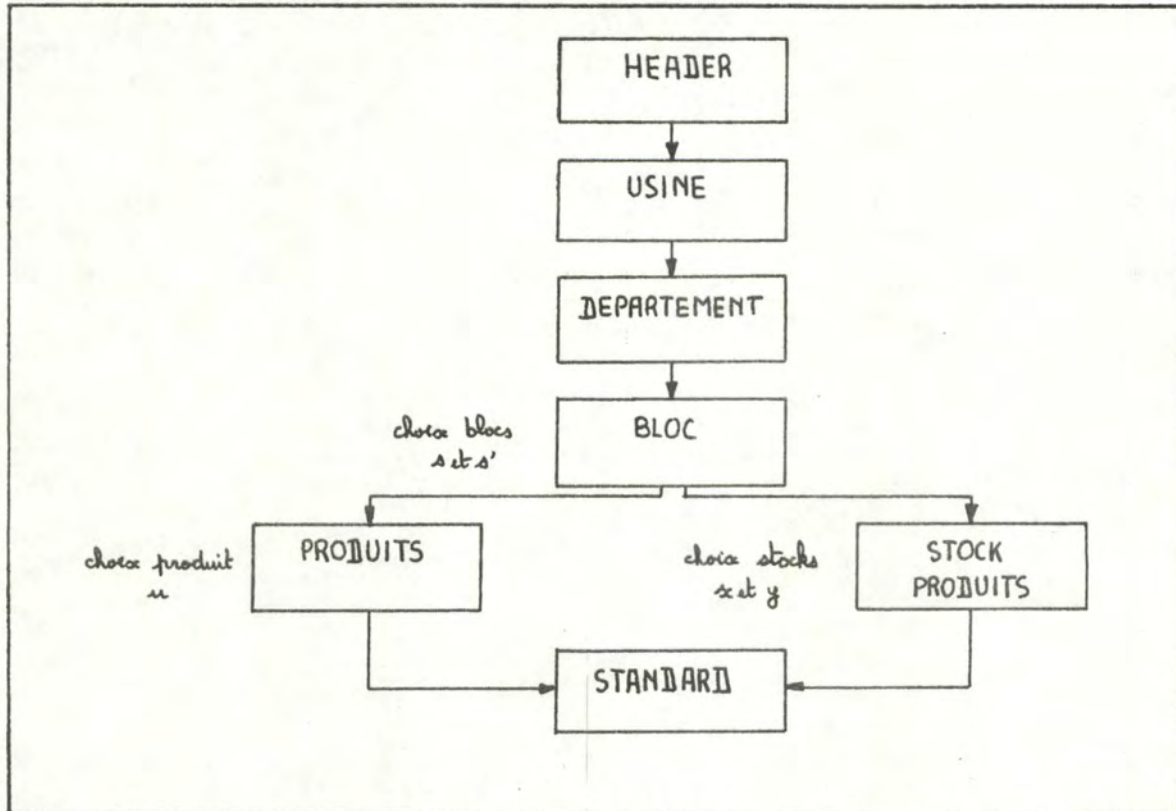


Fig. 2.3.

La démarche sera la suivante : on choisira d'abord le produit u du bloc s puis on accédera aux produits x et y du bloc s' . On accédera alors au standard qui existera ou non. S'il n'existe pas, cela signifie que le produit u ne consomme pas du produit x ou y . En supposant qu'il consomme les deux, on considérera les standards selon la séquence suivante : ux , uy . On remarque ici que la sélection du produit est indépendante de celle du stock puisque le premier appartient au bloc s et le second au bloc s' . Mais la sélection préalable d'une occurrence de ces deux éléments est indispensable pour pouvoir par la suite vérifier s'il existe bien une relation entre les deux. Le montant de la consommation sera obtenu sur base du niveau de production du produit u , du standard de consommation et du prix de revient du produit dans le stock et on aura la séquence de calcul suivante :

- quantité consommée = niveau de production de u
standard;
 - valeur = quantité prix de revient du produit en stock.
- quel est le montant des achats du client a en produit b et comparer son chiffre d'affaires au montant total des ventes de ce produit sur le marché x . (figure 2.4.)

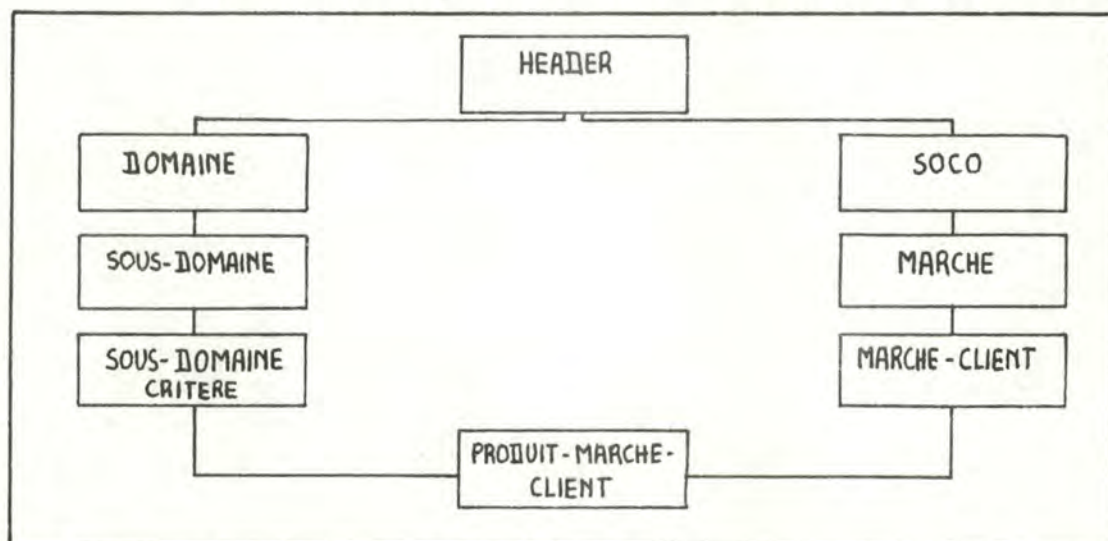


Fig. 2.4.

La démarche à suivre est la suivante : il faut d'abord effectuer le calcul du chiffre d'affaires du client a. Pour cela il faut accéder au produit marché, dont le code est bxa et on obtiendra le montant de l'achat sur base de la quantité vendue et du prix de vente. Pour faire la comparaison, il faut calculer le montant des ventes des produits b vendus sur le marché et cela pour tous les clients. On va donc obtenir la séquence de calcul suivante :

- chiffre d'affaires de a = quantité vendue de b
prix de vente de b.
- chiffre d'affaires du produit b sur le marché x =
(quantité de b vendue à i prix de vente de b)
client i
marché x
- % = (chiffre d'affaires de a 100)/chiffre d'affaires du
produit b sur le marché x.

On remarque l'apparition de trois nouveaux opérateurs de calcul :

- opérateur de division
- opérateur de calcul d'un pourcentage
- opérateur \sum qui sera appelé opérateur de cumul.
client $i \in$ marché x

2.2.2. Conclusions de cette description:

2.2.2.1. Liste des opérateurs à mettre à la disposition de l'utilisateur.

Suite à une analyse identique de toutes les questions, on en a déduit qu'il serait utile de donner à l'utilisateur la possibilité d'utiliser les opérateurs arithmétiques traditionnels à savoir l'addition, la soustraction, la multiplication et la division, mais aussi des opérateurs plus particuliers tels pourcentage et cumul. La description plus détaillée de ces deux derniers sera faite dans le chapitre suivant mais il est peut-être utile de justifier ici leur apparition. L'opérateur pourcentage a été mis en place pour éviter l'introduction de littéraux (comme 100) dans la définition d'un calcul. Le second a été introduit pour des raisons de structure de la banque de données. En effet, certains articles comme produit - marché - client sont détaillés de deux lignes hiérarchiques distinctes tout en ayant une codification propre. Pour l'obtention d'une donnée globalisée en tenant compte de critères propres à chacune des hiérarchies, il faudra sommer sur un certain nombre d'occurrences de l'article produit marché client (voir § 2.3.2.6.)

2.2.2.2. Paramètres à faire figurer dans la requête :

Il faudra indiquer les articles qui seront utilisés dans la requête. Ceci permettra de déterminer quelle structure hiérarchique devra être considérée. On leur associera les occurrences désirées. Elles seront spécifiées de 2 manières : la prioritaire étant celle relative à la codification.

Elle est prioritaire car c'est uniquement sur base de ce critère qu'il sera possible d'accéder aux occurrences des articles. On raffinera la sélection en lui ajoutant une possibilité de choix basée sur le test de valeurs de données.

Il faudra ensuite indiquer quels sont les calculs à effectuer. Pour cela, on utilisera les noms des opérateurs qui auront été définis. Le seul problème sera de rattacher les opérandes cités à l'article dans lequel on pourra en trouver la valeur.

Il reste enfin à définir les données qui seront éditées, où l'on rencontrera le même problème que pour les opérandes. Il sera en plus nécessaire de définir ici la manière dont on veut que les données soient imprimées c'est-à-dire définir la structure de l'état imprimé. Il sera possible sur base de ces renseignements de répondre à une question posée par l'utilisateur.

2.2.2.3. Types de questions pouvant être posées :

On aura deux types de questions :

- questions dites "ponctuelles" : ces questions ne concerneront qu'un article de la banque de données.
Ex.: on demande le niveau d'activité, le prix de revient marginal et complet de l'ensemble des N sections de production du département 2 de l'usine 8.
On sortira donc une liste d'informations suivante :

SECT 1	NIVACT 1	PR-MARG 1	PR-COMP 1
-	-	-	-
SECT N	NIVACT N	PR-MARG N	PR-COMP N

- questions relatives à une association : ce sont des questions qui, pour être résolues, nécessitent la mise en commun des données figurant dans deux articles distincts de la banque. Pour une question de réalisation, l'association est matérialisée par un troisième article qui sera donc également nécessaire pour pouvoir résoudre le problème.

Ex.: si on demande les consommations de fournitures primaires des sections auxiliaires, on considérera les articles section de production, stock de fournitures primaires et l'article standard matérialisant l'association.

- Dans chacune de ces questions on peut demander d'effectuer une comparaison avec une donnée de même nature figurant dans un article de niveau hiérarchique supérieur et qui sera donc plus globale.

Ex.: comparer pour les n sections administratives de l'usine 8 les consommations de fournitures primaires avec la consommation totale de ces sections.

On aura une liste d'informations du type :

(1)	(2)	(3)		
SECT-AD 1	DEP-FP 1	DEP-FP totale-SAD	% = (1)	100/(3)
-	-	-	-	-
-	-	-	-	-
SECT-AD n	DEP-FP n	DEP-FP totale-SAD	% = (2)	100/(3)

On peut également demander de faire une comparaison avec une donnée figurant dans un des articles déjà cités et possédant la même structure de sélection. On ne mentionnera pas dans ce cas une seconde fois l'article (sauf dans le cas particulier de l'opérateur

CUMUL : § 3.3.2.6)

DEUXIEME PARTIE

DEFINITION ET PROPRIETES DU LANGAGE.

Nous présenterons tout d'abord la structure générale d'une requête qui se compose de trois parties : la partie localisation, la partie calcul qui est facultative et la partie édition. Ensuite nous les reprendrons une à une pour les définir dans le détail. Lorsque la définition du langage est terminée, nous présenterons la grammaire formelle lui correspondant. Sur base des propriétés de cette dernière, nous définirons l'algorithme d'analyse. La grammaire étant en fait régulière, nous pouvons la représenter au moyen d'un automate fini déterministe et l'algorithme consiste en un parcours d'un tableau le représentant.

CHAPITRE 3.

SYNTAXE DU LANGAGE.

3.1 Aperçu général de la structure d'une requête.

La requête se subdivisera en trois parties qui seront :

- localisation : on y mentionnera les noms des articles qui seront traités dans la requête. Ils seront précédés par un libellé du type E_i (i variant de 1 à 4) qui permettra par la suite de rattacher toute donnée à l'article dans lequel elle figure. Ils seront suivis des critères de sélection des occurrences qui porteront en premier lieu sur la codification des articles (cette partie sera toujours présente) et puis porteront sur les données figurant dans chaque article (ceci étant facultatif);
- calcul : on fera figurer dans cette partie toutes les manipulations de données qui se borneront à un ensemble assez restreint de calculs arithmétiques. Chaque donnée calculée qui contiendra le résultat de l'opération, sera notée R_i . La liste des opérandes sera entre parenthèses derrière le nom de l'opérateur. Pour localiser l'article dans une occurrence duquel on pourra trouver la valeur de la donnée, on prefixera le nom de cette dernière par l'ensemble " E_i -" faisant référence aux articles cités dans la partie localisation. Toute donnée calculée peut être opérande d'une autre donnée calculée. Cette partie est facultative;
- édition : on y trouvera trois paramètres :
 - ETAT = ce paramètre permettra à l'utilisateur de définir le contenu de l'en-tête de son état budgétaire. Ce sera fait au moyen d'éléments figurant dans la partie codification d'un article (voir structure d'un article fig. 3.1)
 - SOUS-TOTAL = l'utilisateur y mentionnera les niveaux auxquels ils désirent voir apparaître des totaux intermédiaires. Ceci sera également indiquer au moyen d'éléments figurant dans la partie codification d'un article. Ce paramètre est facultatif.

- LISTE = l'utilisateur y indiquera les données qu'il veut voir entrer dans la constitution d'une ligne à éditer. Ces données soit figureront dans la banque de données, soit seront le résultat d'un calcul.

- Exemples :

1) on désire connaître le niveau d'activité des sections de production du bloc 22 du département 1 de l'usine 8 et connaître l'écart par rapport à l'activité maximale.

DEBUT LOCALISATION

E1 : S-PR US = 8 DEPT = 1 BLOC = 22.

CALCUL

R1 = SOUST (E1-CAPX, E1-NIV-ACT).

EDITION

ETAT : BLOC

LISTE : E1-COS-PR, E1-LIB, E1-NIV-ACT, E1-CAPX, R1.

FIN

02	COS-PR	partie codification de la section de production
	03 NOUS	n° usine
	03 NODT	n° département
	03 NOBLOC	n° bloc
	03 NOSECTION	n° section
	03 FILLER	
02	D-S-PR	Donnée concernant la section de production
	03 CAPN	capacité minimale de la section
	03 CAPX	capacité maximale de la section
	03 LIB	libellé de la section
	03 LIB-UN	libellé de l'unité d'oeuvre
02	R-S-PR	Résultats de la simulation concernant la section
	03 NIVACT	niveau d'activité
	03 PRMARG	prix de revient marginal
	03 PRCOMP	prix de revient complet

fig. 3.1. Structure de l'article section de production (= S-PR)

Le mot BLOC (mis pour stade de production) dans le paramètre ETAT indiquera que l'en-tête sera constitué par le nom de l'usine, du département, du bloc. Les activités des sections figureront donc sur un même document.

- 2) on désire connaître les consommations en unités d'oeuvre des sections auxiliaires 222,223,224 de l'usine 8 par les sections de production du bloc 22 du département 1 de l'usine 8.

DEBUT LOCALISATION

E1 : S-PR US = 8 DEPT = 1 BLOC = 22

E2 : S-AUX US = 8 SECT = 222,223,224

E3 : CONSOMMATIONS-UO-SAUX : GLOBALES.

CALCUL

R1 = MULT (E1-NIVACT, E3-STD)

R2 = MULT (R1, E2-PR-COMP).

EDITION

ETAT = SECT

SOUS-TOTAL = BLOC

LISTE = E1-COS-AUX, E1-LIB, R1, E2-PR-COMP, R2.

FIN

Remarque : le tiret correspondant au E1- nom de donnée est le grand c'est-à-dire celui dont le code EBCDIC est 6D (109 décimal) tandis que celui qui apparaît dans PR-COMP est celui dont le code est 60 (96 décimal).

On obtiendra ici un état imprimé par section de production (indiqué par SECT) qui comprendra toutes les consommations de la section. Le total toutes sections du stade de production (= BLOC) réunies sera obtenu grâce au paramètre SOUS-TOTAL.

3.2 Définitions préliminaires.

Avant de présenter dans le détail la syntaxe du langage, il est utile de préciser :

- quelques définitions qui permettront de mettre en évidence quelques termes utilisés par la suite;
- les objectifs du langage.

3.2.1 Définitions :

- ##### 3.2.1.1. Générique de codification : on a montré fig 3.1. que tout article de la banque de données se subdivise en trois parties :
- codification relative à l'article;
 - données dont la valeur est obtenue à partir des fichiers de base de l'entreprise et qui comprennent des libellés ainsi que les éléments servant d'entrée au simulateur;
 - données résultats qui sont garnies par le simulateur.

La partie codification permet de résoudre deux problèmes :

- elle définit au niveau d'un ensemble d'occurrences d'un même article laquelle (ou lesquelles) on désire : Ex : au niveau de l'article S-PR (mis pour section de production), l'élément NOSECT (n° de section) permet de préciser les valeurs désirées;
- elle définit également la structure dans laquelle l'article désiré s'insère :
Exemple : - pour l'article S-PR, les éléments US (= usine), DEPT (= département), BLOC =(stade de production) insèrent la section dans son environnement. Ces éléments sont importants pour la recherche car ils spécifient le chemin d'accès à l'article désiré;

- pour l'article produit-marché-client (PR-M-CL), les éléments DOM (= domaine : float, verre à vitre,...) SDO (sous-domaine)... définissent l'environnement produit de cet article tandis que M (marché), M-CL (client) définissent l'environnement marché. Ceci correspond en fait à deux chemins d'accès possibles à cet article.

Les éléments tels que US, DEPT, BLOC, SECT, DOM, SDO, M, M-CL constitueront les génériques de codification.

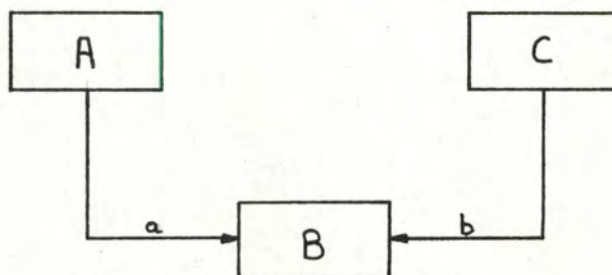
3.2.1.2. Identifiant : est l'ensemble des génériques de codification dont la valeur détermine de manière unique une occurrence d'un article.

Exemple : si la partie localisation d'une requête est la suivante :

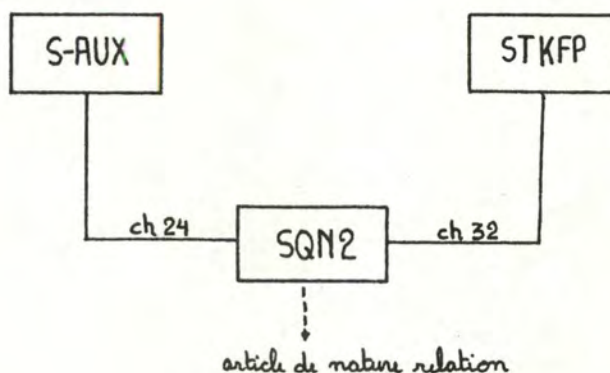
US = 8 DEPT = 2 BLOC = 30 SECT = 222,223
 on aura les identifiants suivants ;
 8230222 et 8230223

3.2.1.3. Dans ce qui précède, on a souvent parlé "d'article". On va préciser cette notion qui présente en fait deux natures :

- nature entité : est un article qui possède une codification propre sur quatorze caractères. Cela signifie que toute occurrence peut être atteinte en donnant son identifiant.
Ex: article section de production constitue une entité puisqu'il possède une partie codification propre composée du code usine, département, bloc, section et un identifiant sera composé de 7 chiffres et 7 blancs;
- nature relation : cette notion recouvre les articles auxquels on a jugé utile de ne point attribuer de codification propre. Ceci est dû à la nécessité de représenter une association de type (m-n) entre deux articles de nature entité. Il fallait passer par un article intermédiaire assurant la liaison entre les deux. L'identifiant d'une occurrence d'un tel article sera constitué de la concaténation des identifiants des deux occurrences des articles entre lesquels existe une association. Les deux sont nécessaires car pour vérifier si celle-ci existe, le système procédera de la façon suivante : on prendra l'identifiant de l'article A et en parcourant la chaîne a (= technique physique de représentation d'une relation maître - esclave), dont il est le maître, on prendra chaque occurrence de l'article B et on essaiera de trouver en remontant la chaîne b une occurrence de l'article c dont l'identifiant est égal au second donné. Si on obtient ce résultat, il existe une association, sinon il n'en existe pas entre les deux occurrences.



Exemple : si on désire connaître les consommations des sections auxiliaires en matières premières, il faudra considérer les articles S-AUX, STKFP et le standard de consommation (SQN2) qui seront disposés comme suit :



Remarque : par la suite, nous appellerons article de nature entité, entité et article de nature relation, entité relation.

3.2.2. Objectifs du langage :

- Ce langage a pour but de permettre à l'utilisateur :
- de choisir les données qu'il veut éditer c'est-à-dire non seulement de spécifier celles qui figurent dans la banque et qu'il désire voir éditées mais aussi de créer au moyen d'opérateurs adéquats (on se bornera ici à quelques opérateurs de calcul) de nouvelles données;
 - de définir le format de sortie des données c'est-à-dire leur disposition dans l'état imprimé.

3.3. Syntaxe du langage.

Toute requête devra toujours être comprise entre un "DEBUT" et un "FIN". C'est sur base de ces éléments que sera déterminé l'ensemble des données qui entreront dans la constitution des états imprimés. On va développer successivement chacune des trois parties constituant la requête.

3.3.1. Partie " LOCALISATION "

La localisation comprend deux éléments essentiels :

- désignation des entités interrogées dans la requête en spécifiant leur nom;
- critères de sélection des occurrences de ces entités qui seront de deux types :
 - sélection sur base des valeurs de leurs identifiants;
 - sélection sur base de comparaison des valeurs de données figurant dans ces entités avec une valeur fixée par l'utilisateur.

3.3.1.1. Localisation d'une entité par son nom :

Etant donné que toute donnée figurant dans la banque n'est pas unique mais qu'elle l'est par entité, il est donc nécessaire de spécifier les entités considérées dans la requête et de rattacher toute donnée mentionnée à l'entité dans laquelle elle figure. On procédera de la manière suivante :

- tout nom d'entité sera précédé d'un symbole "Ei";
- tout nom de donnée sera préfixé d'un symbole "Ei-".

La correspondance entre donnée et entité à laquelle elle appartient sera donc réalisé au moyen de ce symbole. La valeur du "i" n'étant pas quelconque, on va préciser son utilisation:

- pour une question ponctuelle : on utilisera au maximum deux entités. La première sera précédée de "E1" tandis que la seconde (qui est facultative) sera précédée de "E4";
- pour une question relative à une association: on utilisera trois ou quatre entités. Sachant que la quatrième est facultative, elles seront précédées dans l'ordre de E1, E2, E3 et E4.

Remarque : le nom de l'entité mentionné après E3 sera remplacé par le nom de l'association existant entre les deux premières entités.

Exemple : 1) on demande les niveaux d'activité et la capacité maximum d'activité des sections auxiliaires de l'usine 8:

```
DEBUT LOCALISATION
      E1 : S-AUX : US = 8.
CALCUL
      R1 = ADD(E1-NIV-ACT-FIXE, E1-NIV-ACT-VAR).
EDITION
      ETAT : US
      LISTE: E1-COS-AUX, E1-LIB, R1, E1-CAPX.
```

FIN

- 2) on demande les consommations fixes en la fourniture primaire 222 du type 20 pour les mêmes sections et de comparer par rapport à la quantité totale consommée de cette fourniture.

```
DEBUT LOCALISATION
      E1 : S-AUX : US = 8
      E2 : STKFP : US = 8 DEPT = 3 TYPFP = 20
          NAT = 222 .
      E3 : CONSOMMATIONS-FP : FIXES.
CALCUL
      R1 = MULT(E1-NIV-ACT-FIXE, E3-STD)
      R2 = MULT(R1, E2-PRX-CONS-MOY)
      R3 = PRCT(R1-Q-TOT-CONS).
```

EDITION

ETAT : SECT

LISTE: E2-COSTKFP, E2-LIB, R1, E2-PRX-CONS-
MOY, R2, R3.

FIN

3.3.1.2. Critères de sélection des occurrences :**3.3.1.2.1 Sélection sur base des valeurs des identi-
fiants :**

Cette sélection consiste à obtenir une occurrence d'entité sur base de sa codification. On va profiter du fait que la codification d'une entité reprend toute la structure hiérarchique dans laquelle cette dernière s'insère. On ne spécifiera donc à un niveau donné que les codes s'y rapportant.

Exemple : si on considère une section de production, on remarque qu'elle fait partie d'un stade de production appartenant à un département d'une usine. La codification de la section de production devra donc refléter cette structure :

USINE	générique correspondant US
-------	----------------------------

DEPARTEMENT	DEPT
-------------	------

BLOC	BLOC
------	------

SECTION	SECT
---------	------

Elle sera donc identifiée univoquement si on spécifie chacun des éléments suivants : US/DEPT/BLOC/SECT. Ceux-ci seront suivis de leurs codes respectifs: (Ex:US=8 DEPT=2 BLOC=22 SECT=333 ce qui donne la section d'identifiant 8222333).

Ceci va poser trois problèmes :

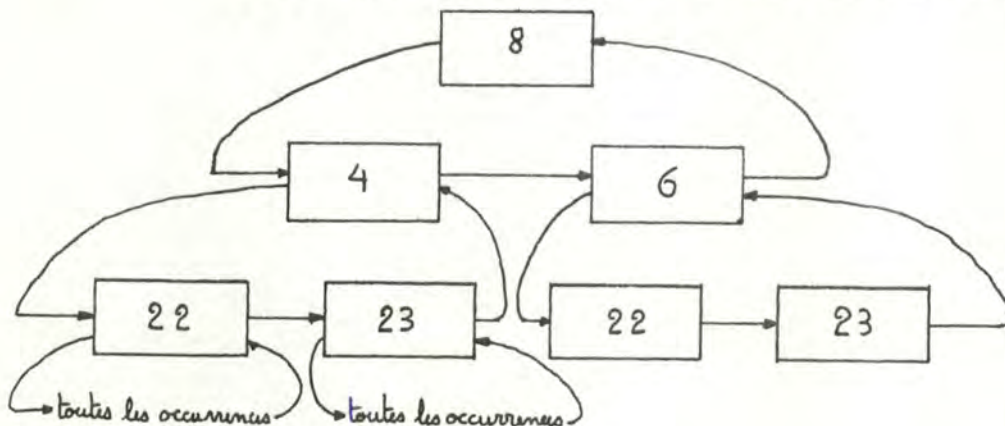
- comment exprimer une suite de codes relatifs à 1 seul niveau hiérarchique et comment reconstituer l'identifiant de l'entité?
 - comment réagir lorsque certains génériques sont omis, cela rendant l'identifiant incomplet?
 - certains génériques de codification différents recouvrent une même zone de l'identifiant ou un même générique couvre des zones différentes. Comment le spécifier?
- 1) expression des codes : les codes relatifs à un même générique seront séparés de ce dernier par le symbole "=" et séparés entre eux par le symbole ",". La reconstitution de l'identifiant d'une occurrence d'entité suivra une philosophie identique à celle de la constitution de la banque de données où un ensemble d'occurrences d'une entité sont reliées à une occurrence de l'entité du niveau hiérarchique supérieur.

Exemple : on désire obtenir les niveaux d'activité minima et maxima des sections de production des blocs 22 et 23 appartenant aux départements 4 et 6 de l'usine 8. On aura la requête suivante :

```

DEBUT   LOCALISATION
        E1 : S-PR: US = 8  DEPT = 4,6  BLOC = 22,23.
        EDITION
        ETAT = BLOC
        LISTE : E1-COS-PR, E1-LIB, E1-CAPN, E1-CAPX.
FIN
  
```

La reconstitution suivra le processus suivant :



Ceci peut poser le problème suivant : il est possible que le bloc 22 n'appartienne pas au département. On lancera donc une recherche de ce bloc qui se soldera par un échec qui sera signalé à l'utilisateur par un message d'erreur lui signifiant la non existence de ce bloc dans la banque de données alors que lui n'avait pas demandé cela. C'est un inconvénient, résultat du choix d'une syntaxe de forme familière, simple d'utilisation et indépendante de la structure de la codification.

Remarque : Alors que l'on désirait obtenir des informations concernant les sections de production, le générique SECT correspondant ne figure pas dans la partie localisation de la requête. L'omission d'un générique entraîne l'obtention de toutes les occurrences d'entités lui correspondant.

Dans le cas cité, on obtiendra donc toutes les sections de production appartenant au bloc 22 et 23 des départements mentionnés.

- 2) correspondance générique de codification - niveau hiérarchique
(fig. 3.2.: structure et schémas généraux de codification)

Exemple : on va dessiner la structure hiérarchique relative à un produit (FLPR)

structure hiérarchique générique correspondants structure de l'identifiant

HEADER		
USINE	US	US
DEPARTEMENT	DEPT	US/DEPT
BLOC	BLOC	US/DEPT/BLOC
PRODUIT	SECT/DOM/T/EP/CH/CT	US/DEPT/BLOC/SECT/ DOM/T/EP/TM/CH/CT/

STRUCTURE ET SCHEMAS GENERAUX DE CODIFICATION.

