78|736

Sc N 78/22A

Université de
Nancy I

U.E.R.
Sciences Mathématiques

CONTRIBUTION À L'ANALYSE DES STRUCTURES EXTERNES ET INTERNES ET DE LEUR CORRESPONDANCE DANS UN S.G.B.D. (SYSTÈME PIVOINES)

oooooooo

THESE

UNIVERSITÉ NANCY I BIBLIOTHÈQUE

Pour l'obtention
du Doctorat de Spécialité
Informatique

Soutenue le 12 mai 1978 par :
Jacques LONCHAMP

JURY : M. C. PAIR — Président
M. A. CABANES — Examinateur
Mme M. CREHANGE — Examinateur
M. JC. DERNIAME — Examinateur
Mme C. ROLLAND — Examinateur

Publication du CENTRE de RECHERCHE en INFORMATIQUE de NANCY (C.R.I.N.)
(Référence : CRIN 78 - T - 020)

BIBLIOTHEQUE SCIENCES NANCY 1
D 095 180521 1

Université de
Nancy I

U.E.R.
Sciences Mathématiques

CONTRIBUTION À L'ANALYSE DES STRUCTURES EXTERNES ET INTERNES ET DE LEUR CORRESPONDANCE DANS UN S.G.B.D. (SYSTÈME PIVOINES)

oooooooo

THESE

Pour l'obtention
du Doctorat de Spécialité
Informatique

Soutenue le 12 mai 1978 par :
Jacques LONCHAMP

JURY : M. C. PAIR — Président
M. A. CABANES — Examinateur
Mme M. CREHANGE — Examinateur
M. JC. DERNIAME — Examinateur
Mme C. ROLLAND — Examinateur

Publication du CENTRE de RECHERCHE en INFORMATIQUE de NANCY (C.R.I.N.)
(Référence : CRIN 78 - T - 020)

Ce travail s'inscrit dans le cadre d'une convention

avec le Ministère de l'Industrie.

(Contrat B.N.I.S.T. n° 291-770).

Je remercie vivement Monsieur le Professeur C. PAIR, Président de l'Institut National Polytechnique de Lorraine, pour l'honneur qu'il me fait de présider le jury.

Je remercie tout particulièrement Madame Marion CREHANGE, responsable du projet PIVOINES, qui a su, avec gentillesse, attention et compétence, me diriger tout au long de l'élaboration de cette thèse.

Que Madame C. ROLAND, Messieurs A. CABANES et J.C. DERNIAME, soient sincèrement remerciés pour l'intérêt qu'ils ont porté à ce travail et l'honneur qu'ils me font de participer au jury.

Merci à J.F. BUECHER pour son aide précieuse et sympathique dans la mise en oeuvre d'une application et à Madame D. MARCHAND pour la réalisation matérielle de la thèse.

Enfin je tiens à exprimer ma profonde reconnaissance à Monsieur Y. GOUSTY, responsable du Département de Gestion et d'Economie Industrielle de l'Université de Metz, qui, en m'accueillant dans son département, a créé des conditions excellentes pour la poursuite de ce travail de recherche.

° ° ° ° °

«Loi de Hoare des grands programmes : A l'intérieur de chaque grand logiciel il existe un petit logiciel qui ne demande qu'à sortir.»

JOHN PEERS , humoriste
(Les lois de l'informatique)

SOMMAIRE GENERAL

CHAPITRE 0 : Introduction

CHAPITRE I : L'utilisation du modèle de PIVOINES au niveau externe

CHAPITRE 2 : L'utilisation du modèle de PIVOINES au niveau interne

CHAPITRE 3 : Résolution du problème de la"correspondance élargie"

CHAPITRE 4 : Un exemple de mise en oeuvre du système

CHAPITRE 5 : Conclusion

oooooooo

CHAPITRE 0 :

INTRODUCTION

0.I. L'objectif principal ..... p. I

0.2. Le cadre - le système PIVOINES ..... p. 3

0.2.I. Le système existant

0.2.I.I. D'un point de vue "fonctionnel"

0.2.I.2. D'un point de vue "organique"

0.2.2. Le contexte de cette étude

0.2.2.I. Confrontation à l'architecture de l' ANSI

0.2.2.2. Le cadre retenu

0.3. Les grandes lignes de l'étude ..... p. I5

0.3.I. Généralités

0.3.2. Le problème central : la différenciation des classes de repères logiques des classes de repères de PHYLOG

0.3.3. Le plan du travail

BIBLIOGRAPHIE ..... p. 2I

o o o o

0.I. L'objectif principal :

Cette thèse aborde un certain nombre de problèmes relatifs à la correspondance ("mapping") entre structures externes et structure interne d'un système de gestion de base de données (S.G.B.D.). Elle a pour cadre le système PIVOINES ([0.I]).

Le mot structure est ici utilisé comme traduction des termes anglais "schema" ou "model" : une structure est la description, indépendante des valeurs présentes à un instant donné, d'un ensemble d'informations à l'aide d'un modèle de données ou "data model" (modèles hiérarchiques, en réseau, relationnel de Codd etc.... cf. [0.2]).

Les termes de structure interne et de structure externe sont employés dans le sens défini par l' ANSI ([0.3] p.I-5) : « les informations comme elles sont vues par la machine et les informations comme elles sont vues par le programmeur ».

Ces termes sont identiques aux termes physique et logique traditionnellement utilisés dans le contexte des fichiers classiques.

Par définition, une base de données est un objet communautaire, c'est à dire partagé entre plusieurs utilisateurs(cf. ente autres la définition de C.J.DATE [0.2] p.I : « la base de données est intégrée. Cela signifie que la base contient des informations pour plusieurs utilisateurs et non juste pour un (...) »).

Il est couramment admis, et le rapport de l' ANSI/SPARC le reflète, que des catégories distinctes d'utilisateurs doivent pouvoir interagir avec la base selon des "visions" qui leur soient propres et qui se concrétisent dans des structures externes propres ([0.3] p.III-7). On parle souvent à ce sujet de "logical data independence" ([0.4,0.5...]). On peut déjà noter avec PIROTTE ([0.4] p.5) que les différenciations admises pour ces visions ne sont pas encore claires.

Il est évident que cette exigence pose avec acuité deux catégories de questions fondamentales concernant :

- la conception des structures : comment concevoir les structures externes, la structure interne ? comment s'assurer de la cohérence sémantique des

structures ? etc.....

+ l'architecture des systèmes : comment y intégrer les structures externes et interne ? selon quels objectifs ? comment réaliser les correspondances? etc.....

L'objectif principal de ce travail est de confronter le système PIVOINES (et plus précisément son sous-système "interrogation" existant, qui sera décrit au paragraphe suivant) avec le problème de la multiplicité des visions propres d'utilisateurs, en déterminant une position globale cohérente face aux problèmes d'architecture et de conception, mais en repectant aussi le parti-pris de mise en œuvre effective qui caractérise tout le projet PIVOINES : l'approfondissement de certains problèmes sera donc différé quand ils ne mettent pas en cause directement la mise en œuvre (ici une grande partie des problèmes de conception) et des hypothèses restrictives seront posées à condition qu'elles s'inscrivent dans un projet global cohérent et bien défini (ici l'étude sera une nouvelle fois restreinte à la partie interrogation du projet PIVOINES).
L'objectif principal est donc la mise en oeuvre opérationnelle du sous-système interrogation "multi-visions".

- - - - - -

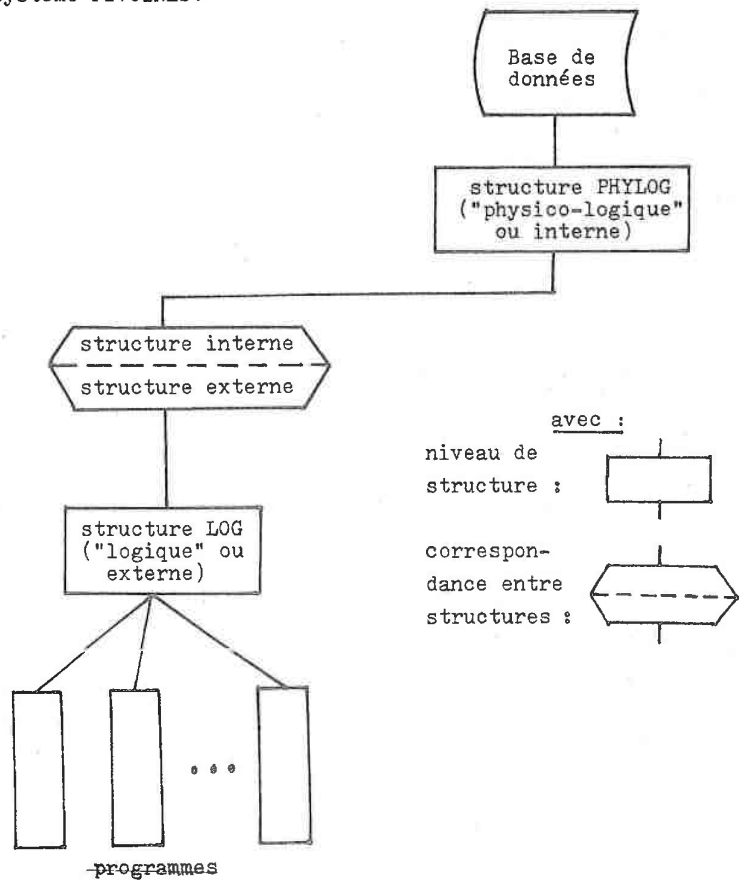
0.2. Le cadre - le système PIVOINES :

0.2.I. Le système existant :

0.2.I.I. D'un point de vue "fonctionnel" :

Il est indispensable de rappeler tout d'abord en quelques mots l'organisation, d'un point de vue "fonctionnel", et la terminologie du système PIVOINES existant, en adoptant par exemple la représentation schématique proposée dans [0.6] :

fig. 0-I
L'organisation "fonctionnelle"
du système PIVOINES.



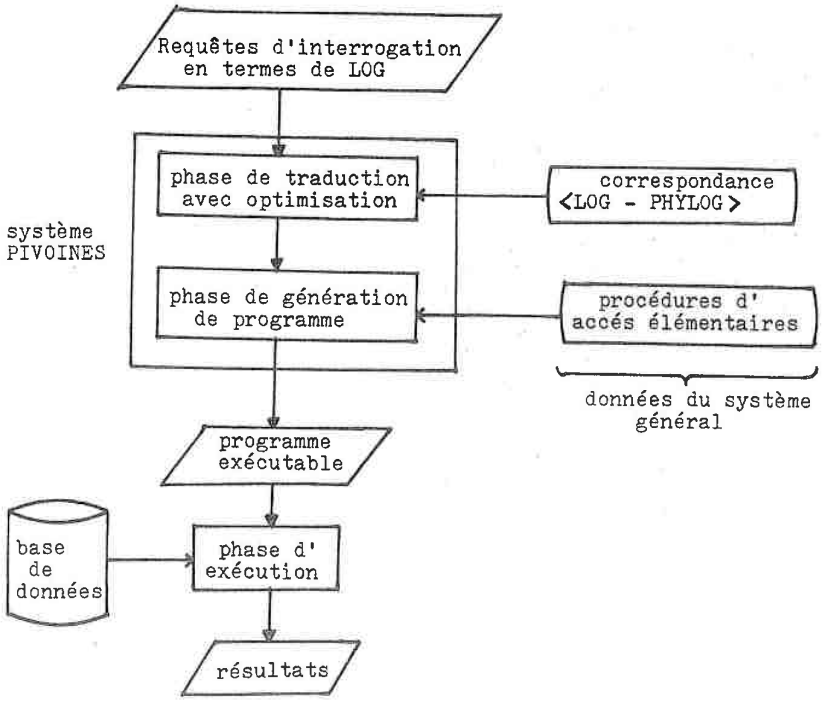
Les deux structures LOG et PHYLOG sont formalisées selon un même modèle de données très général (modèle introduit dans [0.7], présenté en détail dans [0.I] et que nous analyserons dans les chapitres I et 2).
L'organisation schématisée dans la figure 0.I a pour but essentiel d'assurer un bon degré de "physical data independence" ([0.4,0.5,...]) : il s'agit de l'indépendance des programmes par rapport aux choix d'implémentation des données; bon nombre des modifications à ce dernier niveau doivent se répercuter uniquement sur la structure PHYLOG et le mapping <LOG-PHYLOG>, sans affecter la structure LOG et donc les programmes écrits en termes de la structure LOG.

°°°

0.2.I.2. D'un point de vue "organique" :
D'un point de vue plus "organique", la correspondance <LOG-PHYLOG> est une donnée du système général qu'est PIVOINES.
A partir des requêtes d'interrogation (*) exprimées selon un langage déclaratif en termes de LOG, le système construit, grâce à la correspondance <LOG-PHYLOG>, une stratégie d'accès optimisée dans le contexte de PHYLOG. La concrétisation de cette stratégie d'accès en un programme exécutable se réalise dans une seconde phase, grâce à des procédures élémentaires liées aux accès de PHYLOG; ces procédures élémentaires constituent d'autres données du système général (nous détaillerons ces mécanismes aux paragraphes 2.I et 2.2).

(*) seul le sous-système "interrogation" de PIVOINES a été étudié dans le détail et a été réalisé.

fig. 0-2
Schéma "organique"
du système PIVOINES.



0.2.2. Le contexte de cette étude :

0.2.2.I. Confrontation à l'architecture de l'ANSI :
L'organisation fonctionnelle qui intègre plusieurs structures externes et qui dérive immédiatement de l'organisation actuelle (fig. 0.I.) est schématisée dans la figure 0.3.
Elle est à rapprocher de l'organisation à trois niveaux retenue par l'ANSI, pour discussion(cf. fig. 0.4 tirée de [0.6] p.30).
La différence fondamentale est l'existence dans l'organisation de l'ANSI d'une structure conceptuelle commune.

fig. 0-3
Organisation "fonctionnelle"
immédiate à plusieurs structures
externes

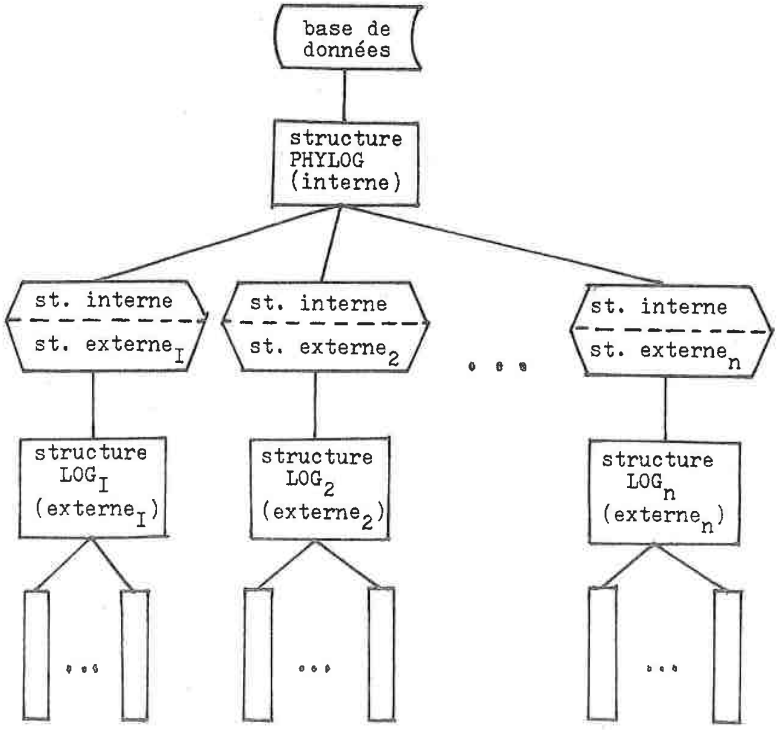
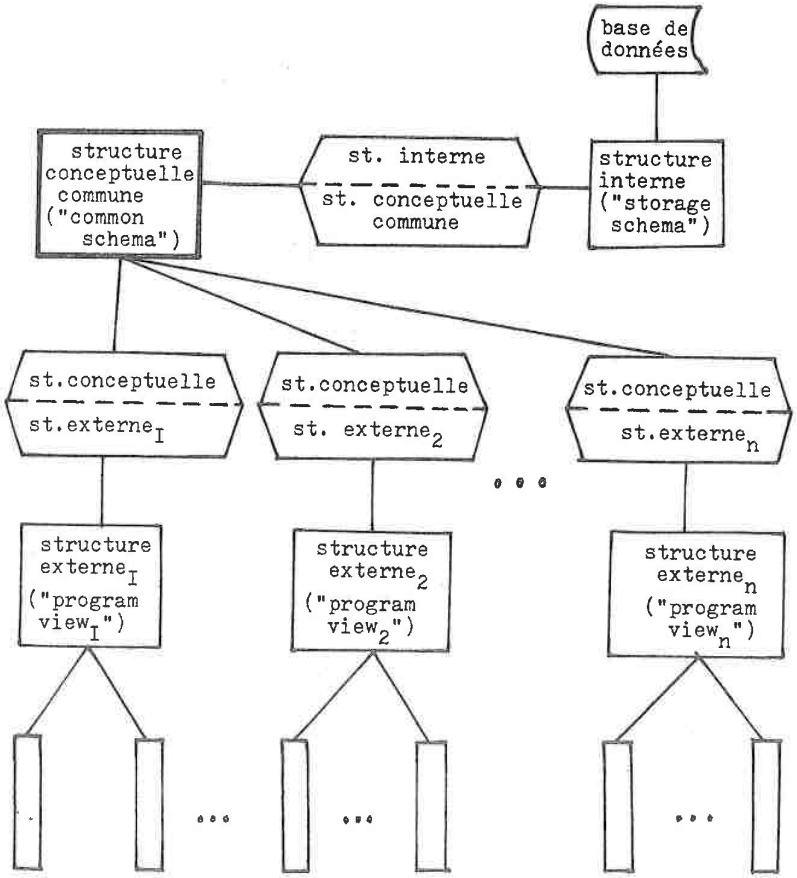


fig. 0-4
Organisation à
3 niveaux (ANSI)



Pour l'ANSI ([0.3] p.I-5) « le schéma conceptuel représente la vue qu'a l'entreprise de la structure qu'elle tente de modéliser dans la base de données ».En d'autres termes il s'agit de la vue logique commune de la base.Toutes les définitions qui vont dans ce sens ne sont accompagnées que de peu d'indications sur le processus d'obtention de la structure conceptuelle et sur ses liens avec les différentes visions des utilisateurs.On peut citer à ce propos les mots de "superposition" [0.8] et d' "intégration" [0.9] : « une description fidèle d'une portion du monde réel pourrait donc être immaginée comme une sorte de superposition de toutes les perceptions particulières possibles de cette portion du réel » ([0.8] p.4) et « nous admettons que le monde réel est perçu selon des"structures logiques" spécifiques à chaque observateur. La structure conceptuelle intègre toutes ces"structures logiques" dans une description unique qui supprime toute redondance sémantique (*) » ([0.9] p.I9).

L'interêt de la structure conceptuelle peut être analysé sous deux aspects bien distincts :

+ lié à la définition initiale des structures, lors de la conception d'un système (ou lors des éventuelles restructurations)
+ lié au fonctionnement des systèmes, comme élément de leur architecture.

a) la structure conceptuelle, résultat d'une étape du processus de conception des SGBD :

Le processus de conception des SGBD est souvent décrit en trois étapes ([0.10,0.11]),ce qu'illustre la figure 0.5 .

(*) il y a redondance sémantique lorsqu'un même objet est pris en compte de plusieurs manières différentes, sous des désignations différentes dans les diverses"structures logiques" .

fig. 0.5
Les étapes de la conception
d'un SGBD.

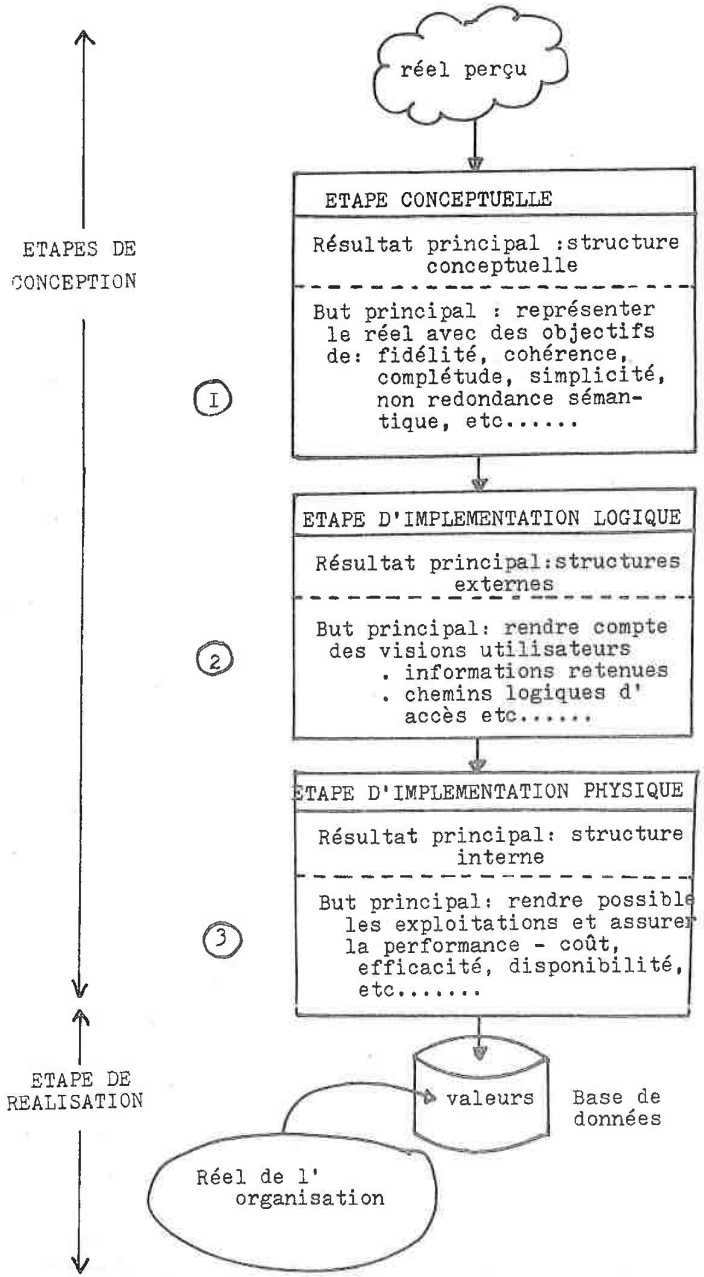
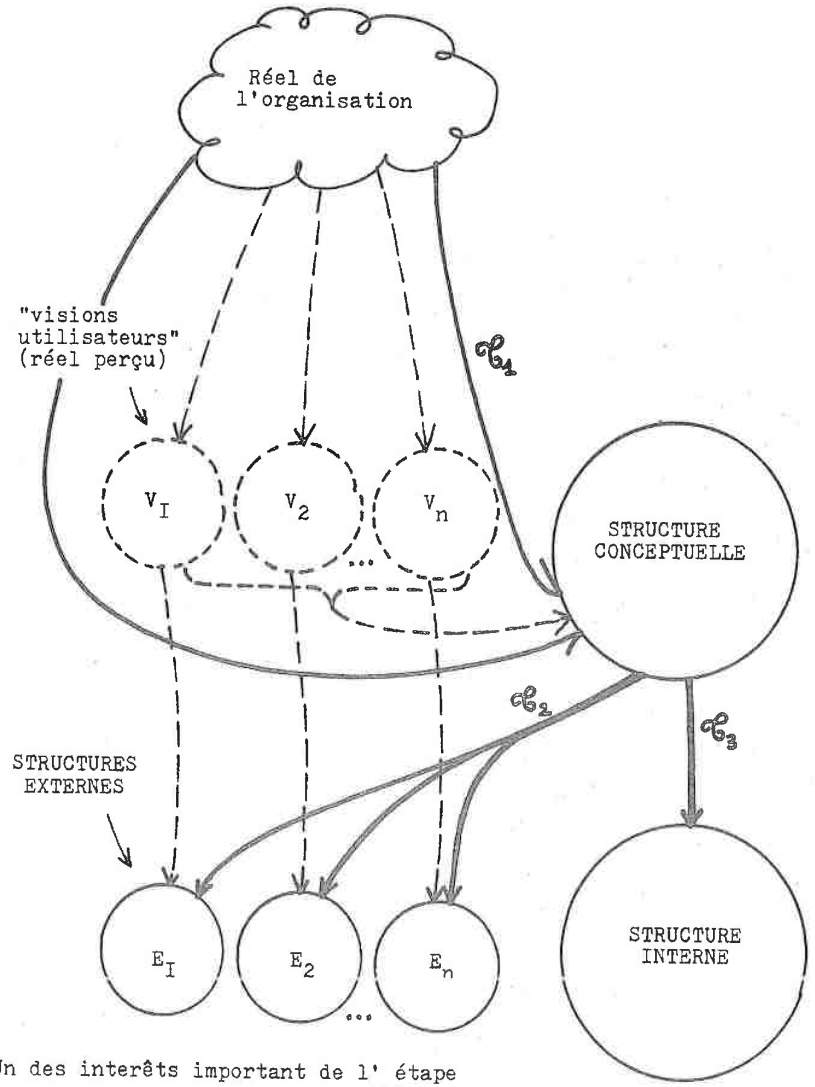


fig. 0-6
Les transformations dans
le processus de conception
d'un S.G.B.D.



Un des interêts important de l' étape conceptuelle est qu'elle permet de s'assurer de la cohérence sémantique des structures externes et de la structure interne vis à vis de la structure conceptuelle et donc de la cohérence des structures externes vis à vis de la structure interne. Ceci, en

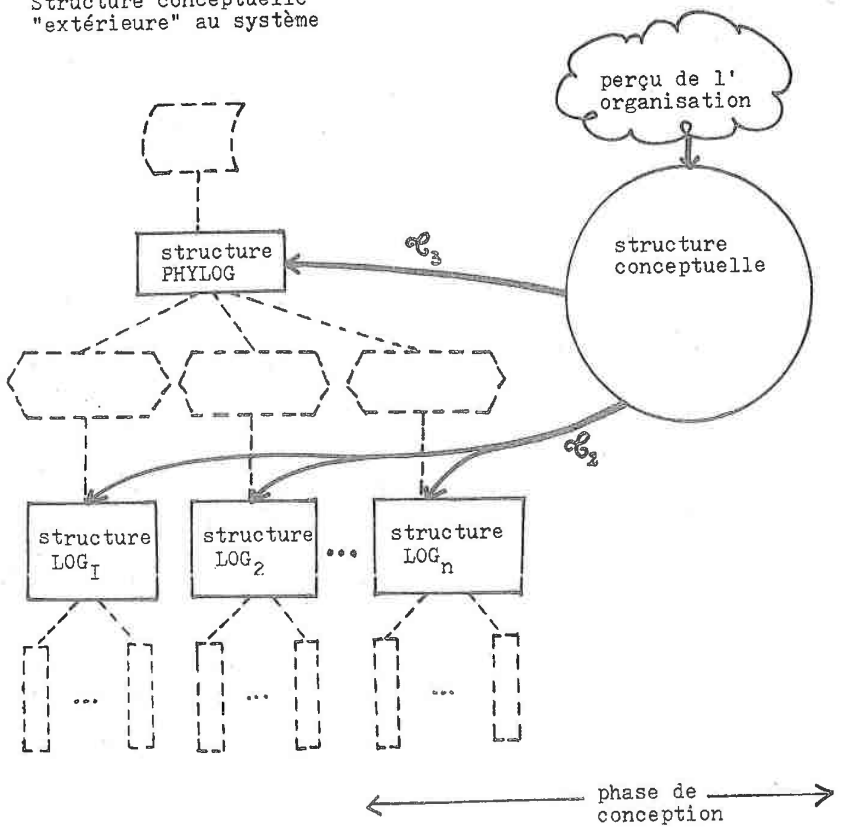
contrôlant les transformations appliquées à la structure conceptuelle lors de la conception des structures externes ($\mathcal{C}_2$) et de la structure interne ($\mathcal{C}_3$) (fig. 0.6).

C'est ainsi que doit être vérifiée en particulier l' "équivalence sémantique" entre la structure conceptuelle et la structure interne.

Si l'on revient à la figure 0.3, il faut noter tout d'abord que la structure PHYLOG est bien la structure interne (elle reflète la répartition des données dans l'espace de mémorisation avec les accès existants - cf. chapitre 2) et non pas la structure logique commune.

Par contre, cette structure logique commune peut parfaitement avoir été conçue préalablement et les autres structures en dériver, ce qui doit permettre d'assurer en particulier la cohérence des structures

fig. 0-7
Structure conceptuelle "extérieure" au système



b) la structure conceptuelle, niveau supplémentaire dans l'architecture des SGBD :

La propriété majeure de ce niveau est la stabilité qui « est la caractéristique de n'être pas touché par les variations aux niveaux interne ou externe » ([0.I2] p.50). Cette stabilité le désigne particulièrement pour acceuillir la centralisation de contrôles souvent indépendants de ces variations et rendus indispensables par l'intégration et le partage des données. « Le contrôle centralisé de la sécurité et de l'intégrité doit être assuré dans un module à travers lequel tout le "traffic" passe. En principe il y a deux possibilités : la structure interne et la structure conceptuelle (...) il est clair que l'endroit approprié n'est pas la structure interne, car la sécurité et l'intégrité seront alors (dans le cas de la structure conceptuelle) exprimés d'une manière indépendante de la "représentation" et cela procure plus de stabilité. La structure conceptuelle est le bon endroit pour décrire les aspects qui sont valides pour la communauté des utilisateurs » ([0.6]p.37).

Par "sécurité", le rapport ANSI/SPARC ([0.3]) entend la protection de l'information contre la destruction, la modification, la recherche délibérée par un individu qui n'en a pas les droits.

Par "intégrité", le même rapport entend la protection de l'information contre la destruction ou la modification accidentelle par un individu ou par tout évènement imprévisible (ce qui recouvre en particulier le contrôle de la nature et du type de l'information cf.[0.I3] parag.I.2) .

°°°°

0.2.2.2. Le cadre retenu :

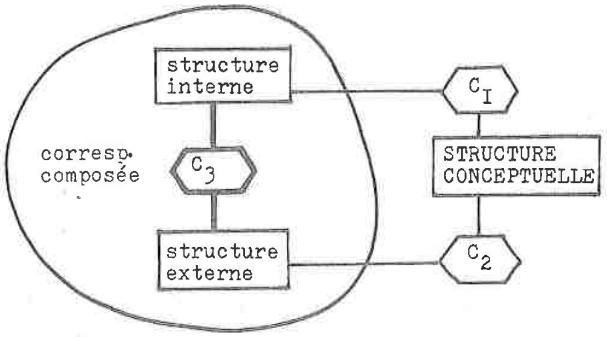
Dans notre cas, il faut noter que la réalisation du système PIVOINES est planifiée selon trois sous-systèmes fonctionnels :

+ d'interrogation
+ d'acquisition
+ de modification

Le sous-système d'interrogation, qui a été seul étudié jusqu'à présent, n'est spécifiquement concerné que par le problème des accès délibérés à des zones interdites pour certains utilisateurs. Il nous semble que les contrôles correspondants (identité de l' utilisateur et prise en compte de ses droits d'accès) peuvent être réalisés de façon satisfaisante aux seuls niveaux externe et

interne.
Il en est de même pour les problèmes généraux d'intégrité, c'est à dire de protection contre les altérations accidentelles des informations.
On peut donc affirmer que les justifications "dynamiques" (c'est à dire liées au fonctionnement) à l'existence d'un niveau conceptuel entre les niveaux externe et interne ne sont pas décisives en ce qui concerne un sous-système interrogation. On peut alors éventuellement considérer que les correspondances de la fig. 0.3 sont des correspondances"composées" , destinées à la seule interrogation et s'inscrivant dans une architecture plus complexe à trois niveaux . (sans préjuger des résultats des études en cours sur les autres sous-systèmes, il semble cependant a priori plus difficile d'assurer les fonctions correspondantes complètement - modification et acquisition - sans la présence d'un niveau conceptuel commun, cf. [0.6] p.37).

fig. 0-8
Principe des correspondances "composées"



Il faut noter que le rapport ANSI/SPARC prévoit explicitement la possibilité de telles correspondances composées ("composite mapping") : « il serait possible de réaliser des correspondances directes entre les structures externes et interne pour gagner l'

efficacité d' avoir un niveau d'indirection en moins lorsque l' on traite des requêtes externes, en acceptant la perte de degrés supplémentaires de "data independence" que peut procurer la structure conceptuelle » ([0.3] p.III-7) .
En effet, il faut noter que si un bon degré de "physical data independence" peut être assuré par une organisation du type de celle présentée à la figure 0.3, c'est au prix de la modification de la structure interne et des n mappings ⟨structure interne-structure externe⟩ alors que dans l'organisation à trois niveaux de la figure 0.4 un seul mapping ⟨structure interne-structure externe⟩ est mis en cause.

Nous pouvons donc maintenant préciser le cadre de notre étude : nous considèrerons n structures externes et une structure interne cohérentes entre elles (le processus de conception, non décrit, l'assurant); nous étudierons les problèmes de correspondance dans une optique purement interrogation qui justifie l'organisation "fonctionnelle" retenue, illustrée par la figure 0.3 .Nous espérons de cette manière conserver presqu'intégralement les processus internes du système déjà réalisé, ce qui nous donnera du point de vue "organique" le système décrit par la figure 0.9 :

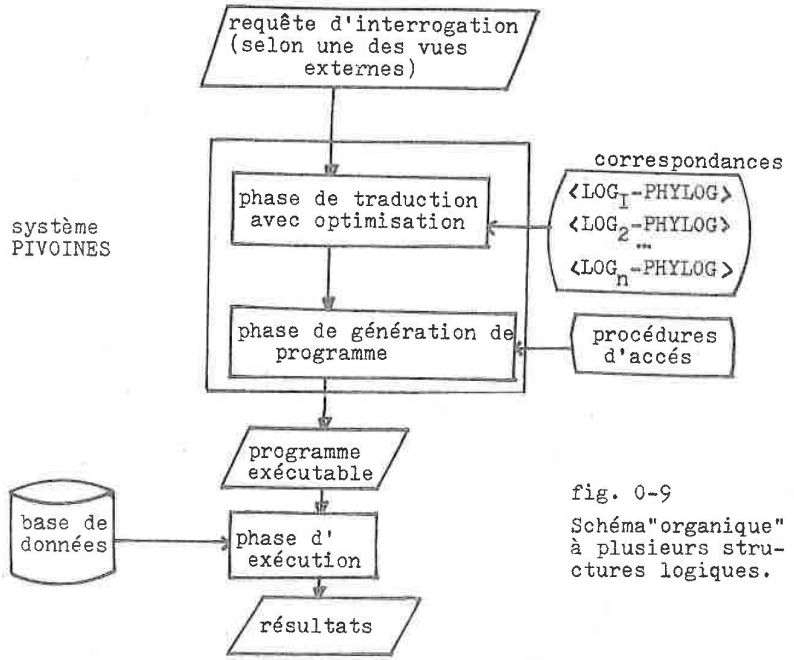


fig. 0-9
Schéma"organique" à plusieurs structures logiques.

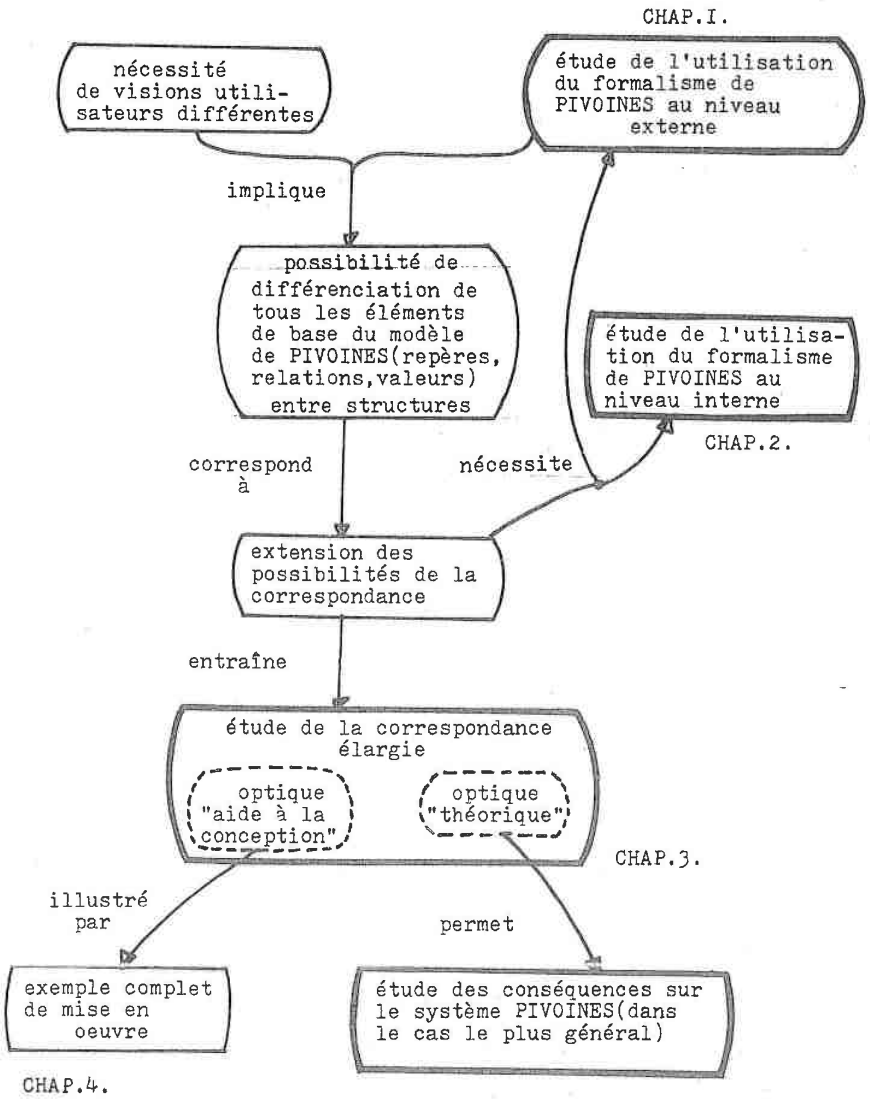
0.3. Les grandes lignes de l'étude :

0.3.I. Généralités :

L' étude de l'utilisation du formalisme de PIVOINES (à base de repères, de valeurs et de relations) au niveau externe (cf. chapitre I), nous indique que les multiples visions des utilisateurs se concrétiseront essentiellement par l'existence de classes de repères (*) et de classes de relations (*) différentes d'une structure externe à l'autre et donc aussi entre les structures externes et la structure interne.
Par ailleurs la nature même de cette structure interne doit être précisée: en particulier sa"place"par rapport à l' implémentation, en vue d'assurer une efficacité maximum (cf. chapitre 2).
La possibilité d'avoir des classes de repères au niveau extern différentes des classes de repères du niveau interne constitue une extension importante aportée au système actuel(qui n' admettait que la différenciation des classes de relations).
L'essentiel de ce travail est consacré à la résolution de ce problème (que nous désignerons par "problème de la correspondance élargie").Lui apporter une solution est indispensable pour mettre en oeuvre un sous-système d'interrogation "multi-visions" , ce qui constitue notre objectif fondamental.
L'étude se fera dans une optique "théorique", afin de déterminer la forme la plus générale de correspondance et ses conséquences sur le système et dans une optique "aide à la conception", afin de fournir un outil pratique permettant de faciliter la mise en oeuvre de chaque cas précis.
La figure 0.I0 schématise cette articulation des éléments de notre travail avec leurs liens logiques.

(*) on parlera de repère et de relation au niveau des occurences et de classe de repères ou de classe de relations au niveau des ensembles d'occurrences.

fig. 0.I0
Articulations logiques du travail.



Peu de travaux abordent cette catégorie de problèmes relatifs à la réalisation des correspondances entre structures. Quelques études concernent la réalisation de mappings entre structures de "type relationnel" et structures de "type accès" (cf. le "translator/interpreter" de NEUHOLD dans[0.I4]). D'autres études se limitent à des descriptions formelles de ces problèmes de correspondance [0.5,0.I5,0.I6 ...]. Des questions proches sont d'actualité par le biais des bases de données réparties où l'on distingue une vue globale et des vues locales : il s'agit de transformer des requêtes exprimées selon la vue globale en requêtes exécutables au niveau local, les modèles de structure pouvant être différents (cf. le "cooperation system translator" de[0.I7]).

°°°°

0.3.2. Le problème central : la différenciation des classes de repères logiques des classes de repères de PHYLOG :

Dans le système actuel la structure logique ne pouvait différer de la structure PHYLOG que par le "réseau" des classes de relations, la "charpente" des classes de repères restant la même. Tout repère logique avait un correspondant direct dans PHYLOG, l'inverse n'étant pas obligatoire. La correspondance se caractérisait donc par :

fig.0.II(a)

LOG	PHYLOG
classe de repères →	classe de repères
classe de relations (ou groupement traduisible cf.[0.I]) →	classe de relations ou composition de celles-ci

Dans la présente étude la structure logique peut différer de PHYLOG à la fois par ses classes de repères et par ses classes de relations (les classes de valeurs se retrouvent aux deux niveaux à des différences de "présentation" près). Il existe

donc des classes de repères sans équivalent direct dans PHYLOG. Afin de conserver le même processus général de traduction, la "correspondance élargie" sera caractérisée par :

fig.0.II(b)

LOG	PHYLOG
classe de repères →	classe de repères ou d' "objets équivalents"(0.E.)
classe de relations (ou groupement traduisible) →	classe de relations ou composition de celles-ci (avec gestion des 0.E.)

Un"objet équivalent" est un objet à base de repères, de relations et de valeurs de PHYLOG qui est associé univoquement à chaque repère logique d'une classe de repères sans équivalent direct dans PHYLOG.

Cette méthode nous permet de continuer à faire réaliser par le système une véritable "simulation" des accès logiques dans le contexte de PHYLOG.

La difficulté réside dans la détermination de ce que peut être l' 0.E. d'un repère logique quelconque.

Il nous est apparu pratique, pour ce faire, de changer de formalisme afin qu'un mapping LOG-PHYLOG puisse être exprimé à l' aide d'un petit nombre d'opérateurs élémentaires, donc en utilisant un langage simple de description de la correspondance. De tels opérateurs sont difficiles à concevoir dans le cadre de PIVOINES alors qu'ils existent dans le cadre relationnel. Nous avons donc utilisé le formalisme relationnel de Codd [0.I8] selon des règles précises qui nous permettent de conserver clairement présente la notion de repère dans l' "expression relationnelle" d'une structure PIVOINES. Nous montrons qu'il est alors possible d'exprimer d'une manière simple, à l'aide d'une algèbre relationnelle, tout mapping entre LOG et PHYLOG ou plus exactement entre l' "expression relationnelle" de LOG (ERLOG) et l' "expression relationnelle" de PHYLOG (ERPHYLOG). Dans cette formulation de la correspondance, à chaque classe de repères logiques est

associée une expression relationnelle. Comme nous sommes capables d'interpréter en termes d"objets équivalents" l'effet de chaque opération de l'algèbre et de leur composition, nous pouvons déterminer la forme des objets équivalents dans chaque cas particulier ainsi que la forme la plus générale des O.E.

fig. 0-II (a)
Changement de formalisme.

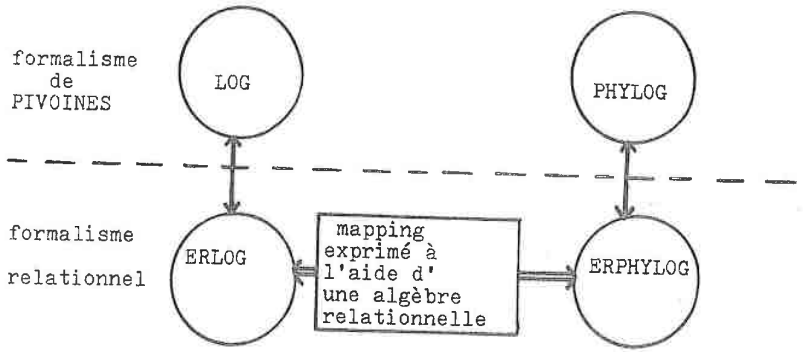
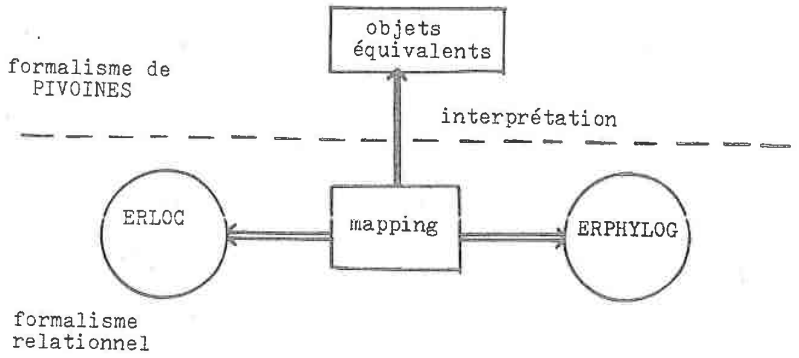


fig. 0 - II (b)
Détermination des objets équivalents.



C'est à partir de cette forme générale que nous étudierons les

conséquences sur le système PIVOINES. Il faut bien souligner que ce passage par le modèle relationnel est réalisé à des seules fins de "démonstration" et n'a aucune "matérialisation" dans le système.

°°°°

0.3.3. Le plan du travail :

Notre travail comporte quatre parties qui regroupent respectivement :

+ tout ce qui concerne le niveau externe (étude de l' utilisation du formalisme de PIVOINES au niveau externe, définition de ERLOG)
+ tout ce qui concerne le niveau interne (étude de l' utilisation du formalisme de PIVOINES au niveau interne, définition de ERPHYLOG)
+ tout ce qui concerne l'étude "théorique" de la "correspondance élargie"
+ tous les aspects d'une mise en oeuvre effective.

