



HAL
open science

Transformation de structures de données dans le cadre d'un système automatique d'aide à la conception de systèmes d'information

Salwa Taha Ahmed

► **To cite this version:**

Salwa Taha Ahmed. Transformation de structures de données dans le cadre d'un système automatique d'aide à la conception de systèmes d'information. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1979. Français. NNT : . tel-00290423

HAL Id: tel-00290423

<https://theses.hal.science/tel-00290423>

Submitted on 25 Jun 2008

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

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

THESE

présentée à

Institut National Polytechnique de Grenoble

pour obtenir le grade de

**DOCTEUR INGENIEUR
(Génie Informatique)**

par

Salwa TAHA AHMED



**TRANSFORMATION DE STRUCTURES DE DONNEES
DANS LE CADRE D'UN SYSTEME AUTOMATIQUE
D'AIDE A LA CONCEPTION
DE SYSTEMES D'INFORMATION.**



Thèse soutenue le 10 décembre 1979 devant la Commission d'Examen :

Président : Monsieur G. VEILLON

**Examineurs : Messieurs C. DELOBEL
F. PECCOUD
J. KOULOUMDJAN**

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Année universitaire 1977-1978

Président : M. Philippe TRAYNARD

Vice-présidents : M. René PAUTHENET

M. Georges LESPINARD

PROFESSEURS TITULAIRES

MM. BENOIT Jean	Electronique - automatique
BESSON Jean	Chimie minérale
BLOCH Daniel	Physique du solide - cristallographie
BONNETAIN Lucien	Génie chimique
BONNIER Etienne	Métallurgie
* BOUDOURIS Georges	Electronique - automatique
BRISSONNEAU Pierre	Physique du solide - cristallographie
BUYLE-BODIN Maurice	Electronique - automatique
COUMES André	Electronique - automatique
DURAND Francis	Métallurgie
FELICI Noël	Electronique - automatique
FOULARD Claude	Electronique - automatique
LANCIA Roland	Electronique - automatique
LONGEQUEUE Jean-Pierre	Physique nucléaire corpusculaire
LESPINARD Georges	Mécanique
MOREAU René	Mécanique
PARIAUD Jean-Charles	Chimie - physique
PAUTHENET René	Electronique - automatique
PERRET René	Electronique - automatique
POLOUJADOFF Michel	Electronique - automatique
TRAYNARD Philippe	Chimie - physique
VEILLON Gérard	Informatique fondamentale et appliquée
* en congé pour études	

PROFESSEURS SANS CHAIRE

MM. BLIMAN Samuël	Electronique - automatique
BOUVARD Maurice	Génie mécanique
COHEN Joseph	Electronique - automatique
GUYOT Pierre	Métallurgie physique
LACOUME Jean-Louis	Electronique - automatique
JOUBERT Jean-Claude	Physique du solide - cristallographie

.../...

MM. ROBERT André	Chimie appliquée et des matériaux
ROBERT François	Analyse numérique
ZADWORNY François	Electronique - automatique

MAITRES DE CONFERENCES

MM. ANCEAU François	Informatique fondamentale et appliquée
CHARTIER Germain	Electronique - automatique
CHIAVERINA Jean	Biologie, biochimie, agronomie
IVANES Marcel	Electronique - automatique
LESIEUR Marcel	Mécanique
MORET Roger	Physique nucléaire - corpusculaire
PIAU Jean-Michel	Mécanique
PIERRARD Jean-Marie	Mécanique
SABONNADIÈRE Jean-Claude	Informatique fondamentale et appliquée
Mme SAUCIER Gabrielle	Informatique fondamentale et appliquée
M. SOHM Jean-Claude	Chimie Physique

CHERCHEURS DU C.N.R.S. (Directeur et Maîtres de Recherche)

M. FRUCHART Robert	Directeur de Recherche
MM. ANSARA Ibrahim	Maître de Recherche
BRONOEL Guy	Maître de Recherche
CARRE René	Maître de Recherche
DAVID René	Maître de Recherche
DRIOLE Jean	Maître de Recherche
KLEITZ Michel	Maître de Recherche
LANDAU Ioan-Doré	Maître de Recherche
MATHIEU Jean-Claude	Maître de Recherche
MERMET Jean	Maître de Recherche
MUNIER Jacques	Maître de Recherche

Personnalités habilitées à diriger des travaux de recherche (décision du Conseil Scientifique) E.N.S.E.E.G.

MM. BISCONDI Michel	Ecole des Mines St. Etienne (dépt. Métallurgie)
BOOS Jean-Yves	Ecole des Mines St. Etienne (Métallurgie)
DRIVER Julian	Ecole des Mines St. Etienne (Métallurgie)

.../...

MM. KOBYLANSKI André
 LE COZE Jean
 LESBATS Pierre
 LEVY Jacques
 RIEU Jean
 SAINFORT
 SOUQUET
 CAILLET Marcel
 COULON Michel
 GUILHOT Bernard
 LALAUZE René
 LANCELOT Francis
 SARRAZIN Pierre
 SOUSTELLE Michel
 THEVENOT François
 THOMAS Gérard
 TOUZAIN Philippe
 TRAN MINH Canh

Ecole des Mines St. Etienne (Métallurgie)
 Ecole des Mines St. Etienne (Métallurgie)
 Ecole des Mines St. Etienne (Métallurgie)
 Ecole des Mines St. Etienne (Métallurgie)
 Ecole des Mines St. Etienne (Métallurgie)
 C.E.N. Grenoble (Métallurgie)

U.S.M.G.

Ecole des Mines St. Etienne (Chim. Min. Ph.)
 Ecole des Mines St. Etienne (Chim. Min. Ph.)
 Ecole des Mines St. Etienne (Chim. Min. Ph.)
 Ecole des Mines St. Etienne (Chim. Min. Ph.)
 Ecole des Mines St. Etienne (Chim. Min. Ph.)
 Ecole des Mines St. Etienne (Chim. Min. Ph.)
 Ecole des Mines St. Etienne (Chim. Min. Ph.)
 Ecole des Mines St. Etienne (Chim. Min. Ph.)
 Ecole des Mines St. Etienne (Chim. Min. Ph.)
 Ecole des Mines St. Etienne (Chim. Min. Ph.)
 Ecole des Mines St. Etienne (Chim. Min. Ph.)

E.N.S.E.R.G.

MM. BOREL
 KAMARINOS

Centre d'études nucléaires de Grenoble
 Centre national recherche scientifique

E.N.S.E.G.P.

M. BORNARD
 Mme CHERUY
 MM. DAVID
 DESCHIZEAUX

Centre national recherche scientifique
 Centre national recherche scientifique
 Centre national recherche scientifique
 Centre national recherche scientifique

- à mon père

- à ma mère

qui m'ont toujours encouragés

- avec mes amis.

Je tiens particulièrement à remercier ici :

Monsieur Gérard VEILLON, Professeur à l'Institut National Polytechnique de Grenoble, pour l'honneur qu'il me fait de présider ce jury,

Monsieur Claude DELOBEL, Professeur à l'Université de Grenoble I, qui m'a accueilli dans son équipe et qui est à l'origine de mon sujet de recherche. Je veux aussi lui exprimer ma plus sincère gratitude pour la confiance qu'il m'a toujours témoignée, pour sa totale disponibilité à mon égard, pour ses conseils indispensables, ces critiques constructives et encouragements sans lesquels ce travail n'aurait pas été possible,

Monsieur Jacques KOULOUMDJIAN, Professeur à l'I.U.T Informatique de Lyon, pour avoir accepté de juger ce travail et pour l'aide qu'il m'a apportée par ces remarques et ces critiques concernant le manuscrit de cette thèse,

Monsieur François PECCOUD, Professeur à l'I.U.T II Informatique de Grenoble, qui a bien voulu participer au jury,

Monsieur Michel LEONNARD, Professeur à l'Université de Genève avec qui j'ai eu des échanges fructueux. Il a toujours répondu avec toute compétence à mes nombreuses questions et m'a beaucoup aidé de ses conseils dans le développement de mon travail,

Tous ceux qui ont contribué par leurs critiques et leurs suggestions à l'amélioration de la rédaction du manuscrit ; je me dois de mentionner Andrée STIERS, Monique MORESTIN et André MORESTIN.

Madame Solange ROCHE qui a assuré avec beaucoup de patience et de gentillesse la dactylographie d'un manuscrit souvent illisible,

Enfin, Monsieur IGLESIAS et toute son équipe pour le soin qu'ils ont apporté à la réalisation matérielle de ce document.

PLAN DE THESE

AVANT-PROPOS

0. INTRODUCTION

- 0. 1 Présentation générale et problèmes de la conception des Systèmes de Bases de données actuels.
- 0. 2 Position du problème traité et objectifs de travail.

CHAPITRE I : Le Modèle Multigraphe

- 1.1 Définition de la notion de Multigraphe
- 1.2 Représentation d'un Multigraphe
 - 1.2.1 Implantation Informatique des catégories
 - 1.2.2 Opération sur les catégories
 - 1.2.3 Fonctions et procédures associées aux opérations sur les catégories
 - 1.2.4 Implantation Informatique des fonctions d'accès
- 1.3 Informations spécifiques au système informatique
- 1.4 Procédures spécifiques au système
- 1.5 Procédures spécifiques au processus de transformation de structures de données

CHAPITRE II : Les Principales étapes et l'objectif du processus de conception de systèmes de Bases de données

- 2.1 La structure de l'information et les besoins en informatique
- 2.2 La conception du schéma conceptuelle : l'approche relationnelle
- 2.3 Point de départ de la démarche de transformation de structures de données
 - 2.3.1 Etablissement du catalogue de constituants
 - 2.3.2 Représentation de la liste de relations fonctionnelles
- 2.4 Exemple
 - 2.4.1 Description de propriétés sémantiques d'un système d'information
 - 2.4.2 Construction de l'ensemble "DEP" de dépendance fonctionnelle de départ
 - 2.4.3 Construction de la table de constituants
 - 2.4.4 Obtention de la couverture minimale de l'ensemble "DEP"
 - 2.4.5 Etablissement du catalogue de la couverture minimale de DEP
- 2.5 Description de la vue relationnelle de départ : l'ensemble EREL de relations

- 2.6 Plan de l'étude-processus de transformation
- 2.7 Conclusion

CHAPITRE III La transformation d'un ensemble de relations en une structure réseau de type SOCRATE ou CODASYL

- 3.1 La transformation d'un ensemble "EREL" de relations en une structure relationnelle : "SREL"
 - 3.1.1 Définition d'une structure SREL
 - 3.1.2 Spécification d'une solution canonique
- 3.2 La transformation de la structure "SREL" en une structure opérationnelle : "SRELR"
 - 3.2.1 Définition d'une structure SRELR
 - 3.2.2 Modification de SREL
- 3.3 La transformation de la structure SRELR en une structure réseau indépendante de S.G.B.D actuels
 - 3.3.1 Caractéristiques générales d'une structure réseau
 - 3.3.2 Définition d'une structure réseau indépendante : "SRES"
 - 3.3.3 Contraintes associées à la structure SRES
- 3.4 La transformation de la structure SRES en une structure réseau SOCRATE ou CODASYL

CHAPITRE IV : La mise en oeuvre à partir du modèle Multigraphe du processus de transformation d'un ensemble de relation en une structure réseau en termes de S.G.B.D actuels

- 4.1 La transformation "EREL" → "SREL" - Construction du Multigraphe : "SREL"
 - 4.1.1 Construction des noeuds et rattachement des constituants et des clés
 - 4.1.2 Création des liaisons hiérarchiques
 - 4.1.3 Création des liaisons fonctionnelles
 - 4.1.4 Construction des arêtes hiérarchiques et fonctionnelles
 - 4.1.5 Conclusion
 - 4.1.6 Exemple
- 4.2 La transformation "SREL" → "SRELR"
 - 4.2.1 Opération de construction du Multigraphe SRELR
 - 4.2.1.1 Construction de noeuds et des arêtes supplémentaires
 - 4.2.1.2 Changement de type d'associations
 - 4.2.1.3 Rattachement d'un noeud à un autre
 - 4.2.2 La schéma SRELR
 - 4.2.3 Conclusion

4.3 La transformation : "SRELR" → "SRES"

4.3.1 Opération de transformation de SRELR

4.3.1.1 Destruction de la redondance de constituants

4.3.1.2 Création de liaisons bianires

4.3.1.3 Détermination de liaisons hiérarchiques

4.3.2 Le schéma "SRES"

4.4 La transformation de SRES en une structure SOCRATE ou CODASYL

4.4.1 Paramètre de la transformation et le besoin d'un apport d'informations externes

4.4.2 Exemple

CHAPITRE V : Conclusion et Prolongation du travail

AVANT PROPOS

Notre travail aborde un domaine informatique auquel sont consacrés aujourd'hui de nombreux efforts de recherche et de développement : La création des méthodes et des outils informatiques d'aide à la conception des systèmes d'information.

Notre apport est centré autour des mécanismes de traduction de structures de données, avec comme objectif la conception et la réalisation d'un système de transformation quasiautomatique, permettant d'affirmer que l'on peut construire, à partir d'un schéma relationnel, une structure réseau de type SOCRATE ou CODASYL.

Dans ce but, nous avons essayé de réaliser un logiciel général et un ensemble de fonctions permettant de faciliter le passage entre deux niveaux :

1) Le niveau conceptuel, qui à partir d'un ensemble de propriétés sémantiques donné comme informations extérieures, cherchera à construire convenablement une structure réseau compatible. Les propriétés sémantiques seront exprimées en termes de dépendances fonctionnelles, structurant les données élémentaires, que doit contenir la base en construction.

A ce niveau, la prise en compte de la composante temporelle est un complément nécessaire et important à considérer, pour aboutir à une structure opérationnelle, conforme à la réalité et aux vœux des usagers.

2) Le niveau d'implémentation logique, qui correspond pratiquement à l'expression en Langage de Définition de Données (L.D.D.) spécifique du Système de Gestion de Bases de Données (S.G.B.D.) utilisé de la structure réseau précédemment décidée.

Le support du logiciel, que nous présentons ici, est la notion de "Multigraphe", qui permet l'expression canonique de plusieurs structures de données.

Cet outil appliqué à la conception des systèmes de bases de données existants, nous a permis de définir une architecture fonctionnelle, servant de cadre de référence à la constitution effective de systèmes d'information, en perpétuelle mutation.

D. INTRODUCTION

D. 1 Présentation générale et problèmes de la conception des systèmes de bases de données actuels

Les bases de données constituent un phénomène marquant de notre époque dans l'évolution du traitement de l'information.

En effet, l'approche bases de données, introduite depuis une quinzaine d'années, avait eu pour tâche essentielle de supprimer les inconvénients multiples d'une informatique de gestion basée sur la création et l'exploitation de fichiers spécifiques. Dans ce but, cette nouvelle conceptualisation cherche à créer un enrichissement sémantique en rendant possible la centralisation et l'intégration complète des informations structurées dans une description unique, tout en veillant à supprimer les redondances. Celle-ci est mise à la disposition d'une communauté d'utilisateurs pour des besoins divers, d'une ou plusieurs applications interdépendantes. De ce fait on désire obtenir :

- Une meilleure cohérence des informations.
- Une réduction des redondances sémantiques et physiques.
- Une modélisation plus fidèle du monde réel.
- Une grande indépendance de données vis-à-vis des programmes de traitement et des supports physiques.
- Une réduction des efforts de saisie et de mise-à-jour de l'information

Une prise de conscience des concepts sous-jacents à l'approche : "Bases de Données" en tant que formalisation effective de traitement de l'information, nous semble aujourd'hui, avoir une importance accrue pour la plupart des applications informatiques. Celle-ci est imposée, d'une part, par la complexité et la diversité des informations à traiter, et d'autre part, par l'hétérogénéité des demandes à satisfaire.

Concevoir une base de données, c'est tout d'abord modéliser une certaine partie du monde réel, en essayant de caractériser la nature des objets manipulés et les événements, qui présentent de l'intérêt vis-à-vis des applications informatiques, que l'on veut mettre en place.

Ainsi, une base de données ne peut pas être considérée comme une simple collection de données reliées ; elle représente, en permanence, l'image aussi fidèle que possible du monde réel, et se trouve ainsi régie par un ensemble de règles spécifiques, qui lui permettent d'accomplir sa fonction.

L'emploi du concept : "Modèle de Données" est le reflet d'un progrès essentiel lors de l'évolution historique vers les bases de données. Pour nous, un modèle de données est un ensemble de concepts destinés à décrire les connaissances, que l'on a du monde réel et ceci de manière indépendante de tout traitement informatique.

Le rôle joué par un modèle de données est fondamental dans la perception correcte du monde réel et dans la description effective des informations perçues. C'est à travers la définition d'un modèle général, que nous pourrons trouver les invariants du monde réel, dont chaque application ne sera qu'un cas particulier.

Ce modèle doit être, avant tout, réaliste : c'est à dire capable de fournir, à tout instant, des renseignements pertinents sur la signification des éléments décrits.

En particulier, il doit répondre à l'évolution des besoins en information et donner à la base les moyens de subir sans à coup les modifications se produisant au stade de son exploitation :

- L'introduction de nouvelles informations et de contraintes d'intégrité nouvelles.
- Le changement de règles de cohérence.

En outre, un bon modèle doit non seulement assurer une certaine finesse dans la description des informations mais, également, dans l'expression des traitements, qui leur seront appliqués.

Les besoins constants d'une modélisation toujours plus fidèle des faits réels a jusqu'à présent, conduit à une croissance du nombre de modèles de données; chacun d'entre eux ayant ses propres concepts, et sa propre terminologie. Cependant, ces modèles ne fournissent pas tous le même niveau sémantique de description de la réalité. [TL 77], [Dat 77], [Nij 77].

Sur le plan informatique, on assiste, depuis une dizaine d'années, au développement de logiciels permettant une meilleure utilisation des bases de données et assurant d'avantage l'intégrité et la sécurité des informations partagées : Les Systèmes de Gestion de Bases de Données, ou encore, appelés : S.G.B.D [FS 76]

Plusieurs architectures ont été proposées, dont les différences résident essentiellement dans la conception du modèle général, qui peut être hiérarchique en réseau ou relationnel.

Parmi les nombreux S.G.B.D. actuellement commercialisés, il est possible de distinguer deux grandes tendances pour la réalisation d'un tel système :

a) La famille de S.G.B.D. hiérarchiques, dont les plus représentatifs sont :

- I.M.S. (Information Management System)
- G.I.S. (General Information System)

On y trouve, également, le Système 2000.

La conception de ces systèmes s'appuie sur un modèle hiérarchique de données, où les informations sont organisées en arbres. L'ensemble des associations du monde réel ne peut se traduire par un tel schéma, sans entraîner des redondances sémantiques et physiques importantes.

b) La famille de S.G.B.D. en réseaux, qui englobe tous les systèmes construits selon les spécifications CODASYL-D.B.T.G. comme :

- I.D.M.S. (Integrated Data Management System)
- I.D.S. (Integrated Data Store)
- D.M.S.100 (Data Management System)
- et le Système TOTAL

Le modèle en réseau a été introduit en 1971, [CSL 71] puis révisé en 1973 par le groupe CODASYL, auquel on doit la définition du langage COBOL. Il fournit des structures, qui peuvent se visualiser à l'aide de graphes, où chaque noeud représente un type d'entité du monde réel^{*}, et chaque arête une liaison entre ces entités. Le schéma produit correspond à la combinaison de plusieurs hiérarchies arborescentes, distinctes, associées par des liens entre hiérarchies.

Il faut signaler, également, dans cette famille de S.G.B.D. le rôle joué par un système comme SOCRATE, conçu à l'Université de Grenoble [Abr 72], dans le développement des modèles réseaux, car les principes sur lesquels il repose sont différents. [SOC 73]

Les modèles hiérarchisés ou en réseau sont, en fait, trop liés à la technique informatique, et en particulier, aux organisations des structures de mémoire. Leur formalisme est plus proche de celui d'un langage de programmation. Ceci est limité à l'interprétation structurelle de données et à la logique de programmes de traitement. Ce sont, en premier lieu, les critères de performances, par le biais de chemins d'accès, qui déterminent la structure. Le schéma ainsi construit, implique le choix des méthodes d'organisation, les formats de données, le regroupement logique de données en segments, et la délimitation de zones physiques de données.

En effet, les modèles hiérarchiques ou en réseau apparaissent d'avantage comme des modèles d'implantation physique des données. Les S.G.B.D. construits autour d'eux sont, de ce fait, bâtis comme des outils de traitement souvent figés dans un ensemble de contraintes d'utilisation répondant à des besoins entièrement spécifiés.

Ainsi la prise en compte de l'évolution du monde réel est malaisé dans le cadre de ces systèmes, où toute modification du schéma obtenu nécessite une adaptation d'importance variable mais rarement indolore pour les usagers. Le choix de la structure est, en fait, primordial pour l'avenir de la base :

* Un enregistrement

une fois, que cette structure est fixée, et la base créée, il est très fastidieux, pour ne pas dire impossible, de la modifier à un coût raisonnable.

Toute introduction de nouveaux éléments, la modification de types d'association et la création de nouveaux chemins d'accès provoquent des changements profonds dans la déclaration de la structure qui se répercutent tous au niveau de la structure physique.

Une évolution s'est alors faite dans la recherche des modèles de données permettant d'affronter ces problèmes. C'est ainsi, que sont apparus les modèles relationnels dont les plus significatifs sont :

- a) Le modèle n-aire de E.F.Codd [Cod 70]
- b) Le modèle Data Semantics de J.R. Abrial [Abr 74]
- c) Le modèle Entité-Relation de P.S Chen [Che 76]

La famille des modèles relationnels se distingue par le compromis réalisé entre les exigences contradictoires de simplifier et de banaliser l'utilisation des systèmes d'information. Celles-ci représentent, à l'heure actuelle, des situations de plus en plus complexes.

Ainsi, le fil directeur de ces travaux était d'arriver à une indépendance de plus en plus grande entre les données et les programmes qui les manipulent, ou les manipuleront.

De plus, leur formalisme représente un effort certain pour autoriser l'introduction dans les modèles de la sémantique, qui leur est attachée, en cherchant, en même temps, à faciliter son maniement.

Basés sur des fondements mathématiques (théories des ensembles, de relations) la plupart de ces modèles proposent au niveau du schéma conceptuel, une représentation simple, qui ne définit que les noms de données et les liaisons logiques entre ces données, à l'exclusion de toute caractéristique liée au chemin d'accès.

Cette qualité est fondamentale dans l'élaboration de schémas conceptuels: elle permet au concepteur de s'affranchir de la définition de chemins d'accès, et de concentrer son attention sur la sémantique de données, contrairement à ce qui se passe dans les modèles hiérarchisés ou en réseau .