		PREMIER GROUPE											SECOND GROUPE.			
Article	Cas particuliers	Génériques	US	D ^t	BLOC		SECTION			DOM	T	Ep	TM	CH	C ^t	
			1	2	3	4	5	6	7	8	9	10	11	12	13	14
<u>Schéma 1</u> : " Section-Produit-Usine "																
1. Usine	-	US	x	-	-	-	-	-	-	-	-	-	-	-	-	
2 Département	-	DEPT	x	x	-	-	-	-	-	-	-	-	-	-	-	
3 Bloc	-	BLOC	x	x	x	x	-	-	-	-	-	-	-	-	-	
4 Direction générale	-	DGEN	1	-	x	x	-	-	-	-	-	-	-	-	-	
5 S ^{on} Production	-	S-PR	x	x	x	x	x	x	x	-	-	-	-	-	-	
6 S ^{on} Auxiliaire	-	S-AUX	x	x	-	-	x	x	x	-	-	-	-	-	-	
7 S ^{on} Administrative	-	S-AD	x	x	-	-	x	x	x	-	-	-	-	-	-	
8 S ^{on} Vente	-	S-VEN	0	9	-	-	x	x	x	-	-	-	-	-	-	
9 S ^{on} Adm.générale	-	S-ADG	1	-	x	x	x	x	x	-	-	-	-	-	-	
10 Flux Produits	*normal	FLPR	x	x	x	x	x	x	x	x	x	x	x	x	x	
	*caisse	FLPR	x	x	x	x	x	x	x	9	9	x	x	-	-	
11 Stock Produits	*normal	STKPR	x	x	x	x	x	x	-	x	x	x	x	x	x	
	*caisse	STKPR	x	x	x	x	x	x	-	9	9	x	x	-	-	
12 Cessions inter-départements	*normal	C-DPT	x	x	x	x	x	x	x	x	x	x	x	x	x	
	*caisse	C-DPT	x	x	x	x	x	x	x	9	9	x	x	-	-	
<u>Schéma 2</u> : " Fournitures "			US	D ^t	TYPFP		NAT			FOURNISSEUR						
			1	2	3	4	5	6	7	8	9	10	11	12	13	14
Type F.Primaire	-	TYPFP	-	-	x	x	-	-	-	-	-	-	-	-	-	
FP-Totale	-	FPTOT	-	-	x	x	x	x	x	x	x	x	x	x	x	
FP-Usine	-	FPUS	x	x	x	x	x	x	x	x	x	x	x	x	x	
FP-Vente	-	FPVE	0	9	x	x	x	x	x	x	x	x	x	x	x	
FP Adm. Gén.	-	FPDG	1	-	x	x	x	x	x	x	x	x	x	x	x	
Stock-FP	-	STKFP	x	x	x	x	x	x	x	-	-	-	-	-	-	
Entité Extér.	-	E-EXT	-	-	-	-	-	-	-	x	x	x	x	x	x	

Fig. 3.2.

On remarque que pour atteindre un niveau hiérarchique déterminé, il est nécessaire que l'identifiant reprenne tous les codes correspondant aux niveaux hiérarchiques supérieurs. C'est à cette seule condition que l'accès direct à une occurrence d'entité sera possible. Si maintenant, on omet un code relatif à un niveau hiérarchique supérieur, l'accès direct devient impossible. On parcourt alors les chaînes conduisant à l'occurrence d'entité désirée, à partir du niveau hiérarchique pour lequel l'identifiant est complet (c'est-à-dire qu'il reprend tous les codes nécessaires). Exemple : on désire obtenir des renseignements sur un bloc et la requête est de la forme :

$$E_1 : \text{BLOC} : \text{US} = 8 \quad \text{BLOC} = 22,23$$

L'accès direct se fera au niveau usine et puis on parcourera les chaînes.

Ceci introduit la particularité suivante : le générique département étant omis, le système va considérer tous les départements de l'usine 8 et rechercher dans chacun de ceux-ci les blocs 22 et 23. Ceci peut introduire une perte de temps par suite de recherches inutiles si les blocs 22 et 23 ne figurent pas dans tous les départements de l'usine 8. Si l'utilisateur veut éviter cette perte de temps, il doit citer explicitement les départements désirés.

Il reste le cas particulier de certains génériques n'ayant aucune raison d'être à certains moments. Il se présentera pour les entités FLPR, STKPR et CDEPT.

Exemple : le générique conditionnement (CT) n'a pas de raison d'être dans la codification d'un produit qui se trouve au stade fusion. Celui-ci ne possèdera que les critères domaine et teinte.

Ceci va avoir pour conséquence de rendre incomplet le niveau hiérarchique correspondant à produit. On ne pourra donc y accéder directement. Etant donné que plusieurs génériques représentent ce niveau, il faudra effectuer des tests de correspondance entre les codes des occurrences d'entités et les codes mentionnés par l'utilisateur pour les génériques figurant dans la requête. Ces tests ne doivent pas être effectués pour les génériques omis et dont on désire toutes les occurrences, ainsi que pour les génériques n'ayant aucune raison d'être. Ceci peut être indiqué de deux manières distinctes :

- pour les utilisateurs connaissant parfaitement la codification en ce sens qu'ils ont une connaissance parfaite de la position occupée par chacun des génériques (par exemple que le code domaine occupe les positions 8 et 9), on leur permettra de rassembler tous les critères d'un produit en cours de fabrication sous un seul générique qui sera PROD (= produit). Il sera équivalent à la suite des génériques SECT, DOM, T, EP, TM, CH, CT. Les codes que l'on juge inutile seront remplacés par une astérisque de même que ceux dont la valeur est indifférente (c'est-à-dire dont on veut toutes les occurrences).

Exemple : on désire obtenir les quantités de produits du domaine 11, toutes teintes, épaisseur 4 et 8 mm produites par la section 222 du bloc 3 du département 2 dans l'usine 8.

DEBUT LOCALISATION

E1 : FL-PR : US = 8 DEPT = 2 BLOC = 3

PROD = 22211*4***, 22211*8***.

EDITION

ETAT : BLOC

LISTE : E1-COFL-PR, E1-LIB, E1-NIV-PROD.

FIN

- pour les utilisateurs ayant une connaissance moins précise de la codification en ce sens qu'ils savent qu'un produit répond à certains critères tels domaine, épaisseur, teinte, mais sans savoir exactement la place occupée par ceux-ci dans l'identifiant, on leur laissera la possibilité de sélectionner les occurrences d'une entité en passant par l'intermédiaire des génériques. Ceux qui sont jugés inutiles ainsi que ceux dont la valeur est indifférente (c'est-à-dire dont on désire toutes les occurrences) ne seront pas mentionnés dans la requête.

Exemple : dans le cas de l'exemple précédent, on aura la requête suivante :

DEBUT LOCALISATION

E1 : FL-PR : US=8 DEPT=2 BLOC=3 SECT=222 DOM=11
EP=4,8.

EDITION

ETAT : BLOC

LISTE : E1-COFL-PR, E1-LIB, E1-NIV-PROD.

FIN

3) Problèmes du recouvrement de zones d'identifiant par plusieurs génériques.

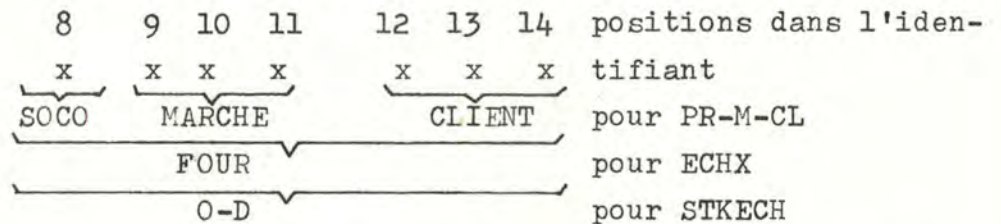
Le fait de permettre à l'utilisateur de spécifier le critère de sélection des occurrences des entités au moyen des génériques peut entraîner certains problèmes qui peuvent être de 2 types et considérés comme résultat des particularités de la codification.

- problème de type structurel : il se situe à deux endroits :
 - au niveau des FL-PR, STKPR, CDEPT : le code section est décomposé de deux manières différentes. Pour le FLPR, il se présente sur trois positions et représente la section. Pour le STKPR, il se présente sur 2 positions (les deux premières, la troisième étant à blanc) et indiquer la filière par laquelle le produit est passé. Pour CDEPT, les 2 premières positions représentent toujours le code filière tandis que la troisième représente le département receveur. On a donc une même zone qui se découpe et qui s'appelle de différentes façons. On a décidé d'introduire de nouveaux génériques qui seront :

5	6	7	positions dans l'identifiant
x	x	x	

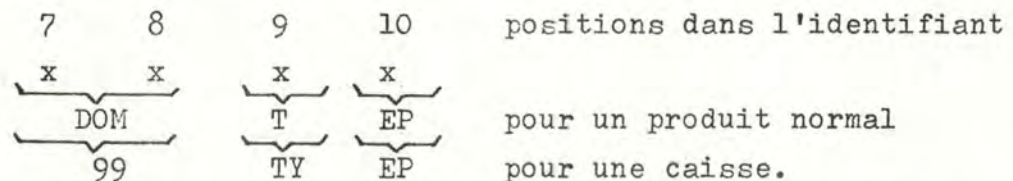
- au niveau du PRMCL, ECHX, STKECH: la partie du code représentant le marché et le client dans PRMCL représente :
 - le fournisseur pour ECHX;
 - une origine - destination pour STKECH.

On aura les appellations suivantes :



- problème de type occurrenceiel : ce problème est unique et apparaît dans le cas suivant : pour les entités FL-PR, STKPR et CDEPT, on trouve le cas particulier des caisses qui ont un code domaine spécial (=99) mais où le critère teinte signifiera type de caisse. Il est anormal dans ce cas pour un utilisateur d'utiliser le générique "T" (mis pour teinte). Celui-ci sera remplacé par "TY" (mis pour type de caisse).

On aura donc la structure suivante :



Remarque : dans le cas de la caisse, les critères TM, CH, CT n'ont pas de signification et ne seront donc jamais utilisés.

3.3.1.2.2. Sélection par comparaison des valeurs des données avec une valeur fixée par l'utilisateur.

Après avoir déterminé un sous-ensemble d'occurrences d'une entité au moyen de la sélection sur base d'identifiants, on peut demander que chacune d'entre elles vérifie un certain nombre de critères définis sur les valeurs de certaines données figurant dans chacune des occurrences de l'entité considérée.

Exemple : parmi les sections de production du bloc 30 du département 3 de l'usine 9, on peut exiger de ne prendre en considération que celles dont le niveau d'activité est supérieur à 200 unités d'oeuvre et le prix de revient complet inférieur ou égal à 150 F.

On aura la requête suivante :

DEBUT LOCALISATION

El : S-PR : US=9 DEPT=3 BLOC=30;

NIVACT >'200' ET PR-COMP <= '150'.

EDITION

ETAT = BLOC

LISTE= El-COS-PR, El-LIB, El-PR-MARG, El-PR-COMP.

FIN

Deux types d'opérateurs seront utilisés dans cette partie:

- les opérateurs de comparaison : ils sont les suivants:
 - égal qui sera représenté par "="
 - supérieur qui sera représenté par ">"
 - supérieur ou égal qui sera représenté par ">="
 - inférieur ou égal qui sera représenté par "<="
 - inférieur qui sera représenté par "<"
 - différent qui sera représenté par "!="
- opérateurs de combinaison : ils seront deux : ET,OU pour lesquels les priorités habituelles restent applicables à savoir que le ET est plus prioritaire que le OU. Celles-ci ne pourront être modifiées par l'usage de parenthèses puisqu'il est interdit de les utiliser.

Remarques :

- toute valeur servant de critère de choix sera comprise entre des guillemets simples.
Ex : le niveau d'activité supérieur à 200 unités d'oeuvre est traduit dans la requête par NIVACT >'200'
- cette seconde méthode de sélection est facultative et séparée de la première par un ";",
- Dans la banque de données existent deux types de données : les données simples et les données tabulaires.
Exemple : le prix de revient d'une section de production est présenté :
 - globalement sous le nom de PR-COMP;
 - sous forme tabulaire et ventilé par types de fournitures primaires sous le nom de TAB-PR-COMP (nom du tableau) et ELPR-COMP (nom d'une composante).

Le problème est de savoir comment indiquer le numéro de la composante désirée.

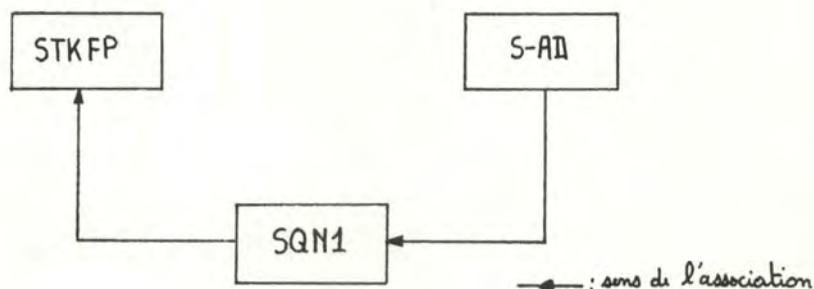
- une première solution est d'obliger l'utilisateur à indiquer le numéro I de la composante sous la forme ELPR-COMP (I) d'où nécessité pour lui de connaître la correspondance entre un type de fourniture primaire et la position qu'elle occupe dans le tableau. Ceci n'étant pas simple d'utilisation, on ne retiendra pas cette solution.
- une seconde solution est de laisser au système le soin de faire la correspondance entre le numéro de type de fourniture primaire et la position équivalente dans le tableau. Le problème est alors de savoir où introduire ce numéro de type. On pourrait désigner une composante sous la forme ELPR-COMP (numéro de type) mais il sera impossible de reconnaître l'existence ou non d'une telle donnée au cours de l'analyse de la requête car le nom qui figurera dans le dictionnaire des données (qui sera défini plus loin) sera ELPR-COMP. On rejettera également cette solution.

La sélection ne pourra donc s'effectuer que par rapport à des données simples.

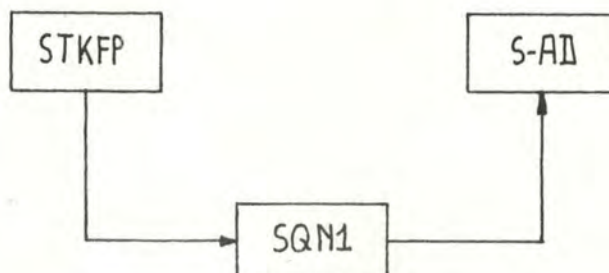
3.3.1.3. Associations reconnues dans le système.

Dans le cas d'une question relative à une association, il faudra donc mentionner trois entités à savoir les deux effectivement concernées par l'association et celle qui la matérialise physiquement. Etant donné que les noms désignant cette dernière ne sont pas très expressifs (Ex: le standard de consommation des fournitures primaires par une section administrative s'appelle SQN1), il a semblé utile de donner à l'utilisateur un autre moyen de désignation. Dans une première approche, on omettait le nom de l'entité relation étant donné que celle-ci était déterminée par les noms des entités entre lesquelles existent l'association et cela, quel que soit le sens dans lequel on la considère.

Exemple : - si on désire les consommations de fournitures primaires par les sections administratives, on a la structure suivante:



- si on désire les sorties de stock de fournitures primaires à destination des sections administratives, on a la structure :



- partie localisation dans chacun des cas :

LOCALISATION

E1 : S-AD:....

E2 : STKFP:....

LOCALISATION

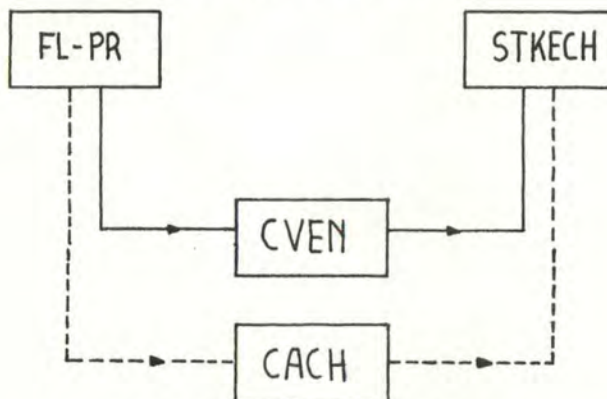
E1 : STKFP:....

E2 : S-AD:....

On remarque donc que l'entité origine de l'association sera mentionnée en E1 et l'entité destination sera en E2.

Cette façon de faire est correcte à la condition qu'il existe pour chaque association autorisée une seule entité relation lui correspondant. Or ceci n'est pas le cas. Il faut donc trouver un moyen de distinguer les associations entre elles. On a pris l'option, qui sera étendue à tous les cas, de désigner ces dernières par leur nom et c'est celui-ci qui figurera dans la requête.

Exemple : si on considère les entités FL-PR et STKECH, on aura deux associations possibles :



→ : sortie vers le stock d'échange des produits du dernier stade de fabrication

- - -> : consommation de produits en provenance de l'extérieur

Les associations qu'il sera possible d'utiliser dans le système d'interrogation de la banque de données sont reprises dans le tableau de la fig. 3.3. Voici maintenant la liste de ces dernières avec le nom qui devra figurer dans la requête :

1. Entrées en stock :
 - 1.1 Entrées-fournitures-primaires dans le stock correspondant.
 - 1.2 Entrées-produits-fabriqués : dans le stock correspondant.
 - 1.3 Entrées-stock-échange:
 - 1.3.1 entrées du produit correspondant du dernier stade de fabrication.
 - 1.3.2 entrées de produits déclassés.
 - 1.3.3 entrées de produits approvisionnés à l'extérieur.
 - 1.3.4 entrées de produits en provenance d'un département.

	FPUS	STKFP	SAD	SAUX	SPR	FLPR	FPVE	FPDG	SVEN	PRMCL	SADG	STKPR	CIPT	DEPT	STKECH	ECHX	FLIC
FPUS																	
STKFP	1.1		2.1.1	2.1.2	2.1.3	2.1.4											
SAD		3.1.1		4.1.1													
SAUX		3.1.2		4.1.2													
SPR		3.1.3	5.1.1	5.1.2	5.1.3												
FLPR		3.1.4			4.2							6.1.1	7.1.1		6.1.2		
FPVE								2.1.5	2.1.6			7.1.2			7.1.2		
FPDG											2.1.7						
SVEN						3.1.5			5.3								
PRMCL						3.1.6		4.3							8.1		
SADG								3.1.7									
STKPR						1.2							2.2.1				
CIPT						2.2.2											
DEPT						10.1.1											
STKECH						1.3.1							9.1.1		9.1.2		
ECHX						2.3.3									10.1.2		
FLIC															1.3.4	1.3.3	1.3.2
														2.3.4			2.3.1

Fig. 3.3.

2. Sorties de stock: 2.1 Sorties - fournitures - primaires
- 2.1.1 sorties de fournitures primaires vers les sections administratives
 - 2.1.2 sorties de fournitures primaires vers les sections auxiliaires
 - 2.1.3 sorties de fournitures primaires vers les sections de production
 - 2.1.4 sorties de fournitures primaires vers les produits fabriqués
 - 2.1.5 sorties de fournitures primaires vers les sections de vente
 - 2.1.6 sorties de fournitures primaires vers les produits marché
 - 2.1.7 sorties de fournitures primaires vers les sections d'administration générale
- 2.2 Sorties - produits - fabriqués
- 2.2.1 sorties de produits fabriqués vers d'autres départements appartenant à la même usine
 - 2.2.2 sorties de produits fabriqués vers les stades ultérieurs
- 2.3 Sorties - stock - échange
- 2.3.1 sorties de produits déclassés
 - 2.3.2 sorties de produits vers les marchés
 - 2.3.3 sorties de produits vers les stades de fabrication
 - 2.3.4 sorties de produits vers les départements
3. Consommations : 3.1 Consommations - fournitures - primaires :
- 3.1.1 consommations de fournitures primaires par les sections administratives
 - 3.1.2 consommations de fournitures primaires par les sections auxiliaires
 - 3.1.3 consommations de fournitures primaires par les sections de production

- 3.1.4 consommations de fournitures primaires par les produits fabriqués
- 3.1.5 consommations de fournitures primaires par les sections de vente
- 3.1.6 consommations de fournitures primaires par les produits marché
- 3.1.7 consommations de fournitures primaires par les sections d'administration générale
- 4. Consommations-UO : 4.1 Consommations - UO - sections - auxiliaires
 - 4.1.1 consommations par les sections administratives
 - 4.1.2 consommations par les sections auxiliaires
 - 4.1.3 consommations par les sections de production
- 4.2 Consommations - UO - sections - production par les produits fabriqués
- 4.3 Consommations - UO - sections - vente: par les produits marché
- 5. Prestations : 5.1 prestations - sections - auxiliaires :
 - 5.1.1 prestations aux sections administratives
 - 5.1.2 prestations aux sections auxiliaires
 - 5.1.3 prestations aux sections de production
- 5.2 prestations - section - production : aux produits fabriqués
- 5.3 prestations - section - vente : aux produits marché
- 6. Destination des produits : 6.1 destination - produits - fabriqués :
 - 6.1.1 produits destinés au stock correspondant
 - 6.1.2 produits (du dernier stade de fabrication) destinés au stock d'échange.

7. Consommations- produits : 7.1 consommations - produits :
- 7.1.1 consommations de produits en provenance d'autres départements e même usine
- 7.1.2 consommations de produits en provenance de l'extérieur
- 7.1.3 consommations de produits en provenance des stades amonts.
8. Achats client : 8.1 Achats - client : de produits en provenance des stocks d'échange.
9. Réception de produits :
- 9.1 Réception - département :
- 9.1.1 réception d'un département de produits cédés par d'autres départements
- 9.1.2 réception d'un département de produits en provenance des stocks d'échange.
10. Cession de produits :
- 10.1 Cession - département :
- 10.1.1 cession par un département de produits à d'autres produits pour leur fabrication
- 10.1.2 cession par un département de produits aux stocks d'échange.
11. Consommations de fournitures primaires :
- 11.1 Consommations - types - FP: consommations par types de fournitures primaires.

Remarques :

- le nom qui est souligné en pointillés sera celui qui devra figurer dans les requêtes.
- l'association 1.1.1 ne figure pas dans le tableau de la fig 3.3 car elle constitue un cas particulier. Etant donné que les données indicées sont interdites dans le système, il faut trouver un moyen pour éditer une partie ou toutes les composantes d'un tableau rassemblant les ventilations par types de fournitures primaires de différentes données.

La requête sera exprimée sous la forme d'une association dont l'origine sera l'entité dans laquelle figure le tableau et la cible sera l'entité TYPFP. Sa codification servira au choix des composantes du tableau à éditer.

Exemple : on désire obtenir les composantes du prix de revient marginal et complet des sections de production du bloc 23 du département 2 de l'usine 8 relatives aux types de fournitures primaires 10,20,30,50,80.

```
DEBUT   LOCALISATION
        E1 : S-PR : US : 8  DEPT = 2  BLOC = 23
        E2 : TYPFP : TYPFP = 10,20,30,50,80
        E3 : CONSOMMATIONS-TYPES-FP
EDITION
        ETAT : BLOC
        LISTE: E1-COS-PR, E1-LIB, E1-EL-PR-MARG, E1-EL-PR-COMP.
```

FIN

- problème des caractéristiques d'association : jusqu'à présent on avait permis une sélection plus fine des occurrences d'entités en jouant sur les valeurs des données.

Exemple : pour obtenir les dépenses de sous-traitance des sections auxiliaires, on aura la requête suivante :

```
DEBUT   LOCALISATION
        E1 : S-AUX :  US = 8;  TY = 3
CALCUL
        R1 = MULT(E1-NIVACT, E1-COUT).
EDITION
        ETAT = US
        LISTE= E1-COS-AUX, E1-LIB, E1-COUT, R1.
```

FIN

Pour l'entité relation, on n'a pas encore mentionné des possibilités de ce type. En ce qui concerne les standards de consommation, il faudrait pouvoir distinguer les frais fixes des frais variables. L'utilisateur pourra le faire par l'intermédiaire de la caractéristique d'association qui existera uniquement dans ce cas. Elles auront pour libellé : fixes, variables, totales (dans le cas où l'on désire les deux types de frais).

Exemple : on désire obtenir les consommations fixes en fournitures primaires des sections administratives de l'usine 8 et département 4.

```
DEBUT   LOCALISATION
        E1 : S-AD:  US=8  DEPT=4
        E2 : STKFP:  US=8  DEPT=4
        E3 : CONSOMMATIONS-FP : FIXES.
CALCUL
        R1 = MULT(E3-STD, E2-PRX-CONS-MOY).
EDITION
        ETAT = SECT
        LISTE= E2-COSTKFP, E2-LIB, R1, E2-UN, E2-PRX-
              CONS-MOY, R1.
```

FIN.

3.3.1.4 Conclusions concernant la partie localisation :

- Pour une question ponctuelle, on aura au maximum deux entités dans la requête qui sera de la forme suivante:

```
LOCALISATION
```

```
E1 : ENTITE 1 :....
```

```
E4 : ENTITE 2 :....
```

On remarque donc que les données servant de base de référence pour une comparaison avec celles des différentes occurrences de ENTITE 1 seront toujours mentionnées dans l'entité E4 sauf si cette dernière est identique à celle mentionnée en E1. Dans ce cas, on ne la spécifiera pas une seconde fois.

- Pour une question relative à une association, il y aura au maximum quatre entités dont trois seront toujours présentes. L'entité dont les données servent de référence pour une comparaison sera toujours spécifiée en E4 sauf si elle est identique à une de celles spécifiées en E1 ou E2.

La partie localisation sera de la forme :

```
LOCALISATION      E1 : ENTITE 1 :...
                  E2 : ENTITE 2 :
                  E3 : libellé de l'association
                  E4 : ENTITE 4 :
```

3.3.2 Partie " CALCUL " de la requête.

On va décrire exactement chaque opérateur permettant des manipulations de données. Cette partie sera séparée de la localisation par le symbole " CALCUL ".

3.3.2.1 Opérateur d'addition : " ADD ".

C'est un opérateur à trois opérandes maximum qui permet la totalisation de données de même nature (c'est-à-dire qui ont même unité). Le test sur la compatibilité de la nature des différents opérandes ne sera pas fait par le système et cela est donc laissé à l'entière responsabilité de l'utilisateur.

Les opérandes seront soit des données figurant dans la banque, soit des données figurant dans la banque, soit des données résultats de calculs précédents.

Ex : on désire connaître le niveau d'activité fixe, variable et total des sections auxiliaires de l'usine 8.

```
DEBUT      LOCALISATION
           El : S-AUX : US=8.
           CALCUL
           R1 = ADD(El-NIVACT-FIXE, El-NIVACT-VAR).
           EDITION
           ETAT = US
           LISTE= El-COSAUX, El-LIB, El-NIVACT-FIXE,
                 El-NIVACT-VAR, R1.
```

FIN.

3.3.2.2 Opérateur de soustraction : " SOUST ".

C'est un opérateur binaire qui soustraira le second opérande du premier. Il faudra donc toujours la compatibilité de la nature de ceux-ci et elle ne sera toujours pas vérifiée par le système. Les opérandes seront soit des données de la banque, soit des données résultats de calculs précédents.

Ex : on désire connaître le niveau de sous-emploi des sections auxiliaires de l'usine 8.

DEBUT LOCALISATION

E1 : S-AUX : US = 8.

CALCUL

R1 = ADD(E1-NIV-ACT-FIXE, E1-NIV-ACT-VAR)

R2 = SOUST(E1-CAPX, R1)

EDITION

ETAT = US

LISTE= E1-COS-AUX, E1-LIB, R1, R2.

FIN

3.3.2.3 opérateur de multiplication : " MULT "

C'est un opérateur binaire pour lequel la compatibilité de nature entre les données n'est plus nécessaire (Ex : on peut multiplier une quantité par un prix) mais la responsabilité de la signification réelle de l'opération est toujours laissée à l'utilisateur (Ex : à lui de savoir si la multiplication de deux quantités est une opération sensée). Les opérandes seront soit des données de la banque, soit des données résultats de calculs précédents.

Ex : on désire la valeur des services prestés par les sections auxiliaires de l'usine 8.

DEBUT LOCALISATION

E1 : S-AUX : US = 8.

CALCUL

R1 = ADD(E1-NIV-ACT-FIXE, E1-NIV-ACT-VAR)

R2 = MULT(R1, E1-PR-COMP).

EDITION

ETAT : US

LISTE: E1-COS-AUX, E1-LIB, R2.

FIN

3.3.2.4 opérateur de division : " DIV "

C'est un opérateur binaire pour lequel l'ordre de spécification des opérands est important car ce sera toujours le premier opérande qui sera le dividende et le second le diviseur. Le fait de juger, que l'opération est sensée ou non, est laissé à l'entière responsabilité de l'utilisateur.

Ex : on désire connaître le nombre d'heures prestées par les S-AUX 222,223 du département 2 aux section 224,225, 226 du même département de l'usine 8.

DEBUT LOCALISATION

E1 : S-AUX : US = 8 DEPT = 2 SECT = 222, 223

E2 : S-AUX : US = 8 DEPT = 2 SECT = 224,225,226.

E3 : PRESTATION-SECTION-AUXILIAIRE : TOTALES.

CALCUL

R1 = MULT(E2-NIV-ACT, E3-STD)

R2 = DIV(R1, E3-STD).

EDITION

ETAT = SECT

LISTE= E2-COS-AUX, E2-LIB, R1, R2.

FIN

3.3.2.5 opérateur de calcul de pourcentage " PRCT "

C'est un opérateur binaire où le premier opérande sera celui dont on désire connaître l'importance qu'il prend par rapport à une donnée de référence qui constitue le second opérande.

Ex : on désire connaître la part prise par les achats du client 222 du marché 4141.

DEBUT LOCALISATION

E1 : PR-M-CL : MARCHE = 4141, M-CL = 222

E4 : M-CL : MARCHE = 4141, M-CL = 222.