- - - - -

BIBLIOGRAPHIE :

[0.I] Description formelle, représentation, interrogation des informations complexes-système PIVOINES
M.CREHANGE
Thèse d'Etat (Nancy I)
(Decembre I975)

[0.2] An introduction to database systems
C.J.DATE
Addison-Wesley
(I975)

[0.3] ANSI/X3/SPARC Interim Report
Study group on DBMS
(I975)

[0.4] Explicit description of entities and their manipulation in languages for the relational data base model
A.PIROTTE
M.B.L.E. Bruxelles (Report R336)
(Septembre I976)

[0.5] Multiple view support within the ANSI/SPARC framework
A.KLUG,D.TSICHRITZIS
Proceedings Very Large Database
Tokyo
(Octobre I977)

[0.6] A gross architecture for the next generation DBMS
G.M.NIJSSEN
IFIP-TC2 Working conference on modelling in DBMS (Freudenstadt)
(Janvier I976)

[0.7] Cours sur les structures d'information
C.PAIR
(I97I)

[0.8] The EPA model.An information-oriented database model
A.PIROTTE
M.B.L.E. Bruxelles (Report R343)
(Mars I977)

[0.9] Data structure models for information systems (Introductory report of the international workshop - Namur, Mai I974)
BENCI,BODART,CABANES,DEHENEFFE,HAINAUT,LEROY,RANDON,SAVOYSKY, THULY
Institut d'informatique de Namur
(I975)

[0.I0] La conception des systèmes de données en vue de la réalisation d'une base de données
G.E.BENCI
Convention Informatique
(I976)

[0.II] La conception des bases de données et les modèles de données
G.BENCI,C.ROLLAND
Polycopié (Nancy-IRIA)

[0.I2] Binary logical associations in data modelling
G.BRACCHI,P.PAOLINI,G.PELAGATTI
(Freudenstadt)
(Janvier I976)

[0.I3] Les systèmes de base de données
C.DELOBEL
Polycopié (Grenoble)
(Juin I975)

[0.I4] Data mapping: a formal hierarchical and relational view(I973)
et
The use of formal description techniques in large data banks
(I972)
E.J.NEUHOLD

[0.I5] Formal definition of mappings in a database
P.PAOLINI,G.PELAGATTI
SIGMOD Toronto
(I977)

[0.I6] Mappings in database systems
G.PELAGATTI,P.PAOLLINI,G.BRACCHI
IFIP Congress proceedings-North Holland publishing Compagny
(I977)

[0.I7] The problem of the cooperation between different DBMS
M.ADIBA,C.DELOBEL
IFIP-TC2 working conference on modelling in DBMS (Nice)
(Janvier I977)

[0.I8] A relational model of data for large shared data banks
E.F.CODD
Communication of ACM (Vol I3,n°6)
(Juin I970)

- - - - -

CHAPITRE I :

L'UTILISATION DU MODELE DE PIVOINES AU NIVEAU EXTERNE

I.I Introduction ..... P.I

I.2 Définition d'un modèle "naturel" d'expression des perceptions P.3

I.2.I. Le processus d'abstraction

I.2.2. Les concepts de base du modèle de perception

I.2.3. La structuration de la perception
- I.2.3.I. Notions de classe et de type
- I.2.3.2. Les classes d'entités
- I.2.3.3. Les classes d'associations
- I.2.3.4. Les classes de propriétés et de valeurs

I.2.4. Un formalisme de type réseau
- I.2.4.I. Au niveau des occurrences
- I.2.4.2. Au niveau des classes

I.3. La notion de vision d'une classe d'utilisateurs ... P.I4

I.4. Le formalisme de PIVOINES au niveau externe P.I6

I.4.I. Au niveau des occurrences
- I.4.I.I. La prise en compte des entités
- I.4.I.2. La prise en compte des associations
- I.4.I.3. La prise en compte des propriétés

I.4.2. Au niveau des classes

I.4.3. Eléments de réflexion
- I.4.3.I. Sur la nature des concepts de base de PIVOINES
- I.4.3.2. Sur la prise en compte des visions utilisateurs
- I.4.3.3. Sur la correspondance⟨LOG-PHYLOG⟩

I.5. L' "expression relationnelle" d'une structure logique (ERLOG) P.25

I.5.I. Rappels sur le modèle relationnel de Codd

I.5.2. L'utilisation d'un modèle du type "entité-relation"

I.5.3. Le passage aux identifiants

I.5.4. Les règles de construction de l' "expression relationnelle" d'une structure logique

I.6. Conclusion P.36

BIBLIOGRAPHIE P.37

° ° ° °

MARCY

I.I Introduction :

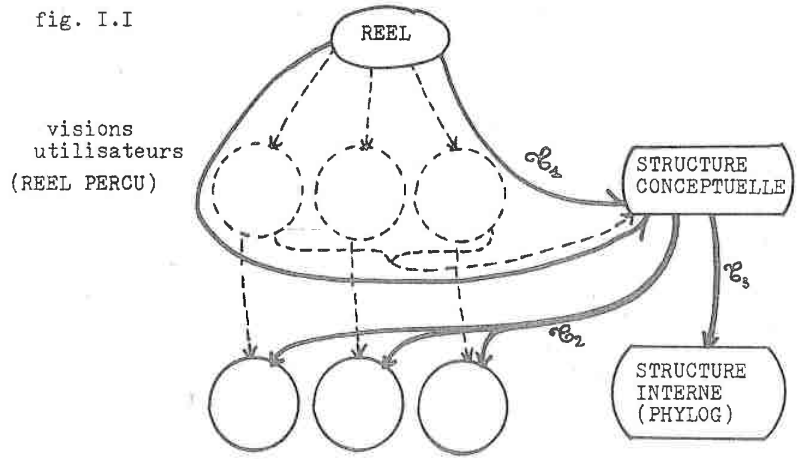
Ce premier chapitre regroupe des éléments d'appréciation et d' approfondissement, concernant l'utilisation du formalisme de PIVOINES au niveau externe.

Le niveau externe, ou "logique" dans la terminologie de PIVOINES, a déjà été caractérisé dans l'introduction générale.

Si on se reporte à la figure 0.6 reprise ci-dessous (fig.I.I), on est conduit à soulever les problèmes suivants :

+ le formalisme de PIVOINES permet-il de rendre compte correctement des différentes visions des utilisateurs? que sont ces visions?
+ de quelle manière ces différentes visions vont-elles se concrétiser dans les structures logiques? quelles vont être les "déformations" admissibles entre les structures logiques et vis à vis de la structure interne (PHYLOG)?

fig. I.I



Pour répondre à ces questions nous serons conduits à :

+ définir un modèle "naturel" d'expression des perceptions (qui se situe donc au niveau noté REEL PERCU ci-dessus)
+ caractériser la notion de vision utilisateur
+ confronter le formalisme de PIVOINES (de type "accès") avec ce formalisme "naturel".

Ces éléments ne constituent pas la description d'un processus de conception (les transitions $\mathcal{C}_2$ et $\mathcal{C}_3$ de la fig. I.I ne sont pas explicitées), pas même dans le cas d'une vision unique qui élimine le problème de la définition d'une structure conceptuelle commune.
Ils constituent cependant un premier élément d'aide à la conception et permettent de donner une certaine consistance à des notions comme celle de repère, au niveau externe.
Enfin, un dernier paragraphe s'attache à définir l'expression relationnelle d'une structure logique qui nous sera utile dans la suite au chapitre 3.

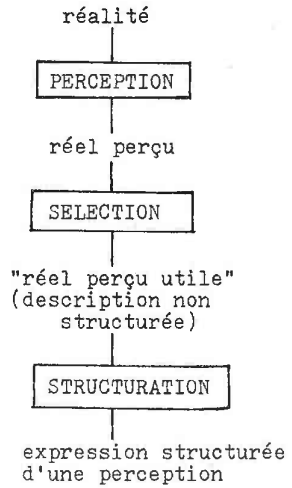
- - - - -

I.2. Définition d'un modèle "naturel" d'expression des perceptions :

I.2.I. Le processus d'abstraction :

Etant donné une certaine classe d'utilisateurs et une portion du monde réel, un certain processus d'abstraction conduit à une interprétation de cette portion de la réalité qui convienne à cette classe d'utilisateurs et qui s'exprime par une description formalisée selon un certain modèle. Ce processus d'abstraction est souvent décomposé en plusieurs étapes (cf. par exemple [I.I]); on peut proposer les étapes suivantes :

fig. I-2
le processus
d'abstraction



La phase de sélection fait la part, dans l'univers des choses perçues, de ce qui est intéressant pour la classe d'utilisateurs concernée (en fonction de ses besoins). La description à ce niveau peut relever des langues naturalles, de restrictions de celles-ci ou de formalismes particuliers(prédicatif dans [I.I], par exemple). Nous ne détaillerons pas cette phase initiale extrêmement importante.
La partie intéressante de la perception va être structurée.
A cet effet, il est nécessaire de définir :

+ un petit nombre de concepts de base (ex: entités, associations etc.....)

+ des types permettant de classer les occurrences des concepts de base observées, en ensembles homogènes (ex: type d'entités "clients", type d'association "fournisseur-produit" ....)
+ des conventions de dénomination et un mode de représentation de la structure ainsi définie.

°°°°

I.2.2. Les concepts de base du modèle de perception :

Un certain consensus se dégage quant à des concepts de base "naturels" observables dans le monde réel (cf. [I.2,I.3,I.4, I.5,I.6,etc....]).

On peut postuler que le réel peut être décrit en termes :

+ d'entités qui sont "définies" (les guillemets vont dans le sens de la remarque de [I.7] «nous devons abandonner l'espoir de définir les entités») par différents auteurs comme :
  - « une chose, abstraite ou concrète, qui a une existence propre et qui forme un tout qui peut être distingué de son environement et en particulier des autres entités» [I.8]
  - « une personne, un lieu, une chose, un concept ou un évènement, réel ou abstrait qui présente de l' interêt pour l'entreprise» [I.9]
  - etc....

  exemples: ce client, ce projet, cette transaction, cette qualité morale etc.....
+ d'associations ("relationships") entre entités, définies comme :
  - « des possibles interactions entre entités » [I.4]
  - « des connexions entre deux ou plusieurs entités, présentant de l'interêt» [I.2]
  - etc....

  exemples: ce lien de parenté, l'appartenance de ce produit à cette commande, etc.....
+ de propriétés des entités et des associations ,définies comme :
  - « des caractéristiques significatives » [I.2]
  - « des qualités que l'on attribue à l'entité ou à l'association » [I.5]
  - etc.....

  exemples: le nom de ce client, la quantité à commander propriété de l' association "fournisseur-produit", etc.....

Dans notre modèle de perception on retrouvera des entités, des associations et des propriétés associées aux entités, associations et propriétés du monde réel.

On peut faire dès maintenant trois remarques importantes sur les concepts de base du modèle de perception :

+ les propriétés se concrétisent par des valeurs obtenues en général par observation ou par mesure. Ces valeurs n'ont pas d'existence par elles-mêmes .

  exemples: le nom de ce client est "TOTO", le pids de cette pièce est en kg "I20"
+ les entités ne peuvent se réduire à la "somme" de leurs propriétés. Par exemple, deux entités distinctes peuvent avoir les mêmes valeurs pour toutes leurs propriétés.
+ l'existence d'une association dépend de l'existence des entités impliquées; de même l'existence d'une propriété dépend de l'existence de l'entité ou de l'association qu'elle caractérise.

°°°°

I.2.3. La structuration de la perception :

I.2.3.I. Notions de classe et de type :

La définition de classes ("sets") et de types va nous permettre de structurer cette perception.

Une classe d'objets (entités, associations, propriétés, valeurs) est un ensemble d'objets "de même nature". Cette nature commune est formalisée dans la notion abstraite de type qui peut être définie à l'aide d'un prédicat d'appartenance ("membership predicate") permettant de décider pour tout objet s'il appartient ou non à la classe du type considéré. Ainsi la notion de classe varie dans le temps, par son contenu, alors que la notion de type est indépendante du temps.

Les entités, les associations, les propriétés et les valeurs sont ainsi regroupées en classes, possédant chacune un type. Dans la suite, pour simplifier, nous parlerons :

+ d'entités, d'associations, de propriétés et de valeurs au niveau des occurrences
+ de classes d'entités, de classes d'associations, de classes de propriétés et de classes de valeurs au niveau des ensembles d'occurrences de même nature (sans conserver la distinction entre classe et type).

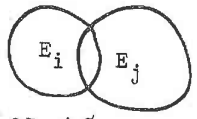
Nous allons approfondir ces diverses notions.

I.2.3.2. Les classes d'entités :

Soit e une entité. Elle appartient à une classe E_i. Il est évident que dans certains cas des utilisateurs"voient" naturellement des classes d'entités non disjointes ([I.3,I.4.]) On peut distinguer essentiellement :

+ des classes ayant une intersection non vide

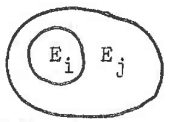
ex: "clients" et "fournisseurs" (certains fournisseurs peuvent être des clients)



$E_i \cap E_j \neq \emptyset$
($E_i \cup E_j$ peut être nommée ou non)

+ des classes totalement incluses dans d'autres classes

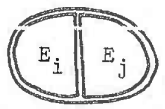
ex: "managers" parmi "employés"



$E_i \subset E_j$

+ des partitions disjointes de classes

ex: "véhicules" regroupant "véhicules légers" et "poids lourds"



E_k

$E_k = E_i \cup E_j$ et $E_i \cap E_j = \emptyset$

Un modèle "naturel" doit intégrer ces cas: il peut le faire en admettant des classes "composées" comportant des classes "composantes", disjointes ou non.

I.2.3.3. Les classes d'associations:

$(e_I, e_2, \ldots, e_n)$ est une association où les e_i sont les entités impliquées. C'est un élément de la classe d'association de nom R :

$$R = \{ (e_I, e_2, \ldots, e_n) \mid e_I \in E_I, \ldots, e_n \in E_n \}$$

On peut écrire aussi : $R \subseteq E_I x E_2 x \ldots x E_n$ (produit cartésien)

L'ordre des entités dans une association peut être rendu non significatif à l'aide d'un "nom de rôle" qui identifie la fonction de l'entité dans l'association :

$[e_I/r_I, e_2/r_2, \ldots, e_n/r_n]$ e_i :entité r_i :nom de rôle

Un modèle "naturel" doit admettre des associations où :

+ les E_i ne sont pas distincts
  ex : association "mariage" $M \subseteq$ PERSONNExPERSONNE
+ l'arité (n) est supérieure à 2
  (quand n = 2 on parle d'association binaire)
+ on retrouve les mêmes produits cartésiens de classes d'entités
  ex : sur COMMANDExPRODUIT peut être définie l'association entre commande et produit de la commande de même que l'association entre commande et produit prioritaire de la commande.

I.2.3.4. Les classes de propriétés et de valeurs :

Les valeurs de propriété sont regroupées en classes V_i ($v \in V_i$). Les classes de propriétés peuvent être définies comme des fonctions totales P_i :

$$P_i : D \longrightarrow \mathcal{V}^* \cup \{\omega\} \quad \text{où} \quad \begin{cases} D = E_i \text{ ou } R_i \\ \mathcal{V} = V_i \text{ ou } V_{i_I} \times V_{i_2} \times \ldots \times V_{i_n} \\ \mathcal{V}^* = \mathcal{V} \times \mathcal{V} \times \ldots \times \mathcal{V} \\ \omega \text{ est la valeur "valeur existante mais non connue".} \end{cases}$$

Une propriété est un triplet (P_i, d, v) avec $d \in D$, $v \in \mathcal{V}^* \cup \{\omega\}$

C'est la présentation de PIROTTE [I.8], qui distingue quatre catégories de classes de propriétés :

+ "simple non répétitive" $P_i : D \longrightarrow V_i \cup \{\omega\}$
  ex: NOM est une classe de propriétés simple non répétitive de la classe d' entités PERSONNES

+ "complexe non répétitive" $P_i : D \rightarrow V_{i_1} \times V_{i_2} \times \ldots \times V_{i_n} \cup \{\omega\}$

ex: ADRESSE est une classe de propriétés complexe non répétitive de PERSONNES

+ "simple répétitive" $P_i : D \longrightarrow V_i^* \cup \{\omega\}$

ex: DISTRACTIONS est une classe de propriétés simple répétitive de PERSONNES

+"complexe répétitive" $P_i : D \longrightarrow (V_{i_1} \times V_{i_2} \times \ldots \times V_{i_n})^* \cup \{\omega\}$

ex: ENFANTS est une classe de propriétés complexe répétitive de PERSONNES

ENFANTS : PERSONNES $\longrightarrow$ (PRENOM, AGE)$^* \cup \{\omega\}$

Il existe une différence fondamentale entre une entité et une propriété complexe. Cette dernière n'a pas d'existence par elle-même et elle se réduit à la "somme" de ses valeurs constituantes. Ce qui n'est pas le cas des entités ainsi que cela a été souligné plus haut.

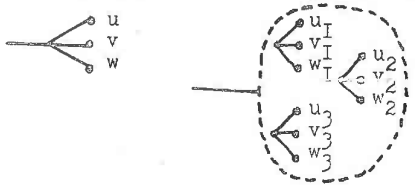
°°°°

I.2.4. Un formalisme de type réseau :

I.2.4.I. Au niveau des occurrences :

Il s'agit d'un réseau d'occurrences ("network of instances" [I.4])

- entité: O
- association: (binaire) (arité=3) etc..
- propriété:

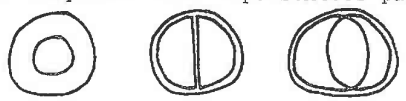


I.2.4.2. Au niveau des classes :

Un réseau d'occurrences se suffit à lui même : il exprime toute la sémantique sélectionnée. En structurant en classes, on noie les occurrences dans des ensembles homogènes. Une représentation au niveau des classes ("type level") perd toujours une certaine part de la sémantique. Par exemple, le fait que lorsqu'un numéro INSEE commence par 2 il existe pour l'entité PERSONNE correspondante un nom de jeune fille, apparaît dans le réseau des occurrences mais disparaît en général dans une représentation au niveau des classes. Un cerain nombre de prescriptions structurelles devront donc être exprimées en dehors de la représentation au niveau des classes, dans un langage adapté. Elles sont communément appelées contraintes d'intégrité (cf. [I.3, I.4, I.5, I.IO, ...]).

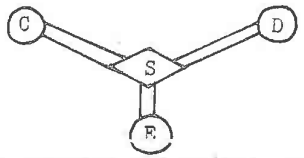
Notre représentation en réseau au niveau des classes s'inspire du "entity-relationship diagram" de CHEN [I.3] et du "EPA network" de PIROTTE [I.4] .

- une classe d'entités est représentée par un noeud du réseau, contenant le nom de la classe
  Les classes composées sont représentées par (cf. parag.I.2.3.2.) :

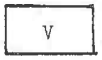


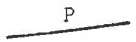
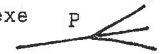
- une classe d'associations est représentée par un losange contenant le nom de la classe et connecté aux classes d'entités concernées.

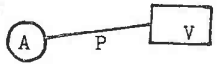
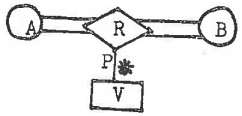
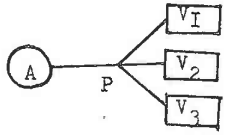
  classe binaire :

  classe d'arité 3 : 

  (les noms de rôle sont sous-entendus chaque fois que la classe d'entités ne joue qu'un seul rôle; ils sont alors confondus avec le nom de la classe d'entités. La fig. I.2 illustre des cas où ces noms de rôle sont indispensables).
- une classe de valeurs est représentée par un rectangle contenant le nom de la classe :

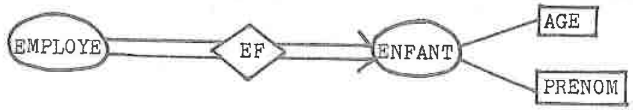


• une classe de propriétés est représentée par un arc simple  ou complexe  ,accompagné du nom de la classe de propriétés et d'une éventuelle astérisque ✱ marquant le caractère répétitif.

- classe de propriétés simples non répétitive d'une classe d'entités 
- classe de propriété simple répétitive d'une classe d' association 
- classe de propriété complexe non répétitive d'une classe d'entités 
- etc..... (cf. fig. I.2)

Un certain nombre de prescriptions structurelles complémentaires importantes peuvent être intégrées à la représentation. C'est le cas de :

• l' "existence dependency" concernant certaines classes d' entités, introduite dans [I.3] et notée ici par une flèche.

ex: 

l'existence d'une entité "enfant" dépend de l' existence de l'entité "employé" qui lui est obligatoirement associé par EF .

A première vue cette possibilité de dépendance va à l' encontre de la définition même d'une entité comme un tout, ayant une existence propre,indépendante de l'environement. Cette possibilité permet de traduire comme entité des choses usuellement perçues comme des entités (ex: des enfants) mais qui, dans la portion restreinte du monde réel qui nous intéresse devraient avoir le statut de propriété.

ex: (cf. exemple ci-dessus)

C'est également le moyen d'exprimer des visions hiérarchisées

(courantes chez certains utilisateurs) ainsi que d'éventuelles connexions entre propriétés.

• la "cardinalité" des classes d'associations : pour le cas binaire on trouve dans [I.II] une caractérisation de la cardinalité par 4 entiers (i,j,k,l) tels que :

$i \leq$ nb. d'entités de Y associées à une entité quelconque de X $\leq j$

$k \leq$ nb. d'entités de X associées à une entité quelconque de Y $\leq j$

($j \geq i ; l \geq k ; j$ et/ou l peuvent être ∞)

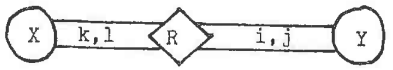
Donc, si par commodité d'expression on "oriente" les associations on peut dire :

si
- $i > 0$ la classe d'associations de X vers Y est totale
- $i = 0$ " " " " " " partielle
- $k > 0$ " " " de Y vers X est totale
- $k = 0$ " " " " " " partielle
- $j = I$ " " " de X vers Y est fonctionnelle
- $j > I$ " " " " " " multivoque
- $l = I$ " " " de Y vers X est fonctionnelle
- $l > I$ " " " " " " multivoque

Cela généralise les caractérisations I:I,I:n,n:I,n:m [I.3]

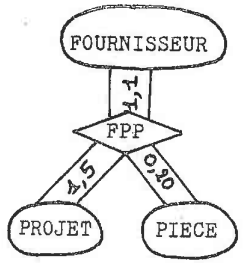
I:I ⇔ (...,I,...,I)
I:n ⇔ (...,n,...,I) avec n>I
n:I ⇔ (...,I,...,n) "
n:m ⇔ (...,m,...,n) avec n et m>I

On utilisera la notation :



Cette caractérisation peut être étendue aux classes d'arité supérieure à 2. Soit une classe d'associations d'arité p+I : les entiers a,b sur une branche représentent respectivement le nombre minimum et maximum d'entités associées à un p-uple quelconque d'entités des p autres branches.

exemple:

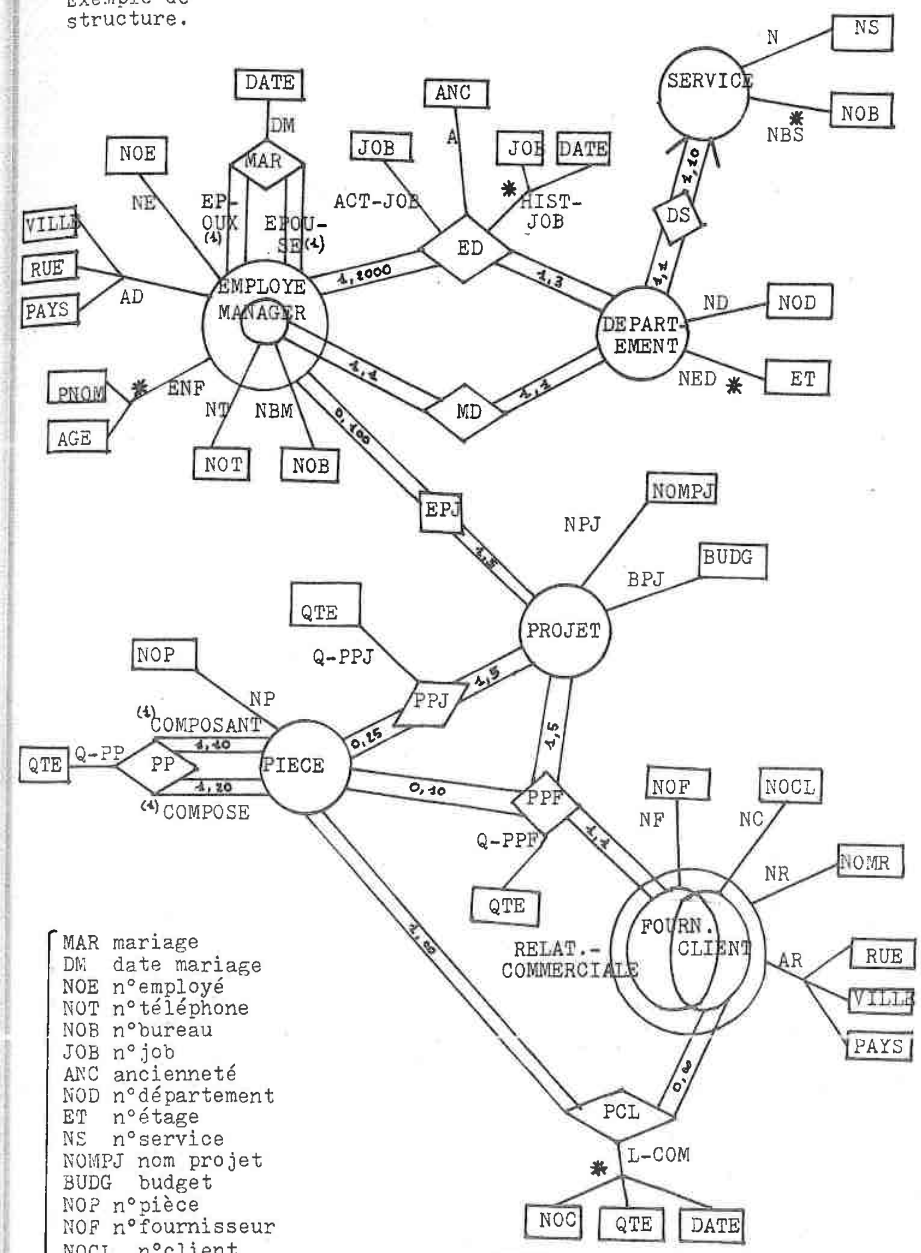


- une pièce d'un projet a un et un seul fournisseur
- un fournisseur approvisionne de 0 à 20 pièces pour un projet donné
- une pièce d'un fournisseur peut concerner de I à 5 projets

La figure I.2 présente un exemple complet de structure: la description est claire et lisible car le formalisme s' appuie sur des concepts "naturels".

- - - - -

fig. I.2
Exemple de structure.



MAR mariage
DM date mariage
NOE n°employé
NOT n°téléphone
NOB n°bureau
JOB n°job
ANC ancienneté
NOD n°département
ET n°étage
NS n°service
NOMPJ nom projet
BUDG budget
NOP n°pièce
NOF n°fournisseur
NOCL n°client
NOMR nom relation commerciale
NOC n°commande

(4) nom de rôle de la classe

I.3. La notion de vision d'une classe d'utilisateurs :

Un schéma comme celui de la figure I.2 permet d'exprimer la vision d'une classe d'utilisateurs sur une portion du monde réel.

En premier lieu,il faut noter que tous les utilisateurs d'une base ne s'intéressent pas obligatoirement à la même portion du monde réel. Par rapport à la structure conceptuelle commune, la vision d'un utilisateur est réduite à un sous-ensemble; on retrouve la notion de "subschema" de CODASYL [I.I2].

En second lieu, la définition des éléments de base et leur structuration en classes peuvent différer d'un utilisateur à l'autre. Par exemple, face aux mêmes entités du monde réel, les utilisateurs peuvent superposer des définitions relatives à l'interêt qu'ils ont pour elles.

exemple : un premier utilisateur peut n'être concerné que par les caractéristiques de personne des relations commerciales (nom,adresse,n°téléphone,etc....); il définira une classe d'entités "relations commerciales".
Un second utilisateur, qui les appréhende en tant que "fournisseurs" et "clients" définira deux classes d'entités.

Par rapport à une même classe d'entités, les utilisateurs prendront en compte un plus ou moins grand nombre de propriétés et sous des formes qui peuvent être différentes.

Le problème est particulièrement net pour les classes d'associations. Il est clair qu'un utilisateur n'est pas concerné par toutes les associations "potentielles" entre classes d'entités et qu'il doit sélectionner celles qui sont significatives par rapport à son travail. Mais le choix même de leur forme est délicat : choix de l'arité, niveau de redondance entre les associations etc....

La vision de l'utilisateur est précisément le reflet de tels choix. Les techniques visant à extraire des associations fondamentales et donc à exprimer la sémantique d'une "manière propre" (formes normales de Codd [I.I3], associations primitives de Pirotte [I.4], associations irréductibles de Rissanen et Delobel [I.I4] etc....) relèvent plus du niveau conceptuel que de l'expression des perceptions des utilisateurs.

Enfin, de nombreux travaux ont mis l'accent sur la relativité

même du choix entre entité, association et propriété ([I.3,I.4, I.5,I.6,etc...]).

exemples :

- dans [I.3] on trouve l'exemple du mariage qui peut être vu comme :
  + une entité (par exemple pour l'état-civil)
  + une association entre des entités "personne"
- dans [I.4] et [I.5] on trouve l'exemple de la notion de couleur qui peut être vue comme :
  + une propriété d'une entité (couleur d'une voiture, des cheveux d'une personne...)
  + une entité (pour les fabriquants de couleurs ou dans un contexte tel que "le jaune a pour longueur d'onde 5800 Å ")
- dans [I.6] on souligne que ces différentes visions transparaissent au niveau du langage courant, comme dans les expressions suivantes :
  + "c'est un parent" (les entités sont les parents)
  + "cette personne a des enfants" (les entités sont les personnes avec des propriétés"enfants")
  + "cette personne est le parent de cet enfant" (les entités sont les personnes et les enfants reliées par des associations de parenté).

Pour toutes ces raisons les diverses visions peuvent être très différentes les unes des autres. Il semble important, afin que les interfaces entre la base et les utilisateurs soient naturels et pas trop contraignants, que les structures externes puissent refléter une grande part de la spécificité des visions utilisateurs. Nous allons dans le paragraphe suivant confronter le modèle de PIVOINES à cette exigence, en le confrontant d'abord au modèle naturel que nous venons de définir.

- - - - -

I.4 Le formalisme de PIVOINES au niveau externe :

I.4.I. Au niveau des occurrences :

Le "graphe généralisé" (cf. [I.I6] p.27-28, [I.I7] p.595) est une représentation au niveau des occurrences que l'on peut rapprocher du "réseau des occurrences" du paragraphe I.2.4.I.

I.4.I.I. La prise en compte des entités :

Aux entités correspondent des "repères" PIVOINES qui constituent les noeuds du graphe généralisé.

• r_i

I.4.I.2. La prise en compte des associations :

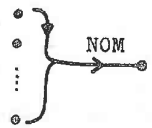
Aux associations correspondent grossièrement des "relations PIVOINES" représentées par des arcs ou des arcs généralisés dans le graphe. Les deux notions sont différentes dans leur esprit : une association exprime l'existence d'une certaine connexion, significative pour l'utilisateur, entre des entités alors qu'une relation PIVOINES est l'image d'un mécanisme logique qui permet de cheminer dans un certain sens depuis un (ou des) repère(s) vers un repère but.
Le passage à une "structure d'accès" peut impliquer des choix supplémentaires quant aux cheminements les plus adaptés : en particulier une association significative peut ne pas donner naissance à un chemin d'accès si cet accès est possible par d'autres voies.
Dans le cas général :

+ à une association binaire peut correspondre une relation PIVOINES d'arité 2, représentée par un arc :

(NOM est le nom de la classe de relations)

Le choix d'une certaine orientation ne porte pas à conséquences car le langage d'accès autorise l'emploi de l'accès inverse (NOM^{-I}).

+ à une association d'arité supérieure à 2 peut correspondre une relation PIVOINES de même arité, représentée par un "arc généralisé"

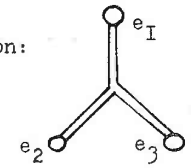


(la flèche noire indique l'ordre du n-uple de départ)

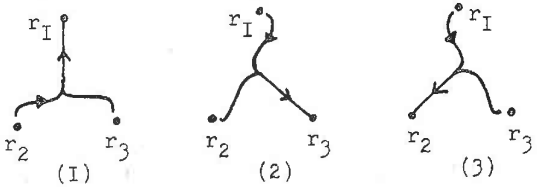
Le choix d'une orientation impose à priori une certaine configuration.

exemple:

à l'association:

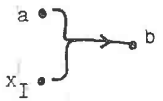


peuvent être associées aussi bien les arcs généralisés (I),(2) ou (3).



Il faut noter cependant, d'une part qu'il existe des cas où une configuration est effectivement prépondérante et d'autre part que le langage d'accès permet de faire comme s'il n'y avait pas de configuration privilégiée.

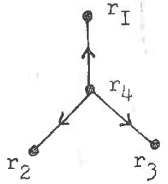
exemple :



où a ret b sont des sommets connus; on peut en déduire les valeurs de l'inconnue x_I.

Une autre solution, qui est la solution classique des modèles exclusivement binaires (cf. [I.I8]) consiste à introduire un repère supplémentaire qui "matérialise" l'occurrence de l'association.

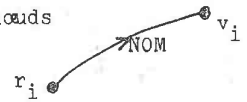
exemple: dans le cas de figure ci-dessus, on aurait



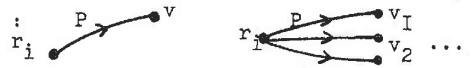
I.4.I.3. La prise en compte des propriétés :

Il est nécessaire de distinguer plusieurs cas :

+ aux propriétés simples d'entités correspondent des relations d'accès binaires à des valeurs (les valeurs constituant d'autres nœuds du graphe généralisé de PIVOINES)



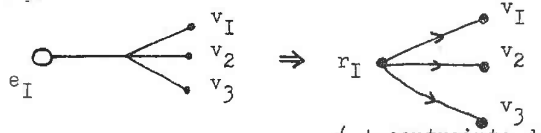
Suivant que la propriété est répétitive ou non, on aura plusieurs ou une seule relation d'accès aux valeurs pour un même repère :



On note qu'il n'y a pas dans le modèle de PIVOINES de distinction entre association et propriété : les deux s' "expriment" par des relations d'accès.

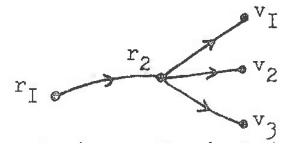
+ aux propriétés complexes non répétitives d'entités peuvent correspondre des accès simples aux valeurs et une contrainte d'intégrité indiquant que ces valeurs constituent un tout (pour les distinguer de plusieurs propriétés simples): soit elles sont toutes définies, soit elles sont toutes indéfinies.

exemple :



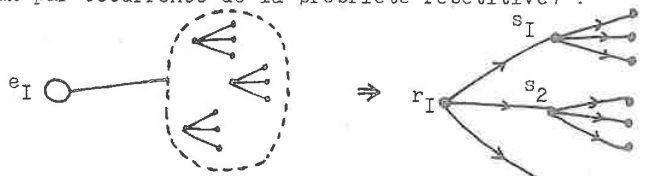
(+ contrainte d'intégrité)

La tendance qui consiste à "matérialiser" la propriété par un repère semble néfaste, car inutile. C'est une des raisons de la "prolifération des types de repères" lors de la conception des structures constatée dans [I.I9].



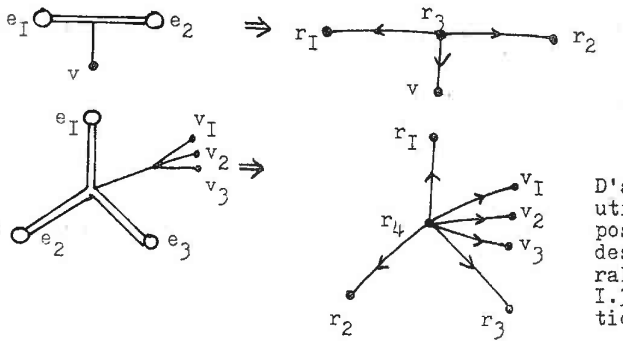
(la contrainte d'intégrité signalée ci-dessus est toujours nécessaire et il en faut une seconde indiquant que l'existence de r_2 est liée à celle de r_I).

+ aux propriétés complexes répétitives d'entités doivent correspondre nécessairement des repères supplémentaires (un par occurrence de la propriété répétitive) :



(avec une contrainte d'intégrité indiquant que l'existence des s_i est liée à l'existence de r_I)

+ les propriétés (simples ou complexes, répétitives ou non) d'associations n'ont pas de correspondant direct. Leur prise en compte peut se faire en "matérialisant" les associations par des repères et en associant les propriétés à ces repères :



D'autres solutions sont possibles avec des arcs généralisés (cf. fig. I.3 l'association PPJ).

I.4.2. Au niveau des classes :

Les repères, les relations et les valeurs sont regroupées en classes. Ces classes sont nommées.
Nous allons tout d'abord présenter une représentation au niveau des classes :

classe de repères (n est le nom de la classe ou un numéro qui renvoie à une table des noms)

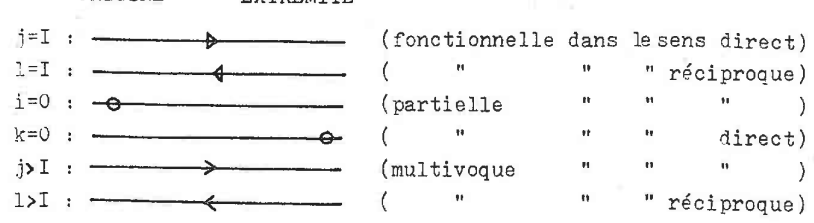
classe de valeurs

classe de relations binaires

classe d'arcs généralisés

Le nom de la classe de relations est porté du côté de la flèche caractérisant le sens direct :

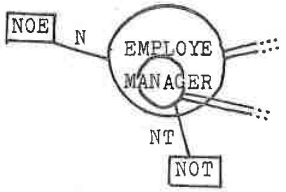
Pour indiquer la cardinalité nous adopterons des conventions proches de celles de [I.20], à savoir :
(les notations i,j,k,l sont présentées en I.2.4.2)



Soit en particulier :

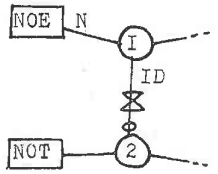
Un point important est qu'un repère ne peut avoir qu'un type unique; la prise en compte des classes d'entités composées implique des schémas assez peu naturels où une même entité est représentée par plusieurs repères distincts.

exemple :



où { NOE : n° employé
NOT : n° téléphone d'un manager

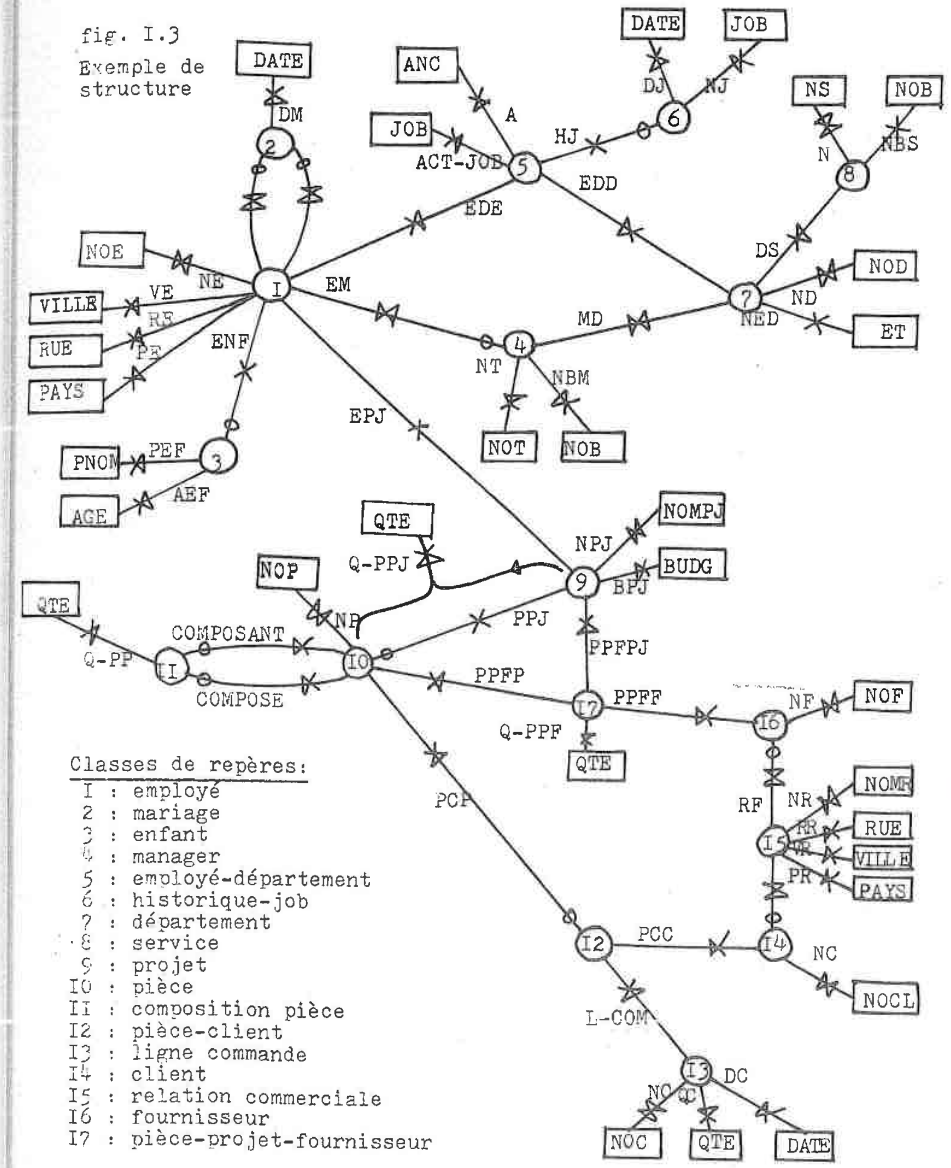
peut être représenté par :



où { (I) est la classe de repères "employés"
(2) " " " " " "managers"
ID est une relation I : I, partielle de (I) vers (2), qui indique l'identité des entités représentées à la fois dans (I) et (2).

D'autres exemples apparaissent dans la figure I.3, qui représente un schéma PIVOINES susceptible d'être associé au réseau de la figure I.2 .

fig. I.3
Exemple de structure



Classes de repères:

I : employé
2 : mariage
3 : enfant
4 : manager
5 : employé-département
6 : historique-job
7 : département
8 : service
9 : projet
I0 : pièce
II : composition pièce
I2 : pièce-client
I3 : ligne commande
I4 : client
I5 : relation commerciale
I6 : fournisseur
I7 : pièce-projet-fournisseur

I.4.3. Eléments de réflexion :

I.4.3.I. Sur la nature des concepts de base de PIVOINES :

Il s'agit des notions de relation et de repère.

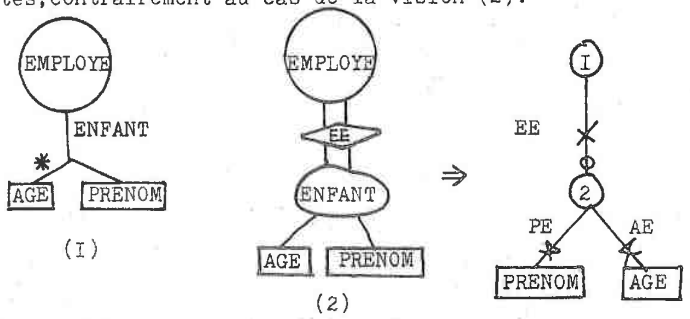
a) la notion de relation PIVOINES : elle correspond à un mécanisme d'accès(*) orienté; mais le langage d'accès permet de ne pas tenir compte du sens (c.à.d. que le sens direct peut perdre son privilège vis à vis du sens réciproque) et de ne pas différencier le but des autres arguments pour les relations d'arité supérieure à 2.
De ce point de vue, la notion de relation PIVOINES s'apparente assez bien à la notion d'association.
La différence majeure est l'impossibilité d'associer des propriétés à une relation.

b) la notion de repère PIVOINES : un repère peut correspondre à une entité de l'utilisateur mais aussi à une propriété complexe répétitive, à la "matérialisation" d'une association avec propriété ou d'une association d'arité supérieure à 2 etc....
La comparaison des figures I.2 et I.3 illustre parfaitement ce fait : I7 types de repères correspondent à seulement 9 types d'entités (6 types"principaux" et 3 types "composés").
La notion de repère risque donc de ne pas être directement compréhensible aux utilisateurs. Des réalisations effectives, portant sur des cas très simples [I.I9,I.2I] confirment quelque peu ces craintes. Le modèle "naturel" du paragraphe I.2. pourrait peut être servir d'intermédiaire, plus proche des utilisateurs, dans le processus de conception (bien que ce modèle puisse conduire à des choix quelque peu artificiels -ex: entre entité dépendante et propriété complexe répétitive).
~~Enfin le nombre plus élevé de contraintes d'intégrité~~ on PIVOINES (certaines sont notées au paragraphe I.4.I.) porrait également caractériser certaines faiblesses du modèle de PIVOINES, si on accepte l'interprétation des

(*) mécanisme logique, qui peut être très différent des mécanismes d'accès physiques.

contraintes d'intégrité donnée dans [I.8] (P.23) : « ces contraintes ont aussi un rôle structurel évident, qui est de pallier aux faiblesses du formalisme principal ».
Ces quelques remarques critiques ne doivent pas cacher les qualités évidentes du formalisme de PIVOINES (simplicité, homogénéité,...) qui ont permis de développer un langage d'accès et d'interrogation particulièrement naturel (cf. [I.16]).

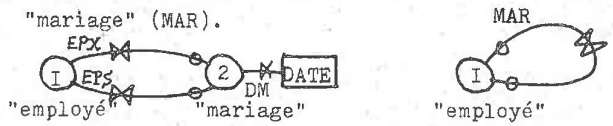
I.4.3.2. Sur la prise en compte des visions utilisateurs :
En conséquence des remarques précédentes, il faut noter qu'il n'est pas toujours possible de rendre compte nettement de visions différentes.
Exemple : aux 2 visions ci-dessous corresponderont une même structure PIVOINES. Dans la vision (I) les enfants ne sont que des propriétés des entités "employé" : ils n'ont pas d'existence propre et sont supprimés en même temps que les entités, contrairement au cas de la vision (2).



(seule une contrainte d'intégrité pourra rendre compte de cette différence).

Dans la majorité des cas, les visions utilisateurs peuvent être prises en compte et traduites :

+ par des classes de repères ou des classes d'associations différentes
  ex: classe de repères "relations commerciales" concurremment aux classes "clients" et "fournisseurs".
+ par des classes de repères à la place de classes de relations ou inversement.
  ex: classe de repères "mariage" et classe de relations "mariage" (MAR).



I.4.3.3. Sur la correspondance LOG - PHYLOG :
Plus que les différences entre visions utilisateurs, concrétisées par des structures logiques différentes, nous sommes concernés par les différences entre structures logiques et structure interne (PHYLOG).
L'existence de classes de repères différentes entre structures logiques pour une même notion et le grand interêt d'une prise en compte unique au niveau interne, sous forme d'une classe de repères de PHYLOG par exemple, est un bon exemple où l'on trouve une classe de repères logiques sans correspondant direct dans PHYLOG.
exemple : prenons le cas élémentaire suivant

niveau LOG
+ vision I : classe de repères "employés"
+ vision 2 : classe de repères "cadres" (qui correspond à une partie de la classe "employés")

niveau PHYLOG : classe de repères "employés"
La classe de repères logique "cadre" n'a pas de correspondant direct dans PHYLOG.

Une véritable prise en compte des visions utilisateurs passe donc par la résolution du problème de la différenciation des classes de repères logiques des classes de repères de PHYLOG ("correspondance élargie"), la différenciation des classes d'associations étant chose acquise.

- - - - -

I.5. L'expression relationnelle d'une structure logique (ERLOG) :

Nous avons précisé au paragraphe 0.3.2. son interêt. Nous avons également montré que le point crucial pour la correspondance <LOG-PHYLOG> concernait les classes de repères. Les expressions relationnelles de structures PIVOINES devront donc clairement conserver présente la notion de repère.

I.5.I. Rappels sur le modèle relationnel de Codd :

Dans ce modèle [I.22] ,une base de données est représentée par un ensemble de relations. Chaque relation peut être caractérisée par un nom et une liste de domaines constitutifs. Un domaine est une classe de valeurs, possédant un nom ("attribute name" dans[I.23]quand le nom est préfixé par un "role name" qui rend l'ordre des domaines non significatif) .Une relation est un sous-ensemble du produit cartésien des domaines qui la constituent et donc un ensemble de n-uples.