De par leur simplicité dans la description logique des structures de données comme des tables de valeurs, les modèles relationnels assurent ainsi une grande indépendance vis-à-vis des représentations physiques de données. Cet aspect avait conduit à trouver un processus de déduction logique, qui compte tenu de l'évolution du monde réel, permet de passer d'un schéma relationnel R à un autre R' et ainsi de suite répondant le mieux aux objectifs de la conception.

De par leur conception même, ils possèdent des notions intéressantes pour décrire entièrement la complexité et le dynamisme du monde réel. Cet aspect important est dû à une adaptabilité à la fois structurelle et logique, qui a permis de prévoir la définition des conditions et des moyens pour prendre en charge des phénomènes externes, non-déterminés auparavant mais susceptibles d'advenir et de modifier les règles d'utilisation précédemment décidées.

La conjonction de ces deux aspects conduit aujourd'hui, à la réalisation de systèmes d'information évolutifs : c'est à dire capables d'être, à tout instant, le reflet exact des situations changeantes, sur lesquelles on pourra définir des nouvelles normes d'exploitation.

En outre, la théorie relationnelle fournit des outils mathématiques (Algèbre relationnelle) dont l'avantage est de pouvoir proposer une approche systématique, qui permet de détecter les anomalies d'une relation. Cette méthode appelée : " Processus de Normalisation " est, en fait, un processus de décomposition de relations, fondé sur la notion de dépendances fonctionnelles. [Cod 71], [Del 73], [Dat 77].

Elle offre une meilleure caractérisation de la sémantique, qui permet de construire un schéma relationnel ayant certaines propriétés. Celles-ci peuvent garantir, lors de l'utilisation de la base, que toute opération de mise-à-jour, de suppression et d'insertion n'ont aucune repercussion sur les autres relations.

Cet avantage a motivé l'investigation de diverses techniques de normalisation, qui a conduit à la définition de plusieurs Formes Normales :

- a) Deuxième et troisième formes normales de E.F. Codd [Cod 71, 72]
- b) Forme normale de Boyce-Codd [Cod 73]
- c) Quatrième forme normale de R. Fagin [Fag 77b]
- d) Faible Boyce-Codd forme normale [PID 78]

Pour une étude complète des Formes Normales, le lecteur peut consulter les articles de [Fag 79a], [Fag 79b] et [Osb 77]

En outre, les travaux de C. Delobel [Del 71], [Del 72], [Del 73] étudient de manière plus approfondie les propriétés de la décomposition fonctionnelle et dégagent des algorithmes de génération automatique de relations en 3FN. Depuis, plusieurs algorithmes de normalisation ont été proposés, parmi eux nous citons à titre d'exemples ceux publiés dans [WW 75], [Bar 76], [BB 77] [Leo 76], [PID 78] et [LT 78]

La notion de dépendances fonctionnelles ne pouvant permettre d'exprimer toute la sémantique du monde réel [BST 75], R. Fagin propose dans [Fag 77b] la notion de dépendances Multivaluées. Une étude complète de leur propriétés mathématiques est donnée dans [Zan 76] et [BFH 77]

C. Delobel a, également, montré dans [Del 73] et [Del 78], que l'on peut rencontrer entre constituants d'autres types de dépendances qu'il appelle : " Dépendance Hiérarchique de Premier et Deuxième Ordre ".

De plus, les dépendances mutuelles sont apparues récemment pour caractériser une méthode de décomposition sans perte d'information [Nic 78a]. Toutefois, elles peuvent être considérées comme un cas spécial des " Joint dependencies " [Ris 78]. Celles-ci permettent en plus de modéliser les différents types de dépendances, à l'exclusion d'une dépendance hiérarchique. Pour le faire, Delobel propose dans [Del 79] le concept de " Embedded joint dependencies "

De gros efforts sont actuellement consacrés à l'implantation de prototypes de S.G.B.D. relationnels, construits, en particulier, autour du modèle n-aire de E.F. Codd. Parmi eux les plus importants sont :

- a) Un Modèle sous MULTICS [MUL 78]
- b) Le Système P.R.T.V. (Peterlee Relational Test Vehicule) [Tod 76]
- c) Le Système INGRES (Interactive Graphic Relational System) [Sto 76]
- d) Le Système R [Ast 76]

Les prototypes relationnels contiennent des caractéristiques intéressantes pour arriver à des S.G.B.D. plus simples à utiliser et où la cohérence des informations est renforcée. Ils offrent des possibilités de non compatibilité avec les S.G.B.D. actuels.

Cependant une représentation relationnelle doit être, enfin, traduite en une structure d'accès. La question soulevée est : " Comment, les promoteurs des prototypes relationnels envisagent-ils de décrire effectivement les chemins d'accès sans pour cela recourir à quelques sortes de structures réseaux ? "

De plus, les autres modèles hiérarchiques ou en réseaux ne sont pas oubliés pour autant, étant donné le nombre de S.G.B.D. actuellement commercialisés.

Face à une telle situation, le besoin d'un modèle de synthèse orienté vers le passage : formalisme relationnel, modèles classiques d'accès est très net.

Cette démarche permet, d'une part, la construction de schémas conceptuels gérés par les S.G.B.D. actuels, à partir de l'étude des propriétés sémantiques intrinsèques aux données pour arriver à une meilleure représentation des faits modélisés.

D'autre part, elle permet de proposer une nouvelle norme de développement des S.G.B.D. Relationnels, basée sur des réalisations existantes.

0. 2 Position de problème traité et objectifs de travail

Le modèle relationnel grâce à l'utilisation des concepts mathématiques de relations et du calcul relationnel représente un modèle de tout premier ordre. Il conduit, aujourd'hui à prévoir la mise en place d'une méthodologie d'analyse et de conception de bases de données, remplaçant le doigté et les expérimentations successives.

De là, l'objectif du présent travail, qui consiste à établir un pont entre le modèle relationnel, comme modèle conceptuel, et le modèle réseau comme modèle d'accès classique, est de pouvoir placer le problème de la conception d'un système d'information automatique dans le cadre d'une démarche logique analytique et rigoureuse.

Le but recherché est double :

a) Il s'agit dans un premier temps d'apporter aux utilisateurs familiers avec la terminologie de S.G.B.D. construits autour d'un modèle réseau la possibilité de se préoccuper de façon rationnelle des problèmes logiques liés à la conception de données, progressivement développées à partir de l'analyse et la mise en oeuvre d'un système d'information. Ces problèmes sont essentiellement d'ordre méthodologique. Leur étude peut se faire en utilisant une approche relationnelle et en définissant des méthodes et des outils algorithmiques permettant aux utilisateurs de s'affranchir des contraintes structurelles, que leur impose la structure réseau. Ces contraintes constituent, jusqu'à présent, une source de blocage à la constitution effective de systèmes d'information, à la fois, réalistes et évolutifs.

b) La construction d'une base de données SOCRATE ou CODASYL peut alors se faire de manière rigoureuse et précise, suivant une démarche rationnelle, à partir d'un modèle mathématiques en utilisant des opérations bien définies. (projection, composition, etc....)

Une telle démarche, doit nous permettre, en particulier, de prendre automatiquement en considération toutes les informations supplémentaires et de là, de nous délivrer des aléas d'une définition exhaustive de l'information au moment de la conception.

Le sujet abordé se subdivise logiquement en deux grandes classes de problèmes :

a) Les problèmes liés à la sémantique de données : élimination des imperfections, processus de normalisation et de génération des relations en 3FN etc....

b) Les problèmes liés à la transformation permettant la création de bases de données efficaces en termes de S.G.B.D. actuels.

La première classe de problèmes a été traitée en détail dans les travaux de [Del 73], [DC 73], [Ber 76], [Leo 76], [BBG 79] etc...

En ce qui nous concerne nous avons essayé plus particulièrement d'aborder les problèmes de la transformation de structures de données, et d'explorer d'avantage les concepts fournis par le modèle n-aire de Codd pour engendrer, à partir d'une vue relationnelle, une structure réseau compatible.

Cependant cette vue relationnelle n'est pas unique ; elle dépend des chemins d'accès que l'on veut favoriser pour aboutir à la structure opérationnelle répondant le mieux à l'évolution des demandes immédiates et anticipées.

Avant de commencer ce travail, il nous a paru intéressant d'étudier les travaux entrepris par de nombreux auteurs abordant les problèmes de la correspondance entre les modèles hiérarchiques ou en réseau et le modèle n-aire de Codd.

L'étude de la possibilité de transformation entre un schéma CODASYL et un modèle relationnel avait été souligné en 1971 par E.F. Codd [Cod 71], sans toutefois préciser les conditions de sa réalisation.

M. Adiba et M. Léonard avaient mis en évidence cette possibilité, et la difficulté dans la réalisation automatique, car la transformation ne peut s'effectuer que par un apport d'informations supplémentaires non contenues dans la description du modèle de départ. [Adi 75], [AL 75]

Dans un article de synthèse [ALD 76] C. Delobel, M. Adiba et M. Léonard ont proposé un processus de transformation des structures hiérarchiques ou en réseau, en une structure relationnelle avec, en tenant compte de dépendances fonctionnelles intrinsèques aux données : l'élimination des imperfections. (Processus de normalisation, passage à 3FN)

Depuis, M. Léonard a développé et réalisé un ensemble de programmes traduisant quasi-automatiquement des structures réseau de type SOCRATE en un ensemble de relations n-aire [Leo 76]

Un projet ENSIMAG [BD 76] a traité le cas général des structures réseau.

Ces travaux ont permis de vérifier les facteurs suivants :

- Une structure SOCRATE ou réseau en général, contient moins d'informations sémantiques que l'ensemble des relations associées. Cette remarque montre l'importance du modèle relationnel dans la représentation effective du monde réel.

- La définition d'une structure quelconque peut provoquer des imperfections au niveau de l'exploitation. Celles-ci peuvent être évitées grâce à une modélisation relationnelle.

Dans [Leo 76], M. Léonard a montré, qu'il suffit pour les éliminer de supprimer les relations redondantes, en particulier celles associées aux chemins d'accès, que l'utilisateur avait introduit pour mieux utiliser sa base.

c) Pour un système donné, le schéma logique de la description des informations peut être réduit à une structure canonique faisant apparaître les éléments fondamentaux (les données élémentaires), ainsi que les liaisons logiques entre ces éléments.

Dans le cadre du présent travail, nous nous plaçons dans une optique légèrement différente : à savoir appliquer les outils conceptuels développés à propos du modèle Multigraphe, sur lesquels nous reviendrons ultérieurement, pour aider, le concepteur à mieux construire une structure réseau dépourvue de redondance à partir d'un ensemble de propriétés sémantiques initiales.

Celles-ci décrivent à l'aide des dépendances fonctionnelles la nature des faits élémentaires et les contraintes notamment fonctionnelles, qui les régissent.

L'étude de cette transformation est indispensable dans le cadre du développement de la conception de S.G.B.D. existants. Elle constitue, à l'heure actuelle, un passage obligé très contraignant, puisque les S.G.B.D. actuels sont loins de pouvoir représenter toute la sémantique et la dynamique du monde réel. En outre, elle offre à l'utilisateur une liberté de raisonnement et de conception nécessaire pour élargir la puissance de systèmes disponibles, en les adaptant à des conditions réelles variant avec le temps. Il sera donc possible d'étendre les services rendus par ces systèmes.

Elle peut servir, également, de base à la création des outils conceptuels de restructuration et de coopération de bases de données.

Une telle démarche soulève les questions suivantes :

- Comment doit-on aborder l'analyse et la conception de données pour construire convenablement une structure réseau en termes des S.G.B.D. actuels ? Cette structure doit tenir compte de toute la sémantique du monde réel telle qu'elle est perçue par la communauté d'utilisateurs, mais ceci de manière à laisser ouverte toute possibilité de prise en compte de nouvelles normes d'exploitation.

- Comment peut on déterminer, à partir d'un ensemble de propriétés sémantiques initiales, les relations et les liaisons que doit contenir implicitement la structure en construction ?

- L'évolution de la base est nécessaire, d'une part, pour faire face aux changements constants de son environnement et d'autre part, pour améliorer, au fur et à mesure la qualité et l'efficacité d'utilisation de celle-ci. Quels sont les paramètres à définir et les opérations de restructuration à entreprendre pour effectuer les modifications du schéma rendues nécessaires, afin de satisfaire en partie aux prévisions de la mise en place d'une structure d'accès performante, et pour continuer à assurer la cohérence globale du schéma conceptuel.

Etant donné la complexité du processus de transformation de structure de données et l'incapacité du concepteur de prendre simultanément en considération un trop grand nombre de paramètres, il nous paraît nécessaire de conduire ce processus par étapes ; nous distinguons trois étapes.

- La première consiste à ordonner l'ensemble de relations de départ en une structure relationnelle, que nous appellerons : "SREL" et dont la définition sera donnée ultérieurement. Le but de cette étape est de faciliter le passage à la structure réseau finale.

- La deuxième consiste à modifier cette structure initiale pour répondre à l'évolution des besoins en information. Elle doit nous conduire à une structure relationnelle plus performante au niveau des accès mais conservant les acquis précédemment obtenus. Nous l'appellerons "SREL". A la fin de cette étape toutes les relations à traduire dans la structure réseau finale sont contenues dans le Multigraphe correspondant à "SREL".

- La dernière étape consiste à obtenir, à partir de cette structure, la structure réseau en termes des S.G.B.D. actuels.

En cherchant à rendre la plus grande partie de cette transformation indépendante des systèmes utilisés, nous nous proposons de décomposer cette dernière étape en deux parties :

a) La première étudie la transformation de "SREL" en une structure réseau générale indépendante du S.G.B.D. utilisé, qui sera appelée "SRES".

b) La deuxième cherche ensuite à transformer "SRES" (la structure réseau précédente), en une structure SOCRATE ou CODASYL.

A chacune de ces étapes nous avons tenté d'associer une formalisation spécifique qui lui est adaptée et un ensemble de paramètres permettant de prendre en considération les contraintes relatives à ce niveau.

Le passage d'un niveau à l'autre doit se faire sans appauvrissement de la sémantique. Aussi la structure réseau finale comportera un minimum de redondance, mais, elle doit permettre un contrôle de cohérence.

Dans toutes les étapes de la démarche, nous travaillons uniquement au niveau des nom, ceci signifie, que nous ne traitons pas les données associées à la structure.

L'implantation informatique du processus de transformation repose sur la définition d'un modèle général de représentation de données, que nous appellerons "Multigraphe".

Le Modèle Multigraphe servira, d'une part, à gérer les informations concernant les structures en cours d'élaboration, et d'autre part, à faciliter l'implantation informatique des algorithmes de transformation de structures de données.

Le chapitre I sera consacré à la représentation des concepts et des fonctions développées à propos, ainsi que la définition des opérations et des procédures réalisées à partir de ces concepts.

Nous avons retenu le modèle n-aire de Codd comme modèle de référence car il nous semble le mieux adapté à la mise en place d'un processus automatique de conception.

En effet, au cours des temps, par suite de l'évolution quant à l'utilisation désirée de l'information, le concepteur peut souhaiter y apporter des compléments, ou effectuer des modifications afin d'assurer la cohérence des informations et pour répondre rapidement aux questions qui seront posées à la base.

Dans ce cas, l'intervention de l'utilisateur est indispensable pour expliciter les ambiguïtés sémantiques qui pourraient survenir à cause de l'introduction des informations supplémentaires.

CHAPITRE I

LE MODELE : " MULTIGRAPHE "

I. 1 Définition de la notion de Multigraphe

Le modèle Multigraphe repose sur la théorie des relations ainsi que sur les notions développées par J.R. Abrial dans Data Semantics [Abr 74]

Depuis, ce modèle de données a été utilisé par de nombreuses personnes dans la représentation d'un schéma conceptuel comme l'on a vu dernièrement chez B. Bastaraud et M. Devy [BD 76] dans le cadre d'un programme de transformation d'une structure de la modélisation SOCRATE, ou réseau en général dans une modélisation relationnelle.

Il se présente comme un ensemble d'outil conceptuels pour la description des informations concernant les schémas élaborés et pour l'expression des opérations qu'il est possible de leur appliquer lors du processus de transformation des structures de données.

Dans ce but, le modèle Multigraphe utilise les notions de catégories d'éléments et de fonctions pour représenter les objets du monde réel et les associations entre ces objets.

Or, le monde réel va nécessairement évoluer. A ce niveau il est possible d'appliquer les notions d'opérations sur les catégories et les fonctions pour effectuer les modifications nécessaires apportées aux schémas déjà construits, afin de prendre en compte cet aspect dynamique.

Grâce à ses caractéristiques et sa généralité il nous a permis de traiter les problèmes suivants :

a) Le problème de l'homogénéisation des représentations informatiques des structures de données en cours d'élaboration, que celles-ci soient décrites à l'aide d'un formalisme relationnel ou à l'aide d'un modèle réseau de données.

b) Le problème de traitement des informations relatives aux sous-transformations, ce qui permet d'avoir un cadre général dans lequel se développe facilement l'implantation informatique des algorithmes de transformation.

c) Le problème de l'adaptabilité à l'évolution des structures de données construites. Ce point est particulièrement important dans une approche ascendante de conception. Il facilite, compte tenu de l'évolution de besoins en information, la définition d'un autre schéma relationnel, à partir de celui de départ et ainsi de suite jusqu'à l'obtention d'une représentation conforme à la réalité et aux vœux des usagers.

De part sa construction même, le modèle Multigraphe nous a permis d'implanter facilement d'autres procédures et de modifier celles déjà existantes pour les adapter aux besoins de notre démarche.

L'architecture d'un tel système repose sur l'implantation d'un ensemble d'hypergraphes multiples: Un hypergraphe H est défini par le couple : $H = \{X, \mathcal{E}\}$ où $X = \{x_1, x_2, \dots, x_n\}$ est un ensemble fini et $\mathcal{E} = (E_i | i \in I)$ une famille de parties de X tel que $E_i \neq \emptyset (i \in I)$ et $\bigcup_{i \in I} E_i = X$

Les éléments (x_1, x_2, \dots, x_n) de X sont les sommets de l'hypergraphe, que l'on appelle noeuds et les ensembles \mathcal{E} , que l'on note (E_1, E_2, \dots, E_n) sont les arêtes.

Entre un modèle de données et un multigraphe nous pouvons établir la correspondance élémentaire suivante :

- A chaque noeud du multigraphe nous faisons correspondre un bloc de constituants caractérisant une entité du monde réel. Un sous-ensemble de ces constituants forme la clé du noeud.

Chaque arête correspondra à une liaison entre ces noeuds. Celle-ci est orientée d'un noeud de départ vers un noeud d'arrivée et sa signification correspond à celle d'une association entre deux blocs de constituants. Elle peut avoir un type pour exprimer la nature du lien qu'elle représente. Cette notion de type sera définie ultérieurement.

I. 2 Représentation d'un Multigraphe : les fonctions , les opérations et les procédures associées.

La modélisation informatique d'un Multigraphe, représentant une structure de données quelconque, comprend quatre catégories d'éléments :

NOEUD, ARETE, CONSTITUANT et CLE

et les fonctions d'accès suivantes :

Facc 1 : NOEUD $\frac{\text{départ}}{\text{noeud départ}}$ ARETE départ (n) = ensemble des arêtes qui partent du noeud n
noeud départ (a) = noeud d'où part l'arête a

Facc 2 : NOEUD $\frac{\text{arrivée}}{\text{noeud d'arrivée}}$ ARETE arrivée (n) = ensemble des arêtes qui arrivent au noeud n
noeud d'arrivée (a) = noeud auquel arrive l'arête a

Facc 3 : NOEUD $\frac{\text{composé}}{\text{utilisé}}$ CONSTITUANT composé (n) = liste des constituants d'un noeud
utilisé (c) = noeud auquel est attaché le constituant c

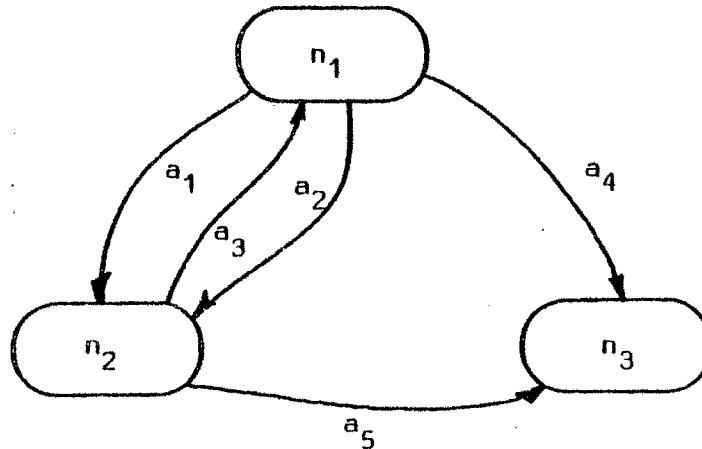
Facc 4 : NOEUD $\frac{\text{clé d'un noeud}}{\dots\dots\dots}$ CLE

Facc 5 et Facc 6 étant inverse l'une de l'autre, on peut simplement écrire :

$$\text{CONSTITUANT} \frac{\text{Facc 5}}{\text{Facc 6}} \text{CLE}$$

Exemple : Pour donner la signification d'une fonction d'accès nous considérons que si $k \in \text{CLE}$ alors $\text{Facc 6}(k)$ est l'ensemble des constituants qui figurent dans la clé k .

La représentation d'un Multigraphe peut être illustrée à l'aide du schéma suivant :



I. 2.1 Implantation informatique des catégories

A chaque catégorie correspond une structure arborescente à deux niveaux, le premier relatif au nom de la catégorie, le second à ses propriétés. Celles-ci sont d'origine :

a) sémantique :

1) La table : NOEUD

NNI : numéro interne formé automatiquement par le système pour le discriminer d'autres noeuds

NNOM : nom symbolique tel qu'il est défini par l'utilisateur

NNBCONST : nombre de constituants d'un noeud

NNBCCLE : nombre de constituants formant la clé

NNBADEP : nombre d'arêtes de départ

NNBAARR : nombre d'arêtes d'arrivée

2) La table : ARETE

ANI : numéro interne désigné par le système

ANOM : nom symbolique

ATYPE : type de l'arête (qui peut être l'un des caractères : " F ", " H " ou " B " et dont la signification sera donnée ultérieurement)

3) la table : CONST

CNI : numéro interne unique (comme dans la table de constituan
CNOM : nom symbolique
CEXT : type de const (valant "1" dans le cas d'un constituant
clé)

b) système : provenant de la méthode d'implantation choisie pour les
fonctions d'accès

1) NOEUD -

NTCONST : tête de liste NOEUD $\frac{\text{Facc } 3}{\dots\dots}$ CONSTITUANT

NTARDEP : tête de liste NOEUD $\frac{\text{Facc } 1}{\dots\dots}$ ARETE

2) ARETE -

ANDEP : numéro interne du noeud de départ

ANARR : numéro interne du noeud d'arrivée

ASUIV : numéro suivant dans la liste NOEUD $\frac{\text{Facc } 1}{\dots\dots}$ ARETE

3) CONSTITUANT -

CINF : si le constituant appartient réellement au noeud

CACLE : chaîne de bit CONCT → CLE

CSUIV : numéro suivant dans la liste NOEUD $\frac{\text{Facc } 3}{\dots\dots}$ CONST.

