82.773

UNIVERSITE DE NANCY I U.E.R. DE SCIENCES MATHEMATIQUES

CENTRE DE RECHERCHE EN INFORMATIQUE DE NANCY

Sc N 82/57A

MODELE ET OUTIL POUR LA CONCEPTION DES SYSTEMES D'INFORMATION DANS LES ORGANISATIONS

Projet REMORA

UNIVERSITE * NANCY * BIBLIOTHEQUE-SCIENCES

THESE

présentée et soutenue publiquement le 11 juin 1982
devant la commission d'examen
A L'UNIVERSITE DE NANCY I
pour l'obtention du grade de
DOCTEUR ES-SCIENCES MATHEMATIQUES
par

Odile FOUCAUT

Président du Jury : . . . . . J. LEGRAS

Rapporteurs : . . . . . C. ROLLAND
. . . . . F. BODART
. . . . . C. PAIR

Examinateurs : . . . . . J.C. BOUSSARD
. . . . . M. CREHANGE
. . . . . C. DELOBEL
. . . . . J.C. DERNIAME

UNIVERSITE DE NANCY I U.E.R. DE SCIENCES MATHEMATIQUES

CENTRE DE RECHERCHE EN INFORMATIQUE DE NANCY

MODELE ET OUTIL POUR LA CONCEPTION DES SYSTEMES D'INFORMATION DANS LES ORGANISATIONS

Projet REMORA

UNIVERSITE * NANCY * BIBLIOTHEQUE-SCIENCES

THESE

présentée et soutenue publiquement le 11 juin 1982
devant la commission d'examen
A L'UNIVERSITE DE NANCY I
pour l'obtention du grade de
DOCTEUR ES-SCIENCES MATHEMATIQUES
par

Odile FOUCAUT

Président du Jury : ..... J. LEGRAS

Rapporteurs : ..... C. ROLLAND
..... F. BODART
..... C. PAIR

Examinateurs : ..... J.C. BOUSSARD
..... M. CREHANGE
..... C. DELOBEL
..... J.C. DERNIAME

SOMMAIRE

ARRIVÉE ANNULÉE
3 MARS 1983
CENTRE D'INFORMATION SCIENTIFIQUE ET TECHNIQUE ET DE TRANSFERTS TECHNOLOGIQUES

INTRODUCTION

PREMIERE PARTIE

CHAPITRE 1 : PROBLEMATIQUE DE LA CONCEPTION DES SYSTEMES D'INFORMATION 1

I - ETUDE DES SYSTEMES D'INFORMATION 1

I-1.- Les systèmes d'information sont des représentations codées 2

I-2.- Ce que représentent les systèmes d'information : la sémantique des SI ... 4

I-3.- Caractéristiques des représentations 6

I-4.- L'amélioration du service rendu par les SI nécessite l'amélioration des représentations 8

II - NOTRE POLITIQUE DE LA CONCEPTION DES SI .. 10

II-1.- Problèmatique de la conception des SI 10

II-2.- Choix d'une solution 10

II-3.- Schéma conceptuel de SI et modèle conceptuel de SI 14

III - ANALYSE DES MODELES EXISTANT ACTUELLEMENT 22

III-1.- Grille d'évaluation des modèles .. 22

III-2.- Etude des modèles de SI 23

DEUXIEME PARTIE : MODELE CONCEPTUEL DE SYSTEME D'INFORMATION

CHAPITRE 2 : MODELE DESCRIPTIF POUR L'ANALYSE D'UNE ORGANISATION

INTRODUCTION 43

I - DEFINITION D'UNE ORGANISATION ET INTRODUCTION DES CATEGORIES DE PHENOMENES REELS 44

I-1.- Définition d'une organisation 44

I-2.- Le modèle descriptif et son rôle .. 45

I-3.- Analyse dynamique de l'organisation : introduction des catégories 46

II - ANALYSE ET DESCRIPTION DES CATEGORIES .. 49

II-1.- Choix d'un mode de description ... 49

II-2.- Structure de la catégorie objet .. 50

II-3.- Structure de la catégorie opération 57

II-4.- Structure de la catégorie événement 64

III - FORMALISATION DU MODELE DESCRIPTIF 73

III-1.- Définition générale de la dynamique d'une organisation 73

III-2.- Expression formelle de la dynamique du sous-système causal d'une organisation 76

CONCLUSION 84

CHAPITRE 3 : UN MODELE CONCEPTUEL POUR LA STRUCTURATION DES SI

INTRODUCTION 86

I - INTRODUCTION DU TEMPS ET DES TYPES DANS LE MODELE RELATIONNEL 87

I-1.- Rappels sur le modèle relationnel . 87

I-2.- Introduction du temps dans le modèle relationnel 103

I-3.- Les types de relation et les types de constituants 111

II - NORMALISATION ET REPRESENTATION DE LA CATEGORIE OBJET 114

II-1.- Objectifs de la représentation de la catégorie OBJET 114

II-2.- Mode d'expression retenu : relation permanente en forme normalisée ... 115

II-3.- Méthode de représentation de la catégorie OBJET - Concepts de C-OBJET et C-CLASSE 115

II-4.- Types de C-OBJETS et sémantique associée 121

II-5.- Expression du schéma conceptuel .. 123

II-6.- Représentation d'un état et d'un changement d'état dans un SI construit à l'aide du modèle conceptuel 124

III - NORMALISATION ET REPRESENTATION DE LA CATEGORIE OPERATION 128

III-1.- Critères de normalisation des catégories OPERATION et EVENEMENT 128

III-2.- Normalisation et représentation des classes d'opérations - Concept de C-OPERATION 128

III-3.- Représentation des C-OPERATIONS . 130

IV - NORMALISATION ET REPRESENTATION DE LA CATEGORIE EVENEMENT 131

IV-1.- Normalisation des classes d'événements Concept de C-EVENEMENT 131

IV-2.- Représentation des C-EVENEMENTS .. 132

IV-3.- Représentation des classes d'associations "DECLENCHE" - Concept de C-DECLENCHE 134