Précisons encore deux définitions fondamentales qui nous seront utiles dans la suite :

+ le domaine D_I est fonctionnellement dépendant du domaine D_2 (noté : $D_2 \longrightarrow D_I$) si, pour tout $d_2 \in D_2$, il y a 0 ou I élément $d_I \in D_I$ associé à d_2.
+ une clé K d'une relation R est une combinaison de domaines (éventuellement un seul) possédant les propriétés PI et P2 suivantes :
  - PI ("unique identification") : chaque n-uple de R est identifié univoquement par la valeur de K
  - P2 ("non redundancy") : aucun domaine de K ne peut être retiré sans que disparaisse la propriété PI.

I.5.2. L'utilisation d'un modèle du type "entité-relation" :

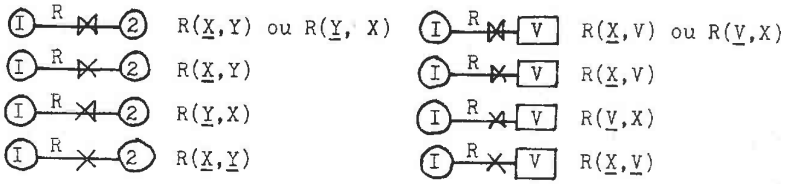
Plusieurs "extensions" du modèle relationnel classique, où est conservée explicitement la notion d'entité, ont étées définies (modèles "entité-relation" de [I.3,I.7,..]).Aux classes d'entités correspondent des domaines des relations dont les valeurs sont des codes internes qui repèrent univoquement chaque entité. Ces codes sont baptisés "surrogates" dans[I.7] ,ce qui peut être traduit par "substituts d'entités".

Notre formalisme "naturel" du paragraphe I.2.4. pourrait ainsi

être exprimé sous une forme relationnelle.

Pour notre problème (recherche d'une forme relationnelle qui conserve présente la notion de repère) il est assez naturel de suivre la même démarche, dont les règles sont les suivantes:

+ à chaque classe de repères correspond une classe de "substituts de repères" (qui peut être domaine d'une ou plusieurs relations)
+ les classes de valeurs sont les autres domaines des relations
+ à chaque classe de relations PIVOINES correspond une relation dont on montre facilement qu'elle prend une des formes suivantes selon les caractéristiques de la classe de relations PIVOINES (la clé est soulignée) :



où X et Y sont les classes de substituts de repères associées à (I) et (2) .

+ à chaque classe de relations d'arité supérieure à 2 de PIVOINES correspond une relation comprenant toutes les classes de substituts de repères et de valeurs concernées.

L'expression relationnelle ainsi obtenue est unique (au choix des clés de certaines relations près).

exemple : à la structure de la fig.I.3 correspond l'expression relationnelle suivante (le nom des relations est celui des relations de PIVOINES et le nom des domaines, celui des classes de valeurs) :

NE(EMPLOYE,NOE)
VE(EMPLOYE,VILLE)
RE(EMPLOYE,RUE)
PE(EMPLOYE,PAYS)
DM(MARIAGE,DATE)
EPOUX(EMPLOYE,MARIAGE)
EPOUSE(EMPLOYE,MARIAGE)
ENF(EMPLOYE,ENFANT)
PEF(ENFANT,PNOM)
AEF(ENFANT,AGE)
EDE(ED,EMPLOYE)
ACT-JOB(ED,JOB)

A(ED,ANC)
HJ(ED,HJOB)
DJ(HJOB,DATE)
NJ(HJOB,JOB)
EDD(ED,DEPARTMT)
EM(EMPLOYE,MANAGER)
NT(MANAGER,NOT)
NBM(MANAGER,NOB)
MD(MANAGER,DEPARTMT)
DS(SERVICE,DEPARTMT)
N(SERVICE,NS)
NBS(SERVICE,NOB)
ND(DEPARTMT,NOD)
NED(DEPARTMT,ET)
EPJ(EMPLOYE,PROJET)
NPJ(PROJET,NOMPJ)
BPJ(PROJET,BUDG)
PPJ(PROJET,PIECE)
Q-PPJ(PROJET,PIECE,QTE)
NP(PIECE,NOP)
COMPOSANT(CP,PIECE)
COMPOSE(CP,PIECE)
Q-PP(CP,QTE)
PCP(PCL,PIECE)
L-COM(PCL,LCOM)
NC(LCOM,NOC)
QC(LCOM,QTE)
DC(LCOM,DATE)
PCC(LCOM,CLIENT)
NC(CLIENT,NOCL)
RC(CLIENT,RELCOM)
NR(RELCOM,NOMR)
RR(RELCOM,RUE)
VR(RELCOM,VILLE)
PR(RELCOM,PAYS)
RF(RELCOM,FOURN)
NF(FOURN,NOF)
PPFPJ(PPF,PROJET)
PPFP(PPF,PIECE)
PPFF(PPF,FOURN)
Q-PPF(PPF,QTE)

la correspondance entre les classes de substituts de repères et les classes de repères est la suivante :

EMPLOYE	(I)
MARIAGE	(2)
ENFANT	(3)
MANAGER	(4)
ED	(5)
HJOB	(6)
DEPARTMT	(7)
SERVICE	(8)
PROJET	(9)
PIECE	(10)
CP	(11)
PCL	(12)
LCOM	(13)
RELCOM	(14)
FOURN	(15)
PPF	(16)

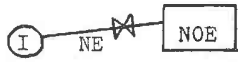
Malheureusement, il n'est pas possible de conserver les substituts de repères dans l'expression de la correspondance <LOG-PHYLOG>, car on ne retrouve pas obligatoirement les mêmes classes de substituts aux deux niveaux de structure, c.à.d. dans ERLOG et ERPHYLOG (on suppose ERPHYLOG construite de la même manière que ERLOG).
Une solution peut consister à remplacer les classes de substituts par des "identifiants" (c'est la démarche décrite par CHEN [I.3] comme passage à un niveau de structure où toutes les notions « sont représentées par des données de la base »).Le paragraphe suivant nous présente cette démarche.

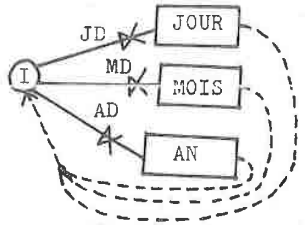
I.5.3. Le passage aux identifiants :

L'identifiant d'une classe de repères est un ensemble de classes de valeurs de la base liées à cette classe de repères, dont les valeurs identifient univoquement les repères (cf. la notion d' "identifier of an entity class" [I.4]).
On associera un identifiant à chaque classe de repères; soit cet identifiant s'impose naturellement, soit il est le résultat d'un choix.
exemples :

- soit le type de repères (10) de la figure I.3 ("pièces"). Il possède un identifiant "naturel", c'est à dire une classe de valeurs qui est utilisée à des fins d'identification dans l'organisation étudiée : le n° de pièce (NOP)
- soit le type de repères (2) de la figure I.3 ("mariage"). Un mariage peut être identifié indifféremment par le n° employé de l'époux ou par celui de l'épouse (classe de valeurs NOE via la relation EPOUX ou classe de valeurs NOE via la relation EPOUSE). Il s'agit ici d'un choix arbitraire.

On pourra faire figurer sur les schémas de structure les identifiants, ce qui en précisera la définition. On distinguera :

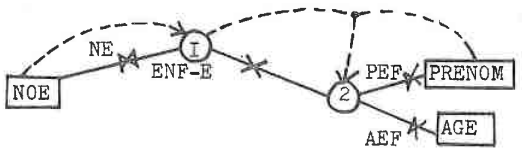
- des identifiants simples (une seule classe de valeurs)
  notation : ----->
  ex : 

  (cf. FIG. I.3)
- des identifiants multiples (plusieurs classes de valeurs associées à la classe de repères)
  notation :
  ex : 

  (I) : date
- des identifiants composés (plusieurs classes de valeurs de plusieurs classes de repères liées)
  notation :

+ la mention "via ....." est portée quand il y a plusieurs chemins possibles
+ si une flèche part d'une classe de repères, l'identifiant de cette classe fait partie de l'identifiant désigné.

ex:



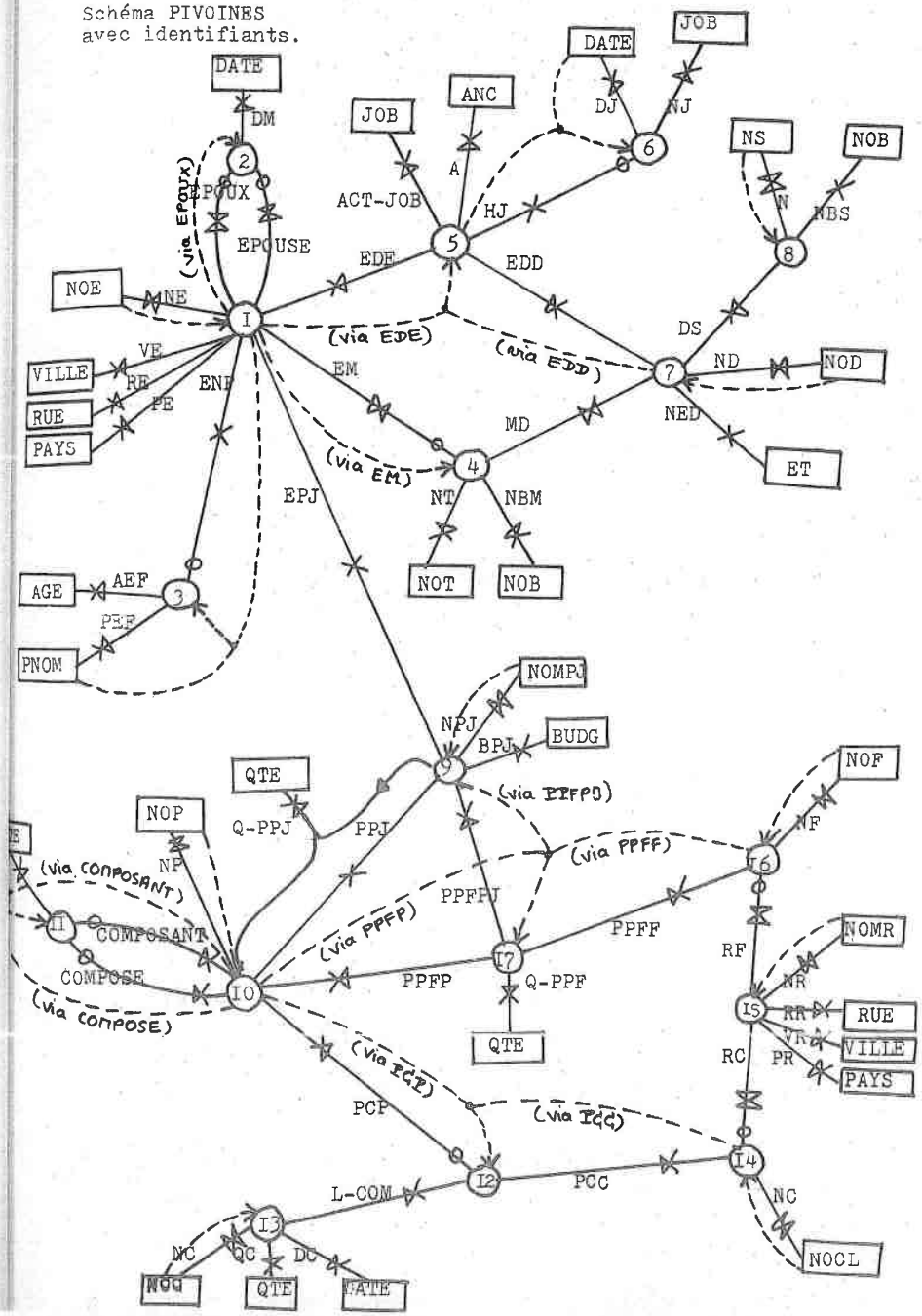
(cf. fig.I.3)

L'identifiant de (2) est constitué de l'identifiant de (I) (NOE) et de PRENOM.

{ (I) : employé
{ (2) : enfant

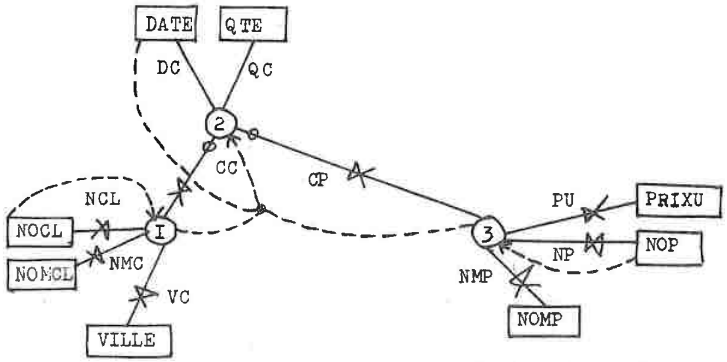
La figure I.4 reprend la structure représentée dans la figure I.3 avec tous les identifiants : elle constitue une représentation plus explicite de la structure PIVOINES au niveau des classes.

Fig. I.4.
Schéma PIVOINES
avec identifiants.



Revenons aux expressions relationnelles en remplaçant les classes de substituts de repères par les identifiants des classes de repères. Cette substitution va entraîner une importante redondance parmi les relations : en effet, les identifiants composés intègrent certaines relations PIVOINES entre classes de repères, qu'il devient redondant de traduire à leur tour. D' autres relations vont se simplifier. Enfin la structure va "s'applatir", c'est à dire que n'apparaîtront plus clairement ni les classes de repères, ni les liaisons entre celles-ci.

Montrons le sur un exemple simple, pris au niveau des occurrences :



classes de repères :

- I : client
- 2 : commande journalière par client et par produit
- 3 : produit

Traduit par :

VC(CLIENT,VILLE)
NCL(CLIENT,NOCL)
NMC(CLIENT,NOMCL)
CC(CJCP,CLIENT)
DC(CJCP,DATE)
QC(CJCP,QTE)
CP(CJCP,PRODUIT)
NP(PRODUIT,NOP)
NMP(PRODUIT,NOMP)
PU(PRODUIT,PRIXU)

où la correspondance entre classes de substituts de repères et classes de repères est : CLIENT (I), CJCP (2), PRODUIT (3)

Avec au niveau des occurrences, par exemple :

NCL

CLIENT	NOCL
c_1	I
c_2	3
c_3	9
c_4	I4
..	..

NMC

CLIENT	NOMCL
c_1	TOTO
c_2	ARTHUR
c_3	JOE
c_4	DUPONT
..	...

VC

CLIENT	VILLE
c_1	METZ
c_2	METZ
c_3	NANCY
c_4	METZ
..	...

CC

CJCP	CLIENT
e_I	c_I
e_2	c_I
e_3	c_2
e_4	c_3
e_5	c_3
e_6	c_3
..	..

DC

CJCP	DATE
e_I	II/I/78
e_2	II/I/78
e_3	I2/I/78
e_4	I2/I/78
e_5	I2/I/78
e_6	I3/I/78
..	...

QC

CJCP	QTE
e_I	IO
e_2	I
e_3	2
e_4	5
e_5	I00
e_6	8
..	..

CP

CJCP	PRODUIT
e_I	p_I
e_2	p_2
e_3	p_I
e_4	p_I
e_5	p_2
e_6	p_I
..	..

NP

PRODUIT	NOP
p_I	I00
p_2	I02
p_3	I03

NMP

PRODUIT	NOMP
p_I	PELLE
p_2	PIOCHE
p_3	RATEAU

PU

PRODUIT	PRIXU
p_I	I5,25
p_2	I08,00
p_3	205,30

Après passage aux identifiants (NOCL pour ①, NOCL NOP DATE pour ②, NOP pour ③) on obtient :

NCL(NOCL,NOCL)	qui se simplifie en : NCL(NOCL)	
NMC(NOCL,NOMCL)		
VC(NOCL,VILLE)		
CC(NOCL,NOP,DATE,NOCL)	" " "	: CC(NOCL,NOP,DATE)
DC(NOCL,NOP,DATE,DATE)	" " "	: DC(NOCL,NOP,DATE)
QC(NOCL,NOP,DATE,QTE)		
CP(NOCL,NOP,DATE,NOP)	" " "	: CP(NOCL,NOP,DATE)
PU(NOP,PRIXU)		
NP(NOP,NOP)	" " "	: NP(NOP)
NMP(NOP,NOMP)		

On note les importantes redondances (entre CC,DC,CP et QC) dûes au fait que l'identifiant de ② "contient" les relations PIVOINES DC,CC,CP.
Après simplification, on a :

RI(NOCL,VILLE)
R2(NOCL,NOMCL)
R3(NOCL,NOP,DATE,QTE)
R4(NOP,NOMP)
R5(NOP,PRIXU)

Soit au niveau des occurrences :

RI

NOCL	VILLE
I	METZ
3	METZ
9	NANCY
I4	METZ
..	..

R2

NOCL	NOMCL
I	TOTO
3	ARTHUR
9	JOE
I4	DUPONT
..	...

R3

NOCL	NOP	DATE	QTE
I	I00	II/I/78	IO
I	I02	II/I/78	I
3	I00	I2/I/78	2
9	I00	I2/I/78	5
9	I02	I2/I/78	I00
9	I00	I3/I/78	8
:	:	:	:

R4

NOP	NOMP
I00	PELLE
I02	PIOCHE
I03	RATEAU
:	:

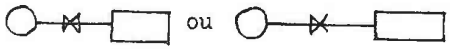
R5

NOP	PRIXU
I00	I5,25
I02	I08,00
I03	205,30
:	:

On note bien sur cet exemple le phénomène d'"applatissement" et la"disparition" des classes de repères ou de relations. Or dans l'expression relationnelle que nous souhaitons construire nous voulons que les classes de repères apparaissent clairement. Il nous faut donc abandonner les démarches classiques et nous définir des règles précises de construction d'expressions relationnelles répondant à nos voeux.

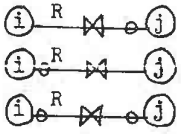
I.5.4. Les règles de construction de l'expression relationnelle d'une structure logique :

La règle fondamentale est d'associer à chaque classe de repères une relation. La clé de cette relation sera bien entendu l'identifiant de la classe de repères. Les classes de valeurs hors identifiant, rattachées par une classe de relations fonctionnelles, constituent des domaines supplémentaires de la relation :



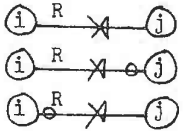
Un certain nombre de classes de relations sont déjà prises en compte "à l'intérieur" des identifiants composés. Les autres sont prises en compte de la manière suivante (les démonstrations sont immédiates ; on note R_i la relation associée au type de repères ① et C_i son identifiant) :

+ classes de relations PIVOINES I:I :

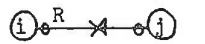


par $R_i(\underline{C_i},C_j..)$ ou par $R_j(\underline{C_j},C_i,..)$
par $R_j(\underline{C_j},C_i,..)$
par $R_i(\underline{C_i},C_j,..)$
par $R(\underline{C_i},C_j)$ ou par $R(\underline{C_j},C_i)$ (I)

+ classes de relations PIVOINES I:n :



par $R_j(\underline{C_j},C_i,..)$
par $R_j(\underline{C_j},C_i,..)$
par $R(\underline{C_j},C_i)$ (2)



par $R(\underline{C_j}, C_i)$ (3)

+ classes de relations PIVOINES n:m :

par $R(\underline{C_i}, \underline{C_j})$ (quelle que soit la totalité) (4)

par $R(\underline{C_i}, \underline{V})$ (5)

+ classes de relations PIVOINES d'arité supérieure à 2 :

par une relation de degré(*) égal à l'arité de la relation PIVOINES et comportant tous les identifiants et toutes les classes de valeurs impliquées. (6)

L'expression relationnelle comprend donc une relation par type de repères plus un certain nombre de relations correspondant aux cas notés (I),(2),(3),(4),(5) et (6) ci dessus.
Pour une structure donnée cette expression relationnelle est unique.

Exemple : construisons l'expression relationnelle de la structure représentée par la figure I.4 .

+ une relation est associée à chaque classe de repères avec son identifiant comme clé et les classes de valeurs qui lui sont fonctionnellement liées comme autres constituants:

EMPLOYE(NOE,VILLE,RUE,PAYS,....)
MARIAGE(NOE,DATE,....)
ENFANT(NOE,PNOM,AGE,.....)
MANAGER(NOE,NOT,NOB,.....)
ED(NOE,NOD,JOB,ANC,.....)
HJ(NOE,NOD,DATE,JOB,.....)
DEPARTEMENT(NOD,.....)
SERVICE(NS,.....)
PROJET(NOMPJ,BUDG,.....)
PIECE(NOP,.....)
CP(NOP/COMPOSE,NOP/COMPOSANT,QTE,....)
PCL(NOP,NOCL,.....)
LCOM(NOC,QTE,DATE,....)
CLIENT(NOCL,.....)
RELCOM(NOMR,RUE,VILLE,PAYS,....)
FOURN(NOF,.....)
PPF(NOP,NOMPJ,NOF,QTE,.....)

+ certaines relations sont complétées pour prendre en compte des accès I:I non encore exprimés :

MARIAGE(NOE/EPOUX,**NOE/EPOUSE**,DATE,.....)
MANAGER(NOE,NOT,NOB,**NOD**,.....)
CLIENT(NOCL,**NOMR**,.....)
FOURN(NOF,**NOMR**,.....)
SERVICE(NS,**NOD**,.....)

(*) le degré est le nombre des constituants (nb. de "colonnes")

LCOM(NOC,QTE,DATE,**NOP,NOCL**,....)

+ des relations supplémentaires sont ajoutées qui correspondent aux accès n:m ou d'arité supérieure à 2, ici :

ETAGE(NOD,ET)
BUREAU(NS,NOB)
EPJ(NOE,NOMPJ)
PPJ(NOP,NOMPJ)
Q-PPJ(NOP,NOMPJ,QTE)

Soit en définitive l'expression relationnelle suivante :

EMPLOYE(NOE,VILLE,RUE,PAYS)
MARIAGE(NOE/EPOUX,NOE/EPOUSE,DATE)
ENFANT(NOE,PNOM,AGE)
MANAGER(NOE,NOT,NOB,NOD)
ED(NOE,NOD,JOB,ANC)
HJ(NOE,NOD,DATE,JOB)
DEPARTEMENT(NOD)
SERVICE(NS,NOD)
PROJET(NOMPJ,BUDG)
PIECE(NOP)
CP(NOP/COMPOSANT,NOP/COMPOSE,QTE)
PCL(NOP,NOCL)
LCOM(NOC,QTE,DATE,NOP,NOCL)
CLIENT(NOCL,NOMR)
RELCOM(NOMR,RUE,VILLE,PAYS)
FOURN(NOF,NOMR)
PPF(NOP,NOMPJ,NOF,QTE)
ETAGE(NOD,ET)
BUREAU(NS,NOB)
EPJ(NOE,NOMPJ)
PPJ(NOP,NOMPJ)
Q-PPJ(NOP,NOMPJ,QTE)

A toute structure logique on peut associer une telle expression relationnelle dont les relations sont par construction en seconde forme normale. Ces expressions seront utilisées au chapitre 3 dans l'étude de la correspondance <LOG-PHYLOG>.

- - - -

I.6. Conclusion :

Outre les résultats déjà mis en évidence au cours de ce premier chapitre, on peut souligner pour conclure que le rapprochement opéré entre le modèle de PIVOINES et un modèle à base d'entités, de propriétés et d'associations doit permettre au système PIVOINES de tirer profit plus directement des études générales menées dans le domaine fondamental et délicat de la conception des structures.

- - - - -

BIBLIOGRAPHIE :

[I.1] An analysis of some constructs for conceptual model
H.A.SCHMD
IFIP-TC-2 Working conference on modelling in DBMS, Nice
(Janvier I977)

[I.2] Conceptual model as a data base design tool
P.MOULIN,J.RANDON,M.TEBOUL,H.TARDIEU,S.SAVOYSKY,S.SPACCAPIETRA
IFIP-TC-2 Work. conf. on modelling in DBMS, Freudenstadt
(Janvier I976)

[I.3] The entity-relatioship model- toward a unified view of data
CHEN P.P.
Conference on"Very large data base", Framingham,Mass.
(Septembre I975)

[I.4] Explicit description of entities and their manipulation in languages for the relational data base model
A.PIROTTE
Report R 336 MBLE
(Septembre I976)

[I.5] Concepts for the design of a conceptual schema
E.BENCI,F.BODART,H.BOGAERT,A.CABANES
Freudenstadt - IFIP
(Janvier I976)

[I.6] Entities and relationships in information
W.KENT
Nice - IFIP
(Janvier I977)

[I.7] Relations and entities
P.HALL,J.OWLETT,S.TODD
Freudenstadt - IFIP
(Janvier I976)

[I.8] The EPA model : an information-oriented data base model
A.PIROTTE
Report R343 MBLE
(Mars I977)

[I.9] Interim REport ANSI/X3/SPARC
Study Group on DBMS
ANSI
(Février I975)

[I.I0] A procedural language for expressing integrity constraints in the COEXISTENCE model
C.MACHGEELS
Freudenstadt - IFIP
(Janvier I976)

[I.II] Data semantics
J.R.ABRIAL
IFIP-TC-2 Working Conference on DBMS, Cargèse
(Avril I974)

[I.I2] CODASYL DBTG Report
ACM, New-york
(I97I)

[I.I3] Normalized data base structure : a brief tutorial
E.F.CODD
Proc. ACM-SIGFIDET
(I97I)

[I.I4] Decomposition of files, a basis for data storage and retri
IBM Report RJ I220
RISSANEN,DELOBEL
(Mai I973)

[I.I5] A uniform approach to data base management
E.FALKENBERG
Freudenstadt - IFIP
(Janvier I976)

[I.I6] Description formelle,représentation,interrogation des informations complexes-système PIVOINES
M.CREHANGE
Thèse d'Etat (Nancy I)
(Décembre I975)

[I.I7] Interrogation de données à structure complexe-système PIVOINES
M.CREHANGE
Congrès AFCET
(Novembre I976)

[I.I8] Binary logical associations in data modelling
G.BRACCHI,P.PAOLINI,G.PELAGATTI
Freudenstadt - IFIP
(Janvier I976)

[I.I9] Réflexions à propos d'une application simple du système PIVOINES
F.MEYER
D.E.A. Nancy I
(I977)

[I.20] Un modèle d'accès standard dans les SGBD
A.CABANES
Bulletin du C.B.D.,n° I6
(Juin I976)

[I.2I] Une application du système PIVOINES à la cartographie automatique
D.BARBIER
D.E.A. Nancy I
(I977)

[I.22] A relational model of data for large shared data banks
E.F.CODD
Comm. ACM I3
(Juin I970)

[I.23] A data base sublanguage founded on the relational calculus
E.F.CODD
Proc. ACM SIGFIDET
(I97I)

- - - -

CHAPITRE 2 :

L' UTILISATION DU MODELE DE PIVOINES AU NIVEAU INTERNE

2.I. Introduction ..... p. I

2.2. La correspondance entre structure externe et implémentation - généralités ..... p. 3
- 2.2.I. Les requêtes de l'utilisateur
- 2.2.2. Les étapes de la correspondance

2.3. Le mécanisme de"traduction optimisée" et ses conséquences sur PHYLOG ..... p. 7
- 2.3.I. Les étapes du mécanisme
- 2.3.2. La représentation de LOG dans PHYLOG
- 2.3.3. L'exploitation de la correspondance <LOG-PHYLOG> et le programme maquette
- 2.3.4. Les conséquences sur PHYLOG

2.4. Le modèle de PIVOINES et la prise en compte de la forme interne - la structure PHYLOG ..... p. I3
- 2.4.I. En mémoire homogène
  - 2.4.I.I. Modélisation des mémoires homogènes
  - 2.4.I.2. La prise en compte "immédiate" d' informations enregistrées en mémoire homogène
  - 2.4.I.3. Les simplifications admissibles
- 2.4.2. En mémoire hétérogène
  - 2.4.2.I. Modélisation des mémoires hétérogènes
  - 2.4.2.2. La prise en compte "immédiate" d' informations enregistrées en mémoire hétérogène
  - 2.4.2.3. Les simplifications admissibles
- 2.4.3. La structure PHYLOG et l'expression relationnelle de la structure PHYLOG (ERPHYLOG)

2.5. Illustration de la différenciation des types de repères de LOG et PHYLOG p.30

2.6. Conclusion p.36

BIBLIOGRAPHIE p.37

° ° ° °

2.I. Introduction :

Le rapport de l'ANSI/SPARC [2.I] définit la structure interne comme « une abstraction valide entre les technologies d'implémentation et la structure conceptuelle » . Il ne faut donc pas confondre le niveau interne où se place cette structure et le niveau physique de l'implémentation.

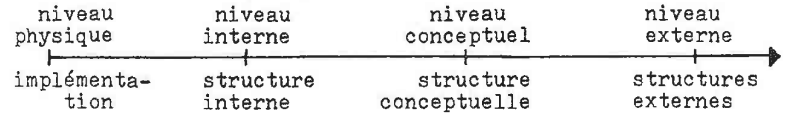


fig. 2.I.(a)

Dans le système PIVOINES, la structure PHYLOG joue ce rôle de structure intermédiaire. Elle est définie comme « la structure logique de l'organisation physique interne » ([2.2]). La terminologie de PIVOINES est rappelée sur l'axe ci-dessous qui est à rapprocher du précédent (on rappelle qu'on se place dans l'hypothèse d'un mapping direct entre niveau externe et interne; le niveau conceptuel n'est donc pas représenté).

fig. 2.I.(b)

PHYLOG est présentée comme un compromis. En effet, la proximité de la forme interne facilite le mapping <PHYLOG-forme interne> (qui est décrit au paragraphe 2.2) alors que son éloignement dégage PHYLOG des détails de l'implémentation ce qui la rend probablement plus simple et plus stable. Le problème de son positionnement sur l'axe de la figure 2.I. (en particulier vis à vis de la forme interne) n'est pas clairement résolu : ce sera l'objet principal de cette seconde partie.

Nous approfondirons notre réfléxion selon deux voies :

+ en analysant les mécanismes internes du système PIVOINES pour dégager les éventuelles exigences qui en découlent pour la structure PHYLOG.
+ en examinant l'utilisation du modèle de PIVOINES pour la représentation des informations enregistrées. Nous sommes capables, face à des implémentations, d'en donner des

descriptions "immédiates" en termes du modèle de PIVOINES, c'est à dire rendant compte de tous les détails d'une façon naturelle et "quasi photographique". Puis nous rechercherons les simplifications admissibles, gommant des détails d'implémentation sans conséquences, ce qui positionnera précisémment PHYLOG vis à vis de la forme interne. Après avoir modélisé les mémoires homogènes et hétérogènes, nous pourrons interpréter la nature des concepts de base du modèle de PIVOINES au niveau interne, en termes de ce modèle de mémoire.

Deux paragraphes complémentaires nous permettrons de définir l' "expression relationnelle" d'une structure PHYLOG (ERPHYLOG), puis d'illustrer par des exemples, des cas caractéristiques de différenciation des repères logiques vis à vis des repères de PHYLOG.

- - - - - -

2.2. La correspondance entre structure externe et implémentation - généralités :

2.2.I. Les requêtes de l'utilisateur :

Considérons un utilisateur du système PIVOINES, désireux d'accéder à certaines informations d'une base. Il dispose:

+ de la description de "sa" structure logique
+ du langage PIVOINES (cf. [2.2])

exemple : introduisons ici une structure logique simple qui nous servira dans la suite. On y distingue 5 classes de repères :

I	fournisseurs
2	pièces
3	dépôts
4	sections de dépôt
5	stock (d'une pièce dans une section)

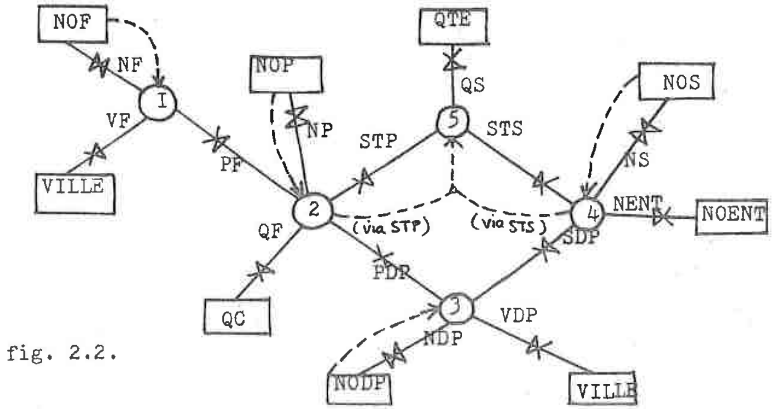


fig. 2.2.

NOF	: n°fournisseur
NOP	: n°pièce
QC	: qté à commander
NOS	: n°stock
NOENT	: n° entrée
QTE	: qté en stock
NODP	: n° dépôt

On y note en particulier que chaque pièce a un fournisseur exclusif, mais qu'elle peut être stockée dans plusieurs dépôts. Un dépôt comporte plusieurs sections où sont regroupés par affinités les stocks des différentes pièces. Dans un dépôt une pièce n'appartient qu'à une seule section: cette prescription structurelle n'apparaît pas sur cette structure logique.

Une requête utilisateur est exprimée dans le langage PIVOINES en termes de la structure logique de cet utilisateur. Rappelons

que le langage PIVOINES est de type déclaratif : la partie accès d'une requête prend la forme d'un"filtre" qui est une expression booléenne dont les termes sont appelés "motifs". Les motifs sont des requêtes "élémentaires".

exemple : soit la requête sur la structure précédente « quels sont les numéros et les quantités des pièces stockées dans le dépôt I3 et fournies par le fournisseur DUPONT dont on veut connaître également l'adresse ? ».

Cette requête peut, par exemple, prendre la forme d'un filtre comportant 2 motifs. Chaque motif peut être représenté graphiquement par un "pseudo-graphe" ou "pochoir" (cf. [2.2]):

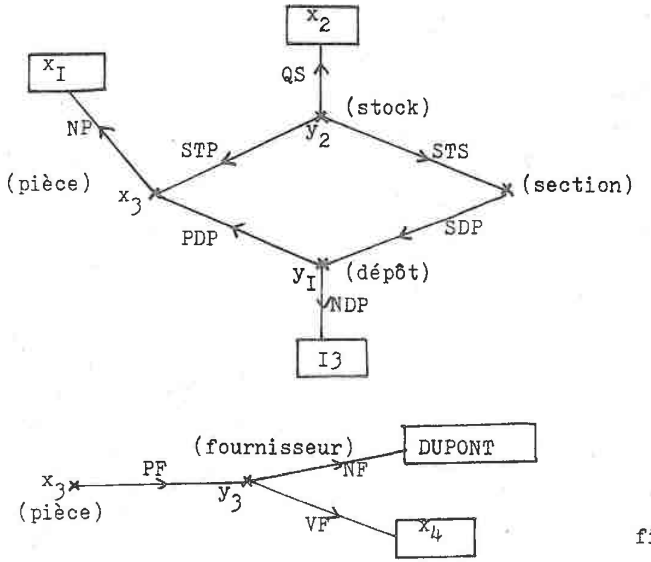
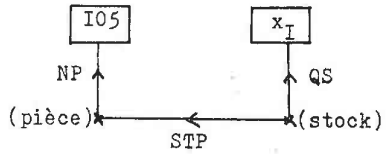


fig. 2.3.

Un tel pochoir exprime de façon très visuelle la requête, quand on la rapproche de la portion de structure concernée. Les noms des classes de repères sont entre parenthèses sur le dessin. Les x_i sont les inconnues du filtre; les y_i sont les inconnues locales des motifs; on note que dans les x_i on trouve des "inconnues-résultats" (encadrées) et des inconnues qui assurent les liens entre les motifs du filtre (ici x_3). Le langage PIVOINES permet de "linéariser" un tel pochoir. On peut écrire, par exemple:

$$F = \text{"I3"}\ NDP^{-I}\ y_I < PDP\ x_3 < NP\ x_I, STP^{-I}\ y_2 >, SDP^{-I}\ STS^{-I}\ y_2$$
$$QS\ x_2 > \underline{et}\ x_3\ PF\ y_3 < NF\ \text{"DUPONT"}, VF\ x_4 >$$

Il faut remarquer que pour une requête telle que « quelle est la quantité stockée totale de la pièce "I05" ? » il faut, en plus de la partie purement accès (cf. pochoir et filtre ci-dessous), une partie exploitation de l'accès (pour la totalisation), qui a été appelée partie "cadrage" de la requête ([2.2]).



$F = \text{"I05"}\ NP^{-I}\ STP\ QS\ x_I$

fig. 2.4.

2.2.2. Les étapes de la correspondance :

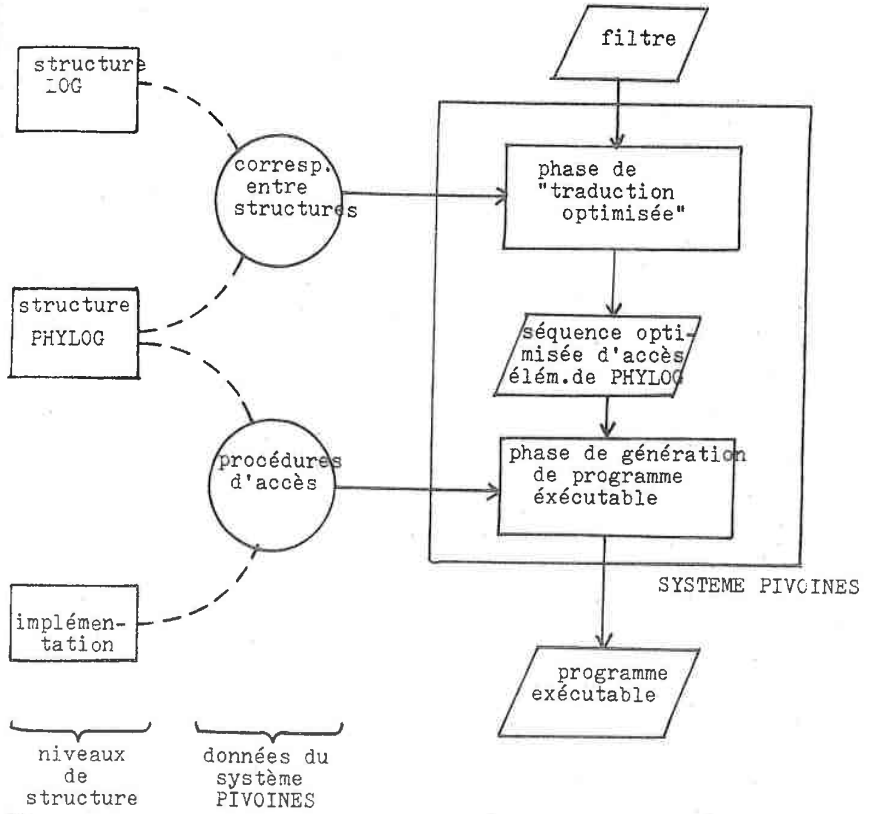
On voit bien sur ces exemples comment la requête de l'utilisateur est liée à la structure logique qu'il utilise. Or le programme qui sera généré à partir de cette requête par le système PIVOINES devra agir bien entendu sur la forme interne des informations. Se pose donc le problème de la correspondance entre structure externe et implémentation. Cette correspondance a été spécifiée en deux étapes :

+ une première étape de correspondance entre structures ; la structure PHYLOG rend compte de l'organisation logique de l'implémentation ; elle est formalisée selon le même modèle que LOG. Les requêtes sont tout d'abord traduites en termes de PHYLOG, sous forme d'une séquence optimisée d'accès élémentaires.
+ une seconde étape permet grâce à des "procédures d'accès" et à la mise en place de la gestion des résultats de passer à un véritable programme d'accès destiné à travailler sur les informations enregistrées. La figure 2.5 schématise cette démarche.

Avant d'aller plus loin dans l'étude de PHYLOG, il est nécessaire de préciser le mécanisme de "traduction optimisée" (c.à.d. de traduction avec optimisation) du système. En effet, PHYLOG est conçue pour répondre aux besoins de ce mécanisme qu'il faut donc détailler.

- - - - -

FIG. 2.5.


filtre
structure LOG
corresp. entre structures
phase de "traduction optimisée"
structure PHYLOG
séquence optimisée d'accès élém.de PHYLOG
procédures d'accès
phase de génération de programme éxécutable
SYSTEME PIVOINES
implémentation
programme exécutable
niveaux de structure
données du système PIVOINES

2.3. Le mécanisme de traduction optimisée et ses conséquences sur PHYLOG :

2.3.1. Les étapes du mécanisme :

On peut remarquer tout d'abord qu'il n'y a optimisation qu'à l'intérieur des motifs. Les différents motifs d'un filtre sont en effet exploités dans l'ordre de leur écriture de gauche à droite. Chaque motif est traité en deux temps :

+ une partie purement syntaxique, classique, mais qui aboutit à un découpage original facilitant la seconde partie (arborescences droite-gauche et gauche-droite)
+ une partie "groupement-traduction-choix" qui constitue la véritable traduction optimisée et qui sera détaillée dans la suite : elle utilise la représentation de la structure LOG dans la structure PHYLOG et aboutit au "programme maquette" du motif qui est la séquence optimisée des accès à réaliser. Le programme maquette complet est composé des programmes maquettes des motifs avec les instructions correspondant à la prise en compte de l'expression booléenne que constitue le filtre.

La figure 2.6. ,tirée de [2.2],illustre cette démarche (ainsi que son prolongement jusqu'au programme éxécutable après "intégration" des procédures d'accès).

2.3.2. La représentation de LOG dans PHYLOG :

Dans la réalisation actuelle les structures LOG et PHYLOG ne diffèrent que par le réseau des classes de relations, la "charpente" des classes de repères restant la même. La représentation de LOG dans PHYLOG se résume donc en une table de traduction qui met en présence :

+ les symboles relationnels (noms de classes de relations) de LOG et certaines combinaisons de ces symboles avec éventuellement des points intermédiaires nommés (ces différentes formes étant regroupées sous le vocable de "groupement traduisible" de LOG).
+ les traductions sous forme d'un nom de "procédure primaire" par groupement traduisible. Cette procédure primaire appelle une ou plusieurs "procédures secondaires",correspondant à des accès élémentaires de PHYLOG.

Fig. 2.6. Les étapes du mécanisme.

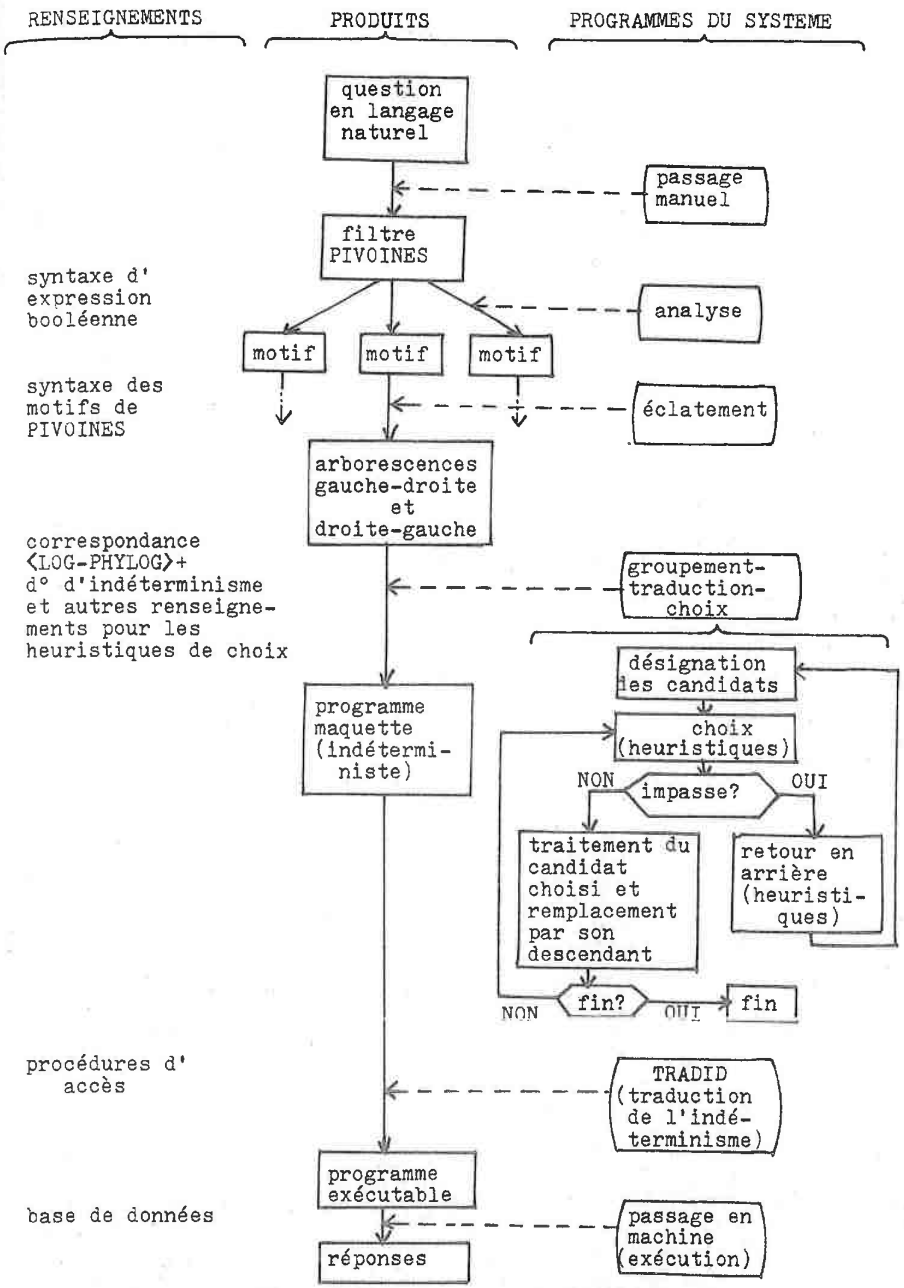


Fig. 2.7.
Table de traduction.

groupements traduisibles de LOG	traductions (PHYLOG)
.....	nom de procédure primaire
⋮	⋮

On peut montrer que théoriquement tous les symboles relationnels de LOG devraient posséder une traduction en considérant le cas limite de filtres dont tous les motifs se réduiraient à une relation unique. Il faudrait nécessairement une traduction pour chacune. Dans le système actuel des exceptions sont admises (la traduction de certaines relations de LOG prises isolémment peut être très complexe).