I. 2.2 Opérations définies sur les catégories.

- 1) Création d'un élément dans une catégorie
- 2) Suppression d'un élément dans une catégorie
- 3) Enumération de tous les éléments d'une catégorie
- 4) Regarder si un élément donné appartient à une catégorie donnée

I. 2.3 Procédures associées aux opérations définies sur les catégories.

a) Les fonctions de création

a.1) NCREER (P₁, P₂, P₃, CØ)
=====

Créer le noeud de numéro interne P₃ dans le graphe P₁ et
lui associer le nom symbolique P₂. (CØ = code de message
d'erreur) Un noeud est créé dans l'état actif et si l'uti-
lisateur ne veut pas associer un nom symbolique au noeud
qu'il créé, alors P₂ = "∇" ;

a.2) ACREER ($P_1, P_2, P_3, P_4, P_5, P_6, C\emptyset$)
=====

Créer une arête de numéro interne P_4 du noeud de départ P_3 au noeud d'arrivée P_5 sur le graphe P_1 , lui associer un nom symbolique P_2 et un type P_6 . Une arête est toujours créée dans l'état actif. Au moment de la création P_2 peut avoir la valeur " ∇ "

a.3) CCREER ($P_1, P_2, P_3, P_4, C\emptyset$)
=====

Créer un constituant de numéro interne P_3 associé au noeud P_2 du graphe P_1 . Un constituant est créé dans l'état non virtuel * l'utilisateur peut lui associer un type P_4 .

a.4) CRCLE ($P_1, P_2, P_3, P_4, C\emptyset$)
=====

Créer une clé associée au noeud P_2 du graphe P_1 . La clé est formée à partir des constituants figurant dans la table P_3 de numéros internes de constituants. P_4 donne le nombre d'éléments significatifs.

b) Fonction de changement de l'état ou de type d'un élément

b.1) SUPNOEU ($P_1, P_2, C\emptyset$)
=====

Rendre inactif le noeud de numéro interne P_2 du graphe P_1 .
Remarque : Cette fonction entraîne obligatoirement la désactivation des arêtes partant de ce noeud et de celles arrivant à ce noeud ainsi que le passage à l'état virtuel des constituants qui y sont attachés.

b.2) SUPARET ($P_1, P_2, P_3, C\emptyset$)
=====

Rendre inactif l'arête de numéro interne P_3 partant de noeud P_2 du graphe P_1 .

b.3) VIRCONS ($P_1, P_2, P_3, C\emptyset$)
=====

Faire passer le constituant de numéro interne P_3 du noeud P_2 du graphe P_1 de l'état non virtuel à l'état virtuel. Cette fonction associe la valeur "1" à CINF, qui autrement a la valeur "0"

* Un constituant est dit dans l'état virtuel s'il appartient réellement à aucun noeud du Multigraphe.

b.4) CREPL ($P_1, P_2, P_3, P_4, C\emptyset$)
=====

Modifier le type d'un constituant de numéro interne P_3 attaché au noeud P_2 du graphe P_1 . Lui associé le type P_4 .

c) Fonctions d'énumération des éléments d'une catégorie.

c.1) LSTNOEU ($P_1, P_2, P_3, C\emptyset$)
=====

Ranger dans le tableau P_2 les numéros internes des noeuds du graphe P_1 . Donner dans P_3 le nombre total de noeuds actifs.

c.2) LSTARET ($P_1, P_2, P_3, P_4, P_5, P_6, P_7, C\emptyset$)
=====

Pour un noeud de numéro interne P_2 du graphe P_1 fournir dans $P_3(i)$ le numéro interne d'une de ces arêtes partant dans $P_4(i)$ le numéro interne de son noeud d'arrivée dans $P_5(i)$ le nom symbolique de l'arête dans $P_6(i)$ son type. P_7 donne le nombre total des arêtes partant du noeud P_2 .

c.3) LSTCONS ($P_1, P_2, P_3, P_4, P_5, P_6, P_7, C\emptyset$)
=====

Pour un noeud de numéro interne P_2 et de nom P_3 associé au graphe P_1 donner dans $P_4(i)$ le numéro interne d'un de ces constituants, dans $P_5(i)$ l'état de ce constituant, dans $P_6(i)$ son type. P_7 donne le nombre total de constituants attachés au noeud P_2 .

d) Fonctions de recherche d'un élément d'une catégorie

d.1) HIERARC ($P_1, P_2, P_3, P_4, P_5, C\emptyset$)
=====

Chercher une arête de type P_5 partant du noeud P_2 du graphe P_1 . Rendre dans P_3 le numéro interne de l'arête dans P_4 le numéro interne de son noeud d'arrivée.

d.2) CHN ($P_1, P_2, P_3, C\emptyset$)
=====

Retrouver dans P_3 l'emplacement physique (indice sur la table NTAB) correspondant à un noeud de numéro interne P_2 associé au Multigraphe P_1

d.3) CHCONST (P₁, P₂, P₃, P₄, CØ)
=====

Rechercher dans la liste de constituants attachés au noeud P₂ du Multigraphe P₁ un constituant de numéro interne P₃. Si l'élément est retrouvé, retourner dans P₄ l'indice du constituant qui lui précède dans le chaînage sinon retourner avec P₄ = "0"

d.4) CHA (P₁, P₂, P₃, P₄, CØ)
=====

Rechercher dans la liste d'arêtes partant du noeud P₂ du graphe P₁ une arête de numéro interne P₃. Si l'arête existe alors retourner dans P₄ l'indice sur l'arête qui précède dans le chaînage, sinon retourner P₄ = "0"

e) Fonctions diverses

e.1) TRI (P₁, P₂, P₃, CØ)
=====

Ranger dans le tableau P₂ les numéros internes des noeuds actifs associés au graphe P₁ suivant le préordre indiqué :

Soit N_i et N_j deux noeuds actifs de clés respectives K(N_i) et K(N_j) alors

$$N_i < N_j \iff \text{cardinal}(K(N_i)) = \text{cardinal}(K(N_j)) + 1$$

Fournir dans P₃(i) les constituants clé du noeud de rang i.

Remarques :

a) Chaque fonction renvoie un code d'erreur renseignant l'utilisateur sur l'exécution de son appel.

b) Un noeud est accédé par le nom du graphe auquel il appartient et à l'intérieur de ce graphe par son numéro interne. Le système assurera donc, qu'il ne peut exister dans un graphe donné deux noeuds de même numéro interne.

c) Un constituant est seulement accessible par le numéro interne du noeud auquel il est associé, et à l'intérieur du noeud par son numéro interne. Le système garantira qu'il ne peut exister deux constituants de même numéro interne associés à un même noeud d'un graphe.

d) Une arête est accessible par le numéro interne de son noeud de départ et par son numéro interne unique par rapport à ce noeud.

I. 2.4 Implantation informatique des fonctions d'accès.

NOEUD	<u>Départ</u>	ARETE	:	structure de liste de tête NTARDEP et de pointeur de chaînage ASUIV
NOEUD	<u>Composé</u> <u>Utilisé</u>	CONSTITUANT	:	structure de liste de tête NTCONST et de pointeur de chaînage CSUIV
NOEUD noeud départ	ARETE	:	pointeur ANDEP
NOEUD noeud d'arrivée	ARETE	:	pointeur ANARR
CONSTITUANT	<u>Facc 5</u> <u>Facc 6</u>	CLE	:	chaîne de bit CCLE interprétation des chaînes précédentes

Remarque : La relation NOEUD → CLE se déduit des relations NOEUD → CONST
et de CONST → CLE

I. 3 Informations spécifiques au système informatique.

a) liées aux catégories :

a.1) NTAB : table de correspondance entre le numéro interne d'un
noeud et l'emplacement de ce noeud dans la table NOEUD
de colonnes respectives :

NTNI : no interne de noeud

NPTR : pointeur sur la table NOEUD

a.2) ATAB : table de correspondance entre le numéro interne d'une
arête et son emplacement physique dans la table ARETE,
de colonnes respectives :

ATNI : numéro interne d'arête

APTR : pointeur sur la table ARETE

Un noeud (resp. arête) inactif figurera dans la table NOEUD (resp.
ARETE) mais ne sera pointé par aucun élément de NTAB (resp. ATAB). Comme
la recherche d'un noeud (arête) est effectué sur NTAB, (ATAB) le noeud (l'arête)
sera dans ce cas inaccessible.

b) liées au graphe :

GRAPHE : Nom de graphe
NBNOEUD : nombre de noeuds actifs
NBNMAX : nombre de noeuds effectivement créés
NBARETE : nombre d'arêtes actives
NBAMAX : nombre d'arêtes effectivement créés
NBCONST : nombre de constituants associés au graphe.

I. 4 Procédures spécifiques au système.

1) INIGRAP (P₁, P₂, P₃, CØ)
=====

Créer un nouveau graphe de nom P₂ à partir d'un fichier P₁ ou
rechercher un graphe P₂ sur mémoire secondaire. P₃ est un indicateur.

2) SAVGRAP (P₁, P₂, CØ)
=====

Assurer la sauvegarde d'un graphe P₂ sur un fichier P₁.

3) EDTGRAP (P₁, P₂, CØ)
=====

Créer un fichier P₂ d'édition du graphe à partir d'un fichier P₁.

Le système pouvant prendre en compte plusieurs graphes, il tient à
jour les informations suivantes :

NBG : nombre de graphes déclarés

NBE : nombre de graphes effectivement initialisés.

La procédure NBGRAP permet de déclarer au système le nombre de graphes
que l'on veut gérer, le nombre maximum de ceux-ci étant limité à quatre.

Chacun peut comporter au maximum 100 Noeuds, 300 Arêtes et 500 Constituants.

I. 5 D'autres Procédures spécifiques à la transformation.

1) RECNS : A partir du numéro interne d'un noeud donner son nom
symbolique.

2) RLIST : Obtenir la liste de tous les noeuds.

3) RECNI : A partir du nom symbolique d'un noeud trouver son numéro
interne.

4) RECCNI : A partir du nom symbolique d'un constituant et du numéro
interne d'un noeud, rechercher le numéro interne du
constituant.

5) NEWCO : Créer un nouveau constituant à partir du numéro interne
d'un noeud.

6) MODSYN : Modifier le numéro interne d'un constituant dans un autre.

CHAPITRE II

LES PRINCIPALES ETAPES ET L'OBJECTIF DU PROCESSUS DE CONCEPTION

II. 1 La structure de l'information et les besoins en information.

Le monde réel représente un système immense, qui est perçu à travers l'ensemble des applications informatiques que l'on cherche à mettre en place. On peut les classer en deux parties :

- Les demandes précises et directement exploitable par l'utilisateur.
- Les demandes potentielles, non formalisées et qui ne seront prises en compte qu'ultérieurement, mais dont les besoins en informations ont été recensés.

Les applications sont segmentées en classes de besoin, chaque classe correspond à la vision spécifique d'un utilisateur ou d'un ensemble d'utilisateurs effectuant des activités semblables ou interchangeable. Celle-ci forme une perception particulière d'une partie du monde réel, qu'il faudra d'une part analyser en tant que telle, et d'autre part intégrer avec d'autres perceptions chaque fois que celles-ci porteront sur un même réel.

La phase : " Analyse de l'environnement réel " est l'aboutissement d'une longue démarche, qui doit conduire à la construction d'une description formelle des situations modélisées. Le but sera d'établir une liste aussi complète que possible des données utilisées en fonction des problèmes que doit résoudre l'existence de la base en construction.

Remarquons que les utilisateurs n'ont pas tous la même vue des données. Celle-ci diffère non seulement d'un individu à l'autre, mais encore, chez le même individu selon les moments. Un utilisateur n'est également concerné que par un sous-ensemble de ces données.

Les informations recueillies lors de cette activité représentent une première tentative de regroupement de données en classe d'objets de même nature.

Par exemple, dans une gestion de personnel nous pourrions voir les classes : " employé ", " département ", " poste de travail " etc....

Chaque classe possède des caractéristiques qui vont permettre de décrire les objets associés d'une manière plus fine. On parlera donc de constituants. Parmi ces constituants, il faut caractériser d'une part, ceux qui vont servir à identifier les objets, les uns par rapport aux autres, et d'autre part, ceux qui vont donner des informations supplémentaires.

C'est ainsi qu'une classe : " employé " pourra être décrite par un no-insee, un nom, une adresse, un salaire, etc....

A l'intérieur toutefois de cette classe d'objets, un employé donné pourra être identifié par son no-insee ou par son nom.

Chaque constituant représente en lui même une classe d'objets à laquelle correspond un ensemble de valeurs. De même un constituant peut avoir plusieurs caractéristiques, par exemple : " rue ", " ville " et " département " peuvent être les caractéristiques du constituant " adresse ".

Il est fondamental de décider quelles seront les classes d'objets qui existeront comme des entités indépendantes ou encore, si une classe d'objets est une simple caractéristique d'une autre classe (entité).

Ces données par elles-mêmes ne sont porteuses d'aucune information. Elles deviennent des informations utilisables, lorsqu' une association a été construite entre ces classes d'objets.

Ainsi pour chaque classe d'objets, il faut considérer la nature des liens sémantiques qu'elle peut avoir avec une ou plusieurs autres.

Par exemple :

" Un employé travaille dans un département "

" Une commande correspond à un client mais concerne plusieurs produits "

....

Une association entre deux classes d'objets peut appartenir à l'une des catégories suivantes :

a) association de type 1 : 1

" A un élément de la première classe, il ne correspond qu'un seul élément de la deuxième classe et inversement ".

b) association de type 1 : n

" A un objet de la première classe, il correspond n-objets de la seconde classe, et à un objet de la seconde classe il ne correspond qu'un objet de la première ". Dans ce cas on parlera d'associations fonctionnelles.

c) association de type n : n

" A un objet de la première classe il correspond n-objets de la seconde et inversement ".

Comme une classe d'objets peut être soit une entité soit un constituant, ceci amène à envisager deux sortes de liaisons.

1) Les liaisons entre une entité et un constituant.

2) Les liaisons entre entités. Celles ci peuvent avoir leurs propres constituants.

Enfin, il faut indiquer un certain nombre de règles sémantiques ou de règles de cohérence que devront vérifier les objets. Par exemple :

" L'âge d'un employé doit être compris entre 18 et 65 ".

" Un employé a toujours un sexe déterminé et une date de naissance ".

Il est évident, que la description des faits réels ainsi obtenue ne peut être complète. Elle contient éventuellement des redondances et des contradictions.

Cependant elle constitue une base de départ, à partir de laquelle, il sera possible d'effectuer des recherches supplémentaires auprès des utilisateurs, en vue de cerner le mieux possible leur vues partielles du monde réel, avant de recommencer une analyse plus approfondie pour en contrôler la cohérence et voir les redondances.

Au cours de cette activité, les gestionnaires devront s'entendre sur une désignation et une description unique des données élémentaires et des contraintes sémantiques qui existent entre elles, activité qui ne doit en aucun cas être sous estimée.

Les résultats obtenus conduiront à établir une vue intégrée des éléments ayant chacun un intérêt pour le système de gestion, et dont la connaissance est nécessaire pour réaliser un système d'information utile et efficace.

Toutefois, l'ensemble de ces éléments ne représente pas nécessairement un ensemble non-modifiable de données. Bien au contraire, ces composants sont appelés à être modifiés avec le temps pour permettre à la base de continuer à être l'expression la plus fidèle de la réalité.

Seule une conception progressive, à l'aide d'une approche relationnelle de modélisation, permettra de rendre aisée l'évolution de la base.

Cette évolution se traduit lors de l'apparition de nouveaux besoins par une saisie satisfaisante des informations supplémentaires et leur intégration facile avec le schéma précédemment décidé, sans pour cela remettre en cause tout ce schéma.

Remarque : Dans la suite de ce travail, le lecteur peut se reporter à [Dat 77], [IL 77] et [Del 73] pour avoir une description complète du modèle relationnel. En ce qui nous concerne, nous retiendrons seulement les éléments dont nous avons besoin. Certains des concepts seront rappelés au moment de leurs utilisations.

II. 2 La conception du schéma conceptuel : L'approche Relationnelle.

Dans l'approche relationnelle de conception associée au modèle n-aire de Codd, le monde réel s'exprime par la conjonction d'une liste $C = UC^1$ de constituants et d'un ensemble de relations $R = \{R_1(C^1), R_2(C^2), \dots, R_n(C^n)\}$ R_i dénote le nom de la relation et C^i l'ensemble de constituants sur lesquels la relation est construite.

Chaque constituant est caractérisé par un nom et un domaine ; pour chaque domaine, le type de valeurs associées et une éventuelle restriction doivent être explicités.

Les constituants appartenant à une relation sont reliés par un même prédicat. Il existe entre eux des liaisons. En particulier, celles que l'on peut déduire des clés présentent un intérêt particulier dans un processus de regroupement de données. Ces liaisons qualifiées de liaisons intra-relation sont implicites, puisque la propriété disparaît lorsque la relation est supprimée.

Les relations sont également reliées par des liens sémantiques différents appelés : liaisons inter-relations. Celles-ci doivent être découvertes par les personnes familières à l'élaboration du système d'information.

On ne saurait définir un schéma relationnel sans définir les contraintes d'intégrité que devront vérifier les données et qui peuvent être :

a) des contraintes liées aux valeurs d'une donnée, par exemple :

$$18 \leq \text{âge d'employé} \leq 65$$

$$\text{Salaire net} = \text{Salaire brut} - \{\text{Impôts} + \text{Charges Sociales}\}$$

b) des contraintes liées à la structure de l'information et la façon dont apparaissent les données.

Par exemple :

" Un employé ne peut travailler que dans un département ", ce qui traduit une association de type n:1 entre " employé " et " département ".

Dans la pratique, il nous semble, que s'il est naturel et simple pour un utilisateur de définir ses besoins en termes de données élémentaires et de règles d'intégrité à conserver, il est en revanche beaucoup plus délicat de les formuler en termes de relations. Ceci est dû au fait que la théorie relationnelle ne fournit aucune indication sur la façon de choisir les relations qu'il faudra prendre en compte pour aboutir à une représentation arbitraire de la sémantique de données.

En effet, un des problèmes du modèle conceptuel est justement de trouver un regroupement des informations en un schéma relationnel comportant un ensemble minimal de relations, de telle sorte que les mises-à-jour et les règles d'intégrité concernant aussi bien les relations traduites, que celles que l'on peut déduire d'elles par projection ou composition, ne causent aucune imperfection intra ou inter-relations : [Leo 76]

- incohérence d'informations.
- règles d'intégrité non assurées.
- redondances d'informations.

Ainsi, la méthodologie de conception que nous allons adopter, suit une démarche ascendante pour construire le schéma conceptuel de départ. Partant des faits élémentaires les utilisateurs sont habitués à manipuler les constituants et connaissent bien les contraintes notamment fonctionnelles qui les régissent. C'est pourquoi, une fois obtenues, à partir de toutes les règles de gestion, la liste des constituants et leurs liaisons sémantiques, ces dernières pourront être utilisées comme concept de base pour produire automatiquement un ensemble minimal de relations. Ceci contiendra éventuellement toutes les relations qu'il faut et qu'il suffit de traduire dans un schéma relationnel pour garantir à l'utilisateur un découpage de l'information, qui diminue le nombre d'anomalies, lors des opérations de mise-à-jour, de suppression et d'insertion de données.

La notion " de dépendance fonctionnelle ", introduite par E.F. Codd en 1971 [Cod 71] permet d'exprimer d'une manière précise les associations intra-et inter-relations, ainsi que certaines propriétés sémantiques.

En effet, les dépendances fonctionnelles forment une partie importante de la sémantique d'une relation facile à formaliser en des termes mathématiques précis. C'est la raison pour laquelle plusieurs travaux leur ont été dévoués.

Pour une étude exhaustive de leurs propriétés le lecteur peut se reporter aux travaux de [Arm 74], [Ber 76], [Del 73], [Cod 73], [BB 77]. Nous nous limiterons ici à une présentation informelle de ce concept.

" Une relation fonctionnelle indique que la connaissance de la valeur d'un constituant implique la connaissance de la valeur d'un autre constituant ".

Dans une application, les relations fonctionnelles peuvent être considérées comme une propriété intrinsèque de l'information. Cette propriété existe soit par conception soit comme résultat de la façon dont apparaissent les données.

Par exemple, le fait de dire " qu'un employé ne peut travailler que dans un département ", peut s'exprimer de la manière suivante :

" La connaissance du nom d'un employé entraîne la connaissance de son numéro de département ".

Cette relation fonctionnelle sera notée symboliquement par l'écriture :

rf : nom employé \rightarrow no de département

Dans la pratique, les liaisons qui existent entre constituants ne sont pas toutes de nature fonctionnelle. On rencontre souvent des liaisons non fonctionnelles, où à une valeur d'un constituant donné, il correspond plusieurs valeurs d'un autre constituant et inversement.

Par exemple, considérons une association entre " employé et " diplôme ' qui traduit la contrainte suivante :

" A un employé donné il correspond une liste parfaitement définie de diplômes et inversement" .

donc :

rf_1 : nom employé \rightarrow nom diplôme
 rf_2 : nom diplôme \rightarrow nom employé

Cependant, toute association de ce même type peut être représentée au moyen d'une relation fonctionnelle, en adjoignant aux constituants concernés par l'association un constituant fictif θ , qui servira de compteur de nombre d'occurrence d'un cosntituant par rapport à un occurrence d'un autre.

Dans le cas de notre exemple, nous obtiendrons la relation fonctionnelle notée :

rf : nom employé, nom diplôme $\rightarrow \theta$

Le constituant fictif θ est un constituant unique de la relation rf , qui ne peut apparaître dans une autre relation fonctionnelle. Dans le cas, où il existe plusieurs associations non-fonctionnelles entre deux mêmes constituants, chaque relation fonctionnelle représentant l'une d'entre elles aura son propre constituant fictif afin de préserver le caractère unique du nom.

Cette transformation de liaisons non-fonctionnelles en des relations fonctionnelles nous permettra d'avoir une représentation informatique homogène de toutes les propriétés de notre modèle.

En effet, les relations fonctionnelles présentent un intérêt dans la mesure où elles ont déjà fait l'objet de plusieurs études, qui ont permis de mettre en évidence certaines propriétés mathématiques.[Arm 74], [DC 73], [Ber 76].

Dans la perspective d'une conception automatique de schéma relationnels, le fait de savoir, que ces propriétés peuvent être formulées en termes de l'Algèbre de Boole [DC 73] et [Mil 65] va permettre au concepteur de les explorer d'avantage dans la structuration de l'information, dans la mesure où les calculs booliens sont très simplement effectués par programme.