V - FORMULATION DE L'APPLICATION DECL DANS LE MODELE CONCEPTUEL 136

V-1.- Applications entre ensembles de classes de phénomènes 136

V-2.- Profil des textes des C-OPERATIONS 137

V-3.- Profil des prédicats des C-EVENEMENTS 137
V-4.- Profil des conditions de déclenchement 138
V-5.- Profil des facteurs de déclenchement 138
V-6.- Formulation de DECL dans le modèle conceptuel 139

VI - REPRESENTATION DE L'APPLICATION DECL DANS LE MODELE CONCEPTUEL. METASTRUCTURE DE L'ORGANISATION 141

VI-1.- Définition de la métastructure de l'organisation 141
VI-2.- Représentation instantanée de la métastructure 141
VI-3.- Représentation historique de la métastructure 145
VI-4.- Expression de DECL à partir de la métastructure 148

CONCLUSION 148

TROISIEME PARTIE : LA MODELISATION

CHAPITRE 4 : ANALYSE STRUCTURELLE ET REGLES DE VALIDITE D'UN SCHEMA CONCEPTUEL DE SI

INTRODUCTION 151

I - EXEMPLE DE SCHEMA CONCEPTUEL 151

I-1.- Enoncé : gestion des prêts-client . 151
I-2.- Schéma conceptuel 153

II - LE SOUS-SCHEMA DES DONNEES 158

II-1.- Définition 158
II-2.- Structure du sous-schéma des données 159
II-3.- Représentation graphique du sous-schéma des données 161

II-4.- Sous-schéma des données de l'exemple gestion des prêts-client 163

III - LE SOUS-SCHEMA DE LA DYNAMIQUE 165

III-1.- Définition 165
III-2.- Structure du sous-schéma de la dynamique 165
III-3.- Sous-schéma de la dynamique de l'exemple gestion des prêts-client 169

IV - REGLES DE VALIDITE D'UN SCHEMA CONCEPTUEL 176

IV-1.- Schéma conceptuel valide 176
IV-2.- Rappel du schéma conceptuel de la métastructure d'un SI 176
IV-3.- Règles de validité syntaxique 178
IV-4.- Règles de validité sémantique 182

CONCLUSION 191

CHAPITRE 5 : DEFINITION CONCEPTUELLE D'UN SI ET TYPE DE LANGAGE ASSOCIE 192

INTRODUCTION 192

I - LE SYSTEME D'INTEGRITE 192

I-1.- Nécessité du système d'intégrité .. 192
I-2.- Réduction du système d'intégrité d'un SI par rapport à celui d'une base de données 192

II - LANGAGE NECESSAIRE POUR EXPRIMER LA DESCRIPTION CONCEPTUELLE D'UN SI 199

II-1.- Ce que doit permettre d'exprimer le langage 199
II-2.- Les langages existants 200

QUATRIEME PARTIE

CHAPITRE 6 : SYSTEME D'AIDE A LA CONCEPTION D'UN SI

INTRODUCTION 203

I - SYSTEME DE GESTION DE SYSTEMES D'INFORMATION 104
I-1.- Rôles du SGSI 204
I-2.- Architecture du SGSI réalisé dans l'équipe REMORA 207

II - POURQUOI UN SYSTEME D'AIDE A LA CONCEPTION D'UN SI 208

III - FONCTIONS DE L'OUTIL 210
III-1.- Aide à la conception du SI initial 210
III-2.- Documentation du schéma conceptuel 213
III-3.- Gestion de l'évolution de structure du SI 213

IV - ARCHITECTURE DU SYSTEME REALISE 215
IV-1.- Schéma conceptuel de la métabase . 216
IV-2.- Schéma logique de la métabase 221

V - FONCTION : AIDE A LA CONCEPTION DU SCHEMA CONCEPTUEL INITIAL 226
V-1.- Méthode de conception d'un schéma conceptuel de SI 227
V-2.- Langage de définition d'un schéma conceptuel de SI 227
V-3.- Les programmes de contrôle d'un schéma conceptuel 231

VI - FONCTION DOCUMENTATION 234
VI-1.- Programmes d'édition du système .. 234
VI-2.- Macrolangage de documentation 237

VII - GESTION DE L'EVOLUTION DE STRUCTURE ... 240
VII-1.- Types de modification de structure de l'organisation 240
VII-2.- Modifications concernant le sous-schéma des données 241
VII-3.- Modifications concernant le sous-schéma de la dynamique 246
VII-4.- Macrolangage de définition d'évolution de structure 249
VII-5.- Répercussion des modifications du schéma conceptuel sur la structure interne du SI dans le cas de l'implantation sous la forme d'une base de données relationnelle 254

CONCLUSION 255

CONCLUSION 258

ANNEXE I : EXEMPLE DE SCHEMA CONCEPTUEL DE SI ET D'UTILISATION DE L'OUTIL

INTRODUCTION 268

I - DEFINITION DE L'ORGANISATION REELLE A MODELISER 269
I-1.- Présentation de l'organisation et de ses objectifs 269
I-2.- Description détaillée de la gestion des commandes prévisionnelles 273
I-3.- Gestion des commandes urgentes 278

II - LE SCHEMA CONCEPTUEL DU SI 280
II-1.- La Métastructure du SI 280
II-2.- La structure du SI 283

III - TESTS DE VALIDITE DU SCHEMA CONCEPTUEL .. 295

IV - SPECIFICATION DES CONSTITUANTS DE TYPE TEXTE 297

V - EXEMPLES DE RENSEIGNEMENTS QUE L'ON PEUT OBTENIR A PARTIR DU SCHEMA CONCEPTUEL ET DU SI 306

V-1.- Questions sur le fonctionnement posées sur la métabase 306

V-2.- Questions sur la situation présente 307

V-3.- Questions sur le passé 308

V-4.- Questions sur l'origine d'une situation passée 309

V-5.- Questions concernant le futur 310

ANNEXE II : PROGRAMMES PRINCIPAUX DE L'OUTIL 312

BIBLIOGRAPHIE 321

C'est un grand plaisir pour moi de remercier ici les professeurs qui depuis mes débuts de jeune chercheur m'ont aidée et guidée :