2.3.3. L'exploitation de la correspondance⟨LOG-PHYLOG⟩et le programme maquette :

L'exploitation de cette table de correspondance consiste à parcourir les arborescences droite-gauche et gauche-droite (c'est à dire le motif réexprimé sous une forme adaptée) à partir de leurs extrémités (c'est à dire les sommets connus du graphe du motif et des extrémités imposées par l'utilisateur éventuellement), en choisissant à chaque étape le groupement traduisible candidat le meilleur. Ce choix se fait grâce au "degré d'indéterminisme",qui est une valeur associée à chaque groupement et qui doit être minimum(la valeur 0 caractérise les fonctions qui sont déterministes; les autres accès sont caractérisés selon un barême d'indéterminisme).
Il faut remarquer que la présence de groupements traduisibles comportant plusieurs relations et l'absence possible de certaines traductions peuvent conduire à des impasses nécessitant des retours en arrière selon des heuristiques adaptées (cf. fig.2.6).
Le programme maquette d'un motif apparaît comme une suite optimisée d'appels de procédures primaires et donc indirectement comme une suite optimisée d'accès élémentaires de PHYLOG à effectuer. Il est indéterministe dans la mesure où, à chaque étape, il peut y avoir choix entre plusieurs possibilités (plusieurs occurrences de l'arc cherché) et que le programme n'a pas la possibilité de faire ce choix. Le passage au programme exécutable correspond essentiel-

lement à la mise en place de la gestion des valeurs des inconnues sous forme d'une gestion globale : toutes les occurrences sont déterminées au même moment.

exemple : à la structure logique de la figure 2.2, nous associons la structure PHYLOG suivante (construite ici comme une autre structure logique et non par référence à une implémentation précise)

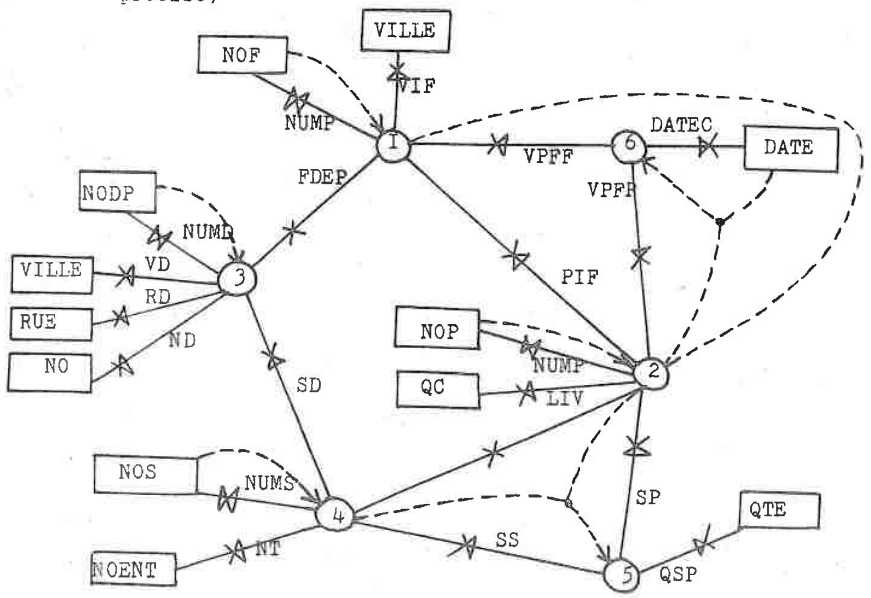


Fig. 2.8.

classe de repères:

1	fournisseurs
2	pièce
3	dépôt
4	section
5	stock
6	commande

On y note un type de repère "commande" qui n'apparaît pas au niveau logique, ainsi que des classes de valeurs supplémentaires (RUE,NO,DATE) et un réseau de classes de relations différent. La table de correspondance pourra prendre la forme suivante:

groupements traduisibles	procédures primaires	accès de PHYLOG correspondants (proc.secondaires)	d°indéterminisme et autres indications de choix
NF	RI	NUMF	...
VF	R2	VIF	...
PF	R3	PIF	...
NP	R4	NUMP	...
QF	R5	LIV	...
STP	R6	SP	...
QS	R7	QSP	...
STS	R8	SS	...
NS	R9	NUMS	...
NENT	RIO	NT	...
PDP	RII	SD^{-I}. SERP	...
SDP	RI2	SD	...
NDP	RI3	NUMD	...
VDP	RI4	VD	...
PDP^{-I}	RI5	$SERP^{-I}$. SD	...
PF^{-I}. PDP^{-I}	RI6	FDEP	...
PDP . PF	RI7	$FDEP^{-I}$	...
STS^{-I} . STP	RI8	SERP	...
STP^{-I} . STS	RI9	$SERP^{-I}$	...

2.3.4. Les conséquences sur PHYLOG :

Les considérations précédentes nous conduisent à trois types de remarques :

a) une remarque générale sur les possibilités assez restreintes de différenciation entre les structures LOG et PHYLOG : en dehors de l'aspect fondamental de"sous-schéma" (au sens de sous-ensemble) que possède LOG vis à vis de PHYLOG, seuls les cheminements peuvent différer. Au paragraphe 2.5 nous illustrerons par des exemples,la grande souplesse qu' apporte la faculté de différencier les types de repères entre les deux niveaux.

b) la seconde remarque est liée au coeur même de cette étude, c'est à dire au cas où des repères logiques n'ont pas de correspondant direct dans PHYLOG. On voit que, pour pouvoir conserver le processus précédemment décrit, il est indispensable que les procédures associées à des groupements traduisibles ayant pour but ces types de repères délivrent des valeurs associées chacune à "quelque chose dans PHYLOG". Ces "quelque choses" seront appelés les "objets équivalents" (O.E.) des repères logiques.

c) la troisième remarque concerne la nature même de la stru-

cture PHYLOG. De quelle manière intervient-elle dans le processus décrit ci-dessus ? D'une part par les traductions qui correspondent à des chemins dans PHYLOG et d'autre part par les critères de choix (d° d'indéterminisme et autres) qui les caractérisent. Ces deux catégories d' informations suffisent au choix d'une stratégie d'accès dont la qualité sera fonction essentiellement de la qualité des critères de choix.

On peut immédiatement constater que cela ne nous apporte aucune contrainte impérative quant à la définition de la structure PHYLOG, vis à vis de l'implémentation. En effet le système peut ainsi fonctionner autant avec une structure PHYLOG décrivant dans ses moindres détails l'organisation physique des données qu'avec une structure PHYLOG plus éloignée encore que la structure logique de l'implémentation !

Le choix de PHYLOG n'est pas pour autant indifférent : l' efficacité du système PIVOINES se mesurera pour une bonne part en fonction de l'efficacité des programmes générés, elle même étant directement liée à l'efficacité des procédures d'accès élémentaires associées à chaque accès de PHYLOG. Or,une structure PHYLOG mal choisie et trop éloignée de l'implémentation impliquera sans doute des procédures compliquées et peu efficaces. Quelle que soit la valeur de l'optimisation dans le choix de l'ordre d'appel des procédures (qui est la forme d'optimisation étudiée jusqu'à présent- cf. 2.2), le résultat ne sera pas bon dans ce cas.Il semble logique que PHYLOG rende compte des grandes lignes de l'implémentation (regroupements d'informations, accès physiques existants,....),à l'exclusion des "détails" sans importance, afin que les procédures ne réalisent pas une seconde traduction cachée.

Le paragraphe suivant va s'attacher à caractériser la prise en compte "immédiate" (c'est à dire de façon naturelle et dans tous ses détails) d'une implémentation ainsi que les simplifications admissibles de cette prise en compte immédiate. Nous définirons ainsi avec plus de précision le positionnement de PHYLOG vis à vis de la forme interne.

- - - - -

2.4. le modèle de PIVOINES et la prise en compte de la forme interne- la structure PHYLOG :

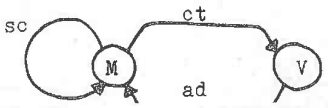
2.4.I. En mémoire homogène :

2.4.I.I. Modélisation des mémoires homogènes :

Le but de cette modélisation est de nous fournir un langage de référence et des notations précises pour parler des mémoires homogènes, c'est à dire de la mémoire centrale réelle ou virtuelle. Elles peuvent être décrites dans les termes suivants (cf.[2.4,2.3]) :

+ un ensemble M de "mots" (ou unités d'adressage)
  $M = \{m_1, m_2, \ldots\ldots, m_n\}$
+ un ensemble V de valeurs susceptibles d'être contenues dans les mots
  $V = \{v_1, v_2, \ldots\ldots, v_q\}$
  (ex: ensemble des suites de 32 bits si la capacité des mots est de 32 bits)
+ un sous-ensemble A de V, ou ensemble des adresses, en relation biunivoque avec l'ensemble des mots
  $A = \{a_1, a_2, \ldots\ldots, a_n\}$
  (ex: sous-ensemble de V,dont les éléments ont les I6 premiers bits à 0).
+ trois fonctions élémentaires:
  - "adressage" ad : $A \longrightarrow M$
    (à une adresse a A, correspond un et un seul mot m = ad(a) et réciproquement par ad^{-1}).
  - "contenu" ct : $M \longrightarrow V$
    (à un instant donné le contenu d'un mot ct(m) est unique)
  - "succession" sc : $M \longrightarrow M$
    (qui réalise un ordre sur les mots M; comme les adresses sont en général les entiers de 0 à Card(M)-I, sc(m) est définie par : sc(m) = ad(a+I) si m = ad(a)).

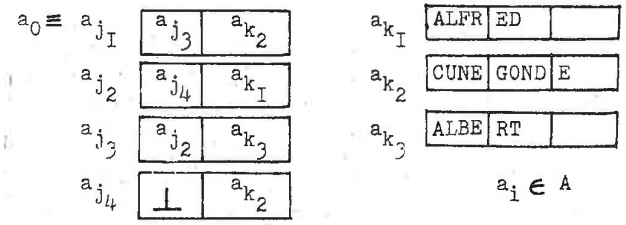
fig. 2.9



2.4.I.2. La prise en compte "immédiate" d'informations enregistrées en mémoire homogène :

Considérons des informations enregistrées en mémoire homogène. Elles peuvent naturellement et directement être décrites en termes du modèle de PIVOINES (repères, valeurs, relations) au niveau des occurrences, d'une manière que nous analyserons au vu de quelques exemples simples caractéristiques.

EXEMPLE I : soit une liste chaînée de pointeurs vers les éléments d'un ensemble E, qui prend ses valeurs dans V^3.



(liste chaînée) (éléments de E)

Cette information peut être décrite d'une manière "immédiate" par :

- des repères $r_{j_I}, r_{j_2}, \ldots\ldots$ (repères de pointeurs)
- des repères $r_{k_I}, r_{k_2}, \ldots\ldots$ (repères d'éléments de E)
- les valeurs de E
- une relation de succession entre repères de pointeurs, s : $s(r_{j_I}) = r_{j_3}$ $s(r_{j_2}) = r_{j_4}$

  $s(r_{j_3}) = r_{j_2}$ $s(r_{j_4}) = \perp$ ("n'existe pas")
- une relation d'accès des repères de pointeurs aux repères d'éléments de E, r :

  $r(r_{j_I}) = r_{k_2}$ $r(r_{j_2}) = r_{k_I}$ $r(r_{j_3}) = r_{k_3}$

  $r(r_{j_4}) = r_{k_2}$
- une relation d'accès aux valeurs de E depuis les repères d'éléments de E, v :

  $v(r_{k_I})$ = ALFRED $v(r_{k_2})$ = CUNEGONDE

  $v(r_{k_3})$ = ALBERT

Soit: $r_0 \equiv r_{j_I}$

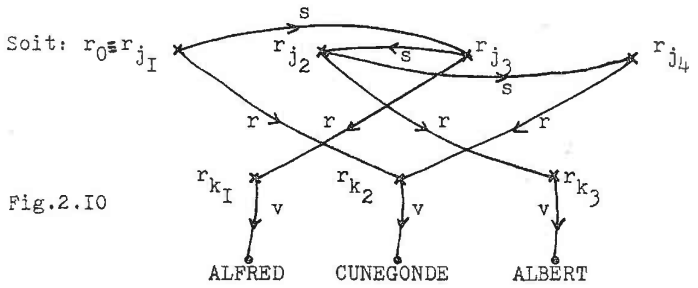


Fig.2.IO

On remarque que dans cette description les repères correspondent soit à des mots soit à des champs élémentaires formant un tout et constitués de plusieurs mots consécutifs :

r_{k_2} correspond au champ élémentaire constitué par les mots d'adresse $a_{k_2}, a_{k_2}+I, a_{k_2}+2$

Les relations d'accès correspondent ici à des accès physiquement matérialisés par des pointeurs (cf. r,s) ou à des accès aux valeurs des champs (v). Elles peuvent facilement être explicitées à l'aide des fonctions élémentaires du modèle de mémoire en remplaçant les repères par les mots (ou leurs adresses) qui correspondent :

s : soit $m_{j_i} = ad(a_{j_i})$ $s(m_{j_i}) = ad(ct(m_{j_i}))$

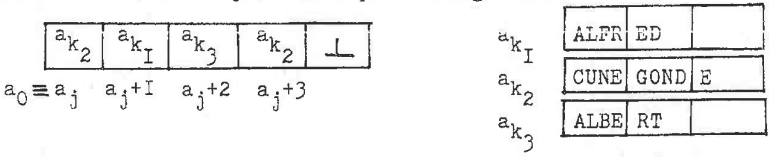
v : soit $m_{k_i} = ad(a_{k_i})$ $v(m_{k_i}) = ct(m_{k_i}) \,||\, ct(cs(m_{k_i})) \,||\, ct(sc^2(m_{k_i}))$

(où || représente la concaténation des valeurs)

r : $r(m_{j_i}) = ad(ct(sc(m_{j_i})))$

Les procédures correspondantes seront très simples.

EXEMPLE 2: il s'agit de la même information qu'à l'exemple I, mais avec une liste de pointeurs par contigüité :

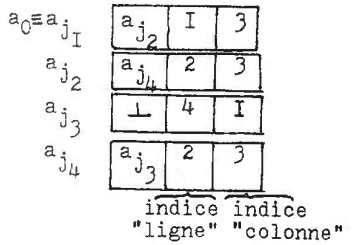


La description en termes de PIVOINES peut être la même que dans l'exemple I. La différence de nature de s (contigüité physique à la place de chaînage) ne transparaissant pas.

Il faut remarquer la différence entre la contigüité des mots d'

adresse $a_j, a_j+I, a_j+2, \ldots$ interprétée comme un accès par contigüité entre des champs élémentaires distincts (et donc des repères distincts) et la contigüité physique des mots d'adresse $a_{k_I}, a_{k_I}+I, a_{k_I}+2, \ldots$ à l'intérieur d'un même champ élémentaire (caractérisé donc par un repère unique). Il s'agit d'un choix de nature sémantique.

EXEMPLE 3: soit une liste chaînée d'indices d'un tableau de nombres (4x3) représenté ligne par ligne de façon contigüe



a_k	II	-40	32
a_k+3	30	28	I7
a_k+6	46	6I	20
a_k+9	I3	-2I	-56

La description "immédiate" peut être la suivante :

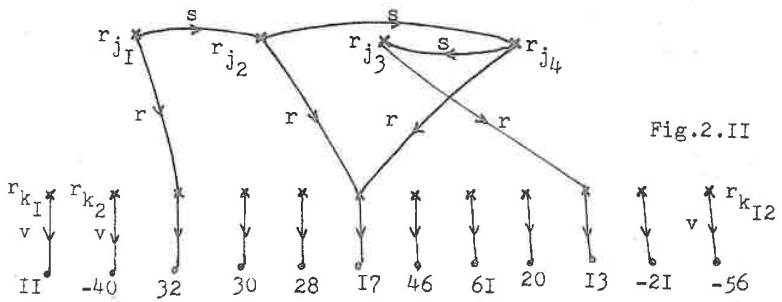


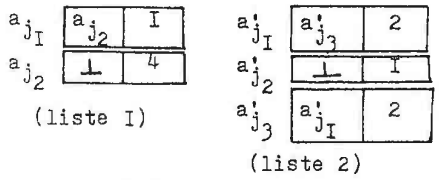
Fig.2.II

On remarque qu'ici la relation r correspond à un calcul d'adresse à base des indices de ligne et de colonne du tableau :

soit $m_{ji} = ad(a_{j_i})$

$r(m_{ji}) = ad(a_k + (ct(sc(m_{ji})-I)x3 + ct(sc^2(m_{ji}))-I)$

EXEMPLE 4: il s'agit du même tableau que précédemment mais dont les indices sont rangés dans deux listes distinctes. (l'accès au tableau étant possible à partir de n'importe quel couple d'éléments des 2 listes).



a_k	II	-40	32
a_k+3	30	28	I7
a_k+6	46	6I	20
a_k+9	I3	-2I	-56

La description "immédiate" peut être la suivante :

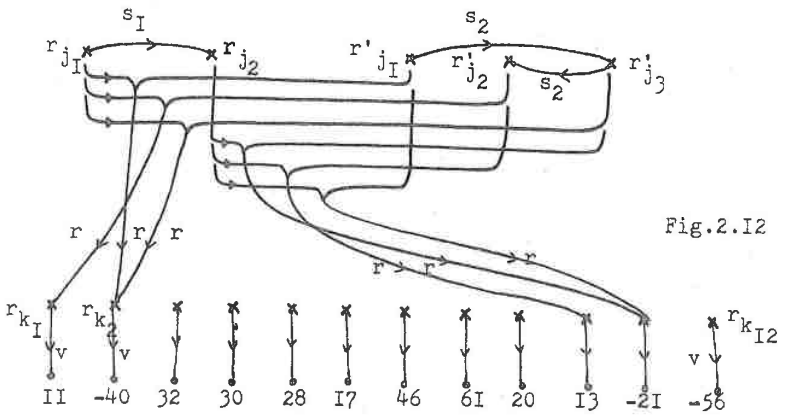


Fig.2.I2

La relation r est ici d'arité 3: elle est définie pour tous les couples de repères (r_{j_i}, r'_{j_i}).

soit $m_{ji} = ad(a_{j_i})$ et $m'_{ji} = ad(a'_{j_i})$

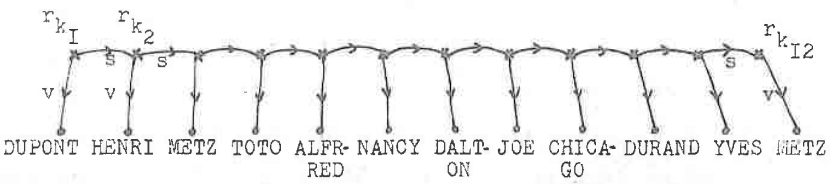
$r(m_{ji}, m'_{ji}) = ad(a_k + (ct(sc(m_{ji}))-I)x3 + ct(sc(m'_{ji}))-I)$

EXEMPLE 5: soit une liste contigüe comportant les noms, prénoms et domiciles de personnes. Chaque valeur appartient à V^2.

a_k	DUPO	NT	HENR	I	METZ	
a_k+6	TOTO		ALFR	ED	NANC	Y
a_k+I2	DALT	ON	JOE		CHIC	AGO
a_k+I8	DURA	ND	YVES		METZ	

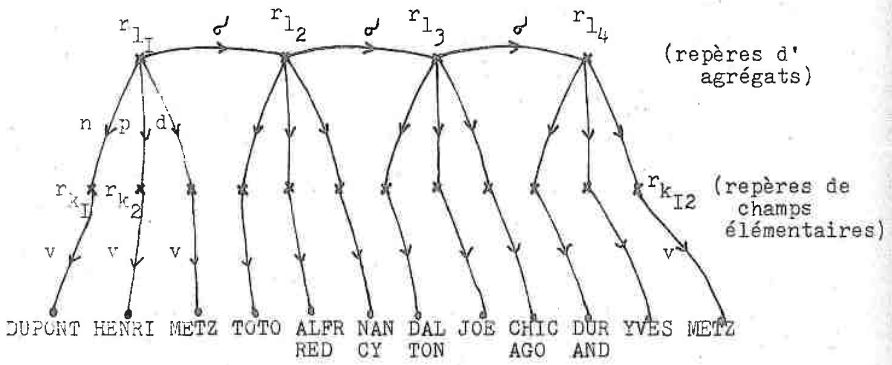
La description en liste simple n'est pas satisfaisante :

Fig.2.I3



Elle ne rend absolument pas compte du lien sémantique qui lie le nom, le prénom et le domicile d'une même personne. On peut en rendre compte en introduisant la notion d' "agrégat de champs élémentaires", qui est un ensemble de champs élémentaires contigüs constituant un tout (d'un point de vue sémantique). Cette définition ne retient que la possibilité d'agrégats par contigüité, ce qui nous semble une interprétation "naturelle" d'une implémentation. La description "immédiate" serait alors la suivante :

Fig.2.I4



(on peut bien entendu avoir plusieurs niveaux d'agrégation des champs élémentaires).

Au vu de ces exemples, qui couvrent les principales structures et méthodes d'accès en mémoire homogène (cf §2.4), on peut dire que l'interprétation directe et naturelle en termes du modèle de PIVOINES consiste à :

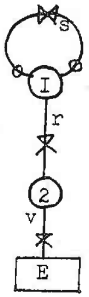
+ associer des repères aux champs élémentaires ou aux agrégats de champs élémentaires que l'on peut discerner au sein de l'information
+ de rendre compte par des relations des accès physiquement matérialisés entre ces objets internes (champs et agrégats). Ces accès se font par contigüité physique, par chaînage ou par calcul d'adresse : ce qui n'apparaît pas dans la description en termes du modèle de PIVOINES. Certains accès peuvent être d'arité supérieure à 2.

Il s'agit bien d'une interprétation de nature sémantique (sélection des champs et des agrégats et de certains accès, par contigüité en

particulier).
Cette description "immédiate" gomme déjà un certain nombre de détails de l'implémentation (nature des accès, forme interne des valeurs,...). Nous présenterons dans la suite d'autres formes de simplifications. Il est bien entendu également possible de donner une représentation au niveau des classes (classes de repères, de relations et de valeurs).

exemple: à la figure 2.I0 on peut faire correspondre le schéma de structure suivant : (les notations sont celles introduites en I.4.2)

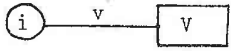
Fig.2.I5



classes de repères :
① repères de pointeurs
② repères d'éléments de E

2.4.I.3. Les simplifications admissibles :

a) dans la représentation "immédiate" toute valeur est attachée à un repère qui correspond au champ élémentaire (mot ou ensemble de mots) la contenant . Au niveau des classes, on a :



{ ① classe des repères de champs élémentaires
v fonction d'accès à la valeur

On peut sans inconvénients passer sous silence cette classe de repères, si elle ne joue que ce seul rôle.
exemple: la figure 2.I5 se simplifie alors en :

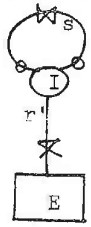


Fig.2.I6

L'interêt de cette simplification, comme celui de beaucoup des suivantes, est :

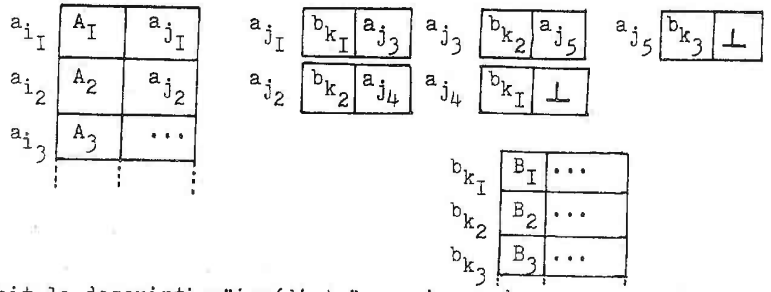
+ le remplacement de 2 procédures d'accès élémentaires par une seule, ce qui minimise le nombre d'appels de procédures

et augmente donc l'efficacité

+ la suppression d'un type de repères qui permet une gestion des résultats plus souple et donc un gain d'espace mémoire. En effet dans le cas de la figure 2.I5, la procédure associée à r doit déterminer et stocker tous les repères du type 2 qui serviront d'argument à la procédure associée à v. Dans le cas de la figure 2.I6 la gestion est plus souple car interne à la procédure associée à r' .

b) dans la représentation "immédiate" les relations n:m sont obligatoirement décomposées en relations fonctionnelles avec des repères supplémentaires.

exemple: soit une liaison n:m entre 2 ensembles de valeurs A et B matérialisée à l'aide d'une liste chaînée.



Soit la description"immédiate" au niveau des occurrences :

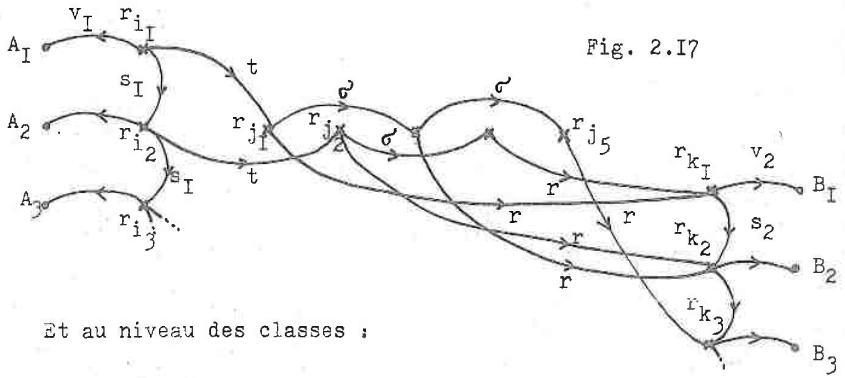
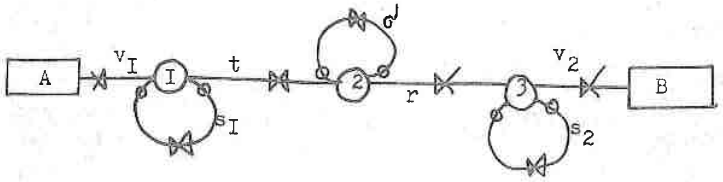


Fig. 2.I7

Et au niveau des classes :



Les repères associés à la liste chaînée n'ont pas en général d'interêt par eux-mêmes. On peut donc réintroduire les relations n:m dans la représentation PIVOINES (avec les mêmes avantages qu'au paragraphe a).

exemple : la figure 2.I7 se simplifie alors en :

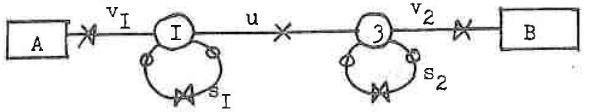
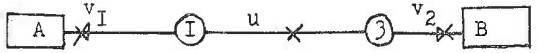


Fig.2.I8

Ces deux premières simplifications consistent à ne pas représenter des éléments de la forme interne que nous appellerons "éléments techniques ou système", car ils relèvent de la nature du système de mémorisation ou des techniques d'implémentation et non de la logique propre à l'implémentation considérée. On peut en citer d'autres exemples : chaînages inverses pour accélérer certains accès, chaînage des emplacements libres dans une liste pour faciliter les modifications, etc.....

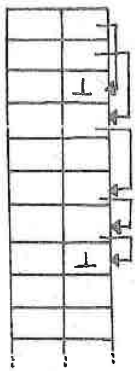
c) au niveau des notations, on peut considérer que la relation triviale de succession entre repères d'une même classe (par contigüité,par chaînage,.....) peut être sous-entendue.

exemple : les relations s_1 et s_2 de la fig. 2.I8. peuvent être sous-entendues



Il peut exister des successions multiples; une seule est alors implicite.

exemple: une liste contigüe avec des sous-listes chaînées.



Dans ce cas il semble naturel de rendre explicite la succession par chaînage, celle par contigüité étant implicite :

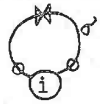


Fig. 2.I9.

En conclusion, en mémoire homogène, les repères correspondent aux champs élémentaires ou aux agrégats de champs à l'exclusion des zones "techniques" ou "système" sans interêt quant à la logique de l'implémentation. Les relations correspondent à des accès entre champs ou agrégats matérialisés par un ou plusieurs accès physiques par contigüité, chaînage ou calcul. La structure PHYLOG,ainsi conçue, comprend donc toutes les catégories de relations (I:I,I:n,n:m,d'arité égale ou supérieure à 2). Elle reste très proche de l'implémentation, dont elle reflète en détail l'organisation.

2.4.2. En mémoire hétérogène :

2.4.2.I. Modélisation des mémoires hétérogènes : cette modélisation nous permet de définir un vocabulaire commun aux différents types de mémoires auxilliaires et à la mémoire centrale.
Une mémoire hétérogène est un ensemble de mémoires de caractéristiques différentes liées (M.C.,disques,autres mémoires adressables ou non). Nous considèrerons que globalement cet ensemble constitue un espace mémoire unique, linéaire, dont les unités élémentaires de mémorisation , que l'on nommera des "mots", sont de types différents suivant les zones dont ils font partie. Les zones de mémoires secondaires sont scindées en volumes.

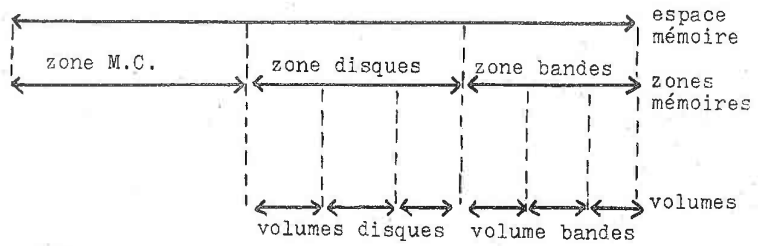
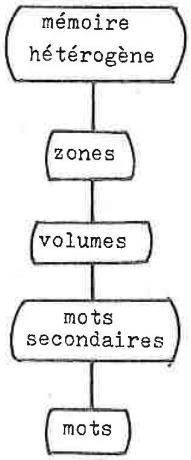


Fig. 2.20

Chaque volume de mémoire secondaire comporte un ensemble d'unités naturelles d'accès ou "mots secondaires" (MS = $\{ms_i\}$). Ces mots secondaires sont soit accessibles directement (secteurs ou pistes des disques sectorisés ou non etc...) soit accessibles séquentiellement(enregistrements physiques sur bandes etc....). Les mots secondaires sont eux-mêmes décomposables en "mots" consécutifs non adressables (octets par exemple):

Fig. 2.2I



Une mémoire hétérogène peut donc être caractérisée par (cf. [2.3]) :

+ un ensemble M de "mots" $\{m_1, m_2, \ldots\ldots, m_n\}$ .Certains de ces mots sont adressables,d'autres non.
+ un ensemble MS de "mots secondaires" $\{ms_1, ms_2, \ldots\ldots, ms_p\}$. Certains sont adressables,d'autres non.
+ un ensemble V des valeurs susceptibles d'être contenues dans les mots $\{v_1, v_2, \ldots\ldots, v_q\}$.
+ un sous-ensemble A de V, ou ensemble des adresses $\{a_1, a_2, \ldots\ldots, a_r\}$ (avec l'hypothèse simplificatrice que toute adresse en mémoire centrale ou en mémoire secondaire peut être contenue dans un mot).
+ 7 fonctions élémentaires :
  - "adressage" ad : A ⟶ M
  - "adressage secondaire" ad_2 : A ⟶ MS
    (adressage par secteur,par piste,...)
  - "contenu" ct : M ⟶ V
  - "premier mot" prm : MS ⟶ M
    ($prm(ms_i) = m_{i_{min}}$ ou $m_{i_{min}}$ est le mot sans prédécesseur de ms_i : $m_{i_{min}} \rightarrow$ mot secondaire ms_i)
  - "succession" sc : M ⟶ M
  - "succession secondaire" sc_2 : MS ⟶ MS
  - "début volume non adressable"
    dv : A ⟶ MS
    (qui associe à une adresse de volume non adressable, le premier mot secondaire du volume).

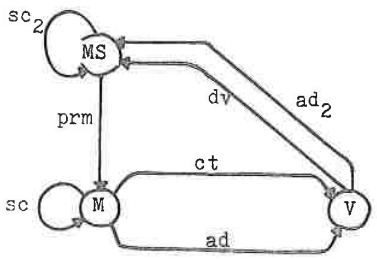


Fig. 2.22

2.4.2.2. La prise en compte "immédiate" d'informations enregistrées en mémoire hétérogène :

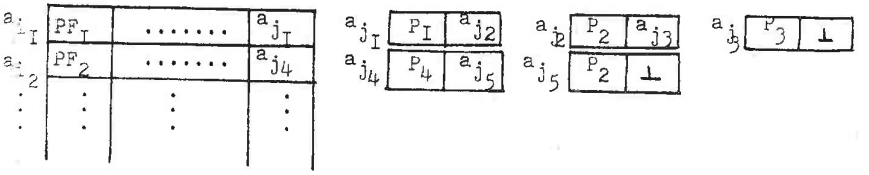
Considérons des informations enregistrées en mémoire hétérogène, c'est à dire partiellement utilisées à travers des systèmes de gestion de fichiers plus ou moins évolués (nous avons en effet rattaché le cas des mémoires secondaires adressables utilisées en mémoire virtuelle au cas des mémoires homogènes et nous ne considèrerons pas le cas où l'on n'utilise pas de système de gestion de fichiers).

exemple : considérons le cas simple d'une liste chaînée en mémoire centrale qui associe à un produit finis ses éléments composants. Chaque composant fait référence (par une clé) à un article d'un fichier "pièces" en séquentiel indexé. Chaque enregistrement du fichier"pièces" comporte un matricule fournisseur qui permet d'accéder à un enregistrement du fichier "fournisseur" (en organisation calculée).

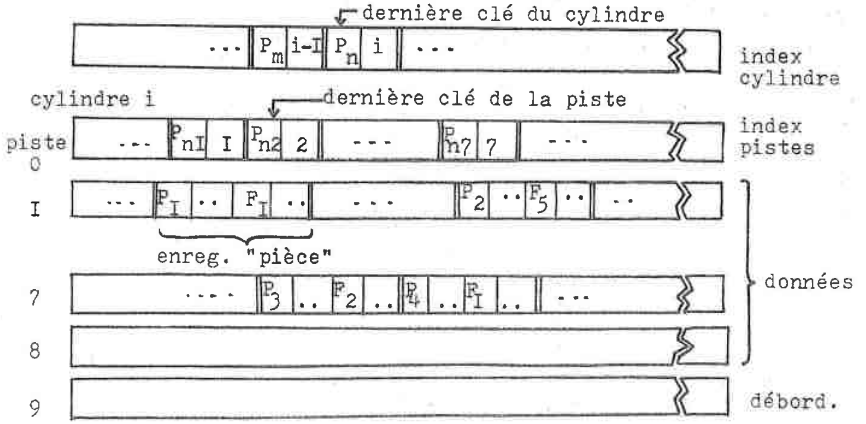
Les mémoires secondaires sont des disques non sectorisés (mot secondaire = piste).

La figure 2.23 représente cette information.

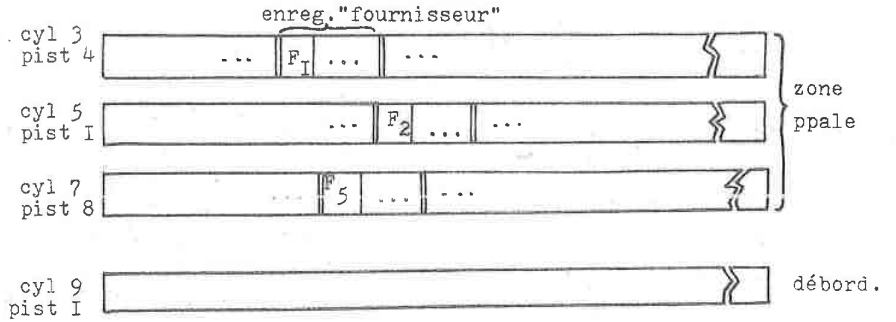
Fig. 2.23



volume v_1 : fichier "pièces"



volume v_2 : fichier "fournisseurs"



$\{P_i$ clés du fichier "pièces"
$\{F_i$ matricules "fournisseurs"

on suppose ici :
$P_m < P_i \leq P_n$
et: $P_I, P_2 \leq P_{nI}$
$P_{n6} < P_3$, $P_4 \leq P_{n7}$

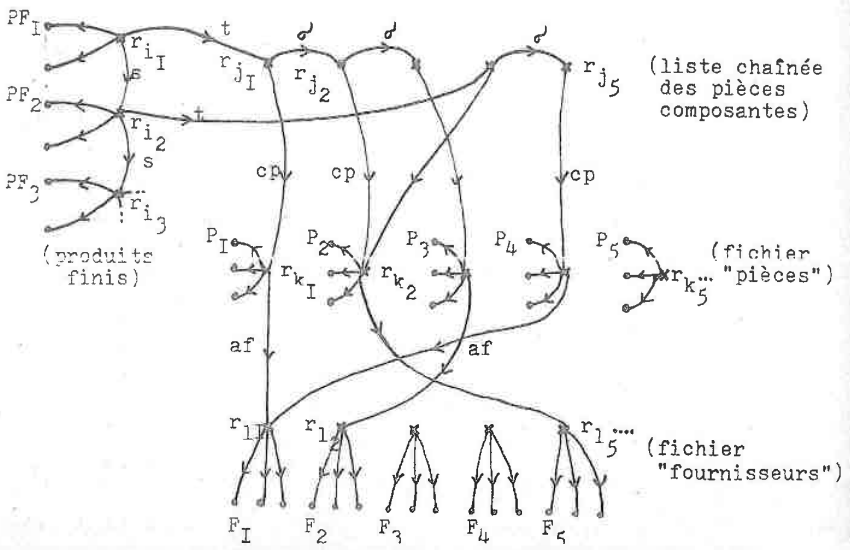
Il serait possible de donner une représentation "immédiate" en termes de PIVOINES de cette information : elle décrirait en détail toutes les zones et leurs liaisons, comme par exemple les différents niveaux d'index du fichier "pièces".
Il est clair qu'il s'agirait d'un effort inutile, car les relations ainsi mises en évidence ne pourraient être utilisées ainsi : cela équivaudrait à reécrire par morceaux des procédures complexes qu'offrent les systèmes de gestion de fichiers.
Cet exemple simple illustre bien la nécessité qu'il y a de prendre en compte les moyens système qui accompagnent les mémoires secondaires. Prendre en compte ces moyens équivaut à apporter de très importantes simplifications à la représentation "immédiate". Les informations sont vues "à travers" un certain système de gestion.

2.4.2.3. Les simplifications admissibles :

En ce qui concerne les zones de mémoire secondaire ces simplifications sont importantes et liées à l'environement système qui accompagne les données.

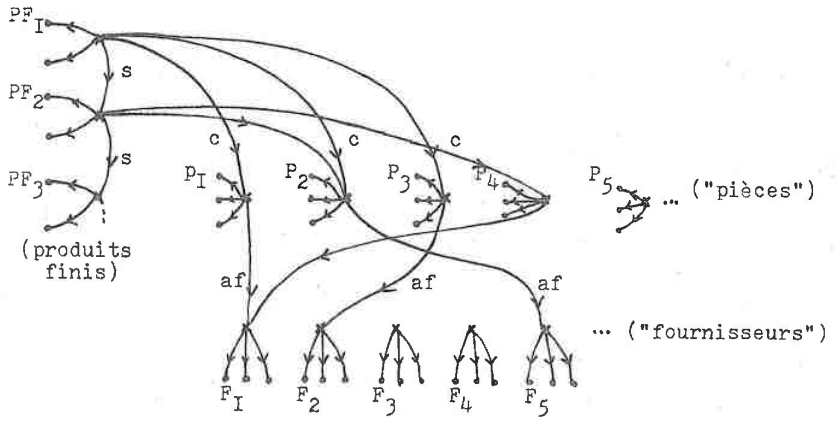
Exemple: dans l'exemple du paragraphe précédent, un enregistrement du fichier "pièces" (ou "fournisseurs") sera représenté comme un agrégat de champs élémentaires, caractérisé par un repère et accessible par une relation PIVOINES qui recouvre tout le mécanisme complexe d'accès géré par le système de gestion de fichiers (accès par tables d'index ou calculé).

Fig. 2.24



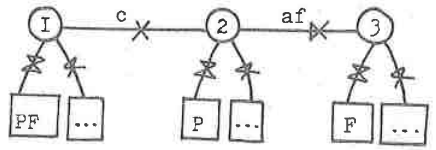
Ce schéma peut être simplifié en réintroduisant la relation n:m qui lie les produits finis et les pièces, ce qui efface la liste chaînée matérialisant cette relation (cf.2.4.I.3.b).

Fig.2.25



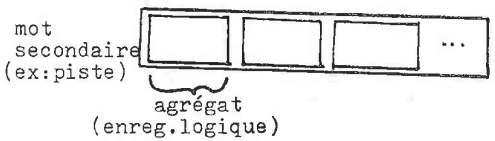
Soit le schéma au niveau des classes : (la classe de relations s, de succession entre les repères de produits finis est sous-entendue-cf.2.4.I.3.c).

Fig. 2.26



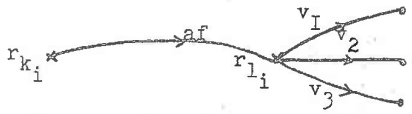
classe de repères:
- I : produits finis
- 2 : pièces
- 3 : fournisseurs

L'interprétation des concepts de PIVOINES en termes du modèle de mémoire est moins immédiate pour les zones de mémoire secondaire que pour la mémoire centrale. On peut noter avec [2.4]que la notion de fichier en mémoire secondaire « recouvre en fait la notion de liste(fichiers séquentiels), de table(fichiers en rangement calculé), de table ordonnée(fichiers à organisation séquentielle indexée) » .Leur prise en compte en termes de PIVOINES n'est donc pas un problème spécifique. Les repères correspondent aux agrégats de champs élémentaires (enregistrements logiques), qui sont des ensembles de champs consécutifs accessibles en séquence depuis le début du mot secondaire qui les acceuillent:



Les relations d'accès correspondent à des mécanismes d'accès complexes perçus à travers un système de gestion de fichiers (tables d'index hiérarchisées, chaînages vers des zones de débordement....)
On peut citer comme autre exemple de simplification, la non prise en compte explicite des fichiers inverses dans une représentation en termes de PIVOINES: il ne s'agit en effet que d'un mécanisme d'accélération de la sélection, depuis une classe de valeurs, des repères associés à un type d'enregistrements donné.
On peut donc conclure en notant que la représentation en termes de PIVOINES est beaucoup plus "éloignée" de l'implémentation pour les zones de mémoire secondaire que pour les zones de mémoire centrale.

Remarque : il sera nécessaire de tenir compte du fait que les procédures des systèmes de gestion de fichiers réalisent en général,l'accès à l'ensemble des valeurs d'un enregistrement, qui sont disponibles dans un buffer désigné par l'utilisateur.
ex:dans l'exemple précédent il y aurait prise en compte simultannée de af et des relations d'accès aux valeurs v_I, v_2,v_3.



Il semble donc peu naturel et peu efficace d'associer des procédures individualisées à af,v_I,v_2 et v_3, comme le système PIVOINES nous contraint à le faire jusqu'à présent.

2.4.3. La structure PHYLOG et l'expression relationnelle de la structure PHYLOG (ERPHYLOG) :

Quel que soit le type de mémoire, les repères de PHYLOG correspondent à des agrégats homogènes de zones à l'exclusion des zones purement techniques ou système. Les relations de PHYLOG correspondent aux accès existant physiquement entre ces agrégats: ils sont élémentaires en mémoire centrale et plus complexes en mémoire secondaire(accès via un système de gestion de fichiers).

Un schéma au niveau des classes ainsi conçu (cf.fig.2.26) est tout à fait semblable à un schéma de structure logique : on y retrouve des classes de repères avec leurs classes de valeurs "accrochées en étoile" et des classes de relations d'arité et de cardinalité quelconque qui les joignent.
Il nous sera donc possible, en reprenant les règles du paragraphe I.5.4., de construire une "expression relationnelle" d'une structure PHYLOG (ERPHYLOG). Ces expressions relationnelles nous seront utiles au chapitre 3, dans l'étude de la correspondance LOG-PHYLOG .
Exemple : expression relationnelle associée à la structure de la figure 2.26

PRODUIT(PF,....) correspondant à (1)
PIECE(P,F,.....) " " " (2)
FOURNISS(F,....) " " " (3)
C(PF,P) " " " la relation n:m c .

- - - - -

2.5. Illustation de la différenciation des types de repères de LOG et PHYLOG :

Nous présenterons tout d'abord un certain nombre de cas typiques (limités à la définition d'un type de repères logiques dérivé d'un ou plusieurs types de repères de PHYLOG).

Exemple 1:

au niveau PHYLOG : type de repères "employé" avec une classe de valeurs "catégorie d'emploi"

au niveau LOG : type de repères "cadre"

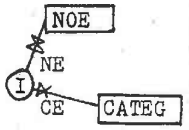


Fig. 2.27

Il s'agit de la simple restriction du type de repères de PHYLOG selon une valeur donnée d'une de ses classes de valeurs (ici : CATEG = "cadre").

Ce procédé se généralise à des restrictions beucoup plus complexes.

Exemple 2:

au niveau PHYLOG :

types de repères:
- I : fournisseur
- 2 : pièce
- 3 : fourn.-pièce avec la qté à livrer

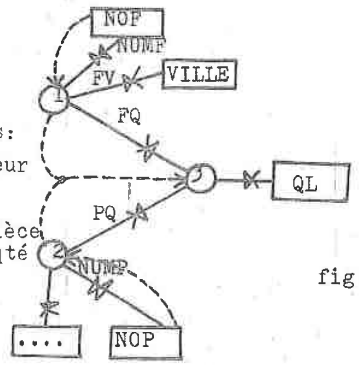


fig. 2.28(a)

au niveau LOG:

types de repères:
- I : pièce
- 2 : ville (localisation pièce)

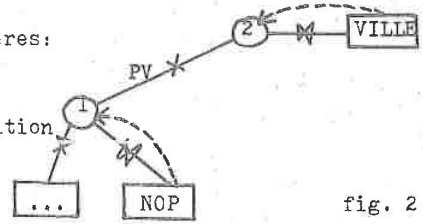


fig. 2.28(b)

Il s'agit de la "projection" du type de repères de PHYLOG sur une de ses caractéristiques (ici, la ville).Cette structure peut être pratique pour un utilisateur essentiellement intéressé par des problèmes de localisation (villes ou l'on trouve la pièce x ? pièces que l'on trouve dans la ville y ?).

Exemple 3:

au niveau PHYLOG:

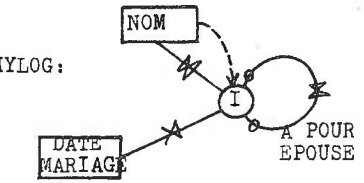


fig. 2.29(a)

(ou valeur nulle)

type de repères:
I : personnes

au niveau LOG :

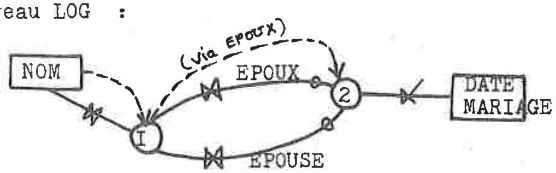


fig. 2.29(b)

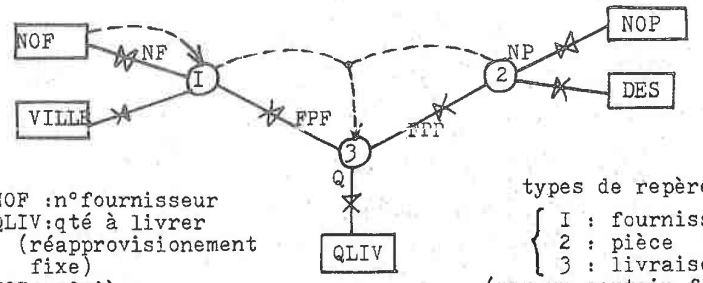
NOM : nom ou nom de jeune-fille pour les femmes mariées (on suppose que 2 personnes n'ont jamais le même nom)

types de repères:
- I : personnes
- 2 : mariage

Le type de repères (2) de LOG correspond à l'interprétation par un utilisateur du mariage en tant qu'entité (traduite par un repère) et non en tant qu'association (traduite par une relation PIVOINES). On pourrait tout aussi bien inverser LOG et PHYLOG et considérer le cas de repères de mariage au niveau PHYLOG n'apparaissant pas au niveau logique: il n'y aurait pas dans ce cas de problème de correspondance entre classes de repères et nous savons déjà traiter un tel cas.