De ce fait, les relations fonctionnelles sont significatives, à la fois, pour les utilisateurs dans la compréhension des propriétés intrinsèques des données et pour le concepteur dans la construction effective du schéma conceptuel. Celui-ci peut faire des raisonnements purement formels sur ces dépendances, sans avoir besoin de se souvenir de la sémantique qui y est

attachée et qui a conduit à leur établissement. En outre elles lui donnent les moyens de recherche automatique des clés associés à une relation, ce qui, autrement, n'est pas toujours facile. [BB 77], [YJ 76], [LD 75] et [Leo 76].

De plus, elles permettent d'avoir une indépendance de plus en plus grande entre les données et les programmes de manipulation, puisque les traitements seront modifiés si et seulement si l'ensemble de dépendances fonctionnelles concerné a changé. Cependant, ce changement est au niveau sémantique et donc ne peut pas être évité.

II. 3 Le point de départ de la démarche de transformation.

En ce qui nous concerne, la démarche proposée commence lorsqu'on a pu obtenir un ensemble de dépendances fonctionnelles structurant les constituants rassemblés par l'utilisateur dans la phase : " Structure de l'information ". Celles-ci sont exprimées sous formes de relations fonctionnelles.

L'ensemble " Dep " de dépendances fonctionnelles de départ n'est pas nécessairement donné en extension. Par contre, il peut comporter un certain nombre de redondances qu'il faudra ensuite éliminer.

De toute façon, le concepteur n'a aucun contrôle sur cette structure il ne peut faire que des constatations et à partir de ces constatations proposer un processus de regroupement naturel des informations adapté aux différents traitements. Ceci peut consister à appliquer à l'ensemble " Dep " de relations fonctionnelles données comme informations extérieures, un algorithme dit de synthèse, par exemple celui proposé dans [BB 76] ou dans [Leo 76], qui à l'aide des propriétés mathématiques (transitivité, pseudotransitivité, etc..) produit automatiquement la fermeture et la couverture minimale de " Dep ".

Remarque : Les lecteurs non familiers avec l'approche de synthèse peuvent se référer à [De1 73], [WW 75], [Ber 76], [Leo 76], [BB 76] et [PiD 78].

L'obtention de la fermeture de l'ensemble " Dep " revient à la recherche d'une base complète de la forme relationnelle associée à l'ensemble C de constituants qui figurent dans la base.

Ceci permet au concepteur de fournir au niveau de l'implémentation logique de la base tous les chemins d'accès à une information donnée.

Trouver la couverture minimale de " Dep " revient à la suppression des relations redondantes. Or, à cause de la double signification des relations fonctionnelles, qui sont considérées, à la fois, comme des relations et des règles d'intégrité, l'obtention de la couverture minimale servira à éliminer les règles d'intégrité redondantes et à trouver un ensemble minimal de

règles dont il suffit d'assurer la cohérence pour obtenir la cohérence globale de la base de données en construction.

Le schéma relationnel ainsi produit, ne contiendra aucune information superflue, puisque chaque relation fonctionnelle, qui est donnée comme information extérieure, et soit contenue dans le schéma, soit déduite d'autres relations fonctionnelles par projection.

Dans le cadre du développement d'une méthodologie et d'un système de transformation effective des structures d'une base, la démarche adoptée pour construire le schéma de départ présuppose le stockage en machine d'un catalogue de constituants et d'une liste de relations fonctionnelles appartenant à la couverture minimale de l'ensemble " DEP ".

La construction de ces catalogues de départ peut être réalisée en machine de la manière suivante :

II.3.1 Etablissement du catalogue de constituants.

Les constituants choisis par l'utilisateur dans la phase " Structure de l'information " doivent être désignés d'une manière précise. Ils seront ensuite introduits dans le système automatisé et stockés dans un catalogue.

Chaque constituant est défini par le couple : nom et domaine. Le nom doit être suffisamment précis pour qu'il n'y ait pas d'ambiguïté au sujet de l'information désignée.

Le système affectera à chaque constituant un numéro interne unique pour le discriminer. Ce numéro servira à établir un accès facile au constituant concerné. Lors de l'introduction d'un nouveau constituant dans le catalogue, le système vérifie si son nom figure déjà dans le catalogue avant de lui affecter un numéro interne. Dans le cas où ceci se produirait, le système enverrait un message d'erreur indiquant à l'utilisateur le constituant refusé. L'utilisateur de son côté est amené à intervenir pour proposer un autre nom, ou bien pour introduire le constituant suivant.

À la fin de cette démarche toutes les polysémies sont reconnues automatiquement et rejetées.

Quant aux synonymes, ils ne peuvent pas être détectés de la même manière. Pour le système " nom de client " et " no-de-client " sont totalement différents. Cependant le système doit permettre au concepteur d'obtenir à la fin de cette étape une liste complète des constituants définis.

Le concepteur peut à l'aide de cette liste vérifier manuellement l'existence de synonymes et les indiquer au système. Celui-ci les considère comme équivalents et change leurs numéros internes en affectant à tous le même

numéro. Les noms symboliques pourtant restent intouchés, pour permettre aux utilisateurs de les référencer ultérieurement. Le système assurera au moyen d'une procédure spécifique la recherche des numéros internes correspondant aux noms symboliques référencés pour réaliser les accès à ces informations.

Un domaine représente l'ensemble des valeurs prises par un constituant. Ceci peut être défini en compréhension par le type de caractères utilisés. Le système peut admettre trois types :

- a) type " MOT " : chaîne alpha numérique dont la longueur max. est donnée en paramètre
- b) type " NUM " : valeur numérique entière définie entre une borne inférieure et une borne supérieure
- c) type " LVAL " : liste de valeurs prise dans un domaine prédéfini en extension par une référence explicite.

II. 3.2 Représentation en machine de la liste de relations fonctionnelles.

Les relations fonctionnelles canoniques ^{*} appartenant à la couverture minimale de l'ensemble " DEP " de départ et ayant une même partie gauche sont regroupées en une seule relation. Chaque relation est désignée par un nom et un numéro interne unique. La représentation en machine de celle-ci est composée de deux chaînes binaires de longueur 2 n (si n est la dimension de l'ensemble C des constituants). A chaque position binaire correspond un numéro interne d'un constituant. Chaque élément d'une telle liste aura la forme suivante :

		IMPG					NBD		IMPD					NBD	
Nom symbolique	No interne	0	1	0	0	0	1	1	0	0	1	1	3
de relation	de relation														

Si le $i^{\text{ème}}$ bit de la chaîne IMPG vaut " 1 " le $i^{\text{ème}}$ élément de la liste de constituants donne le numéro interne du constituant appartenant à la clé de la relation fonctionnelle concernée, NBD donne le nombre de constituants formant la clé.

Si le $i^{\text{ème}}$ bit de la chaîne IMPD vaut " 1 " le $i^{\text{ème}}$ élément du catalogue de constituants donne le numéro interne d'un des constituants atteint grâce à la clé, NBD donne leur nombre.

^x Une relation fonctionnelle est dite canonique [Ber 77] si sa partie droite est formée d'un simple constituant élémentaire

II. 4 Exemple 1

Nous allons illustrer cette démarche à l'aide d'un exemple qui envisage le sous-ensemble d'une entreprise, concernant les personnels, les produits, les unités de fabrication, les dépôts, les clients et leurs commandes.

Nous commençons par une description des propriétés du système d'information concerné.

II. 4.1 Description des propriétés du système d'information.

- 1) Une unité de fabrication peut fabriquer plusieurs produits.
Un produit élémentaire est fourni par un fournisseur unique.
- 2) Un dépôt peut stocker plusieurs produits.
- 3) Un produit peut être fabriqué par plusieurs unités de fabrication.
- 4) Un produit peut être stocké dans plusieurs dépôts.
- 5) Parmi les produits, certains sont des produits finis.
- 6) Une quantité unitaire d'un produit fini ou semi-fini est fabriquée à partir de quantités déterminées d'autres produits qu'une unité de fabrication commande à un dépôt.
- 7) A tout instant on sait pour une unité de fabrication et pour un produit si cette unité peut fabriquer, la quantité en cours de fabrication et les capacités maximales et minimales de fabrication.
- 8) On veut, également, connaître à chaque instant pour un dépôt et pour un produit qui y est stocké, la quantité disponible et la quantité en commande (aux unités de fabrication).
- 9) Lorsqu'aucune unité de fabrication ne fabrique, ou ne pourra fabriquer un produit et qu'aucun dépôt n'en stocke ce produit disparaît pour l'entreprise.
- 10) Une commande est relative à un client ; elle concerne un ou plusieurs produits finis dont elle spécifie la quantité demandée.
- 11) Un client est caractérisé par son nom, son adresse et peut s'adresser à un seul représentant.
- 12) On désire également connaître les produits et les quantités que ce client a déjà commandé.
- 13) Un employé ne peut travailler que dans une unité de fabrication on désire connaître les postes de travail que l'employé a déjà occupé.

A partir de ces informations le concepteur peut extraire les relations fonctionnelles suivantes, qui vont constituer l'ensemble " DEP ".

II. 4.2 Expression des relations fonctionnelles implicites : la construction de l'ensemble "DEP " de départ.

- rf1 = Unité-de-fabrication : no-unité → nom-unité, description, adresse,
nb-employé
- rf2 = Employé : no-insée → nom, sexe, résidence, âge, grade, nb-heures, taux,
no-unité
- rf3 = Poste-occupé : no-insée, no-poste → fonction, date-arrivée, date-départ
- rf4 = Produit : no-produit → libellé, désignation, stade-d'élab., prix-unitaire,
no-fournisseur
- rf5 = Fournisseur : no-fournisseur → nom, région, delai-de-livraison,
mode-de-règlement
- rf6 = Dépôt : no-dépôt → nom dépôt, lieu, surface
- rf7 = Fabrication : no-unité, no-produit → Qté-en-cours, Cap-min, Cap-maxi
- rf8 = Ligne-de-produit : no-produit, no-produit-composant → Qté-nécessaire
- rf9 = Stock : no-dépôt, no-produit → Qté-disponible
- rf10 = Mouvement-stock : no-unité, no-produit, no-dépôt → date-mouvement,
code-mouvement,
Qté-mouvementée
- rf11 = Commande : no-commande → date-de-commande, état-commandé, mode-d'ex-
pédition, no-client
- rf12 = Ligne-de-commande : no-commande, no-produit → Qté-demandée
- rf13 = Client : no-client → nom client, adresse-client, nom représentant
- rf14 = Historique-commande-client : no-client, no-produit → Qté-vendue,
taux-de-remise
- rf15 = Représentant : nom représentant → secteur, montant-vendu-mois
- rf16 = Règlement-commande-client : no-règlement → montant, date-de-règlement,
no-facture
- rf17 = Facture : no-facture → date-facture, no-bon-de-livraison
- rf18 = Bon-de-livraison : no-bon-de-livraison → date-sortie, no-commande

II. 4.3 Construction de la table de constituant.

Les constituants figurant dans " DEP " sont ensuite introduits dans le catalogue correspondant, après qu'un numéro interne unique ait été affecté à chacun.

Ainsi la table de constituants aura la forme suivante :

CINT	CNSYM	CType	Restriction sur les types de valeurs
001	no-unité	MOT	longueur max 6 caractères
002	nom-unité	MOT	longueur max 30 caractères
003	description	MOT	longueur max 500 caractères
004	adresse	MOT	longueur max 60 caractères
005	nb-employé	NUM	de 0 à 9999
006	no-insee	MOT	longueur max 6 caractères
007	nom employé	MOT	longueur max 30 caractères
008	sexe	LVAL	(M, F) 2, 10

..... et ainsi de suite.

II. 4.4 Obtention de la couverture minimale de l'ensemble "DEP".

Il s'agit maintenant d'éliminer les redondances qui peuvent éventuellement exister dans l'ensemble de dépendances fonctionnelles de départ et de trouver, à l'aide des propriétés mathématiques (transivité, pseudo-transitivité etc...) un ensemble minimal de dépendances les recouvrant toutes. Ces redondances infiltrées au niveau du modèle conceptuel présentent une source d'incohérence des informations qui se répercutera au niveau de l'implémentation logique de la base.

Dans le cas de notre exemple, toutes les relations fonctionnelles données au départ comme informations extérieures sont élémentaires et directes.

Par conséquent, on peut constater que la couverture minimale de " DEP " sera égale à " DEP ".

Une relation fonctionnelle $X \xrightarrow{R} Y$ sera dite élémentaire si pour toute partie $X' \subset X$, $X' \xrightarrow{R} Y$ n'est pas une relation fonctionnelle et si elle n'est pas triviale.

Une relation fonctionnelle élémentaire $X \xrightarrow{R} Y$ sera dite directe s'il n'existe aucun constituant composé Z n'appartenant pas à X tel que :

$$X \rightarrow Z \rightarrow Y$$

où les relations fonctionnelles $X \rightarrow Z$ et $Z \rightarrow Y$ sont élémentaires.

Remarque : Notre exemple de référence a été forcé de façon à simplifier notre tâche. Les informations prises dans un tel cas ne l'auraient peut être pas été dans un cas réel.

II. 4.5 Etablissement du catalogue de la couverture minimale de " DEP "

Chaque relation fonctionnelle appartenant à la couverture minimale de l'ensemble " DEP " sera introduite dans le système automatique et stockée dans le catalogue correspondant comme nous venons de le préciser dans un paragraphe précédent c'est ainsi que la relation fonctionnelle : " Unité de fabrication " aura à titre d'exemple la forme suivante :

Unité de fabrication	001	1	0	0				1	0	1	1	1	1				4
		1					n	1	2							n	
nom symbolique	no				IMPG		NBG			IMPD						NBD	
	interne																

II. 5 Comment engendrer la vue relationnelle de départ.

En résumé, on peut conclure, que la démarche de la transformation de structure de données commence lorsqu'on a pu obtenir en machine :

- Un catalogue de constituants fournis à priori
- Un ensemble dit EREL de relations en 3.F.N. tel que

$$EREL = \{ R_1(C^1) , R_2(C^2) , \dots , R_n(C^n) \}$$

Chaque relation est associée bijectivement à une relation fonctionnelle appartenant à la couverture minimale de l'ensemble de dépendances fonctionnelles de départ.

Elle peut avoir une seule clé ; celle-ci peut être simple ou composée.

Or, il se trouve dans certains cas, qu'une relation produite par l'algorithme de synthèse possède des clés équivalentes. Dans ce cas, il appartient au concepteur de recourir à l'une de ces clés, qui sera reconnue comme clé primaire par tous les utilisateurs. Il est nécessaire de refuser, que cette clé puisse prendre des valeurs inconnues lors de l'introduction des données.

Remarque : Cette restriction évite la prise en compte de circuits dans l'ensemble des relations traduites. Cependant elle modifie l'ensemble de relations de départ, car toutes les relations issues des clés non-choisies seront ignorées. De plus, cette restriction aura des conséquences ultérieures sur la façon d'utiliser la base en question.

Dans EREL nous pourrions distinguer trois catégories de relations :

a) Les relations bâties sur des constituants qui ne constituent pas la clé de n'importe quelle autre relation de l'ensemble.

Exemple :

Dans notre exemple de référence on peut voir les relations :
Unité de fabrication (no-unité, nom unité, description, adresse,
nb-employé)

Dépôt (no-dépôt, nom dépôt, lieu, surface)

etc....

N.B. Les clés sont soulignées.

b) Les relations bâties sur des constituants qui constituent.

1) La clé de la relation

2) Des constituants éventuellement non-clé

3) Des constituants clé provenant d'autres relations chacun symbolisant une association de type n:1

Exemple :

Reprenons la relation "Commande" précédemment définie :

Commande (no-commande, date-commande, état-commande, mode-d'expédition,
no-client)

On voit que le constituant "no-client" provient de la relation "Client" et forme la clé de celle-ci. Son apparition dans la relation "Commande" reflète une association de type n:1 entre commande et client, notamment pour exprimer qu'à une commande donnée il ne correspond qu'un seul client.

c) Des relations où tous les constituants sont des clés provenant d'autres relations. La clé d'une telle relation est formée de la concaténation de tous ces constituants. La relation fonctionnelle associée aura comme partie droite l'un des constituants fictifs :

$\theta_1, \theta_2, \dots, \theta_m$

Exemple :

Si dans l'exemple cité la réalité évolue et selon les conditions de travail, un employé peut travailler pour plusieurs unités de fabrication, on aura de plus la relation : "Emploi" définie comme suit :

Emploi (no-unité, no-insee, θ_1),

Ce qui symbolise une association de type n : m entre "Unité de Fabrication" et "Employé".

II. 6 La transformation d'ensemble EREL de relation en une structure réseau de type SOCRATE ou CODASYL

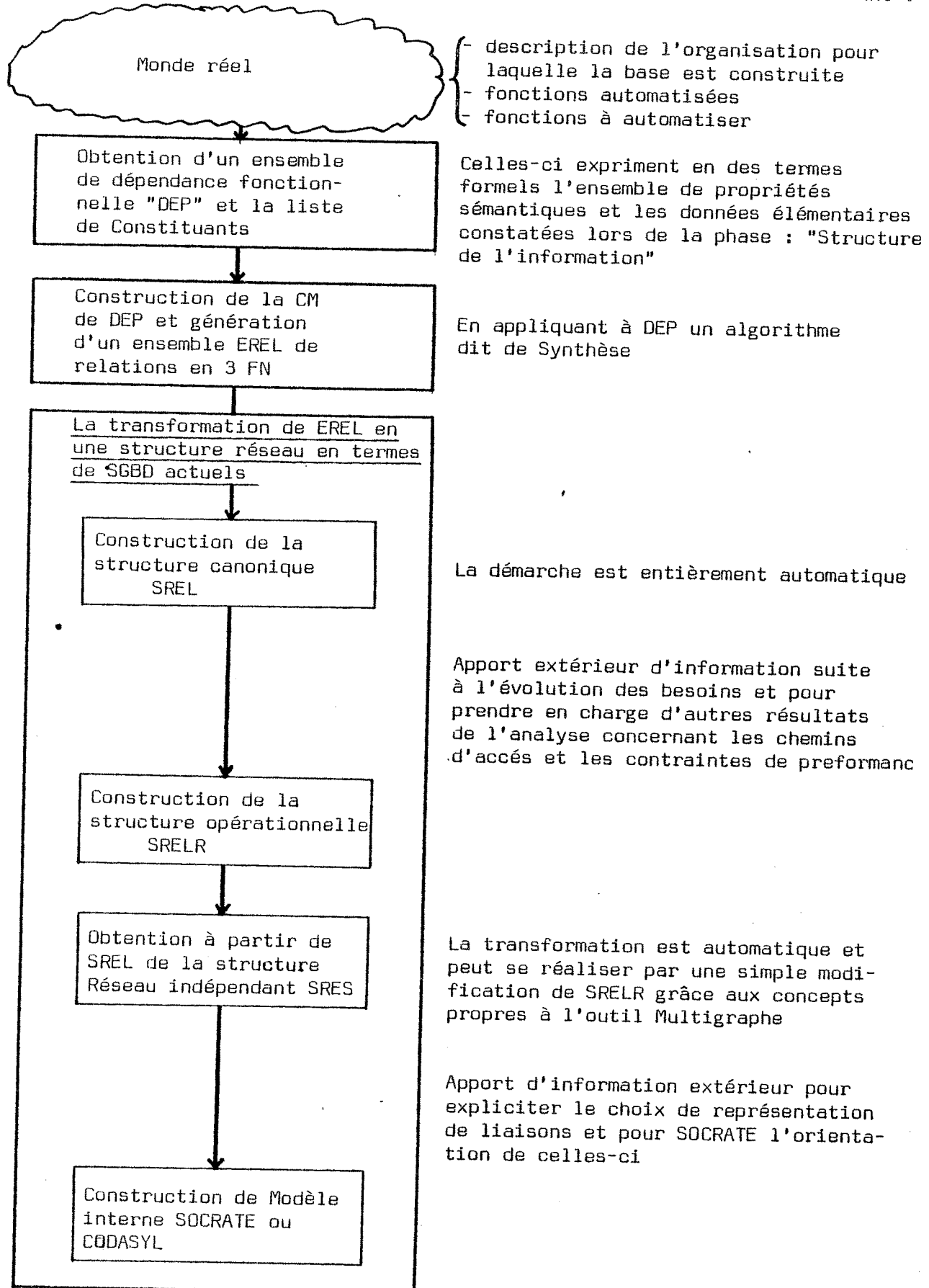
La deuxième partie de ce rapport est consacrée à la description détaillée et à la mise en oeuvre du processus de transformation de structures de données. Ceci permet tout d'abord, à partir d'un ensemble de relations " EREL " d'aboutir à une structure relationnelle " SREL " correspondante. Cette structure initiale est modifiée pour donner la structure opérationnelle "SRELR".

Il s'agit ensuite de transformer " SRELR " en une structure réseau compatible : " SRES " pour, enfin, engendrer à partir d'elle la structure logique associée à un système déterminé de base de données : par exemple SOCRATE ou CODASYL.

Nous nous proposons de conduire ce processus par étapes, certaines de ces étapes sont automatisables, d'autres nécessitent un apport d'informations extérieures.

II. 7 Conclusion

Nous pouvons illustrer toute cette démarche à l'aide du schéma suivant :



CHAPITRE III

LA TRANSFORMATION D'UN ENSEMBLE DE RELATIONS EN UNE STRUCTURE
RESEAU DE TYPE SOCRATE OU CODASYL

Cette démarche consiste à mettre au point une méthode et un ensemble d'outils informatiques qui peuvent servir de trame à la construction effective des structures logiques de système de bases de données gérés par les S.G.B.D. actuels.

Pour atteindre un tel objectif nous donnerons les spécifications d'un processus de transformation de structures de données, dont l'intérêt est d'aborder la sémantique des liaisons entre les différents constituants de la base exprimées à l'aide du modèle n-aire de E.F. Codd pour enfin, engendrer à partir d'elles la structure réseau associée à un système déterminé de base de données, par exemple : SOCRATE ou CODASYL.

Comme cette démarche est l'inverse de celle étudiée dans [Leo 76] et [BD 76] nous nous sommes servis des notions déjà présentées et des résultats obtenus, afin de garder certaines analogies avec ces travaux.

En cherchant à rendre cette transformation la plus indépendante possible du S.G.B.D. utilisé nous nous proposons de la conduire par étapes :

III. 1) La première étape consiste à ordonner l'ensemble "EREL" de relations de départ en une structure Relationnelle initiale que nous appellerons "SREL". L'intérêt est de pouvoir faciliter ensuite le passage à la structure réseau finale. Grâce aux concepts développés à propos du modèle Multigraphe la transformation : EREL \rightarrow SEREL est entièrement automatique.