CALCUL

R1 = MULT(E1-Q-VENDUE, E1-PV)

R2 = PRCT(R1, E4-CAB).

EDITION

ETAT = M-CL

LISTE= E1-COPR-M-CL, E1-LIB, R1, R2.

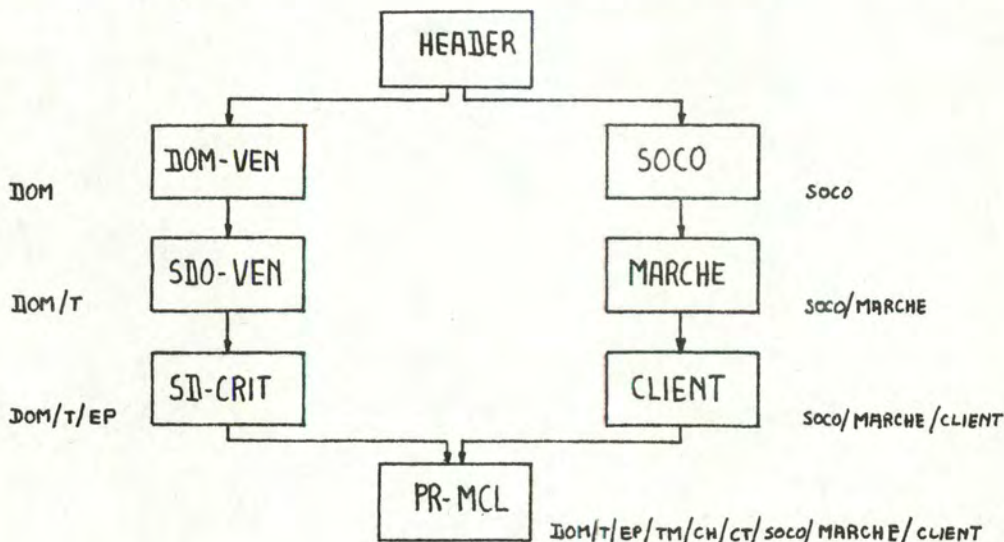
FIN

3.3.2.6 opérateur " CUMUL ".

Jusqu'à présent on a défini des opérateurs permettant d'effectuer, dans le cas d'une question ponctuelle, des calculs relatifs à chaque occurrence de l'entité spécifiée en E1 et, dans le cas d'une question relative à une association, des calculs relatifs à chaque paire d'occurrences des entités spécifiées en E1 et E2; le résultat est utilisé dans la constitution de chaque ligne détail correspondant à l'occurrence ou à la paire.

On va définir maintenant un opérateur dont le résultat sera également nécessaire à la constitution de la ligne détail relative à une ou à la paire d'occurrences, mais dont le calcul nécessitera la prise en compte de plusieurs occurrences d'une même entité. Cette situation est due à une particularité de la structure de la banque de données où certaines entités ont une existence propre (c'est-à-dire une codification propre) tout en faisant partie de deux lignes hiérarchiques distinctes :

Exemple :



On remarque que la codification de PR-M-CL contient des renseignements de type produit et de type marché alors que les autres n'en contiennent que d'un seul type. Si l'on désire définir une donnée de référence pour le calcul d'un pourcentage, qui tienne compte des deux critères, il faudra donc en effectuer le calcul.

Cet opérateur ne pourra donc être utilisé que comme second opérande de l'opérateur pourcentage. Il sera binaire et ses opérandes seront toujours des données d'occurrences de l'entité spécifiée en E4. Le premier sera un générique de codification tandis que le second sera une donnée figurant dans la banque ou sera une donnée calculée.

Exemple : on désire obtenir les achats du client 222 sur le marché 4141 dans le domaine 20 et comparer le chiffre d'affaires de chaque produit au chiffre d'affaires total en ce domaine.

DEBUT LOCALISATION

E1 : PR-MCL DOM = 20 MARCHE = 4141 CLIENT = 222

E4 : PR-MCL DOM = 20 MARCHE = 4141 CLIENT = 222.

CALCUL

R1 = MULT(E1-Q-VENDUE, E1-PV)

R2 = MULT(E4-Q-VENDUE, E4-PV)

R3 = CUMUL(E4-DOM, R2)

R4 = PRCT(R1, R3).

EDITION

ETAT = DOM

LISTE= E1-COPR-M-CL, E1-LIB, R1, R3, R4.

FIN

Remarque : on a défini deux fois le même calcul en R1 et R2 pour séparer complètement le calcul relatif à l'opérateur CUMUL des autres calculs.

3.3.2.7 Remarque :

On autorise l'utilisation d'au maximum vingt données calculées.

Le " après " fait référence à la séquence de base fournie par la partie codification d'une entité.

Ex : on désire obtenir les quantités approvisionnées en fournitures primaires de type 80.

DEBUT LOCALISATION

El : FPTOT : TYPFP = 80.

EDITION

ETAT : TYPFP

SOUS-TOTAL : NAT

LISTE : El-COFPTOT, El-LIB, El-Q-APPROV.

on aura les lignes détails dans l'ordre suivant :

TYPFP _i	NAT _{ij}	FOUR _{ij1}
		FOUR _{ij2}
		FOUR _{ijn}
		<hr/>
		TOTAL NAT _{ij}
	NAT _{ik}	FOUR _{ik1}
		FOUR _{ikm}
		<hr/>
		TOTAL NAT _{ik}

Si la codification se décompose en deux groupes, tout générique de niveau inférieur au premier figurant dans le paramètre ETAT entraînera un sous-total à l'intérieur de l'état.

Ex : pour la séquence de base : DOM T EP TM CH CT SOCO MARCHE
CLIENT

si le paramètre état est : EP, MARCHE et sous-total:

DOM, SOCO, la séquence de rupture sera :

DOM, EP, MARCHE, SOCO

Si le paramètre état ne contient qu'un générique du second groupe, tous ceux du premier figurant dans sous-total entraîneront des totaux à l'intérieur de l'état.

Ex : même structure de codification que précédemment, même sous-total mais un paramètre ETAT = MARCHE. La séquence de rupture sera : SOCO, MARCHE, DOM.

Lorsque l'utilisateur désirera obtenir des totaux sur base de plusieurs états, il faudra que le générique indiquant ce total soit hiérarchiquement supérieur au premier mentionné dans ETAT.

Ex : cas de DOM et de SOCO dans les deux exemples précédants.

Remarque :

- si les génériques de codification mentionnés par l'utilisateur dans le paramètre sous-total ne respectent pas l'ordre indiqué par la séquence de base, la requête sera refusée.
- ordre d'utilisation des génériques de rupture :

Ex : si on a le paramètre sous-total DOM MARCHE, les totaux seront produits de la manière suivante :

```

                                détail 1
                                détail 2
total MARCHE 1
                                détail 3
                                détail 4
total MARCHE 2
total DOMAINE 1.
```

3.3.3.3 Données figurant dans les lignes détails.

La liste des données constituant la ligne détail à imprimer sera spécifiée par le paramètre LISTE.

Elles seront soit :

- les données figurant dans la banque : elles seront mentionnées par le nom figurant dans la banque et préfixées par le symbole Ei (i allant de 1 à 4) permettant de relier le nom de la donnée à l'entité à laquelle elle appartient ;
- les données calculées : elles seront mentionnées par le nom qui leur a été attribué dans la partie calcul.

Exemple : on désire connaître par produit le chiffre d'affaires du client 122 sur le marché 4141.

DEBUT LOCALISATION

E1 : PR-MCL : SOCO=4 MARCHE=141 CLIENT=122.

CALCUL

R1 = MULT(E1-Q-VENDUE, E1-PV).

EDITION

ETAT = CLIENT

LISTE= E1-COPR-M-CL, E1-LIB, R1.

FIN

3.3.4 Remarques :

Le triplet localisation, calcul, édition pourra être reproduit entre un "début" et un "fin" à la condition que l'entité spécifiée en E1 soit la même dans chaque partie et que les critères de mise en page soient identiques car la génération de la structure de l'état imprimé sera générée une seule fois lors de l'analyse du premier triplet.

Exemples : on désire connaître les consommations en FP et UO de sections auxiliaires des sections auxiliaires de l'usine 8.

DEBUT LOCALISATION

E1 : S-AUX : US = 8

E2 : STKFP : US = 8

E3 : CONSOMMATIONS-FP : TOTALES.

CALCUL

R1 = MULT(E1-NIV-ACT, E3-STD)

R2 = MULT(R1, E2-PRX-CONS-MOY).

EDITION

ETAT = SECT

LISTE= E2-COSTKFP, E2-LIB, R1, R2.

LOCALISATION

E1 : S-AUX : US = 8

E2 : S-AUX : US = 8

E3 : CONSOMMATIONS-UO-SAUX: TOTALES.

CALCUL

R1 = MULT(E1-NIV-ACT, E3-STD)

R2 = MULT(R1, E2-PR-COMP).

EDITION

ETAT = SECT

LISTE= E2-COS-AUX, E2-LIB, R1, R2.

FIN

CHAPITRE 4.

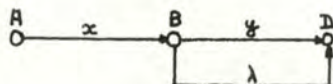
Algorithme d'analyse

du langage.

4.1 Grammaire formelle du langage

<requête> ::= DEBUT<requêtel>FIN
 <requêtel> ::= <requête2> | <requête2><requêtel>
 <requête2> ::= <localisation><actions>
 <actions> ::= <édition> | <calcul><édition>
 <localisation> ::= LOCALISATION<liste d'entités>
 <liste d'entités> ::= E1:<sélection d'entité><liste d'entités1> |
 E1:<sélection d'entité>E2:<sélection d'entité>
 E3:<sélection d'association><liste d'entités1>
 <liste d'entités1> ::= . | E4:<sélection simple>
 <sélection d'entité> ::= <nom d'entité>:<critères de sélection>
 <critères de sélection> ::= <sélection sur codification> |
 <sélection sur codification>;<sélection sur nom de donnée>
 <sélection sur codification> ::= ALL | <éléments de codification>
 <éléments de codification> ::= <codification><éléments de codification> |
 <codification>
 <codification> ::= <générique de codification>=<liste de codes>
 <liste de codes> ::= <code>,<liste de codes> | <code>
 <code> ::= <zone «v»>
 <nom d'entité> ::= nom figurant dans le modèle I.D.S
 <générique de codification> ::= générique figurant dans la partie
 codification de chaque article de la banque de données
 <sélection sur nom de donnée> ::= <donnée de sélection> | <donnée de
 sélection><opérateur de combinaison><sélection sur
 nom de donnée>
 <donnée de sélection> ::= <nom de donnée><opérateur de comparaison>
 <valeur>
 <nom de donnée> ::= nom figurant dans les parties données et résultats
 de chaque article de la banque de données
 <opérateur de comparaison> ::= <|> | <=> | <|> = | <|> =
 <valeur> ::= 'zone «v»'
 <opérateur de combinaison> ::= ET | OU
 <sélection simple> ::= <nom d'entité>:<sélection sur codification>
 <sélection d'association> ::= <libellé d'association>:<libellé de
 caractéristique> | <libellé d'association>
 <libellé d'association> ::= voir dictionnaire des associations

Exemple :



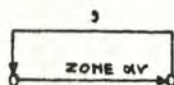
l'automate fini ainsi obtenu est non-déterministe car à l'état A, correspondent les états B et D.

Le problème est de rendre l'automate obtenu déterministe. Pour cela, on va d'abord exprimer graphiquement chacune des règles de déduction (ou production) de la grammaire, puis en déduire l'automate non-déterministe correspondant. Sur base de celui-ci, on construira le déterministe équivalent.

4.2.1 Représentation graphique de quelques productions de la grammaire (voir fig.4.1)

Pour constituer l'automate fini non-déterministe, il suffit de partir d'une production terminale, de la placer dans celle où elle apparaît en partie droite et ainsi de suite.

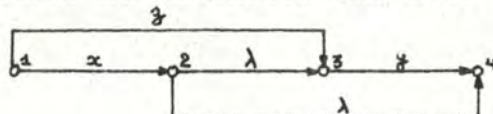
Exemple : on replace code dans liste de codes et on obtient :



4.2.2 Automate fini non-déterministe : (voir fig.4.2)

4.2.3 Passage à l'automate fini déterministe équivalent :

Si on considère la situation suivante :



le terminal x fait passer de l'état 1 aux états 2,3,4 ce qui sera schématisé dans l'automate déterministe par la transition suivante :



On va construire un tableau (fig.4.3) dans lequel chaque état (excepté l'état initial) est représenté par un quadruple constitué des éléments suivants :

- état atteint par la production appliquée
- élément de l'alphabet d'entrée lu pour atteindre cet état
- numéro du quadruple de l'état successeur. On peut trouver également la valeur V (= vrai) signifiant que la chaîne de caractères lue est correcte.

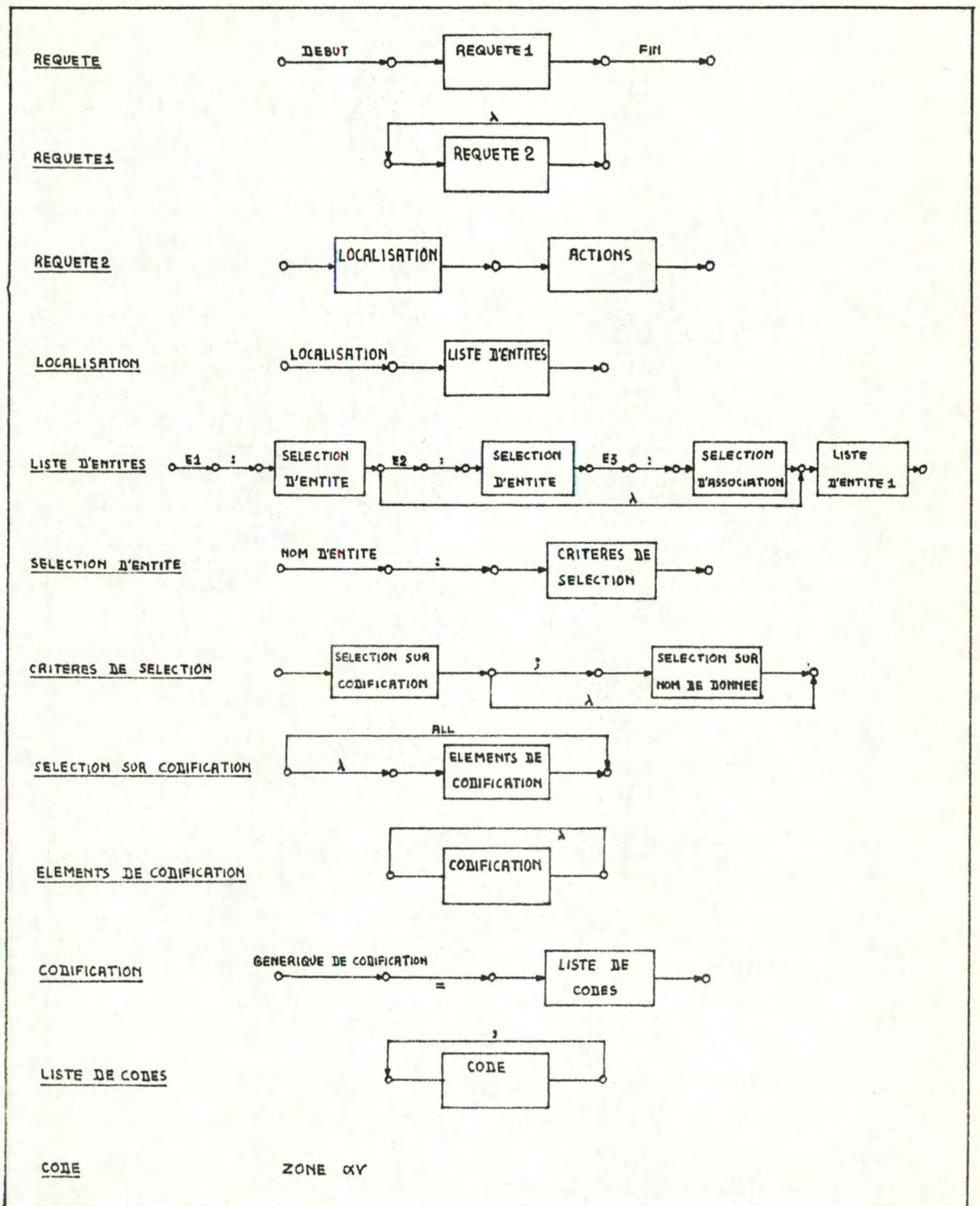


Fig. 4.1.

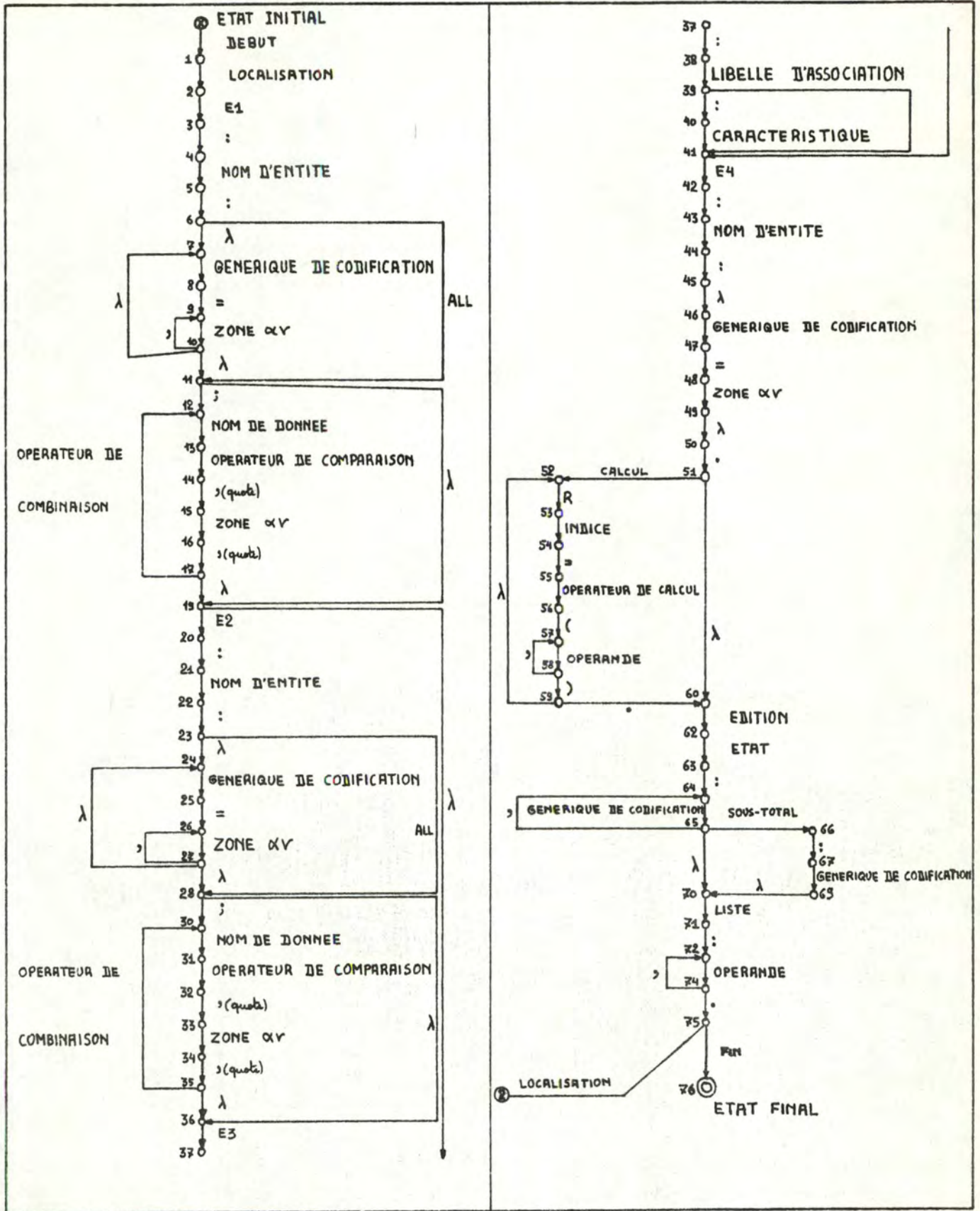


Fig. 4.2.

NUM°	ETAT	TERMINAL LU	SUCC	ALT	NUM°	ETAT	TERMINAL LU	SUCC	ALT
1	1	DEBUT	2	F	35	22	nom-d'entité	36	F
2	2	LOCALISATION	3	F	36	23,24	:	37	F
3	3	E 1	4	F	37	25	générique	39	38
4	4	:	5	F	38	28,36	ALL	44	F
5	5	nom-d'entité	6	F	39	26	=	40	F
6	6,7	:	7	F	40	24,27,28,36	zone	42	F
7	8	générique	9	8	42	26	'	40	43
8	11,19,41,50	ALL	13	F	43	25	générique	39	44
9	9	=	10	F	44	30	;	48	45
10	10,11,7,19,41,50	zone	11	F	45	37	E 3	61	F
11	9	'	10	12	48	31	nom de donnée	49	F
12	8	générique	9	13	49	32	=	55	50
13	12	;	19	14	50	32	<	55	51
14	20	E 2	34	15	51	32	>	55	52
15	42	E 4	67	16	52	32	<=	55	53
16	51,60	.	77	F	53	32	>=	55	54
19	13	nom de donnée	20	F	54	32	7=	55	F
20	14	=	26	21	55	33	'	56	F
21	14	<	26	22	56	34	zone	57	F
22	14	>	26	23	57	35,36	'	58	F
23	14	<=	26	24	58	30	ET	48	59
24	14	>=	26	25	59	30	OU	48	60
25	14	7=	26	F	60	37	E 3	61	F
26	15	'	27	F	61	38	:	62	F
27	16	zone	28	F	62	39,41,50	association	63	F
28	19,17,41,50	'	29	F	63	40	:	66	64
29	12	ET	19	30	64	42	E 4	67	65
30	12	OU	19	31	65	51,60	.	77	F
31	20	E 2	34	32	66	41,50	caractéristique	64	F
32	42	E 4	67	33	67	43	:	68	F
33	51,60	.	77	F	68	44	nom-d'entité	69	F
34	21	:	35	F	69	45,46	:	70	F

Fig. 4.3.

NUM°	ETAT	TERMINAL LU	SUCC	ALT	NUM°	ETAT	TERMINAL LU	SUCC	ALT
70	47	générique	72	71	102	58	indice	103	F
71	50	ALL	76	F	103	57	'	89	104
72	48	=	73	F	104	59,52)	105	F
73	46,49,50	zone	74	F	105	53	R	80	106
74	47	générique	72	75	106	60	.	78	F
75	48	'	73	76	107	63	ETAT	108	F
76	51,60	.	77	F	108	64	:	109	F
77	52	CALCUL	79	78	109	65,70	générique	110	F
78	62	EDITION	106	F	110	64	'	109	111
79	53	R	80	F	111	66	SOUS-TOTAL	113	112
80	54	indice	81	F	112	71	LISTE	117	F
81	55	=	82	F	113	67	:	114	F
82	56	ADD	88	83	114	69,70	générique	115	F
83	56	SOUST	88	84	115	67	'	114	116
84	56	MULT	88	85	116	71	LISTE	117	F
85	56	DIV	88	86	117	72	:	118	F
86	56	PRCT	88	87	118	74	E 1	119	121
87	56	CUMUL	88	F	119	74	-	120	F
88	57	(89	F	120	74	nom de donnée	132	F
89	58	E 1	90	92	121	74	E 2	122	124
90	58	-	91	F	122	74	-	123	F
91	58	nom de donnée	103	F	123	74	nom de donnée	132	F
92	58	E 2	93	95	124	74	E 3	125	127
93	58	-	94	F	125	74	-	126	F
94	58	nom de donnée	103	F	126	74	nom de donnée	132	F
95	58	E 3	96	98	127	74	E 4	128	130
96	58	-	97	F	128	74	-	129	F
97	58	nom de donnée	103	F	129	74	nom de donnée	132	F
98	58	E 4	99	101	130	74	R	131	F
99	58	-	100	F	131	74	indice	132	F
100	58	nom de donnée	103	F	132	72	'	118	133
101	58	R	102	F	133	75	.	134	F
					134	2	LOCALISATION	3	135
					135	76	FIN		F

C'est à partir de ce dernier que s'effectue l'analyse requête par requête. Lorsque l'une d'entre elles est refusée, on progresse dans ce fichier jusqu'à la rencontre du symbole de base FIN relatif à la requête erronée. La séquence normale d'analyse reprend au début pour la nouvelle requête.

Cette décomposition est réalisée au moyen d'un module qui, sur base des séparateurs mis en évidence, va isoler un à un les symboles de base figurant dans la requête.

4.2.4.1.2. Analyse d'un élément de base

Elle est réalisée grâce au tableau représentant l'automate fini déterministe que l'on va parcourir. Chaque élément de base isolé dans la requête va être comparé avec ceux figurant dans les quadruples représentant les états auxquels il est sensé conduire. Si l'on n'obtient jamais l'égalité entre les éléments de base comparés, la requête sera fausse.

4.2.4.1.2.1 Algorithme de parcours du tableau

On rappelle qu'un quadruple du tableau contient les informations suivantes :

- état auquel on se trouve;
- terminal lu pour atteindre cet état;
- numéro du quadruple de l'état successeur;
- numéro du quadruple de l'état alternant.

On suppose se trouver dans l'état x. Pour l'atteindre, on a reconnu l'élément de base A. On recherche le suivant. L'état, que l'on doit atteindre, étant donné par le paramètre SUCC, on va comparer l'élément de base isolé avec celui se trouvant dans la partie "terminal lu" du quadruple désigné par SUCC. S'ils sont identiques, on a atteint l'état y et s'ils ne le sont pas, on regarde la partie "ALT". Si elle contient "F", cela signifie que le seul chemin possible permettant de quitter x était celui arrivant y et donc que la requête est fausse. Si elle contient un numéro de quadruple, on va comparer l'élément de base à ce qui se trouve dans la partie "terminal lu". On continue de la sorte jusqu'à ce que, soit la comparaison donne un résultat positif, soit on trouve un "F" signifiant que la requête est fausse. Cette dernière sera correcte dès que l'on rencontre un V dans la partie "SUCC" (l'algorithme complet figure en annexe B).

n°	ETAT	TERMINAL LU	SUCC	ALT
0	x	A	1	F
1	y	B	3	F
2	z	C	4	F
3				
4				

4.2.4.1.2.2. Vérification de l'élément de base

Le tableau représentant l'automate déterministe contient, pour chacun des éléments de base du langage, un indicateur permettant de vérifier, directement ou dans une table, si l'élément reconnu est correct ou non. Voici la liste des indicateurs :

- 0 : précède tous les symboles de base ainsi que les opérateurs et chaînes de caractères. On ne considère pas une table spéciale associée à ces éléments car ils figurent textuellement dans le tableau. La vérification peut donc se faire immédiatement.

Exemple : le tableau contient dans la colonne "terminal lu" 0, LISTE. Cela signifie que l'élément de base qui a été isolé doit être LISTE.

- 1,2,3,4 : précède un nom-de-donnée dont il faut vérifier l'existence en allant consulter les entrées du dictionnaire relatives à la requête examinée. Elles sont stockées dans un tableau (fig.4.4) qui a la structure suivante :

1	2	3	4
entrée du dictionnaire relative à l'entité spécifiée en			
E1	E2	E3	E4

fig.4.4 tableau des entités relatives à la requête.

Toute donnée mentionnée dans la requête et figurant dans ce tableau voit, dans les descripteurs, son nom remplacé par un couple (x,y) où :

x sera un numéro entre 1 et 4

y sera le numéro d'ordre de la donnée dans l'entrée du dictionnaire.

Exemple : DEBUT LOCALISATION
 E1 : S-AD : US = 8
 EDITION
 ETAT : US
 LISTE: E1-COS-AD, E1-LIB, E1-DUR.

FIN

Seule la première composante du tableau est garnie et elle contient l'entrée du dictionnaire relative à la section administrative.

- 5 : précède un nom-d'entité. Dès que l'on rencontre cet indicateur, il faut parcourir la table des entités (fig.4.5) qui permet de faire la correspondance entre le nom et le numéro de cette dernière. Si l'on regarde l'exemple précédent, on remarque que la section administrative a été mentionnée au moyen du libellé S-AD. Or le module que l'on utilisera par la suite pour effectuer l'accès (= module I.D.S.) n'utilise que les numéros associés aux entités qui figurent sur le modèle I.D.S. Ces derniers serviront également de clé d'accès au dictionnaire des données et c'est grâce à eux que l'on pourra garnir le tableau décrit dans la figure 4.4.

H	U	D	
E	S	O	
A	I	M	
D	N	-	
E	E	V	
R		E	
		N	
0	0	0	
0	0	0	
1	2	3	

fig.4.5 Tableau des entités.

- 6 : précède un libellé d'association. Pour vérifier si l'association est autorisée entre l'entité spécifiée en E1 et celle spécifiée en E2, il faut, sur base des numéros de ces entités, parcourir le dictionnaire des associations et vérifier si elle y figure.
- 7 : précède un libellé de caractéristique dont la validité est également vérifiée par consultation du dictionnaire des caractéristiques.

4.2.4.2 Analyse sémantique.

Cette phase va consister, pour chacun des éléments de base reconnus, en la réalisation d'un certain nombre d'actions ayant pour résultat la génération des informations nécessaires à la bonne réalisation de l'édition. Cette phase n'est pas séparée de la première c'est-à-dire que, pour chaque élément de base reconnu, on accomplira les actions correspondantes. Le résultat de ces actions est présenté au chapitre 7.

TROISIEME PARTIE

ANALYSE DE L'IMPLEMENTATION.