- J. LEGRAS, Professeur à l'Université de NANCY I, pour l'honneur qu'il me fait en présidant ce jury. Il fut mon premier directeur de recherche et c'est lui qui, il y a une douzaine d'années, m'a orientée vers l'informatique d'organisation. Je lui suis très reconnaissante de m'avoir fait, à l'époque, approcher les problèmes "sur le vas" dans des administrations et des entreprises. Cette expérience fut très bénéfique au développement de ce travail.

- C. ROLLAND, Professeur à l'Université de PARIS I, pour tout ce que je lui dois, aussi bien dans le domaine de l'enseignement que dans celui de la recherche. Elle est l'instigatrice du projet de recherche REMORA dans lequel s'inscrit ce travail ; elle a su créer, diriger et animer autour de ce projet une équipe de recherche dans laquelle il faisait "bon vivre et travailler". Je la remercie, vivement, pour toutes ces années passées côte à côte, pendant lesquelles, elle fut pour moi bien davantage qu'un directeur de recherche.

- C. PAIR, Professeur à l'Institut National Polytechnique de NANCY, actuellement Directeur des Lycées au Ministère de l'Education Nationale, pour avoir accepté il y a deux ans d'être rapporteur de ce travail. Je lui suis reconnaissante de m'avoir aidée à la formulation des concepts développés dans cette thèse. Cette tâche, plus théorique, effectuée seule et non plus en équipe, fut pour moi une épreuve difficile mais qui s'est avérée très utile. En effet elle a permis un rapprochement avec les préoccupations de chercheurs du CRIN travaillant dans le domaine de la théorie de la programmation.

./...

Mes remerciements sincères vont également à :

- F. BODART, Professeur à l'Institut Universitaire Notre Dame de la Paix à NAMUR, pour l'intérêt qu'il a toujours porté au projet REMORA et à mon travail en particulier. Les discussions et les échanges très enrichissants que nous avons eus, notamment lors des colloques INFORSID, ont contribué à l'avancement de ce travail. Je le remercie tout particulièrement d'avoir accepté d'en être rapporteur.

Que les membres du jury trouvent ici également l'expression de ma gratitude.

- J.C. BOUSSARD, Professeur à l'Université de NICE, a bien voulu s'intéresser à mon travail et a accepté un long déplacement pour participer à la commission d'examen de cette thèse, je l'en remercie vivement.

- C. DELOBEL, Professeur à l'Université de GRENOBLE, m'a aidée à la mise en forme final de ce travail, par les remarques et les conseils qu'il m'a adressés à la suite de la lecture de la première version. Je lui en suis très reconnaissante et le remercie également d'avoir accepté d'effectuer un long voyage pour siéger à ce jury.

- J.C. DERNIAME, Professeur à l'Université de NANCY I, actuel Directeur du CRIN, m'a apporté son aide par l'intermédiaire des nombreux échanges et contacts que nous avons pu avoir au CRIN tout au long de ces années passées dans le même service. Je le remercie tout particulièrement pour le soutien qu'il a toujours apporté à l'équipe de recherche REMORA. Merci à lui d'avoir bien voulu juger cette thèse.

- M. CREHANGE, Professeur à l'Université de NANCY II, a toujours su m'accueillir chaleureusement et les nombreuses discussions que nous avons eues à propos des bases de données laissent présager une collaboration efficace entre nous à l'avenir. Sa participation à ce jury est un grand plaisir pour moi.

./...

J'adresse également mes remerciements à tous les chercheurs qui, au cours des années, ont participé au projet REMORA. Sans leur activité, leurs idées, leur soutien et leurs conseils ce travail n'aurait certainement pas existé puis abouti. Je remercie plus spécialement O. THIERY pour sa constante collaboration.

J'associe à ces remerciements M.T. DRIQUERT et B. MALTHIERY qui se sont chargés avec compétence et bonne humeur de la réalisation matérielle de cet ouvrage.

Enfin ces remerciements ne seraient pas complets si je n'évoquais pas ici ma famille.

Je remercie Jean-François pour le soutien moral et matériel qu'il m'a toujours apporté lors des difficultés rencontrées tout au long de ce travail, et, je demande pardon à Alix, Lucie, Jean-Baptiste, Elise et Jean-Yves, de ne pas leur avoir toujours accordé, à cause de cette thèse, l'attention qu'ils méritaient.

INTRODUCTION

Ce travail se situe dans le courant de recherches concernant la conception des systèmes d'information (SI) pour les organisations.

Il est une des études menées depuis 1973 par l'équipe de recherche dirigée par Madame ROLLAND dans le cadre du projet REMORA dont l'objectif est la définition d'un système de pilotage pour l'analyse et la conception des systèmes d'information.

L'insuffisance des moyens mis à la disposition des concepteurs de systèmes d'information a entraîné, en France, à partir des années 1970, le développement de recherches dans le domaine de l'informatique d'organisation. Le projet Remora se situe dans ce domaine de rerecherches.

Le groupe INFORSID, créé par l'INRIA, a notamment contribué à ce développement par l'organisation de séminaires et la création de groupes de travail chargés en particulier de définir les axes de recherche possibles dans ce domaine (INF 75_1, 75_2, 76, 78_1, 78_2, 80).

Dans cette optique, le groupe INFORSID2 (INF 76) a classé les recherches à entreprendre en matière de conception de systèmes d'information en quatre axes : modèle, méthode, langage, outil.