Définition d'une structure "SREL"

Une structure relationnelle "SREL" est formée :

- des relations R de l'ensemble considéré "EREL". A chaque relation $R_i \in \text{EREL}$ est associé un ensemble de constituants C^i dont un sous-ensemble forme la clé $K(R_i)$ de R_i .
- de liaisons entre deux relations ; elles sont de deux types :
Etant données deux relations R_i et R_j bâties sur les ensembles de constituants C^i et C^j et de clé respectives : $K(R_i)$ et $K(R_j)$, on aura :
 - . une liaison fonctionnelle de R_i vers R_j (type F)

Dans ce cas, $K(R_j)$ appartient à C^i sans être contenue dans $K(R_i)$; la relation fonctionnelle élémentaire $K(R_i) \rightarrow K(R_j)$ existe sans être triviale.

- . une liaison hiérarchique de R_i vers R_j (type H)

Dans ce cas, $K(R_j)$ appartient à C^i et est contenue dans $K(R_i)$; puisque $K(R_j) \subset K(R_i)$, la relation fonctionnelle élémentaire $K(R_i) \rightarrow K(R_j)$ existe et est triviale.

Remarquons qu'en raison de la définition même d'une structure relationnelle "SREL" : A un ensemble de relations particulier, il ne correspond qu'une structure "SREL" et réciproquement. De plus, dans la suite du processus de transformation de "SREL" en une structure réseau, il ne sera plus possible de rajouter ou d'enlever des relations.

Il s'agit donc au cours de cette étape de choisir l'ensemble de relations que l'on veut traduire dans la structure réseau finale.

Deux critères d'efficacités influencent ce choix :

- a) la base de données doit comporter un minimum de redondances et permettre un contrôle de cohérence des informations.
- b) la structure finale doit être construite en fonction des renseignements que les futures utilisateurs souhaitent obtenir.

Ces deux critères sont en général contradictoires.

L'une des difficultés de la conception réside dans la recherche d'un juste équilibre entre ces deux objectifs.

Toutefois, nous avons montré l'intérêt, qu'il y aurait à prendre "EREL" (ensemble non-redondant de relations) comme ensemble de relation à traduire. Le schéma final de la base ne contiendra alors aucune imperfection intra ou inter-relations. Cet avantage nous fait retenir, au départ "EREL" comme solution canonique. La structure relationnelle "SREL" correspondante aura les particularités de n'accepter aucune information superflue et de ne privilégier aucun chemin d'accès. En outre, elle représente, au moment de la conception, l'image aussi fidèle que possible du système d'information concerné.

Or tout système d'information évolue vite sous l'effet de phénomènes extérieurs, tel que l'arrivée d'une nouvelle information ou la prise en compte d'une décision qui déclenche un ou plusieurs traitements. Il est donc impossible de retenir la solution canonique pour longtemps, si l'on cherche à construire d'une manière rigoureuse et précise une structure de données répondant le mieux à l'évolution constante des besoins en information et aux changements de règles de fonctionnement de l'organisation pour laquelle la base a été créée. Ces évolutions interdisent, en général, à la base d'avoir une structure figée.

Il est aussi évident, que si l'on désire construire une base de données dotée de bonnes performances, il s'agit, à ce niveau, de prendre en compte d'autres résultats de l'analyse relatives aux chemins d'accès et aux mécanismes de mise-à-jour et de destruction de données pour améliorer les performances et faire face au manque d'efficacité de traitements, qui réclament la création de nouveaux chemins d'accès logiques. Ces contraintes obligent, généralement, le concepteur à réviser la structure canonique "SREL" pour prendre en charge

cette évolution.

A ce propos, l'utilisateur est autorisé à introduire de la redondance par retour à la 2 FN et même à la 1 FN pour des raisons d'efficacité.

L'évolution du système d'information peut se traduire lors de l'apparition de nouveaux besoins ou lors du changement de règles d'intégrité par :

- a) l'introduction de nouvelles relations et la modification structurale de celles existantes.
- b) l'introduction de nouveaux constituants et la création de redondances.
- c) l'introduction de nouvelles liaisons intra et inter-relations, même redondantes, causée par la fréquence importante d'utilisation de ces liaisons.
- d) la modification de types d'association causée par un changement dans l'environnement modélisé.

Cette situation pose des problèmes délicats de restructuration. Elle nous amène, avant d'exécuter le passage à la structure réseau finale, à proposer une étape intermédiaire au cours de laquelle on étudie la modification de "SREL" et la mise en place d'une structure opérationnelle plus performante au niveau des accès, répondant le mieux à l'évolution des besoins.

III. 2) De là, la deuxième étape du processus de transformation consiste à modifier la structure canonique "SREL" et à déterminer à partir des opérations sur les catégories et les fonctions relatives au modèle Multigraphe, les différentes possibilités de restructuration et de modification à effectuer en vue d'intégrer les informations complémentaires et d'aménager un accès plus facile à des demandes diverses.

Le but de cette étape intermédiaire entre "SREL" et la structure réseau sera de réaliser une optimisation de la structure conceptuelle relative aux accès conformément à l'évolution des situations modélisées et ceci en conservant un maximum des acquis sémantiques déjà obtenus. Cette étape et cette optimisation sont particulièrement justifiées dans le cadre d'une modélisation relationnelle, où les liaisons établies sont des relations fonctionnelles n'impliquant aucune notion d'accès, rendant ainsi explicitement les choix à faire.

Dans un premier temps nous allons donner la définition d'une structure relationnelle opérationnelle que nous appellerons : "SRELR". La mise en oeuvre de la transformation. : "SREL" en "SRELR" est donnée en détail dans le chapitre IV

Définition d'une structure : "SRELR"

Une structure SRELR est définie par un Multigraphe orienté tel que :

- Un bloc est formé de la réunion des constituants d'une ou plusieurs relations de "SREL" ou des relations supplémentaires que l'utilisateur souhaite y introduire. Un bloc a une seule clé, celle-ci peut comporter plusieurs constituants comme dans "SREL".
- Une liaison entre deux blocs X et Y peut avoir deux types.
 - . fonctionnelle (type F) allant de X vers Y
 - . hiérarchique (type H) allant de X vers YElles sont définies comme dans SREL
- Un constituant peut se trouver dans plusieurs blocs simultanément comme dans SREL.

Le schéma "SRELR" ne sera plus un système sans redondances. Celles-ci provoquent des imperfections de fonctionnement notamment des contrôles de cohérence plus longs. Leur introduction à ce niveau semble aller à contre courant et détruire les acquis précédents.

Or, les redondances envisagées ne seront prises en compte que si elles entraînent une amélioration véritable de l'efficacité d'utilisation. Généralement cela ne doit pas remettre en cause entièrement le processus de construction de la structure "SREL". Au contraire les algorithmes proposés nécessitent pour une utilisation optimale d'avoir comme point de départ la structure canonique. Dans ce cas, le système automatique étudie le comportement de ces modifications sur la structure et continue à maintenir la qualité de la base et à préserver la cohérence des données.

Toutefois, certains changements (l'introduction de nouvelles données venant perturber les règles précédemment admises) peuvent imposer une nouvelle exécution des algorithmes de normalisation et les modifications associées. Ce processus très lourd ne sera utilisé que si les changements sont suffisamment importants pour être justifiés.

Pour le moment, les initiatives de modification sont laissées au concepteur, qui doit savoir si les changements présentent un intérêt opérationnel suffisant par rapport aux imperfections qu'ils entraînent.

Cependant, il s'agit de lui indiquer les moyens et les opérations à entreprendre pour modifier la structure "SREL" et de lui fournir également des renseignements sur les fréquences d'accès aux différents éléments de la structure. Ces statistiques peuvent être calculées à partir de l'analyse des requêtes émises par les utilisateurs lors de l'exploitation de la base. Il est évident, qu'il faut disposer d'un analyseur de requêtes relationnelles et d'un interpréteur pour réaliser l'interface avec le schéma global.

Après leur analyse, les requêtes peuvent être paramétrisées pour préparer certains arguments de recherche, qui permettent au concepteur de vérifier d'une part, que toutes les modalités d'utilisation prévues par les traitements décrits sont bien satisfaits par le schéma existant et de préciser, d'autre part, les nouveaux besoins avant de les intégrer.

L'intervention des utilisateurs est nécessaire pour expliciter les ambiguïtés sémantiques relevées par les questions, en particulier, les problèmes de synonymes et de polysémies.

Le système doit aussi permettre aux concepteurs de connaître l'impact de chaque modification sur la structure.

III. 3 La troisième étape de notre démarche consiste à passer de la structure opérationnelle "SRELR" vers une structure réseau en termes de S.G.B.D actuels.

Nous rappellerons avant tout les caractéristiques générales d'une structure réseau.

Caractéristiques générales d'une structure Réseau :

Une structure Réseau est composée :

- de groupes de constituants G , à chaque groupe $G_1 \in G$ est associée une clé $K(G_1)$; celle-ci est telle qu'à une de ces réalisations il correspond une seule réalisation de groupe.
- de liaisons directionnelles entre deux groupes, et à les propriétés suivantes :
- chaque groupe peut atteindre plusieurs groupes et être atteint par plusieurs.
- plusieurs liaisons peuvent exister entre deux mêmes groupes.

Pour pouvoir prendre en compte différents S.G.B.D. Réseau, nous nous sommes servis de la notion introduite dans [Leo 78] d'une structure Réseau Indépendante des systèmes utilisés SOCRATE, CODASYL mais dont la définition est basée sur leurs caractéristiques communes.

Cette structure que nous appellerons structure "SRES" peut aussi bien décrire une structure SOCRATE qu'une structure CODASYL.

Cependant elle présente de multiples avantages par rapport à ces structures sources :

- a) au niveau du modèle logique la structure "SRES" ne contiendra aucune information superflue liée aux caractéristiques d'un L.D.D particulier.
- b) elle permet de prendre en considération les problèmes logiques qui se poseront sur toutes structures de types réseau quelque soit le S.G.B.D. utilisé. Ainsi la transposition de "SRES" à des

systemes autres que SOCRATE et CODASYL pourra se faire d'une maniere analogue.

Ces avantages nous ont amené à décomposer la transformation : Structure "SREL" → Structure logique SOCRATE ou CODASYL et à donner premièrement les éléments qui permettent de passer d'une structure "SREL" à une structure "SRES" pour enfin engendrer à partir de celle-ci la structure réseau en termes du S.G.B.D. concerné.

Nous rappelons d'abord la définition de la structure réseau indépendante comme elle sera interprétée dans notre démarche.

Définition d'une Structure Réseau : "SRES"

Une structure "SRES" est formée :

- de groupes de constituants G . A chaque groupe $G_i \in G$ sera associé son ensemble de constituants C^i et sa clé $K(G_i)$. Celle-ci ne peut plus être composée. Les groupes proviennent des noeuds associés au Multigraphe : "SREL".
- de liaisons entre deux groupes, tel que G_i et G_j désignent deux groupes. Elles proviennent des relations associées à la structure "SREL" et sont de trois types :

Etant données deux relations R_i et R_j associées bijectivement dans "SREL" aux noeuds correspondants aux groupes G_i et G_j , on aura dans "SRES".

- . une liaison fonctionnelle allant de G_i vers G_j (type F)

Dans ce cas la relation fonctionnelle élémentaire $K(R_i) \rightarrow K(R_j)$ existe sans être triviale.

- . une liaison hiérarchique allant de G_i vers G_j (type H)

Dans ce cas :

- R_i n'est pas une relation de la forme $[K(R_i), \theta_i]$
- $K(R_j) \subset K(R_i)$

- . deux liaisons binaires l'une allant de G_i vers G_j et l'autre allant de G_j vers G_i (type B)

Dans ce cas, $K(R_j) \subset K(R_i)$ et la relation R_i est de la forme $[K(R_i), \theta_i]$

En vue de réduire l'intervention de l'utilisateur nous avons tenté d'associer à la structure "SRES" les contraintes suivantes :

- Un constituant ne peut appartenir qu'à un seul groupe.

- Une liaison hiérarchique peut être aussi représentée par une liaison fonctionnelle puisque dans ce cas la relation fonctionnelle élémentaire : $K(R_i) \rightarrow K(R_j)$ existe et est triviale. Alors si d'un groupe G partent plusieurs liaisons hiérarchiques, il s'agit d'en privilégier une parmi elles, qui sera matérialisée par une liaison hiérarchique dans "SRES". Les autres sont transformées en liaisons fonctionnelles.

Nous reviendrons en détail sur ces contraintes dans le chapitre IV.

III. 4 La dernière étape de notre processus de transformation correspond pratiquement à l'obtention, à partir de la structure "SRES" de la structure réseau finale exprimée en termes du S.G.B.D concerné.

Ce n'est qu'à ce niveau qu'il faudra prendre en considération les caractéristiques spécifiques d'un L.D.D particulier.

La transformation de la structure SRES en une structure de type SOCRATE ou CODASYL peut être réalisée par un interface simplifié qui nécessite toutefois l'intervention du concepteur pour spécifier les choix de types de liaisons et l'orientation de celles-ci.

Soit G_i et G_j deux groupes appartenant à la structure "SRES". A G_i et G_j sont associés leurs ensembles de constituants et leurs clés respectives $K(G_i)$ et $K(G_j)$.

La transformation à l'un des S.G.B.D. utilisés peut alors se faire d'une manière simple à partir du tableau suivant :

Structures SRES	SOCRATE	CODASYL
Groupe G_i de clé $K(G_i)$	ENTITE X constituant DISCR = $K(G_i)$	RECORD X Location mode de type CALC ou DIRECTE
liaison fonctionnelle (type F) allant de G_i vers G_j	REFERENCE de nom L de l'ENTITE X vers l'ENTITE Y	LIEN avec RECORD Y : Parent X : Enfant
liaison hiérarchique (type H) allant de G_i vers G_j	ENTITE - ENGLOBEE X dans Y	LIEN HIERARCHIQUE avec Y : Parent X : Enfant ou encore X GROUPE REPETITIF de Y
liaison binaire (type B) allant de G_i vers G_j et/ou de G_j vers G_i	INVERSE de nom I ou ENTITE-REFERENCE de X vers Y ou de Y vers X <u>Remarque</u> : Dans certaines versions de SOCRATE une liaison binaire peut aussi être représentée par une REFERENCE avec ANNEAU [S ₁ 76]	ENREGISTREMENT-LIAISON de nom L et LIENS avec X : Parent (puis Y) L : Enfant

En résumé, on peut dire que nous avons essayé de définir dans chaque étape de la transformation un ensemble de paramètres dont la détermination des valeurs conduit à établir une structure réseau opérationnelle et effective.

Nous rappelons, que nous travaillons uniquement au niveau des noms, cela signifie que nous ne traitons pas les données associées à la structure.

CHAPITRE IV

LA MISE EN OEUVRE, A PARTIR DU MODELE MULTIGRAPHE DU PROCESSUS DE
TRANSFORMATION D'UN ENSEMBLE DE RELATION EN UNE STRUCTURE RESEAU DE TYPE
SOCRATE OU CODASYL

Dans ce chapitre, nous précisons comment un tel système peut être réalisé, à partir des concepts de catégories et de fonctions développées pour le modèle Multigraphe.

La démarche de transformation se décompose du point de vue informatique en plusieurs phases ; chacune d'elle conduit à une classe de structures résultat. La détermination d'une structure particulière dans chacune de ces classes dépendra d'un apport d'information extérieures. Aussi l'automatisation d'un tel système ne peut, en effet, être complète.

Toute référence à l'outil Multigraphe sur lequel repose l'implantation informatique des algorithmes de transformation est faite en citant la fonction concernée si elle a été déjà définie dans le chapitre I ou bien en précisant les spécifications selon lesquelles il faudra modifier une fonction ou en créer une nouvelle adaptée à notre démarche.

IV. 1 Première Phase : la Transformation "EREL" → "SREL"

Le processus de transformation d'un ensemble de relation "EREL" en une structure relationnelle "SREL" correspond à l'obtention, à partir de EREL, d'un Multigraphe orienté de noeuds et des arêtes : Les relations sont aux noeuds, les liaisons entre les relations sont représentées par les arêtes. A la fin de cette étape toutes les relations à traduire à partir d'une vue relationnelle de départ sont contenues dans le Multigraphe ("SREL")

Celui-ci est formé :

- de noeuds représentant les relations. Chaque noeud est associé bijectivement à une relation $R_i \in \text{EREL}$, formée de la réunion des constituants d'une ou plusieurs relations fonctionnelles élémentaires ayant même partie gauche. Un ou plusieurs de ces constituants forment la clé $K(R_i)$ de R_i , qui à son tour donne la clé du noeud. (Rappelons ici, qu'une relation appartenant à l'ensemble "EREL" ne peut avoir qu'une clé et que cette clé peut être simple ou composée).
- des arêtes : celles-ci sont orientées dans le même sens que les liaisons entre relations. Elles sont de deux types notés :
 - . F, ce qui correspond à une liaison fonctionnelle entre deux relations R_i et R_j allant de R_i vers R_j .
 - . H, ce qui correspond à une liaison hiérarchique entre deux relations R_i et R_j allant de R_i vers R_j .

La construction du graphe "SREL" est réalisée automatiquement grâce à l'outil Multigraphe.

IV. 1.1 Construction des noeuds : rattachement de constituants et des clés

Chaque relation de EREL donnera naissance dans le graphe "SREL" à un noeud, auquel on associe un nom, un numéro interne et l'ensemble de ces constituants. Pour le faire, il suffit d'exécuter les appels aux procédures suivantes :

- 1) NCREER : Créer dans le graphe "SREL" un noeud de nom R_i et de numéro interne (i)
- 2) Pour chaque constituant associé à R_i faire :
CCREER : Créer un constituant de numéro interne (k) lui associé au noeud R_i
- 3) CRCLE : Créer une clé parmi les constituants associés au noeud R_i

La difficulté de cette étape est de déterminer les liaisons hiérarchiques et fonctionnelles que l'on prendra effectivement en compte dans "SREL"

Nous allons proposer par la suite une démarche à suivre pour établir la nature des liaisons existantes et pour enfin les représenter dans le graphe "SREL" à l'aide de l'outil Multigraphe.

IV. 1.2 Création des liaisons hiérarchiques.

Pour la mise en évidence des liaisons hiérarchiques nous allons définir tout d'abord un certain nombre d'éléments qui serviront de support à cette partie de la démarche :

- 1) Créer une liste L_0 des noeuds rangés par cardinalité décroissante tel que :

$$R_i < R_j \iff \text{card}(K(R_i)) = \text{card}(K(R_j)) + 1$$

- 2) A chaque noeud R_i nous associerons la liste L_{R_i} des noeuds R_j tel que :

$$K(R_j) \subset K(R_i) \quad (i < j \leq n)$$

- 3) Chaque élément R de la liste L_0 ou d'une liste L_R est composé de :
 - la clé du noeud concerné
 - d'un pointeur Point (R) sur la liste L_R si celle-ci a été créée
 - d'une marque, MARK1(R) valant 0 ou 1
- 4) La création de deux nouvelles fonctions utilisant le Multigraphe.
 - a) CRLIST (R)

Pour un élément R de la liste courante cette fonction crée la liste L_R par recherche séquentielle dans la liste courante des éléments R_i tel que $K(R_i) \subset K(R)$

Le pointeur associé à un élément de L_R est initialisé à nil et la marque à 0. De plus cette fonction met à jour le pointeur associé à l'élément R ($\text{Point}(R) = L_R$)

b) SUIVL (L)

Cette fonction donne le premier élément non encore traité d'une liste L. (C'est à dire le premier élément tel que $\text{MARK 1}(R) = 0$) De plus elle fait de celui-ci l'élément courant.

Si tous les éléments ont été traités alors $\text{SUIVL}(L) = 0$

Remarques :

- a) on appellera "élément courant" (respectivement "liste courante") l'élément (resp. la liste) que l'on est entrain de traiter.
- b) on appellera "liste origine" la liste où se trouve l'élément associé à la liste courante dans le contexte où l'on se trouve.

IV. 1.2.1 Algorithmes de recherche de liaisons hiérarchiques.

Début

Liste courante = L_0 (la liste des noeuds rangés par cardinalité décroissante)

Liste origine (L_0) = vide

(i) Initialisation

Pour tout élément R_{oi} de L_0 ($1 \leq i \leq n$), on fait :

$\text{MARK1}(R_{oi}) = 0$

$\text{POINT}(R_{oi}) = \text{NIL}$

(ii) élément courant = SUIVL (liste courante)

Si élément courant = 0 ((il n'y a plus d'éléments à traiter)

Si liste courante = L_0 alors aller à Fin

Sinon liste courante = liste origine aller à (ii)

(iii) Si POINT (élément courant dans L_0) = Nil

MARK1 (élément courant) = 1

CRLIST (élément courant)

Sinon POINT (élément courant) = POINT (élément courant dans L_0)

liste courante = POINT (élément courant)

Aller à (ii)

Fin

IV. 1.2.2 Obtention des liaisons hiérarchiques

- a) A chaque liste L_0 et L_R on peut associer un ensemble d'éléments R_1, \dots, R_n POINT (R_i) \neq NIL ($1 \leq i \leq n$)
- b) A chacun de ces éléments R_i est associée une liste L_{R_i} .
- c) A chaque liste L_{R_i} correspond un ensemble d'éléments : R_{i1}, \dots, R_{ij} .
POINT (R_{ij}) \neq Nil ($1 \leq j \leq l$)

On construit la liaison hiérarchique

$$R_i \xrightarrow{H} R_{ij}$$

IV. 1.3 Création des liaisons fonctionnelles

Nous utiliserons les notions et les résultats de (IV.1.1). De plus nous associerons à chaque élément de la liste L_0 deux nouvelles marques : MARK 2 et MARK 3 et nous définirons la fonction suivante :

SUIVLI (L_0)

Cette fonction donne le premier élément non encore traité de L_0 tel que : MARK 3 = 1 et MARK 2 = 0

IV. 1.3.1 Algorithme de recherche de liaisons fonctionnelles

La partie de l'algorithme que nous décrivons ici donne les liaisons fonctionnelles partant d'un noeud pour obtenir toutes les liaisons fonctionnelles de "SREL" il faut le réappliquer à chaque noeud.

Soit le noeud R_i de clé $K(R_i)$

Début

(i) Initialisation

Pour tout élément de L_0

MARK 2 = 0

MARK 3 = 0

(ii) Pour tout élément R_j de L_0 ($i \neq j$)

Si $K(R_j) \subset C^i$

et $K(R_j) \neq K(R_i)$

alors MARK 3(R_j) = 1

- (iii) élément courant = SUIVL1 (L_0)
 Si élément courant = 0 alors aller à (iiii)
 Si Point(élément courant) = Nil alors
 MARK 2 (élément courant) = 1
 aller à (iii)
- (iiii) Pour tout élément R_j de L_0
 Si MARK 3(R_j) = 1 et
 MARK 2(R_j) = 1
 ou construit la liaison fonctionnelle
- $$R_i \xrightarrow{F} R_j$$

Fin

IV. 1.4 Construction des arêtes hiérarchiques et fonctionnelles

A ce niveau on pourra représenter les liaisons hiérarchiques et fonctionnelles déjà obtenues en faisant appel à la fonction ACREER :
 Créer une arête de type donné d'un noeud de départ R_i et d'un noeud d'arrivée R_j .