Nous présentons succinctement le système permettant de répondre aux requêtes introduites par un utilisateur. Il permet :

- l'analyse syntaxique de la requête;
- la réalisation des accès aux données et l'élaboration des fichiers à éditer;
- la génération de la description de l'état imprimé;

0. Définition préliminaire :

Dans l'exposé, nous utilisons "entité" pour type d'entité, "numéro d'entité" pour numéro du type d'entité (c'est le numéro associé à chaque entité dans le modèle I.D.S.) et "occurrence d'entité" pour occurrence d'entité.

1. Analyse syntaxique de la requête :

Elle a pour but de vérifier si la requête introduite est correcte. Pour cela, le système dispose de trois dictionnaires qui sont :

- le dictionnaire des données (une donnée étant tout nom de niveau inférieur ou égal au niveau 02 dans la description COBOL de la banque de données) : il permet au système de vérifier, sans accès à la banque, que toute donnée mentionnée par l'utilisateur dans la requête figure bien dans l'entité à laquelle il l'a rattachée;
- le dictionnaire des associations (association = mise en correspondance de deux entités) : il permet de vérifier si l'association citée est autorisée entre les deux entités figurant dans la requête;

- le dictionnaire des caractéristiques (les consommations des sections peuvent être décomposées en consommations fixes et variables; ces éléments constituent les caractéristiques): l'ensemble des données relatives à l'association peut dans certains cas, être décomposé en sous-ensembles. Ce dictionnaire a pour but de vérifier si le sous-ensemble mentionné existe.

Simultanément à la validation de la requête, on génère un ensemble de descripteurs qui seront utilisés dans les deux phases suivantes.

2. Réalisation des accès et élaboration des fichiers à éditer.

La sélection des occurrences d'entités se fait en deux phases. La première consiste en l'accès aux occurrences spécifiées dans les descripteurs d'accès sur base de la codification. Dans l'ensemble ainsi déterminé, on peut, si l'utilisateur le désire, effectuer une sélection plus fine en comparant la valeur d'une donnée appartenant à une occurrence d'entité à une valeur de référence mentionnée par l'utilisateur. Lorsque cela est effectué, on dispose en mémoire centrale d'une occurrence d'entité à traiter. On peut d'abord calculer un certain nombre de données sur base des opérandes et opérateurs mentionnés par l'utilisateur et qui sont stockés dans la table des données calculées. Ensuite, on passe à la construction de deux fichiers. Le premier est celui des lignes à éditer qui est créé sur base du descripteur contenant la description de la ligne constituant l'état budgétaire. Lorsque ce fichier est créé, il faut le trier pour répondre au désir des utilisateurs qui souhaitent voir les lignes imprimées selon l'ordre croissant des identifiants. Ce tri est nécessaire car l'ordre existant dans la banque de données est celui du stockage qui peut être différent de l'ordre désiré. Le second fichier est celui des libellés qui renfermera les en-têtes d'état. Il est créé sur base des descripteurs d'en-tête et sous-totaux, et est également trié selon le critère cité précédemment. Ces fichiers triés seront les entrées du module d'édition.

3. Génération de la description de l'état imprimé.

Etant donné que la structure du budget est définie par l'utilisateur, il est nécessaire de passer par une phase de génération, qui, sur base des renseignements figurant dans la partie EDITION d'une requête, fournira en sortie un texte source COBOL contenant la description de l'état imprimé ainsi que les traitements à accomplir lors de la production de certains groupes d'édition.

Ce fichier sera également une entrée du module d'édition puisqu'il contient une partie de sa zone "data". Ce module devra donc être compilé avant de pouvoir être exécuté et fournira en sortie les budgets demandés.

L'architecture du système est schématisé par la fig.III.

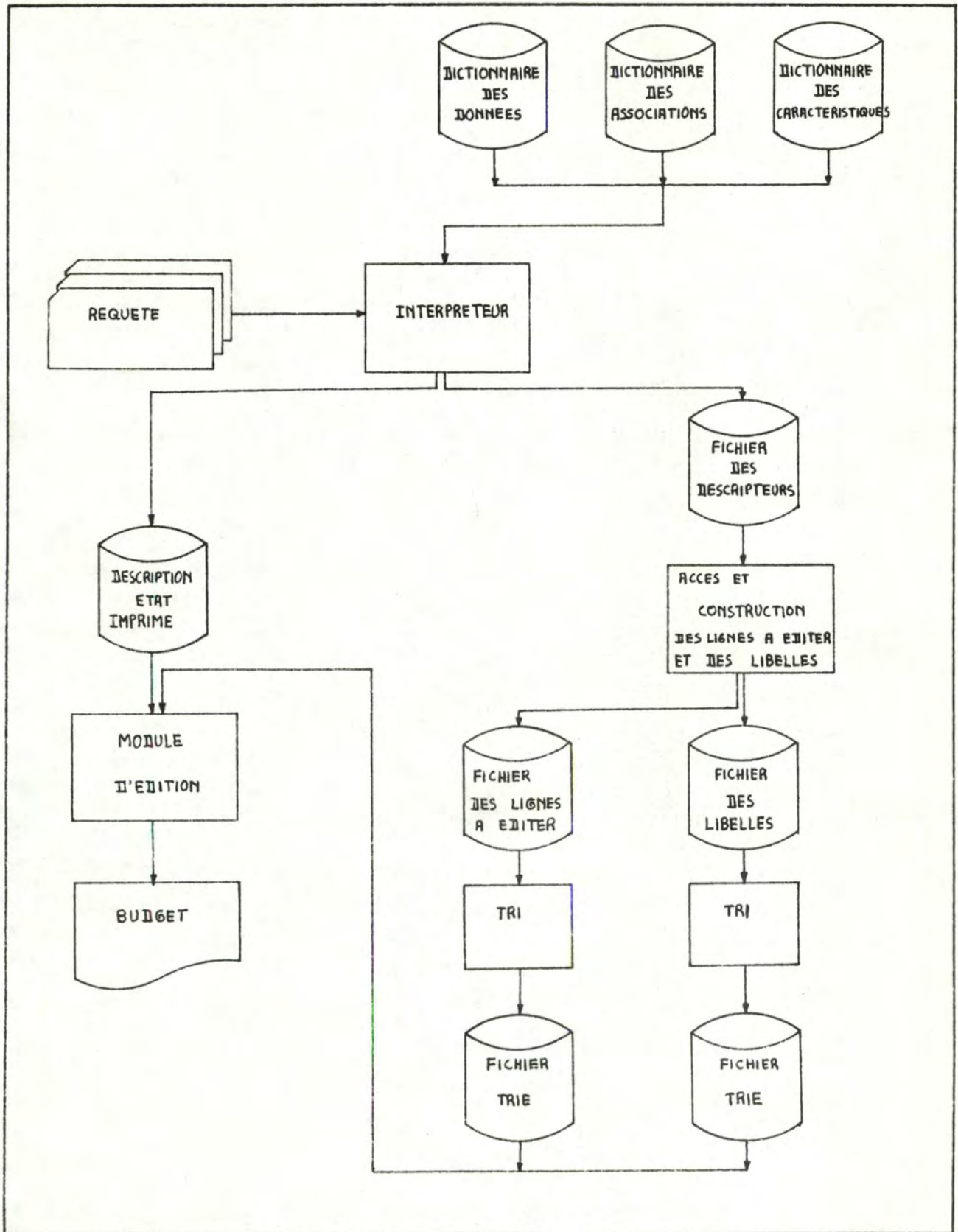


Fig. III

CHAPITRE 5.

Dictionnaires du système.

5.0 Introduction.

Nous présenterons dans ce chapitre les différents dictionnaires nécessaires à la vérification d'une requête et à la réalisation de la réponse. Ce sont :

- le dictionnaire des associations : qui a pour but de permettre au système de vérifier, lors de l'analyse de la requête, si l'association entre deux entités mentionnées par l'utilisateur est autorisée et si elle est matérialisée par une "entité relation". Il contient également les informations permettant le traitement de l'association;
- le dictionnaire des caractéristiques : certaines associations regroupent un ensemble d'informations plus finement décomposable. Ce dictionnaire mentionne les sous-ensembles que l'on peut distinguer. Il s'agira en fait des consommations que l'on peut décomposer en consommations fixes et consommations variables;
- le dictionnaire des données : dont le but est de permettre au système de vérifier, lors de la phase d'analyse, si la donnée mentionnée par l'utilisateur figure dans l'entité à laquelle il l'a rattachée. Il doit également contenir toutes les informations nécessaires pour réaliser l'accès et l'édition des données.

5.1 Définitions préliminaires.

- nom d'une donnée : c'est le nom attribué, dans la description COBOL de la banque de données, à une zone contenant une valeur.
Exemple : dans l'entité "section auxiliaire" (S-AUX) la zone contenant le niveau d'activité fixe est appelée NIV-ACT-FIXE et celle contenant le total des dépenses fixes en fournitures primaires directement incorporées par les sections auxiliaires est appelée TOT-DEP-FIXE-DIR-FP
- libellé associé à une donnée : les noms attribués aux données peuvent ne pas être toujours suffisamment explicites s'il fallait les faire apparaître sur un budget. On leur associe un libellé plus significatif qui sera utilisé lors de l'édition.
Exemple : dans l'entité section auxiliaire, on associe le libellé "unité" à la donnée UN et le libellé "libellé" à la donnée LIB.
- séquence de base : liste de génériques de codification figurant dans la partie codification d'une entité.
Exemple : la séquence de base d'une section de production est US/DEPT/BLOC/SECT.

5.2 Dictionnaire des associations.

5.2.1 Structure et constitution du dictionnaire.

Il a la forme d'un vecteur (fig.5.1) dont une composante constitue une entrée du dictionnaire et correspond à une association. Son contenu est défini sur base du tableau des associations permises établi au chapitre 3. Etant donné que la banque de données est supposée invariable, ce dictionnaire est invariable également et est chargé manuellement.

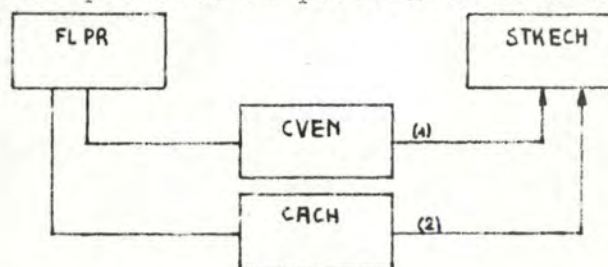
5.2.2 Contenu du dictionnaire.

(1)	(2)	(3)	(4)	(5)	(6)	(7)
018	016	0	CONSOMMATION-FP	040	000	1

- (1): numéro de l'entité origine de l'association :
- (2): numéro de l'entité cible de l'association :
- (3): un indicateur permettant de savoir s'il existe une ou plusieurs associations entre deux entités : lorsque la valeur de cet indicateur est 0, cela signifie qu'il n'existe plus d'autre association entre les deux entités spécifiées aux points 1 et 2 tandis que lorsqu'il possède la valeur 1, l'entrée du dictionnaire suivante contient également une association entre ces deux entités .

Exemple : entre un produit et un stock d'échange il existe les deux associations suivantes :

- 1): produits destinés aux stocks d'échange
 2): consommation de produits en provenance de l'extérieur



- (4): nom de l'association : nom de remplacement autorisé dans les requêtes.

- (5): numéro de l'entité qui matérialise l'association : pour une association de type n-n, la représentation I.D.S nécessite la définition d'une troisième entité. Le numéro de cette dernière est stocké à cet endroit. Lorsque l'association est de type l-n, on y place la valeur 000;
- (6): second numéro d'entité matérialisant l'association : dans le cas unique de l'association entre un produit (FL-PR) et un stock de produits (STKPR) il existe deux "entités relation" à savoir PRLIE et DPP. Le numéro de DPP figure à cet endroit. Dans les autres cas, il contient la valeur 000;
- (7): indicateur de caractéristique : pour les associations entre le stock de fournitures primaires et les entités qui en consomment, on peut arriver à une décomposition plus fine en faisant la distinction entre les consommations fixes et variables. Cet indicateur aura la valeur 0 dans le cas où il n'y a pas de caractéristique et la valeur 1 dans les autres cas;
- Explication de l'exemple : les consommations de fournitures primaires définissent la seule association entre l'entité section auxiliaire (018) et l'entité stock de fournitures primaires (016). Ceci explique le 0 dans la colonne (3). Cette association est matérialisée par la seule entité relation 040 ce qui explique le 000 dans la colonne (6). Elle possède des caractéristiques (1 dans la colonne 7).

5.3 Dictionnaire des caractéristiques.

5.3.1 Structure et composition du dictionnaire.

C'est un vecteur (fig.5.2) qui comprend trois composantes associées à chacune des caractéristiques fixes, variables, totales. Etant donné que les associations sont supposées invariantes, les caractéristiques le sont également. Le contenu de ce dictionnaire est donc fixe et est constitué manuellement.

5.3.2 Contenu du dictionnaire.

(1)	(2)	(3)	(4)
fixes	consommations fixes	3	1
variables	consommations variables	1 2 4	2
totales		1 2 3 4	

fig.5.2 Dictionnaire des caractéristiques.

- (1): libellé de la caractéristique : c'est celui qui figure dans la requête;
- (2): libellé associé : c'est celui qui sera utilisé lors de l'édition;
- (3): valeurs possibles du paramètre figurant dans la banque de données et correspondant à la caractéristique : étant donné qu'il est interdit de faire une sélection sur la valeur d'une donnée pour une "entité relation" et que la distinction entre consommations fixes et variables se fait précisément à ce niveau (la donnée RELAT), on va associer à chaque caractéristique les valeurs de RELAT qui lui sont associées;
- (4): code édition associé à la caractéristique : lors de l'édition du budget, on regroupe les consommations fixes d'un côté et les consommations variables de l'autre. Le but du code est de permettre d'effectuer ce regroupement au moment du tri du fichier des lignes à éditer.
- Explication de l'exemple : pour les consommations fixes, la valeur du paramètre RELAT, nécessaire pour la recherche dans la banque de données, est 3 et le code d'édition associé aux consommations fixes est 1. C'est la valeur qui prendra la variable CARACT dans la clé associée aux fichiers des lignes à éditer et des libellés. (voir §9.2.1.2)

5.4 Dictionnaire des données.

5.4.1 Structure du dictionnaire.

Il contient une entrée par entité définie dans la banque de données. Il est stocké sur un fichier à accès par clé, celle-ci étant constituée du numéro d'entité. Chaque entrée a la structure d'un vecteur dont chaque composante correspond à une donnée figurant dans l'entité et contient toutes les informations qui lui sont relatives. Chaque entrée dispose d'un même nombre de composantes qui est égal à la somme des données de niveau O2, O3, O4. On ajoute à ce nombre trois composantes dont le rôle sera explicité dans le paragraphe suivant. (schéma de la structure du dictionnaire : fig.5.3)

CLE D'ACCES = 001	CLE D'ACCES = 002	
	COUSINE (= codification usine)	
	NOUS (= numéro usine)	
	D-USINE (= données usine)	
	LIB (= libellé)	
ENTREE DU DICTIONNAIRE		

Fig. 5.3.

5.4.2 Contenu du dictionnaire. (fig. 5.4)

CLE D'ACCES		(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)
US	USINE	5	X	X	0	0	1	1	1; 002			0; 002	
PV	PRIX DE VENTE	13	C2	S9(7)V9(5)		0	12						

Fig. 5.4. Schéma d'une entrée du dictionnaire.

- (1) : nom de la donnée;
- (2) : libellé associé à la donnée : ce libellé constituera l'en-tête de la colonne du budget dans laquelle viendront s'inscrire les valeurs de cette donnée;
- (3) : longueur de ce libellé : actuellement, on impose que la longueur du libellé soit inférieure ou égale à dix-huit caractères. Afin qu'il soit toujours suffisamment explicite, il faudrait augmenter la longueur de ce libellé. Ce problème devrait être étudié ultérieurement;
- (4) : format d'édition de la donnée : toutes les données de la banque ne possédant pas le même format d'édition, il est nécessaire de spécifier lequel leur a été attribué;
Exemple : -pour un prix, le format est S9(7)V9(5)
-pour une quantité, le format est S9(10)V99
- (5) : format interne de représentation de la donnée : toutes les données de la banque ne possédant pas le même format de représentation, il est intéressant de spécifier lequel leur a été attribué afin d'assurer la compatibilité du transfert de la valeur d'une donnée d'une zone à une autre;

- (6) : un déplacement en mots : qui indique, pour chacune des données, leur position par rapport au début de la zone de "working-storage section" dans laquelle l'occurrence d'entité est introduite en mémoire centrale;
- (7) : un déplacement en caractères : certaines données alpha-numériques n'étant pas alignées sur un mot, il est nécessaire de savoir à partir de quelle position, à l'intérieur d'un mot, elles commencent;
- (8) : longueur en caractères de la donnée : ces trois dernières informations doivent permettre le calcul de l'adresse de la donnée dont on veut connaître la valeur et de réaliser son extraction;
- (9) : indicateur du groupe pour les génériques : la codification d'une entité pouvant se décomposer en deux groupes, il faut indiquer, pour chaque générique, le groupe auquel il appartient
Exemple : la codification d'un FL-PR (= produit) se décompose en:
 - caractéristiques fonctionnelles : US/DEPT/BLOC/SECT
 - caractéristiques de produit : DOM/T/EP/TM/CH/CT
- (10) : indicateur et numéro d'entité du niveau hiérarchique du générique : cette information est nécessaire pour déterminer, lors de l'analyse des éléments de codification introduits par l'utilisateur, le niveau hiérarchique pour lequel l'identifiant obtenu est complet ainsi que l'entité à laquelle on peut accéder directement;
- (11) : numéro de la composante du générique alternant : dans la définition de la syntaxe du langage nous avons spécifié que certains génériques pouvaient être remplacés, dans certains cas particuliers, par un ou plusieurs autres, plus explicites pour un utilisateur. C'est par le biais de ce numéro que l'on insère le (ou les) substitut(s) dans la séquence normale.
Exemple : la séquence de base d'un produit est US/DEPT/ELOC/SECT/DOM/T/EP/TM/CH/CT. Une caisse est un produit particulier pour lequel la teinte n'a aucun sens et est remplacé par type de caisse. La séquence de base devient : US/DEPT/BLOC/SECT/DOM/TY/EP;

- (12): numéro de la composante du générique successeur : lorsqu'on rencontre un cas particulier dans la codification, on s'écarte de la séquence de base et l'on continue la nouvelle séquence grâce à ce numéro ;
- (13): indicateur permettant de savoir si le libellé correspondant à un code figure dans une autre entité ou dans une table de codification : la partie de l'en-tête relative à un générique de codification se compose de trois parties :
 - (1): libellé complet du générique qui est le libellé associé à une donnée mentionné au point 2
 - (2): libellé correspondant au code
 - (3): code

Exemple: -BLOC:COMPOSITION /8120
 (1) (2) (3)
 -DOMAINE: FLOAT / FL

Le libellé COMPOSITION est obtenu en accédant à la donnée LIB du bloc 8120 tandis que FLOAT l'est en consultant la table associée au critère domaine (dessin d'l table de codification : fig.5.5). Cet indicateur a la structure suivante :

| 1 | 2 |

où 1 : prend la valeur 0 lorsque le libellé appartient à une entité et 1 dans l'autre cas;

2 : contient le numéro de l'entité ou le numéro de la table

VALEUR DU CODE	LIBELLE ASSOCIE
FL	FLOAT

fig.5.5 Table de codification (cas du critère domaine)

- (14) : adresse en "working-storage section" du début de la zone réservée à l'entité :

cette adresse correspond en fait à celle du premier niveau 02 de la description COBOL de l'entité et elle peut être obtenue en consultant les "field définition" de la table .IDS... Elle est mémorisée dans une composante particulière de l'entrée du dictionnaire qui contient, outre cette adresse, le libellé définissant le budget.

Celui-ci se superpose aux autres libellés définis aux points 1 et 2.

Exemple : si une requête commence par : E1 : S-PR :, le titre du budget est contenu dans cette composante et sera " budget des sections de production".

- Explication des exemples de la fig. 5.4 :

Exemple 1 : première ligne du tableau

- col 1 : code générique US, désigne le type usine
- col 2 : le libellé de ce type
- col 3 : longueur de ce libellé : 5 caractères alpha-numériques
- col 4 : format de la représentation interne : alpha-numérique représenté par X
- col 5 : format de sortie. Comme il s'agit d'un code numérique sur un caractère, le format utilisé est X
- col 6 : le générique usine représentant la première donnée d'une entité, son déplacement par rapport au début de l'entité est nul
- col 7 : pour la même raison, le déplacement par rapport au début d'un mot est nul
- col 8 : comme il s'agit d'un code numérique sur un caractère, la longueur occupée est un
- col 9 : le générique usine appartient au premier groupe de génériques
- col 10 : le générique usine fait partie du premier niveau hiérarchique de tout chemin d'accès et l'entité correspondant à ce niveau hiérarchique est l'entité usine de numéro 002
- col 13 : une valeur du générique usine peut être 8 qui correspond à l'**usine** de MOUSTIER. Ce nom sera trouvé par un accès à la donnée LIB (=libellé) de l'entité (ceci justifie le 0) usine (ceci justifie le numéro 002) d'identifiant 8

Exemple 2 : seconde ligne du tableau

- col 1 : PV désigne le type prix de vente
- col 2 : libellé de ce type
- col 3 : longueur de ce libellé : douze caractères alpha-numériques
- col 4 : format de la représentation interne : virgule flottante double précision représenté par C2

- col 5 : format de sortie : pour tous les prix, il sera S9(7)V9(5)
- col 6 : le déplacement de la donnée par rapport au début de l'entité figurera dans cette colonne
- col 7 : étant toujours alignée sur un mot, le déplacement de cette donnée par rapport au début du mot est nul
- col 8 : occupant deux mots, la longueur de cette donnée en caractères est douze

5.4.3 Constitution du dictionnaire.

On a fait l'hypothèse que la description des entités reste invariable dans le temps. Conséquemment, la seule information variable concernant une entité est l'adresse début de son implantation en "working-storage section". Il faut donc, après le chargement de la banque de données en mémoire centrale, consulter la table .IDS.. et mettre le dictionnaire à jour. Toutes les autres informations sont fixes et chargées manuellement.

CHAPITRE 6.

MODULE I.D.S.

Ce chapitre a pour but de présenter brièvement les objectifs du module écrit à l'Institut d'Informatique [1] et de décrire les ordres qui seront utilisés lors de la recherche dans la banque de données.

6.1 Objectifs du module.

6.1.1 Modularisation des programmes du simulateur sans nécessité de décrire la Banque de données I.D.S dans chaque module d'exploitation.

Il est en effet avantageux de :

- ne pas passer, dans chaque module d'exploitation, par une étape de pré-compilation I.D.S. de la banque de données car celle-ci est assez longue;
- n'avoir qu'un seul gestionnaire de la banque de données. La responsabilité de la gestion de celle-ci n'incombe qu'à une seule personne, ce qui permet d'en avoir à tout moment une version unique;
- pouvoir modifier certaines caractéristiques purement I.D.S de la banque de données sans devoir modifier les modules d'exploitation. A partir du moment où la description de la banque de données et tous les ordres I.D.S sont cantonnés dans un même module, toute optimisation d'accès d'ordre purement technique reste du ressort de ce module sans entraîner une maintenance des modules d'exploitation. Tout accès à la banque de données par ces derniers est considéré comme un accès logique, indépendant de la technique utilisée pour le réaliser.

6.1.2 Extension de la notion d'accès logique à celle de fonction logique standard.

Si un accès logique d'un module d'exploitation nécessite une séquence d'instructions COBOL-IDS et si l'accès envisagé peut être demandé par plusieurs modules, il est avantageux de rendre la séquence correspondante commune à l'ensemble des modules d'exploitation :

- gains de temps lors de la programmation et de la mise au point des modules d'exploitation;
- gains de temps lors des maintenances, une modification inhérente à la logique de la fonction ou à l'optimisation technique de celle-ci n'ayant aucune répercussion dans les modules d'exploitation;
- possibilité d'optimisation poussée de la fonction, cette optimisation, étant profitable à l'ensemble des modules utilisateurs.

Dans la suite, on ne parlera plus que de FONCTIONS du module I.D.S. Le module possède plusieurs points d'entrée, chacun d'eux étant une séquence d'instructions permettant de réaliser physiquement une fonction logique standard pour les modules d'exploitation.

6.2 Fonctions qui seront utilisées.

La partie sélection sur base de la codification ayant pour but de permettre à l'utilisateur de spécifier les identifiants des occurrences d'entités auxquelles le système devra accéder, les fonctions utilisées seront celles de recherche sur base d'un identifiant.

6.2.1 Recherche d'une occurrence d'entité sur base de l'identifiant et transfert en mémoire centrale (WSS)

- paramètres nécessaires :

- 1) numéro de l'entité désirée
- 2) identifiant de l'occurrence d'entité
- 3) code de retour : sa valeur est 1 si l'occurrence d'entité recherchée n'existe pas dans la banque de données, 0 sinon.

- point d'entrée : IRTRIM. L'appel au module I.D.S se fera donc par l'instruction CALL IRTRIM (liste de paramètres)

Exemple: la recherche du département 1 de l'usine 8 sera réalisée grâce à l'appel : CALL IRTRIM (006, 81, SW).

6.2.2 Recherche d'une occurrence d'entité sur base de l'identifiant de l'émetteur et du récepteur avec transfert en mémoire centrale (WSS)

- paramètres nécessaires :

- 1) numéro de l'entité désirée
- 2) identifiant de l'émetteur
- 3) identifiant du récepteur
- 4) type de relation : dans le cas d'un standard de consommation de fournitures primaires, il existe un paramètre RELAT permettant de faire la distinction entre consommations fixes et variables. C'est la valeur désirée de ce paramètre qui doit figurer à cet endroit.
- 5) code de retour : sa valeur est 1 si l'occurrence d'entité recherchée n'existe pas dans la banque de données, 0 sinon.

- point d'entrée : IRTISM. L'appel au module I.D.S se fera donc par l'instruction CALL IRTISM (liste de paramètres)

Exemple : la recherche du standard de consommation fixe de la fourniture primaire 10222 par la section auxiliaire 81 230 est obtenu par :
CALL IRTISM (40, 10222, 81 230, 3, SW).

On remarque notamment dans le cas 6.2.1 que la recherche n'est possible que si l'on dispose de l'identifiant de l'occurrence d'entité. Si ce n'est le cas, il est alors obligatoire d'effectuer la recherche en parcourant les chaînes. La fonction correspondante est :

6.2.3 Recherche de l'occurrence d'entité suivante dans une chaîne avec transfert en mémoire centrale (WSS)

- paramètres nécessaires :

- 1) numéro de l'entité maître de la chaîne
- 2) numéro de l'entité détail de la chaîne
- 3) numéro d'ordre de la chaîne si l'on a une chaîne double (fig.6.1), sinon 0.

cha : chaîne de gauche qui aura le numéro 1

chb : chaîne de droite qui aura le numéro 2.

- point d'entrée : IRTRNM. L'appel au module I.D.S se fera par l'instruction CALL IRTRNM (liste de paramètres)

Exemple : la recherche d'un département en parcourant la chaîne reliant département à usine se fait par : CALL IRTRNM (002, 006, 0).

6.3 Autres fonctions du module.

- ouverture de la banque de données (IOPEN1)
- fermeture de la banque de données (ICLOS1)
- suppression d'un article (IDLTE)
- modification des champs d'un article (IMODFY)
- transfert de l'article courant absolu en working-storage section (IMOVE)
- recherche d'un article courant du type (I RTRC) + transfert en working-storage section (IRTRCM)
- recherche de l'article précédent dans une chaîne (IRTRP) + transfert en working-storage section (IRTRPM)
- recherche du maître d'une chaîne (IRTRM) + transfert en working-storage section (IHEAD)
- recherche d'un article sur base de son référence-code (IRTRD) + transfert en working-storage section (IRTRDM)
- création d'un article (ISTORE)

CHAPITRE 7.

SORTIE DE L'INTERPRETEUR.

7.0 Introduction.

Nous allons présenter dans ce chapitre les éléments créés par l'interpréteur et sur base desquels sont réalisées la sélection des occurrences d'entités et la génération de la description de l'état imprimé. La sélection est réalisée sur base de deux types de descripteurs : le premier permet l'accès à une occurrence d'entité tandis que le second permet, parmi l'ensemble des occurrences retenues suite à l'accès, de n'en considérer qu'un sous-ensemble. Ensuite, on construit, si nécessaire, une table dans laquelle on indique, pour chaque donnée à calculer, la liste des opérandes ainsi que l'opérateur. Enfin, pour permettre, tant la génération de la description de l'état que la constitution des fichiers des lignes à éditer et des libellés, on génère trois descripteurs : celui de l'en-tête, celui des sous-totaux à imprimer et celui de la composition d'une ligne à éditer.

7.1 Génération des informations sur base de la partie localisation.

Le premier problème est de vérifier si les génériques de codification sont mentionnés dans l'ordre correspondant à celui indiqué par la séquence de base. On compare le premier générique cité avec le premier de cette séquence. S'ils sont égaux, on analyse le suivant dans la requête et s'ils ne le sont pas, on progresse dans la séquence de base. Après l'analyse d'un générique, on voit apparaître les codes lui correspondant. Etant donné que la codification est présentée par l'utilisateur sous une forme décomposée, il est nécessaire de stocker les codes de manière telle que la reconstitution d'un identifiant soit facilement réalisable. On considère pour cela un tableau (fig.7.1.) composé de dix vecteurs car une entité comporte au maximum dix génériques dans sa codification. Chaque vecteur comporte trente composantes car le nombre de valeurs distinctes que peut prendre un générique est trente (cas de SECT pour les sections auxiliaires). Après avoir vérifié que le générique mentionné figure au bon endroit dans la séquence de base, on peut en déduire un indice qui correspond à la place occupée par ce générique dans la séquence de base. Celui-ci servira à indiquer dans quelle ligne du tableau devront être stockés les différents codes mentionnés dans la requête.