Exemple 4 :

au niveau PHYLOG: fig. 2.30(a)

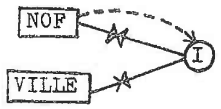


- NOF :n°fournisseur
- QLIV:qté à livrer (réapprovisionement fixe)
- NOP :n°pièce
- DES : désignation

types de repères:
- I : fournisseur
- 2 : pièce
- 3 : livraison

(par un certain fournisseur d'une certaine pièce)

au niveau LOG :

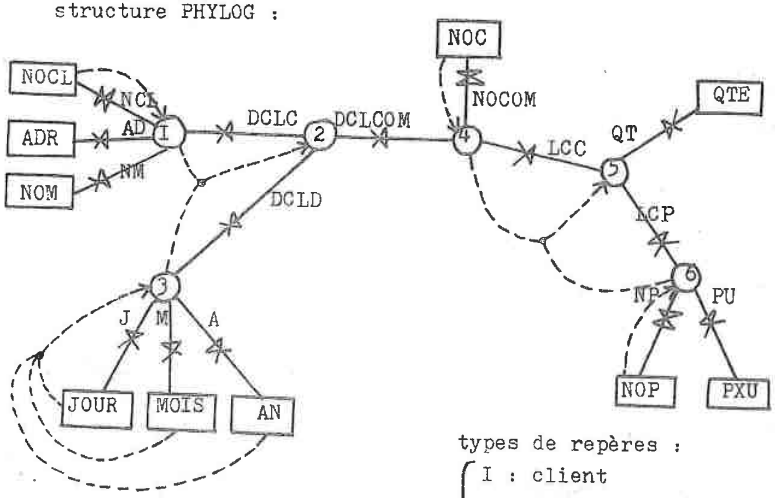


type de repères:
I : fournisseur fournissant toutes les pièces

fig.2.30(b)

Il s'agit ici d'un type de repères dont la définition fait "intervenir la notion de quantificateur universel".
Il est bien clair que la complexité de définition de tels types de repères n'a pas de limites a priori, ce qui justifie le type de démarche adopté pour étudier leur correspondance avec PHYLOG.
Exemple 5:il s'agit d'un premier exemple complet (structures LOG, PHYLOG et implémentation),qui illustre un autre cas de définition de repères logiques.

structure PHYLOG :



NOCL :n° client
ADR :adresse client
NOC :n° commande
NOP :n° produit
PXU :prix unitaire

types de repères :
- I : client
- 2 : date-client
- 3 : date
- 4 : commande
- 5 : ligne de commande(une seule ligne par produit)
- 6 : produit

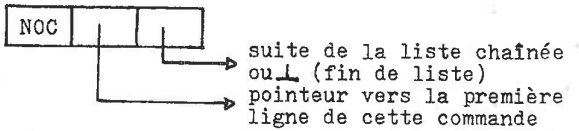
fig. 2.3I(a)

L'implémentation correspondante est la suivante :

+ une liste contigüe de "date-client" en mémoire centrale, avec comme élément de la liste:

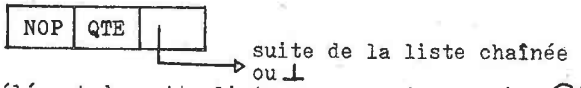
(à chaque élément de cette liste correspond un repère ② ;à chaque élément de la liste contigüe des jours ouvrables correspond un repère ③).

+ une liste chaînée en mémoire centrale des commandes pour un jour et un client donné, avec comme élément :



(à chaque élément de cette liste correspond un repère ④)

+ une liste chaînée en mémoire centrale des lignes d'une commande, avec comme élément:

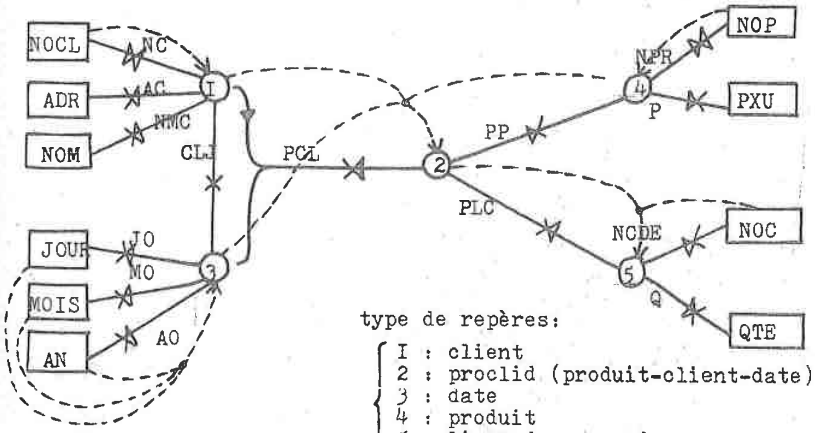


(à chaque élément de cette liste correspond un repère ⑤)

+ un fichier "client" en accès direct (clé NOCL) (rep. Ⓘ)

+ un fichier "produit" en accès direct (clé NOP) (rep. ⑥)

structure LOG :



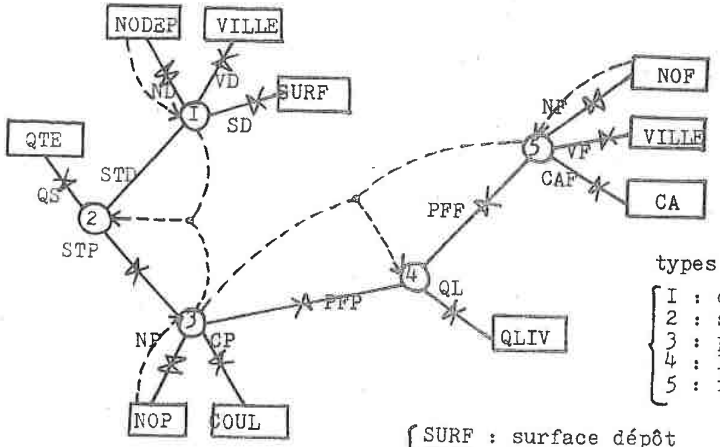
type de repères:
- I : client
- 2 : proclid (produit-client-date)
- 3 : date
- 4 : produit
- 5 : ligne de commande

fig. 2.3I(b)

Le type de repères "proclid" est défini par le "rapprochement" des notions de produit, client et date. C'est l'illustation d'un type courant de structuration pour les utilisateurs : ② joue en quelque sorte le rôle de "carrefour" pour les relations d'accès à Ⓘ , ③ , ④ et ⑤.

Exemple 6: il s'agit d'un second exemple complet qui nous permettra d'illustrer la notion de cohérence sémantique entre structures évoquée au paragraphe 0.I.

structure PHYLOG : fig. 2.32(a)



types de repères :
- I : dépôt
- 2 : stock
- 3 : pièce
- 4 : livraison
- 5 : fournisseur

- SURF : surface dépôt
- COUL : couleur pièce
- CA : chiffre d'affaires fournisseur
- QLIV : qté à livrer (réapprovisionement fixe)

L'implémentation correspondante est la suivante :

+ une liste des "stocks" ,en mémoire centrale, avec comme élément :

NODEP	NOP	QTE

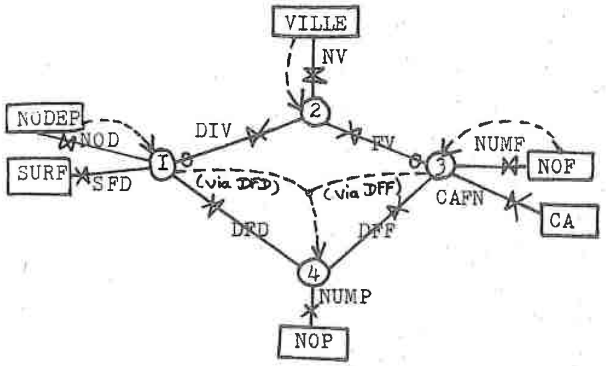
+ une liste des "livraisons", en mémoire centrale, avec comme élément :

NOP	NOF	QLIV

+ 3 fichiers en accès direct :
  - "dépôt" (clé NODEP)
  - "pièce" (clé NOP)
  - "fournisseur" (clé NOF)

structure LOG : fig. 2.32(b)

types de repères :
- I : dépôt
- 2 : localisation
- 3 : fournisseur
- 4 : dépôt-fournisseur



Présentées sans autres indications, ces deux structures ne sont pas sémantiquement cohérentes. En effet, rien dans PHYLOG ne correspond aux repères "dépôt-fournisseur" : on y trouve les pièces fournies par les fournisseurs et les pièces stockées dans les dépôts, mais pas les dépôts approvisionnés par les fournisseurs. Il n'y a cohérence que dans l'hypothèse où un fournisseur approvisionne tous les dépôts ou est stockée une pièce dont il est le fournisseur. C'est lors de la conception des structures ,ainsi que nous l'avons déjà signalé, que devra être vérifiée la cohérence sémantique des structures (cf.0.2.2.I (a)).

- - - - -

2.6. Conclusion :

Cette seconde partie est très liée au système particulier étudié, dans une optique pratique : elle tente de répondre au besoin des utilisateurs d'une caractérisation moins théorique et plus précise du niveau interne. Elle devra se prolonger par une étude qualitative des choix d'implémentation possibles.

Au terme de ces deux premières parties, les "extrémités" de la correspondance <LOG-PHYLOG> étant plus précisémment analysés, nous allons pouvoir aborder dans la troisième partie le problème spécifique de cette correspondance et de son "élargissement".

°°°°°

BIBLIOGRAPHIE :

[2.I] Interim Report ANSI/X3/SPARC
ANSI
(Février I975)

[2.2] Description formelle, représentation, interrogation des informations complexes- système PIVOINES
M.CREHANGE
Thèse d'Etat (Nancy I)
(I975)

[2.3] Descriptions d'organisations physiques de données
A. CABANES
I.I.E.-CNAM
(Janvier I973)

[2.4] Cours sur les structures d'informations
C.PAIR
(I97I)

° ° ° ° °

CHAPITRE 3 :

RESOLUTION DU PROBLEME DE LA "CORRESPONDANCE ELARGIE"

3.I. Introduction ..... p. I

3.2. Présentation de l'algèbre relationnelle utilisée .... p. 3

3.2.I. L'algèbre relationnelle de Codd

3.2.I.I. Les opérations ensemblistes classiques
3.2.I.2. La projection
3.2.I.3. La jointure
3.2.I.4. La restriction
3.2.I.5. La division

3.2.2. L'algèbre relationnelle de PRTV

3.2.2.I. La sélection
3.2.2.2. La différence sur constituants communs
3.2.2.3. La jointure sur constituants communs

3.2.3. L'algèbre relationnelle utilisée dans la suite de l'étude

3.3. L'algèbre relationnelle, langage d'expression de la correspondance ..... p. II

3.4. L' "interprétation" des opérations de l'algèbre relationnelle dans le modèle de PIVOINES ..... p. I4

3.4.I. Interprétation de la sélection

3.4.2. Interprétation de la projection

3.4.3. Interprétation de la jointure naturelle

3.4.3.I. La jointure selon l'ensemble vide (ou produit cartésien)
3.4.3.2. La jointure selon un constituant commun

3.4.4. Interprétation de la division

3.4.5. Interprétation des opérations ensemblistes classiques

3.4.6. Conclusion

3.5. Les "objets équivalents" au résultat d'une expression quelconque de l'algèbre réduite ..... p. 24

3.5.I. Justification de l'algèbre réduite

3.5.2. La simplification et l'optimisation des expressions de l'algèbre réduite
3.5.2.I. la composition des opérations élémentaires
3.5.2.2. la transformation des exressions relationnelles
3.5.3. Les objets équivalents

3.6. L'élargissement de l'algèbre ..... p. 40
3.6.I. L'utilisation de la division
3.6.2. L'utilisation de la réunion,de l'intersection et de la différence
3.6.3. Conclusion

3.7. La gestion des objets équivalents parle système PIVOINES..p.46
3.7.I. Généralités
3.7.2. La gestion mémoire
3.7.3. Essai d' évaluation "économique" de l'utilisation des objets équivalents.

3.8. Conclusion ..... p. 49

BIBLIOGRAPHIE ..... p. 50

°°°°°°°°

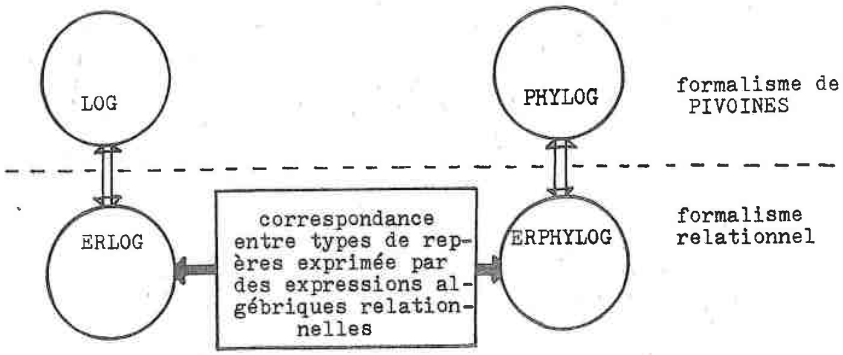
3.I. Introduction :

Dans les chapitres précédents nous avons mis en évidence un certain nombre de points importants dont les trois principaux sont :

+ la nécessité de résoudre le problème de la "correspondance élargie" (c.à.d. de la différenciation des types de repères logiques par rapport aux types de repères de PHYLOG) pour permettre de prendre en compte fidèlement les "visions" différentes des utilisateurs (cf. paragraphe I.4.3.3.). Il faut remarquer qu'il existe d'autres justifications possibles à la correspondance élargie, même dans le cas d' une vision utilisateur unique : en effet, l'évolution dans le temps des besoins peut avoir pour conséquence une modification de cette vision unique, entraînant des distorsions avec la vue interne, povant se traduire par une non correspondance des types de repères aux deux niveaux.
+ la complexité arbitrairement grande que peut prendre la définition des repères logiques vis à vis de PHYLOG (cf. paragraphe 2.5.).
+ la possibilité de se concentrer principalement sur les repères pour étudier la correspondance LOG-PHYLOG élargie, en raison du fonctionnement du système (cf. paragraphe 2.3.-b).

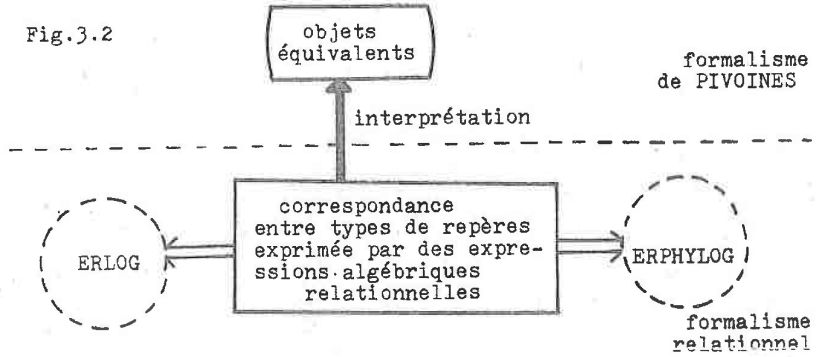
Les deux derniers points justifient la démarche retenue pour résoudre dans une optique générale le problème posé: ses grandes lignes ont étées présentées au paragraphe 0.3.2.
Nous avons vu aux paragraphes I.5.4. et 2.4.3. comment associer des "expressions relationnelles" ERLOG et ERPHYLOG aux structures LOG et PHYLOG, avec une relation associée à chaque type de repère. La correspondance entre types de repères pourra s'exprimer par des expressions algébriques donnant les relations associées aux types de repères logiques en fonction des relations de ERPHYLOG (associées aux types de repères de PHYLOG ou à des relations de PHYLOG -CF. paragraphes I.5.4 et 2.4.3.)

Fig. 3.I



Le paragraphe 3.I. précisera l'algèbre relationnelle utilisée et sa puissance.
Le paragraphe 3.2. illustrera sur les exemples du paragraphe 2.5 ce mode de description de la correspondance. Il faut bien noter que cette correspondance est une donnée du problème : nous n' indiquons qu'une méthode de description.
Les paragraphes suivants permettront, par l' "interprétation" des opérations algébriques dans le contexte de PIVOINES, de définir la forme la plus générale des objets équivalents qui vont "matérialiser" les repères logiques sans correspondants directs dans PHYLOG.

Fig.3.2



Le dernier paragraphe étudiera , dans le cas le plus général, les conséquences de l'introduction des objets équivalents dans le système PIVOINES et proposera des solutions adéquates.

3.2. Présentation de l'algèbre relationnelle utilisée :

Plusieurs algèbres relationnelles ont été proposées. Nous en analyserons deux, parmi les plus connues: celle de Codd, le fondement de toutes, présentée dans un contexte plus théorique que pratique [3.I] et celle de PRTV [3.2], plus destinée à l' utilisation, car à la base d'un langage de requête (ISBL).
Nous en déduirons une algèbre adaptée pour nos besoins.
Pour Codd, 4 opérations spécifiques (jointure, projection, restriction et division) s'ajoutent aux opérations ensemblistes classiques (produit cartésien, union, intersection, différence).
Leurs définitions sont précisées au paragraphe 3.2.I.
Dans ISBL, on retrouve les opérations ensemblistes et 3 opérations spécifiques (sélection, jointure et projection). Leurs définitions, parfois différentes des précédentes même si les opérations portent le même nom, sont précisées au paragraphe 3.2.2.
Le paragraphe 3.2.3 présente et justifie l'algèbre utilisée dans cette étude.

3.2.I. L'algèbre relationnelle de Codd :

3.2.I.I Les opérations ensemblistes classiques:

Une relation est un ensemble (cf.I.5.I): on peut donc appliquer aux relations les opérations ensemblistes classiques : produit cartésien, union, intersection et différence. Pour les trois dernières opérations il est nécessaire que les relations soient "union-compatibles" (cf. [3.I], parag.2.2).

+ produit cartésien :
  $R * S = \{ r \hat{} s \mid r \in R \wedge s \in S \}$
  notations : $\wedge$:"et" $\vee$:"ou"
  $r \hat{} s$ concaténation de r et s
+ union :
  $R+S = \{ t \mid t \in R \vee t \in S \}$
+ intersection :
  $R.S = \{ t \mid t \in R \wedge t \in S \}$
+ différence :
  $R-S = \{ t \mid t \in R \wedge t \notin S \}$

3.2.I.2. La projection :

Soit $L = (l_1, l_2, \ldots\ldots, l_m)$ une liste de noms de constituants d'une relation n-aire donnée ($n \geq m$), R.

On note : $\begin{cases} r \text{ un n-uple de R} \\ r[j] \text{ la valeur du constituant j de R} \\ r[L] = (r(l_1), r(l_2), \ldots, r(l_m)) \end{cases}$

La projection de R sur L, notée $R[L]$ ou $\prod_L R$, est définie par:

$$R[L] = \{ r[L] \mid r \in R \}$$

Exemples: CLIENT(NOCL, NOM, ADRESSE, AGE)

I	TOTO	NANCY	33
2	JOHN	METZ	39
3	JOE	NANCY	45
4	MIKE	METZ	39
5	HUBERT	METZ	27

CLIENT [ADRESSE]

NANCY
METZ

CLIENT [ADRESSE, AGE]

NANCY	33
METZ	39
NANCY	45
METZ	27

CLIENT [NOM]

TOTO
JOHN
JOE
MIKE
HUBERT

3.2.I.3. La jointure :

Soit θ un opérateur de comparaison $=, \neq, >, <, \geq, \leq$. La θ-jointure d'une relation R (n-aire) sur le constituant (*) A avec une relation S (p-aire) sur le constituant B est la relation (n+p-aire) définie par :

$$R[A \theta B] S = \{ (r \hat{} s) \mid r \in R \wedge s \in S \wedge (r[A] \theta s[B]) \}$$

($a \theta b$ doit être défini pour tout $a \in A$ et tout $b \in B$; on dit que les domaines A et B doivent être "θ-comparables" cf. [3.I]).

Exemples :

ARRIVEE (NOVOLA, H-ARR)

I0I	9
I05	I3
307	2I

DEPART (NOVOLD, H-DEP)

605	I0
I03	I8

avec: NOVOLA: n° vol arrivée
NOVOLD: n° vol départ
H-ARR : heure d'arrivée
H-DEP : heure départ

La relation donnant les correspondances possibles du jour est définie par:

ARRIVEE H-ARR H-DEP DEPART = CORR(NOVOLA, H-ARR, NOVOLD, H-DEP)

I0I	9	605	I0
I0I	9	I03	I8
I05	I3	I03	I8

Si l'opérateur est = , on parle d' "équi-jointure". Dans ce cas , deux constituants de la relation résultat ont des contenus

(*) constituant simple ou ensemble de constituants simples (dans toute la suite)

identiques. Quand on en supprime un, le résultat est appelé "jointure naturelle". Elle s'effectue très souvent sur des relations ayant un constituant commun et se note alors :

$R[A]\ S$ si $\begin{cases} R(\ldots\ldots, A, \ldots) \\ S(\ldots\ldots, A, \ldots) \end{cases}$

Exemple:

COMMANDE (NOC , NOCL, ...)

I	I00
2	I0I
3	I00
4	I02
5	I0I

{NOC: n° cde
{NOCL: n° client

CLIENT (NOCL , NOM , ...)

I00	TOTO
I0I	JOHN
I02	JOE

COMMANDE [NOCL] CLIENT = CC (NOC , NOCL , NOM , ...)

I	I00	TOTO
2	I0I	JOHN
3	I00	TOTO
4	I02	JOE
5	I0I	JOHN

3.2.I.4. La restriction :

Soit une relation R et deux de ses constituants A et B. La restriction de R selon $A \theta B$, notée $R[A \theta B]$ est définie par :

$$R[A \theta B] = \{ r \mid r \in R \wedge (r[A] \theta r[B]) \}$$ (sous réserve de θ-compatibilité)

Exemple :

PIECE (NOP , QALERTE , QSTOCK , ..)

I	I00	80
5	20	I350
8	I0	I7
I0	500	420

QALERTE: qté d'alerte
QSTOCK: qté en stock

REAPPROV (NOP , ...) = PIECE [QSTOCK < QALERTE]

I ..
I0 ..

3.2.I.5. La division :

a) notion d'ensemble image : soit $R(X, \overbrace{\ldots\ldots}^{Z})$ et z_j un p-uple quelconque de Z (Z est l'ensemble des constituants de R autres que X).

On appelle ensemble image de $x_i \in X$: $g_R(x_i) = \{ z_j \mid (x_i, z_j) \in R \}$

Soit $R(\underbrace{\ldots,\ldots}_{A}, \underbrace{\ldots\ldots}_{\overline{A}})$ et $a_i \in A$, $\overline{a}_j \in \overline{A}$.

On appelle ensemble image du k-uple $a_i \in A$:

$$g_R(a_i) = \{ \overline{a}_j \mid (a_i, \overline{a}_j) \in R \}$$

Exemple: soit la relation CLIENT du paragraphe 3.2.I.2.

$g_{CLIENT}(NANCY) = \{(I,TOTO,33),(3,JOE,45)\}$

b) division de R par S relativement à A et B :

A et B sont des constituants respectifs de R et S, union-compatibles. On a :

$$R\,[A \div B]\;S = \{\, r[\bar{A}] \mid r \in R \wedge S[B] \subseteq g_R(r[\bar{A}]) \,\}$$

Exemple : LIGNE-COMMANDE(NOC , NOP) PIECE(NOP ,PXU ,...)

NOC	NOP
I0	I
I0	2
II	I
II	2
II	3
I2	3
I3	I

NOP	PXU
I	I00
2	I0
3	375

LIGNE-COMMANDE [NOP ÷ NOP] PIECE = R (NOC = II)

en effet: PIECE [NOP] = {I,2,3}

$g_{L-C}(I0) = \{I,2\}$

$g_{L-C}(II) = \{I,2,3\}$

$g_{L-C}(I2) = \{3\}$

$g_{L-C}(I3) = \{I\}$ (avec L-C ≡ LIGNE-COMMANDE)

On peut donner l'interprétation "en clair" suivante de la division: le "quotient" (R [A ÷ B] S) est l'ensemble des complémentaires dans R du "dividende" (A) dont les ensembles images contiennent toutes les valeurs dans S du "diviseur"(B). Dans l'exemple ci-dessus, R donne les commandes comportant tous les produits. La division constitue un "équivalent algébrique" du quantificateur universel.

3.2.2. L'algèbre relationnelle de PRTV :

Cette algèbre est intéressante car elle constitue la base d'un langage d'interrogation (du type manipulation algébrique) et qu'elle a donc été conçue avec le souci d'assurer une utilisation facile.

Elle est moins riche que l'algèbre de Codd: pas de division ou d'opération équivalente.

Les opérations ensemblistes (union,intersection,différence, produit cartésien -nommé "full quadratic join" car R * S=R[Ø]S) et la projection sont identiques aux opérations de Codd. Trois opérations ont des définitions originales : la sélection, la

différence sur constituants communs et la jointure sur constituants communs.

3.2.2.I. La sélection :

Il s'agit d'une généralisation triviale de la restriction de Codd. Elle est notée :

R : filtre où { R est une relation ; le filtre est une expression booléenne dont les éléments sont des comparaisons entre 2 constituants ou d'un constituant avec une constante.

Le résultat est une relation R' qui est un sous-ensemble de R dont tous les uples vérifient le filtre.

Exemple : soit LIVRE(NUMLIV, AUTEUR , TITRE , ANNEE) et

LIVRE : (ANNEE = 75 ∨ ANNEE = 76) ∧ AUTEUR = DUPONT .

Dans l'algèbre de Codd il aurait été nécessaire d'écrire:

(LIVRE [ANNEE = 75] + LIVRE [ANNEE = 76]) . LIVRE [AUTEUR = DUPONT]

3.2.2.2. La différence sur constituants communs :

Soient 2 relations R et S ayant une partie Z commune (constituant ou ensemble de constituants);

$R - S = \{\, r \in R \mid r[Z] \notin S[Z] \,\}$ (la différence se fait donc selon les seuls constituants communs).

Exemple : LIVRE (NUMLIV, AUTEUR, TITRE, ANNEE)
(livres de la bibliothèque)
PRET (NUMLIV, NOM, ADRESSE)
(livres prêtés par la bibliothèque)
LIVRE - PRET = PRESENT(NUMLIV, AUTEUR, TITRE,ANNEE)
(livres présents en bibliothèque)

Dans l'algèbre de Codd il aurait été nécessaire d'écrire:

(LIVRE [NUMLIV] - PRET [NUMLIV]) [NUMLIV] LIVRE

3.2.2.3. La jointure sur constituants communs:

Il s'agit d'une jointure naturelle qui s'effectue uniquement selon les constituants communs; s'il n'y en a pas on retrouve le produit cartésien.

Exemple : dans l'exemple du paragraphe 3.2.2.2

LIVRE ⊛ PRET = R(NUMLIV,AUTEUR,TITRE,ANNEE,NOM,ADRESSE)

(⊛ : jointure selon constituants communs)

Dans l'algèbre de Codd il aurait fallu écrire:
LIVRE [NUMLIV] PRET (l'indication de NUMLIV étant ici superfétatoire car c'est la seule possibilité de jointure).

Il faut cependant remarquer que dans les opérations sur constituants communs (c.à.d. de même nom), l'utilisateur n'est pas maître du choix des constituants impliqués, ce qui peut être gênant. Il peut le redevenir s'il a la faculté de renommer les constituants : c'est le cas ici, grâce à la projection. L'indication X→Y dans une projection indique un changement de nom lors de la projection.

exemple: la projection (notée % dans PRTV) :
LIVRE % NUMLIV → NOL, AUTEUR, TITRE
donne la relation
LIVRE(NOL, AUTEUR, TITRE) à la place de LIVRE(NUMLIV, AUTEUR, TITRE, ANNEE).

3.2.3. L'algèbre relationnelle utilisée dans la suite de l'étude:

Elle comprend : la sélection, la projection, la jointure naturelle, la division et les opérations ensemblistes (produit, union, intersection, différence).

La sélection a été préférée à la restriction pour sa plus grande généralité.

La jointure naturelle est présentée avec la liste explicite des constituants concernés. La θ-jointure n'est pas utile car:

$$R\,[A\,\theta\,B]\,S = R * S : A\,\theta\,B$$

La projection est l'opération classique, avec la notation de Codd.

La division a une définition un peu différente de celle de Codd, à savoir :

$$R\,[A, X \,(\div)\, B]\,S = \{r \mid r \in R \wedge S[B] \subseteq g_{R[A,X]}(r[\bar{A}])\}$$
avec: $X \subseteq \bar{A}$

Cette définition généralise celle de Codd. En effet:

$$R\,[A \div B]\,S = (R\,[A, A\,(\div)\,B]\,S)\,[\bar{A}]$$

Elle permet d'obtenir, grâce à sa souplesse, un résultat directement significatif, alors que le résultat de la division de Codd ne l'est pas toujours en dehors du cas simple des relations binaires (le seul retenu par Date [3.3], où X et $\bar{A}$ sont nécessairement identiques et égaux au second constituant de R).

Exemple : soit: APPROV(NOF, NOP, DELAIS, PRIXU)
PIECE(NOP, DESIGNATION)
et: S_I = APPROV [NOP ÷ NOP] PIECE
S_2 = APPROV [NOP, NOF (÷) NOP] PIECE

Au niveau des occurrences :

PIECE

NOP	DESIGNATION
I	pelle
2	pioche

APPROV

NOF	NOP	DELAIS	PRIXU
I00	I	3	200
I00	2	3	200
I0I	2	5	205
I02	I	5	203
I02	2	5	205
I03	I	3	I95

S_I

NOF	DELAIS	PRIXU
I00	3	200

S_2

NOF	NOP	DELAIS	PRIXU
I00	I	3	200
I00	2	3	200
I02	I	5	203
I02	2	5	205

La sigification de S_2 est immédiate: il s'agit des approvisionnements des fournisseurs fournissant toutes les pièces. La signification de S_I l'est beaucoup moins (car dans $\bar{A}$ intervient aussi DELAIS et PRIXU). Pour obtenir S_2 avec la division de Codd il faut écrire :

((APPROV[NOF, NOP]) [NOP ÷ NOP] PIECE) [NOF] APPROV

L'ensemble des opérations et leurs notations sont rappelés dans le tableau de la page 3.I0 .

Cette algèbre "généralise" celle de Codd sur quelques détails : sélection, division modifiée. Dans les deux cas l'opération de Codd est un cas particulier des opérations retenues ici. Notre algèbre est donc complète au sens de Codd. (cf. [3.I])

- - - - -

NOM OPERATION	NOTATION	Paragraphe ou cette opération est définie
sélection	R : filtre	3.2.2.I.
projection	R [L]	3.2.I.2.
jointure naturelle	$R_I [L] R_2$	3.2.I.3.
division	$R_I[L_I, L_2 \textcircled{\div} L_3]R_2$	3.2.3.
produit cartésien	$R_I * R_2$ ou $R_I[\emptyset] R_2$	3.2.I.I.
union	$R_I + R_2$	3.2.I.I.
intersection	$R_I \cdot R_2$	3.2.I.I.
différence	$R_I - R_2$	3.2.I.I.

avec : $\begin{cases} R_i & \text{relation} \\ L_i & \text{liste de constituants simples} \end{cases}$

3.3. L'algèbre relationnelle, langage d'expression de la correspondance :

A chaque type de repères logiques est associée une relation de ERLOG. Nous exprimerons la correspondance entre ces types de repères et PHYLOG en donnant la façon dont la relation de ERLOG dérive des relations de ERPHYLOG par une séquence d'opérations algébriques relationnelles. Il peut ne pas y avoir unicité : on choisira la dérivation la plus simple.
On peut noter que DATE ([3.3]p.II6,II7) utilise une démarche similaire pour définir, dans le modèle relationnel, les sous-schémas utilisateurs d'un schéma global : chaque relation du sous-modèle doit être dérivable des relations du modèle global par une séquence d'opérations algébriques relationnelles.
Nous allons dans la suite illustrer cette forme d'expression de la correspondance en reprenant les 6 exemples du paragraphe 2.5. Les expressions relationnelles ERLOG et ERPHYLOG sont construites selon les règles du paragraphe I.5.4.

Exemple I:

ERPHYLOG : EMPLOYE(NOE,CATEG,.....)
ERLOG : CADRE(NOE,.....)
correspondance :
| CADRE = EMPLOYE : CATEG = "cadre"

Exemple 2:

ERPHYLOG : FOURN(NOF,VILLE,....) associée à ① de PHYLOG
PIECE(NOP,.....) associée à ② "
FP(NOF,NOP,QL) associée à ③ "
ERLOG : LOCP(VILLE) associée à ② de LOG
PIECE2(NOP,.....) associée à ① "
PV(NOP,VILLE) associée à la relation PV
et: | LOCP = FOURN [VILLE]
| PIECE2 = PIECE
(PV = (FP [NOF] FOURN)[NOP,VILLE] ne nous intéresse pas, car PV n'est pas associée à un type de repères logiques).

Exemple 3:

ERPHYLOG : PERSONNE(NOM,DATE-MARIAGE,....) associée à ①
A-P-EP(NOM/EPOUX,NOM/EPOUSE) associée à la relation A POUR EPOUSE

ERLOG : PERSONNE2(NOM) associée à ① de LOG
MARIAGE(NOM/EPOUX,NOM/EPOUSE,DATE-MARIAGE)
associée à ② de LOG

correspondance :
- PERSONNE2 = PERSONNE [NOM]
- MARIAGE = A-P-EP [NOM/EPOUX] PERSONNE

Exemple 4:

ERPHYLOG : FOURN(NOF,VILLE,....) associée à ① de PHYLOG
PIECE(NOP,DES,.....) associée à ② "
LIVR(NOF,NOP,QLIV) associée à ③ "

ERLOG : FTOT(NOF,VILLE,.....) associée à ① de LOG

correspondance :

FTOT = ((FOURN[NOF] LIVR)[NOP,NOF ⊜ NOP] PIECE)[NOF,VILLE, ...]

Exemple 5:

ERPHYLOG : CLIENT(NOCL,ADR,NOM) associée à ① de PHYLOG
DATE-CLT(NOCL,JOUR,MOIS,AN) " ② "
DATE(JOUR,MOIS,AN) " ③ "
CDE(NOC,NOCL,JOUR,MOIS,AN) " ④ "
LIGNE(NOC,NOP,QTE) " ⑤ "
PRODUIT(NOP,PXU) " ⑥ "

ERLOG : CLIENT2(NOCL,ADR,NOM) associée à ① de LOG
DATE2(JOUR,MOIS,AN) " ③ "
PROCLID(NOCL,JOUR,MOIS,AN,NOP) ② "
CLJ(NOCL,JOUR,MOIS,AN) " la relation CLJ de LOG
PRODUIT2(NOP,PXU) " ④ "
LC(NOC,NOP,NOCL,JOUR,MOIS,AN,QTE) ⑤ "

correspondance:
- CLIENT2 = CLIENT
- DATE2 = DATE
- PRODUIT2 = PRODUIT
- PROCLID = (CDE[NOC]LIGNE)[NOCL,JOUR,MOIS,AN,NOP]
- LC = LIGNE [NOC] CDE

Exemple 6 :

ERPHYLOG : DEPOT(NODEP,VILLE,SURF) associée à ① de PHYLOG
STOCK(NODEP,NOP,QTE) " ② "
PIECE(NOP,COUL) " ③ "
LIVR(NOP,NOF,QLIV) " ④ "
FOURN(NOF,VILLE,CA) " ⑤ "

ERLOG : DEPOT2(NODEP,SURF,VILLE) associée à ① de LOG
LOCAL(VILLE) " ② "
FOURN2(NOF,CA,VILLE) " ③ "
DF(NODEP,NOF) " ④ "
NUMP(NODEP,NOF,NOP) " à la relation NUMP de LOG

correspondance :
- DEPOT2 = DEPOT
- FOURN2 = FOURN
- LOCAL = DEPOT [VILLE] + FOURN [VILLE]
- DF = (STOCK [NOP] LIVR) [NODEP,NOF]

Remarque: si on voulait avoir au niveau logique les localisations communes aux fournisseurs et aux dépôts, on aurait:

LOCAL2 = DEPOT[VILLE]. FOURN[VILLE]

si on voulait les localisations de dépôts sans fournisseurs, on aurait:

LOCAL3 = DEPOT[VILLE]- FOURN[VILLE]

Les 6 exemples ci-dessus mettent en évidence l'utilisation de toutes les opérations de l'algèbre. Ils permettent d'illustrer le degré de complexité (en général assez peu élévé) des expressions traduisant le mapping.

- - - - -

3.4. L'interprétation des opérations de l'algèbre relationnelle dans le modèle de PIVOINES :

Les données du problème sont les suivantes :

+ S relation de ERLOG, associée à un type de repères logiques sans équivalent direct dans PHYLOG
+ R, R_I, R_2 des relations de ERPHYLOG telles que:
  - soit $S = R_I \ Op_2 \ R_2$ où Op_2 est un opérateur binaire de l'algèbre relationnelle
  - soit $S = Op_I \ R$ où Op_I est un opérateur unaire de l'algèbre relationnelle

Les relations de ERPHYLOG sont associées essentiellement aux classes de repères de PHYLOG et parfois à des relations PIVOINES non traduisibles autrement (cf. paragraphe I.5.4).

Interpréter une opération de l'algèbre relationnelle dans le modèle de PIVOINES consistera à :

+ déterminer la nature de "quelque chose" dans PHYLOG ayant une occurrence pour chaque n-uple de la relation S, donc pour chaque repère logique ("objet équivalent" ou O.E.).
+ préciser la nature des accès et des algorithmes permettant de construire ces O.E.

Ce paragraphe ne constitue qu'une première étape, car se posera ensuite le problème de la composition des opérations et de ses conséquences.

3.4.I. Interprétation de la sélection :

Soit : S = R : filtre.

Le filtre porte sur un nombre quelconque de constituants de R qu'il compare entre eux ou avec des constantes, le tout constituant une expression booléenne de comparaisons élémentaires.

Dans le cas général R est associée à un type de repères (i) de PHYLOG.

A chaque occurrence de S correspond un repère du type (i) dont les valeurs qui lui sont associées (valeurs des constituants de R) vérifient le filtre.

exemple I : CADRE = EMPLOYE : CATEG = "cadre" (cf. exemple I des paragraphes 2.5 et 3.3)

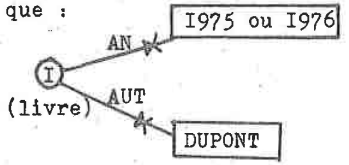
A chaque occurrence de CADRE correspond un repère du type (I) de PHYLOG tel que :

fig. 3.3

exemple 2: DUPONT56 = LIVRE : (ANNEE = 75 ∨ ANNEE = 76) ∧ AUTEUR = "DUPONT" (cf.3.2.2.I.)

A chaque occurrence de DUPONT56 correspond un repère du type "livre" tel que :

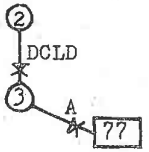
fig. 3.4



exemple 3: R = DATE-CLT : AN = 77 (cf. exemple 5 de 3.3et 2.5).

A chaque occurrence de R correspond un repère du type (2) de PHYLOG tel que:

fig. 3.5



Opérer une sélection sur une relation peut donc s'interpréter comme opérer une sélection sur la classe de repères associée, selon un algorithme plus ou moins complexe qui peut éventuellement nécessiter, comme dans l'exemple 3, des accès à d'autres types de repères de PHYLOG. Le "résultat" de la sélection est un sous-ensemble de la classe de repères considérée : c'est l'ensemble des O.E.

Dans le cas particulier où la relation R de ERPHYLOG serait associée à une classe de relations et non à une classe de repères, le processus de sélection serait du même type, mais s'exercerait sur des couples (ou des n-uples) de repères liés par la relation considérée.

3.4.2. Interprétation de la projection :

On considère la projection d'une relation R de ERPHYLOG sur un sous-ensemble $\mathcal{Y}$ de ses constituants (on notera $\bar{\mathcal{Y}}$ le sous-ensemble complémentaire de $\mathcal{Y}$): $S = R[\mathcal{Y}]$

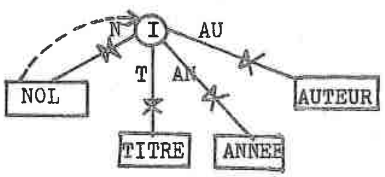
Dans le cas général R est associée à un type de repères (i) de PHYLOG. On peut éliminer le cas trivial où la clé de R est contenue toute entière dans $\mathcal{Y}$: dans ce cas une occurrence de $\mathcal{Y}$ est associée à chaque occurrence de R et donc à chaque repère (i). Il n'y a pas de problème d'objet équivalent.

On considère maintenant le cas où une partie au moins de la

clé de R n'est pas contenue dans $\mathcal{F}$.
A chaque occurrence de $\mathcal{F}$ peuvent correspondre plusieurs occurrences de R dont les valeurs des constituants de $\mathcal{F}$ sont identiques mais qui diffèrent par les valeurs des constituants de $\bar{\mathcal{F}}$.

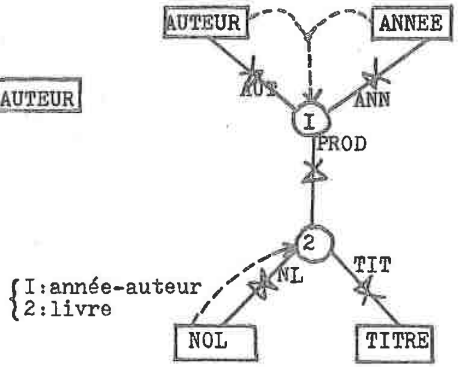
exemple : soit LIVRE(NOL,TITRE,AUTEUR,ANNEE) de ERPHYLOG
et: AN-AUT(ANNEE,AUTEUR) = LIVRE[AUTEUR,ANNEE] de ERLOG.

PHYLOG:



I:livre

LOG:



{ I:année-auteur
{ 2:livre

fig. 3.6

au niveau des occurrences:

LIVRE

NOL	AUTEUR	TITRE	ANNEE	repères de PHYLOG
I0I	ARTHUR	t_1	I975	r_1
I02	TOTO	t_2	I977	r_2
I03	TOTO	t_3	I974	r_3
I04	ARTHUR	t_4	I975	r_4
I05	ARTHUR	t_5	I978	r_5
I08	ARTHUR	t_6	I975	r_6
II0	DUPONT	t_7	I975	r_7

AN-AUT

ANNEE	AUTEUR	repères de LOG
I975	ARTHUR	s_1
I977	TOTO	s_2
I974	TOTO	s_3
I978	ARTHUR	s_4
I975	DUPONT	s_5

A l'occurrence (ARTHUR,I975) de AN-AUT correspondent les occurrences (I0I,ARTHUR,t_1,I975),(I04,ARTHUR,t_4,I975) et (I08,ARTHUR,t_6,I975) de LIVRE. Le problème est de déterminer si l'O.E. au repère logique s_1 associé à (ARTHUR,I975) est l'ensemble des repères r_1,r_4,r_6 associés aux 3 occurrences de LIVRE ci-dessus

ou si l'on peut en choisir un comme "représentant" ? Il est clair qu'en ne conservant pas les autres repères on perd de l'information : on n'est plus capable à partir du seul représentant d'accéder à toutes les valeurs de NOL ou de TITRE associées à s_1. On conservera donc comme objet équivalent l'ensemble des repères de PHYLOG associés aux mêmes valeurs de $\mathcal{F}$. Cela nécessite donc l'accès à ces valeurs puis le regroupement des repères de PHYLOG "semblables" en ensembles. On discutera plus loin les problèmes que peuvent poser de tels objets équivalents.

exemple : dans l'exemple précédent

à s_1 correspond $\{r_1,r_4,r_6\}$
à s_2 " $\{r_2\}$
à s_3 " $\{r_3\}$
à s_4 " $\{r_5\}$
à s_5 " $\{r_7\}$

Dans le cas particulier où la relation R serait associée à une classe de relations et non à une classe de repères, le processus serait le même, mais s'exercerait sur des couples (ou des n-uples) de repères liés par la relation considérée.

3.4.3. Interprétation de la jointure naturelle :

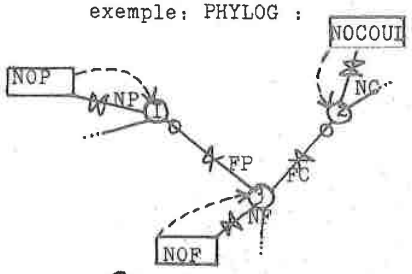
Nous étudierons dans un premier temps la jointure selon l'ensemble vide ou produit cartésien. Puis nous étudierons la jointure naturelle selon un constituant commun.

3.4.3.I. La jointure selon l'ensemble vide(ou produit cartésien) :

Soit : $S = R_1[\emptyset]R_2 = R_1 * R_2$

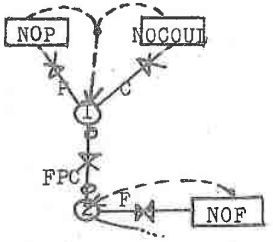
Dans le cas général R_1 et R_2 sont associées à des types de repères (i) et (j) de PHYLOG. A chaque occurrence de S correspond un couple de repères de (i) et (j). Les O.E. sont ces couples. Il est donc nécessaire de déterminer tous les repères des deux types et de construire tous les couples possibles.

exemple: PHYLOG :



{ 1 : pièces
{ 2 : couleurs
{ 3 : fournisseurs

LOG:



{ 1 : "pièce-couleur"(toutes les associations possibles)
{ 2 : fournisseurs

fig.3.7

ERPHYLOG : PIECE(NOP,NOF,.....) Associée à ① de PHYLOG
COULEUR(NOCOUL,NOF,.....) " ② "
FOURN(NOF,.....) " ③ "
ERLOG : PICOUL(NOP,NOCOUL) Associée à ① de LOG
FOURN(NOF,.....) " ② "
FPC(NOP,NOCOUL,NOF) " à la relation FPC

correspondance:

⌊ PICOUL = PIECE [NOP] ✻ COULEUR [NOCOUL]

au niveau des occurrences :

PIECE

NOP	NOF
I	I0
2	II
3	I0
5	I2
8	II

COULEUR

NOCOUL	NOF
I00	II
I0I	II
I02	I0

FOURN

NOF
I0
II
I2

FPC

NOP	NOCOUL	NOF
I	I02	I0
3	I02	I0
2	I00	II
2	I0I	II
8	I00	II
8	I0I	II

PIECE [NOP] ✻ COULEUR [NOCOUL] = PICOUL

NOP	NOCOUL
I	I00
I	I0I
I	I02
2	I00
2	I0I
2	I02
3	I00
3	I0I
3	I02
5	I00
5	I0I
5	I02
8	I00
8	I0I
8	I02

Les objets équivalents aux repères "pièce-couleur" sont tous les couples de repères ① et ② de PHYLOG.

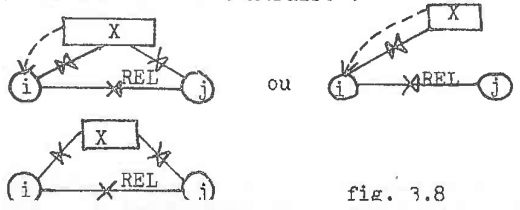
3.4.3.2. La jointure selon un constituant commun :

Soit: $S = R_I [X] R_2$

Dans le cas général R_I et R_2 sont associées à des types de repères (i) et (j) de PHYLOG.

On peut distinguer deux cas : soit (i) et (j) sont liés par une relation PIVOINES traduisant l'existence du constituant commun, soit non.