IV. 1.5 Conclusion

Les algorithmes de recherche de liaisons associées à la structure "SREL" nécessitent la modification de la procédure TRI pour effectuer le tri des noeuds suivant l'ordre défini dans (IV. 1.1), notamment

$$R_1 < R_2 \iff \text{card}(K(R_1)) = \text{card}(K(R_2)) + 1$$

La procédure LSTCONS est aussi modifiée pour donner le nombre de constituants qui forment la clé d'un noeud.

En plus, nous avons réalisé la programmation de nouvelles procédures correspondantes aux fonctions : CRLIST (L), SUIVL(L) et SUIVL1 (L_0)

Le tableau suivant donne en résumé les opérations de construction du graphe "SREL" à partir de l'ensemble EREL.

Fonction de transformation	Fonction du Multigraphe
(1) Création d'un NOEUD	NCREER CCREER CRCLE
(2) Création de liaisons hiérarchiques et fonctionnelles	TRI LSTCONS CRLIST SUIVL SUIVL 1 ACREER

IV. 1.6 Exemple

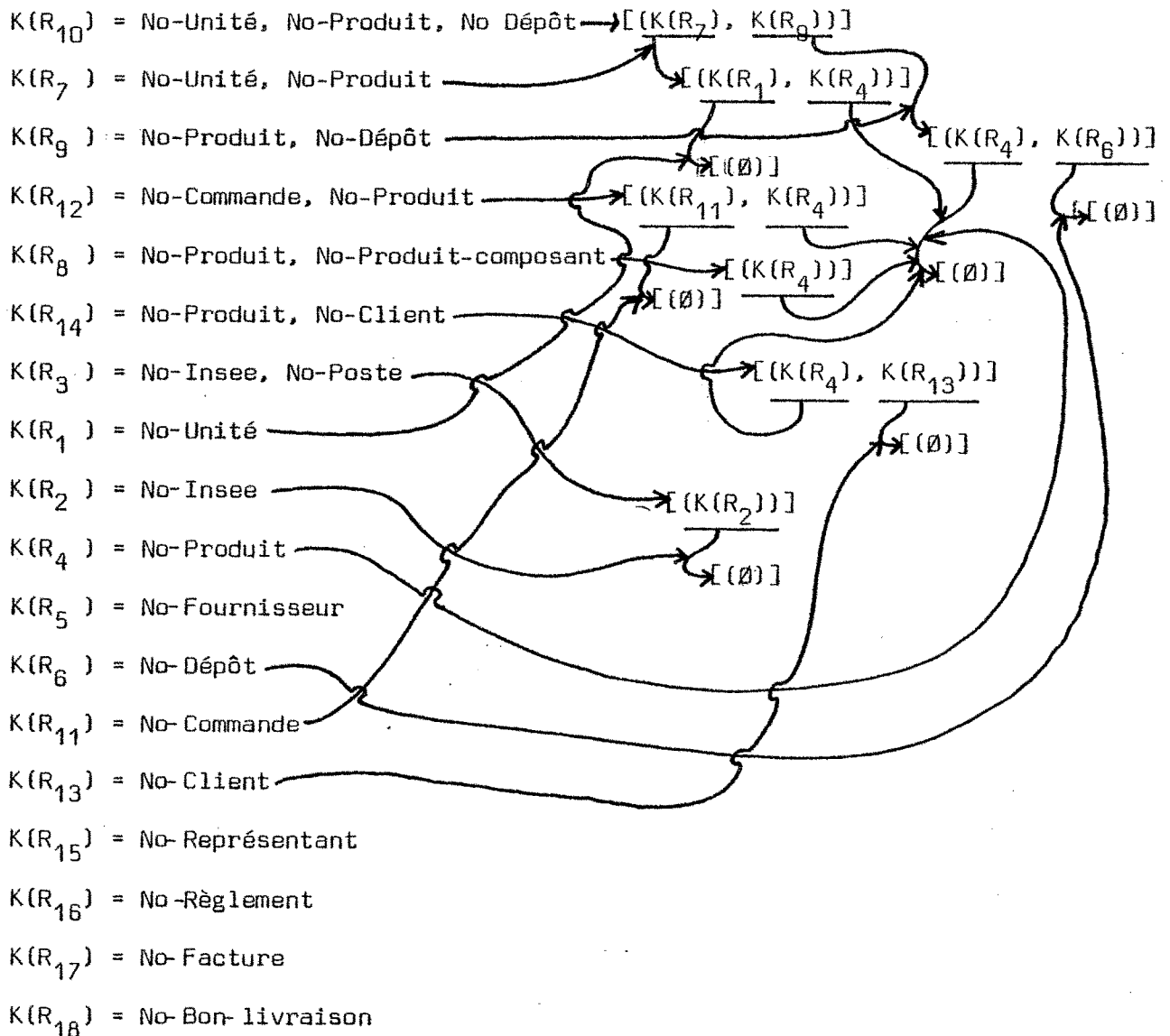
Nous allons illustrer cette démarche en faisant toujours référence à l'exemple cité précédemment.

Etant donné l'ensemble de relations EREL = {Unité de fab., Employé, Poste-occupé, Produit, Fournisseur, Dépôt, Fabrication, Ligne-de-produit, Stock, Mouvement-stock, Commande, Ligne-de-commande, Historique-commande-client, Client, Représentant, Règlement-commande-client, Facture, Bon-de-livraison}.

1) on crée dans SREL un ensemble de noeuds, chacun est associé bijectivement à une relation $R_i \in EREL$ auquel on attache son ensemble de constituants et sa clé.

2) Détermination des liaisons hiérarchiques.

- a) tri des noeuds et création de la liste L_0
- b) obtention des liaisons hiérarchiques.



On aura alors les liaisons hiérarchiques suivantes :

- | | | |
|------------------------|---------------------------|----------------------------|
| 1) R_{10} vers R_7 | 5) R_9 vers R_4 | 9) R_8 vers R_4 |
| 2) R_{10} vers R_9 | 6) R_9 vers R_6 | 10) R_{14} vers R_4 |
| 3) R_7 vers R_1 | 7) R_{12} vers R_{11} | 11) R_{14} vers R_{13} |
| 4) R_7 vers R_4 | 8) R_{12} vers R_4 | 12) R_3 vers R_2 |

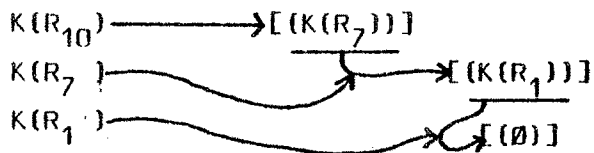
3) Détermination des liaisons fonctionnelles

Soit le noeud "Employé" avec comme relation associée :

Employé (No-Insee, Nom, Sexe, Résidence, Age, Grade, Nb-Heure, Taux,
No-Unité)

Obtention de liaisons fonctionnelles partant de "Employé"

Si dans L_0 on a la structure suivante tel que $MARK\ 3 = 1$

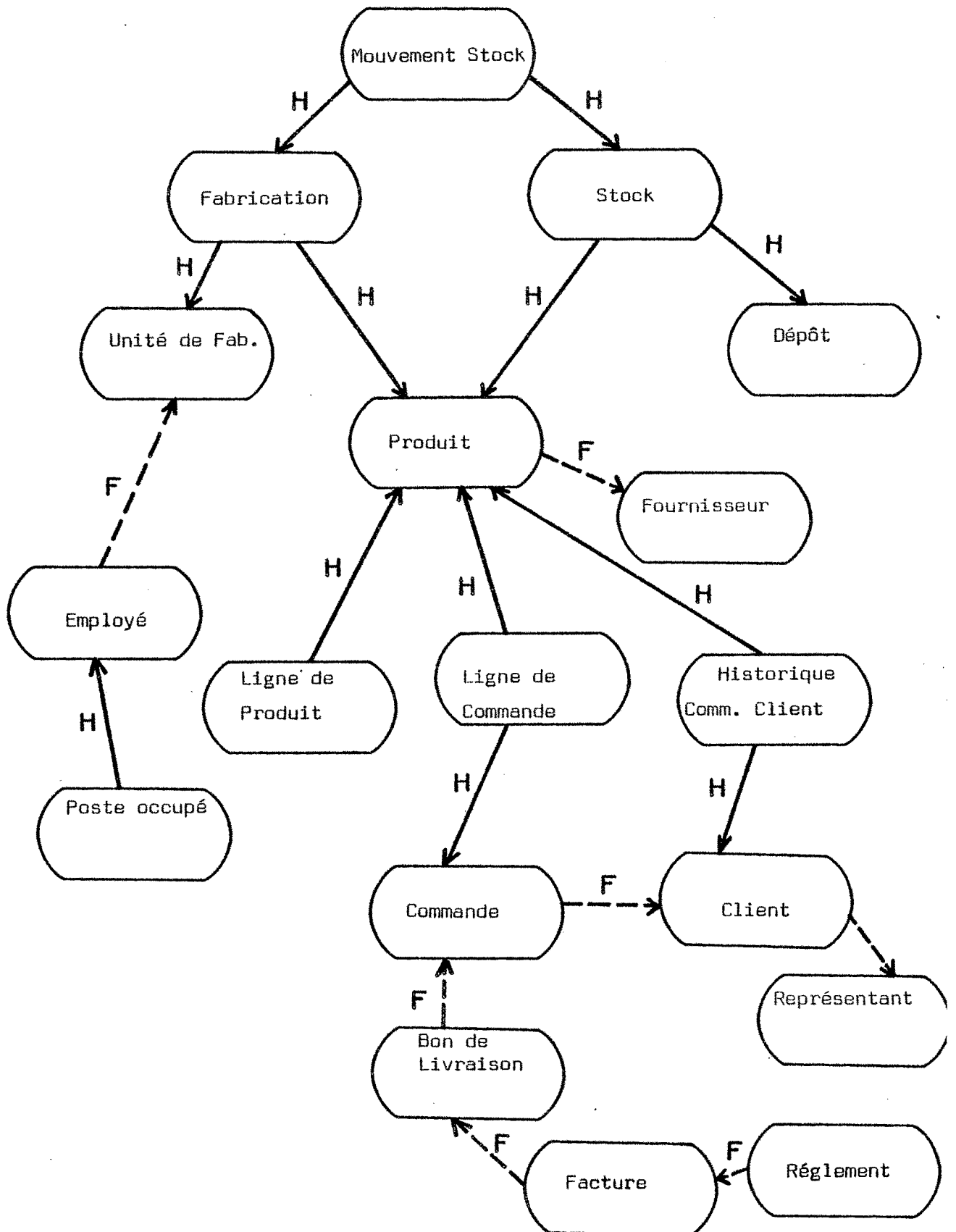


Dans ce cas $MARK\ 2(R_1) = 1$

On construit une liaison fonctionnelle de R_2 vers R_1 .

La même démarche est répétée pour chaque noeud jusqu'à l'obtention de toutes les liaisons fonctionnelles.

Voici la structure "SREL" obtenue sous forme de graphe



A un noeud nous avons mentionné le nom du groupe et à une arête le type de la liaison.

IV. 2 Deuxième Phase : La transformation "SREL" → "SREL R"

La transformation de " SREL" en "SREL R" correspond à l'étude d'une nouvelle structuration des informations permettant de passer de la structure canonique "SREL" vers une structure d'accès plus performante au plan de l'exploitation de données, mais conservant les acquis sémantiques du niveau conceptuel. En outre, elle doit permettre à la base de répondre facilement aux questions qui seront posées à la base lors de l'évolution des besoins en information et le changement des règles d'utilisation.

Le Multigraphe "SREL R" découlera de la structure "SREL" déjà construite, mais doit respecter les spécifications de la transformation.

Dans cette étape de notre démarche on peut distinguer trois types fondamentaux de modification du schéma :

- . l'insertion de nouvelles relations de nouveaux constituants et la construction des liaisons hiérarchiques et fonctionnelles non-prises en compte dans la structure "SREL".
- . Le changement de type d'association et la modification des liaisons construites, pour privilégier certains chemins d'accès.
- . La fusion de relations et leur modification structurelle.

Rappelons ici, que la notion de dépendances fonctionnelles a été l'outil conceptuel qui a permis jusqu'à présent la création d'une structure donnant une bonne présentation du monde réel. Il convient donc de préserver cet acquis en ne rajoutant que des relations fonctionnelles, puisque le changement du monde réel peut être interprété par un changement dans l'ensemble de dépendances fonctionnelles de départ et par l'introduction de nouveaux éléments dans le catalogue de constituants.

Ensuite nous allons essayer d'explorer davantage les concepts développés par l'outil Multigraphe et les fonctions associées pour définir :

- la façon de traiter l'intégration de nouveaux éléments de la structure
- la manière de répercuter les opérations de modification.

Le passage de la structure canonique "SREL" à la structure opérationnelle "SREL R" est essentiellement interactive pour affronter aux problèmes de synonymes et de polysémies.

IV. 2.1 Opérations de construction du Multigraphe : "SREL"

2.1.1 Construction de noeuds et des arêtes supplémentaires.

La création de nouvelles liaisons hiérarchiques et fonctionnelles ainsi que la construction de noeuds supplémentaires sont motivées par un changement de la réalité et par l'évolution constante des besoins recensés de par l'analyse des requêtes émises à la base.

Ceci montre la façon arbitraire dont un utilisateur décide à priori la nature de données.

Exemple :

Dans l'application précédente, on peut se trouver dans l'un des cas suivants :

a) l'utilisateur souhaite pouvoir connaître l'histoire du salaire d'un employé et non plus le salaire du mois en cours. Il exprime sa demande par l'introduction d'une nouvelle relation fonctionnelle "Paie", notée :

Paie : No-Insee, Date → Salaire

La prise en charge de cette nouvelle information provoque une modification dans l'ensemble d'information de départ. Face à de tel changement, le concepteur doit avant tout intégrer les nouveaux constituants concernés dans le dictionnaire de constituants et rajouter une nouvelle entrée dans la table de relations fonctionnelles.

Il cherche, ensuite, à construire le Multigraphe "SREL" en modifiant la partie correspondante de la structure "SREL" déjà obtenue.

Pour cela il doit :

1) Créer dans le graphe "SREL" un noeud supplémentaire de nom "Paie" auquel il attache la clé et l'ensemble de constituants associés à l'aide des fonctions :

NCREER

CCREER

et CRCLE

2) Construire une nouvelle liaison hiérarchique entre "Paie" et "Employé", ce qui donne naissance, dans le graphe "SREL" à une nouvelle arête de type (H) allant de "Paie" vers "Employé".

3) Supprimer du noeud "Employé" les constituants relatifs au salaire en se définissant une nouvelle fonction : "SUPCONS"

b) Si le concepteur s'aperçoit de par les questions posées à la base, que l'utilisateur souhaite très souvent pouvoir accéder à une "Unité-Fabrication" à partir de son lieu et non pas à partir de son nom. Dans ce

cas, le concepteur peut extraire de la relation "Unité-de-Fabrication" les informations concernant le lieu et créer dans le Multigraphe "SRELR" un nouveau noeud : "Lieu-de-Fabrication" auquel il attache les constituants extraits.

Ensuite il crée une arête de type (F) pour les associer.

2.1.2 Changement de type d'associations

Un tel changement est motivée par un changement dans le monde réel.

Exemple :

Dans le cas de notre exemple si la réalité a évolué et un employé peut travailler, selon les conditions de fabrication, pour plusieurs usines. Ceci indique que l'association entre "Employé" et "Unité-de-Fabrication" est réellement passée de type n:1 au type n:m.

Cette évolution s'exprime par l'introduction de la part de l'utilisateur d'une nouvelle relation fonctionnelle notée :

Emploi : no-Insee, No-Unité \rightarrow θ_1

Le concepteur à son tour modifie la table de relations fonctionnelles avant d'effectuer les changements nécessaires apportés à la structure "SREL". Pour cela il doit :

- 1) Créer dans le graphe "SRELR" un noeud supplémentaire de nom "Emploi" auquel il attache les constituants : No-Insee, No-Unité et θ_1 ainsi que la clé composée. (No-Insee, No-Unité)
- 2) Supprimer l'arête fonctionnelle entre "Employé" et "Unité-de-Fabrication" à l'aide de fonction "SUPARET".
- 3) Créer dans "SRELR" deux nouvelles arêtes de types (H) partant du noeud "Emploi", l'une allant vers le noeud "Unité-de-Fabrication" et l'autre vers le noeud "Employé".

2.1.3 Rattachement d'un noeud à un autre

Soit deux noeuds X_1 et X_2 et les relations fonctionnelles associées

rf1 : $K(R_1) \rightarrow \alpha_1$, $K(R_2)$ { α_1 et α_2 sont des constituants
rf2 : $K(R_2) \rightarrow \alpha_2$ { appartenant à aucune clé

On pourra si l'on veut raccourcir le chemin d'accès de X_1 à X_2 et rattacher le noeud X_2 au noeud X_1 .

Une telle transformation est intéressante si le noeud X'_1 obtenu est souvent demandé ou bien si X_2 est rarement mis-à-jour.

La relation associée à X'_1 est en 2 FN.

Cependant, elle permet de retrouver par projection les relations fonctionnelles irrédondantes. De plus elle ne risque pas de causer trop d'incohérence puisque X_2 est rarement mis-à-jour.

Exemple :

Si de par les statistiques concernant les fréquences d'accès à un noeud donné, le concepteur voit que les informations associées à un "Représentant" sont accédées par tous les traitements concernant un client et que de plus ces informations sont rarement mises-à-jour il peut décider de rattacher le noeud "Représentant" au noeud "Client".

Les modifications provoquées peuvent être effectuées dans "SRELR" en suivant les opérations ici indiquées:

- a) Rattachement des constituants de noeud X_2 au noeud X_1 . Le noeud X'_1 obtenu est formé de la réunion de constituants : $C^1 \cup C^2$.
Pour que cette opération ait un sens il s'agit maintenant de construire l'ensemble de relations fonctionnelles élémentaires :
 $rf'1 = rf1 \cup rf2$.
Cet ensemble peut donner naissance à des relations non-directes qu'il faudra ensuite éliminer pour continuer à préserver la propriété de non-redondance sémantique du niveau conceptuel.
- b) Toutes les arêtes partant du noeud X_2 doivent maintenant partir du noeud X'_1 .
- c) Toutes les arêtes arrivant au noeud X_2 doivent arriver au noeud X'_1 .
- d) Quand toutes ces opérations ont été effectuées le noeud X_2 est supprimé.

Ces modifications sont effectuées directement à l'aide de l'outil Multigraphe.

Pour le faire, le concepteur peut utiliser la fonction HIERARC afin de connaître les arêtes partant du noeud X_2 et la fonction ACREER pour les reconstruire à partir du noeud X'_1 .

Nous avons réalisé une nouvelle fonction : AREER qui donne toutes les arêtes arrivant à un noeud donné. Pour chaque arête elle retournera le type de l'arête et le numéro interne de son noeud de départ.

Le concepteur peut, ensuite, utiliser la fonction ACREER pour recréer ces arêtes avec comme noeud d'arrivée X'_1 .

La fonction SUPNOEU permettra de rendre inactif le noeud X_2 , celui-ci devient inaccessible par l'utilisateur.

Conclusion :

La transformation de "SREL" en "SRELR" permet à la base de ne pas avoir une structure figée et d'aménager un accès plus facile à des demandes diverses.

Elle repose sur les résultats de l'analyse de traitements décrits par les différentes requêtes, qui permettent de clarifier les facteurs suivants :

- la fréquence des accès aux différents éléments de la structure NOEUD, CONSTITUANT, ARETE et leur volume de mis-à-jour.
- la fréquence des accès par structure d'accès
- le temps de réponse prévu
- le mode de fonctionnement de la base

Les traitements peuvent être considérés soit comme un ensemble de constituants à connaître simultanément, soit comme des chemins d'accès à traverser. Pour chaque traitement la fréquence des accès à un élément de la structure pourra être définie comme le nombre de réalisations consultées ou mis-à-jour pour satisfaire au traitement et ceci par rapport à une période de référence donnée. La fréquence d'accès par structure d'accès déduite de la fréquence de traitement permet au concepteur d'étudier une restructuration des données sous l'angle des accès. Cependant l'optimisation de la structure d'accès doit se faire selon des règles strictes garantissant la propriété de non-redondance sémantique du niveau conceptuel.

IV. 3 Troisième Phase : La transformation de "SRELR" en "SRES"

Cette transformation doit nous conduire à une structure réseau : "SRES" qui conforme à la définition donnée dans le chapitre III. Elle doit se faire sans transfert d'informations externes mais par simple modification de "SRELR" grâce aux concepts développés à propos de l'outil Multigraphe.

IV. 3.1 Opérations de transformation : "SRELR" → "SRES"

- a) Destruction de la redondance des constituants.
- b) Création des liaisons binaires, qui n'existent pas explicitement dans "SRELR"
- c) Choisir la représentation de toutes les liaisons hiérarchiques.
En effet, il existe une correspondance biunivoque entre une structure relationnelle et une classe de structure réseau. Pour nous, deux structures réseaux appartiennent à la même classe, si

et seulement si elles ne diffèrent que par la représentation des liaisons hiérarchiques. Il s'agit donc au cours de cette étape de déterminer une structure parmi cette classe, qui sera représentée par le Multigraphe "SRES".

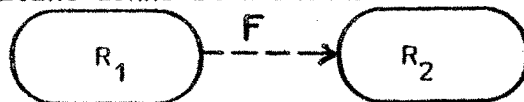
Dans ce qui suit, nous allons voir comment ces trois opérations peuvent être effectuées directement à l'aide du Modèle Multigraphe.

3.1.1 Opération de destruction de la redondance des constituants

Contrairement aux noeuds associés à la structure "SREL" ou à la structure "SREL", un noeud de "SRES" peut ne pas contenir tous les constituants qui lui sont associés, en particulier, ceux qui appartiennent à sa clé.

a) Cas d'une liaison fonctionnelle

Etant donné la structure "SREL" suivante :



avec comme relations fonctionnelles associées

$$\begin{array}{l} \text{rf1 : } K(R_1) \rightarrow K(R_2), \alpha_1 \\ \text{rf2 : } K(R_2) \rightarrow \alpha_2 \end{array} \quad \left\{ \begin{array}{l} \alpha_1 \text{ et } \alpha_2 \text{ sont des constituants} \\ \text{non-clé de } C \end{array} \right.$$

On peut remarquer, que les constituants composant la clé $K(R_2)$ du noeud R_2 forment une partie redondante de l'ensemble C^1 de constituants associés au noeud R_1 .

b) Cas d'une liaison hiérarchique

Etant donné la structure "SREL" suivante :



avec comme relations fonctionnelles associées

$$\begin{array}{l} \text{rf1 : } K(R_1) \rightarrow \alpha_1 \\ \text{rf2 : } K(R_2) \rightarrow \alpha_2 \end{array} \quad \left\{ \begin{array}{l} \alpha_1 \text{ et } \alpha_2 \text{ sont des constituants} \\ \text{non-clé de } C \end{array} \right.$$

et tel que $K(R_2) \subset K(R_1)$

Dans ce cas les constituants composant la clé de R_2 sont redondants dans R_1

Il s'agit lors du passage de "SREL" à "SRES" :

- Soit de supprimer l'arête fonctionnelle ou hiérarchique entre les deux blocs concernés et de garder la redondance de constituants. Dans ce cas la fonction "SUPARET" permettra de supprimer cette liaison. La fonction SUPCONS supprimera de R_1 les constituants appartenant à C^1 n: $K(R_2)$.