Exemple : on considère la codification de l'entité S-PR.

E1 : S-PR : US = 8 DEPT = 1 BLOC = 20,21,22.

La séquence de base est US/DEPT/BLOC/SECT On obtiendra le tableau suivant :

8			
1			
20	21	22	

fig. 7.1 tableau de mémorisation des codes.

L'algorithme de recombinaison du tableau figure dans l'annexe B au point 12 où ce tableau a pour nom : TAB-IDENT.

L'identifiant qui est reconstruit à partir de ce tableau est placé dans un descripteur (fig.7.2) qui contient également deux numéros d'entités : le premier est celui de l'entité dont on désire connaître certaines informations, le second est celui de l'entité à laquelle on peut accéder directement sans parcourir les chaînes. S'il est différent du premier, cela signifie que l'identifiant est incomplet pour l'entité demandée. On accède alors à un niveau hiérarchique supérieur indiqué par le second numéro et on parcourt les chaînes à partir de celui-ci.

Exemple : Dans le précédent, on désire obtenir des renseignements sur les sections de production mais l'identifiant reconstitué ne mentionne que le bloc. On obtient alors :

| clé de stockage | 019 | 011 | 8120 |

fig.7.2 Descripteur concernant la codification.

La sélection peut également se faire sur base de noms de données d'où nécessité de générer un nouvel ensemble de descripteurs (fig.7.3) contenant toutes les informations nécessaires à cette sélection.

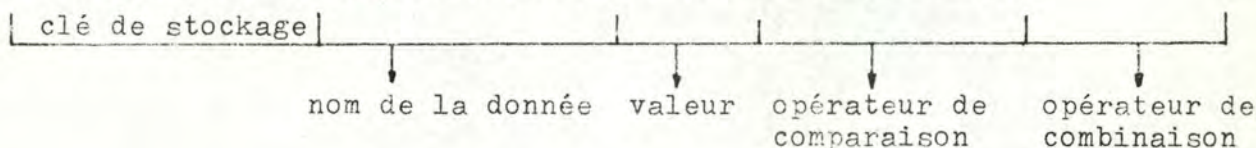


fig.7.3 Descripteur de sélection sur nom-de-donnée.

Le nom-de-donnée figure dans ce descripteur sous la forme d'un couple (x,y) où

- x indique l'entité à laquelle cette donnée appartient
- y indique le numéro d'ordre de la donnée dans la composante x.

En ce qui concerne la valeur, il se pose le problème de savoir quel est le format de la donnée. La banque de données contenant deux formats distincts pour ses données (le format alpha-numérique et le format virgule flottante double précision), il faut considérer deux zones pour la mémorisation. Le choix se fait sur base du format de la donnée. La zone occupée est marquée 1 et la zone inoccupée 0. L'opérateur de comparaison est recopié à l'intérieur du descripteur dans la case qui lui est réservée.

Pour les opérateurs de combinaison, il faut tenir compte du fait que le ET est prioritaire sur le OU. De plus, on désire générer ceux-ci selon un ordre correspondant à celui de la notation polonaise postfixée. Pour ce faire, on considère une pile dans laquelle on stocke tous les opérateurs de combinaison jusqu'à ce que l'on en trouve un qui soit de priorité inférieure ou égale à celle de l'opérateur figurant au sommet de la pile. Dès que c'est le cas, on vide cette dernière jusqu'à ce que le dernier opérateur lu aie une priorité supérieure à celui se trouvant au sommet de la pile. (l'algorithme détaillé est expliqué en annexe B au point 29).

Exemple : E1 : S-PR : US = 8 DEPT = 1 BLOC = 20;
NIV-ACT >= '200'.

On obtient le descripteur suivant sachant que NIV-ACT est une donnée de format virgule flottante

clé de stockage | (1,y) | 0; | 1; 200 | >= |

Pour une question relative à une association, il est nécessaire de disposer d'un descripteur (fig.7.4) indiquant quelle association a été considérée dans la requête. Il contient les numéros des entités unies par cette dernière et son libellé.

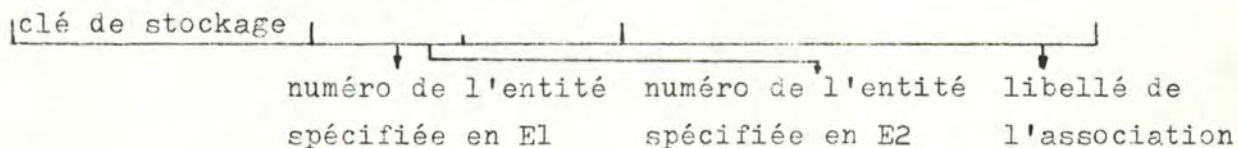


fig.7.4 Descripteur d'une association.

7.2 Génération d'informations sur base de la partie CALCUL.

On va garnir la table des données calculées (fig.7.5). Le seul problème que l'on rencontre est celui des données de référence.

Elles ne sont calculées ou accédées qu'une seule fois tandis que les autres sont calculées à chaque coup. Pour les distinguer, on fait précéder chaque opérande par un indicateur qui est positionné à 1 dans le cas d'une donnée de référence et à 0 dans les autres cas. Lorsque l'opérateur est précédé d'un indicateur valant 1, cela signifie que le résultat de l'opération est une donnée de référence. Toutes les données manipulées sont des données qui ont un format virgule flottante double précision. La table sera donc aussi de ce type.

Exemple : DEBUT LOCALISATION

E1 : S-AD : US=8 DEPT=1
 E2 : STKFP : US=8 DEPT=1
 E3 : CONSOMMATIONS-FP : FIXES
 E4 : USINE : US=8.

CALCUL

R1 = MULT(E1-NIVACT, E3-STD)
 R2 = MULT(R1, E2-PRX-CCNS-MOY)
 R3 = PRCT(R2, E4-TOTDEP-FPSAD).

EDITION

ETAT : SECT

LISTE: E2-COSTKFP, E2-LIB, R1, E2-UN, R2, R3.

FIN

donnée calculée	1er opérande	2iè opérande	3iè opérande	OPERATEUR	RESULTAT
R1	0 (1,x)	0 (3,y)	1	0 MULT	
R2	0 (0,1)	0 (2,z)	1	0 MULT	
R3	0 (0,2)	1 (4,u)	1	0 PRCT	

fig.7.5 table des données calculées.

Les opérandes sont données sous la forme (x,y) où

x = indique l'entité à laquelle la donnée appartient ou indique que la donnée est calculée (0)

y = numéro d'ordre de la donnée.

Remarque : pour les opérateurs à deux opérandes, le troisième est toujours marqué 1 car si les deux premiers le sont déjà, il faut que l'opérateur le soit également or le positionnement de l'indicateur relatif à l'opérateur se fera par un ET logique sur ceux des opérandes. (l'algorithme complet de garnissage de la table figure en annexe B au point 91).

7.3 Génération d'informations sur base de la partie EDITION.

Les informations à générer concernent :

- la définition de l'en-tête d'un budget : c'est sur base des génériques introduits par l'utilisateur que le système va déduire l'ensemble des données qui vont constituer l'en-tête. Ce sont tous les génériques dont les numéros d'ordre sont supérieurs ou égaux à celui du générique spécifié dans ETAT.

Exemple : DEBUT LOCALISATION
 E1 : S-PR : US = 8 DEPT = 1
 EDITION
 ETAT : BLOC
 LISTE: E1-COS-PR, E1-LIB, E1-NIV-ACT.

FIN

La séquence de base de la section de production étant US/DEPT/BLOC/SECT, les génériques qui constituent l'en-tête sont : US/DEPT/BLOC. Les génériques introduits par l'utilisateur sont stockés dans un descripteur d'en-tête (fig7.6) sous la forme d'un couple (1,y) où y est le numéro d'ordre dans la séquence de base.

(1,3)

fig.7.6 Descripteur d'en-tête

- la définition de la liste des sous-totaux à imprimer : elle est également définie par l'utilisateur au moyen de génériques relatifs à l'entité spécifiée en E1. Ceux-ci doivent être cités dans un ordre correspondant à celui de la séquence de base et sont stockés dans un descripteur des sous-totaux (fig.7.7) qui pourra contenir au maximum dix génériques sous la forme d'un couple (1,y)

Exemple : DEBUT LOCALISATION
 E1 : S-PR : US=8 DEPT=1
 E2 : STKFP: US=8 DEPT=1
 E3 : CONSOMMATIONS-FP: TOTALES.
 CALCUL
 R1 = MULT(E1-NIV-ACT, E3-STD).
 EDITION
 ETAT : SECT
 SOUS-TOTAL : US, DEPT, BLOC
 LISTE : E2-COSTKFP, E2-LIB, E2-UN, R1.

FIN

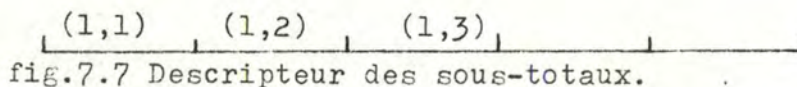


fig.7.7 Descripteur des sous-totaux.

- la définition des données constituant la ligne à éditer : ces données font partie d'une des entités relatives à la requête ou sont des données calculées. Après avoir vérifié qu'elles existent, on les stocke sous la forme d'un couple (x,y) dans le descripteur des données à éditer (fig.7.8)

Exemple : descripteur relatif à l'exemple précédent

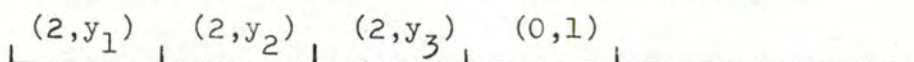
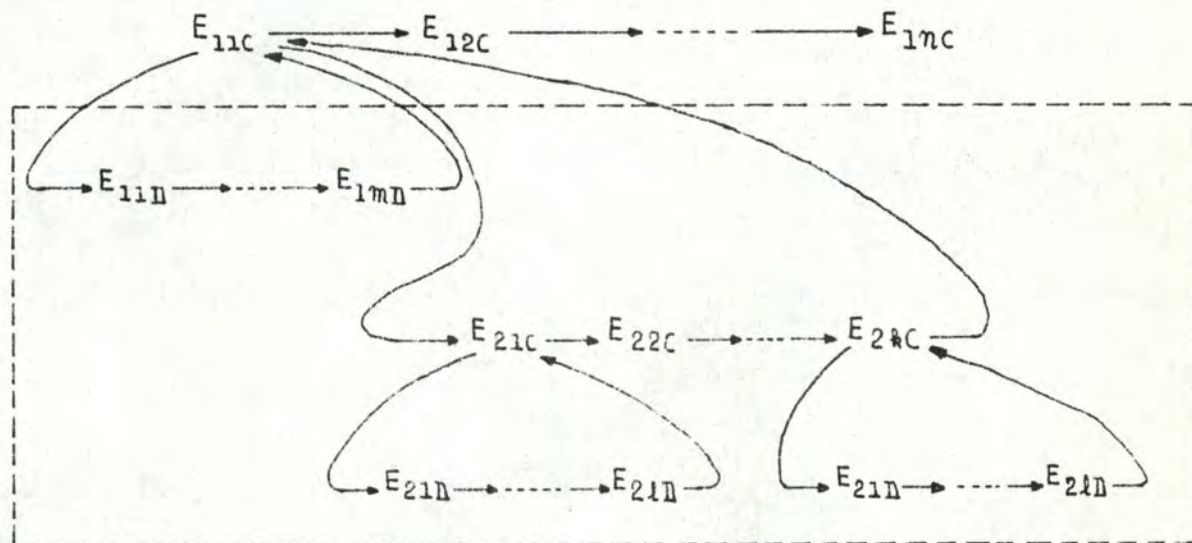


fig.7.8 Descripteur des données à éditer.

7.4 Stockage de ces descripteurs.

7.4.1 Stockage des descripteurs de sélection.



Si on appelle E_{1iC} et E_{2iC} les descripteurs de sélection sur base de la codification, et E_{ijD} et E_{2jD} les descripteurs de sélection sur base de nom de donnée, la structure entourée en pointillés doit être reproduite pour chaque E_{1iC} . Celle-ci ressemble fortement à une structure I.D.S. Pour l'utiliser de la sorte, il faudrait, soit considérer une seconde banque de données, soit l'introduire dans la description de la banque de données figurant dans le module I.D.S. On rejette ces deux solutions car d'une part, il est impossible d'utiliser simultanément, à l'exécution, deux banques de données, et d'autre part, on ne désire pas toucher au module I.D.S.

La solution retenue est de considérer un fichier à accès par clé dans lequel les éléments E_{11C} , E_{11D} , E_{21C} , E_{21D} sont stockés à un endroit fixe. Les boucles sont alors réalisées en effectuant le retour à cet endroit. Viendront s'ajouter dans ce fichier le descripteur de l'association ainsi que la liste des descripteurs relatifs à l'entité spécifiée en E4 si elle existe.

7.4.2 Stockage de la table des données calculées et des descripteurs d'édition.

La table de données calculées est stockée dans un fichier différent de celui décrit en 7.4.1 qui peut être éventuellement vide.

Les descripteurs d'édition seront également enregistrés à un endroit fixe dans le fichier décrit en 7.4.1 car il est nécessaire de pouvoir y accéder directement indépendamment de toute autre information lors de la génération de la description de l'état imprimé.

CHAPITRE 8.

SELECTION DES OCCURRENCES

ET CONSTITUTION

DES LIGNES A EDITER.

8.0 Introduction.

Nous allons décrire l'exploitation des éléments générés lors de l'analyse dans le but de créer les fichiers à éditer, en particulier :

- la sélection des occurrences d'entités désirées : elle peut être réalisée de deux manières distinctes :
 - sélection sur base des valeurs des identifiants : qui est obligatoire. Elle est fondée sur le descripteur contenant les informations suivantes : numéro de l'entité désirée, numéro de l'entité dont une occurrence est désignée par l'identifiant constituant la troisième information de ce descripteur :
Deux cas peuvent se présenter :
 - les deux numéros sont identiques : cela signifie que l'on peut accéder directement à l'occurrence de l'entité désirée;
 - les deux numéros sont distincts : l'accès est alors réalisé en deux étapes . On effectue tout d'abord un accès à l'occurrence de l'entité désignée par le second numéro. Ensuite, on parcourt le chemin constitué des chaînes reliant cette entité à celle désirée. Il a été nécessaire de définir tous les chemins d'accès possibles.
 - sélection par comparaison des valeurs des données avec une valeur fixée par l'utilisateur : cette sélection est facultative et permet d'écartier certaines occurrences d'entités retenues par la phase précédente. On se sert également de descripteurs contenant le nom de la donnée, l'opérateur de comparaison, la valeur introduite par l'utilisateur, l'opérateur de combinaison. Ces descripteurs sont obtenus et traités selon un ordre défini par la notation polonaise postfixée pour tenir compte des priorités habituelles définies sur les opérateurs de combinaison (ET,OU);
- le calcul des données : qui consiste en l'appel de modules spécialisés que l'on actualise au moyen des informations figurant dans la table des données calculées et qui fournissent en retour une valeur qui est stockée dans la zone résultat de la composante de la table correspondant à la donnée calculée;
- la constitution des fichiers à éditer : le premier contient les lignes à éditer dont la composition est donnée par le descripteur de la ligne à éditer. Le second contient les libellés dont la composition est déduite du descripteur d'en-tête et de celui des sous-totaux à imprimer.

8.1. Sélection des occurrences désirées.

8.1.1. Sélection sur base des valeurs des identifiants.

Lors de la définition de la syntaxe du langage, on a fait apparaître que certains génériques, partie de la codification de l'entité considérée, peuvent être omis dans une requête. Un identifiant reprenant toute la structure hiérarchique dans laquelle l'entité s'insère, on est amené à distinguer deux cas :

- l'identifiant est complet (aucun générique omis) ;
- l'identifiant est incomplet.

8.1.1.1. Cas de l'identifiant complet.

Lors de l'analyse, le système a généré un descripteur dans lequel les deux numéros d'entité sont identiques. L'occurrence d'entité, dont l'identifiant se trouve dans le descripteur, sera obtenue par un accès direct. Celui-ci sera réalisé par l'appel du module I.D.S au point d'entrée IPTRIM. Cette recherche nécessite les paramètres suivants :

- identifiant complet correspondant à l'occurrence d'entité considérée;
- numéro de l'entité considérée.

Exemple : on demande les activités des sections de production 122, 123 du bloc 20 du département 1 de l'usine 8.

```
DEBUT   LOCALISATION
        EL : S-PR : US=8  DEPT=1  BLOC=20  SECT=122, 123.
EDITION
        ETAT : BLOC
        LISTE: EL-COS-PR, EL-LIB, EL-NIV-ACT.
```

FIN

Les descripteurs générés sont : |019|019|8120122| et |019|019|8120123|
 La séquence de base de la section de production est US/DEPT/BLOC/SECT.
 On remarque donc que tous les génériques sont présents et que l'on se trouve bien dans le cas d'un identifiant complet. Les paramètres pour la recherche sont : 019, 8120122 et 019, 8120123.

8.1.1.2. Cas de l'identifiant incomplet.

Lors de l'analyse, le système a généré un descripteur dans lequel figurent deux numéros d'entité; le premier étant celui de l'entité considérée et le second celui de l'entité pour laquelle on peut déduire, des génériques et des codes correspondants mentionnés par l'utilisateur, un identifiant complet.

Cet artifice permet d'éviter un balayage trop important de la structure hiérarchique dans laquelle l'entité considérée s'insère. L'accès sera donc réalisé en deux phases :

- la première consiste en un accès direct à l'entité mentionnée par le second numéro du descripteur;
- la seconde consiste en un balayage de la structure hiérarchique de l'entité mentionnée par le second numéro jusqu'à celle désirée. Ceci est réalisé par l'appel au module I.D.S au point d'entrée IRTRNM et nécessite les paramètres suivants :
 - numéro de l'entité maître de la chaîne;
 - numéro de l'entité détail de la chaîne;
 - numéro de la chaîne dans le cas d'une chaîne double.

Exemple : on demande les activités des sections de production 122, 123 du bloc 20 de l'usine 8.

DEBUT LOCALISATION

 E1 : S-PR : US=8 BLOC=20 SECT=122,123.

EDITION

 ETAT : BLOC

 LISTE: E1-COS-PR, E1-LIB, E1-NIV-ACT.

FIN

Les descripteurs générés sont :

019,002,8 20122 et 019,002,8 20123

On remarque que par rapport à la séquence de base citée dans l'exemple précédent, on a omis le niveau hiérarchique département. La recherche s'effectuera comme suit :

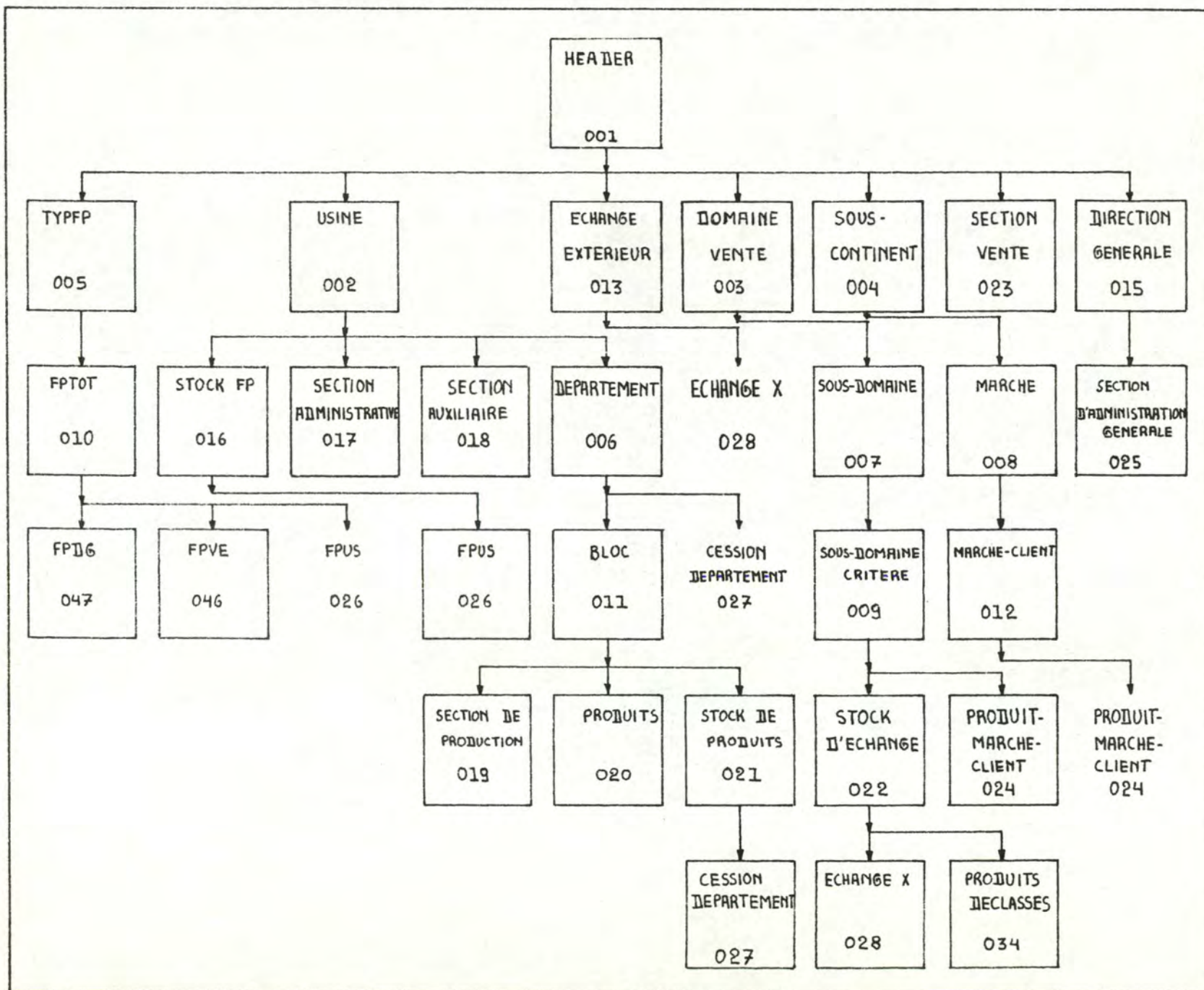
- accès direct à usine avec les paramètres : 002,8
- balayage des chaînes connectant
 - usine à département avec les paramètres 002,006,0;
 - département à bloc avec les paramètres 006, 011, 0;
 - bloc à section avec les paramètres 011, 019, 0.

Lorsqu'on aura accédé au bloc, il faudra comparer le code associé au générique BLOC à 20 et le code associé au générique SECT à 122 et 123 lorsqu'on aura accédé aux sections.

8.1.1.3. Algorithme de sélection.

Nous mettons d'abord en évidence (fig.8.1) les différentes structures hiérarchiques que l'on peut distinguer dans le modèle I.D.S et sur lesquelles est basé l'algorithme de sélection.

Fig. 8.1.



On peut remarquer deux particularités :

- la première est que certaines entités telles FPUS, PRMCL, figurent dans deux structures hiérarchiques distinctes. Le chemin à suivre pour y accéder est celui dans lequel l'entité est encadrée;
- la seconde est que toutes les " entités relations " sont exclues car elles seront toujours accessibles directement . (voir 8.1.1.6).

Nous présentons tout d'abord, une description d'un descripteur qui sera utilisée plus tard dans l'algorithme de sélection :

N1	N2	ID
----	----	----

N1 = premier numéro d'entité du descripteur

N2 = second numéro d'entité

RECORD-TYPE = variable qui contiendra le numéro de l'entité obtenue lors de l'accès à la banque de données

ID = identifiant figurant dans le descripteur.

L'élément central de cet algorithme de recherche est le second numéro d'entité à savoir N2. C'est sur base de celui-ci que l'on plonge à un endroit déterminé de la structure hiérarchique et que l'on accède directement à une occurrence d'entité qui est :

- soit l'occurrence désirée dans le cas d'un identifiant complet
- soit une occurrence à partir de laquelle il faut parcourir des chaînes pour obtenir celle désirée.

La distinction entre le cas complet et incomplet se fait sur base de la comparaison de N1 et N2. S'ils sont égaux, on est dans le cas complet et l'occurrence accédée est celle désirée.

Dans le cas d'un générique incomplet, on effectue les opérations suivantes :

1. recherche d'une occurrence d'entité au moyen de l'appel au module I.D.S au point d'entrée IPTNMM avec transmission d'une liste de paramètres;
2. si RECORD-TYPE = N2 alors on sort de ce module et la recherche d'entités relatives au descripteur reçu est terminée;
3. si RECORD-TYPE = numéro d'entité maître de la chaîne alors rechercher le suivant de ce maître;
4. si N1 ≠ numéro d'entité du niveau hiérarchique atteint alors ajuster la liste de paramètres relative à IPTNMM car il faut descendre dans la structure hiérarchique pour trouver l'entité désirée et retour au point 1;

001	A-001
002	A-002
003	A-003
004	A-004

TAB.1.

A-001

CALL IRTRIM (001, ID, SW).
 IF SW = 1 THEN la banque de données n'existe pas
 IF N1 = 001 THEN traitement 1

002	RECH-002
003	RECH-003
004	RECH-004
017	RECH-002

TAB.2.

A-002

CALL IRTRIM (002, ID, SW).
 IF SW = 1 THEN l'occurrence d'entité recherchée ne figure pas
 dans la banque de données
 IF N1 = 002 THEN traitement 1

016	RECH-016
017	RECH-017
018	RECH-018
006	RECH-006

TAB.2.

RECH-002

CALL IRTRNM (001, 002, 0).
 IF RECORD-TYPE = N2 THEN fin de recherche
 IF N1 ≠ 002 THEN

016	RECH-016
017	RECH-017
018	RECH-018
006	RECH-006

TAB.2.

tester si les portions d'identifiants sont identiques
 traitement 2

A-017

CALL IRTRIM (017, ID, SW).
 IF SW = 1 THEN l'occurrence d'entité recherchée ne figure pas
 dans la banque de données
 traitement 1

RECH-017

CALL IRTRNM (002, 017, 0).
 IF RECORD-TYPE = N2 THEN fin de recherche
 IF RECORD-TYPE = 002 THEN aller à RECH-002
 tester si les portions d'identifiants sont identiques
 traitement 2

Fig. 8.2.

5. sinon on a atteint l'entité désirée; on sort du module de recherche.

La structure générale de l'algorithme est donnée dans la fig.8.2 qui est réalisée dans le cas d'une section administrative.

8.1.1.4. Remarques concernant l'algorithme de la fig. 8.2.

- utilisation des tables de branchement.

(1) : rôle de TAB.1 : cette table est utilisée lorsque, sur base du numéro d'entité figurant dans N2, on plonge à un endroit précis de la structure hiérarchique pour y faire un accès direct à une occurrence d'entité;

(2) : rôle de TAB.2 : si, après avoir accédé à une occurrence d'entité sur base de son identifiant, on remarque que ce n'est pas une occurrence de l'entité désirée, on va devoir parcourir un certain nombre de chaînes pour y accéder. Le second type de table permet, pour chaque entité, de reconstituer la structure hiérarchique dans laquelle elle s'insère et qu'il faudra parcourir pour y accéder;

- contenu de traitement 1 : lorsqu'on a trouvé une occurrence de l'entité désirée par un accès direct, on continue en séquence l'algorithme de constitution des lignes à éditer. La sélection n'est en réalité qu'une étape dans la création du fichier des lignes à éditer. Le retour dans cette procédure de recherche se fera avec un nouvel identifiant figurant dans un nouveau descripteur.

- contenu de traitement 2 : ce traitement doit permettre de résoudre les problèmes suivants :

- (1) : retour au bon endroit dans la structure hiérarchique : lorsqu'on a extrait les données nécessaires à la constitution d'une ligne à éditer d'une occurrence d'entité accédée par le parcours d'une chaîne, l'accès à l'occurrence suivante ne se fera pas sur base d'un nouveau descripteur. Il faudra, en fait, lancer un nouvel accès par le même ordre que celui qui a été utilisé pour obtenir l'occurrence d'entité déjà traitée. Il est donc indispensable de mémoriser l'étiquette précédant cet ordre.

Exemple : on a atteint la section administrative 222 au moyen de l'ordre CALL ITRNM(002,017,0).

Pour accéder à la suivante, il faut continuer à parcourir la chaîne indiquée par les numéros d'entités 002, 017 et cela n'est possible qu'en réutilisant l'ordre CALL IRTRNM(002,017,0).

Il faut donc après avoir traité la section 222 revenir dans le programme d'accès à l'endroit où se trouve cet ordre. Celui-ci est repéré grâce à l'étiquette RECH-017 que l'on va donc mémoriser. L'acquisition d'un nouveau descripteur se fera lorsqu'on passera par " fin de recherche " (voir fig. 8.2). Cela signifie que l'on a retrouvé l'occurrence d'entité accédée directement grâce à la portion d'identifiant complète qui figurait dans le descripteur.

- (2) : reconstitution de la situation initiale pour la recherche d'une occurrence d'entité par parcours de chaîne : pour une question relative à une association, on peut avoir la situation suivante :

DEBUT LOCALISATION

E1 : S-AUX : US=8 DEPT=1

E2 : S-AUX : US=8 DEPT=1

E3 : CONSOMMATIONS-UC-SAUX: FIXES.

Ceci signifie que pour chaque section auxiliaire relative à E1, il faudra considérer chacune des sections mentionnées en E2.