Ce travail s'appuye sur les résultats qui ont été produits par l'équipe REMORA de 1973 à 1977 et notamment sur les éléments du système de pilotage qui ont été mis au point pour résoudre les problèmes d'analyse et de description des faits de gestion et qui abordent chacun de ces axes de recherche.

Citons en particulier :

- une méthode d'analyse des problèmes organisationnels mise au point par W. PEREA (PER 76).

- un langage de description de SI, associé à un générateur de programmes développé par D. LAMY (LAM 77), (FOU 79).

- un outil de pilotage de la conception du SI guidant le concepteur et contrôlant ses définitions, défini et réalisé par O. THIERY (THI 76), ($ROL_2$80).

Il s'inspire par ailleurs des travaux de recherche du domaine des bases de données et de l'expérience pratique de Madame ROLLAND dans ce domaine (BEN 79).

Il est, en effet, apparu clairement que l'introduction de nouveaux outils, les systèmes de gestion de bases de données, permettant de gérer et d'utiliser de grands volumes de données n'apportait pas une solution complète au problème de la conception de la base elle-même.

Ces outils résolvent des problèmes techniques mais n'aident pas réellement les concepteurs à construire des représentations aussi fidèles que possible des organisations réelles.

Ainsi se sont développées de nouvelles recherches dont l'objectif était la résolution du problème de représentation. Ces recherches ont conduit à la proposition de modèles de données, dits conceptuels, indépendants de toute contrainte technique d'implantation et d'utilisation de données sur du matériel informatique et dont la sémantique des concepts permet une interprétation de la représentation qui était difficile avec les modèles associés aux premiers systèmes de gestion de bases de données.

Les actes des congrès de Namur (IFI 75) et de Freundenstadt (IFI 76) du groupe IFIP TC2 présentent un panorama des principaux résultats obtenus dans ce domaine de la représentation des données à un niveau conceptuel.

Ces modèles, cependant, ne résolvent qu'une partie du problème de la conception d'un système d'information.

Une base de données n'est qu'un sous-ensemble du système d'information. Le système d'information comprend, en plus des données, des programmes et des commandes règlant l'exécution des programmes sur les données. Les modèles conceptuels de données n'apportent aucune aide pour la conception des programmes et des commandes.

Il fallait donc, à notre avis, étendre les modèles de données pour englober une représentation plus large des phénomènes réels.

Le problème ainsi posé à l'équipe de recherche a trouvé une solution dans le modèle qui est présenté dans cette thèse.

Elle comporte quatre parties, découpées en chapitres, et intitulées respectivement :

- Problématique de la conception des systèmes d'information
- Modèle conceptuel de système d'information
- Modélisation conceptuelle
- Système d'aide à la conception d'un SI.

Une annexe est consacrée à la présentation de l'application du modèle et du système à un exemple d'organisation proche d'une organisation réelle.

La première partie (chapitre 1) développe tout d'abord une analyse de la sémantique des données, des programmes et des commandes d'un SI d'une organisation. Nous y montrons que certains aspects de la réalité, utiles aux gestionnaires, sont encore mal ou incomplètement représentés dans les SI actuels, notamment tous les aspects dits "dynamiques" qui concernent l'évolution des constituants de l'organisation au cours du temps.

Nous y exposons ensuite une politique de conception des systèmes d'information qui consiste, comme en matière de constructions informatiques dans d'autres domaines, programme ou logiciel par exemple, à commencer par effectuer une représentation abstraite, indépendante de toute caractéristique du matériel ou du logiciel informatique, de l'ensemble des phénomènes réels à représenter dans le SI.

Nous étudions les problèmes posés par l'élaboration d'une telle représentation et effectuons une étude comparative des modèles proposés actuellement pour les résoudre. Nous montrons qu'il reste certains points sans solution satisfaisante : la représentation du temps, l'intégration de la représentation des aspects dynamiques à la représentation des aspects statiques.

La seconde partie est consacrée à la présentation du modèle conceptuel, elle comprend deux chapitres.

Le premier (chapitre 2), présente une analyse du monde réel des organisations. Elle conduit à un modèle que nous qualifions de "descriptif". Il permet de décrire l'ensemble des phénomènes existant ou se produisant dans une organisation,ainsi que leurs interactions sous la forme d'une unique structure. Pour exprimer cette structure, nous utilisons les notions simples de propriété, classe et association introduites habituellement pour la représentation des données. La formalisation associée à ce modèle montre sa complêtude en ce qui concerne la représentation de la dynamique.

Le second (chapitre 3), définit le modèle conceptuel de SI. Nous en donnons tout d'abord les caractéristiques générales : c'est un modèle de nature relationnel, normatif, intégrant le temps, et utilisant des types de relationset de constituants. Ce modèle introduit, dans le modèle relationnel, une nouvelle forme de dépendance fonctionnelle et une nouvelle forme de relation qui permettent une représentation minimale, dans le temps, des phénomènes et de leur structure.

La troisième partie étudie le type de modèlisation à laquelle conduit le modèle.

Son premier chapitre (chapitre 4), consiste en une analyse de la structure de la collection de relations appelée schéma conceptuel du SI, à laquelle conduit l'usage du modèle pour une organisation donnée. Nous y proposons également un mode de représentation graphique de deux sous-schémas particuliers du schéma conceptuel, qui facilite son interprétation. Enfin nous donnons une liste de règles que doit vérifier un schéma conceptuel de SI pour être valide.

Le second chapitre (chapitre 5), discute :

- d'une part,des limites de la modèlisation et montre que, comme dans une modèlisation limitée aux données, il est nécessaire de complèter le schéma conceptuel par des contraintes d'intégrité,
- d'autre part,du type de langage nécessaire pour exprimer une description conceptuelle complète, c'est-à-dire le schéma conceptuel et les contraintes d'intégrité.

La quatrième partie est consacrée à la description d'un système d'aide à la conception de système d'information que nous avons réalisé.