+ dans le premier cas, la relation PIVOINES est fonctionnelle si le constituant commun est une clé d'une des 2 relations et n:m dans le cas contraire :



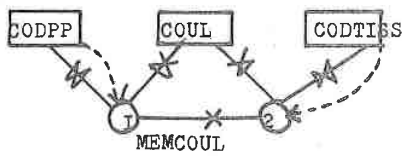
REL traduisant le partage de la clé de (i)

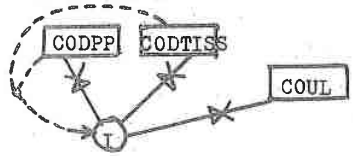
fig. 3.8

A chaque occurrence de S correspond un couple de repères de (i) et (j), liés par REL. Les objets équivalents sont des couples de repères liés.

exemple :

au niveau PHYLOG: au niveau LOG:





①: papier-peint
②: tissu

①: "coordonné" (papier+tissu de même couleur

{CODPP : code papier-peint
CODTISS : code tissu

ERPHYLOG : PAP(CODPP,COUL) associée à ① de PHYLOG
TISS(CODTISS,COUL) " ② "
MEMCOUL(CODPP,CODTISS) " la relation MEMCOUL

ERLOG : COORD(CODPP,CODTISS,COUL)

correspondance :

⌊ COORD = PAP [COUL] TISS

Les objets équivalents aux repères "coordonné" sont les couples de repères de ① et ② liés par MEMCOUL.

Remarque: dans le cas où la relation est fonctionnelle dans un sens au moins, on pourrait ne conserver qu'un repère (celui situé à l'origine du sens fonctionnel) comme O.E. On gagnerait en ce qui concerne le volume des O.E. mais on perdrait probablement en nombre d'accès (dans beaucoup de cas il faudrait accéder à nouveau au second repère du couple).

+ dans le deuxième cas, où la relation PIVOINES traduisant l'existence du constituant commun n'existe pas, on peut écrire:

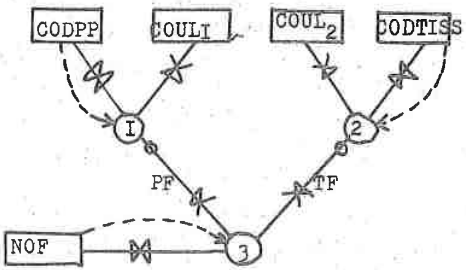
$$S = R_I [X] R_2 = (R_I \text{ ✻ } R_2) [X_I = X_2]$$

(X_I et X_2 sont les constituants respectifs de R_I et R_2 définis sur le même domaine, selon lesquels est réalisée la jointure).

Il suffit donc de prendre en compte tous les couples possibles de repères de (i) et (j) et de sélectionner ceux associés à la même valeur de X.

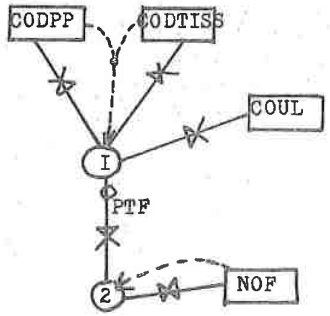
exemple:

au niveau PHYLOG:



{ I : papier-peint
2 : tissu
3 : fournisseur

au niveau LOG:



{ I : "coordonné"(cf.ex. précédent)
2 : fournisseur

ERPHYLOG :

PAP(CODPP, $COUL_I$, NOF)

TISS(CODTISS, $COUL_2$, NOF)

FOURN(NOF,.....)

ERLOG :

COORD(CODPP, CODTISS, COUL, NOF)

FOURN2(NOF,.....)

correspondance :

FOURN2 = FOURN

COORD = (PAP ✱ TISS) : $COUL_I$ = $COUL_2$

Les objets équivalents aux repères "coordonné" sont des couples de repères de (1) et (2) associés aux mêmes valeurs de couleur.

✚ Dans le cas particulier où une des relations R_I ou R_2 serait associée à une classe de relations et non à une classe de repères, l'interprétation des jointures serait la même, mais s'exercerait sur des couples (ou des n-uples) de repères liés par la relation considérée.

3.4.4. Interprétation de la division :

Soit : $S = R_I [A, X \textcircled{\div} B] R_2$

Dans le cas général, R_I et R_2 sont associées à des types de repères (i) et (j) de PHYLOG.

D'après la définition même de la division (cf. 3.2.3), à chaque occurrence de S correspond un repère du type (i) associé à une valeur x de X telle que, l'ensemble des valeurs de A pour tous les repères (i) associés à x, contienne toutes les valeurs de B associées aux repères du type (j).

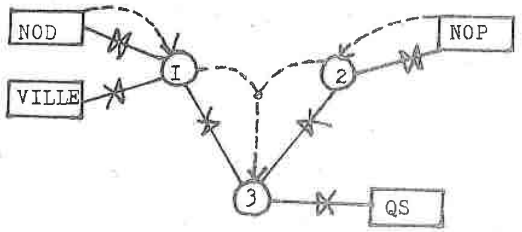
La détermination des repères associés aux occurrences de S nécessite donc :

a) l'accès de (j) à toutes les valeurs possibles de B
b) la constitution de tous les ensembles de repères (i), associés aux mêmes valeurs de X
c) la sélection des ensembles où toutes les valeurs possibles de B sont contenues dans l'ensemble des valeurs de A associées aux repères (i) de ces ensembles.

Les O.E. sont donc des repères du type (i) sélectionnés par la démarche en trois points ci-dessus.

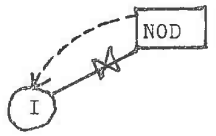
exemple :

au niveau PHYLOG :



{ I : dépôt
2 : pièce
3 : stock

au niveau LOG :



{ I : dépôt contenant toutes les pièces

{ NOD : n° dépôt
NOP : n° pièce
QS : qté en stock

ERPHYLOG :

DEPOT(NOD, VILLE)

PIECE(NOP)

STOCK(NOD, NOP, QS)

ERLOG :

DTP(NOD)

correspondance :

DTP = (STOCK [NOP, NOD $\textcircled{\div}$ NOP] PIECE)[NOD]

Au niveau des occurrences :

STOCK	NOD	NOP	QS	repères ③
	I0	I	I00	r_1
	II	I	I05	r_2
	II	2	I	r_3
	II	3	875	r_4
	I2	2	20	r_5
	I2	3	85	r_6
	I3	I	7	r_7
	I3	2	85	r_8
	I3	3	I	r_9

PIECE	NOP	repères ②
	I	s_1
	2	s_2
	3	s_3

DEPOT	NOD	VILLE	repères ①
	I0	METZ	d_1
	II	NANCY	d_2
	I2	METZ	d_3
	I3	TOUL	d_4

Si l'on reprend les 3 points de la détermination des repères associés à DTP, on a :

a) $\{I,2,3\}$ (toutes les valeurs possibles de NOP associées aux repères ②).

b) $\{r_1\}$ associé à NOD = I0 ⇒ $\{I\}$ pour NOP
$\{r_2,r_3,r_4\}$ " = II ⇒ $\{I,2,3\}$ "
$\{r_5,r_6\}$ " = I2 ⇒ $\{2,3\}$ "
$\{r_7,r_8,r_9\}$ " = I3 ⇒ $\{I,2,3\}$ "

c) r_2,r_3,r_4,r_7,r_8,r_9 sont donc les 6 occurrences associées à (STOCK[NOP,NOD ÷ NOP]PIECE)

d) $\{r_2,r_3,r_4\}$, $\{r_7,r_8,r_9\}$ sont donc les 2 occurrences associées à DTP

DTP	NOD
	II
	I3

3.4.5. Interprétation des opérations ensemblistes classiques :

Nous avons déjà traité le cas du produit cartésien au paragraphe 3.4.3.I. Les trois autres opérations (union, intersection, différence) opèrent sur des relations de même nature (c.à.d. comportant des constituants deux à deux union-compatibles). Elles sont dans le cas général associées à des types de repères (i) et (j). Les interprétations en termes d'O.E. sont immédiates :

+ soit $S = R_1 \circ R_2$; à chaque occurrence de S correspond un couple de repères ((i), (j)) dont les valeurs de tous les constituants sont identiques.
+ soit $S = R_1 + R_2$; à chaque occurrence de S correspond soit un couple ((i), (j)) dont les valeurs de tous les constituants sont identiques, soit un repère (i), soit un repère (j) quand il n'y a pas de repères des 2 types dont les valeurs de tous les constituants soient identiques.
+ soit $S = R_1 - R_2$; à chaque occurrence de S correspond un repère (i) n'ayant pas de "semblable" dans (j) (c.à.d. de repère associé aux mêmes valeurs pour tous leurs constituants).

3.4.6. Conclusion :

Dans ce paragraphe 3, nous avons mis en évidence les O.E. que l'on peut associer au résultat d' une opération de l'algèbre relationnelle. Il nous reste à étudier les O.E. que l'on peut associer au résultat d'une expression relationnelle complexe, comportant plusieurs opérations . Après avoir analysé l'intérêt, dans le contexte de notre problème, des différentes opérations nous procèderons en deux étapes :

+ dans la première étape nous nous restreindrons à une algèbre réduite comprenant la sélection, la projection, la jointure naturelle (et donc le produit cartésien). Nous montrerons qu'il s'agit des opérations de très loin les plus importantes dans le contexte de notre problème. Puis nous déterminerons la forme générale des O.E. associés au résultat d'une expression quelconque de l'algèbre réduite.
+ dans la seconde étape nous considèrerons la division et les opérations ensemblistes non encore étudiées(union,intersection,différence). Nous montrerons dans quels cas particuliers leur utilisation est nécessaire et nous résoudrons ces cas particuliers.

- - - - -

3.5. Les objets équivalents au résultat d'une expression quelconque de l'algèbre réduite :

3.5.I. Justification de l'algèbre réduite :

L'algèbre réduite comporte la sélection, la projection, la jointure naturelle. Ont donc étées laissées de côté, la division et les opérations ensemblistes union, intersection et différence.
En ce qui concerne la division on peut noter tout d'abord qu' ISBL, le langage de manipulation algébrique de PRTV, ne l' intègre pas. D'autres pensent ([3.4]) que « l'utilité de la division est surtout théorique : démonstration de l'inclusion de l'algèbre relationnelle dans le calcul relationnel ».
Dans le contexte de notre problème, et sous une forme un peu modifiée, nous avons montré son utilité pour définir un type de repères logiques "incluant" un quantificateur universel dans sa définition (cf. ex.$_4$ du paragraphe 3.3). Nous nous restreindrons à cette seule forme (un seul niveau de quantification) que nous étudierons par la suite.
L'union, l'intersection et la différence ne peuvent opérer que sur des relations de même nature("union-compatibles"). Les expressions relationnelles ne contiennent pas de telles relations. Le seul cas plausible est celui illustré par l'exemple $_6$ du paragraphe 3.3 : on obtient deux relations de même nature à partir de deux relations ayant des constituants communs, par projection sur ces constituants communs. Là également, nous nous restreindrons à cette seule forme que nous étudierons dans la suite.
Les autres cas, s'ils se présentaient devraient faire l'objet d'une étude spécifique.
Pour les opérations de l'algèbre réduite nous n'apporterons aucune restriction : elles doivent permettre à elles seules de caractériser la très grande majorité des repères logiques.

3.5.2. La simplification et l'optimisation des expressions de l'algèbre réduite :

3.5.2.I. La composition des opérations élémentaires :

Nous allons présenter un certain nombre de propriétés de l' algèbre réduite, rendant possible une simplification et une optimisation des expressions de cette algèbre (cf.[3.6,3.5]).
Par simplification nous entendons réduction du nombre d'

opérations dans l'expression et par optimisation, minimisation du volume des relations résultats intermédiaires (par diminution de l'arité ou de la cardinalité). Ce sont des objectifs qui peuvent être contradictoires : certaines optimisations entraînent une augmentation du nombre des opérations.
Notre objectif est d'obtenir une forme générale optimisée des expressions de l'algèbre réduite. L'optimisation nous aidera également à améliorer la mise en oeuvre des procédures qui construiront les O.E.(cf.3.7).

a) composition des sélections : une séquence de sélections sur une relation R peut être remplacée par une sélection unique dont le filtre est la conjonction des filtres originaux. Cela constitue à la fois une simplification et une optimisation(suppression des résultats intermédiaires) :

$$(\ldots..((R:f_I):f_2)\ldots):f_n = R : f_I \wedge f_2 \wedge \ldots. \wedge f_n$$

b) composition des projections : une séquence de projections sur une relation R se simplifie de la manière suivante :

$$(\ldots..((R[L_I])[L_2])\ldots.)[L_n] = \begin{cases} R[L_n] \text{ si } L_n \subset L_{n-I} \subset \ldots \subset L_I \\ \emptyset \text{ sinon} \end{cases}$$

L_i : liste de constituants

c) priorité des sélections sur les projections :
Soit une projection et une sélection portant sur une même relation. Il est toujours possible d'effectuer la sélection avant d'effectuer la projection. En effet, lorsque $R[L]:f$ existe on a :

$$R[L]:f \Rightarrow (R:f)[L]$$

Il s'agit d'une optimisation si la sélection a pour conséquence une forte diminution de la cardinalité de R (a priori importante dans les applications réelles), entraînant une diminution de volume plus importante que la projection sur L. Ce doit être le cas le plus fréquent (la même hypothèse est retenue dans [3.6]).
exemple :

PERSONNE

NOM	PRENOM	VILLE	AGE	NB.ENF
ALFRED	LEON	METZ	30	0
DUPONT	JOE	NANCY	25	2
MACHIN	HUBERT	METZ	33	I
TRUC	JACQUES	NANCY	I7	0
BIDULE	LOUIS	PARIS	58	2
WILSON	JOHN	LONDRES	45	4
ZWIFFT	HELMUTT	BERLIN	74	I
PAUL	PIERRE	TOUL	28	0

PERSONNE [NOM , PRENOM , AGE] ⇒ (PERSONNE [NOM , PRENOM , AGE]) : AGE > 40

ALFRED	LEON	30
DUPONT	JOE	25
MACHIN	HUBERT	33
TRUC	JACQUES	I7
BIDULE	LOUIS	58
WILSON	JOHN	45
ZWIFFT	HELMUTT	74
PAUL	PIERRE	28

BIDULE	LOUIS	58
WILSON	JOHN	45
ZWIFFT	HELMUTT	74

PERSONNE : AGE > 40 ⇒ (PERSONNE : AGE > 40) [NOM, PRENOM, AGE]

BIDULE	LOUIS	PARIS	58	2
WILSON	JOHN	LONDRES	45	4
ZWIFFT	HELMUT	BERLIN	74	I

BIDULE	LOUIS	58
WILSON	JOHN	45
ZWIFFT	HELMUTT	74

En conséquence, dans une expression ne comportant que des projections et des sélections, il est possible d'effectuer une sélection unique suivie d'une projection unique :

$((((R:f_I)[L_I]):f_2)[L_2]):f_3$ (avec: $L_2 \subset L_I$ et f_2 portant sur L_I et f_3 portant sur L_2).

$((((R:f_I):f_2)[L_I])[L_2]):f_3$

$(((R:f_I \wedge f_2)[L_I]):f_3)[L_2]$

$(((R:f_I \wedge f_2):f_3)[L_I])[L_2]$

$((R:f_I \wedge f_2 \wedge f_3)[L_I])[L_2]$

$(R:f_I \wedge f_2 \wedge f_3)[L_2]$

d) priorité des sélections sur les jointures naturelles (et le produit cartésien comme cas particulier) :

Soient deux relations R et S et leur jointure selon L (constituant commun ou ∅). Une sélection porte sur cette jointure.

On a: $(R[L]S):f \Rightarrow ((R:f_I)[L](S:f_2)):f_3$

avec:
- $f = f_I \wedge f_2 \wedge f_3$
- f_I est la partie de f (éventuellement vide) agissant sur les seuls constituants de R (y compris L)
- f_2 est la partie de f (éventuellement vide) agissant sur les seuls constituants de S (y compris L)
- f3 est la partie de f agissant sur les constituants de R et de S

IL est donc possible en général d'effectuer en priorité tout ou partie des sélections en les décomposant (le seul cas où l'on ne peut rien faire correspond à $f = f_3$); il s'agit dans presque tous les cas d'une optimisation (bien que le nombre d'opérations puisse augmenter).

exemples : CLIENT

NOCL	ADRESSE	NOM
I00	METZ	DUPONT
I0I	METZ	DUPOND
I02	NANCY	DURAND
I03	NANCY	TOTO
I04	METZ	MACHIN
I05	TOUL	DUPONT

CDE

NOCL	NOP	QTE
I00	I	I0
I00	2	I
I0I	4	I50
I03	I	3
I04	I	2
I04	2	I5
I04	3	I00
I05	I	I
I05	3	I

α) (CLIENT[NOCL]CDE):ADRESSE="METZ" ⇒ (CLIENT:ADRESSE="METZ") [NOCL] CDE

CLIENT:ADRESSE="METZ" ⇒ (CLIENT:ADRESSE="METZ")[NOCL]CDE

I00	METZ	DUPONT
I0I	METZ	DUPOND
I04	METZ	MACHIN

I00	METZ	DUPONT	I	I0
I00	METZ	DUPONT	2	I
I0I	METZ	DUPOND	4	I50
I04	METZ	MACHIN	I	2
I04	METZ	MACHIN	2	I5
I04	METZ	MACHIN	3	I00

CLIENT[NOCL]CDE ⇒ (CLIENT[NOCL]CDE):ADRESSE="METZ"

I00	METZ	DUPONT	I	I0
I00	METZ	DUPONT	2	I
I0I	METZ	DUPOND	4	I50
I03	NANCY	TOTO	I	3
I04	METZ	MACHIN	I	2
I04	METZ	MACHIN	2	I5
I04	METZ	MACHIN	3	I00
I05	TOUL	DUPONT	I	I
I05	TOUL	DUPONT	3	I

I00	METZ	DUPONT	I	I0
I00	METZ	DUPONT	2	I
I0I	METZ	DUPOND	4	I50
I04	METZ	MACHIN	I	2
I04	METZ	MACHIN	2	I5
I04	METZ	MACHIN	3	I00

β) (CLIENT[NOCL]CDE):NOCL > I02 ∧ (NOP=I ∨ NOP=2) ⇒ (CLIENT:NOCL > I02)[NOCL](CDE:NOCL>I02 ∧ (NOP=I ∨ NOP=2))

CLIENT : NOCL > I02

I03	NANCY	TOTO
I04	METZ	MACHIN
I05	TOUL	DUPONT

CDE : NOCL > I02 ∧ (NOP=I ∨ NOP=2)

I03	I	3
I04	I	2
I04	2	I5
I05	I	I

(CLIENT:NOCL>I02)[NOCL] (CDE:NOCL > I02 ∧ (NOP=I ∨ NOP=2)) ⇒

I03	NANCY	TOTO	I	3
I04	METZ	MACHIN	I	2
I04	METZ	MACHIN	2	I5
I05	TOUL	DUPONT	I	I

CLIENT[NOCL]CDE (cf. p.3.27)

(CLIENT[NOCL]CDE):NOCL>I02∧(NOP=I∨NOP=2)

I03	NANCY	TOTO	I	3
I04	METZ	MACHIN	I	2
I04	METZ	MACHIN	2	I5
I05	TOUL	DUPONT	I	I

δ) (CLIENT[NOCL]CDE): (ADRESSE="METZ") ∨(QTE>I00), ne peut être transformé ($f = f_3$)

ε) priorité des projections sur les jointures naturelles:
Soient deux relations R et S, leur jointure naturelle selon L (constituant commun ou ∅) et une projection de la jointure sur une liste de constituants P :

$$(R[L]S)[P] \Rightarrow \begin{cases} (R[PI])[L](S[P2]) & \text{si } L \subset P \\ ((R[PI])[L](S[P2]))[P] & \text{sinon} \end{cases}$$

où { PI comprend L et les éléments de P figurant dans R
{ P2 " " " " S

Il est donc possible d'effectuer en priorité tout ou partie des projections; il s'agit dans presque tous les cas d'une optimisation bien que le nombre des opérations soit en augmentation.

exemples : on reprend les relations de l'exemple précédent

(CDE[NOCL]CLIENT)[NOCL,NOM,NOP] ⇒ (CDE[NOCL,NOP])[NOCL] (CLIENT[NOCL,NOM])

CDE[NOCL]CLIENT (cf. p.3.27)

(CDE[NOCL]CLIENT)[NOCL,NOM,NOP]

I00	DUPONT	I
I00	DUPONT	2
I0I	DUPOND	4
I03	TOTO	I
I04	MACHIN	I
I04	MACHIN	2
I04	MACHIN	3
I05	DUPONT	I
I05	DUPONT	3

CDE[NOCL,NOP]

I00	I
I00	2
I0I	4
I03	I
I04	I
I04	2
I04	3
I05	I
I05	3

CLIENT[NOCL,NOM]

I00	DUPONT
I0I	DUPOND
I02	DURAND
I03	TOTO
I04	MACHIN

⇒ (CDE[NOCL,NOP])[NOCL](CLIENT[NOCL,NOM])

I00	DUPONT	I
I00	DUPONT	2
I0I	DUPOND	4
I03	TOTO	I
I04	MACHIN	I

I04	MACHIN	2
I04	MACHIN	3
I05	DUPONT	I
I05	DUPONT	3

3.5.2.2. La transformation des expressions relationnelles :
Introduisons une représentation en arbre des expressions, selon les définitions ci-dessous.

+ représentation des opérations :

$R : f_I$ — $S(f_I)$ — R

où { S indique la nature de l'opération (sélection)
{ le lien vertical donne le (ou les)opérande(s)
{ les rectangles entourent les noms des relations

R[A,B,C] — P(A,B,C) — R

où { P indique la nature de l'opération (projection)

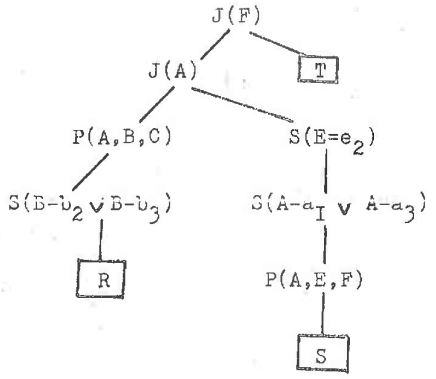
R[A]S — J(A) — R, S

où { J indique la nature de l'opération (jointure)

+ représentation des expressions : les compositions se traduisent par des liens verticaux.

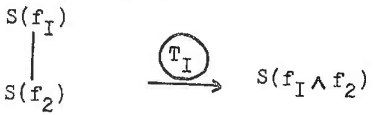
exemple : $(((R:B=b_2 \vee B=b_3)[A,B,C])[A](((S[A,E,F]):A=a_I \vee A=a_3):E=e_2))[F]T$

peut être représentée par l'arbre suivant :

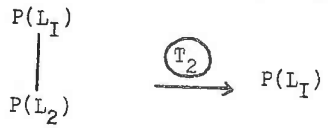


Les propriétés des paragraphes précédents (a,b,c,d,e) peuvent être traduites en transformations élémentaires des arbres :

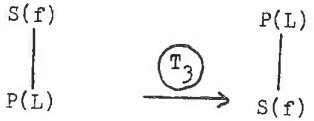
+ composition des sélections :



+ composition des projections : $(L_I \subset L_2)$

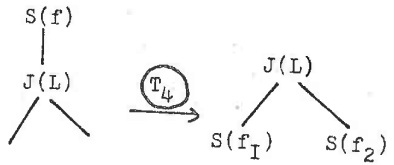


+ priorité des sélections sur les projections :

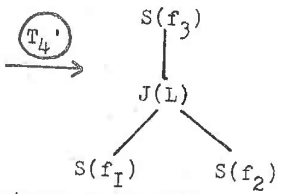


+ priorité des sélections sur les jointures naturelles :

• si f est séparable en f_I et f_2 portant exclusivement sur les opérandes I et 2 ($f = f_I \wedge f_2$)

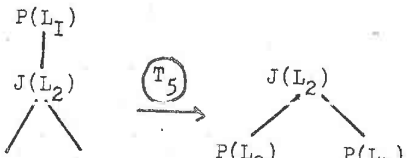


• sinon ($f = f_I \wedge f_2 \wedge f_3$ portant respectivement sur les opérandes I,2 et sur la jointure)

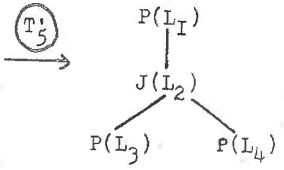


+ priorité des projections sur les jointures naturelles:

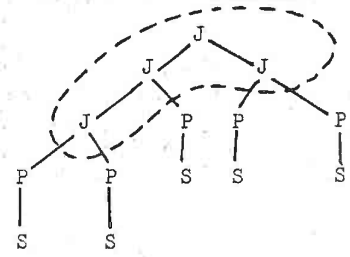
• si $L_2 \subset L_I$



• si $L_2 \not\subset L_I$



La composition de ces transformations élémentaires permet de modifier l'arbre (et donc l'expression relationnelle) : les jointures "montent" dans l'arbre, alors que les projections et les sélections "descendent" et s'agrègent. Ces agrégations conduisent en général à une diminution du nombre global des opérations, l'ensemble de l'expression étant par ailleurs optimisée. Tout arbre se réduit donc, quel que soit sa complexité, en un arbre de jointures (avec éventuellement des projections et/ou des sélections "résiduelles" dûes aux transformations T'_4 et T'_5), les "feuilles" de cet arbre étant constituées d'une projection et d'une sélection uniques.

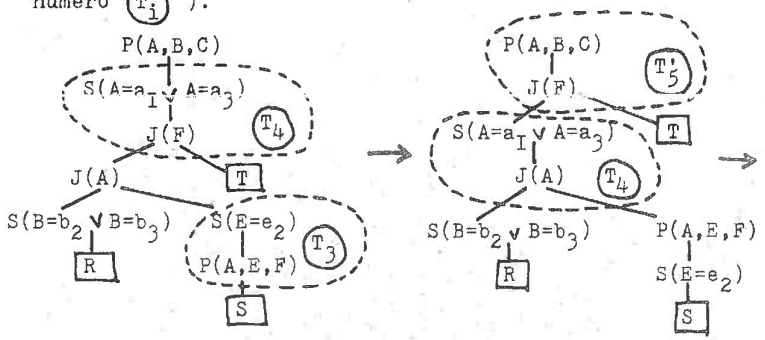


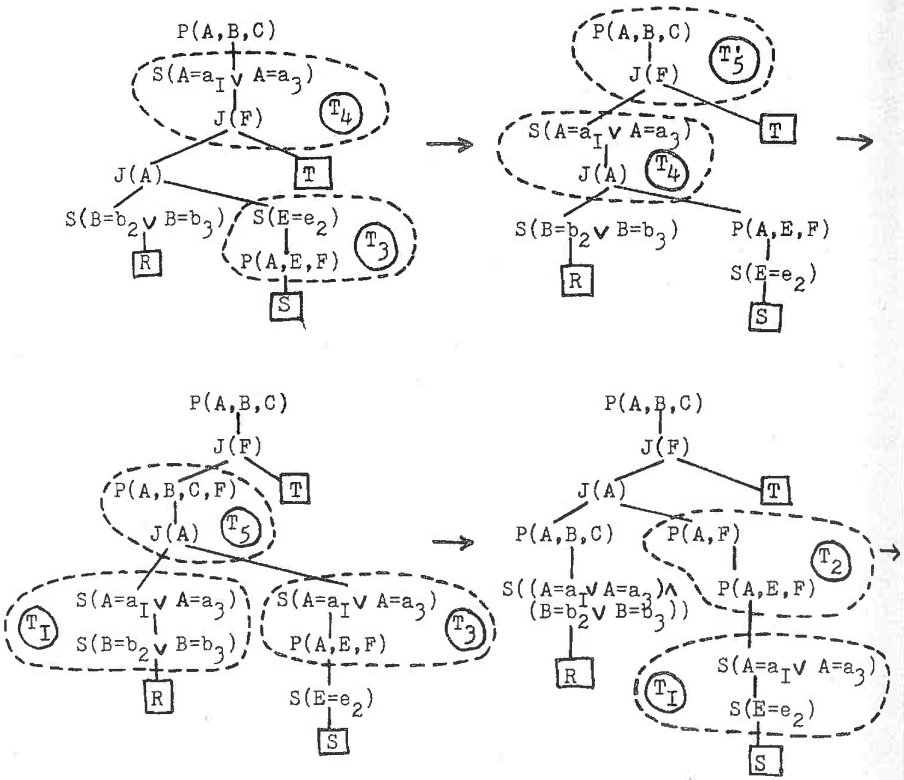
arbre de jointures

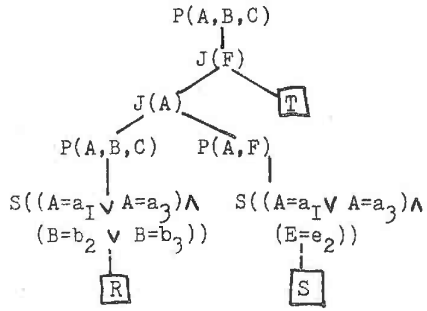
exemple : soient : $\begin{cases} R(A,B,C,D) \\ S(A,E,F,K) \\ T(F,G,H) \end{cases}$

et : $((((R{:}B=b_2 \vee B=b_3)\,[A]\,(S[A,E,F]{:}E=e_2))\,[F]\,T)$
$:A=a_I \vee A=a_3)\,[A,B,C]$

A cette expression correspondent l'arbre et les transformations suivantes(sont entourées les diverses transformations élémentaires opérées successivement sur l'arbre avec leur numéro T_i).







Ce qui correspond à la forme générale annoncée et se retranscrit en :
((((R:(A=a_I∨A=a_3)∧(B=b_2∨ B=b_3))[A,B,C])[A]((S:(A=a_I∨A=a_3) ∧ (E=e_2))[A,F])) [F] T) [A,B,C]

On peut noter que le nombre global d'opérations est resté constant. Par contre il y a eu optimisation: vérifions-la, ainsi que la validité de la transformation, au niveau des occurrences.

Soient: R

A	B	C	D
a_I	b_I	c_I	d_I
a_2	b_2	c_I	d_2
a_3	b_2	c_I	d_2
a_3	b_3	c_2	d_I
a_3	b_3	c_2	d_3
a_4	b_5	c_3	d_I

S

A	E	F	K
a_I	e_I	f_I	k_2
a_2	e_I	f_2	k_I
a_3	e_2	f_I	k_3
a_3	e_2	f_I	k_4

T

F	G	H
f_I	g_I	h_I
f_I	g_I	h_2
f_3	g_2	h_I
f_4	g_2	h_2

Evaluons l'expression initiale :

R:B=b_2 ∨B=b_3

a_2	b_2	c_I	d_2
a_3	b_2	c_2	d_2
a_3	b_3	c_2	d_I
a_3	b_3	c_2	d_3

S[A,E,F]

a_I	e_I	f_I
a_2	e_I	f_2
a_3	e_2	f_I

S[A,E,F] : E=e_2

a_3	e_2	f_I

(R:B=b_2 ∨B=b_3)[A](S[A,E,F]:E=e_2) = X

a_3	b_2	c_I	d_2	e_2	f_I	
a_3	b_3	c_2	d_I	e_2	f_I	
a_3	b_3	c_2	d_3	e_2	f_I	

X[F]T

a_3	b_2	c_I	d_2	e_2	f_I	g_I	h_I
a_3	b_2	c_I	d_2	e_2	f_I	g_I	h_2
a_3	b_3	c_2	d_I	e_2	f_I	g_I	h_I
a_3	b_3	c_2	d_I	e_2	f_I	g_I	h_2
a_3	b_3	c_2	d_3	e_2	f_I	g_I	h_I
a_3	b_3	c_2	d_3	e_2	f_I	g_I	h_2

(X[F]T):A=a_I∨ A=a_3 = X[F] T
((X[F] T): A=a_I∨ A=a_3)[A,B,C]

a_3	b_2	c_I
a_3	b_3	c_2

qui est le résultat cherché. Le volume des résultats intermédiaires est ici de I45 valeurs.

Evaluons l'expression finale :

R:(A=a_I ∨ A=a_3) ∧(B=b_2 ∨ B=b_3) = Y

a_3	b_2	c_1	d_2
a_3	b_3	c_2	d_I
a_3	b_3	c_2	d_3

Y[A,B,C]

a_3	b_2	c_I
a_3	b_3	c_2

S:(A=a_I∨ A=a_3) ∧ (E=e_2) = W

a_3	e_2	f_I	k_3
a_3	e_2	f_I	k_4

W [A,F]

a_3	f_I

(Y[A,B,C])[A](W[A,F]) = Z

a_3	b_2	c_I	f_I
a_3	b_3	c_2	f_I

Z [F] T

a_3	b_2	c_I	f_I	g_I	h_I
a_3	b_2	c_I	f_I	g_I	h_2
a_3	b_3	c_2	f_I	g_I	h_I
a_3	b_3	c_2	f_I	g_I	h_2

(Z [F] T) [A,B,C]

a_3	b_2	c_I
a_3	b_3	c_2

qui est le résultat cherché. Le volume des résultats intermédiaires est ici ramené à 60 valeurs (gain de 60 % environ)

3.5.3. Les objets équivalents :

Pour déterminer les O.E. au résultat d'une expression quelconque de l'algèbre réduite, il nous suffit d' "interpréter" en termes de PIVOINES la composition de jointures opérant chacune sur le résultat d'une sélection et d'une projection (avec éventuellement des projections et des sélections"résiduelles"-cf. 3.5.2) .
La sélection et la projection donnent des ensembles de repères. Les algorithmes de construction de ces ensembles ont été présentés aux paragraphes 3.4.I. et 3.4.2.
Considérons la jointure de relations ainsi obtenues par sélection et projection. A chaque occurrence de la jointure sera associé un ensemble de couples de repères, ces repères étant pris dans les ensembles associés aux occurrences des 2 relations.
Dans le cas du produit cartésien, on prendra tous les ensembles de repères 2 à 2 pour construire les ensembles de couples.
Dans le cas de la jointure naturelle selon un constituant commun, on ne prendra que les ensembles de repères associés aux mêmes valeurs du constituant commun. Il s'agit de la simple généralisation du paragraphe 3.4.3. à des ensembles de repères.

exemple :

R

A	B	C	repères
a_I	b_I	c_I	r_I
a_I	b_2	c_I	r_2
a_2	b_I	c_I	r_3
a_3	b_2	c_2	r_4
a_2	b_2	c_I	r_5
a_4	b_2	c_I	r_6
a_I	b_2	c_2	r_7

S

A	D	repères
a_I	d_I	r'_I
a_I	d_2	r'_2
a_2	d_I	r'_3
a_I	d_I	r'_4

R [A , C] objets équivalents

a_I	c_I	$\rightarrow \{r_I, r_2\}$
a_2	c_I	$\rightarrow \{r_3, r_5\}$
a_3	c_2	$\rightarrow \{r_4\}$
a_4	c_I	$\rightarrow \{r_6\}$
a_I	c_2	$\rightarrow \{r_7\}$

S [A] objets équivalents

a_I	$\rightarrow \{r'_I, r'_2, r'_4\}$
a_2	$\rightarrow \{r'_3, r'_5\}$

T(A,C) = (R[A,C])[A](S[A]) objets équivalents

a_I	c_I	$\rightarrow \{(r_I,r'_I),(r_I,r'_2),(r_I,r'_4),(r_2,r'_I),(r_2,r'_2),(r_2,r'_4)\}$
a_2	c_I	$\rightarrow \{(r_3,r'_3),(r_3,r'_5),(r_5,r'_3),(r_5,r'_5)\}$
a_I	c_2	$\rightarrow \{(r_7,r'_I),(r_7,r'_2),(r_7,r'_4)\}$

Dans les ensembles de couples, tous les repères d'un même type sont associés, dans les couples d'un même ensemble, aux mêmes valeurs des constituants.
Une sélection à ce niveau (sélection "résiduelle") entraînerait la suppression globale de certains ensembles de couples de repères.
Une projection à ce niveau (projection "résiduelle") conduirait à regrouper dans un même ensemble des ensembles de couples de repères associés aux mêmes valeurs des constituants de la projection.
Dans les deux cas, la nature des O.E. ne change pas.

exemples: sur l'exemple précédent

+ soit T(A,C):C=c_I objets équivalents

a_I	c_I	$\rightarrow \{(r_I,r'_I),(r_I,r'_2),(r_I,r'_4),(r_2,r'_I),(r_2,r'_2)(r_2,r'_4)\}$
a_2	c_I	$\rightarrow \{(r_3,r'_3),(r_3,r'_5),(r_5,r'_3),(r_5,r'_5)\}$

Il y a eu suppression en bloc du troisième ensemble de couples correspondant à (a_I, c_2)

+ soit T(A,C)[C] objets équivalents

c_I	$\rightarrow$	$\{(r_I,r'_I),(r_I,r'_2),(r_I,r'_4),(r_2,r'_I),(r_2,r'_2),(r_2,r'_4),(r_3,r'_4),(r_3,r'_5),(r_5,r'_3),(r_5,r'_5)\}$
c_2	$\rightarrow$	$\{(r_7,r'_I),(r_7,r'_2),(r_7,r'_4)\}$

Il y a eu union des 2 premiers ensembles de couples associés tous deux à la valeur c_I de C.

La généralisation à un arbre de jointures conduit donc à des objets équivalents ensembles de n-uples de repères. Ils constituent les O.E. les plus complexes que l'on puisse associer aux expressions de l'algèbre réduite. Les conséquences "économiques" de leur utilisation seront examinées plus loin. Nous allons illustrer ces résultats sur les exemples I,2,3,5 des paragraphes 2.5 et 3.3 (qui mettent en oeuvre les opérations de l'algèbre réduite).

Exemple I : les O.E. sont les repères du type (I) de PHYLOG liés par la relation CE à la valeur "cadre" de CATEG.

Exemple 2 : les O.E. sont des ensembles de repères du type (I) de PHYLOG liés par la relation FV à la même valeur de VILLE.

Exemple 3 : à la relation A-P-EP de ERPHYLOG on peut associer les couples de repères ((I),(I)) liés par la relation PIVOINES A POUR EPOUSE. La jointure consiste alors à associer à ces couples, les repères (I) liés à la même valeur de NOM que le premier repère du couple. Comme NOM est l'identifiant de (I), il n'y en a qu'un seul qui est le premier repère du couple lui-même. Les O.E. peuvent donc ici se simplifier : il s'agit uniquement des couples de repères ((I),(I)) de PHYLOG liés par la relation A POUR EPOUSE.

Exemple 5 : considérons tout d'abord les O.E. des repères logiques (5). Il s'agit des couples de repères de PHYLOG ((4),(5)) liés par la relation PIVOINES LCC.

Considérons ensuite les O.E. des repères logiques (2). Il s'agit d'ensembles de couples de repères ((4),(5)), associés aux mêmes valeurs de NOCL, JOUR, MOIS, AN et NOP.

Vérifions ces résultats au niveau des occurrences :

ERLOG :

CLIENT2

NOCL	ADR	NOM
I00	METZ	TOTO
I01	NANCY	DUPONT
I02	TOUL	JOE

DATE2

JOUR	MOIS	AN
I0	I	I978
I1	I	I978
I3	I	I978

PROCLID

NOCL	JOUR	MOIS	AN	NOP
I00	I3	I	I978	I0
I00	I3	I	I978	I1
I01	I0	I	I978	I0
I01	I0	I	I978	I2
I02	I1	I	I978	I1
I02	I0	I	I978	I2

CLJ

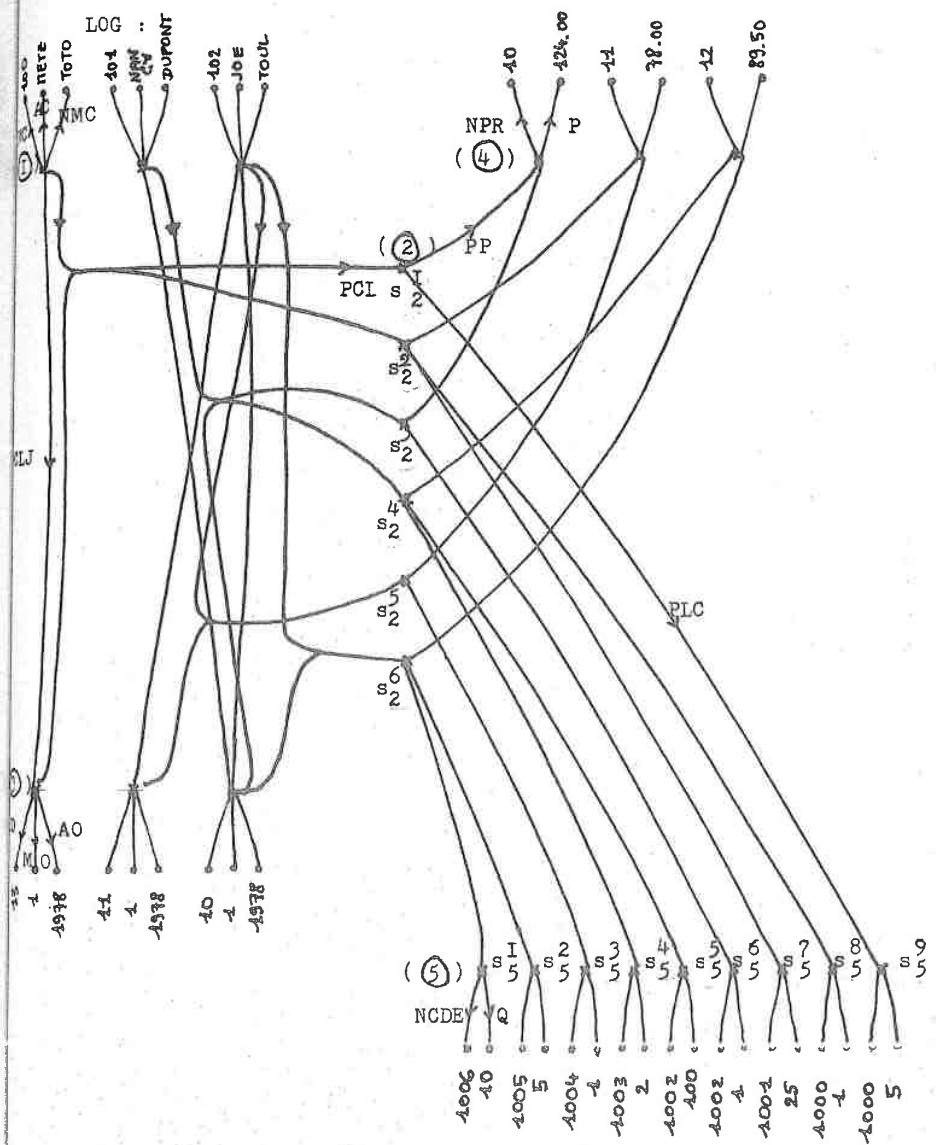
NOCL	JOUR	MOIS	AN
I00	I3	I	I978
I01	I0	I	I978
I02	I1	I	I978
I02	I0	I	I978

PRODUIT2

NOP	PXU
I0	I24.00
I1	78.00
I2	89.50

LC

NOC	NOP	NOCL	JOUR	MOIS	AN	QTE
I000	I0	I00	I3	I	I978	5
I000	I1	I00	I3	I	I978	1
I001	I1	I00	I3	I	I978	25
I002	I0	I01	I0	I	I978	1
I002	I2	I01	I0	I	I978	I00
I003	I2	I01	I0	I	I978	2
I004	I1	I02	I1	I	I978	1
I005	I2	I02	I0	I	I978	5
I006	I2	I02	I0	I	I978	I0



ERPHYLOG :

CLIENT = CLIENT2

DATE = DATE2

PRODUIT = PRODUIT2

DATE-CLT = CLJ

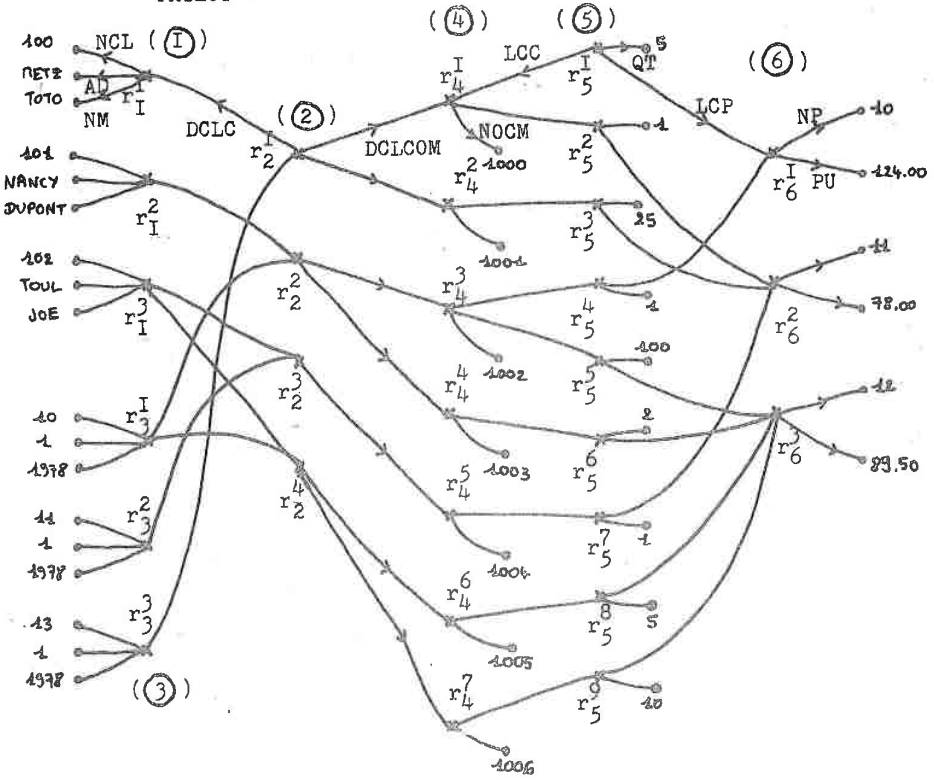
CDE

NOC	NOCL	JOUR	MOIS	AN
I000	I00	I3	I	I978
I00I	I00	I3	I	I978
I002	I0I	I0	I	I978
I003	I0I	I0	I	I978
I004	I02	II	I	I978
I005	I02	I0	I	I978
I006	I02	I0	I	I978

LIGNE

NOC	NOP	QTE
I000	I0	5
I000	II	I
I00I	II	25
I002	I0	I
I002	I2	I00
I003	I2	2
I004	II	I
I005	I2	5
I006	I2	I0

PHYLOG :



Objets équivalents :

repères LOG	LC							objets équivalents
s_5^1 →	I006	I2	I02	I0	I	I978	I0	$\longrightarrow (r_5^9, r_6^3)$
s_5^2 →	I005	I2	I02	I0	I	I978	5	$\longrightarrow (r_5^8, r_6^3)$
s_5^3 →	I004	II	I02	II	I	I978	I	$\longrightarrow (r_5^7, r_6^2)$
s_5^4 →	I003	I2	I0I	I0	I	I978	2	$\longrightarrow (r_5^6, r_6^3)$
s_5^5 →	I002	I2	I0I	I0	I	I978	I00	$\longrightarrow (r_5^5, r_6^3)$
s_5^6 →	I002	I0	I0I	I0	I	I978	I	$\longrightarrow (r_5^4, r_6^I)$
s_5^7 →	I00I	II	I00	I3	I	I978	25	$\longrightarrow (r_5^3, r_6^2)$
s_5^8 →	I000	II	I00	I3	I	I978	I	$\longrightarrow (r_5^2, r_6^2)$
s_5^9 →	I000	I0	I00	I3	I	I978	5	$\longrightarrow (r_5^I, r_6^I)$

repères LOG	PROCLID					objets équivalents
s_2^I →	I00	I3	I	I978	I0	$\longrightarrow \{(r_5^I, r_6^I)\}$
s_2^2 →	I00	I3	I	I978	II	$\longrightarrow \{(r_5^2, r_6^2), (r_5^3, r_6^2)\}$
s_2^3 →	I0I	I0	I	I978	I0	$\longrightarrow \{(r_5^4, r_6^I)\}$
s_2^4 →	I0I	I0	I	I978	I2	$\longrightarrow \{(r_5^5, r_6^3), (r_5^6, r_6^3)\}$
s_2^5 →	I02	II	I	I978	II	$\longrightarrow \{(r_5^7, r_6^2)\}$
s_2^6 →	I02	I0	I	I978	I2	$\longrightarrow \{(r_5^8, r_6^3), (r_5^9, r_6^3)\}$

- - - - -

3.6. L' élargissement de l'algèbre :

Dans ce paragraphe, nous réintroduisons la division ainsi que l' union, l'intersection et la différence, dans les conditions restrictives exposées au paragraphe 3.5.I.