En supposant que le département 1 contienne les sections auxiliaires 222, 223 et 224, on aura la séquence d'accès suivante :

222 (pour E1) (3)
 222 (pour E2)
 223 (pour E2)
 224 (pour E2) (1)
 223 (pour E1) (2)
 222 (pour E2)
 223 (pour E2)
 224 (pour E2)
 224 (pour E1)

Lorsqu'on se trouve dans la situation (1) et que l'on veut passer dans la situation (2) au moyen de l'ordre de recherche IRTRNM (= rechercher le suivant), il faut que le système se trouve en fait dans la situation (3) d'où nécessité de repositionner correctement le système.

Ceci est réalisé en mémorisant le numéro de l'entité à laquelle il faudra accéder ainsi que l'identifiant complet.

Le traitement va donc aboutir à la création de la table (fig.8.3) avant d'autoriser la poursuite normale de l'algorithme de constitution des lignes à éditer. Cette table ne sera construite entièrement que pour E1 tandis que pour E2 elle contiendra uniquement l'étiquette de retour.

E1	étiquette retour	numéro d'entité	identifiant
E2	étiquette de retour		

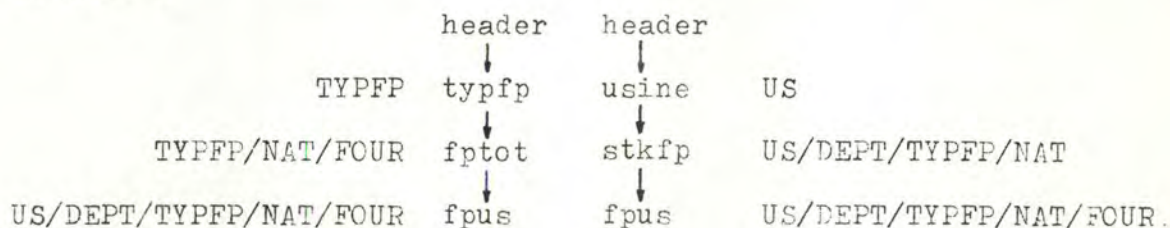
fig.8.3.

La restauration de la situation adéquate au moyen d'un accès direct est suffisante car la table de chaînage gérée par l'IDS contient toujours l'adresse du suivant dans la chaîne.

8.1.1.5. Cas particulier des entités faisant partie de deux structures hiérarchiques.

La fig. 8.1 définit toutes les structures hiérarchiques que l'on rencontre dans la banque de données. Il y apparaît une particularité : certaines entités apparaissent dans des structures hiérarchiques différentes car elles sont détails de deux maîtres distincts. Leur codification renferme celles des deux maîtres. Etant donné que la vérification de la partie localisation d'une requête se fait par rapport à une séquence de base (celle figurant dans le dictionnaire), la structure d'accès à l'entité est unique et correspond à cette séquence.

Ex : les fournitures primaires de l'usine font partie des deux structures suivantes :



C'est la seconde structure qui permet l'accès. Toute entité encadrée indique quel est son chemin d'accès.

8.1.1.6 Cas des entités de nature "relation".

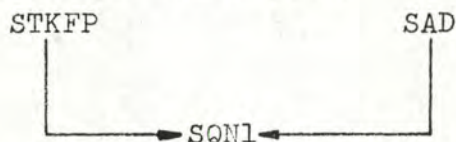
Ceci comprend toutes les entités qui n'ont pas de codification propre tels les standards de consommation. Elles ne figurent pas dans les structures d'accès car dès que l'on a obtenu l'occurrence de l'entité origine de l'association et celle de l'entité cible de l'association, on dispose de tous les éléments pour accéder directement à l'entité de nature relation.

Ex : on désire les consommations en FP de toutes les S-AD de l'usine 8.

```
DEBUT LOCALISATION
      E1 : S-AD : US = 8
      E2 : STKFP : US = 8
      E3 : CONSOMMATIONS-FP : TOTALES
EDITION
      ETAT : SECT
      LISTE : E2-COSTKFP, E2-LIB, E3-STD.
```

FIN

Ceci correspond à la structure suivante :



L'accès à SQN1 est réalisé par l'appel au module I.D.S par l'ordre suivant : CALL IRTISM (n° d'entité de SQN1, identifiant origine, identifiant cible, type de relation, SW)

où - n° d'entité de SQN1 est obtenu en consultant le dictionnaire des associations;

- identifiant origine : obtenu facilement car on dispose déjà de l'occurrence de l'entité origine en working-storage section;
- identifiant cible : même facilité;
- type de relation : ceci correspond au type de standard et permet de distinguer les fixes des variables. La mise à jour de ce paramètre est réalisée grâce au dictionnaire des caractéristiques;
- SW : switch de retour permettant de détecter une erreur quand l'occurrence demandée n'existe pas dans la banque de données.

8.1.2 Sélection par comparaison des valeurs des données avec une valeur fixée par l'utilisateur.

L'analyseur du langage a généré un ensemble de descripteurs dans un ordre correspondant à celui de la notation polonaise postfixée.

Ceci joue un rôle important dans l'utilisation des opérateurs de combinaison.

- procédure de test correspondant aux opérateurs de combinaison (ET,OU)
Elle est basée sur la disposition des descripteurs dans l'ordre de la polonaise postfixée. Pour la réaliser, on considère un vecteur qui joue le rôle d'une pile et dont une composante peut avoir les valeurs suivantes :

- 0 : si le test de comparaison s'est révélé négatif;
- 1 : si le test de comparaison s'est révélé positif.

Ce vecteur à 10 composantes est géré de la manière suivante : après le test de comparaison, on place soit 0, soit 1 dans la composante adéquate et dès que l'on rencontre un opérateur de combinaison, on effectue un test sur les deux dernières positions occupées du vecteur. On diminue l'indice de un, on place le résultat dans la composante pointée par l'indice puis on passe à l'analyse du descripteur suivant. Le résultat de la sélection se trouve toujours dans la première composante du vecteur.

Exemple : On considère l'expression suivante :

A = a OU B = b ET C = c

La forme polonaise étant :

A = a B = b C = c ET OU

Les descripteurs sont générés dans l'ordre :

A	a	=	
B	b	=	
C	c	=	ET
			OU

Les résultats de l'algorithme donnent l'évolution suivante :

I = 0 I = indice du vecteur de test (appelé TEST)

- A = a vérifié I=I+1 (I=1) TEST(1)=1
- B = b non vérifié I=I+1 (I=2) TEST(2)=0
- C = c vérifié I=I+1 (I=3) TEST(3)=1

on rencontre également le 1er opérateur de combinaison ET :

TEST(I) ET TEST(I-1) = 0
I=I-1 (I=2)
TEST(2) = 0

- on rencontre l'opérateur de combinaison OU :

TEST(I) OU TEST(I-1) = 1
I=I-1 (I=1)
TEST(1) = 1

Il n'y a plus de descripteur, on regarde test(1) et on remarque qu'il contient 1. Le résultat est donc positif et l'occurrence d'entité est acceptée.

- procédure de test correspondant aux opérateurs de comparaison

A chaque opérateur de ce type correspondra une procédure préprogrammée dont il reste uniquement à actualiser les opérandes à savoir :

- opérande contenant la valeur de référence (dans l'exemple précédent: a,b,c) : il suffit d'effectuer le transfert de la valeur figurant dans le descripteur dans la variable représentant cet opérande;
- opérande contenant la valeur de la donnée : or ce qui figure dans le descripteur est le nom de la donnée dont on veut utiliser la valeur. Il est nécessaire de passer par l'intermédiaire d'un module écrit dans un langage du type assembleur pour effectuer le transfert. C'est la raison pour laquelle on voit apparaître dans le dictionnaire des données des éléments tels que adresse physique en working-storage section du début de l'article, déplacement par rapport au début de l'article, longueur physique d'une donnée.

La procédure générale de sélection est énoncée en annexe D.

8.2. Calcul de données.

Nous développons dans cette partie les éléments suivants :

- organisation générale du calcul;
- logique interne de chaque opérateur.

8.2.1. Organisation générale du calcul

Lorsque les données calculées passent dans l'analyseur, celui-ci remplit un tableau contenant pour chacune d'entre elles la liste des opérandes ainsi que l'opérateur à appliquer. Le calcul consiste à appeler une routine correspondant à l'opérateur indiqué avec actualisation des opérandes et introduire dans le tableau le résultat de l'opération effectuée. Ceci est réalisé pour chaque occurrence de l'entité spécifiée en E1 pour une question ponctuelle et pour chaque paire d'occurrences des entités mentionnées en E1 et E2 pour une question relative à une association. On exclut cependant de cette procédure les données servant à la détermination d'une autre utilisée comme référence pour des comparaisons et notamment pour le calcul de pourcentages. Une telle donnée sera appelée par la suite : donnée de référence.

Exemples :

1. On désire obtenir pour les sections auxiliaires de l'usine 8 le total des consommations fixes et variables en fournitures primaires.

DEBUT LOCALISATION

E1 : S-AUX : US=8

E2 : STKFP

E3 : CONSOMMATIONS-FP : TOTALES.

CALCUL

R1 = MULT(E1-NIVACT, E3-STD)

R2 = MULT(R1, R2-PRX-CONS-MOY).

EDITION

ETAT : SECT

LISTE: E2-COSKFP, E2-LIB, R1, E2-UN,R2.

FIN

La table des données calculées est la suivante :

1	R1	0	(1,x)	0	(3,y)	1		résultat 1	0	MULT
2	R2	0	(0,1)	0	(2,z)	1		résultat 2	0	MULT
20										

x,y,z = numéros des données dans l'entrée du dictionnaire à laquelle elles appartiennent.

Remarque :

1. On fait figurer dans ce tableau les noms des données calculées (c'est-à-dire R1, R2) car la grammaire du langage n'impose pas la spécification dans **un** ordre croissant, des nombres compris entre 1 et 20. On peut avoir la situation suivante :

R10 (1,x) (3,y) 0 résultat 1 MULT

R2 (0,1) (2,z) 0 résultat 2 MULT

2. la seule restriction émise est la suivante : toute donnée calculée figurant comme opérande d'une autre doit la précéder dans le tableau et par conséquent, doit la précéder dans la rédaction par l'utilisateur de la partie calcul. Ceci est du au parcours séquentiel de la première à la vingtième composante de la table des données calculées .

2. Si on formule la même requête que celle de l'exemple 1 en demandant en plus de comparer les résultats aux consommations totales de fournitures primaires par les sections auxiliaires de l'usine 8, la requête devient :

DEBUT LOCALISATION

E1 : S-AUX : US = 8

E2 : STKFP

E3 : CONSOMMATIONS-FP: TOTALES

E4 : USINE : US = 8.

CALCUL

R1 = MULT(E1-NIVACT, E3-STD)

R2 = MULT(R1, E2-PRX-CONS-MOY)

R3 = PRCT(R2, E4-TOTDEP-DIR-FPSAUX).

EDITION

ETAT : SECT

LISTE: E2-COSTKFP, E2-LIB,R1, E2-UN, R2, R3.

FIN

La table des données calculées devient :

1	R1	0	(1,x)	0	(3,y)	1	résultat1	0	MULT
2	R2	0	(0,1)	0	(2,z)	1	résultat2	0	MULT
3	R3	0	(0,2)	1	(4,u)	1	résultat3	0	PRCT
20									

On remarque que la donnée de référence (TOTDEP-DIR-FPSAUX) est marquée 1. Cela signifie qu'on en effectue la recherche une seule fois puisque c'est toujours la même valeur qui sera utilisée. Mais l'opérateur dans lequel elle figure est marqué 0 ce qui signifie que pour tout couple d'occurrences de E1 et de E2, on effectue le calcul du résultat. Le calcul associé à un opérateur, dont l'indicateur correspondant possède la valeur 1, n'est exécuté qu'une seule fois.

8.2.2. Logique interne de chaque opérateur.

8.2.2.1. Remarque : à chacun des opérateurs devra être associé un module particulier permettant au système d'aller chercher la valeur d'une donnée. Il sera écrit en langage de base par suite de la nécessité de manipuler des adresses. Il aura la structure suivante :

- calcul de l'adresse
- acquisition de la valeur

a) calcul de l'adresse : on se sert du dictionnaire des données où l'on trouve :

- l'adresse d'implantation en WSS de l'entité considérée (1)
- pour chaque donnée figurant dans l'entité
 - déplacement en mots par rapport au début de l'entité (2)
 - déplacement en caractères par rapport au début du mot (3)
 - longueur en caractères de la donnée (4)

Ces informations vont permettre de calculer une adresse et une longueur. La première sera fournie sous la forme du couple ((1)+(2), (3)) et la seconde sera donnée par (4).

b) acquisition de la valeur :

Se pose le problème de la compatibilité entre le format de la donnée et celui de la zone de réception du module. On peut distinguer trois formats dans la banque :

- format x
- format 9 avec usage COMP-1
- format S9(13)V9(5) avec usage COMP-2.

Les deux premiers types de données sont réceptionnés dans une zone tampon définie alphanumérique où le cadrage se fait à gauche. Le dernier l'est dans une zone définie par format S9(13)V9(5).

Le choix de la zone tampon adéquate se fera sur base du format de la donnée qui figure dans le dictionnaire des données.

8.2.2.2. Opérateur d'addition.

C'est le seul opérateur à trois opérandes. Le seul problème pour cet opérateur ainsi que pour les autres est la détermination du type des zones intermédiaires représentant les opérandes et le résultat de l'opération ainsi que celui des zones de réception des valeurs des données figurant dans la banque. Les données déclarées en format X ne seront pas considérées ainsi que celles déclarées en format 9 car elles représentent uniquement des durées d'activité, des durées de simulation ou des "page-range" (notion I.D.S. permettant de localiser en un endroit bien précis de la banque de données un ensemble d'informations). Ces zones auront donc le format unique flottante double précision.

L'algorithme de cette opération figure en annexe C.

8.2.2.3. Opérateurs de soustraction, de multiplication, de division.

Ces opérateurs sont, au contraire du précédent, des opérateurs binaires mais dont la procédure générale de calcul est similaire à celle exposée en annexe C.

8.2.2.4. Opérateur pourcentage.

La procédure de calcul propre à cet opérateur figure en annexe C

8.2.2.5. Opérateur cumul.

C'est un opérateur particulier car son calcul n'est effectué qu'une seule fois et ses opérandes ne font partie que de l'entité spécifiée en E4. Il fournit comme résultat une donnée de référence qui sera utilisée par l'opérateur pourcentage.

Dans le cas où l'entité spécifiée en E4 est identique à celle citée en E1, le fait de manipuler plusieurs occurrences pour E4 va entraîner la perte des informations se trouvant en mémoire et relatives à une occurrence de E1. Il est donc nécessaire de restaurer la situation de E1.

Exemple :

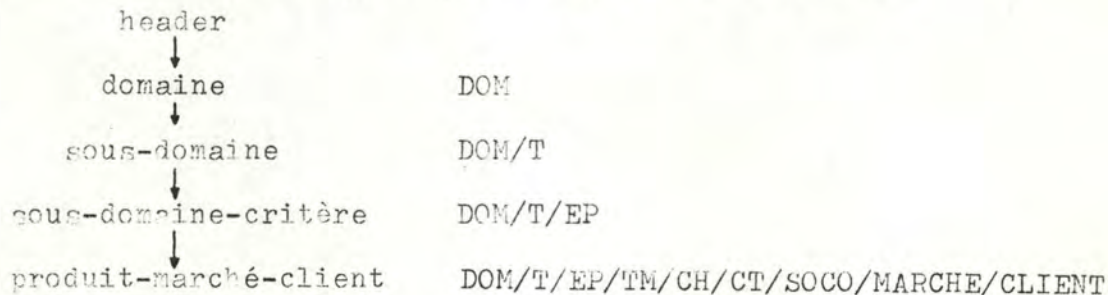
```

DEBUT   LOCALISATION
        E1 : PR-M-CL :  DOM=2  T=1  MARCHE=2222  CLIENT=111
        E4 : PR-M-CL :  DOM=2  T=1  MARCHE=2222.
CALCUL
        R1 = CUMUL(E4-DOM, E4-Q-VENDUE)
        R2 = PRCT(E1-Q-VENDUE, R1).
EDITION
        ETAT : CLIENT
        LISTE: E1-LIB, E1-COPR-M-CL, E1-Q-VENDUE, R1, R2.

FIN

```

La structure hiérarchique à parcourir est la suivante :



Il est fort possible que la valeur du générique EP correspondant à la première occurrence de EI soit différente de celle du même générique relative à la dernière occurrence accédée lors de l'exécution du cumul. Il faut donc restaurer l'ancienne valeur de EP correspondant à EI.

L'algorithme de réalisation figure en annexe C.

8.3. Constitution des fichiers utilisés lors de l'édition.

Ces fichiers sont au nombre de deux :

- fichier des libellés : est celui qui contient toutes les en-têtes, relatives à la partie codification, des états budgétaires.
- fichier d'édition : contient l'ensemble des informations entrant dans la composition de l'état budgétaire et relatives aux parties données et résultats de chaque occurrence d'entité.

8.3.1. Constitution du fichier des lignes à éditer :

un article du fichier se scinde en deux parties (fig.8.4)

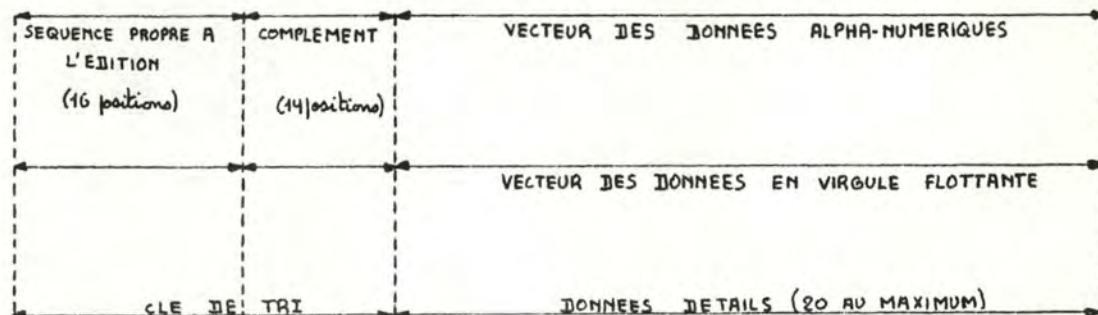


Fig. 8.4.

8.3.1.1. Clé de tri du fichier.

Cette clé permet de ranger l'ensemble des informations dans un ordre correspondant à celui utilisé dans la partie édition.

Exemple :

```

DEBUT   LOCALISATION
        EI : S-PR : US=8  DEPT=1  BLOC=20
        EDITION
        ETAT : DEPT
        SOUS-TOTAL : BLOC
        LISTE : EI-COS-PR, EI-LIB, EI-NIV-ACT.

```

FIN

La séquence de base est US/DEPT/BLOC/SECT.

Il suffit d'y inclure au bon endroit le critère de classe et de caractéristique (celui-ci étant facultatif) : US/DEPT/CLASSE/BLOC/SECT.

Ceci correspond à la séquence par rapport à laquelle sont détectées les ruptures dans la partie édition. Mais on la complète différemment selon que l'on a :

- une question ponctuelle : la partie complément est mise à blanc car l'identifiant de toute l'occurrence de l'entité E1 figure déjà dans la première partie;
- une question relative à une association : on transfère dans la partie complément l'identifiant de toute l'occurrence de E2.

Exemple : DEBUT LOCALISATION

E1 : S-AUX : US = 8 DEPT = 1 SECT = 222

E2 : STKFP :

E3 : CONSOMMATIONS-FP : TOTALES.

CALCUL

R1 = MULT(E1-NIV-ACT, E3-STD)

R2 = MULT(R1, E2-PRX-CONS-MOY).

EDITION

ETAT : SECT

LISTE : E2-COSTKFP, E2-LIB, R1, R2.

FIN

La clé aura la structure suivante :

US	DEPT	SECT	CLASSE	CARACT	code FP
----	------	------	--------	--------	---------

Pour le garnissage de la clé, on rencontre deux problèmes :

- déterminer où introduire la valeur de la classe et de la caractéristique si nécessaire : pour cela il suffit de considérer le (ou les) générique(s) définissant l'état. La valeur de la classe le (ou les) succède. La valeur de la caractéristique est obtenue par consultation du dictionnaire des caractéristiques et suit immédiatement celle de la classe.
- déterminer s'il faut introduire une valeur de caractéristique : pour cela, il faut consulter le dictionnaire des associations et vérifier si celle qui a été utilisée dans la requête possède ou non des caractéristiques.

8.3.1.2. Constitution de la ligne des données détails.

Il faut résoudre ici un problème de compatibilité de zones étant donné les différents formats que l'on rencontre dans la banque. On considère deux vecteurs qui sont parcourus par un indice commun variant de 1 à 10 (car on autorise 10 données au maximum). A une

valeur de ce dernier correspondra une seule valeur qui sera contenue soit :

- dans le vecteur des données alpha-numériques si celle mentionnée possède un format numérique ou alpha-numérique;
- dans le vecteur des données en virgule flottante si celle mentionnée est décrite par un format en virgule flottante.

Exemple : dans l'exemple précédent, on a les données :

E2-COSTKFP qui a un format X(14)

E2-LIB qui a un format X(30)

R1 qui a un format S9(13)V9(5) COMP-2

R2 qui a un format S9(13)V9(5) COMP-2

Les vecteurs seront les suivants :

vecteur alpha-numérique	COSTKFP	LIB			
vecteur virgule flottante			R1	R2	

8.3.2. Constitution du fichier des libellés.

Exemple : DEBUT LOCALISATION
 E1 : S-PR : US = 8 DEPT = 1 BLOC = 20.
 EDITION
 ETAT : BLOC
 LISTE : E1-COS-PR, E1-LIB, E1-NIV-ACT.

FIN

On aura le libellé suivant :

USINE : MOUSTIER	/8
DEPARTEMENT : FLOAT	/81
<u>BLOC</u> : <u>COMPOSITION</u>	<u>/8120</u>
(1)	(2) (3)

- (1) : ceci figure dans le dictionnaire des données associé au générique correspondant;
- (2) : ces libellés seront obtenus en consultant la donnée LIB de l'occurrence d'entité désignée par l'identifiant se trouvant après la "/". Il est possible d'y accéder directement étant donné que le numéro d'entité correspondant au générique figure dans le dictionnaire des données. Celui associé aux critères d'un produit est accessible dans les tables de codification définies au §5.2.3.
- (3) : ceci figure dans l'identifiant de l'occurrence de la section de production accédée et il suffit de l'en extraire.

8.3.2.1. Garnissage de la clé de ce fichier.

Chaque article du fichier est décomposé en une partie clé et une partie libellé. Pour le garnissage de la clé, on doit considérer tous les génériques faisant partie de l'en-tête à savoir tous ceux qui, par rapport à la séquence de base, sont supérieurs ou égaux au premier générique d'état plus ceux mentionnés dans SOUS-TOTAL. On ajoute également l'indicateur de classe et de caractéristique. Le garnissage s'effectue donc sur base du descripteur de la séquence de rupture.

Exemple : on considère la partie édition relative à une requête concernant un stock de produits :

EDITION

ETAT : SECT, T

SOUS-TOTAL : EP

LISTE : ...

La séquence de base étant : US/DEPT/BLOC/SECT/DOM/T/EP/TM/CH/CT

on aura la séquence de rupture : US/DEPT/BLOC/SECT/T/CLASSE/EP.

8.3.2.2. Constitution des libellés.

On a remarqué que trois données doivent figurer dans le libellé associé à un générique :

- libellé complet du générique figurant dans le dictionnaire des données
- le libellé figurant comme donnée dans chaque occurrence d'entité et auquel il faut accéder
- l'identifiant complet correspondant au générique

On associe à chaque occurrence de EL un vecteur de dix composantes (car au maximum dix génériques dans la codification d'une entité) qui seront constituées de trois parties déclarées alphanumériques.

Exemple : structure du libellé associé à la requête de l'exemple précédent. On remarque que le libellé occupe la composante du vecteur correspondant à la position du générique dans la séquence de base.

clé	USINE	MOUSTIER	8	
	DEPARTEMENT	FLOAT	81	
	BLOC	COMPOSITION	8120	
	SECTION			
				DOM absent
	TEINTE			
	EPAISSEUR			
				TM absent
				CH absent
				CT absent

C H A P I T R E 9.

Structure générale de

l'édition.

9.0. Introduction.

Nous présentons dans ce chapitre les éléments nécessaires pour l'édition à savoir :

- les fichiers utilisés : qui sont celui des lignes à éditer, celui des libellés et celui de sortie;
- la définition de l'état imprimé : étant donné que les éléments définissant la constitution de l'état budgétaire sont introduits par l'utilisateur, il est nécessaire de passer par une phase de génération dont le résultat sera un fichier COBOL contenant les différents éléments à insérer dans le texte source du module d'édition. Cette définition comprend :
 - la séquence de rupture : qui est construite sur base des paramètres ETAT et SOUS-TOTAL;
 - les groupes d'édition constituant l'état : aux en-têtes d'état et de classe de données sont associés des groupes de type "control heading" tandis qu'à tous les autres éléments de la séquence de rupture sont associés des groupes de type "control footing";
- les traitements associés à la production d'un groupe (= sections use before reporting) : leur but principal est d'assurer la progression correcte du fichier des libellés. Un autre but est d'assurer les sauts de page adéquats.

9.1. Définition des fichiers utilisés.

- fichier des lignes à éditer : ce fichier est unique par requête et réutilisé par chacune d'entre elles. On en écrira une définition standard et la seule précaution à prendre est d'écraser ce fichier lorsque l'analyse de la requête est terminée;
- fichier des libellés : on agira de la même façon que pour le précédent;
- fichier de sortie : on en écrasera le contenu au début de chaque exécution de la réponse à une requête. On y indiquera également le nom de l'état imprimé.

9.2. Définition de l'état imprimé.

9.2.1. Définition de la séquence de rupture.

C'est à ce niveau que va être définie, sur base des paramètres ETAT et SOUS-TOTAL, la séquence (qui sera décroissante selon l'ordre d'importance) des indicatifs (qui seront les génériques de codification relatifs à l'entité spécifiée en E1) dont le changement de valeur

devra être pris en considération pour la production de l'un ou l'autre groupe d'édition.

9.2.1.1. Contrainte à respecter.

La génération de la séquence de rupture est liée à la clé de tri du fichier des lignes à éditer. Or celle-ci doit respecter la structure hiérarchique de la codification.

Exemple : on demande les consommations de fournitures primaires des sections A et B du département E et des sections A et C du département E. Si l'on omet le département dans la clé, les deux sections A n'en constitueront qu'une seule. Il faut donc faire figurer le département dans la clé.

Elle le doit à une exception près qui est celle où la codification se scinde en deux groupes et que le critère définissant l'état appartient au second groupe.

Exemple : on désire connaître pour tous les clients du marché X les produits qu'ils ont achetés. On aura la partie édition suivante : ETAT = MARCHE, SOUS-TOTAL = CLIENT. Or dans l'entité produit-marché-client, les critères des produits sont hiérarchiquement supérieurs aux critères des marchés. On est obligé d'inverser la priorité des deux groupes de génériques : le second devient donc le plus prioritaire.

9.2.1.2. Construction de la séquence de rupture.

En supposant que le premier générique d'état appartient au premier groupe de codification, on va examiner un à un les éléments à introduire dans la clé de tri du fichier des lignes à éditer :

- la clé de tri comprendra d'abord la séquence de base. Celle-ci peut être modifiée dans le cas où la codification comprend deux groupes de génériques et où :

- le paramètre ETAT renferme un générique de chaque groupe : le second générique se situe immédiatement après le premier;

Exemple : la séquence de base de l'entité produit est : US/DEPT/BLOC/SECT/DOM/T/EP/TM/CH/CT. Si le paramètre ETAT est : SECT, EP, la clé de tri sera : US/DEPT/BLOC/SECT/EP/DOM/T/TM/CH/CT

- le premier générique du paramètre ETAT appartient au second groupe :

l'attitude à avoir a été définie au §9.2.1.1.

- on insérera dans la séquence le niveau de classe de données : ce concept sera introduit à chaque requête même s'il n'apparaît qu'un

seul triplet localisation, calcul, édition que ce niveau permet de distinguer dans le cas où il en existe plusieurs. L'indicateur de classe de données sera directement inférieur au niveau correspondant à ETAT;

Exemple : dans le cas du produit cité précédemment, la clé de tri est :

US/DEPT/BLOC/SECT/EP/CLASSE/DOM/T/TM/CH/CT

- on insérera également le niveau de caractéristique dans le cas d'association qui possède des caractéristiques. Ce niveau sera directement inférieur à celui de classe;

Exemple : dans le cas d'un produit, la cle de tri devient :

US/DEPT/BLOC/SECT/EP/CLASSE/CARACT/DOM/T/TM/CH/CT

- on ajoutera à la séquence de base une zone de quatorze positions qui dans le cas :
 - d'une requête relative à une association sera composée de l'identifiant de l'occurrence d'entité relative à E2;
 - d'une requête ponctuelle sera laissée blanche.

Après avoir vérifié que SOUS-TOTAL respecte l'ordre défini dans la clé de tri, on va déduire la séquence de rupture sur base des paramètres ETAT et SOU-TOTAL. On associe à chaque élément constitutif de cette séquence un code d'édition sur base duquel on décidera du groupe d'édition qu'il faudra adjoindre à l'élément considéré. Ces codes sont les suivants :

- 0 : il correspond à l'ensemble des génériques qui déclenchent des totaux sur base de plusieurs états budgétaires. Ils figurent parmi les génériques qui, dans la clé de tri, précèdent celui (ou ceux) spécifiant l'état;
- 1 : correspond au(x) générique(s) définissant l'état budgétaire;
- 2 : définit la classe de données;
- 3 : définit la caractéristique;
- 4 : il correspond à l'ensemble des génériques qui déclenchent des totaux à l'intérieur d'un état budgétaire. Ils figurent parmi les génériques qui, dans la clé de tri, suivent celui (ou ceux) spécifiant l'état.