Ce système s'intègre dans un outil plus vaste réalisé dans l'équipe REMORA et appelé système de gestion de systèmes d'information.

Son rôle est d'assister le concepteur du SI

- lors de l'élaboration du schéma conceptuel du SI,en lui fournissant des outils de description, de documentation et de contrôle,
- lors de la mise à jour du schéma conceptuel consécutive à des évolutions de structure de l'organisation,en déterminant toutes les conséquences d'une modification élémentaire au niveau du schéma conceptuel et au niveau du SI.

PREMIERE PARTIE

CHAPITRE 1 : PROBLÈMATIQUE DE LA CONCEPTION DES SYSTÈMES D'INFORMATION

INTRODUCTION

Ce chapitre a pour objectif d'introduire les problèmes posés par la conception des systèmes d'information (SI) dans les organisations et de comparer les solutions proposées jusqu'à ce jour pour les résoudre.

Dans la première partie, nous rappelons que les SI actuels sont des représentations codées et partielles des organisations, de plus ou moins bonne qualité, qui ne satisfont pas complètement leurs utilisateurs.

Dans la seconde partie, nous décomposons le problème général de représentation posé par la conception des SI en quatre sous-problèmes. Puis, en nous appuyant sur les résultats obtenus ces dernières années, en matière de conception de constructions informatiques en général, nous faisons l'hypothèse que la résolution de ces sous-problèmes, nécessite le passage par une expression abstraite de l'ensemble de la réalité à représenter. Nous appelons schéma conceptuel de SI cette expression et modèle conceptuel de SI l'ensemble des concepts sur lesquels elle s'appuie.

Des différents sous-problèmes à résoudre nous déduisons un certain nombre de caractéristiques que doit posséder un tel modèle.

Dans la troisième partie, nous nous appuyons sur ces caractéristiques pour étudier les modèles proposés actuellement pour concevoir des SI. Nous mettons ainsi en évidence les problèmes de représentation, résolus, en cours d'étude ou non encore résolus, et situons notre proposition dans cet ensemble.

I - ETUDE DES SYSTEMES D'INFORMATION

Dans cette partie nous analysons les systèmes d'information en tant que représentations particulières du monde réel des organisations.

Nous étudions la sémantique des trois composantes de tout système d'information : les données, les traitements, les commandes dynamiques, et nous montrons qu'il existe certains aspects de la réalité dont la connaissance est nécessaire aux responsables des organisations, qui sont encore mal représentés dans les systèmes d'information actuels.

I.1.- Les systèmes d'information sont des représentations codées

De tout temps,les hommes ont utilisé des représentations codées de faits réels pour effectuer du commerce, comme par exemple les valeurs numériques représentant le poids ou la taille des objets réels exprimées dans une certaine unité.

Puis ils ont inventé des objets abstraits qui leur permettaient de mieux comprendre et de mieux gérer leurs entreprises, ils ont également représenté ces objets par des codes. Citons par exemple les objets abstraits tels que les comptes, les salaires, les taux, les plafonds sans lesquels les entreprises ne sauraient plus fonctionner à l'heure actuelle.

La complexité toujours croissante des entreprises et des sociétés contemporaines a entraîné la multiplication et la diversification de ces codes.

L'apparition et le développement des techniques informatiques ont permis :

- la mémorisation des codes sous la forme de "données", appelées aussi "informations"
- le traitement des codes par des "programmes"

c'est-à-dire la création de systèmes d'information

Un système d'information automatisé se présente actuellement dans toute organisation comme :

- un ensemble de données stockées sur des mémoires magnétiques, en général sous la forme de fichiers indépendants, parfois rassemblées dans une base de données.
- un ensemble de programmes créant ou utilisant ces données pour produire de façon régulière un certain nombre de documents.

Le fonctionnement du système d'information correspond à l'application des programmes sur les données,pour en prendre en compte ou bien en produire de nouvelles.

Dans une organisation,ce fonctionnement est régi par ce que nous avons choisi d'appeler des commandes dynamiques.

Les commandes dynamiques définissent,d'une part,la périodicité d'application des programmes sur les données et,d'autre part,leur

ordonnancement et leurs conditions d'exécution. Ces commandes font, à notre avis, partie intégrante du système d'information. En effet si l'on observe la description manuscrite des systèmes d'information réels contenue dans les dossiers classiques d'analyse,on rencontre :

- la description des informations sous forme de descriptifs de fichiers,
- la description des programmes sous des formes diverses telles que organigramme, table de décision, texte dans un langage particulier,
- la description de l'exploitation des programmes sous forme de diagrammes d'enchaînement définissant des "chaînes de programmes". Ces diagrammes sont complétés par un calendrier spécifiant les périodicités ou les dates de lancement des chaînes.

Ainsi,tout système d'information comprend trois types d'éléments - des données
- des traitements
- des commandes dynamiques

représentés par des codes.

La codification utilisée pour chacun de ces types est différente :

- Pour les données il s'agit de codes numériques, alphabétiques ou alphanumériques.

ex : le prix d'un produit est représenté par un entier compris entre 10 et 200.

La situation familiale d'un agent est représentée par une lettre appartenant à la liste C,V,M,D.

- Les programmes sont exprimés dans des langages de programmation, différents des langues naturelles par la rigueur de leur syntaxe et de leur sémantique.

- Les commandes dynamiques sont exprimées :
  - en partie dans le langage de commande de l'ordinateur utilisé,
  - en partie sous forme de consignes données aux opérateurs, exprimées le plus souvent de façon graphique,parfois en langue naturelle dans les dossiers d'exploitation du SI.

I-2.- Ce que représentent les systèmes d'information : la sémantique des SI

Les systèmes d'information sont utiles à la gestion des entreprises parce que les codes utilisés, ou tout au moins une bonne partie d'entre eux, ont une signification, c'est-à-dire un contenu sémantique qui permet l'interprétation par les gestionnaires des résultats fournis par le système d'information.