La redondance de constituants peut être tolérée dans certains cas pour améliorer les accès, si la fréquence importante d'utilisation simultanée de ces constituants a été soulignée par les traitements. Cependant, elle pose un problème délicat, car dans "SRES" un même constituant ne peut figurer dans deux blocs simultanément. Dans ce cas, le système automatique doit distinguer les deux constituants et fournir au concepteur au même temps que la structure une sémantique associée dans laquelle il faudra lui indiquer les constituants synonymes.

- Soit de conserver la liaison hiérarchique ou fonctionnelle et d'éliminer la redondance de constituants. Dans ce cas, la fonction SUPCONS permettra de supprimer ces constituants. Le choix des redondances, que l'on pourra éventuellement tolérer dans la structure SRES dépendra des statistiques sur les fréquences d'accès. Celles-ci ne seront disponibles qu'après une analyse de requêtes émises à la base lors de son exploitation. C'est donc la raison pour laquelle nous avons choisi pour le moment de supprimer toute la redondance de constituants et de préserver les liaisons.

c) Autre cas

Un constituant peut appartenir à plusieurs noeuds sous d'autres conditions que celles citées dans les cas (a) et (b)

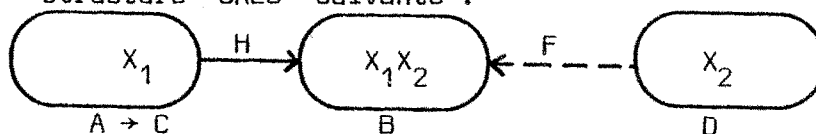
Exemple :

Etant donné deux noeuds X_1 et X_2 et les relations fonctionnelles associées :

rf1 : A, B → C

rf2 : D → B

Dans ce cas, le constituant redondant B peut être mis en commun par la création dans SRES d'un noeud supplémentaire ne contenant que lui. Tous les noeuds les contenant seront reliés à celui-ci par une liaison hiérarchique ou fonctionnelle, ce qui donne la structure "SRES" suivante :



3.1.2 Opération de création de liaisons binaires

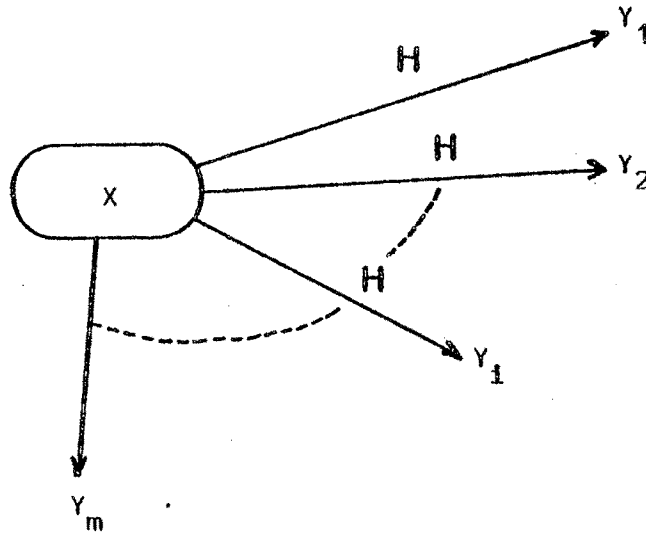
Etant donné un noeud X et un ensemble de noeuds : $\{ Y_1, Y_2, \dots, Y_m \}$
d'une structure "SREL" tels que :

- la relation associée au noeud X soit de la forme :

$$rf : K(X) \rightarrow \Theta_1$$

- les blocs : $\{ Y_1, \dots, Y_m \}$ soient tels qu'il existe une liaison hiérarchique allant du noeud vers chacun des noeuds Y_i .

Soit le schéma "SREL" suivant :



avec comme relations fonctionnelles associées :

$$rf_1 : K(X) \rightarrow \Theta_1$$

$$rf_2 : K(Y_1) \rightarrow \alpha_1$$

$$rf_3 : K(Y_2) \rightarrow \alpha_2$$

$$:$$

$$:$$

$$rf_i : K(Y_i) \rightarrow \alpha_i$$

$$:$$

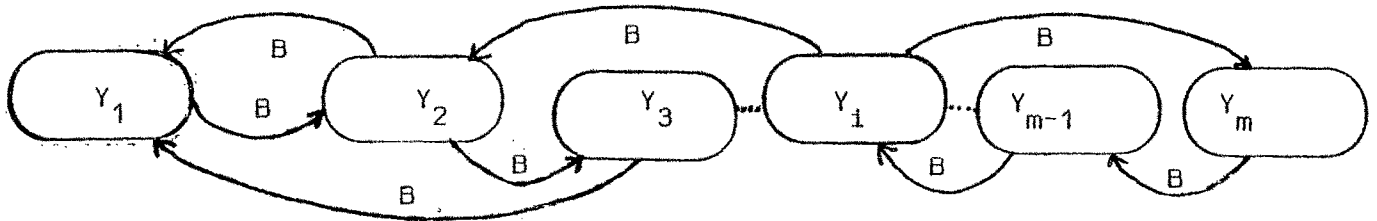
$$:$$

$$rf_m : K(Y_m) \rightarrow \alpha_m$$

et tel que :

$$K(X) = K(Y_1) \cup K(Y_2) \cup \dots \cup K(Y_i) \dots \cup K(Y_m)$$

Nous pourrions la transformer en une structure réseau "SRES" tel que deux noeuds Y_i et Y_j soient reliés par un chemin de liaisons binaires.



Ce chemin peut être déterminé à partir des questions qui seront posées à la base de données lors de son exploitation.

Exemple :

Si à la structure "SREL" les relations fonctionnelles suivantes sont associées :

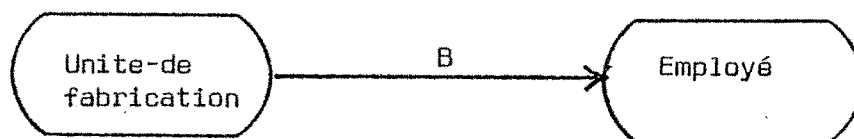
rf1 : No-Unité → NOM Unité,.....

rf2 : No-Insee → NOM Employé,.....

et que l'on peut de plus déduire de l'ensemble de relations associées à la structure "SREL" la relation : EMPLOI

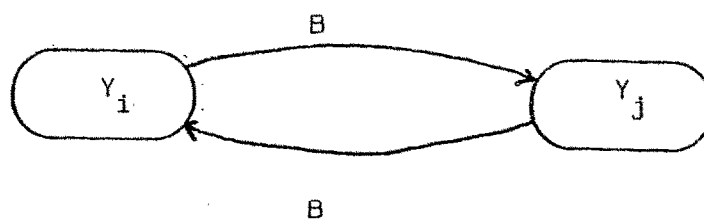
EMPLOI : No-Unité, No-Insee → Θ_i

Il pourra être intéressant, dans le cas où les utilisateurs s'intéressent plus souvent aux employés associés à une Unité-de-fabrication qu'aux Unité-de-fabrication dans lesquelles un employé travaille, d'adopter la structure réseau SRES suivante :



Or, en vue d'une automatisation complète de cette étape de la transformation nous avons choisi de prendre en compte sur SRES tous les chemins d'accès à un noeud donné.

Ainsi pour ne privilégier aucune des liaisons entre blocs nous construirons des liaisons binaires entre chaque couple de l'ensemble Y de la manière suivante :



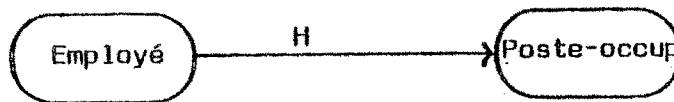
Le noeud X est supprimé. Ces opérations peuvent être effectuées à l'aide des fonctions ACREER et SUPNOEU.

3.1.3 Opération de détermination de liaisons hiérarchiques

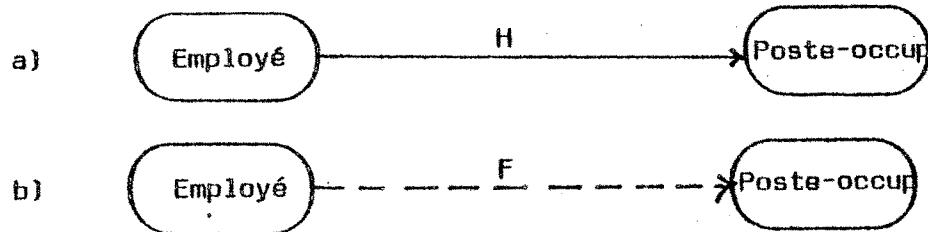
Contrairement à ce qui se passe dans une structure relationnelle, le concept d'une liaison fonctionnelle est plus étendu dans la structure réseau : "SRES" ou une liaison hiérarchique apparaît comme un cas spécial d'une liaison fonctionnelle.

Exemple :

Soit la structure "SREL" suivante :



Cette structure pourra être représentée au niveau de "SRES" de manière équivalente par l'un des deux schémas suivants :



Ainsi, si dans la structure "SREL", on a une liaison hiérarchique entre deux noeuds X_1 et X_2 et si de plus la relation fonctionnelle associée au noeud X_1 n'est pas de la forme $K(X_1) \rightarrow \theta_1$, le choix de la représenter dans "SRES" par une liaison fonctionnelle ou hiérarchique peut être laissé au concepteur.

Ceci se fera selon que si le concepteur veut conditionner l'existence d'une réalisation de X_1 , à une réalisation de X_2 ou non.

Exemple :

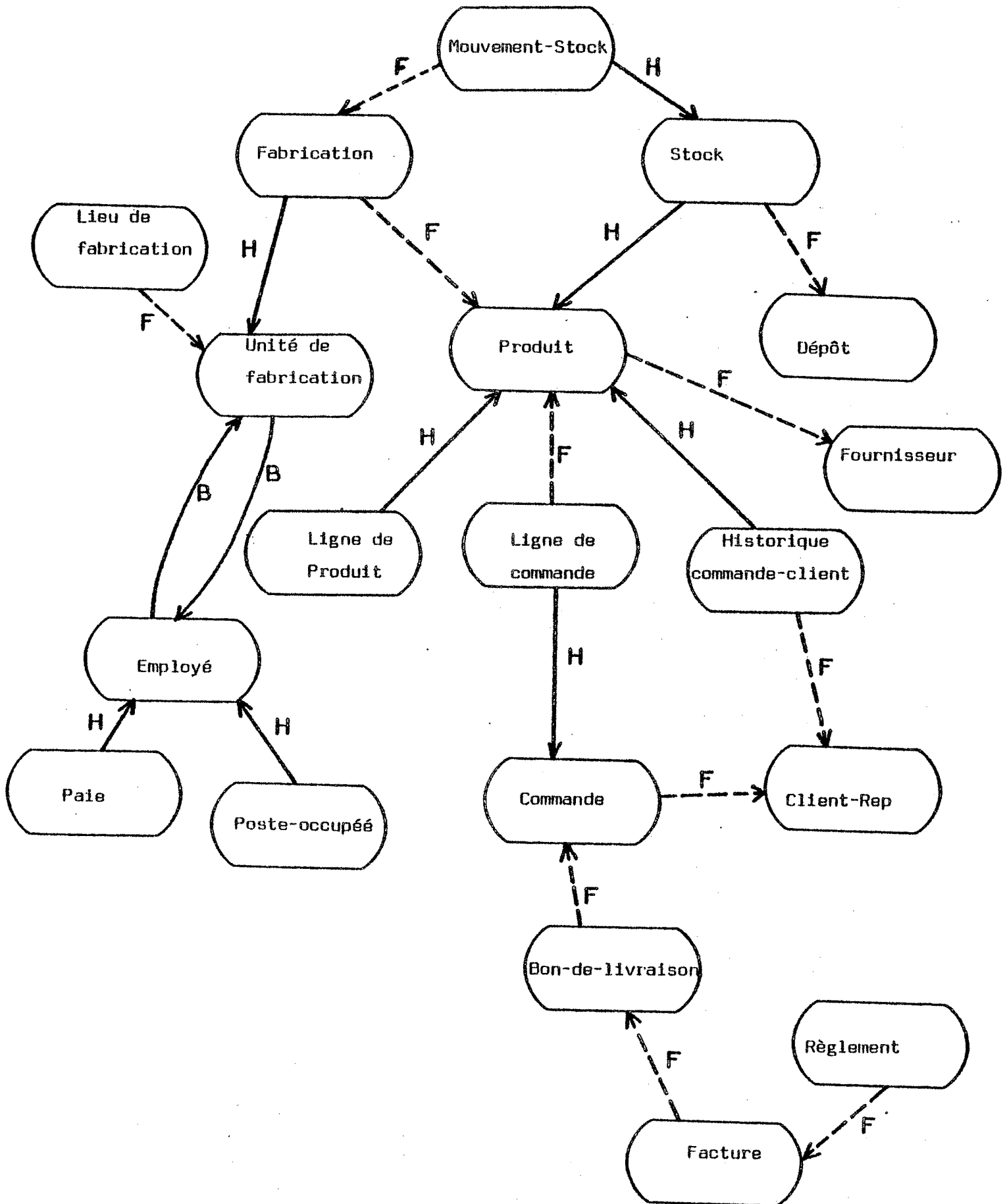
Dans notre exemple, le concepteur peut décider en passant à la structure "SRES" de conserver la liaison hiérarchique allant de "Poste-occupé" vers "Employé", cela conditionnera l'existence d'un poste à celle de son employé. Celle-ci disparaîtra si l'employé quitte son travail, ce qui n'est peut être pas souhaitable si le noeud Poste-occupé est relié à d'autres noeuds de la structure.

En vue d'une automatisation totale de cette étape de transformation il nous semble souhaitable de réduire l'intervention du concepteur ou des utilisateurs en intégrant ces critères de choix dans notre processus :

- a) SI d'un noeud donné part une seule liaison hiérarchique celle-ci est conservée.
- b) Si d'un noeud donné partent plusieurs liaisons hiérarchiques, la première dans la liste est privilégiée. Elle sera représentée dans SRES par une liaison hiérarchique. Les autres sont toutes transférées en des liaisons fonctionnelles.

Cette opération peut être réalisée à l'aide de la fonction CATYP de modification de type d'une arête.

Le Multigraphe correspondant à la structure "SRES" obtenue, peut être illustré à l'aide du schéma suivant :



IV. 4 Quatrième Phase : La transformation de "SRES" en une structure réseau en termes de S.G.B.D. actuels

C'est la dernière étape du processus de transformation pour obtenir une structure réseau en termes de SOCRATE ou CODASYL à partir d'un ensemble de relations. Elle consiste à traduire la structure indépendante "SRES" en termes d'un L.D.D. particulier. Cette traduction peut être faite selon le tableau donné en paragraphe (III. 3) par un simple interface.

Cependant, à une structure "SRES" il correspond par construction même une classe de structure SOCRATE ou CODASYL. Les différences entre deux structures d'une même classe ne sont dues qu'aux différences dans la représentation possible d'un même type de liaison.

Exemple 1 :

Dans le cas de SOCRATE pour la représentation d'une liaison binaire entre deux blocs X_1 et X_2 allant de X_1 vers X_2 , le concepteur aura le choix entre un INVERSE et une ENTITE-REFERENCE.

Ce choix repose, dans le cas d'une solution opérationnelle, sur des critères de masse d'information. En effet si le nombre de réalisations du bloc X_2 inversées par une réalisation du bloc X_1 est peu important il sera préférable d'utiliser une ENTITE-REFERENCE pour représenter cette liaison.

Pour aider le concepteur dans son choix, l'utilisateur doit fournir des renseignements sur le nombre maxi d'uplets que pourra comporter chaque relation lors de l'introduction de données.

Dans le cas de CODASYL le concepteur doit, également, décider sur la représentation des liaisons hiérarchiques qui peut être effectuée soit par un "LIEN-HIERARCHIQUE" soit par un "GROUPE-REPETITIF". Là aussi la décision dépendra des critères de masse d'information.

Exemple 2 :

Pour SOCRATE il y a aussi le choix de l'orientation d'une liaison. En effet une liaison SOCRATE entre deux blocs désignés X_1 et X_2 allant de X_1 vers X_2 correspond à un chemin d'accès non symétrique qui permet d'atteindre X_2 à partir de X_1 et non pas l'inverse.

Le choix du sens d'une liaison SOCRATE, en particulier, dans le cas d'une liaison binaire dépendra essentiellement de la fréquence d'utilisation de cette liaison et de son inverse, indiquée par les différents traitements.

Exemple :

En faisant toujours appel à notre exemple de référence, nous considérons d'abord le passage à la structure SOCRATE et illustrons ici la structure obtenue.

STRUCTURE SOCRATE

Entité 10 Unité-de-Fabrication

Début

No-Unité MOT DISCR
Nom-Unité MOT
Description MOT
Nb-Employé DE 0 A 100
Employés INVERSE toutes Employé

Entité 10 000 Fabrication

Début

Produit-fabriqué REFERENCE un Produit
Qté-en-cours DE 0 A 500
Capacité-Mini DE 0 A 250
Capacité-Maxi DE 0 A 1000
Produit-utilisé REFERENCE un Mouvement-Stock de
un Stock de un Produit

Fin

Fin

Entité 5 Dépôt

Début

No-Dépôt MOT DISCR
Nom-Dépôt MOT
Lieu MOT
Surface DE 0 A 100000

Fin

Entité 10000 Produit

Début

No-Produit MOT DISCR
Libellé MOT
Désignation MOT
Stade-d'élaboration LVAL (Fini, Semi-fini) 2 10
Prix-unitaire DE 0 A 999
Fournition REFERENCE un Fournisseur

Entité 10000 Stock

Début

Lieu-de-stockage REFERENCE un Dépôt

Entité 1000 Mouvement-Stock

Début

Date-Mouvement DE 0 A 999999

Code-Mouvement LVAL (E S) 2 1

Qté-Mouvement DE 0 A 1000

Fin

Qté-disponible

Entité 1000 Historique-Commande-Client

Début

Historique-Client REFERENCE un client

Qté-vendue

Taux-de-remise

Fin

Entité 100 Ligne-de-Produit

Début

No-Produit Composant MOT

Qté-nécessaire

Fin

Fin

Fin

Entité 1000 Commande

Début

No-Commande MOT DISCR

Date-Commande DE 0 A 999999

Etat-Commande LVAL (D ND) 2 2

Mode-d'expédition MOT

Clients REFERENCE un Client

Entité 100 Ligne-de-commande

Début

Produit commandé REFERENCE un Produit

Qté-demandée

Fin

Fin

Entité 1000 Client

Début

No-Client MOT DISCR

Nom-Client MOT

Adresse-Client MOT

Début

Nom-Représentant MOT

Secteur MOT

Montant-Vendu-Mois DE 0 A 9999

Fin

Fin

Entité 1000 Règlement

Début

No-Règlement MOT DISCR

Montant

Date-de-Règlement

Facturation REFERENCE une Facture

Fin

Entité 1000 Facture

Début

No-Facture MOT DISCR

Date-facture DE 0 A 999999

Livraison REFERENCE un bon-de-livraison

Fin

Entité 1000 Bon-de-Livraison

Début

No-bon-de-livraison MOT

Date Sortie

No-Ref REFERENCE une Commande

Fin

Entité 10000 Employé

Début

No-Insee MOT DISCR
Nom-Employé MOT
Sexe LVAL (M F) 2 1
Résidence MOT
Age DE 0 A 65
Grade MOT

Entité 10 Travail

Début

LIEU-de-Travail REFERENCE une Unité-de-Fabrication

Fin

Entité 5 Poste-Occupé

Début

No-Poste
Fonction
Date arrivée
Date départ

Fin

Entité 100 Raie

Début

Date

Début

Nb-heures
taux

Fin

Fin

Fin

Entité 10 Lieu de Fabrication

Début

Nom de Lieu REFERENCE une Unité-de-Fabrication
Adresse MOT

Fin

Etant donné qu'une structure SOCRATE et une structure CODASYL
appartient à une même classe de structure réseau la transformation :

"SRES" → Structure CODASYL ne posera aucun problème

CHAPITRE V

CONCLUSION

CONCLUSION

Dans ce chapitre de conclusion, nous rappelons brièvement les objectifs poursuivis et les prolongements possibles de notre travail et son adaptation envisageable.

Nos objectifs en commençant ce travail étaient :

- d'explorer davantage le modèle relationnel, en particulier, celui de CODD, qui donnait une représentation de la sémantique du monde réel et facilitait son appréhension mais qui avait été conçu indépendamment des chemins d'accès inévitables dans un passage à l'automatisation.
- de faciliter le passage au modèle réseau comme modèle d'implantation logique.

Notre travail se situe donc au niveau conceptuel. Il consiste à mettre au point une méthode et des outils qui aideront à la construction successive de la structure conceptuelle d'un Système d'Information.

La démarche, que nous avons adoptée, est une démarche ascendante, dont l'intérêt est d'aborder la sémantique des liaisons entre les différents constituants de la base pour construire convenablement une structure réseau dépourvue de redondances, en termes de Systèmes de Bases de Données actuels.

En effet, ces liaisons existent, soit par conception, soit par la façon dont apparaissent les données ; il s'agit donc de leur donner une réalité en les explicitant rigoureusement.

Ainsi, elles serviront de trame à la construction effective d'une structure de la base. Pour celles qui s'expriment sous forme de relations fonctionnelles, il est important de construire un ensemble minimal de liaisons qui recouvrent toutes les relations que la structure doit prendre en compte. Ceci permettra d'éliminer les incohérences et de voir les redondances qui peuvent s'infiltrer au niveau du modèle.

A ce stade de notre travail la représentation du monde réel reste, encore imparfaite dans la mesure où les règles d'évolution qui traduisent la dynamique du Système d'Information ne sont pas représentées.