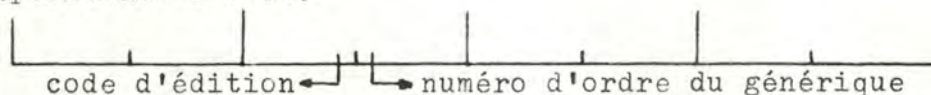


Fig. 9.1.

Pour constituer le descripteur de la séquence de rupture (fig. 9.1), on prend chaque élément cité dans sous-total et on le compare au premier générique d'état. Tant que le numéro d'ordre du générique de sous-total est supérieur ou égal à celui du générique d'état, on le place dans le descripteur avec le code d'édition 1. Dès que cette condition n'est plus vérifiée, on place le (ou les) générique(s) d'état et l'indicateur de classe de données. Ensuite on vérifie, dans le cas d'une association, s'il faut introduire un indicateur de caractéristique. Ensuite on inscrit, dans l'ordre de leur citation, les génériques non encore utilisés du paramètre SOUS-TOTAL.

Le descripteur ainsi constitué va d'abord servir à la définition de la clause CONTROL de la définition de l'état imprimé.

Exemple : on demande les consommations fixes et variables des sections auxiliaires en fournitures primaires. La partie EDITION de la requête contient ETAT = SECT et SOUS-TOTAL = US. Le descripteur de la séquence de rupture est :

1	1	2	3	3	4
---	---	---	---	---	---

La classe de données et la caractéristique ne contiennent pas de numéro de générique. La clause CONTROL sera :
CONTROLS ARE US,ETAT,CLASSE,CARACT,CLE

L'apparition de CLE dans la clause CONTROL va servir pour la mise à jour d'accumulateurs permettant la totalisation de données dont on demande un montant récapitulatif.

Le descripteur de la séquence de rupture va servir également à la génération de la clé de tri du fichier des libellés. Elle ne devra contenir que les génériques supérieurs ou égaux à ceux spécifiés dans ETAT plus ceux de SOUS-TOTAL qui leur sont inférieurs ainsi que l'indicateur de classe et de caractéristique. Trois problèmes importants sont à éclaircir :

- cas d'une classe de données ou d'une caractéristique : en supposant que l'on demande les consommations fixes et variables en fournitures primaires des sections de production A et B avec les paramètres ETAT=BLOC et SOUS-TOTAL = SECT, la clé de tri sera : US/DEPT/BLOC/CLASSE/CARACT/SECT. On obtiendra la présentation suivante :

	FP 1
	FP 2
TOTAL SECT A	
	FP 3
TOTAL SECT B	
TOTAL CONS FIXES	
	FP 4
	FP 5
TOTAL SECT A	
TOTAL CONS VARIABLES	

On remarque que le libellé SECT A doit être utilisé deux fois. Il faut le dédoubler et c'est la raison pour laquelle on a introduit la caractéristique dans la clé du tri. La raison de l'introduction de la classe de données est identique.

- quand construire le libellé : celui-ci étant relatif à une occurrence d'entité de E1, on le construit pour chaque occurrence accédée. Ceci risque d'introduire plusieurs fois un seul libellé.

Exemple : on a pour une section de production, le paramètre ETAT = BLOC et le paramètre SOUS-TOTAL est absent. Il peut se passer la chose suivante :

US 1	DEPT 1	BLOC 1	SECT 1
US 1	DEPT 1	BLOC 1	SECT 2

on obtient deux fois le libellé relatif aux génériques US1/DEPT1/BLOC1. Il faudra en supprimer un et cela sera réalisé, grâce à la clé de tri, lors du tri.

- comment faire progresser ce fichier :

Exemple : on désire obtenir les quantités en stock selon la mise en page suivante :

ETAT = SECT; SOUS-TOTAL = BLOC, EP

et on considère les occurrences suivantes;

US1	DEPT1	BLOC1	SECT1	EP11	LIB1(US1,DEPT1,BLOC1,SECT1,EP11)
				EP12	LIB2(US1,DEPT1,BLOC1,SECT1,EP12)
			SECT2	EP21	LIB3(US1,DEPT1,BLOC1,SECT1,EP21)
				EP22	LIB4(US1,DEPT1,BLOC1,SECT2,EP22)
	BLOC2	SECT3	EP31		LIB5(US1,DEPT1,BLOC2,SECT3,EP31)

Sur section, on définit un groupe "control heading" (CH) et sur les autres un groupe "control footing" (CF). La séquence, dans laquelle les groupes d'édition doivent être pris en compte, est la suivante :

CH_{SECT1} CF_{EP11} CF_{EP12} CH_{SECT2} CF_{EP21} CF_{EP22} CF_{BLOC1}

Si on fait progresser le fichier des libellés à la détection d'une rupture sur chaque groupe d'édition, on obtient la colonne (2) du tableau fig.9.2.

(1) : séquence correcte	(2)	(3)	(4)
CH _{SECT1} LIB1	LIB1	LIB1	LIB1
CF _{EP11} LIB1	LIB2	LIB1	LIB1
CF _{EP12} LIB2	LIB3	LIB2	LIB2
CH _{SECT2} LIB3	LIB4	LIB3	LIB3
CF _{EP21} LIB3	LIB5	LIB3	LIB3
CF _{EP22} LIB4	LIB6	LIB4	LIB4
CF _{BLOC1} LIB4	LIB7	LIB5	LIB4
CH _{SECT3} LIB5	LIB8	LIB6	LIB5

fig.9.2.

On remarque que tout groupe CF succédant directement au groupe CH correspondant à l'en-tête de l'état budgétaire utilise les mêmes informations que ce dernier. Il est donc inutile de faire progresser le fichier à ce moment. Cela sera fait en positionnant dans les traitements associés au groupe CH, un indicateur qui sera positionné à 1 à la production de ce groupe. Il interdira la progression du fichier des libellés à la production du premier groupe CF. C'est à ce moment que l'indicateur sera repositionné à 0. On obtient la colonne(3) Un nouveau décalage apparaît lorsque plusieurs groupes CF se succèdent. En se basant sur le fait que, toute rupture d'un niveau supérieur entraîne une rupture des niveaux inférieurs, on ne fera progresser le fichier des libellés que lors de la production du groupe CF du niveau inférieur car c'est lui qui est produit le premier. Il faut bien sûr que ce niveau ne suive pas directement un groupe CH car dans ce cas, on ne fera pas non plus progresser le fichier. On obtient la colonne (4) qui donne la séquence correcte.

9.2.2 Définition des groupes d'édition et des traitements associés.

9.2.2.1 Définition du groupe d'en-tête = groupe de type "control heading" (CH).

Chaque ligne constituant ce groupe sera composée du libellé complet du générique, du contenu de la donnée LIB de l'occurrence d'entité correspondant à la valeur du générique et du code complet.

Ces données figurent dans un article du fichier des libellés sous la forme décrite au §8.3.2.2. La première ligne de l'en-tête devra également contenir le titre du budget, qui sera accessible dans le dictionnaire des données, ainsi qu'un compteur de pages et la seconde, le trait de soulignement de ce titre.

Exemple : si on interroge les sections de production avec le paramètre ETAT = BLOC, l'en-tête sera :

<u>BUDGET DES SECTIONS DE PRODUCTION</u>	USINE : MOUSTIER	/8
	DEPARTEMENT : FLOAT	/81
	BLOC : COMPOSITION	/8120

Les traitements associés à la production de ce groupe devront réaliser les opérations suivantes :

- avancement dans le fichier des libellés;
- effectuer un saut de page;
- remise du compteur de pages à 1;
- positionner un indicateur associé à la production du groupe à 1.

9.2.2.2 Définition du groupe d'en-tête de classe de données = groupe de type "control heading" (CH).

Ce groupe est composé :

- du libellé de classe de données : la nature de celui-ci dépend du type de question posée :
 - pour une question relative à une association : ce libellé sera le nom de l'association qui est mentionné dans la requête et qui est disponible dans le dictionnaire des associations;
 - pour une question ponctuelle : le libellé d'association sera une zone blanche;
- du libellé d'en-tête, et cela pour chaque donnée, de l'ensemble de ses valeurs : ce libellé est constitué du nom de la donnée qui est disponible dans le dictionnaire des données. Etant donné qu'une requête peut contenir plusieurs classes de données et que la description du groupe d'édition est statique, il est nécessaire d'actualiser, avant la production du groupe, les zones dans lesquelles l'éditeur viendra chercher les valeurs à éditer. Ces zones appartiennent à une table composée de cinq vecteurs, un par classe de données. Chacun d'eux est composé de onze composantes dont la première contiendra le libellé de classe et les dix autres les noms des données utilisées.

Cette table sera garnie avant l'édition et le choix d'une composante se fait sur base de la valeur de l'indicateur CLASSE dans la clé de tri. La possibilité de présenter plusieurs classes de données dans un même état budgétaire va entraîner les restrictions suivantes :

- nécessité pour chaque classe d'avoir la même structure de présentation : si on considère pour une section les consommations de fournitures primaires et d'unités d'oeuvre de sections auxiliaires, et que l'on définit pour le premier type de consommations la présentation suivante :

consommations FP	Quantité	Unité	Prix Unitaire	Valeur
------------------	----------	-------	---------------	--------

 il faut que celle-ci soit également valable pour les unités d'oeuvre de sections auxiliaires. Cela est dû au fait que la description de classe est unique et que les positionnements sur une ligne sont fixés une fois pour toutes;
- nécessité d'un même format d'édition pour les données occupant une même position : ceci est dû à la définition du groupe détail (voir suivant) dans lequel on donne le format d'édition d'une donnée entrant dans la composition d'une ligne à éditer. Il est unique et valable pour toute valeur;

Exemple : à quantité correspond le format S9(10)V99. Toutes les données apparaissant au même endroit que quantité seront éditées selon ce format.
- nécessité pour les données d'une même colonne de posséder les mêmes critères de totalisation : car à chaque donnée, on associe un accumulateur selon que la donnée est totalisable ou non (voir § 9.2.2.4.)

9.2.2.3 Définition du groupe détail (DE) correspondant à la ligne à éditer.

La structure de ce groupe est identique à celle du groupe précédent. Sur base du descripteur de la ligne d'édition, on obtiendra le format d'édition de la donnée ainsi que sa longueur qui se trouvent dans le dictionnaire des données. La valeur de chaque donnée est obtenue en consultant le fichier des lignes à éditer que l'on fera progresser avant de lancer l'édition d'une ligne.

9.2.2.4 Définition du groupe associé à l'indicateur de rupture CLE = groupe de type "control footing" (CF)

Remarque préliminaire : c'est à partir de ce groupe que l'on va définir ceux qui permettront d'effectuer les totalisations intermédiaires.

Le nombre de groupes à générer est fourni par le descripteur de la séquence de rupture.

Dans chaque groupe, sauf pour celui correspondant à la première composante du vecteur, on définit des accumulateurs au moyen de la clause SUM du langage COBOL. Ils servent à cumuler les résultats intermédiaires et seront remis à zéro par le groupe directement supérieur à celui considéré. Pour le plus haut niveau de rupture, on ne déclare aucun accumulateur car son rôle est uniquement d'entraîner l'impression des accumulateurs du niveau inférieur puis de les remettre à zéro. Les données totalisables sont composées des valeurs ou montants. Par suite du manque de temps, on n'a pu traiter complètement le cas des quantités et on les déclare dès lors non totalisables ainsi que les autres types de données.

définition du groupe : c'est dans ce groupe que l'on va associer à chaque donnée totalisable un accumulateur qui est défini de la manière suivante :

- il porte toujours le même nom à savoir TOTAL;
- auquel on adjoint deux indices :
 - i : numéro du groupe de "control footing" auquel appartient cet accumulateur;
 - j : numéro de l'accumulateur correspondant à une donnée totalisable. Etant donné que le nombre de ces données est identique dans chacun des groupes, une même donnée, dans des groupes distincts, est toujours totalisée dans un accumulateur de même numéro j.

9.2.2.5 Définition des groupes associés au code d'édition 4 = groupe de type "control footing" (CF)

Ce sont les groupes correspondant aux génériques de codification indiquant les totaux à produire à l'intérieur de l'état budgétaire. Le total est accompagné d'un libellé qui figure en partie dans le dictionnaire des données(1) et en partie dans un article du fichier des libellés (2).

Exemple : pour un total au niveau d'un domaine, on a le libellé :

```
TOTAL      DOMAINE(1)      FLOAT(2)
```

Dans les traitements associés à la production d'un groupe, il faut, pour le générique correspondant au niveau inférieur, tester si un indicateur associé au groupe d'en-tête a été positionné à 1. Si oui, on le repositionne à 0. Si non, on fait progresser le fichier des libellés.

S'il n'existe aucun générique de code d'édition 4, la progression du fichier des libellés n'est effectuée qu'au niveau du groupe d'en-tête d'état de type CH.

9.2.2.6 Définition des groupes associés au code d'édition 3 = groupe de type "control footing" (CF).

Le problème est de disposer du libellé de caractéristique adéquat. Etant donné que chaque caractéristique possède un code d'édition distinct, fixe a le code 1 et variable le code 2, c'est sur base de ce dernier que l'on actualisera le libellé. Dans le groupe CH associé à la classe de données, on mémorise la valeur de la caractéristique en garnissant une zone "ancienne valeur". C'est sur base de cette zone que l'on mettra à jour le libellé et ensuite, on mettra à jour cette zone lorsque la rupture a été détectée. Ces deux dernières opérations sont réalisées dans les traitements associés à la production du groupe CF.

9.2.2.7 Définition des groupes associés au code d'édition 2 = groupe de type "control footing" (CF).

Le problème est de disposer du libellé de classe de données or celui-ci figure dans la zone intermédiaire où il a été mis à jour par le groupe du type CH associé à la classe. Il suffit donc de le prendre à cet endroit.

9.2.2.8 Définition des groupes associés au code d'édition 0 = groupe de type "control footing" (CF).

Les totaux définis à ce niveau portant sur plusieurs états budgétaires, ils seront imprimés en bas de l'état. Les libellés nécessaires ont déjà mis à jour précédemment d'où il n'est pas nécessaire de faire progresser ce fichier ici.

Remarque : Pour chacun des groupes définis dans les points 9.2.2.5 à 9.2.2.8, on introduit des accumulateurs selon la forme présentée au point 9.2.2.4.

A N N E X E S .

- Expression de l'activité des sections de production

pour un s donné

$$\forall k \quad \text{NIVACT-SPR}(s)(k) = \sum_l \text{NIVPROD}(s)(l) \quad \text{STANDARD de SPR}(k)(l)$$

où s = stade de fabrication

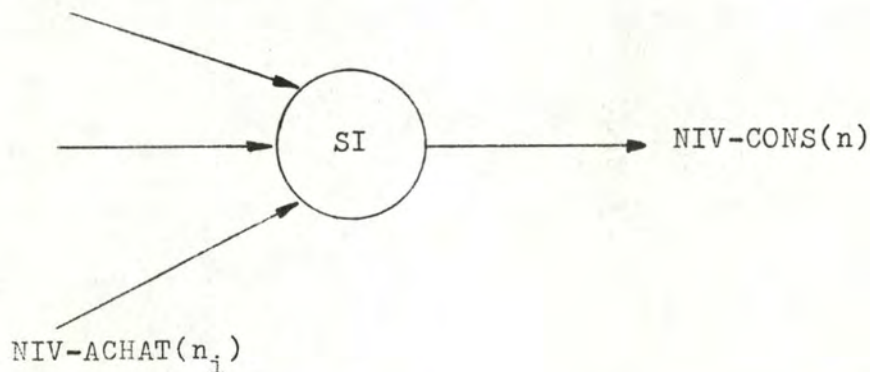
k = indice de la section de production

l = indice de tous les produits intermédiaires
pour lesquels la section k a presté

std = standard de consommation d'unité d'oeuvre
de section de production par un produit.

- Expression du niveau du stock de natures de dépenses.

$$\forall n \quad \text{SI}(n) + \sum_i \text{NIV-ACHAT}(n_i) = \text{NIV-CONS}(n) + \text{SF}(n)$$



où n = nature de dépenses

i = indice des sources d'approvisionnements

SI = stock initial

SF = stock final

Les autres équations peuvent être trouvées dans le document
correspondant relatif au simulateur.

Cet algorithme correspond à la partie analyse de l'élément de base reconnu et analyse du séparateur reconnu.

On aura initialisé à 1 l'indice courant (appelé IC-TAB) du tableau associé à l'automate déterministe.

étape 1 : pas 11 : tester si le terminal lu est identique à celui contenu dans la composante du tableau désignée par IC-TAB.
 si oui : aller à l'étape 2
 si non : aller à l'étape 3

étape 2 : pas 21 : faire l'action correspondant à l'élément de base reconnu.
pas 22 : placer la valeur de SUCC dans IC-TAB
pas 23 : tester si SUCC de IC-TAB est égal à V
 si oui : fin analyse de la requête
 si non : lancer la reconnaissance de l'élément de base suivant

étape 3 : pas 31 : placer la valeur de ALT dans IC-TAB
pas 32 : tester si IC-TAB est égal à F
 si oui : la requête est erronée : message d'erreur à l'utilisateur et on stoppe la requête.
 si non : aller à l'étape 1.

Algorithme d'analyse.

Cet algorithme va reprendre pour chaque numéro de quadruple les actions à entreprendre. Ne seront pas repris, ceux pour lesquels aucune action n'est nécessaire hormis la vérification du fait que l'élément de base reconnu devait effectivement être mentionné.

2(localisation) :

mettre I-GEN-RS à 1. Cette variable permet de n'effectuer qu'une seule fois la génération de la report section dans le cas où plusieurs classes de données figurent dans la requête.

- mise à 0 du compteur de classe de données

- ouverture de la banque de données.

3(EI) :

étape 1 : remettre à blanc le tableau servant à générer les identifiants (TAB-IDENT)

étape 2 : positionner à 1 le 1er indice du tableau qui contiendra les 4 entrées du dictionnaire relatives à la classe de données c'est-à-dire I-TAB-DICT = 1.

5(nom-d'entité) :

étape 1 : tester si cette entité existe

si oui : aller à l'étape 2

si non : erreur; fin de l'analyse de la requête.

étape 2 : charger l'entrée du dictionnaire des données lui correspondant dans le tableau TAB-DICT à la position indiquée par I-TAB-DICT.

7(générique) :

étape 1 :

pas 11 : positionner à 1 l'indice de colonne de TAB-IDENT.

pas 12 : positionner à 0 le compteur de génériques (COMPT-GEN) et un indicateur (INDIC) qui permet de savoir lors de l'analyse des génériques si l'identifiant est complet ou non.

pas 13 : initialiser un tableau qui permet de savoir jusqu'à quel niveau d'une structure hiérarchique l'identifiant est complet et le numéro d'entité correspondant.

Sa valeur initiale est :

	001	0

(1) (2) (3)

- (1): indice du générique correspondant à ce niveau
 (2): numéro d'entité correspondant à ce niveau
 (3): niveau hiérarchique pour lequel l'identifiant est complet.

étape 2 :

- pas 21 : ajouter 1 à COMPT-GEN; le numéro de niveau associé à COMPT-GEN = 5
 si oui : erreur car le générique lu ne figure pas dans la codification de l'entité
 si non : aller au pas 2.2.
- pas 22 : tester si INDIC = 1
 si oui : aller au pas 2.15
 si non : aller au pas 23
- pas 23 : tester si le générique lu est égal au générique de la séquence de base indiqué par COMPT-GEN
 si oui : aller au pas 24
 si non : aller au pas 26
- pas 24 : tester si le niveau du générique lu est supérieur au niveau déjà atteint
 si oui : aller au pas 2.5
 si non : fin de l'analyse du générique lu
- pas 25 : mettre à jour le tableau contenant les niveaux atteints et décrit au pas 1.3. La deuxième ligne de celui-ci vient occuper la première et on y introduit le numéro du générique atteint, le numéro d'entité correspondant et le niveau; fin de l'analyse du générique lu. Le numéro d'entité sera également placé dans RT-CORRECT (c'est celui qui figure dans la seconde ligne)
- pas 26 : mettre des " * " dans les positions relatives au générique indiqué par COMPT-GEN.

- pas 2.7 : ajouter 1 à COMPT-GEN, le numéro de niveau associé à
COMPT-GEN = \bar{b}
si oui : erreur
si non : aller au pas 2.8.
- pas 2.8 : le générique lu est-il égal au générique de la séquence de
base indiqué par COMPT-GEN
si oui : aller au pas 2.6.
si non : aller au pas 2.9.
- pas 2.9 : le niveau du générique lu est-il égal au niveau déjà atteint
si oui : aller au pas 2.10
si non : aller au pas 2.11
- pas 2.10 : mettre dans la variable RT-CORRECT le numéro d'entité pour
lequel l'identifiant est complet, (qui est celui \in lère
ligne), aller pas 2.14
- pas 2.11 : le niveau du générique lu est-il égal au niveau du géné-
rique qui suit celui dont le numéro dans la séquence de
base figure dans la seconde ligne du tableau défini au
pas 1.4.
si oui : aller au pas 2.12.
si non : aller au pas 2.13.
- pas 2.12 : mettre dans la variable RT-CORRECT le numéro d'entité qui
figure dans la seconde ligne; aller au pas 2.14.
- pas 2.13 : mettre dans la variable RT-CORRECT le numéro d'entité qui
figure dans la première ligne;
- pas 2.14 : mettre INDIC à 1, fin de l'analyse du générique.
- pas 2.15 : le générique lu est-il égal au générique de la séquence de
base indiqué par COMPT-GEN
si oui : fin de l'analyse du générique
si non : aller au pas 2.16.
- pas 2.16 : mettre * dans les positions relatives au générique indi-
qué par COMPT-GEN
- pas 2.17 : ajouter 1 à COMPT-GEN, le numéro de niveau associé à
COMPT-GEN = \bar{b}
si oui : erreur car le générique lu ne figure pas dans
la séquence de base
si non : aller au pas 2.15.

8(ALL) :

étape 1 : générer le descripteur suivant :

numéro de l'entité désirée	001	
----------------------------	-----	--

la clé de ce descripteur sera : 001 + un descripteur de fin de génération

10(zone) :

étape 1 : charger cette zone ~~xv~~ dans le tableau des identifiants à l'endroit spécifié par le couple (COMPT-GEN, indice de colonne)

11(,) :

étape 1 : ajouter 1 à l'indice de colonne de TAB-IDENT

12(générique) :

étape 1 : remettre à 1 l'indice de colonne de TAB-IDENT

étape 2 : appeler le module d'analyse du générique qui a été décrit à l'étape 2 du point 7

Les quadruples 13, 14, 15, 16 devront pouvoir effectuer la construction de tous les descripteurs relatifs à l'entité spécifiée en E1.

L'algorithme de composition est le suivant :

étant donné qu'une entité possède 10 génériques au maximum, on aura 10 séquences identiques à celles qui vont être **indiquées** dans le cas de 3 génériques.

CLE-E1 = 001

I1 = 0.

I1-PAR

I1 = I1+1.

IF I1 > 25 OR TAB-IDENT(1,I1) = **†** THEN fin de génération des descripteurs.

placer TAB-IDENT (1,I1) dans la partie identifiant du descripteur

I2 = 0

I2-PAR

I2 = I2 + 1

IF I2 > 25 THEN GOTO I1-PAR

IF TAB-IDENT(2,I2) = **†** THEN IF I2 = 1 THEN générer 1 descripteur;
ajouter 1 à CLE-E1; GOTO I1-PAR

ELSE GOTO I1-PAR.

placer TAB-IDENT(2,I2) dans la partie identifiant du descripteur
I3 = 0.

I3-PAR

I3 = I3 + 1

IF I3 > 25 THEN GOTO I2-PAR.

IF TAB-IDENT(3,I3) = **t** THEN IF I3 = 1 THEN générer 1 descripteur;
ajouter 1 à CLE-E1; GOTO I2-PAR
ELSE GOTO I2-PAR.

placer TAB-IDENT(3,I3) dans la partie identifiant du descripteur.

Remarque : la génération du descripteur renferme les actions suivantes :

- transférer le numéro d'entité associé à E1 dans A
- transférer RT-CORRECT dans B

A	B
---	---

13(;) : on va avoir une sélection sur nom de donnée. Mise de l'indice de la pile à 0. Positionner un indicateur (INDICATEUR) à 1 pour l'initialisation de la clé de stockage des descripteurs

14(E2) : après avoir générer les descripteurs relatifs à E1, mise à **t** de TAB-IDENT et positionner à 2 le premier indice du tableau qui contiendra les entrées du dictionnaire relatives à la requête (I-TAB-DICT = 2)

15(E4) : mise à de TAB-IDENT et positionner à 4 I-TAB-DICT

19(nom-de-donnée) :

étape 1 : réinitialiser (par la mise à **t**) du descripteur de sélection sur nom-de-donnée.

étape 2 : tester si INDICATEUR vaut 1.

si oui : mettre 151 dans CLE-E1-DONNEE; INDICATEUR = 0

si non : aller à l'étape 3 .

étape 3 : tester si le nom de donnée figure dans l'entité mentionnée

si oui : aller à l'étape 4.

si non : erreur car donnée inexistante; fin de l'analyse de la requête.

étape 4 : placer le nom de la donnée dans la descripteur.

pour l'opérateur parmi ceux cités entre 20 et 25 :

étape 1 : placer l'opérateur de comparaison dans le descripteur

27(zone αv) :

étape 1 : placer la valeur de cette zone dans le descripteur.

29(ET) :

étape 1 :

pas 11 : tester l'indice de la pile.

si égal 0 : aller au pas 1.2

si non égal 0 : aller au pas 1.3

pas 12 : ajouter 1 à l'indice et placer l'opérateur de combinaison au sommet de la pile ; générer le descripteur; ajouter 1 à CLE-El-DONNEE;

fin de l'analyse de l'opérateur

pas 13 : tester si la priorité associée à l'opérateur lu est supérieure à celle de l'opérateur figurant au sommet de la pile.

si oui : aller au pas 1.2

si non : aller au pas 1.4

pas 14 : faire $I = 1$

pas 15 : tester si $I = 1$

si oui : aller au pas 1.6

si non : aller au pas 1.9

pas 16 : placer l'opérateur de combinaison se trouvant au sommet de la pile dans le descripteur, le stocker et ajouter 1 à CLE-El-DONNEE

pas 17 : ajouter 1 à I et retrancher 1 à l'indice de pile.

pas 18 : tester si l'indice de la pile = 0

si oui : ajouter 1 à IC-PILE, placer l'opérateur au sommet, fin de l'analyse

si non : aller au pas 1.3

pas 19 : générer un descripteur où seul figurera l'opérateur de combinaison du sommet de la pile.

pas 1.10: ajouter 1 à I et retrancher 1 à l'indice de la pile; aller au pas 1.8.

30(OU) : l'algorithme est identique à celui énoncé pour ET.
 Pour 31(E2), 32(E4) et 33(.), il faut vider le contenu de la pile.

étape 1 :

pas 1.1 : placer l'opérateur de combinaison du sommet de la pile dans le descripteur.

pas 1.2 : retrancher 1 à l'indice de la pile

étape 2 :

pas 2.1 : tester si l'indice de pile = 0

si oui : on a vidé la pile; générer le descripteur de fin

si non : aller au pas 2.2.

pas 2.2 : remettre à **t** la zone du descripteur et ajouter 1 à CLE-EI-DONNEE.

pas 2.3 : placer l'opérateur de combinaison du sommet de la pile dans le descripteur, le stocker et retrancher 1 à l'indice de la pile; aller au pas 2.1.

31(E2) :

étape 1 : mise à **t** de TAB-IDENT

étape 2 : positionner à 2 le premier indice du tableau contenant les entrées du dictionnaire relatives à la requête à savoir I-TAB-DICT.

32(E4) :

étape 1 : mise à **t** de TAB-IDENT

étape 2 : positionner à 4 I-TAB-DICT

35(nom-d'entité) :

étape 1 : tester si cette entité existe

si oui : aller à l'étape 2

si non : erreur; fin de l'analyse de la requête

étape 2 : charger l'entrée du dictionnaire correspondante dans TAB-DICT(I-TAB-DICT)

étape 3 : positionner COMPT-PAS à 0

37(générique) : voir point 7 annexe **B.3**

38(ALL) : voir point 8 annexe **B.5** avec 1 valeur de clé de stockage = 201 + un descripteur de fin de génération

40(zone) :

étape 1 : charger cette zone **vr** dans le tableau des identifiants à l'endroit spécifié par le couple (COMPT-GEN, indice de colonne)

42(,) :

étape 1 : ajouter 1 à l'indice de colonne du tableau des identifiants.

43(générique) :

étape 1 : appeler le module d'analyse du générique qui a été décrit à l'étape 2 du point 7.

étape 2 : remettre l'indice de colonne à 1.

Lorsque l'on rencontre les quadruples 44 (;) ou 45(E3) il faut effectuer la génération des descripteurs relatifs à l'entité spécifiée en E2.

L'algorithme est identique à celui énoncé en annexe B.5 mais où CLE-E1 = 001 sera remplacé par CLE-E2 = 201 et ajouter 1 à CLE-E1 par ajouter 1 à CLE-E2.

44(;) : on a une sélection sur nom-de-donnée. Mise de l'indice de la pile à 0. Positionner l'indicateur INDICATEUR à 1 qui sert à l'initialisation de la clé de stockage des descripteurs.