I-2.1.- Sémantique des données

Les données mémorisées dans les fichiers sont des représentations partielles des constituants de l'organisation et de leur structure.

Par exemple, le fichier client représente des caractéristiques des clients réels de l'organisation : leur nom , leur adresse , etc...

Une base de données commerciale représente à la fois les clients de l'entreprise, leurs commandes en cours et les produits au catalogue.

I-2.2.- Sémantique des traitements

Les programmes se présentent comme des ensembles d'instructions appartenant à deux types :

- des instructions d'entrée-sortie qui permettent l'accès aux données ou la création de données sur des supports

Ex : lire Fichier client

écrire ligne de facture

- des instructions de traitement qui permettent la valorisation de certaines données à partir d'autres

Ex : calculer montant net = montant brut - cotisation.

Les instructions d'entrée-sortie n'ont pas de contenu sémantique vis à vis de l'organisation ; elles n'existent dans les programmes que pour des raisons techniques, elles sont dépendantes du langage utilisé et du mode de stockage des informations.

Les instructions de traitement représentent :

- soit des transformations réelles survenues aux constituants de l'organisation.
  Ex : l'instruction Stock = Stock - quantité livrée
  représente la diminution physique du stock résultant de la livraison d'une commande.
- soit des règles de gestion de l'organisation
  Ex : l'instruction Prime = salaire brut/100
  représente la règle appliquée dans l'organisation pour calculer la prime des employés.

I-2.3.- Sémantique des commandes dynamiques

- Dans les commandes dynamiques, les informations concernant la périodicité d'exécution des programmes n'ont pas toujours de sémantique précise. En effet, cette périodicité est le plus souvent définie à partir de considérations économiques, techniques ou de sécurité, et non pas par référence aux faits réels survenant dans l'organisation.

Par exemple - la mise à jour du fichier Produit est effectuée tous les vendredis soirs, indépendamment de la fréquence d'arrivée réelle des commandes.

- la paie est calculée tous les 22 du mois, alors que la législation n'impose de la verser que le 1er du mois suivant.

- Ce décalage entre les faits réels ou les transformations réelles subies par les objets et leur prise en compte au niveau des données dans le SI rend difficile l'interprétation sémantique des commandes.

Elles représentent partiellement les règles de fonctionnement et de gestion de l'organisation, mais il est impossible de distinguer, dans leur expression, les caractéristiques propres de ces règles, de celles choisies pour des raisons techniques ou économiques.

En fait, remarquons que peu de réflexions ont été faites jusqu'à ce jour sur la sémantique et même sur la codification de ces commandes, essentiellement parce que leur gestion est restée manuelle.

En conclusion, il apparaît que les SI sont des représentations codées et partielles :

- des constituants de l'organisation
- des règles de gestion et de fonctionnement de l'organisation.

L'interprétation, en termes sémantiques, de ces représentations est assez facile pour les données, elle est beaucoup plus difficile pour les traitements et les commandes.

I-3.- Caractéristiques des représentations

Les représentations que sont les SI diffèrent d'une organisation à une autre suivant deux points :

- leur nature par rapport au temps : elles peuvent être instantanées ou historiques,
- leur dépendance ou leur indépendance par rapport à l'utilisation qui en est faite.

I-3.1.- Représentations instantanées ou historiques

La représentation fournie par un SI est le plus souvent instantanée. C'est-à-dire qu'elle représente, en partie, l'état de l'organisation à un instant donné.

Par exemple - le fichier produit représente après chaque mise à jour, l'état des produits de l'entreprise, à une certaine date,

- le fichier personnel représente la situation du personnel pour le mois en cours.

Remarquons qu'il y a en général un décalage entre le moment où l'on dispose de la représentation d'un état dans le SI et l'instant réel où l'organisation est passée dans cet état.

Par exemple si le fichier personnel est mis à jour uniquement le 20 de chaque mois, les changements de situations intervenus au cours du mois, sont soit représentés avec retard dans le SI ou pas représentés (s'ils se produisent après le 20).

Certains SI conservent les versions successives prises par les données au cours du temps. Par exemple les différentes versions mensuelles du fichier Personnel sont souvent archivées, mais, en général, suivant un mode de stockage qui ne permet pas une consultation directe de cet historique.

Ces systèmes fournissent alors une représentation historique partielle de l'organisation. Cette représentation historique se borne à une suite ordonnée dans le temps, de représentations de certains de ses états.

Notons, de plus, que cette représentation historique est, en général, limitée aux données, c'est-à-dire qu'il n'y a pas conservation dans les SI actuels, des versions successives des programmes et des commandes qui pourtant subissent également des modifications au cours du temps.

I-3.2.- Représentations dépendantes ou indépendantes de l'usage

Les SI classiques, organisés autour de fichiers, sont en général très dépendants de l'utilisation qui en est faite.

Au niveau des données, cela se traduit par le regroupement d'informations dans des fichiers pour des raisons pratiques et économiques et non pas pour des raisons sémantiques.

Par exemple, le prix unitaire du produit pourra figurer dans la représentation d'une commande dans le fichier des commandes afin de diminuer le nombre de fichiers à manipuler lors de l'exécution du programme éditant les factures.

Au niveau des programmes et des commandes, la dépendance par rapport à l'usage est encore plus nette ; les programmes sont souvent écrits pour répondre à un besoin spécifique d'un utilisateur, ou plus exactement les besoins en information, des gestionnaires conditionnent la prise en compte des transformations réelles et la périodicité d'exécution des traitements.

Par exemple si les gestionnaires n'ont besoin d'un état du stock des produits qu'une fois par semaine, la mise à jour du fichier produit ne sera effectuée qu'une fois par semaine, quelle que soit la fréquence réelle de livraison dans l'entreprise.

Les bases de données ont été introduites dans les SI en partie pour réduire cette dépendance entre les données et les traitements.