3.6.I. L' utilisation de la division :

Elle permet la définition de types de repères logiques "incluant" un quantificateur universel. Nous avons montré au paragraphe 3.4.4. l'effet d'une division s'appliquant à des relations auxquelles sont associées des types de repères; la division conduisait à la sélection d'un sous-ensemble de repères associés au "dividende", selon un processus assez complexe. Dans le cas le plus général, la division opère sur des relations auxquelles sont associés des ensembles de n-uples de repères. Dans chacun de ces ensembles, tous les repères d' un même type sont associés aux mêmes valeurs des constituants des relations. La sélection qu'opère la division conduira donc à rejeter en bloc certains de ces ensembles et à en conserver d'autres. La nature des O.E. n'est donc pas modifiée. Il faut remarquer que les accès nécessaires pour effectuer cette sélection peuvent être complexes.

Exemple : considérons l'exemple 4 des paragraphes 2.5 et 3.3 . Le "dividende" est la jointure des relations FOURN et LIVR de ERPHYLOG. LES O.E. associés au dividende sont donc des couples de repères (①, ③) de PHYLOG, liés par FPF. Les O.E. associés au "quotient" sont donc également des couples de repères (①, ③), liés par FPF et sélectionnés selon un certain algorithme qui consiste à :

a) rechercher toutes les valeurs de NOP associées aux repères ② de PHYLOG par NP.

b) rechercher tous les ensembles de couples associés aux mêmes valeurs de NOF

c) rechercher pour tous ces ensembles toutes les valeurs de NOP associées et ne conserver que les ensembles tels que l'ensemble des valeurs de NOP déterminé au point a) soit compris dans celui du point c).

Les couples de ces ensembles restants sont les O.E. au quotient.

Illustrons ces résultats au niveau des occurrences :

ERPHYLOG :

FOURN

NOF	VILLE
I0	METZ
II	NANCY
I2	METZ
I3	TOUL

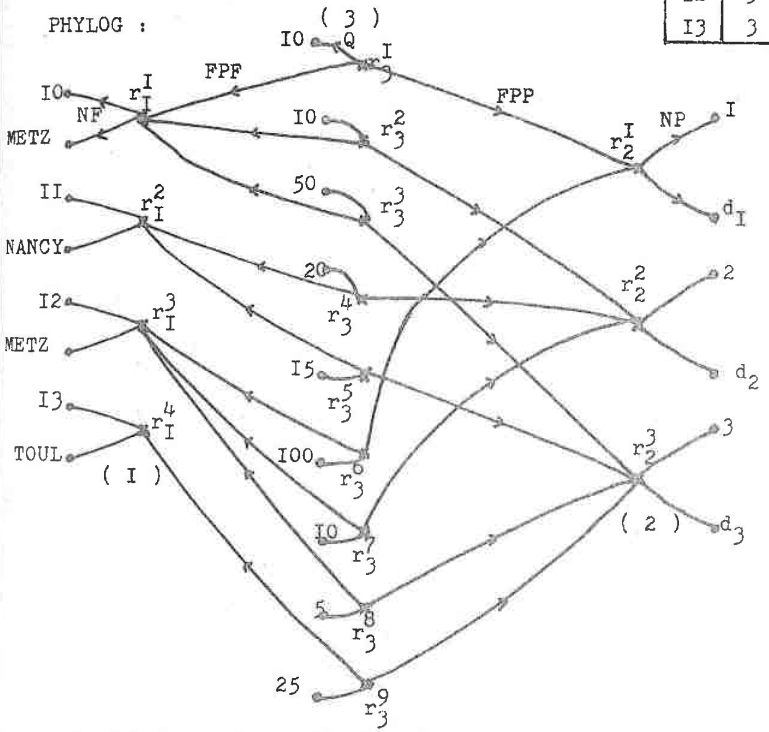
PIECE

NOP	DES
I	d_I
2	d_2
3	d_3

LIVR

NOF	NOP	QLIV
I0	I	I0
I0	2	I0
I0	3	50
II	2	20
II	3	I5
I2	I	I00
I2	2	I0
I2	3	5
I3	3	25

PHYLOG :



Le "dividende" : FOURN [NOF] LIVR — les objets équivalents

I0	METZ	I	I0	$\rightarrow (r_I^I, r_3^I)$
I0	METZ	2	I0	$\rightarrow (r_I^I, r_3^2)$
I0	METZ	3	50	$\rightarrow (r_I^I, r_3^3)$
II	NANCY	2	20	$\rightarrow (r_I^2, r_3^4)$
II	NANCY	3	I5	$\rightarrow (r_I^2, r_3^5)$
I2	METZ	I	I00	$\rightarrow (r_I^3, r_3^6)$
I2	METZ	2	I0	$\rightarrow (r_I^3, r_3^7)$
I2	METZ	3	5	$\rightarrow (r_I^3, r_3^8)$
I3	TOUL	3	25	$\rightarrow (r_I^4, r_3^9)$

Algorithme de construction des O.E. au "quotient"

a) {I , 2 , 3} (ensemble des valeurs de NOP liées aux r_2^i).

b) $\{(r_I^I,r_3^I),(r_I^I,r_3^2),(r_I^I,r_3^3)\}$ associé à NOF = I0

$\{(r_I^2,r_3^4),(r_I^2,r_3^5)\}$ " " = II

$\{(r_I^3,r_3^6),(r_I^3,r_3^7),(r_I^3,r_3^8)\}$ " " = I2

$\{(r_I^4,r_3^9)\}$ " " = I3

c) sont sélectionnés :

$\{(r_I^I,r_3^I),(r_I^I,r_3^2),(r_I^I,r_3^3)\}$

(associés à NOP = I,2 et 3)

$\{(r_I^3,r_3^6),(r_I^3,r_3^7),(r_I^3,r_3^8)\}$

Ces 6 couples sont les O.E. associés à :

(FOURN [NOF] LIVR) [NOP,NOF ⊘ NOP] PIECE

I0	METZ	I	I0	$\rightarrow (r_I^I,r_3^I)$
I0	METZ	2	I0	$\rightarrow (r_I^I,r_3^2)$
I0	METZ	3	50	$\rightarrow (r_I^I,r_3^3)$
I2	METZ	I	I00	$\rightarrow (r_I^3,r_3^6)$
I2	METZ	2	I0	$\rightarrow (r_I^3,r_3^7)$
I2	METZ	3	5	$\rightarrow (r_I^3,r_3^8)$

Les O.E. associés à FTOT = ((FOURN[NOF]LIVR)[NOP,NOF⊘NOP]PIECE) [NOF,VILLE] snt donc :

FTOT

I0	METZ
I2	METZ

$\rightarrow \{(r_I^I,r_3^I),(r_I^I,r_3^2),(r_I^I,r_3^3)\}$

$\rightarrow \{(r_I^3,r_3^6),(r_I^3,r_3^7),(r_I^3,r_3^8)\}$

Après introduction de la division, la forme la plus générale des O.E. reste donc les ensembles de n-uples de repères de PHYLOG.

3.6.2. L'utilisation de la réunion, de l'intersection et de la différence :

Nous avons précisé au paragraphe 3.5.I. les conditions d'utilisation de ces opérations. Les opérandes sont obtenus par projection : les O.E. qui leur sont associés sont donc des ensembles de repères, les repères d'un ensemble étant associés aux mêmes valeurs des constituants de la projection. Soit E_i et F_i ces O.E. associés aux 2 opérandes. L' "interprétation" du paragraphe 3.4.5. s' étend sans difficultés aux O.E. ensembles de repères :

✦ à chaque occurrence d'une intersection est associé un ensemble de couples de repères pris dans E_i et F_i et associés aux mêmes valeurs de tous les constituants

✦ à chaque occurrence d'une réunion correspond soit un ensemble de repères associés au premier opérande (E_i), soit un ensemble de repères associés au second opérande (F_i), soit un ensemble de couples de repères pris dans E_i et F_i et associés aux mêmes valeurs pour tous les constituants quand ils existent

✦ à chaque occurrence d'une différence correspond un ensemble de repères associés au premier opérande (E_i). Ne sont conservés que les ensembles sans "équivalent" parmi les F_i (c.à.d. non liés aux mêmes valeurs de tous les constituants).

Exemple : considérons l'exemple 6 des paragraphes 2.5 et 3.3 .

A chaque occurrence de DEPOT VILLE est associé un ensemble de repères du type ① de PHYLOG, liés à la même valeur de VILLE (E_i).

A chaque occurrence de FOURN VILLE est associé un ensemble de repères du type ⑤ de PHYLOG, liés à la même valeur de VILLE (F_i).

Les O.E. correspondants au type de repères logiques ② sont donc, soit des ensembles de couples pris dans E_i et F_i et associés aux mêmes valeurs de VILLE, soit des ensembles de repères E_i, soit des ensembles de repères F_i.

Illustrons ces résultats au niveau des occurrences :

ERPHYLOG :

DEPOT

NODEP	VILLE	SURF
I00	METZ	I00
I0I	NANCY	500
I02	NANCY	200
I03	METZ	I00
I04	TOUL	200

PIECE

NOP	COUL
I	c_I
2	c_2
3	$\bar{c}_I$
5	c_2

STOCK

NODEP	NOP	QTE
I00	I	50
I00	2	I00
I00	5	50
I0I	2	80
I0I	5	20
I02	3	I00
I03	I	500
I03	2	80
I03	3	I0
I03	5	25
I04	I	I00

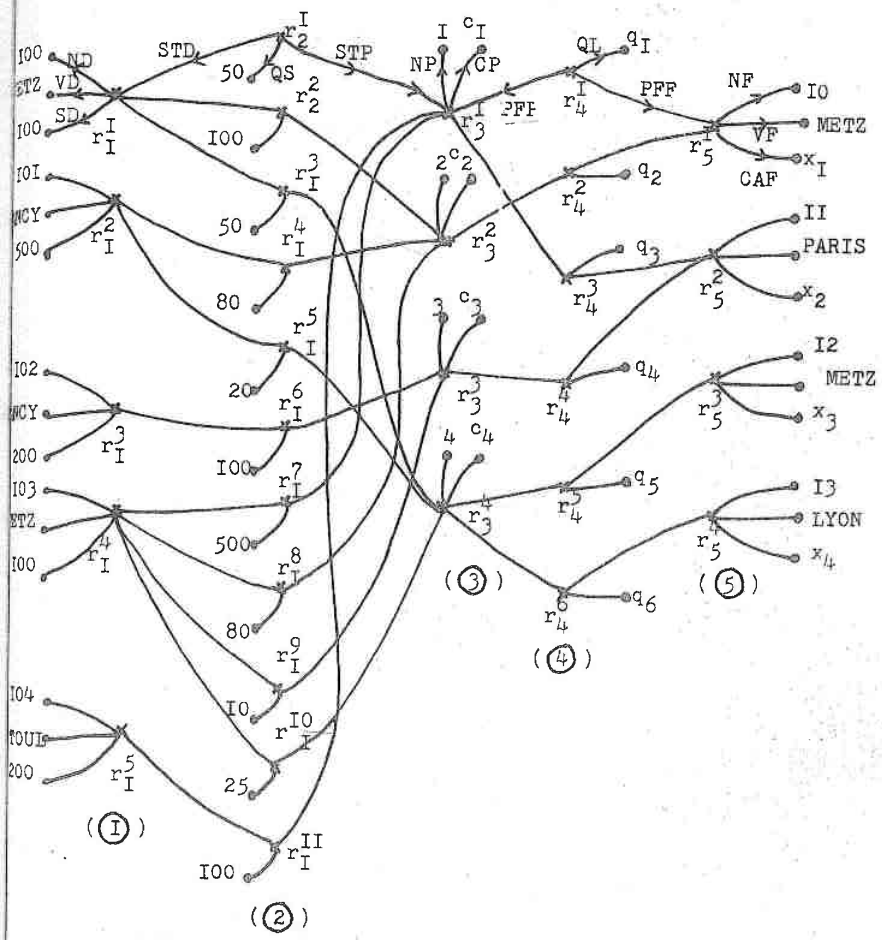
LIVR

NOP	NOF	QTE
I	I0	q_I
2	I0	q_2
I	II	q_3
3	II	q_4
5	I2	q_5
5	I3	q_6

FOURN

NOF	VILLE	CA
I0	METZ	x_I
II	PARIS	x_2
I2	METZ	x_3
I3	LYON	x_4

PHYLOG :



Objets équivalents :

DEPOT[VILLE] O.E.

METZ	$\{r_I^I, r_I^4\}$
NANCY	$\{r_I^2, r_I^3\}$
TOUL	$\{r_I^5\}$

FOURN[VILLE] O.E.

METZ	$\{r_5^I, r_5^3\}$
PARIS	$\{r_5^2\}$
LYON	$\{r_5^4\}$

LOC = DEPOT[VILLE] + FOURN[VILLE] O.E.

METZ	$\longrightarrow \{(r_I^I, r_5^I), (r_I^I, r_5^3), (r_I^4, r_5^I), (r_I^4, r_5^3)\}$
NANCY	$\longrightarrow \{ r_I^2, r_I^3 \}$
TOUL	$\longrightarrow \{ r_I^5 \}$
PARIS	$\longrightarrow \{ r_5^2 \}$
LYON	$\longrightarrow \{ r_5^4 \}$

3.6.3. Conclusion:

Les ensembles de n-uples de repères de PHYLOG restent donc les O.E. les plus complexes après introduction de toutes les opérations de l'algèbre.

Il nous reste donc à étudier les aménagements à apporter au système PIVOINES pour qu'il puisse prendre en compte la gestion de ces O.E. Ainsi qu'à évaluer qualitativement cette utilisation des O.E. Ce sera le but du prochain paragraphe.

- - - - -

3.7. La gestion des O.E. par le système PIVOINES :

3.7.I. Généralités :

Cette mise en oeuvre a été conçue comme une extension du processus décrit aux paragraphes 2.2 et 2.3 .La philosophie générale est la création dynamique des O.E. utiles, lors de l'exécution des requêtes d'accès.Ce choix sera discuté dans la suite.

Cette option fait que la nouveauté se situe essentiellement au niveau des procédures de recherche (qui sont des données du système).

Nous aurons, en plus des procédures secondaires correspondant aux accès élémentaires de PHYLOG, des procédures de "constitution d'O.E." et d' "utilisation d'O.E.". En effet, la forme générale des O.E. (ensembles de n-uples de repères) fait que :

+ une première tâche est d'accèder à ces repères, ce qui peut être réalisé à l'aide des procédures secondaires
+ une seconde tâche est de sélectionner et regrouper ces repères selon les algorithmes précédemment décrits pour constituer les O.E. Les procédures de "constitution d'O.E." réaliseront cette seconde tâche.

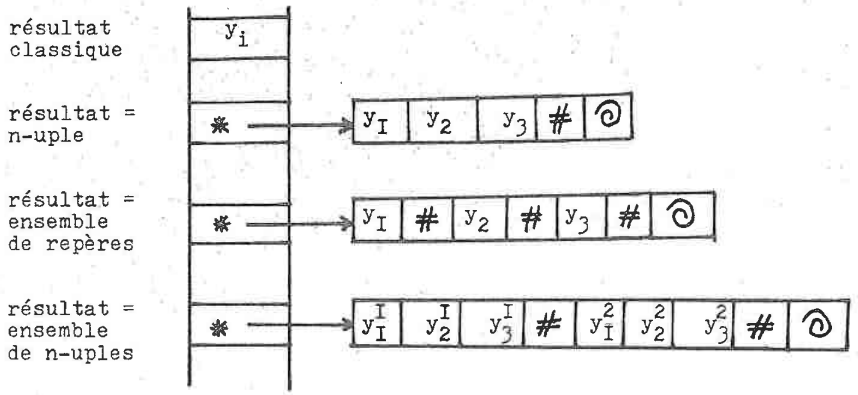
Les procédures utilisant les O.E. doivent le plus souvent, dans un premier temps, "extraire" des O.E. les occurrences de repères indispensables pour la mise en oeuvre des accès voulus. Ce sera le but des procédures d' "utilisation des O.E.".

Dans les cas complexes, les procédures de construction des O.E. pourront s'inspirer des résultats de l'optimisation du paragraphe 3.5.2. pour mettre en oeuvre, par exemple, les algorithmes de sélection avant ceux de constitution des ensembles de n-uples.

3.7.2. La gestion mémoire :

En ce qui concerne le stockage des O.E., il pourra se faire dans les tableaux de résultats classiques du système (cf. [3.7] p.288-292), par adressage indirect : avec une marque d'adressage indirect et un pointeur vers une zone où seront stockés tous les O.E. au fur et à mesure des besoins. Il suffit pour enregistrer les diverses formes d'O.E. d'une marque de fin de n-uple et d'une marque de fin d'ensemble .

exemple : soient
- ✱ la marque d'adressage indirect
- # la marque de fin de n-uple
- ◎ la marque de fin d'ensemble



La durée de vie des O.E. est la même que celle des résultats classiques. Les procédures du système assurant la gestion des résultats peuvent être étendues ici (avec récupération de place à la fois dans le tableau des résultats et dans la zone de stockage).

Les différences de longueur entre O.E. font que la méthode la plus pratique de gestion de la zone de stockage est probablement la suivante :

+ rangement des O.E. dans un tableau linéaire sans récupération de place tant qu'il en reste de disponible
+ rangement dans le premier emplacement libre de taille suffisante si cela devient nécessaire
+ effacement complet après chaque filtre.

D'autres méthodes, plus efficaces mais beaucoup plus coûteuses sont possibles ([3.8] chap. 5).

Au niveau du système de petites modifications doivent être apportées en ce qui concerne la manipulation des résultats (par exemple les comparaisons de résultats lors des intersections sont plus complexes).

3.7.3. Essai d'évaluation "économique" de l'utilisation des O.E.:

Rappelons que le problème de la "correpondance élargie" (c.a.d. de la différenciation des types de repères logiques par rapport aux types de repères de PHYLOG) naît :

+ de la prise en compte de visions utilisateurs différentes
+ de l'évolution dans le temps des besoins (cf. 3.I).

A ces deux causes principales peuvent correspondre des "remèdes" radicaux qui sont respectivement :

+ la redondance interne (on supprime le problème en enregistrant toutes les formes possibles correspondant aux différentes visions utilisateurs)
+ la restructuration systématique (on supprime le problème en faisant évoluer la forme interne à chaque modification des besoins)

Leur application rigoureuse est synonyme de lourdeur et d'instabilité. Notre solution est incontestablement plus souple. Par contre, elle peut être coûteuse en volume mémoire (temporaire) et en temps. Elle se justifie pleinement pour un petit nombre de types de repères logiques sans équivalents directs.

Deux remarques peuvent conforter cette appréciation :

+ la complexité des O.E. n'est pas très grande en général (cf. tous les exemples traités) : les n-uples de grande dimension seront rares.
  Les O.E. du type "ensemble" sont plus inquiétants. Mais le mode global de gestion des inconnues conduit déjà à des manipulations d'informations de ce volume.
+ grâce aux groupements traduisibles de LOG, il est souvent possible d'éviter de construire les O.E.; cette possibilité devra être exploitée systématiquement.
  exemple : considérons l'exemple 5 du paragraphe 2.5 et le filtre suivant :

  $$\text{"I000"} \quad NCDE^{-I} \quad PLC^{-I} \quad PP \quad NPR \quad x_I$$

  (qui donne les n° de produit de la commande n° I000)
  - si $PCL^{-I}\ PP^{-I}$ est défini comme un groupement traduisible de traduction $LCC^{-I}\ LCP$ il n'est pas nécessaire de générer les O.E. associés à (2) de LOG
  - dans le cas contraire c'est indispensable (comme résultat de la procédure associée à PLC^{-I} et pour servir d'origine à la procédure associée à PP^{-I}, par exemple).

En définitive, le coût semble compatible avec les avantages procurés.

- - - - -

3.8. Conclusion :

Toute cette étude est assez liée au système PIVOINES et dépendante de ses particularités. Nous réserverons pour la conclusion générale, les éventuelles leçons à tirer sur un plan plus large.

L'aboutissement logique consiste à présenter en détail un exemple de mise en oeuvre effective du système. C'est le but de la quatrième et dernière partie.

- - - - -

BIBLIOGRAPHIE :

[3.I] Relational completeness of data base sublanguages
E.F.CODD
Data base systems, Courant Computer Sciences Symposia series, Vol 6,Prentice-Hall
(I972)

[3.2] The Peterlee Relational Test Vehicel-a system overview
S.J.P.TODD
IBM System Journal,Vol I5,n°4
(I976)

[3.3] An introduction to database systems
C.J.DATE
Addison-Wesley
(Janvier I976)

[3.4] Le modèle relationnel de Codd
DEHENEFFE,LECHARLIER,PAULUS
Institut d'INFORMATIQUE de NAMUR
(I973)

[3.5] The PRTV optimiser :the current state
J.S.M. VERHOFSTAD
IBM Report UKSC 0083
(Mai I976)

[3.6] Optimisation of a single expression in a relational data base system
P.A.V.HALL
IBM Report UKSC 0076
(Juin I975)

[3.7] Description formelle,représentation,interrogation des informations complexes-système PIVOINES.
M.CREHANGE
(DECEMBRE I975)

[3.8] Les structures d'information et leur représentation en mémoire
C.PAIR
Polycopié (Nancy I).

ooooo

CHAPITRE 4 :

UN EXEMPLE DE MISE EN OEUVRE DU SYSTEME

4.I. Présentation du problème traité .....p. I
4.2. Le choix des structures .....p. 3
4.2.I. La structure PHYLOG
4.2.2. Les structures logiques
4.3. Analyse de la correspondance LOG-PHYLOG .....p. 5
4.3.I. Les objets équivalents
4.3.2. Les caractéristiques de la mise en oeuvre réelle
4.3.3. La table de correspondance
4.4. Exemple d'interrogation de la base .....p. II

Annexe n° I : les procédures .....p. I9
Annexe n° 2 : autres exemples d'interrogation de la base ....p. 26

oooooooo

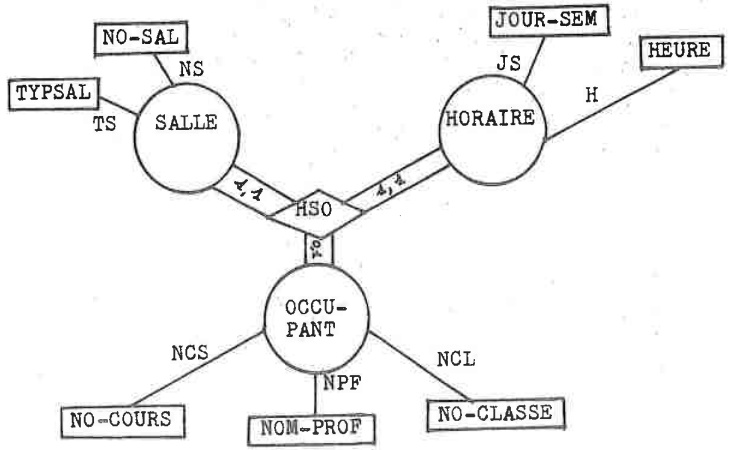
4.I. Présentation du problème traité :

Il s'agit de la base de données d'un établissement scolaire fictif. La base est partagée entre plusieurs utilisateurs ayant des préoccupations diverses. Nous n'en retiendrons que deux :

+ un responsable des emplois du temps
+ un responsable de la scolarité et plus précisément des résultats scolaires.

La perception du premier utilisateur peut être traduite par :

fig. 4.I.



où l'on peut noter que :

+ une entité HORAIRE est associée à chaque case horaire de l'emploi du temps. La durée normale des cours est d'une heure. Les cours plus longs sont considérés comme plusieurs cours consécutifs d'une heure chacun.
+ à chaque unité de cours d'une heure est associé un NO-COURS.

Les questions les plus courantes seront du type :

+ qui occupe la salle n le jour j à l'heure t?
+ quelles salles sont occupées le jour j à l'heure t?
+ quels sont les horaires de la classe m? du professeur p?
+ comment sont occupées les salles de TP CHIMIE? etc.....

La perception du second utilisateur est de type hiérarchique :

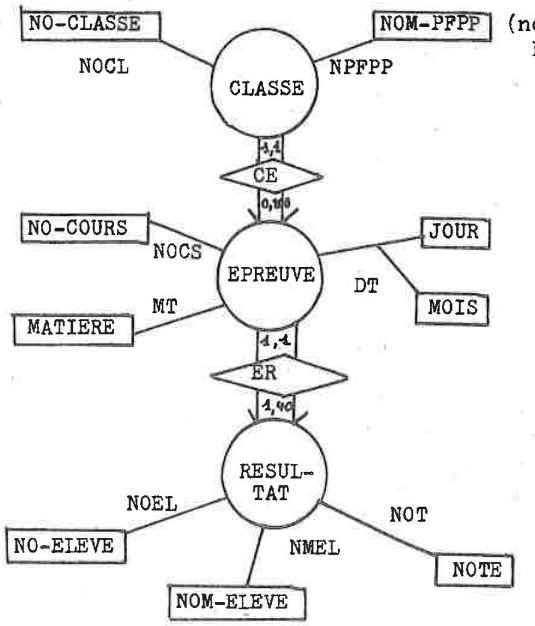
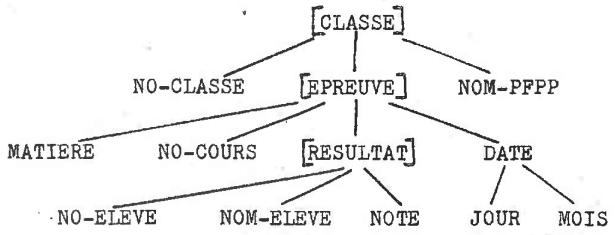


fig. 4.2.

Soit encore avec la notation habituelle pour les hiérarchies :



où [X] est un groupe répétitif.

fig. 4.3.

Les questions les plus courantes seront du type :

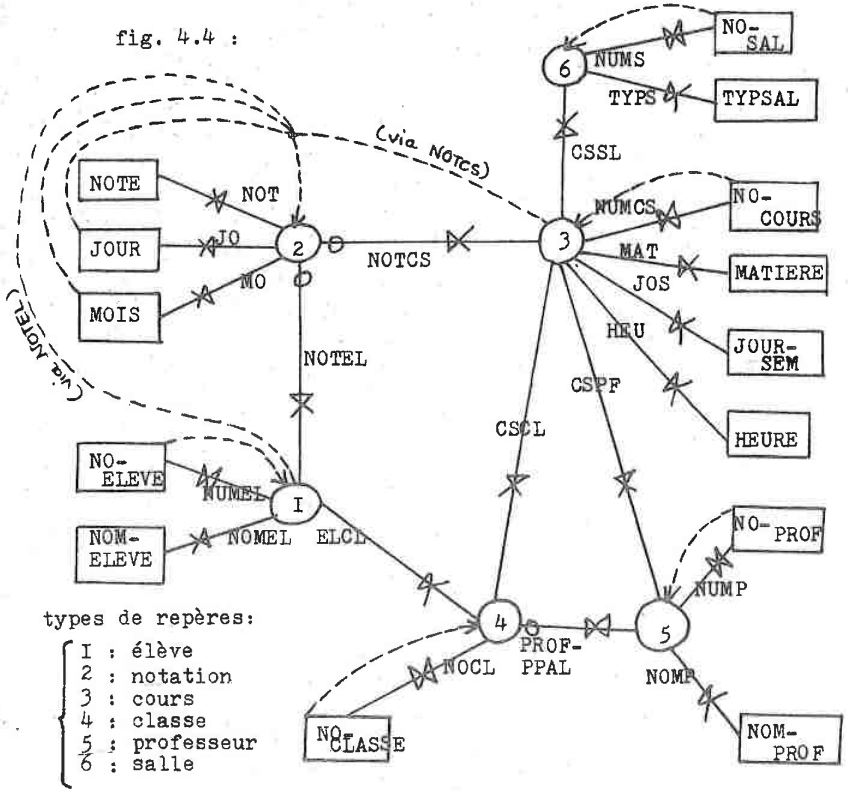
+ quels sont les résultats de l'élève x? dans la matière m? à l'épreuve de MATHS du d?
+ quelles épreuves a effectué la classe c? quelles épreuves se sont déroul ées à la date d?
+ etc.....

4.2. Le choix des structures :

4.2.I. La structure PHYLOG :

Après la synthèse des besoins (étape conceptuelle) et en fonction des contraintes d'implémentation, la structure PHYLOG suivante a été retenue :

fig. 4.4 :



Elle correspond grossièrement à une implémentation organisée autour de 4 fichiers en accès direct interconnectés (élèves, professeurs, classes et salles) et de 2 listes (cours et notes).

4.2.2. Les structures logiques :

Elles correspondent ici à une simple transcription en termes du modèle de PIVOINES des perceptions utilisateurs présentées en 4.I .

Fig. 4.5
Structure LOG_I :

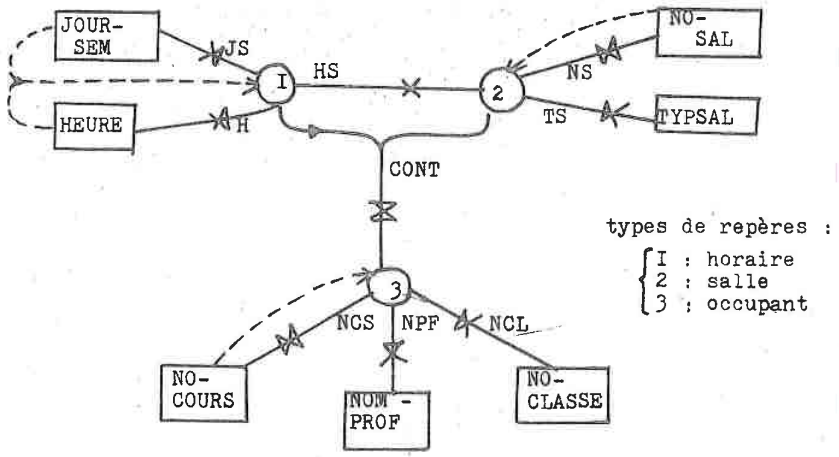
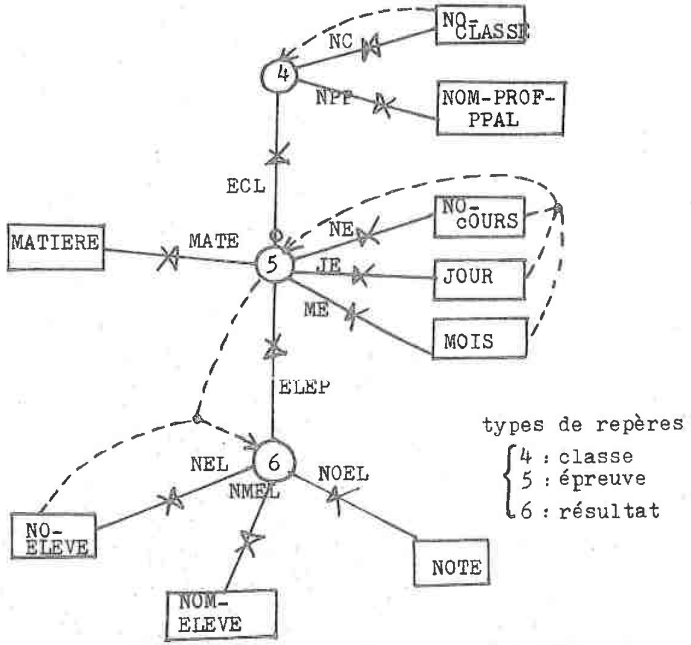


Fig. 4.6
Structure LOG_2



4.3. Analyse de la correspondance LOG-PHYLOG :

4.3.I. Les objets équivalents :

Les expressions relationnelles des trois structures sont les suivantes :

ERPHYLOG :
- ELEVE (NO-ELEVE,NOM-ELEVE,NO-CLASSE)
- NOTATION (NO-ELEVE,NO-COURS,JOUR,MOIS,NOTE)
- COURS(NO-COURS,JOUR-SEM,HEURE,MATIERE,NO-SAL,NO-CLASSE ,NO-PROF)
- CLASSE(NO-CLASSE,NO-PROF-PPAL)
- PROF(NO-PROF,NOM-PROF)
- SALLE(NO-SAL,TYPSAL)

$ERLOG_I$:
- HORAIRE(JOUR-SEM,HEURE)
- SALLE2(NO-SAL,TYPSAL)
- HS(JOUR-SEM,HEURE,NO-SAL)
- OCCUPANT(NO-COURS,NOM-PROF,NO-CLASSE,JOUR-SEM,HEURE, NO-SAL)

$ERLOG_2$:
- CLASSE2(NO-CLASSE,NOM-PROF-PPAL)
- EPREUVE(NO-COURS,JOUR,MOIS,MATIERE,NO-CLASSE)
- RESULTAT(NO-ELEVE,NO-COURS,JOUR,MOIS,NOTE,NOM-ELEVE)

Les correspondances exprimées en termes de l'algèbre relationnelle sont les suivantes :

$\langle ERLOG_I$ - ERPHYLOG$\rangle$:

$$\begin{cases} \text{HORAIRE} = \text{COURS}\,[\text{JOUR-SEM},\text{HEURE}] \\ \text{SALLE2} = \text{SALLE} \\ \text{OCCUPANT} = (\text{COURS}[\text{NO-PROF}]\text{PROF})\,[\text{NO-COURS},\text{NOM-PROF}, \text{NO-CLASSE},\text{JOUR-SEM},\text{HEURE},\text{NO-SAL}] \end{cases}$$

$\langle ERLOG_2$ - ERPHYLOG$\rangle$:

$$\begin{cases} \text{CLASSE2} = \text{CLASSE} \\ \text{EPREUVE} = (\text{NOTATION}\,[\text{NO-COURS}]\,\text{COURS})\,[\text{NO-COURS}, \text{JOUR},\text{MOIS},\text{MATIERE},\text{NO-CLASSE}] \\ \text{RESULTAT} = ((\,\text{NOTATION}\,[\text{NO-ELEVE}]\,\text{ELEVE})[\text{NO-COURS}] \text{COURS})\,[\text{NO-COURS},\text{NO-ELEVE},\text{JOUR},\text{MOIS}, \text{NOTE},\text{NOM-ELEVE}] \end{cases}$$

Les objets équivalents, déduits directement des expressions ci-dessus sont les suivants :

type de repères logiques	relation associée	O.E.
①	HORAIRE	projection ⇒ ensembles des repères ③ de PHYLOG liés aux mêmes valeurs de JOUR-SEM et HEURE
③	OCCUPANT	jointure ⇒ couples de repères (③, ⑤) de PHYLOG
⑤	EPREUVE	projection (selon clé) ⇒ sans effet jointure ⇒ couples de repères (②, ③) de PHYLOG
⑥	RESULTAT	projection ⇒ ensembles de couples (②, ③) liés aux mêmes valeurs de NO-COURS, JOUR, MOIS, NO-CLASSE, MATIERE 2 jointures ⇒ triplets de repères (②, ①, ③) de PHYLOG projection (selon clé) ⇒ sans effet

En application de la remarque du paragraphe 3.4.3.2., on pourrait en raison des liens fonctionnels entre classes de repères, ne conserver comme O.E. que :

- ③ de PHYLOG pour ③ de LOG
- des ensembles de repères ② de PHYLOG pour ⑤ de LOG
- ② de PHYLOG pour ⑥ de LOG

Nous conserverons dans la mise en oeuvre les O.E. complets décrits dans le tableau ci-dessus.

4.3.2. Les caractéristiques de la mise en oeuvre réelle :

Cette mise en oeuvre concerne exclusivement la correspondance entre LOG_I et PHYLOG.

La base a été implémentée d'une manière simple (structures PL/I), en mémoire centrale, selon le jeu d' essai suivant :

TABEL

	n° élève	nom	n° classe	pointeur vers TABNOT
r_I^1	I	TOTO	I	I
r_I^2	2	JOE	2	2
r_I^3	3	DUPONT	I	3
r_I^4	4	ARTHUR	2	4
r_I^5	5	MACHIN	I	5

TABCOURS

	n° cours	jour	heure	matière	n° classe	n° prof.	n° salle
r_3^1	I	LUNDI	IO	FRANCAIS	I	I	I
r_3^2	2	LUNDI	II	MATHS	2	2	I
r_3^3	3	MARDI	I5	ANGLAIS	I	3	I
r_3^4	4	MARDI	I5	CHIMIE	2	I	2
r_3^5	5	MARDI	I5	MATHS	3	3	3
r_3^6	6	MARDI	I6	MATHS	I	I	I
r_3^7	7	MARDI	I6	ANGLAIS	2	2	3

TABPROF

	n° prof.	nom
r_5^1	I	MARTIN
r_5^2	2	HUBERT
r_5^3	3	MIKE

TABCLASS

	n° classe	n° prof. ppal
r_4^1	I	I
r_4^2	2	3
r_4^3	3	2

TABNOT

	note	jour	mois	n° cours	n° élève	pointeur vers note suivante du même élève
r_2^1	8,5	3	2	I	I	6
r_2^2	9	3	2	2	2	0
r_2^3	I2,5	3	2	I	3	7
r_2^4	7	3	2	2	4	0
r_2^5	I6,5	3	2	I	5	8
r_2^6	IO,5	IO	2	4	I	0
r_2^7	I3	IO	2	4	3	0
r_2^8	II	IO	2	4	5	0

TABSAL

	n° salle	type salle
r_6^1	I	COURS
r_6^2	2	TP CHIMIE
r_6^3	3	COURS

Ce jeu d'essai correspond au niveau logique à :

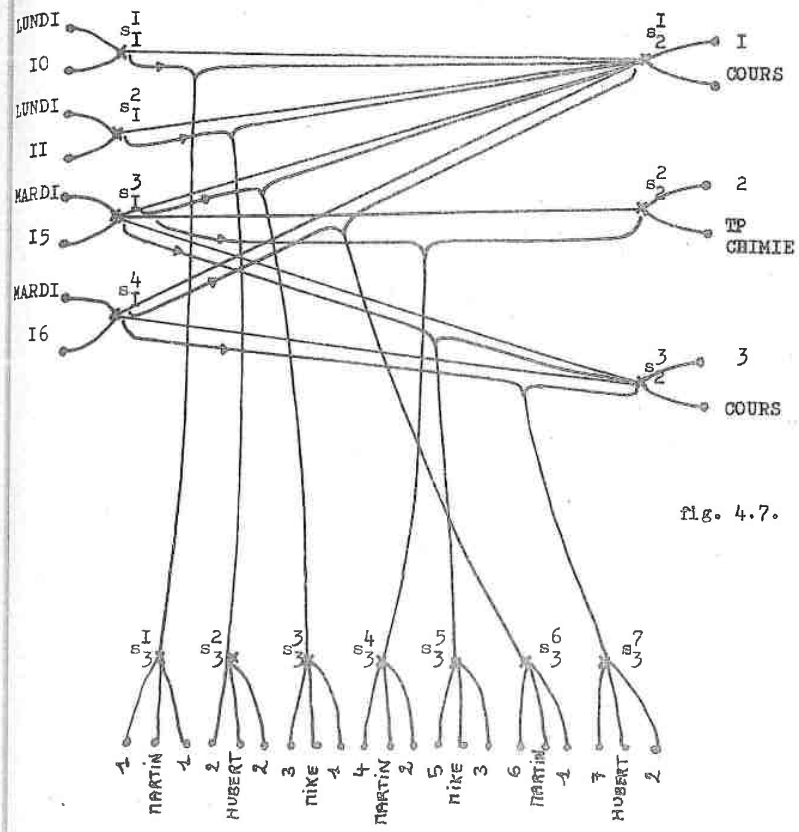


fig. 4.7.

La correspondance au niveau des occurrences sera donc :

$s_I^1 \rightarrow \{r_3^1\}$

$s_I^2 \rightarrow \{r_3^2\}$

$s_I^3 \rightarrow \{r_3^3, r_3^4, r_3^5\}$

$s_I^4 \rightarrow \{r_3^6, r_3^7, r_3^8\}$

$s_3^1 \rightarrow (r_3^1, r_5^1)$
$s_3^2 \rightarrow (r_3^2, r_5^2)$
$s_3^3 \rightarrow (r_3^3, r_5^3)$
$s_3^4 \rightarrow (r_3^4, r_5^1)$
$s_3^5 \rightarrow (r_3^5, r_5^3)$
$s_3^6 \rightarrow (r_3^6, r_5^1)$
$s_3^7 \rightarrow (r_3^7, r_5^2)$

4.3.3. La table de correspondance et les procédures :

La table principale de la correspondance <LOG_I - PHYLOG> apparaît comme suit sur le listing d'acquisition :

```
            *** ACQUISITION DE LA CORRESPONDANCE PHILOG ***

 DESIREZ-VOUS VOIR LE RESULTAT DE CETTE ACQUISITION(O/N)?
        N
 ENTREZ LA CORRESPONDANCE
    1      < XX  JS  XX  :   P50(2,2),
    2       XN  JS-  XX  :   P51(2,1),
    3       XX  H  XX  :   P52(2,2)  ,
    4       XN  H-  XX  :   P53(2,1),
    5       XX  NS  XX  :   P54(2,2) ,
    6       XN  NS-  XX  :   P55(2,0),
    7       XX  TS  XX  :   P56(2,2),
    8       XN  TS-  XX  :   P57(2,1),
    9       XX  HS  XX  :   P58(2,1),
   10       XX  HS-  XX  :   P59(2,2),
   11       XX  NCS  XX  :   P60(2,2),
   12       XN  NCS-  XX  :   P61(2,0),
   13       XX  NPF  XX  :   P62(2,2),
   14       XN  NPF-  XX  :   P63(2,1),
   15       XX  NCL  XX  :   P64(2,2),
   16       XN  NCL-  XX  :   P65(2,1),
   17       XX  CONT@  <  XO NCS XX :  P67(3,1),
   18         XO  NPF  XX  :   P68(3,1),
   19         XO  NCL  XX  :   P69(3,1) ,
   20              XX :  P66(3,2) > > ;

 *      *       *       *       *       *
```

On distingue :

+ dans la partie gauche, toutes les relations de LOG_I, leurs inverses(ce qui n'est pas obligatoire) et des groupements plus complexes dans les lignes I7 à 20 (CONT.NCS,CONT.NPF,CONT.NCL).
+ dans la partie droite, les noms des procédures primaires (P50 à P69) avec, entre parenthèses, l'arité et le "d° d'efficacité" permettant le choix de l'ordonnancement des accès (il est ici réduit à sa plus simple expression de 3 valeurs 0,1,2 marquant une efficacité décroissante).

La table secondaire liant les procédures primaires, les procédures secondaires et les procédures de gestion des O.E. est la suivante (les noms de procédures secondaires sont constitués de PS suivi du nom de l'accès de PHYLOG ;
les noms de procédures de "construction d'O.E." sont CREREPn ;
les noms de procédures d' "utilisation d' O.E." sont UTILREPn) :

H^{-I}	P53	PSUTIL3REP4 PSHEU-I PSCREREP4
NS	P54	PSNUMS
NS^{-I}	P55	PSNUMS-I
TS	P56	PSTYPS
TS^{-I}	P57	PSTYPS-I
HS	P58	PSUTIL2REP4 PSCSSL
HS^{-I}	P59	PSUTIL4REP4 PSCSSL-I PSCREREP4
NCS	P60	PSUTILIREP6 PSNUMCS
NCS^{-I}	P6I	PSUTIL2REP6 PSNUMCS-I PSUTIL3REP6 PSCSPF PSCREREP6
NPF	P62	PSUTIL4REP6 PSNOMP
NPF^{-I}	P63	PSUTIL5REP6 PSNOMP-I PSCREREP62
NCL	P64	PSUTILIREP6 PSCSCL PSUTIL6REP6 PSNOCL
NCL^{-I}	P65	PSUTIL7REP6 PSNOCL-I PSUTIL3REP6 PSCSCL-I PSCREREP63
CONT	P66	PSUTILIREP4ET5 PSCREREP64
CONT.NCS	P67	PSUTILIREP4ET5 PSUTIL8REP6
CONT.NPF	P68	PSUTILIREP4ET5 PSUTIL9REP6
CONT.NCL	P69	PSUTILIREP4ET5 PSUTILIOREP6

Les procédures primaires sont uniquement constituées de la suite des appels des procédures secondaires.

exemple :

```
P53:PROC;
  CALL PSUTIL3REP4;
  CALL PSHEU@1 ;
  CALL PSCREREP4;
END P53;
```

Le texte des procédures secondaires est donné dans l'annexe I (on notera leur simplicité et leur "régularité" de forme, dans la majorité des cas) Des procédures de "génération d'extrémités" doivent aussi être décrites (procédures S20 à S29 -cf. Annexe I).

4.4. Exemples d'interrogation de la base :

Quatre requêtes ont été soumises au système :

Requête I : "nom du professeur et n° de la classe concernée par le cours 4 ?"

Requête 2 : "nom du professeur , n° classe et n° cours ayant lieu le mardi à I5 h dans la salle 2 ?"

Requête 3 : "n° des classes qui travaillent le mardi à I5 h ?"

Requête 4 : "horaires de travail de tous les professeurs ?"

Nous allons présenter à titre d'exemple la requête 3 (les 3 autres sont décrites dans l'annexe ?, avec leurs particularités).

Le filtre de la requête 3 est le suivant :

$$\langle \text{'MARDI'} \quad JS^{-I} \ , \ \text{'I5'} \quad H^{-I} \rangle XI \quad HS \quad X2 \quad \underline{et} \quad \langle XI, X2 \rangle CONT.NCL \ X3$$

(l'inconnue résultat est donc X3).