45(E3) :

étape 1 : placer 301 dans CLE-E3 (= clé de stockage du descripteur de l'association).

48(nom-de-donnée) :

étape 1 : réinitialiser (par la mise à 0) du descripteur de sélection sur nom-de-donnée.

étape 2 : tester si INDICATEUR vaut 1.
si oui : mettre 251 dans CLE-E2-DONNEE, INDICATEUR = 0
si non : aller à l'étape 3.

étape 3 : tester si le nom de donnée figure dans l'entité mentionnée.
si oui : aller à l'étape 4.
si non : erreur car donnée inexistante; fin de l'analyse de la requête.

étape 4 : placer le nom de la donnée dans le descripteur.

49 à 54 (opérateurs de comparaison) : placer l'opérateur reconnu dans le descripteur.

56(zone) :

étape 1 : placer la valeur de cette zone dans le descripteur.

58(ET) :

étape 1 : voir étape 1 du point 29 de l'annexe B.7 où CLE-E2-DONNEE remplace CLE-E1-DONNEE.

59(OU) : l'algorithme est identique à celui énoncé pour ET.

60(E3) :

étape 1 : il faut vider le contenu de la pile. L'algorithme est identique à celui mentionné au point 31,32,33 de l'annexe **B.8** où CLE-E2-DONNEE remplace CLE-E1-DONNEE.

étape 2 : placer 301 dans CLE-E3 (=clé de stockage du descripteur de l'association)

62(libellé d'association) :

étape 1 : accéder au dictionnaire des associations sur base des numéros des entités spécifiées en E1 et en E2.

étape 2 : correspond-t-il une association à ces 2 numéros?

si oui : aller à l'étape 3

si non : erreur car on a spécifié une association inexistante; fin de l'analyse de la requête.

étape 3 : le libellé d'association indiqué dans la requête correspond-t-il à celui figurant dans le dictionnaire ?

si oui : aller à l'étape 4.

si non : erreur, fin de l'analyse de la requête.

étape 4 : générer le descripteur de l'association.

64(E4) :

étape 1 : stocker le descripteur de l'association en CLE-E3.

étape 2 : mise à **t** de TAB-IDENT et positionner à 4 I-TAB-DICT.

65(.) :

étape 1 : stocker le descripteur de l'association en CLE-E3.

66(caractéristique) :

étape 1 : tester par rapport au dictionnaire des caractéristiques si le libellé est correct

si oui : aller à l'étape 2

si non : erreur; fin de l'analyse de la requête.

étape 2 : placer le libellé de la caractéristique dans le descripteur de l'association.

68(nom d'entité) : L'algorithme est identique à celui mentionné au point 35 de l'annexe **B.8**.

70(générique) : L'algorithme est identique à celui du point 7 annexe **B.3**

71(ALL) : L'algorithme est identique à celui du point 8 annexe B.5 où la clé de stockage aura la valeur 401 + générer un descripteur de fin.

73(zone) : charger la valeur de cette zone dans le tableau des identifiants à l'endroit spécifié par le couple (COMPT-GEN, indice de colonne).

74(,) : ajouter 1 à l'indice de colonne du tableau des identifiants

75(générique) :

étape_1 : appeler le module d'analyse du générique qui a été décrit à l'étape 2 du point 7 annexe B.3

étape_2 : remettre l'indice de colonne à 1.

Lorsqu'on rencontre le quadruple 70(.), il faut effectuer la génération des descripteurs relatifs à l'entité spécifiée en E.4. L'algorithme est identique à celui énoncé en annexe B.5 mais où CLE-E1 = 001 sera remplacé par CLE-E4 = 401 et ajouter 1 à CLE-E1 par ajouter 1 à CLE-E4.

77(calcul) : initialisation (par la mise à **1**) de la table des données calculées + mise à 1 de l'indice courant de cette table.

79(édition) : initialisation (par la mise à **1**) des différents descripteurs d'édition.

80(indice) :

étape_1 : vérifier si la valeur de ce dernier est comprise entre 1 et 20.

étape_2 : placer R et valeur de l'indice dans la case correspondante au nom de la donnée de la composante indiquée par l'indice courant.

81 à 86(opérateurs ADD,SOUST,MULT,DIV,PRCT) : placer l'opérateur dans la case correspondante.

87(CUMUL) :

étape_1 : mise à 1 de l'indicateur associé à l'opérateur.

étape_2 : placer l'opérateur dans la case correspondante.

88(parenthèse ouvrante) : positionner le compteur d'opérandes à 1

89(E1) : ajouter 1 au compteur d'opérandes et mettre x (voir le couple x,y défini en 91) à 1.

91(nom-de-donnée) :

étape 1 : vérifier si cette donnée existe.

si oui : aller à l'étape 2

si non : erreur; fin de l'analyse de la requête.

étape 2 : placer cette donnée dans la case correspondante sous la forme (x,y)

où x = composante du tableau contenant les entrées du dictionnaire relatives à la requête.

y = position de la donnée dans l'entrée du dictionnaire.

92(E2) : ajouter 1 au compteur d'opérandes et mettre x à 2.

94(nom-de-donnée) : même raisonnement que pour 91.

95(E3) : ajouter 1 au compteur d'opérandes et mettre x à 3.

97(nom-de-donnée) : même raisonnement que pour 91

98(F4) :

étape 1 : positionner à 1 l'indicateur associé à la donnée.

étape 2 : ajouter 1 au compteur d'opérandes et mettre x à 4.

100(nom-de-donnée) : même raisonnement que pour 91

101(R) : ajouter 1 au compteur d'opérandes et mettre x à 0

102(indice) :

étape 1 : vérifier si ce nom existe dans les composantes déjà garnies de la table des données calculées.

si oui : aller à l'étape 2.

si non : erreur, fin de l'analyse de la requête.

étape 2 : placer cette donnée sous la forme (o,z) dans la case correspondante

où z = numéro d'ordre de cette donnée dans la table

étape 3 : tester si l'indicateur de l'opérateur qui lui correspond vaut 1

si oui : positionner à 1 l'indicateur associé à la donnée

si non : positionner à 0 l'indicateur associé à la donnée

104(parenthèse fermante) :

étape 1 : tester si compteur d'opérandes = 3

si oui : aller à l'étape 2

si non : aller à l'étape 3

étape 2 : tester si les 3 indicateurs associés aux opérandes sont à 1.
 si oui : mettre l'indicateur d'opérateur à 1; fin de l'analyse de la donnée calculée.
 si non : mettre l'indicateur d'opérateur à 0; fin de l'analyse de la donnée calculée.

étape 3 : mettre l'indicateur associé au 3iè opérande à 1, aller à l'étape 2.

105(R) : ajouter 1 à l'indice courant de la table des données calculées

107(ETAT) : mise à 1 du compteur de génériques.

109(générique) : placer le générique dans le descripteur de l'état sous la forme (1,y) où y est la place occupée par le générique dans le dictionnaire des données.

110(,) : tester si le compteur de génériques= 2
 si oui : passer à l'analyse du sous-total éventuel
 si non : continuer l'analyse d'état

111(SOUS-TOTAL) : mise à 0 du compteur de génériques et de N-ORDRE-GEN-PRECEDENT
 mise à 1 de l'indicateur SW.

114(générique) :

étape 1 :

pas 11 : ajouter 1 au compteur de génériques

étape 2 :

pas 21 : tester si le générique appartient au second groupe
 si oui : aller au pas 22
 si non : aller à l'étape 3

pas 22 : tester si le numéro d'ordre du générique est supérieur à N-ORDRE-GEN-PRECEDENT
 si oui : aller à l'étape 4
 si non : erreur dans la requête : l'ordre de spécification des génériques n'est pas respecté.

étape 3 :

pas 31 : SW vaut-il 1 ?
 si oui : mettre N-ORDRE-GEN-PRECEDENT à 0; ajouter 1 à SW
 si non : aller au pas 22.

étape 4 :

pas 41 : placer le générique dans le descripteur de SOUS-TOTAL sous la forme (1,y)

112 et 116(LISTE) : mise à 1 du compteur de données

120,123,126,129(nom-de-donnée) :

étape 1 : vérifier si cette donnée existe

si oui : aller à l'étape 2

si non : erreur ; fin de l'analyse de la requête

étape 2 : placer cette donnée dans la case correspondante sous forme (x,y) où

x = composante du tableau contenant les entrées du dictionnaire relatives à la requête.

y = position de la donnée dans l'entrée du dictionnaire.

étape 3 : ajouter 1 au compteur de données.

131(indice) : rechercher cette donnée dans la table des données calculées et la placer dans le descripteur sous la forme (o,z) où z est son numéro d'ordre dans la table.

133(.) :

étape 1 : tester si I-GEN-RS vaut 1.

si oui : à l'étape 2

si non : aller à l'étape 3

étape 2 : effectuer la génération de la description de l'état-imprimé.

étape 3 : effectuer l'accès aux données et construire les fichiers à éditer.

134(LOCALISATION) : mettre I-GEN-RS à 0

135(FIN) :

étape 1 : effectuer le tri des fichiers à éditer

étape 2 : réaliser l'édition

étape 3 : repositionner l'indice courant du tableau d'analyse à 1.

étape 4 : écraser les fichiers créés à savoir :

- fichier des lignes à éditer

- fichier des libellés

- fichier contenant la description de l'état imprimé

- fichier des descripteurs.

1. - algorithme des opérateurs ADD,SOUST,MULT,DIV:étape 1 :

pas 1.1 : positionner I (=compteur du nombre d'opérandes) à 1.

étape 2 :

pas 2.1 : acquérir la valeur de l'opérande

pas 2.2 : tester si I est égal au nombre maximum d'opérandes

si oui : aller au pas 2.3

si non : ajouter 1 à I; aller au pas 2.1

pas 2.3 : effectuer le calcul mentionné par l'opérateur et stocker le résultat dans la table des données calculées.

Algorithme de réalisation de l'opérateur PRCTétape 1 :

pas 1.1 : positionner I (= compteur du nombre d'opérandes) à 1 et ajouter 1 à K

pas 1.2 : K vaut-il 1 ?

si oui : aller au pas 1.3

si non : aller à l'étape 2

pas 1.3 : accéder au premier descripteur relatif à E4

- si ce descripteur existe : introduction de l'occurrence d'entité en mémoire ; aller au pas 1.4.

- si ce descripteur n'existe pas : aller à l'étape 2.

pas 1.4 : pour toutes les composantes de la table des données calculées dont le numéro de séquence est strictement inférieur à celui de l'opérateur PRCT, on effectue le calcul des opérateurs dont l'indicateur associé vaut 1 puis on positionne ce dernier à 2

étape 2 :

pas 2.1 : acquérir la valeur de l'opérande

pas 2.2 : tester si I vaut 2

si oui : aller au pas 2.3

si non : ajouter 1 à I; aller au pas 2.1

pas 2.3 : effectuer le calcul du pourcentage et stocker le résultat dans la table des données calculées

Algorithme de réalisation de l'opérateur CUMULétape 1 :

pas 1.1 : mise à 0 de la zone résultat de l'opérateur.

- pas 1.2 : acquérir le descripteur relatif à E4
- s'il existe : aller au pas 1.3
- s'il n'existe pas : aller à l'étape 2
- pas 1.3 : accéder à l'occurrence d'entité indiquée dans le descripteur
- pas 1.4 : pour toutes les composantes de la table des données calculées dont le numéro de séquence est strictement inférieur à celui de l'opérateur CUMUL, on effectue le calcul des opérateurs dont l'indicateur associé vaut 1
- pas 1.5 : dès que l'indice de la table = celui de la composante où se trouve l'opérateur CUMUL; on ajoute la valeur du second opérande dans la zone résultat; aller au pas 1:2
- étape 2 :
- pas 2.1 : pour toutes les composantes de la table des données calculées dont le numéro de séquence est supérieur ou égal à celui de l'opérateur CUMUL, mise de l'indicateur valant 1 à 2
- pas 2.2 : restaurer la structure hiérarchique associée à E1.

Algorithme de création des lignes à éditer.

étape 1 :

- pas 11 : ajouter 1 au compteur de classes de données (1 classe correspond à 1 triple localisation, calcul, édition).
- pas 12 : mettre le compteur d'appel à l'opérateur pourcentage à 0.
- pas 13 : mettre 351 dans CLE et acquérir le descripteur de l'association.

étape 2 :

- pas 2.1 : mettre 001 dans CLE-El
- pas 2.2 : accéder au descripteur désigné par CLE-El
- pas 2.3 : tester si le premier numéro d'entité du descripteur est égal à ***
 - si oui : aller au pas 23
 - si non : aller au pas 25
- pas 2.4 : on a terminé l'analyse de la classe de données. On repasse la main à l'analyseur qui effectue la génération des descripteurs relatifs à la classe suivante si elle existe, sinon on clôture.
- pas 2.5 : recherche de l'occurrence de El désirée.
- pas 2.6 : l'occurrence **existe-t-elle** ?
 - si oui : aller à l'étape 3
 - si non : erreur : l'occurrence recherchée ne figure pas dans la banque de données;
- pas 2.7 : si l'on est dans le cas d'un identifiant complet : ajouter 1 à CLE-El; aller au pas 2.2
 - si l'on est dans le cas d'un identifiant incomplet : plonger dans le module d'accès à l'adresse donnée par la table de mémorisation des étiquettes; aller au pas 2.5.

étape 3 :

- pas 3.1 : mettre 101 dans CLE-El-DONNEE
- pas 3.2 : accéder au descripteur désigné par CLE-El-DONNEE.
- pas 3.3 : tester si l'opérateur de combinaison est " * ".
 - si oui : aller à l'étape 4
 - si non : aller au pas 3.4
- pas 3.4 : transmission du descripteur au module de sélection

pas 3.5 : ajouter 1 à CLE-E1-DONNEE; aller au pas 32

étape 4 :

pas 41 : tester si CLE-E1-DONNEE
 si oui : aller à l'étape 5
 si non : aller au pas 42.

pas 42 : tester si TEST[1] = 1
 si oui : remettre I(indice du vecteur test) à 0; aller
 à l'étape 5
 si non : erreur: l'occurrence de E1 est refusée car elle
 ne vérifie pas les conditions imposées I à 0,

pas 43 : aller au pas 2.7

étape 5 :

pas 5.1 : tester si les deux numéros d'entité du descripteur de
 l'association valent : *****
 si oui : aller à l'étape 10
 si non : aller au pas 5.2.

pas 5.2 : tester si le second numéro d'entité du descripteur de
 l'association vaut 005.
 si oui : aller à l'étape 10
 si non : aller au pas 5.3.

pas 5.3 : transfert de l'identifiant de E1 dans la zone correspon-
 dante de l'ordre de recherche d'une occurrence de E3.

pas 5.4 : mettre 201 dans CLE-E2.

pas 5.5 : accéder au descripteur désigné par CLE-E2.

pas 5.6 : tester si le premier numéro d'entité du descripteur vaut ***
 si oui : aller au pas 5.7
 si non : aller au pas 5.8

pas 5.7 : tester si CLE-E2 = 201
 si oui : erreur : on a indiqué une association pour
 laquelle on n'a mentionné aucune occurrence de
 l'entité E2. On ne traite pas cette classe de
 données, on passe à la suivante. Retour à
 l'analyseur.
 si non : aller au pas 2.7.

pas 5.8 : recherche de l'occurrence de E2 désirée.

pas 5.9 : l'occurrence recherchée existe-t-elle ?

si oui : aller à l'étape 7.

si non : erreur : l'occurrence de E2 recherchée ne figure pas dans la banque de données.

pas 5.10 : si l'on est dans le cas d'un identifiant complet : ajouter 1 à CLE-E2; aller au pas 5.5.

si l'on est dans le cas d'un identifiant incomplet : plonger dans le module d'accès à l'adresse donnée par la table de mémorisation des étiquettes; aller au pas 5.8.

étape 6 :

pas 6.1 : mettre 301 dans CLE-E2-DONNEE

pas 6.2 : accéder au descripteur désigné par CLE-E2-DONNEE.

pas 6.3 : tester si l'opérateur de combinaison vaut " * "

si oui : aller à l'étape 7.

si non : transmission du descripteur au module de sélection.

pas 6.4 : ajouter 1 à CLE-E2-DONNEE, aller au pas 6.2

étape 7 :

pas 7.1 : tester si CLE-E2-DONNEE = 301

si oui : aller à l'étape 8

si non : aller au pas 7.2

pas 7.2 : tester si TEST(1) = 1

si oui : remettre I à 0; aller à l'étape 8.

si non : erreur : l'occurrence de E2 est refusée car elle ne vérifie pas les conditions imposées ; remettre I à 0; aller au pas 5.10

étape 8 :

pas 8.1 : tester s'il correspond une entité-relation à l'association

si oui : aller au pas 8.2

si non : aller à l'étape 10

pas 8.2 : tester si l'association possède des caractéristiques

si oui : aller à l'étape 9

si non : mettre " blanc " dans la variable TYPE-RELATION de l'ordre IRTISM

pas 8.3 : mettre le numéro de l'entité-relation dans la variable RT et transférer l'identifiant de E2 dans la variable ID2 de l'ordre IRTISM.

- pas_8.4 : recherche de l'occurrence de l'entité relation
 si elle existe : aller à l'étape 10
 si elle n'existe pas : aller au pas 8.5.
- pas_8.5 : tester s'il existe une autre entité-relation matérialisant
 l'association
 si oui : mettre son numéro dans RT; aller au pas 8.4
 si non : erreur : les occurrences d'entités E1 et E2 ne
 sont pas reliées par l'association mentionnée;
 aller au pas 5.10
- étape 9 :
- pas_9.1 : positionner un compteur de caractéristiques (COMPT-CARACT)
 à 1.
- pas_9.2 : mettre le numéro de l'entité relation dans la variable RT
 et transférer l'identifiant de E2 dans la variable ID2 de
 l'ordre IRTISM
- pas_9.3 : mettre V-CARACT (COMPT-CARACT) dans la variable TYPE-RELA-
 TION (V-CARACT est un tableau figurant dans le diction-
 naire des caractéristiques contenant les valeurs du para-
 mètre RELAT correspondant à la caractéristique).
- pas_9.4 : recherche de l'occurrence de l'entité relation
 si elle existe : aller à l'étape 10
 si elle n'existe pas : aller au pas 9.5
- pas_9.5 : tester si COMPT-CARACT = 4
 si oui : aller au pas 9.7
 si non : ajouter 1 à COMPT-CARACT.
- pas_9.6 : tester si V-CARACT(COMPT-CARACT) = 5
 si oui : aller au pas 9.7
 si non : aller au pas 9.3
- pas_9.7 : erreur car les occurrences des entités E1 et E2 ne sont pas
 reliées par l'association de caractéristiques mentionnées;
 aller au pas 5.10
- étape 10 :
- pas_10.1 : positionner l'indice courant de la table des données
 calculées à 1.
- pas_10.2 : tester si l'indice courant vaut 21
 si oui : aller à l'étape 11.

- pas 10.3 : tester si l'indicateur associé à l'opérateur est différent de 1 :
si oui : aller au pas 10.5
- pas 10.4 : tester si l'opérateur est CUMUL
si oui : faire appel à l'opérateur; aller au pas 10.6
si non : aller au pas 10.6
- pas 10.5 : tester si l'indicateur associé à l'opérateur vaut 2
si oui : aller au pas 10.6
si non : faire appel à l'opérateur; aller au pas 10.6
- pas 10.6 : ajouter 1 à l'indice courant; aller au pas 10.2
- étape 11 :
- pas 11.1 : tester si le second numéro d'entité du descripteur d'association vaut 005
si oui : aller à l'étape 14
- pas 11.2 : transférer dans la zone clé du fichier des lignes à éditer la valeur des génériques supérieurs ou égaux à ceux figurant dans ETAT.
- pas 11.3 : transférer la valeur du compteur de classe de données
- pas 11.4 : tester si on a une association avec caractéristiques
si oui : aller au pas 11.5
si non : aller au pas 11.6
- pas 11.5 : placer la valeur de la caractéristique dans la clé
- pas 11.6 : placer le reste de l'identifiant dans la clé
- pas 11.7 : tester si on a une association :
si oui : placer l'identifiant de E2 dans la clé
si non : laisser cette zone à blanc.
- étape 12 :
- pas 12.1 : positionner à l'indice courant du vecteur de stockage des données (qui sera aussi celui du descripteur de la ligne d'édition)
- pas 12.2 : tester si le numéro d'entité du couple associé à une donnée vaut 0.
si oui : aller au pas 12.5
- pas 12.3 : effectuer la recherche de la valeur de la donnée sur base des informations figurant dans le dictionnaire des données.

- pas 12.4 : tester le format de définition de la donnée
 si C2 : transférer la valeur dans la zone virgule flottante indiquée par l'indice courant; aller au pas 12.6
 sinon : transférer la valeur dans la zone alpha-numérique indiquée par l'indice courant; aller au pas 12.6
- pas 12.5 : transférer le résultat de l'opération dans la zone virgule flottante indiquée par l'indice courant
- pas 12.6 : ajouter 1 à l'indice courant et tester si le numéro d'entité du couple est **t**
 si oui : aller à l'étape 13
 si non : aller au pas 12.2

étape 13 :

- pas 13.1 : pour tous les génériques dont le numéro de séquence est supérieur à ETAT et pour ceux mentionnés dans ETAT, on transfère dans le tableau des libellés:
 - libellé complet du générique
 - libellé associé à la valeur du générique que l'on trouve soit dans une occurrence d'entité, soit dans une table de codification
 - identifiant complet jusqu'au générique considéré
- pas 13.2 : on fait de même pour tous les génériques, dont le numéro de séquence est inférieur à celui du premier générique d'ETAT, figurant dans la séquence de rupture.
- pas 13.3 : tester si les numéros d'entité du descripteur de l'association sont égaux à *********
 si oui : aller au pas 2.7
 si non : aller au pas 5.10

étape 14 : traitement d'une association bidon

remarque : dans une grande partie des entités de la banque de données figurent des données tabulaires représentant une valeur ventilée par type de fourniture primaire

Ex : dans l'entité section de production, on dispose du tableau TAB-DEPFIKE-DIR-FP = tableau des dépenses fixes en FP incorporées par les sections de production, par type de FP.

Lorsque l'on désire connaître la valeur de l'une ou plusieurs composantes de ce tableau, il est utile de faire précéder cette valeur du code et du libellé du type de la fourniture primaire. Pour éviter les accès à la banque de données en vue d'acquérir uniquement des libellés, on construit le tableau suivant.

code typfp.	composante dans l donnée tabulée	libellé typfp

Pour indiquer la composante désirée d'une donnée tabulaire, on utilise le code du type de fourniture primaire et la correspondance avec la composante désirée est réalisée grâce au tableau précédent. La requête sera formulée sous la forme d'une association dans laquelle l'entité E2 est TYPFP

Exemple :

```

DEBUT      LOCALISATION
           E1 : S-PR : US=8  DEPT=1  BLOC=20
           E2 : TYPFP : 10,20,30,40,50,60,70,80,90
           E3 : consommations-types-FP.
EDITION
           ETAT : SECT
           LISTE: E2-COTYPFP, E2-LIB, E1-DEPFIxE-DIR-FP.

```

FIN

- pas 14.1 : mettre 201 dans CLE-E2
- pas 14.2 : accéder au descripteur indiqué par CLE-E2
- pas 14.3 : tester si le premier numéro d'entité = ***
si oui : aller au pas 2.7

- pas 14.4 : stocker la valeur de la composante de la donnée tabulée correspondant au code typfp dans une zone de travail.
- pas 14.5 : effectuer les points 11.2 à 11.6
- pas 14.6 : transférer le code typfp considéré dans la clé
- pas 14.7 : positionner à l l'indice courant du vecteur de stockage des données détails
- pas 14.8 : tester si le numéro d'entité du couple associé à une donnée vaut 2.
si oui : aller prendre sa valeur dans le tableau précédent;
aller au pas 14.10
- pas 14.9 : effectuer les points 12.2 à 12.5
- pas 14.10 : ajouter 1 à l'indice courant et tester si le numéro d'entité du couple est différent de **t**
si oui : aller au pas 14.8
- pas 14.11 : effectuer les points 13.1 à 13.2
- pas 14.12 : ajouter 1 à CLE-E2; aller au pas 14.4

C O N C L U S I O N S .

L'objectif de ce travail était l'étude d'un langage d'édition d'une banque de données budgétaires. L'étude complète devait comporter les aspects suivants :

- la définition d'un langage qui doit être orienté utilisateur;
- l'étude de ses propriétés;
- son implémentation;
- la mise au point de cette implémentation;
- les tests du langage par rapport au premier objectif c'est-à-dire vérifier s'il est bien orienté utilisateur.

Nous avons réalisé :

- la définition des spécifications fonctionnelles du langage. Ce travail qui constitue une analyse de conception a consisté dans l'analyse d'un ensemble de questions relatives au contenu de la banque de données. Nous pouvons ainsi répondre à différents besoins de consultation de la banque de données ressentis par l'utilisateur allant de la consultation pure et simple de la valeur d'un élément précis, - au moyen d'une question ponctuelle -, à la recherche d'informations définies par leur relation avec d'autres éléments, - réalisable au moyen d'une question portant sur une association -. La forme externe permettant d'exprimer ces besoins est relativement simple d'utilisation car la rédaction d'une requête requiert uniquement la connaissance parfaite du contenu de la banque de données ainsi que les différents paramètres, avec leur ordre de citation, intervenant dans la requête;
- la mise en place d'un algorithme d'analyse fondé sur les propriétés théoriques du langage;
- la définition d'une stratégie d'implémentation qui nécessite les outils suivants.
 - Trois dictionnaires : celui des données renfermant toutes les informations nécessaires à la reconnaissance et à la manipulation des données; celui des associations contenant la liste des associations autorisées; celui des caractéristiques spécifiant les sous-ensembles que l'on peut dégager des informations relatives à une association.
 - Un interpréteur : qui fournit en sortie un ensemble de descripteurs destinés à la sélection des occurrences d'entités désirées, à la génération de la description de la structure du budget et à la constitution des lignes à éditer.

- Un module de sélection : dont la construction est basée sur les structures hiérarchiques dégagées du schéma de la banque de données. Lorsque la sélection est terminée, on construit le fichier des lignes à éditer ainsi que celui des libellés, qui seront ensuite triés avant d'être cédés au module d'édition.
- Un module de génération de la description du budget dont le résultat sera intégré dans le module d'édition.

Les éléments que nous venons de présenter sont réalisables mais il nous a été impossible de passer à la mise au point de l'implémentation ainsi qu'à son optimisation. Cependant ce travail nous a permis de couvrir un champ de travail très vaste qui va du besoin des utilisateurs jusqu'à l'implémentation et de nous rendre compte des difficultés inhérentes à chaque stade examiné.

Le travail devrait cependant être complété par la réalisation des tâches suivantes.

- Compléter les spécifications fonctionnelles par l'introduction d'opérateurs plus complexes permettant notamment le calcul de données statistiques en vue de la réalisation de traitements d'analyse et de prévision. On pourrait également définir une implémentation orientée vers un traitement interactif au lieu d'une exploitation en "batch" implicitement choisie dans ce travail. Cela permettrait d'alléger la spécification de la requête dont une partie pourrait être prise en charge par le système. Une telle redéfinition conduirait à revoir, au niveau de l'implémentation, le principe de création de fichiers intermédiaires.
- Améliorer l'implémentation par une optimisation des algorithmes présentés, tant au point de vue du temps de sélection que de la place mémoire. En particulier, le choix de fonctions de sélection plus rapides, une procédure de constitution des identifiants plus simple, l'amélioration de l'organisation générale de la procédure de création des fichiers pour l'édition.
- Définir un programme de formation des utilisateurs et une documentation à leur usage.
- Tester auprès des utilisateurs les propriétés terminales du langage (usage direct du langage sans passer par l'intermédiaire d'un programmeur) et examiner les difficultés d'assimilation de la syntaxe du langage. Ceci devrait permettre de dégager un ensemble de conclusions

relatives à l'amélioration du langage. Nous sommes conscients qu'il n'est pas économique de procéder à ce test après l'implémentation. On ne l'a pas fait par manque de temps et parce que le propos du mémoire était de couvrir les différents aspects de la démarche.

Bibliographie.

- [1] Module I.D.S - R.Patigny (Institut d'Informatique - Juin 1975)
- [2] Simulateur de gestion pour entreprises manufacturières - Institut d'Informatique. Facultés Notre-Dame de la Paix - Namur (Novembre 1971)
- [3] Description du contenu de la banque de données - A-M Hennebert et J-M Leheureux (Namur - Juillet 1975)
- [4] Analyse fonctionnelle de la génération des équations - J-M Leheureux (Namur - Mars 1975)
- [5] Proposition d'un graphe d'activité pour la fabrication, à l'usine de Moustier et la commercialisation du float - Equipe graphe : M.Guillaume - G.Chainiaux. R.Lemaire (Avril 1975)
- [6] Clôture des travaux relatifs à la définition du graphe d'activités de production - L.B Raway (Juin 1975)
- [7] Projet SCAPFACE. Acquisition des structures d'imprimés. Génération des programmes d'édition. (Thèse de 3ème cycle - Toulouse 1974)
- [8] SEQUEL : A structured english query language - Donald D.Chamberlin et Raymond F.Boyce (Mai 1974)
- [9] NUL : A **n**avigational user's language for a network structured Data Base - Claude Dehennette et Henri Hennebert (Institut d'Informatique 21, rue Grand gagnage - B5000 NAMUR)
- [10] Cours de COBOL - André Clarinval (Institut d'Informatique - année académique 1974-1975)
- [11] Cours de théorie des langages : théorie de la sémantique et grammaires formelles - H.Leroy (Institut d'Informatique - année académique 1974-1975).