Cette indépendance a été réalisée par l'introduction de plusieurs niveaux dans la représentation (ANS 75).

- un niveau abstrait, appelé conceptuel, donnant une représentation des constituants et de leur structure indépendante de l'usage que l'on désire faire de la base de données

- un niveau interne, où les choix de structuration sont faits d'après les besoins des utilisateurs et les conditions d'utilisation de la base.

Remarquons que cette recherche d'une représentation indépendante de l'usage et des moyens techniques n'a été faite que pour les données et que les bases de données n'ont apporté au départ aucune modification dans la conception et la représentation des traitements et des commandes dynamiques.

I-4.- L'amélioration du service rendu par les SI nécessite l'amélioration des représentations

I-4.1.- Le service rendu par les SI actuels est encore insuffisant

Dans le domaine de l'informatique appliquée aux organisations, l'insatisfaction des utilisateurs des SI est devenu un lieu commun.

Les inconvénients des SI organisés en chaînes de programmes indépendantes ont été maintes fois énoncés et ont été en partie à l'origine du développement des bases de données ($COD_A 71$), (DAT 77).

Les bases de données ont, certes, apporté une amélioration du service rendu par les SI, en permettant de construire des représentations des constituants de l'organisation et de leur structure, plus fidèles que celles fournies par un ensemble de fichiers, et surtout en les rendant plus accessibles aux utilisateurs.

Cependant, certains besoins en information des gestionnaires sont encore très mal ou pas du tout satisfaits par les SI actuels, comme la connaissance du passé ou la connaissance des règles de gestion et de fonctionnement de l'organisation.

La connaissance du passé est parfois absolument nécessaire aux gestionnaires pour effectuer certaines tâches précises, comme, par exemple, les calculs de retraite, ou les régularisations de salaires. Elle constitue de toute façon un élément d'information important dans toute prise de décision.

La connaissance des règles de gestion et de fonctionnement de l'organisation est bien entendu indispensable à tout gestionnaire. Or dans une organisation de taille importante, ces règles sont nombreuses et complexes et c'est, à notre avis, un des rôles du SI d'apporter cette connaissance.

Par exemple le gestionnaire peut avoir besoin d'informations concernant - les modalités d'application des règles de gestion
- formule de calcul d'un tableau d'amortissement
- seuil à partir duquel on définit la rupture de stock

- l'application effective des règles de gestion au cours du temps comme :
  - la fréquence d'arrivée d'événements qui conditionnent l'application de la règle
    Ex : nombre de ruptures de stock survenues sur une période.

ou - la conséquence de l'arrivée de certains événements
  Ex : influence du retard des paiements des fournisseurs.

Les SI actuels, y compris ceux organisés autour d'une base de données, ne permettent pas la satisfaction de ces besoins pour deux raisons :

- les représentations sont instantanées, ou bien la représentation historique est limitée aux données. En particulier les SI ne gardent pas trace des programmes qui y sont exécutés au cours du temps et ne peuvent donc fournir aucune information sur leur fréquence ou leurs conditions d'exécution.

- les représentations des règles de gestion sont partielles ou déformées. La dépendance des programmes et des commandes dynamiques de l'usage particulier fait du SI et des caractéristiques du matériel utilisé, a pour conséquence, une représentation incomplète ou erronée des règles de gestion, de toute façon inaccessible aux utilisateurs.

I-4.2.- L'amélioration du service rendu passe par l'amélioration des représentations

Pour satisfaire ces besoins et par là améliorer le service rendu par les SI, il est nécessaire d'améliorer les représentations sous deux aspects :

- l'aspect temporel : La prise en compte du temps ne doit pas être limitée aux données mais être effective au niveau des traitements et des commandes dynamiques, de façon à ce que la représentation historique contienne, en plus des représentations des états successifs pris par les constituants au cours du temps, des représentations des faits réels qui les ont fait passer d'un état dans un autre.

- l'aspect dynamique : La représentation des règles de gestion et de fonctionnement de l'organisation doit être améliorée; elle doit être faite, comme celle des constituants dans une base de données, à un niveau abstrait de façon indépendante des conditions d'utilisation du SI. Dans la suite de ce travail nous regroupons sous le terme dynamique l'ensemble des règles de gestion et de fonctionnement de l'organisation.

II - NOTRE POLITIQUE DE CONCEPTION DES SI

II-1.- Problèmatique de la conception des SI

La qualité d'un SI est liée à la qualité des images des faits réels, présents ou passés, de l'organisation qu'il restitue aux gestionnaires.

Plus les images sont fidèles et variées, meilleur est le SI.

Le problème de la conception des SI est donc avant tout un problème de représentation.

Il s'agit de construire des représentations aussi complètes et aussi fidèles que possible de l'ensemble des aspects des phénomènes réels de l'organisation qui présentent un intérêt pour la gestion de celle-ci.

Ce problème de représentation comprend 4 sous-problèmes liés à la nature de la réalité à représenter :

1.- Problème de représentation d'une réalité fortement structurée

Les organisations sont des systèmes complexes dans lesquels les liens entre les constituants sont nombreux. La représentation des organisations pose donc le problème général de la représentation d'une réalité fortement structurée.

2.- Problème de représentation dans le temps

Les organisations sont des systèmes évoluant au cours du temps dont la connaissance de l'histoire est indispensable aux gestionnaires.

Cette particularité entraîne un problème de prise en compte du temps de façon aussi rigoureuse et aussi complète que possible dans la représentation d'un système réel.

3.- Problème de représentation d'une réalité dynamique

Les organisations sont des systèmes dynamiques particuliers dont le fonctionnement est régi par un ensemble de règles déterminées dont la connaissance est utile au gestionnaire.

La nature dynamique des organisations pose donc un problème de représentation d'une réalité dynamique.