C'est donc dans une étape ultérieure qu'il s'agira de prendre en compte :

- de nouvelles règles d'utilisation des informations,
- de nouvelles données venant perturber les règles précédemment admises et ceci en conservant un maximum des acquis déjà obtenus.
- de considérer d'autres résultats de l'analyse du Système d'Information relatifs aux chemins d'accès, aux mécanismes de mises-à-jour et de destruction de données,

avant de passer à la structure réseau finale. Celle-ci peut comporter un minimum de redondance mais doit préserver la cohérence des informations.

Ce processus de transformation de structure de données permettra au concepteur de passer des aspects purement logiques de conception à la construction d'une structure effective en termes S.G.B.D. actuels. Ainsi pourra-t-il prendre en compte une plus grande hétérogénéité des informations et aménager un accès plus facile à des demandes diverses.

Le logiciel que nous avons présenté ici, repose sur l'implantation d'un Multigraphe orienté : $G = [C, \pi]$

où C est l'ensemble de noms de constituants

$\pi \subset C \times C \times \lambda$ est une relation définie par
(A,B, rfi)

Si A et B sont des constituants figurant dans la relation fonctionnelle élémentaire rfi avec A dans la partie gauche et B dans la partie droite.

λ étant l'ensemble de relations fonctionnelles élémentaires :

rf1, rf2, ..., rfm

Nous sommes parfaitement conscients que certains problèmes sont loin d'être résolus, d'autres sont seulement évoqués et peuvent faire l'objet de travaux ultérieurs, d'autres enfin, ne sont même pas abordés dans l'état actuel du développement de ce travail. De plus, la recherche que nous présentons ici pour être menée à bien devrait se dérouler sur plusieurs applications. C'est la raison pour laquelle cette thèse essaie plus de définir le cadre d'une solution possible aux problèmes évoqués que de définir d'une manière détaillée une façon de faire.

Nous pensons toutefois, que la démarche proposée répond en grand partie à nos objectifs. Elle constitue un cadre convenant bien à la construction des Systèmes d'Information évolutifs. Le système présenté peut bien à l'égard être considéré comme une première étape vers des développements ultérieurs plus importants. Ceci tient aux techniques mises en oeuvre :

- l'utilisation des concepts relationnels
- une décomposition de la transformation
- une conception ouverte et segmentée de système correspondant aux algorithmes suffisamment modulaires et généraux pour s'adapter à de nombreux cas pratiques
- un ensemble de fonctions capables de réaliser des opérations élémentaires au niveau de chaque sous transformation.

Nous rappelons que ces fonctions sont de deux types :

- a) fonctions d'introduction des catégories : NOEUD, CONSTITUANT, ARETE et CLE,
- b) fonctions de modification du schéma de la structure.

Nous avons montré en quoi la réalisation des ces modifications dépend des questions que les utilisateurs poseront à la base de données. L'aide de l'utilisateur est alors indispensable, en lui laissant dans chaque cas la responsabilité d'accepter ou non les solutions proposées.

Ce travail pourra être développé dans deux directions :

- la prise en compte de paramètres quantitatifs pour la détermination des structures d'accès et des mesures de performance.
- De partir des questions qui forment l'ensemble des demandes que devra satisfaire la base pour déterminer progressivement la structure finale de la base par intégration successive des structures correspondant à chaque classe de requêtes. Dans ce cas, les requêtes correspondent à des actions ponctuelles sur la base.

Pour aborder un tel problème il peut être intéressant de le décomposer de la manière suivante :

- sous quelle forme prendre les questions,
- quels sont les éléments relationnels d'une question
- quels sont les éléments de la structure concernés par une question et quel est le rôle de ses éléments dans la question
- à partir de quel moment une modification du schéma de la structure sera-t-elle effectivement exécutée.

Pour en arriver là, il faut disposer d'un analyseur de requêtes et d'un interpréteur pour réaliser l'interface avec le schéma global. Les règles de gestion existant dans la méta base peuvent être utilisées pour une génération quasi automatique de la logique de traitement.

BIBLIOGRAPHIE

- [Abr 72] ABRIAL, J.R. et al,
Projet SOCRATE
U.S.M.G., Grenoble, sept. 72
- [Abr 74] ABRIAL, J.R
" Data Semantics "
IFIP TC2 Working Conference,
Cargèse, avril 74
- [ABU 77] AHO A.V, BEERI C., ULLMAN J.D.
" The theory of Joins in Relational Data bases "
Proc. 10th Symposium on Foundations of Computer Science, oct 77
- [AC 78] ARORA A., CARLSON R.
" The information preserving properties of relational
data base transformations "
Proc IV VLDB, Berlin sept 78, pp 352-359
- [AD 77] ARMSTRONG W., DELOBEL C.
" Décompositions and Functional Dependencies in Relations "
Département d'Informatique et Recherche Opérationnelle,
Université de Montréal, oct 77. (submit ACM-TODS)
- [Ad1 75] ADIBA M.,
"Transformation d'une structure SOCRATE en un ensemble
de relation ".
Note technique, IMAG, mars 75
- [Ad1D 76] ADIBA M., DELOBEL C.,
" Les modèles relationnels de Bases de Données ".
Cour IRIA, avril 76
- [AL 75] ADIBA M., LEONARD M.,
TS1 Présentation d'une structure SOCRATE en termes de structure
relationnelle
TS2 Présentation d'une structure CODASYL en termes de structure
relationnelle
Notes techniques, IMAG, juin 75

- [ALD 76] ADIBA M., LEONARD M., DELOBEL C.,
" A unified approach for modeling data in logical database design "
IFIP TC2, Freudenstadt, janvier 76
- [Ant 76] ANTON J.P. et al.,
" Un système de conception automatique et progressive d'un système d'information ". Congrès AFCET, novembre 76
- [Arm 74] ARMSTRONG W.W.,
" Dependency Structures of Data Base Relationships "
Proc IFIP 1974, pp 580-583, North Holland, 1974
- [Art 78] ARTIGUE-ASSIE S.,
" Implantation logique d'une base de données ".
Thèse 3ème Cycle, Université Claude Bernard-Lyon 1, septembre 78
- [Ast 76] ASTRAHAM M. et al.,
" System R : Relational approach to Data Base management ".
ACM TODS, vol.1, n°2, juin 76
- [BB 76] BERNSTEIN P.A., BEERI C.,
" An Algorithmic approach to Normalisation of Relational Data Base schemas ".
Tech. Report CSRG-73, Computer Science Research Group,
University of Toronto, sept 76
- [BB 77] BEERI C., BERNSTEIN P.A.,
" Computational problems related to the design of normal forms for relational schemas ". ACM TODS, vol.4, n°1, mars 79
- [BBBC 76] BENCI E., BODART F., BOGARD H., CABANES A.,
" Concepts for the design of a conceptuel schema ".
Modelling in data base management systems.
G.M. Nijssen, North Holland publ. 1976, pp 181-200
- [BBG 78] BEERI C., BERNSTEIN P.A., GOODMAN N.,
" A sophisticate's Introduction to Data Base Normalization theory
Proc VLDB Conf., Berlin, sept 78, pp 113-124

- [Bee 79] BEERI C.,
" On the role of data dependencies in the construction of
Relational data base Schemas "
Report n° 43, Hebrew University of Jerusalem, janvier 79
- [Ber 76] BERNSTEIN P.A.,
" Synthesizing third normal form relations from functional
dependencies "
ACM TODS 1, 4, pp 277-298, décembre 76
- [Ber 77] BERNSTEIN P.A.,
" Normalization and functional dependencies in the relational
data base model "
Ph.D.Thesis, Department of Computer Science,
University of Toronto, 1975
- [BD 76] BASTARAUD B., DEVY M.,
" Transformations automatiques de structures "
Projet de 3^{ème} Année - ENSIMAG, juin 76
- [Bub 76] BUBINKO J.A. et al.,
" From information requirements to DBTG structures "
Proc. Conference on Data Abstraction, Definition and structure
1976
- [BFH 77] BEERI C., FAGIN R., HOWARD J.H.,
" A computer axiomatization for functional and multivalued
dependencies in Data base relations "
IBM Tech. Rep. RJ 1977, San José, Ca., (also Proc. ACM SIGMOD
Conf., Toronto, août 77, pp 47-61
- [BMSU 78] BEERI C., MENDELZON A., SAGIN Y., ULLMAN J.,
" Equivalence of relational data base schemas "
Princeton University, Decembre 78, to appear in ACM Symp.
on Theory of Computing.
- [BST 75] BERNSTEIN P.A., SWENSON J.R., TSICHRITZIS D.C.,
" A unified approach to Functional Dependencies and Relations "
Proc. ACM SIGMOD Intl. Conf. on Management of Data,
San José, mai 75, pp 237-245

- [BRK 74] BATNI R.P., RUSSEL J.D., KIME C.R.,
" An efficient algorithm for finding an Irredundant Set Cover "
JACM 21,3, pp 351-355, juillet 74
- [BS 78] BANCILHON F., SPYRATOS N.,
" Update semantics of relational views "
Rapport IRIA - Lab., Le Chesnay, France, septembre 78
- [Cab 76] CABANES A.,
" Independance logique dans les systèmes de gestion de bases
de données "
Rapport contrat IRIA, 1976
- [Cad 76] CADIOU J.M.,
" On semantic issues in the relational Model of Data "
Proc. Intern. Symp. on Mathematical Foundations of Computer
Science, Gdansk, Poland septembre 75, Springer-Verlag Lecture
Notes in Computer Science
- [Che 76] CHEN P.S.,
" The Entity Relationship Model towards a unified view of Data "
ACM Trans. of Data Base Systems, vol. 1, n°1, mars 76, pp 9-36
- [Cod 70] CODD E.F.,
" A relational Model of Data for large shared Databanks "
CACM 13, 6, juin 70, pp 377-387
- [Cod 71] CODD E.F.,
" Normalized Data Base structure : A brief tutorial "
ACM SIGFIDET Workingshop on data description access and control,
SanDiago, novembre 71
- [Cod 72] CODD E.F.,
" Further normalizations of Data base Relational Model "
in Data Base systems (R. Rustin ed.), Prentice Hall, Englewood
cliffs, N.J., 1972, pp 33-64
- [Cod 73] CODD E.F.,
" Recent Investigation in Relational Data Base Systems "
Proc. of IFIP 1974, North Holland

- [CSL 71] CODASYL
" DATA BASE TASK GROUP Report "
ACM 10, New York, 1971
- [Dat 77] DATE C.J.,
" An introduction to data base systems "
Second edition, Addison Wesley Publ. Co., 1971
- [DB 78] DAYAL U., BERNSTEIN P.A.,
" On the updatability of relational views "
Proc. IV VLDB, Berlin, septembre 78, pp 368-377
- [DC 73] DELOBEL C., CASEY R.G.,
" Decomposition of a Database and the Theory of Boolean
Switching Functions "
IBM J. Res. Develop. 17, 5, pp 374-386, septembre 73
- [De1 71] DELOBEL C.,
" Aspects théoriques sur la structure de l'information dans
une base de données "
RAIRO, B.3, 1971, pp 37-64
- [De1 72] DELOBEL C.,
" A theory about Data in an information System "
IBM Research Report RJ 964, San José, 1972
- [De1 73] DELOBEL C.,
" Contribution théoriques à la conception et l'évolution d'un
système d'information appliqué à la gestion "
Thèse d'Etat, Université de Grenoble, octobre 73
- [De1 75] DELOBEL C.,
" Les systèmes de Bases de Données "
Ecole d'été de l'AFCEP, 1975
- [De1 76] DELOBEL C.,
" Semantique des relations et processus de normalisation
dans le modèle relationnel "
Rapport de Recherche n° 45, IMAG, 1976 septembre

- [De1 78] DELOBEL C.,
" Normalization and Hierarchical Dependencies in the relational
Data Model "
ACM TODS 3, 3, pp 201-222, septembre 78
- [De1 79] DELOBEL C.,
" An overview on the relational Data Theory "
1979
- [DL 74] DELOBEL C., LEONARD M.,
" The decomposition process in a relational Model "
Intern. Workingshop on Data structures, IRIA, Namur (Belgique)
mai 74
- [DP 78] DELOBEL C., PARKER D.S.,
" Functional and Multivalued dependencies in a relational
Database and the Theory of Boolean Switching Functions "
Tech. Rep. n°142, Dept. de Math. Appl. et d'Informatique,
Université de Grenoble, novembre 78
- [DP1 78a] DELOBEL C., PICHAT E.,
" Application de l'algèbre aux modèles de données relationnels "
1^{er} Colloque AFCET-SMF de Math. Appl., 4-8 septembre 78,
Palaiseau (France), tome II, pp 301-312
- [DP1 78b] DELOBEL C., PICHAT E.,
" The disign of relational information system according to
different kind of dependencies "
Congrés ECI 78, Venise, octobre 78
- [Fag 77a] FAGIN R.,
" Functional dependencies in a relational Data Base and
Propositional Logic "
IBM J. Res. Develop. 21, 6, pp 534-544, novembre 77
- [Fag 77b] FAGIN R.,
" Multivalued Dependencies and a new Normal form for Relational
Data bases "
ACM TODS 2, 3, pp 262-278, septembre 77

- [Fag 77c] FAGIN R.,
" The Decomposition versus the Synthetic Approach to
Relational Database Design "
IBM Tech. Report RJ 1976, San José, CA, 77
- [Fag 79a] FAGIN R.,
" Normal Forms and Relational Database operators "
Proc. ACM SIGMOD, 1979
- [Fag 79b] FAGIN R.,
" A normal form for relational database that is based on
domains and keys "
IBM Research Report RJ 2520, mai 79
- [Flo 77] FLORY A.,
" Un modèle et une méthode pour la conception logique d'une
base de données "
Thèse d'Etat, Université de Lyon, novembre 77
- [FM 77] FLORY A., MOULIN M.,
" L'utilisation de la classification automatique, pour la
conception logique d'une base de données "
Journée intern. d'Analyse de Données et Informatique,
Séminaire INFORSID, Versailles, septembre 77
- [FKB 76] FLORY A., KOULOUMDJIAN J., BOULANGER D.,
" Semantique et structure de données - Application à la
conception de systèmes d'information "
Congrès AFCET, novembre 76
- [FK 78] FLORY A., KOULOUMDJIAN J.,
" A model for the description of the information system
dynamics "
Second Conf. ECI, Venise, octobre 78
- [FS 76] FRY J.P., SIBLEY E.H.,
" Evolution of Database Management Systems "
ACM Computing Surveys, vol. 8, n°1, mars 76

- [Ger 75] GERRITSEN B.,
" A preliminary system for design of D.B.T.G. data structures "
CACM, octobre 75
- [GJP 76] GIRAUDIN J.P., JAHU D., PECCOUD F.,
" Le projet MACSI-P : son modèle de représentation des données
et des traitements "
Séminaire INFORSID Groupe II, Saint Pierre de Chartreuse, oct 76
- [Hea 71] HEATH I.J.,
" Unacceptable file operations in a relational data base "
Proc. 1971, ACM SIGFIDET Workshop on data description,
access and control, San diago, Ca, novembre 11-12 1971
- [HR 75] HUBBARD G., RAVEN N.,
" Automating Logical file design "
Proc. VLDB, septembre 75
- [Isl 78] ISLOOR S.S.,
" An algorithm with logical simplicity desining third normal
form relationl database schema form functional dependencies "
ICMOD 78, 29-30 juin 78, Milano (Italy)
- [Ken 73] KENT W.,
" A Primer of Normal Forms "
IBM Sys. Dev. Div. TR02.600, San José, Ca., 1973
- [Knu 73] KNUTH D.,
" The art of computer Programming "
Vol. 1, 3, Addison Wesley Publ. Compagny
- [Leo 76] LEONARD M.,
" Aides algorithmiques à la conception de bases de données "
Thèse d'Ingénieur, USMG, juin 76
- [LO 75] LUCCHESI C.L., OSBORN S.L.,
" Candidate Keys for relations "
Tech. Rep., University of Waterloo Ontario, Canada, 1971

- [Loz 76] LOZINSKI E.L.,
" Relational transformation and a redundancy in relational
Data bases "
Conf. on VLDB, Bruxelles 76, pp 95-101
- [LR 76] LEONARD M., REYNAUD F.,
" Existence du consensus et caractérisation des couvertures
et bases irrédondantes d'une fonction $\sum_i \mu_i A'_i$ "
Rapport de Recherche, septembre 76, Computing Laboratory,
Université de Grenoble
- [LT 78] LING T.W., TOMPA F.,
" Adequate definition for third normal form "
Research Report CS-78-34, University of Waterloo, july 78
- [Mel 76] MELTZER H.S.,
" Structure redundancy in the conceptual schema in the
administration of very large Data Base "
Conf. on VLDB, Bruxelles 76, pp 13-25
- [M11 65] MILLER R.E.,
" Switching Theory "
Vol. 1 : John Wiley & Sons, 1965
- [MMS 79] MAIER D., MENDELZON A., SAGIV Y.,
" Testing Implications of Data Dependencies "
Proc. ACM-SIGMOD 1974, Int. Conf. on Management of Data,
juin 79
- [MP 76] MIJARES I., PEEBLES R.,
" Structural Semantics of Data Bases "
IFIP TC 2 Working Conf. on Modeling in Data base Management,
Forêt Noire 1976
- [MUL 78] MULTICS Relational data Store, Reference Manuel
juin 78, Honeywell Information System

- [Nic 78a] NICOLAS J.M.,
" Mutual Dependencies and some Results on Undecomposable Relations "
Proc. VLDB Conf., Berlin, septembre 78, pp360-367
- [Nic 78b] NICOLAS J.M.,
" First order logic formalization for functional, multivalued and mutual dependencies "
Proc. ACM SIGMOD Inter. Conf. on Management of Data, Austin, Texas, juin 78, pp 40-46
- [Ng 77] NGUYEN GIA TOAN,
" URANUS: une approche relationnelle à la coopération de bases de données "
Thèses 3^{ème} Cycle, USMG, décembre 77
- [Nij 77] NIJSSSEN G.M.,
" Current issues in conceptuel schema concepts, architectures and models in D.B.M.S. "
IFIP TC2, Nice, janvier 77
- [Osb 77] OSBORN S.L.,
" Normal forms for relational Data Bases "
Ph. D. Thesis, University of Waterloo, nov. 77
- [Osb 78] OSBORN S.L.,
" Existence of a covering Boyce-Codd Normal form "
Information Processing Letter (to appear)
- [Par 79] PAREDAENS J.,
" Transitive dependencies in a database schema "
Tech. Report. Universitaire Antwerpen, Wilrijk (Belgium) 79
- [Pid 79] PICHAT E., DELOBEL C.,
" Designing third normal form for relational database schema "
Rapport de Recherche n°149, Univerité de Grenoble, janv. 79
- [R 77] RHYNE V.T. et al.,
" A new technique for the fast minimization of switching functions " IEEE Trans. Comput. C26, 8, pp 757-764, août 77

- [Ris 77] RISSANEN J.,
" Independent components of Relations "
ACM TODS 2,4, pp 317-325, décembre 77
- [SAAF 73] SENKO M., ALTMAN E., ASTRAHAN M., FEHDER,
" Data structures and accessing in DBMS "
IBM syst. Journal 12, 1, 1973, pp 39-93
- [SF 79] SAGIV Y., FAGIN R.,
" An equivalence between relational data base dependencies
and subclass of propositional logic "
IBM Research Report, RJ 2500, mars 79
- [SOC 73] Manuel d'utilisation SOCRATE
CII, juillet 73
- [SSmi 77] SMITH J.M., SMITH D.C.P.,
" Data Abstractions : Aggregation and generalization "
ACM Trans. on Database Syst. Vol.1, n°2, juin 77, pp 105-133
- [SS 75] SCHIMD H.A., SWENSON J.R.,
" On the semantics of relational Model "
Proc. ACM SIGMOD, W.K.King (ED.), San José, Ca, mai 75,
pp 211-223
- [Sti 76] STIERS A.,
" Comparaison des systèmes SOCRATE disponibles au CICG "
Note technique T 40, octobre 76
- [Sto 76] STONEBRAKER M. et al.,
" The design and implementation of INGERS "
ACM TODS, vol.1, n°3, septembre 76
- [Tar 76] TARDIEU H. et al.,
" Modeling in data base management "
IFIP TC2 Working Conference, Forêt Noire, janvier 76

- [THN 76] TARDIEU H., HECKENROTH, NANJI D.,
" Etude d'une methodologie d'analyse et de conception d'une
base de données "
Contrat IRIA, n°74.180, 1976
- [Tod 75a] TODD S.,
" PRTV : an efficient implementation of large relational
Data Bases "
VLDB Conf., 1975
- [Tod 75b] TODD S.,
" Integrated architecture for transaction specification and
optimisation in relational data base systems "
UKSC Scientific center, Report n° 0085 novembre 75
- [Tod 76] TODD S.,
" Peterlee Relational Test Vehicule : an overview "
IBM System Journal, vol. 15, n°4, novembre 76
- [TL 77] TSICHRITZES D.C., LOCHOYSKY F.H.,
" Data base Management Systems "
Accademic Press 1977
- [WW 75] WANG C.P., WEDEKIND H.H.,
" Segment synthesis in logical data base design "
IBM Journal of Research and Devel, janvier 1975
- [YJ 76] YU C.T., JOHNSON D.T.,
" On the Complexity of Finding the set of candidate Keys
for a given set of functional dependencies "
Information Processing Letters, vol. 5, n°4, octobre 76,
pp 100-101
- [Zan 76] ZANIOLO C.,
" Analysis and Design of Relational Schemata of Database
systems "
Tech. Report UCLA-ENG-7669, Dept. of Computer Science,
University of California, juillet 76

[ZM 78]

ZANIOLO C., MELKANOFF M.A.,

" Relational schemas for Database systems "

Tech. Report UCLA-ENG-7801, University of California,
Los Angeles, 1978

[Vet 77]

VETTER C.M.,

" Data base design by applied data synthesis "

Proc. Third Conference on VLDB Tokyo Japan 1977, pp 428-440

dernière page de la thèse

AUTORISATION DE SOUTENANCE

VU les dispositions de l'article 3 de l'arrêté du 16 Avril 1974,

VU les rapports de présentation de Messieurs :

- C. DELOBEL, Professeur à l'Université Scientifique et Médicale de GRENOBLE

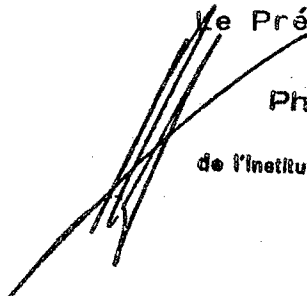
- J. KOULOUMDJIAN, Professeur à l'I.U.T. Informatique de LYON

Mademoiselle Salwa T A H A

est autorisée à présenter une thèse en soutenance pour l'obtention du diplôme de DOCTEUR-INGENIEUR, spécialité "Génie Informatique".

Grenoble, le 22 Novembre 1979

Le Président de l'I.N.P.G.


Ph. TRAYNARD
Président
de l'Institut National Polytechnique