Le programme maquette obtenu est le suivant : (on peut y noter l'ordonnancement des accès retenu).

```
**********************
*PROGRAMME MAQUETTE*
**********************
2 0 0,1000 1P51('MARDI',X 1),1000 1P53('15',X 1),1000 1P58(X 1,X 2)

,3 1,4 1,2 1 1,1000 2P69(X 1,X 2,X 3),3 2,5 1,8 1,9 ,
```

où les différents symboles ont la signification suivante :

```
        SYMBOLES UTILISES DANS LE PROGRAMME MAQUETTE
2 I J   :  2 DEBUT DE MOTIF , I NUMERO DU SOUS-FILTRE CONTEXTE A GAUCHE ,

                J NB D INCONNUES LOCALES UTILISEES DANS LE MOTIF

 3 I    :  3 FIN DE MOTIF , I NUMERO DU MOTIF ;
 4 I    :  B(EI) :   5 I   :  EI
 6 I    :  REUNION(I) :   7   :  EMPILER
 8 I    :  DECHU(I) ;   9   :  FIN DU PROGRAMME MAQUETTE
 1IJK MPROCEDURE(   ,   ,   )   :  IJK NEGATION ET ITERATION , M NB D ARGU
                MENTS BUT NON COMPRIS
```

Le programme exécutable généré est le suivant : (il est encore encombré de quelques procédures du type initialisation ou sortie de résultats intermédiaires qui disparaîtront intermédiaires dans les versions futures).

```
CALL INITIALISATION;
MWLIB=1; NW,COMPTPIL=0; PILBOOL(*)='0'B;
/* TRAITEMENT D UN (J) DE N.  0   AVEC  0 INCONNUES LOCALES */
/* APPEL D UN MODULE DE PROCEDURE DE RECHERCHE */
   BEGIN;
   ARG(0)='                    ';                    } préparation des
   ARG(  1)='MARDI               ';                  } arguments
   IMPLANW(NW+1)=MWLIB; RECUPW(NW+1)='0'B; BOOL='0'B;
     BEGIN;
     DCL TRAV(1:  1) CHAR(18);
 CALL P51 ;                                          } appel de procédure
     IF BRESU='0'B THEN GOTO FIN@MOD@BV;  BOOL='1'B;  principale
/* MODULE IV */
      DO J1=1 TO NRESU; /*MODULE J3*/                } gestion
         TRAV(  1)=RESU(J1);                         } des
         DO J2=1 TO   1; W(MWLIB-1+J2)=TRAV(J2); END; } résultats
         MWLIB=MWLIB+  1; END;
     END;
  FIN@MOD@BV:
   W(MWLIB)=Y; MWLIB=MWLIB+1; NW=NW+1; IF BOOL='0'B THEN GOTO F1 ;
CALL ORANGE;          } procédure d'impression de résultats intermédiaires
   END;
/* APPEL D UN MODULE DE PROCEDURE DE RECHERCHE */
   BEGIN;
   ARG(0)='                    ';
   ARG(  1)='15                  ';
   IMPLANW(NW+1)=MWLIB; RECUPW(NW+1)='0'B; BOOL='0'B;
     BEGIN;
     DCL TRAV(1:  1) CHAR(18);
 CALL P53 ;
     NAD=0;  /*MODULE H1 H2 H3*/
     IF NRESU=0 THEN GOTO MODULE@H2;
ON CONVERSION GOTO CARACT;
     DO T=IMPLANW(    1) BY     1 WHILE (W(T)#=Y);
        J1=0;  L100: J1=J1+1;                        } comparaison
        IF J1<=NRESU THEN DO;                        } des
     J2=RESU(J1); J3=W(T-1+    1);                   } résultats
     IF J2>0 & J3>0 & J2=J3 THEN DO:                  } et
       NAD=NAD+1;AD(NAD)=T-1;GOTO L100;END;           } intersection
     IF J2>0 & J3>0 & J2#=J3 THEN GOTO L100;
     IF J2<0 & J3<0 THEN DO;
```



```
        DO WHILE(TRAV(J2)=0);
           J4=TRAV(J2); J5=TRAV(J3);
           IF J4=J5 THEN GOTO L100;
           J2=J2+1; J3=J3+1; END;
        NAD=NAD+1; AD(NAD)=T-1; END;
     GOTO L100;
     CARACT:IF RESU(J1)=W(T-1+    1) THEN DO;
       NAD=NAD+1;AD(NAD)=T-1;END;
     GOTO L100;END;END;
ON CONVERSION SYSTEM
     MODULE H2:
       IF NAD=0 THEN BRESU='0'B; ELSE BRESU='1'B;
       IF BRESU='0'B THEN GOTO FIN@MOD@BV;  BOOL='1'B;
/* MODULE IV */
       DO J1=1 TO NRES;  /*MODULE J2*/
          TRAV(  1)=W(AD(J1)+  1);
          DO J2=1 TO   1; W(MWLIB-1+J2)=TRAV(J2); END;
          MWLIB=MWLIB+  1; END;
       END;
   FIN@MOD@BV:
     RECUP(    1)='1'B;
     W(MWLIB)=Y; MWLIB=MWLIB+1; NW=NW+1; IF BOOL='0'B THEN GOTO F1 ;
CALL ORANGE;
     END;
/* APPEL D UN MODULE DE PROCEDURE DE RECHERCHE */
     BEGIN;
     ARG(0)='                    ';
     IMPLANW(NW+1)=MWLIB; RECUPW(NW+1)='0'B; BOOL='0'B;
     DO I01=IMPLANW(    2) BY     1 WHILE (W(I01)#=Y);
     ARG( 1)=W(I01-1+    1);
       BEGIN;
       DCL TRAV(1:  2) CHAR(18);
 CALL P52 ;
       IF BRESU='0'B THEN GOTO FIN@MOD@BV;  BOOL='1'B;
/* MODULE IV */
          TRAV(  1)=W(I01+  0);
       DO J1=1 TO NRESU; /*MODULE J3*/
          TRAV(  2)=RESU(J1);
          DO J2=1 TO   2; W(MWLIB-1+J2)=TRAV(J2); END;
          MWLIB=MWLIB+  2; END;
       END;
   FIN@MOD@BV:
     END;
     RECUP(    2)='1'B;
     W(MWLIB)=Y; MWLIB=MWLIB+1; NW=NW+1; IF BOOL='0'B THEN GOTO F1 ;
CALL ORANGE;
     END;
F1 :  /*TRAITEMENT D UN F;*/
DO J1=  1 TO NARG; F(IMPLANW(J1))=Y THEN RECUPW(J1)='1'B; END;
     J1=  2; J2=  2;  CALL MODIF;
IF BOOL='0'B THEN GOTO E1 ;  /*TRAITEMENT D UN B(E1)*/
/* TRAITEMENT D UN (J) DE N.  1   AVEC  1 INCONNUES LOCALES */
/* APPEL D UN MODULE DE PROCEDURE DE RECHERCHE */
     BEGIN;
     ARG(0)='                    ';
     IMPLANW(NW+1)=MWLIB; RECUPW(NW+1)='0'B; BOOL='0'B;
     DO I01=IMPLANW(    3) BY     2 WHILE (W(I01)#=Y);
     ARG( 1)= (I01-1+    1);
     ARG( 2)= (I01-1+    2);
       BEGIN;
```

```
     DCL TRAV(1:   3) CHAR(18);
  CALL RES 1
     IF NRESU='0'B THEN GOTO FIN@MOD@BV;   BOOL='1'B;
/* MODULE IV */
         TRAV(   1)=W(101+   0);
         TRAV(   2)=W(101+   1);
     DO J1=1 TO NRESU; /*MODULE J3*/
         TRAV(   3)=RESU(J1);
         DO J2=1 TO    3; W(NWLIB-1+J2)=TRAV(J2); END;
         NWLIB=NWLIB+   3; END;
     END;
   FIN@MOD@BV:
    END;
    W(NWLIB)=Y; NWLIB=NWLIB+1; NW=NW+1; IF BOOL='0'B THEN GOTO F2 ;
CALL DRANGE;
    END;
F2 :  /*TRAITEMENT E ON F1*/
DO J1=  4 TO NW;IF W(IMPLANW(J1))=Y THEN RECUPW(J1)='1'B; END:
RECUPW(     3)='1'B;
RECUPW(     3)='1'B;
   J1=  3; J2=  3;  CALL MODFIP;
F1 :  /*TRAITEMENT D ON E1*/
/* TRAITEMENT D ON DECHU N.  1    DEJA EFFECTUE */
CALL DRANGE;
DRANGE:  PROC;
    DCL (YY1(1:NTRG),YY2(1:NTRG)) BIT(1);
    DCL BVAL(1:NBL) BIT(1) INIT((NBL)'1'B);
    DO J4=NW-1 TO 1 BY -1;
        IF RECUPW(J4)='0'B THEN GOTO Q150; END;
Q150: J8=IMPLANW(NW); NWLIB,IMPLANW(NW)=IMPLANW(J4+1);
    RECUPW(J4+1)='0'B;
    J1=0;
P10:J1=J1+1; IF J1>NBL THEN GOTO P30;
    IF BVAL(J1)='0'B THEN GOTO P10; J2,J3=J1;
P20: J3=J3+1;
    IF J3>NBL THEN DO;
        DO J6=0 TO NTRG-1;
            W(NWLIB+J6)=W(J8+((J2-1)*NTRG)+J6); END;
        NWLIB=NWLIB+NTRG;BVAL(J2)='0'B; GOTO P10; END;
    IF BVAL(J3)='0'B THEN GOTO P20;
    J12=(J2-1)*NTRG+J8; J13=(J3-1)*NTRG+J8;
    DO J6=0 TO NTRG-1;
        IF W(J12+J6)=W(J13+J6)!W(J12+J6)=YY!W(J13+J6)=YY THEN;ELSE
    J10,J20=0; /*J1 J2 COMPARABLES*/                  GOTO P20;END;
    DO J6=0 TO NTRG-1;
        IF W(J12+J6)=YY THEN DO; YY1(J6+1)='1'B; J10=J10+1; END;
                        ELSE YY1(J6+1)='0'B;
        IF W(J13+J6)=YY THEN DO; YY2(J6+1)='1'B; J20=J20+1; END;
                        ELSE YY2(J6+1)='0'B; END;
    IF J10<=J20 THEN DO;
        DO J6=1 TO NTRG;
            IF YY1(J6)='1'B THEN DO;
                IF YY2(J6)¬='1'B THEN GOTO P20; END; END;
        BVAL(J2)='0'B; J2=J3; GOTO P20; END;
                ELSE DO;
        DO J6=1 TO NTRG;
            IF YY2(J6)='1'B THEN DO;
                IF YY1(J6)¬='1'B THEN GOTO P20; END; END;
        BVAL(J3)='0'B; GOTO P20; END;
P30: W(NWLIB)=Y; NWLIB=NWLIB+1;
```

```
   END MODP;
MODFIP: PROC;
   DO J4=NW-1 TO 1 BY -1;
      IF RECUPW(J4)='0'B THEN GOTO L150 ; END;
   L150: J7=IMPLANW(J4+1); J8=IMPLANW(NW);
   /* J1=INC. GLOB. J2=LONG.LIGNES W  J7=NOUV.TABLEAU J8=NW */
   IF J7=J8 & (J1=J2 ! W(J8)=Y      ) THEN GOTO L300;
   IF W(J8)=Y        THEN DO; W(J7)=Y      ; MWLIB=J7+1; IMPLANW(NW)=J7; END;
   J5,J6=0;
L200: IF W(J8+J6)#=Y        THEN DO;
        DO J3=0 TO J1-1;
           W(J7+J3+J5)=W(J8+J3+J6); END;
        J5=J5+J1; J6=J6+J2;
        GOTO L200; END;
                        ELSE DO;
                        J4=J7+J5; W(J4)=Y; MWLIB=J4+1; IMPLANW(NW)=J7; END;
L300: PUT DATA ((W(I) DO I=1 TO MWLIB)) SKIP;
END MODFIP;
ORANGE: PROC;
PUT DATA (NW,MWLIB,COMPTPIL) SKIP; PUT DATA ((W(I) DO I=1 TO MWLIB-1)) SKIP;
PUT DATA ((IMPLANW(I) DO I=1 TO NW)) SKIP;
PUT DATA ((RECUPW(I) DO I=1 TO NW)) SKIP;
PUT DATA ((PILBOOL(I) DO I=1 TO COMPTPIL)) SKIP;
PUT DATA ((TOE(I) DO I=1 TO NIVOE)) SKIP;
END ORANGE;
FINPPG: END PPG;
```

Le projet étant encore expérimental, les résultats doivent être cherchés dans le tableau suivant :

```
W(1)='    -1         '          W(2)='           1    '
W(3)='           1   '          W(4)='     -1           '
W(5)='           2   '          W(6)='           2    '
W(7)='    -1         '          W(8)='           3    '
W(9)='           3   '          W(10)='#################';
```

Les résultats sont des triplets de valeurs de X_I, X_2, X_3. Ces triplets sont rangés consécutivement dans le tableau des résultats W : il y a ici 3 occurrences de résultat; soit en clair :

variables / n° occurrences	X_I	X_2	X_3
I	-I	I	I
2	-I	2	2
3	-I	3	3

Les n° des classes qui travaillent le MARDI à I5 h sont donc : I,2,3 (valeurs de X_3).

Analysons enfin, plus particulièrement ce qui concerne les O.E. et pour cela suivons pas à pas les résultats intermédiaires correspondant à l'appel de chaque procédure primaire, donc à chaque accès logiques.

Notons que :

- les indicateurs d'indirection, de fin de n-uple, de fin d'ensemb le sont ici respectivement -,-,0 (car la base ne comprend que des valeurs strictement positives).
- la zone de stockage des O.E. s'appelle TOE.
- les valeurs des repères correspondent aux n° des lignes des tableaux PL/I correspondants.

Appel de P5I (c.à.d. JS^{-I}) avec comme argument 'MARDI'. Le résultat (cf. fig. 4.7.) doit correspondre aux 2 repères logiques s_I^3 et s_I^4, soient les O.E. : $\begin{cases} \{r_3^3 , r_3^4 , r_3^5\} \\ \{r_3^6 , r_3^7\} \end{cases}$

C'est bien ce qu'on peut observer dans le premier résultat intermédiaire (on rapelle que dans le cas d' O.E. on trouve dans W, le rang de l' O.E. dans TOE avec un signe - d'indirection).

Appel de P53 (c.à.d. H^{-I}) avec comme argument 'I5' .Le résultat (cf. 4.7.) doit correspondre au repère logique s_I^3 , soit l' O.E. $\{r_3^3, r_3^4, r_3^5\}$.

C'est bien ce qu'on note dans le deuxième résultat intermédiaire (on peut lire le résultat dans W au rang IMPLANW(2) c'est à dire en W(4)). (cf. P. 4.I7)

Appel de P58 (c.à.d. HS) avec comme argument l'intersection des résultats précédents c'est à dire s_I^3 .

Le résultat(cf. fig. 4.7.) doit correspondre aux repères logiques s_2^I , s_2^2 , s_2^3 soit r_6^I , r_6^2 , r_6^3 .

C'est bien ce qu'on note dans le troisième résultat intermédiaire au rang IMPLANW(3) c'est à dire en W(6) (les résultats sont ici des couples de repères logiques (Ⓘ , ②) liés par HS).

(cf. p. 4.I7)

Appel de P69 (c.à.d. CONT.NCL) avec comme arguments les couples (s_I^3, s_2^I) , (s_I^3, s_2^2) , (s_I^3, s_2^3) . Les résultats sont les n° des classes I,2 et 3.

C'est bien ce qu'on note au niveau des résultats définitifs.

(cf. p. 4.I7)

- - - - -

ANNEXE I: Les procédures .

(ne sont données que les procédures effectivement utilisées dans les tests) .

```
PS1:PROC;
    CALL PSUTIL3REP4;
    CALL PSJOS*1;
    CALL PSCREREP4;
END PS1;
PSUTIL3REP4:PROC;
   I=ARG(0);
I =-1;
   IF ARG(0) ¬= ' ' THEN DO ; I = TOE(1);
                              ARG(0) = I;
                         END;
END PSUTIL3REP4;
PSJOS*1:PROC;
  NRESU = 0; I = ARG(0);
  IF ARG(0) ¬= ' ' THEN DO; IF TABCOURS.JOUR@SEM(I) = ARG (1)
                               THEN BRESU = '1'B;
                               ELSE BRESU = '0'B;
                        END;
                   ELSE DO; DO J=1 BY 1   WHILE (TABCOURS.JOUR@SEM
                            (J)  ¬= ' ');
                            IF TABCOURS.JOUR@SEM(J)= ARG(1)
                            THEN DO; NRESU = NRESU + 1 ;
                                     RESU(NRESU)= J;
                                 END;
                        END;
                        IF NRESU = 0 THEN BRESU = '0'B;
                                     ELSE BRESU = '1'B;
                    END;
END PSJOS*1;
PSCREREP4:PROC;
  IF NRESU ¬= 0 THEN
    DO; K = 1 ;
        DO I = 1 TO NRESU ;
           IF RESU(I) ¬= ' ' THEN
           DO; L = 0 ; N =RESU(I);
                DO M = 1 TO NRESU ;
                   J=RESU(M);
                   IF TABCOURS.JOUR@SEM(J) = TABCOURS.JOUR@SEM
                      (N) & TABCOURS.HEUR(J)=TABCOURS.HEUR(N)
                      THEN DO; L=L+1;
                               TOE(NIVOE+L) = -J;
                               RESU(M)= ' ' ;
                           END;
                END;
             J=NIVOE+1;
             RESU(K) = -J;
               TOE(NIVOE + L + 1) = 0 ;
               NIVOE = NIVOE + L + 1 ; K=K+1;
           END;
        END;
        NRESU = K - 1 ;
    END;
END PSCREREP4;
```

```
P53:PROC;
  CALL PSUTIL2REP4;
  CALL PSCREU1 ;
  CALL PSCREREP4;
END P53;
PSREU1:PROC;
  NRESU = 0; I=ARG(0);
  IF ARG(0) ¬= ' ' THEN DO ;IF TABCOURS.HEUR(I) = ARG(1)
                              THEN BRESU = '1'B;
                              ELSE BRESU = '0'B;
                         END;
                    ELSE DO ; DO J=1  BY 1  WHILE (TABCOURS.HEUR(J)
                              ¬= 0);
                              IF TABCOURS.HEUR(J)= ARG(1)
                              THEN DO; NRESU =NRESU + 1;
                                       RESU(NRESU)= J;
                                   END;
                              END;
                              IF NRESU = 0 THEN BRESU = '0'B;
                                           ELSE BRESU = '1'B;
                         END;
END PSREU1;


P55:PROC;
  CALL PSNUMS 1;
END P55;
PSNUMS 1:PROC;
  NRESU=0;
  IF ARG(0) ¬= ' ' THEN DO; IF ARG(0) = ARG(1) THEN
                               BRESU = '1'B; ELSE
                               BRESU = '0'B;
                         END;
                    ELSE DO; NRESU=1;
                         BRESU = '1'B;
                         RESU(1)=ARG(1);
                         END;
END PSNUMS 1;



P58:PROC;
  CALL PSUTIL2REP4;
END P58;
PSUTIL2REP4:PROC;
  NRESU=0; I=ARG(1);
  I =-I;
  N=1;
  DO K = I BY 1 WHILE (TOE(K) ¬= 0);
              ARG(1) =-TOE(K);
              CALL PSCSSL;
              IF (ARG(0) ¬= ' ' & BRESU='1'B)
                        THEN GO TO FIN1;
              IF ARG(0) = ' ' THEN DO;BOUCL(NB)=RESU(1);
                                      NB=NB+1;
                                   END;
```

```
   END;
   IF (ARG(0) = ' ' & NRESU = '1'B) THEN
               DO; RESU(1) = NOUCL(1);
                   IF NB=2 THEN GOTO FIN1;
                   L=2;NB=NB-1;
                   DO I=2 TO NB; INDIC = 0;
                      DO K=1 TO L-1;
                         IF NOUCL(I)  = RESU(K) THEN INDIC = 1;
                      END;
                      IF INDIC = 0 THEN DO; RESU(L)=NOUCL(I);
                                            L=L+1;
                                        END;
                   END;
                   NRESU=L-1;
               END;
   FIN1: END PSUTIL2REP4;
 PSCSSL:PROC;
   NRESU = 0; I=ARG(1);
   IF ARG(0) = ' ' THEN DO; NRESU= 1 ;
                            BRESU='1'B;
                       RESU(1)=TABCOURS.SALLE(1);
                            END;
              ELSE IF TABCOURS.SALLE(I) = ARG(0)
                            THEN BRESU = '1'B;
                            ELSE BRESU ='0'B;
 END PSCSSL;


 PSU:PROC;
   CALL PSUTIL1REP6;
   CALL PSNUMCS;
 END PSU;
 PSUTIL1REP6:PROC;
   I=ARG(1);
   I=-I;
   I=TOP(I);
   ARG(1)=I;
 END PSUTIL1REP6;
 PSNUMCS:PROC;
   NRESU = 0; I=ARG(1);
   IF ARG(0) ^= ' ' THEN DO;
            IF TABCOURS.NOCOURS(I) = ARG(0) THEN BRESU = '1'B;
                                            ELSE BRESU = '0'B;
                            END;
                       ELSE DO; NRESU=1;
                            BRESU='1'B;
                            RESU(1)=TABCOURS.NOCOURS(I);
                            END;
 END PSNUMCS;


 PS2:PROC;
   CALL PSUTIL4REP6;
   CALL PSNUMP;
 END PS2;
 PSUTIL4REP6:PROC;
   I=ARG(1);
   I=-I;
   I=TOP(I+1);
   ARG(1)=-1;
```

```
PSNOMP:PROC;
  NRESU=0; I=ARG(0);J=ARG(1);
  IF ARG(0) ¬= ' ' THEN DO;

          IF TABPROF.NOMPROF(J)=I THEN BRESU = '1'B;
                                  ELSE BRESU = '0'B;
                    END;
               ELSE DO; NRESU=1;
                        BRESU='1'B;
                        RESU(1)= TABPROF.NOMPROF(J);
                    END;
END PSNOMP;

P64:PROC;
  CALL PSUTIL1REP6;
  CALL PSCSCL;
  CALL PSUTIL6REP6;
  CALL PSNOCL;
END P64;
PSCSCL:PROC;
  NRESU = 0; I=ARG(0);J=ARG(1);
  IF ARG(0) ¬= ' ' THEN DO;
        IF TABCOURS.CLASS(J) = I THEN BRESU='1'B;
                                 ELSE BRESU='0'B;
                    END;
               ELSE DO; NRESU=1;
                        BRESU='1'B;
                        RESU(1)=TABCOURS.CLASS(J);
                    END;
END PSCSCL;
PSUTIL6REP6:PROC;
  IF ARG(0) = ' ' THEN ARG(1)=RESU(1);
END PSUTIL6REP6;
PSNOCL:PROC;
  NRESU=0; I=ARG(0);J=ARG(1);
  IF ARG(0) ¬= ' ' THEN DO;
        IF TABCLASS(J,1) = I THEN BRESU='1'B;
                             ELSE BRESU = '0'B;
                    END;
               ELSE DO; NRESU=1;
                        BRESU='1'B;
                        RESU(1)=TABCLASS(J,1);
                    END;
END PSNOCL;



P66:PROC;
  CALL PSUTIL1REP4ET5;
  CALL PSCREREP64;
END P66;
```

```
PSUTIL1REP4ETS:PROC;
  I=ARG(0);J=ARG(1);K=ARG(2);N = 1  ;
  J=-J;
  I=-I;
  DO L = 0 BY 1 WHILE(TOE(L) ^= 0);
    M=-TOE(L);
    IF TABCOURS.SALLE(M) = K   THEN DO;
            ARG(1)=M;
             IF ARG(0) ^= ' ' THEN ARG(0) =-TOE(I+1);
            CALL PSCSPF; IF NRESU = 1 THEN DO;
            BOUCL (N)=M;BOUCL(N+1)=RESU(1);
            N=N+2;
                                          END;
                                    ELSE IF BRESU = '1'B
                                          THEN GO TO FIN3;
            END;
  END;
  NRESU=N-1;
FIN3:END PSUTIL1REP4ETS;
PSCSPF:PROC;
  BRESU=0; I=ARG(0); J=ARG(1);
   IF ARG(0) ^= ' ' THEN DO;
          IF TABCOURS.PROF(J) = I THEN BRESU='1'B;
                                  ELSE BRESU='0'B;
                    END;
                ELSE DO; NRESU=1;
                         BRESU='1'B;
                         RESU(1)=TABCOURS.PROF(J);
                    END;
END PSCSPF;
PSCREREP64:PROC;
  N=NRESU/2;IF N ^=  0  THEN DO;
  DO I = 1 TO N;
    TOE(NIVOE+1)=BOUCL(2*I-1);
    TOE(NIVOE+2)=-BOUCL(2*I);
                          J=NIVOE+1;
                          RESU(I)=-J;
                          TOE(NIVOE+3)=0;
                          NIVOE=NIVOE+3;
  END;
  NRESU=N;
  END;
END PSCREREP64;



Pgm:PROC;
  CALL PSUTIL1REP4ETS;
  CALL PSUTIL9REP6;
END Pgm;
PSUTIL9REP6:PROC;
  N=NRESU/2;IF N ^= 0 THEN DO;
        DO I=1 TO N;
          J = BOUCL(2*I);
          RESU(I)=TABPROF.NUMPROF(J);
END;
        NRESU=N;
        END;
END PSUTIL9REP6;
```

```
P50:PROC;
   CALL PSUTIL1REP4;
   CALL PSJOS;
END P50;
PSUTIL1REP4:PROC;
   I=ARG(1);
I =-1;
   I=TOE(I);
     ARG(1)=-1;
END PSUTIL1REP4;
PSJOS:PROC; DCL R CHAR(8);
   I=ARG(1);
   R=TARCOURS.JOURHSEM(I);
   NRESU=0;
   IF ARG(0)=' ' THEN DO ; NRESU=1 ;
                           BRESU = '1'B ;
                           RESU(1) = R;
                      END;
                 ELSE DO; IF R = ARG(0) THEN BRESU ='1'B;
                                       ELSE BRESU = '0'B;
                      END;
END PSJOS;


P52:PROC;
  CALL PSUTIL1REP4;
  CALL PSHEU;
END P52;
PSHEU:PROC;
  I=ARG(1);
  J=TARCOURS.HEUR(I);
  NRESU=0;
  IF ARG(0) = ' ' THEN DO ; NRESU = 1;
                            BRESU = '1'B;
                            RESU(1) =J ;
                      END;
                 ELSE DO ; IF J = ARG(0) THEN BRESU = '1'B;
                                       ELSE BRESU = '0'B;
                      END;
END PSHEU;



P61:PROC;
 CALL PSUTIL2REP6;
 CALL PSNUMCS#1;
 CALL PSUTIL3REP6;
 CALL PSCSPF;
 CALL PSCREREP6;
END P61;
PSUTIL2REP6:PROC;
  IF ARG(0) ¬= ' ' THEN DO; I=ARG(0);
                            I=-I;
                           I=TOE(I);
                                    BOUCL(I) =-TOE(I+1);
                           ARG(0)=I;
END PSUTIL2REP6;
```

```
PSNUMCS.1:PROC;
  NRESU=0; I=ARG(0); J=ARG(1);
  IF ARG(0) ¬= ' ' THEN DO;
         IF TABCOURS.NOCOURS(I) = J THEN BRESU='1'B;
                                    ELSE BRESU='0'B;
                       END;
                  ELSE DO; NRESU=1;
                     BRESU='1'B;
                     RESU(1)=J;
                    END;
END PSNUMCS.1;
PSUTIL3REP6:PROC;
  IF ARG(0) ¬= ' ' THEN ARG(0) = BOUCL(1);
                    BOUCL(1) = RESU(1);
END PSUTIL3REP6;
PSCREREP6:PROC;
  IF ARG(0) = ' ' THEN DO; TOE(NIVOE+1)=BOUCL(1);
                           I=RESU(1);
                           TOE(NIVOE+2)=-1;
                           TOE(NIVOE+3)=0;
                           I=NIVOE+1;
                           RESU(1)=-I;
                           NIVOE=NIVOE + 3;
                       END;
END PSCREREP6;


P69:PROC;
  CALL PSUTIL1REP4ET5;
  CALL PSUTIL1JREP6;
END P69;
PSUTIL1JREP6:PROC;
  N=NRESU/2; IF N ¬= 0 THEN DO;
        DO I=1 TO N;
          J = BOUCL(2*I-1);
          RESU(I)= TABCOURS.CLASS(J);
        END;
        NRESU=N;
        END;
END PSUTIL1JREP6;




S20:PROC; N=1;
  DO I=1 BY 1 WHILE (TABCOURS.NOCOURS(I) ¬= 0);
    RESU(I)=I;
    N=N+1;
  END;
  NRESU=N-1;
  CALL PSCREREP4;
  BRESU='1'B;
END S20;
```

ANNEXE 2: autres exemples d'interrogation de la base :

a) requête n° I : " nom du professeur et n° de la classe concernée par le cours 4 ?".

Le filtre correspondant est :

$$'4'\ \ NCS^{-I}\ \ y_I < NPF\ x_I\ ,\ NCL\ \ x_2 >$$

(les inconnues résultat sont donc x_I et x_2)

Le programme maquette obtenu est le suivant : (cf. signification des symboles p. 4.I2)

```
**********************
*PROGRAMME MAQUETTE*
**********************
2 0 1,1000 1P61('4',X 3),1000 1P62(X 3,X 1),1000 1P64(X 3,X 2),3 1,9 ,
```

Les résultats intermédiaires correspondant aux appels de P6I, P62, et P64 sont donnés P. 4.27. ; le résultat final se lit dans le troisième tableau, en INPLANW(3) = 6 de W : on y lit un triplet unique de valeurs de x_I, x_2, y_I ; le professeur est donc MARTIN et le n° de la classe 2 (on peut le vérifier p.4.7).

Il s'agit d'un exemple très simple où une seule occurrence d'O.E. associé au type de repères (3) de LOG est utile.

```
=               (1)  MML(3=              3            COMPTPIL=               0;
1)='          -1           '     W(2)='##################';
PLANW(1)=                  1;
CUPW(1)='0'B;

E(1)=                4         TOE(2)=              -1          TOE(3)=            0;


=               (2)  MML(3=              5            COMPTPIL=               0;
1)='          -1           '     W(2)='##################'     W(3)='MARTIN          '     W(4)='       -1'
5)='##################';
PLANW(1)=                  1     IMPLANW(2)=              3;
CUPW(1)='1'B         RECUP.(2)='0'B;

E(1)=                4         TOE(2)=              -1          TOE(3)=            0;


=               (3)  MML(3=              10           COMPTPIL=               0;
1)='          -1           '     W(2)='##################'     W(3)='MARTIN          '     W(4)='       -1'
5)='##################'          W(6)='MARTIN          '       W(7)='          2     '     W(8)='       -1'
9)='##################';
PLANW(1)=                  1     IMPLANW(2)=              3     IMPLANW(3)=          6;
CUPW(1)='1'B         RECUPW(2)='1'B       RECUPW(3)='0'B;

E(1)=                4         TOE(2)=              -1          TOE(3)=            0;
```

b) requête n° 2: "nom du professeur, n° classe, n° cours ayant lieu le MARDI à I5h dans la salle 2 ?".

Le filtre correspondant est :

$$\langle \text{'MARDI'} \quad JS^{-I},\ \text{'I5'}\ H^{-I} \rangle x_I \ HS \ x_2 \ NS \ \text{'2'} \ \underline{et} \langle x_I, x_2 \rangle CONT$$

$$y_I \langle NCS \ x_3,\ NPF \ x_4,\ NCL \ x_5 \rangle$$

(les inconnues résultat sont x_3, x_4, x_5).

Le programme maquette obtenu est le suivant :

```
**********************
*PROGRAMME MAQUETTE*
**********************
```

```
2 0 0,1000 1P55('2',X 2),1000 1P51('MARDI',X 1),1000 1P53('15',X 1),
1000 1P58(X 1,X 2),3 1,4 1,2 1 1,1000 2P66(X 1,X 2,X 6),1000 1
P60(X 6,X 3),1000 1P62(X 6,X 4),1000 1P64(X 6,X 5),3 2,5 1,6 1,9 .
```

Les 8 résultats intermédiaires correspondant aux 8 appels de procédures principales sont donnés dans les pages suivantes. Le résultat final se lit dans le huitième tableau, en IMPLANW(8) = I de W; on y lit un 5-uple unique de valeurs correspondant à x_I, x_2, x_3, x_4, x_5 ; le n° de cours est donc 4, le professeur MARTIN et le n° de classe 2.

Dans cet exemple sont générés les deux types d' O.E.

```
NW=            (1)   MBLIP=              3             COMPTPIL=            0;
W(1)='2                  '      W(2)='##################';
IMPLANW(1)=             1;
RECUPW(1)='0'B;

NW=            (2)   MBLIP=              6             COMPTPIL=            0;
 (1)='2                  '      W(2)='##################'    W(3)='          -1          '     W(4)='          -5'
 (5)='##################';
MPLANW(1)=              1       IMPLANW(2)=              3;
ECUPW(1)='0'B            RECUPW(2)='0'B;

OE(1)=               -3         TOE(2)=               -4      TOE(3)=              -5        TOE(4)=        0
OE(5)=               -6         TOE(6)=               -7      TOE(7)=               0;

N=             (3)   MBLIP=              6             COMPTPIL=            0;
 1)='2                   '      W(2)='##################'    W(3)='          -1          '     W(4)='          -5'
 5)='##################'        W(6)='         -1        '   W(7)='##################';
MPLANW(1)=              1       IMPLANW(2)=              3    IMPLANW(3)=                6;
CUPW(1)='0'B         RECUPW(2)='1'B      RECUPW(3)='0'B;

E(1)=                -3         TOE(2)=               -4      TOE(3)=              -5        TOE(4)=        0
E(5)=                -6         TOE(6)=               -7      TOE(7)=               0        TOE(8)=       -3
E(9)=                -4         TOE(10)=              -5      TOE(11)=              0;

W=             (4)   MBLIP=             11             COMPTPIL=            0;
(1)='2                   '      W(2)='##################'    W(3)='          -1          '     W(4)='          -5'
(5)='##################'        W(6)='         -1        '   W(7)='##################'         W(8)='          -1'
(9)='2                   '      W(10)='##################';
MPLANW(1)=              1       IMPLANW(2)=              3    IMPLANW(3)=                6     IMPLANW(4)=   8;
ECUPW(1)='1'B        RECUPW(2)='1'B      RECUPW(3)='1'B     RECUPW(4)='0'B;

OE(1)=               -3         TOE(2)=               -4      TOE(3)=              -5        TOE(4)=        0
OE(5)=               -6         TOE(6)=               -7      TOE(7)=               0        TOE(8)=       -3
OE(9)=               -4         TOE(10)=              -5      TOE(11)=              0;
(1)='          -1          '    W(2)='2                  '    W(3)='##################'      W(4)='          -5'
```

```
5)='2                    '       W(5)='      -12        '    W(3)='#################'        W(4)='       -1'
PLANW(1)=                1       IMPLANW(2)=            3    W(7)='#################';
PLANW(5)=                4;                                  IMPLANW(3)=              6      IMPLANW(4)=    1
CUPW(1)='1'B        RECUPW(2)='1'B      RECUPW(3)='1'B      RECUPW(4)='0'B        RECUPW(5)='0'B;

E(1)=            -3      TUE(2)=             -4              TUE(3)=           -5        TUE(4)=      0
E(5)=            -6      TUE(6)=             -7              TUE(7)=            0        TUE(8)=     -3
E(9)=            -4      TUE(10)=            -5              TUE(11)=           0        TUE(12)=     4
E(13)=           -1      TUE(14)=             0;

=           (6) ANULI=            13          COMPTPIL=               0;
1)='      -1        '    W(2)='2                 '    W(3)='#################'       W(4)='       -1'
5)='2               '    W(6)='      -12         '    W(7)='#################'       W(8)='       -1'
9)='2               '    W(10)='           4     '    W(11)='      -12         '      W(12)='########
PLANW(1)=        1       IMPLANW(2)=             3    IMPLANW(3)=              6       IMPLANW(4)=    1
PLANW(5)=        4       IMPLANW(6)=             8;
CUPW(1)='1'B       RECUPW(2)='1'B       RECUPW(3)='1'B       RECUPW(4)='0'B       RECUPW(5)='1'B       RECUPW

E(1)=            -4      TUE(2)=             -4              TUE(3)=           -5        TUE(4)=      0
E(5)=            -6      TUE(6)=             -7              TUE(7)=            0        TUE(8)=     -3
E(9)=            -4      TUE(10)=            -5              TUE(11)=           0        TUE(12)=     4
E(13)=           -1      TUE(14)=             0;

=           (7) ANULI=            19          COMPTPIL=               0;
1)='      -1        '    W(2)='2                 '    W(3)='#################'       W(4)='       -1'

5)='2               '    W(6)='      -12         '    W(7)='#################'       W(8)='       -1'
9)='2               '    W(10)='           4     '    W(11)='      -12         '      W(12)='########
3)='      -1        '    W(14)='2                     W(15)='           4     '      W(16)='MARTIN  '
7)='     -12        '    W(18)='####################';
PLANW(1)=        1       IMPLANW(2)=             3    IMPLANW(3)=              6       IMPLANW(4)=    1
PLANW(5)=        4       IMPLANW(6)=             8    IMPLANW(7)=             13;
CUPW(1)='1'B       RECUPW(2)='1'B       RECUPW(3)='1'B       RECUPW(4)='0'B       RECUPW(5)='1'B       RECUPW
CUPW(7)='0'B

(1)=             -3      TUE(2)=             -4              TUE(3)=           -5        TUE(4)=      0
(5)=             -6      TUE(6)=             -7              TUE(7)=            0        TUE(8)=     -3
(9)=             -4      TUE(10)=            -5              TUE(11)=           0        TUE(12)=     4
(13)=            -1      TUE(14)=             0;
```

```
I=             (8)  MWLI =               /              COMPTPIL=               0;
(1)='        -1        '      W(2)='2                '          W(3)='             4      '      W(4)='MARTIN '
(5)='               2        '      W(6)='##################':
MPLANW(1)=              1        IMPLANW(2)=            3        IMPLANW(3)=              6        IMPLANW(4)= 1
MPLANW(5)=              4        IMPLANW(6)=            8        IMPLANW(7)=             13        IMPLANW(8)= 1
ECUPW(1)='1'B        RECUPW(2)='1'B        RECUPW(3)='1'B        RECUPW(4)='1'B        RECUPW(5)='1'B        RECU
ECUPW(7)='1'B        RECUPW(8)='0'B;

OE(1)=              -3        TOE(2)=              -4        TOE(3)=              -5        TOE(4)=   0
OE(5)=              -6        TOE(6)=              -7        TOE(7)=               0        TOE(8)=  -3
OE(9)=              -4        TOE(10)=             -5        TOE(11)=              0        TOE(12)=  4
OE(13)=             -1        TOE(14)=              0;
```

c) requête n° 4: "horaires de travail de tous les professeurs ?".

Le filtre correspondant est :

$$\langle x_I \ JS^{-I}, x_2 \ H^{-I} \rangle \ x_3 \ HS \ x_4 \ \underline{et} \ \langle x_3, x_4 \rangle \ CONT.NPF \ x_5$$

(les variables résultats sont x_I, x_2 et x_5).

Le programme maquette obtenu est le suivant :

```
**********************
*PROGRAMME MAQUETTE*
**********************

2 0 0,1000 1S20(' ',X 3),1000 1P58(X 3,X 4),1000 1P50(X 3,X 1),

1000 1P52(X 3,X 2),3 1,4 1,2 1 1,1000 2P68(X 3,X 4,X 5),3 2,5 1,8 1

,9 .
```

(où S20 est une procédure de génération d'extrémités, ici de repères de date, qui sont affectés à la variable x_3).

Les 5 résultats intermédiaires sont donnés dans les pages suivantes. Le résultat final se lit en IMPLANW(5) = I de W ; il s'agit de 7 5-uples $(x_I, x_2, x_3, x_4, x_5)$.

```
W=              (1)   NMLIN=          6          COMPTPIL=              0;
(1)='     -1        '     W(2)='      -3        '    W(3)='       -5        '    W(4)='      -9
(5)='################'  ';
MPLANW(1)=          1;
ECUPW(1)='0';

OE(1)=           -1     TOE(2)=          0      TOE(3)=           -2     TOE(4)=     0
OE(5)=           -3     TOE(6)=         -4      TOE(7)=           -5     TOE(8)=     0
OE(9)=           -6     TOE(10)=        -7      TOE(11)=           0;

W=              (2)   NMLIN=         21          COMPTPIL=              0;
(1)='     -1        '     W(2)='      -3        '    W(3)='       -5        '    W(4)='      -9
(5)='################'  W(6)='      -1        '    W(7)='        1       '    W(8)='      -3
(9)='        1      '     W(10)='     -5        '    W(11)='       1        '    W(12)='     -5
(13)='       2      '     W(14)='     -5        '    W(15)='       3        '    W(16)='     -9
(17)='       1      '     W(18)='     -9        '    W(19)='       3        '    W(20)='##########
MPLANW(1)=          1     IMPLANW(2)=          0;
ECUPW(1)='1'      RECUPW(2)='0';

OE(1)=           -1     TOE(2)=          0      TOE(3)=           -2     TOE(4)=     0
OE(5)=           -3     TOE(6)=         -4      TOE(7)=           -5     TOE(8)=     0
OE(9)=           -6     TOE(10)=        -7      TOE(11)=           0;

JW=             (3)   NMLIN=         43          COMPTPIL=              0;
W(1)='    -1        '     W(2)='      -3        '    W(3)='       -5        '    W(4)='      -9
W(5)='################'  W(6)='      -1        '    W(7)='        1       '    W(8)='      -3
W(9)='       1      '     W(10)='     -5        '    W(11)='       1        '    W(12)='     -5
W(13)='      2      '     W(14)='     -5        '    W(15)='       3        '    W(16)='     -9
W(17)='      1      '     W(18)='     -9        '    W(19)='       3        '    W(20)='##############
W(21)='LUNDI        '     W(22)='     -1        '    W(23)='       1        '    W(24)='LUNDI
W(25)='    -3       '     W(26)='      1        '    W(27)='MARDI           '    W(28)='      -5
W(29)='      1      '     W(30)='MARDI          '    W(31)='      -5        '    W(32)='       2
W(33)='MARDI        '     W(34)='     -5        '    W(35)='       3        '    W(36)='MARDI
W(37)='    -9       '     W(38)='      1        '    W(39)='MARDI           '    W(40)='      -9
W(41)='     5       '     W(42)='##################';
IMPLANW(1)=          1     IMPLANW(2)=          6      IMPLANW(3)=          21;
RECUPW(1)='1'      RECUPW(2)='1'      RECUPW(3)='0';

TOE(1)=          -1     TOE(2)=          0      TOE(3)=           -2     TOE(4)=     0
TOE(5)=          -3     TOE(6)=         -4      TOE(7)=           -5     TOE(8)=     0
TOE(9)=          -6     TOE(10)=        -7      TOE(11)=           0;
```

```
W=                 MALLE=              72       COMPTPIL=              0;
(1)='        -1       '  W(2)='        -3         '  W(3)='       -5          '  W(4)='        -9      '
(5)='################'  W(6)='        -1         '  W(7)='               1   '  W(8)='        -3      '
(9)='            1    '  W(10)='        -5        '  W(11)='              1   '  W(12)='        -5     '
(13)='           2    '  W(14)='        -5        '  W(15)='              3   '  W(16)='        -9     '
(17)='           1    '  W(18)='        -9        '  W(19)='              3   '  W(20)='#############'
(21)='LUNDI           '  W(22)='        -1        '  W(23)='              1   '  W(24)='LUNDI        '
(25)='       -3       '  W(26)='              1   '  W(27)='MARDI             '  W(28)='        -5     '
(29)='           1    '  W(30)='MARDI             '  W(31)='       -5         '  W(32)='              2'
(33)='MARDI           '  W(34)='        -5        '  W(35)='              3   '  W(36)='MARDI        '
(37)='       -9       '  W(38)='              1   '  W(39)='MARDI             '  W(40)='        -9     '
(41)='           3    '  W(42)='################' W(43)='LUNDI             '  W(44)='        10     '
(45)='       -1       '  W(46)='              1   '  W(47)='LUNDI             '  W(48)='        11     '
(49)='       -3       '  W(50)='              1   '  W(51)='MARDI             '  W(52)='        15     '
(53)='       -5       '  W(54)='              1   '  W(55)='MARDI             '  W(56)='        15     '
(57)='       -5       '  W(58)='              2   '  W(59)='MARDI             '  W(60)='        15     '
(61)='       -5       '  W(62)='              3   '  W(63)='MARDI             '  W(64)='        16     '
(65)='       -9       '  W(66)='              1   '  W(67)='MARDI             '  W(68)='        16     '
(69)='       -9       '  W(70)='              3   '  W(71)='################';
MPLANW(1)=          1    IMPLANW(2)=          6    IMPLANW(3)=          21    IMPLANW(4)=

ECUPW(1)='1'R    RECUPW(2)='1'J    RECUPW(3)='1'R    RECUPW(4)='1'R;

OF(1)=       -1    TUE(2)=        0    TUE(3)=       -2    TUE(4)=        0
OF(5)=       -3    TUE(6)=       -4    TUE(7)=       -5    TUE(8)=        0
OF(9)=       -6    TUE(10)=      -7    TUE(11)=       0;
```

```
W=              5    N-LIN=              37        COMPIPILE=            0;
(1)='LUNDI              '   W(2)='           10      '   W(3)='           -1           '   W(4)='                1'
(5)='MARTIN             '   W(6)='LUNDI              '   W(7)='           11           '   W(8)='          -3     '
(9)='            1      '   W(10)='HUBERT            '   W(11)='MARDI                  '   W(12)='         15      '
(13)='       -5         '   W(14)='            1      '   W(15)='MIKE                   '   W(16)='MARDI            '
(17)='       15         '   W(18)='      -5           '   W(19)='             2         '   W(20)='MARTIN           '
(21)='MARDI             '   W(22)='      15           '   W(23)='        -5             '   W(24)='                3'
(25)='HUBERT            '   W(26)='MARDI              '   W(27)='        16             '   W(28)='          -9     '
(29)='            1     '   W(30)='HUBERT             '   W(31)='MARDI                  '   W(32)='          16     '
(33)='       -9         '   W(34)='            3      '   W(35)='MIKE                   '   W(36)='###############
MPLANW(1)=           1      IMPLANW(2)=          6      IMPLANW(3)=           21      IMPLANW(4)=
MPLANW(5)=           1[illegible]

ECUPW(1)='1'B     RECUPW(2)='1'B     RECUPW(3)='1'B     RECUPW(4)='1'B     RECUPW(5)='0'B;

OE(1)=       -1      TOE(2)=        0      TOE(3)=       -2      TOE(4)=        0
OE(5)=       -3      TOE(6)=       -4      TOE(7)=       -5      TOE(8)=        0
OE(9)=       -6      TOE(10)=      -7      TOE(11)=       0;
                ***********************************
```

Sur le plan du système PIVOINES lui-même, le bilan de ce travail est relativement aisé à établir : l' extension du sous-système d' interrogation à de multiples structures externes pouvant acceuillir des types de repères sans équivalent direct au niveau interne est réalisé.

Il faut cependant constater et peut être regretter l' aspect "procédural" selon lequel la correspondance des types de repères est prise en compte.

Un résultat complémentaire est constitué par l'approfondissement de l'analyse du contenu des structures LOG et PHYLOG dans les chapitres I et 2. Ceci doit faciliter la mise en oeuvre des applications.

Il est beaucoup plus difficile et plus aléatoire de vouloir tirer des leçons générales de ce travail, dont ce n'était pas d' ailleurs l' objet. On peut cependant avancer,avec prudence, les quelques remarques générales suivantes :

+ en ce qui concerne les problèmes d' architecture des S.G.B.D., une correspondance directe entre les niveaux externe et interne, sans intermédiaire par un niveau conceptuel, semble intéressante, au moins pour la fonction interrogation étudiée ici.
+ en ce qui concerne la correspondance entre les structures externes et internes, un maximum de flexibilité doit être offert.
+ cette flexibilité est en particulier indispensable pour rendre compte des diverses visions des utilisateurs, qui peuvent être très différentes les unes des autres, ainsi que nous l'avons montré.

Pour terminer, indiquons que parmi les nombreux axes de développement du projet PIVOINES, ceux qui compléteront le plus directement ce travail sont :

+ l' amélioration du sous-système interrogation (en complétant les possibilités du langage d' interrogation avec en particulier le "cadrage"; en améliorant les critères de choix au niveau du programme maquette; en améliorant la "maniabilité" du système).
+ conception détaillée et mise en oeuvre des sous-systèmes modification et acquisition; c'est à ce moment là seulement, à notre sens, qu'il sera possible de juger complètement des solutions et hypothèses retenues pour cette partie interrogation.

- - - - - -

NOM DE L'ETUDIANT : LONCHAMP Jacques

NATURE DE LA THESE : Doctorat de Spécialité (Informatique)

VU, APPROUVE

et PERMIS D'IMPRIMER

NANCY LE 03 MAI 1978 3016

LE PRESIDENT DE L'UNIVERSITE DE NANCY I