4.- Problème d'intégration des représentations statiques et dynamiques

Comme dans tout système, la distinction entre aspect statique et aspect dynamique d'une organisation, est artificielle. La dynamique d'un système ne peut pas en effet être définie sans faire référence à ses composants ou à leur structure c'est-à-dire à son aspect statique. En conséquence au niveau de la représentation du système, il ne peut y avoir deux représentations indépendantes: l'une représentant les aspects statiques, l'autre représentant les aspects dynamiques, sans perte de sémantique.

II-2.- Choix d'une solution

II-2.1.- Choix d'un passage par une expression abstraite

La solution que nous proposons pour résoudre le problème général de la conception des SI, est identique à celle dévelop-

pée actuellement pour la conception de constructions informatiques quelconques, base de données, programmes, systèmes d'exploitation ou autres.

Elle consiste :

1) à définir une expression abstraite de la réalité à représenter.

2) à choisir une représentation concrète à partir de cette expression.

Dans le cas d'une base de données l'expression abstraite est appelée "schéma conceptuel" (BEN 76), (ANS 75), (NIJ 77), dans le cas d'un programme ou d'un système d'exploitation elle est appelée "spécification" (ABR 78), (PAI 79), (FIN 79).

Cette expression abstraite possède deux caractéristiques fondamentales :

1) Elle est indépendante de tout aspect technique lié à l'utilisation de l'informatique.

Par exemple un schéma conceptuel de base de données ne décrit pas comment organiser les données sur un support magnétique et quelles techniques d'accès à des structures d'information utiliser, mais uniquement les objets à représenter, leurs propriétés et leurs structures.

De même une spécification de programme ne décrit pas un algorithme, mais définit le problème à résoudre.

2) Elle est canonique

Nous entendons par là qu'elle possède des qualités particulières ou vérifie des règles précises.

Par exemple un schéma conceptuel de base de données est une représentation particulière des objets et de leur structure dont la principale qualité, qui lui est en général assignée est d'être minimale (BEE 78). De même une spécification de programme s'appuie, en général, sur un système formel dont elle doit respecter les règles (FIN 79).

Les avantages du passage par un niveau abstrait de définition, dans la conception d'une construction informatique, ont été mis en évidence aussi bien en matière de conception de base de données que de conception de programmes. Citons quelques uns de ces avantages :

Il permet - une meilleure compréhension de la réalité concernée
- un contrôle de la représentation
- un choix diversifié et raisonné des caractéristiques techniques de réalisation.

Il conduit - à des constructions informatiques plus souples et plus extensibles.

II-2.2.- Choix d'une approche modèle

Les moyens utilisés pour construire cette représentation abstraite sont différents suivant les domaines :

- Les théoriciens de la programmation sont partis des langages de programmation et ont cherché de nouveaux langages moins dépendants du matériel.

Les propositions actuelles comprennent notamment :

α) les langages utilisant le concept de type abstrait et les études théoriques faites sur cette notion (GUT 75), (ZIL 75), (DER 79), (LES 79).

Ces langages ont pour objectif de permettre une définition du type, c'est-à-dire d'une classe d'objets abstraits, par un ensemble d'opérations définies par des spécifications syntaxiques et sémantiques indépendantes de la réalisation du type c'est-à-dire de la structure physique choisie pour son implémentation.

Citons le langage CLU (LIS 75), le langage ADA (ICH 79), le langage ATM (MIN 79).

β) Les langages de spécification

Le rôle d'un langage de spécification est de permettre de définir les problèmes en vue de leur résolution par des programmes, de façon à ce que le passage de la spécification au programme puisse être réalisé par des transformations successives de la définition initiale.

Citons dans ce domaine, le langage Z (ABR 78), le langage proposé par FINANCE (FIN 79).

- Les chercheurs de la communauté scientifique des bases de données se sont orientés vers la définition de modèles,

La nécessité de passer par un niveau abstrait de représentation dans la conception d'une base de données est admis par l'ensemble des chercheurs de ce domaine depuis le congrès de Namur en 1975

Le terme de schéma conceptuel, introduit dans de nombreuses publications en 1976 est, depuis cette date, utilisé pour désigner la représentation abstraite de la structure de données à implémenter dans une base.

Pour exprimer ce schéma conceptuel un certain nombre de modèles ont été proposés. Nous rappellerons les plus connus dans l'étude des modèles que nous allons faire dans la troisième partie de ce chapitre.

Ces modèles, appelés modèles conceptuels de données, s'appuient tous sur une analyse du monde réel et leurs concepts sont en général définis sémantiquement à partir de cette analyse.

Le problème de la conception des SI incluant le problème de la conception des bases de données, nous avons choisi une approche modèle plutôt qu'une approche langage.

Remarquons cependant que ces deux approches ne s'opposent pas, mais au contraire se complètent : pour exprimer un schéma conceptuel construit à partir d'un modèle conceptuel de données, il faut utiliser un langage, et il apparaît qu'un langage de spécification ou un langage basé sur les types abstraits sont des moyens possibles d'expression. Réciproquement, tout langage comprend implicitement un modèle correspondant à l'ensemble des concepts manipulés par le langage.

Notons cependant qu'à un même modèle peuvent être associés plusieurs langages possédant des syntaxes différentes, ainsi l'approche modèle nous apparaît plus générale que l'approche langage.

II-3.- Schéma conceptuel de SI et modèle conceptuel de SI

Par analogie avec le domaine des bases de données nous appelons schéma conceptuel de SI, l'expression abstraite du SI et modèle

conceptuel de SI l'ensemble des concepts sur lesquels elle s'appuie.

II-3.1.- Schéma conceptuel de SI, définition, rôle et qualités

II-3.1.1.- Définition

Le schéma conceptuel d'un SI est une expression abstraite de la structure de tous les phénomènes réels de l'organisation à représenter dans le SI.

II-3.1.2.- Rôles du schéma conceptuel

Le schéma conceptuel joue un rôle
- lors de la conception
- lors de l'utilisation du SI

Rôle du schéma conceptuel dans la méthode de conception du SI

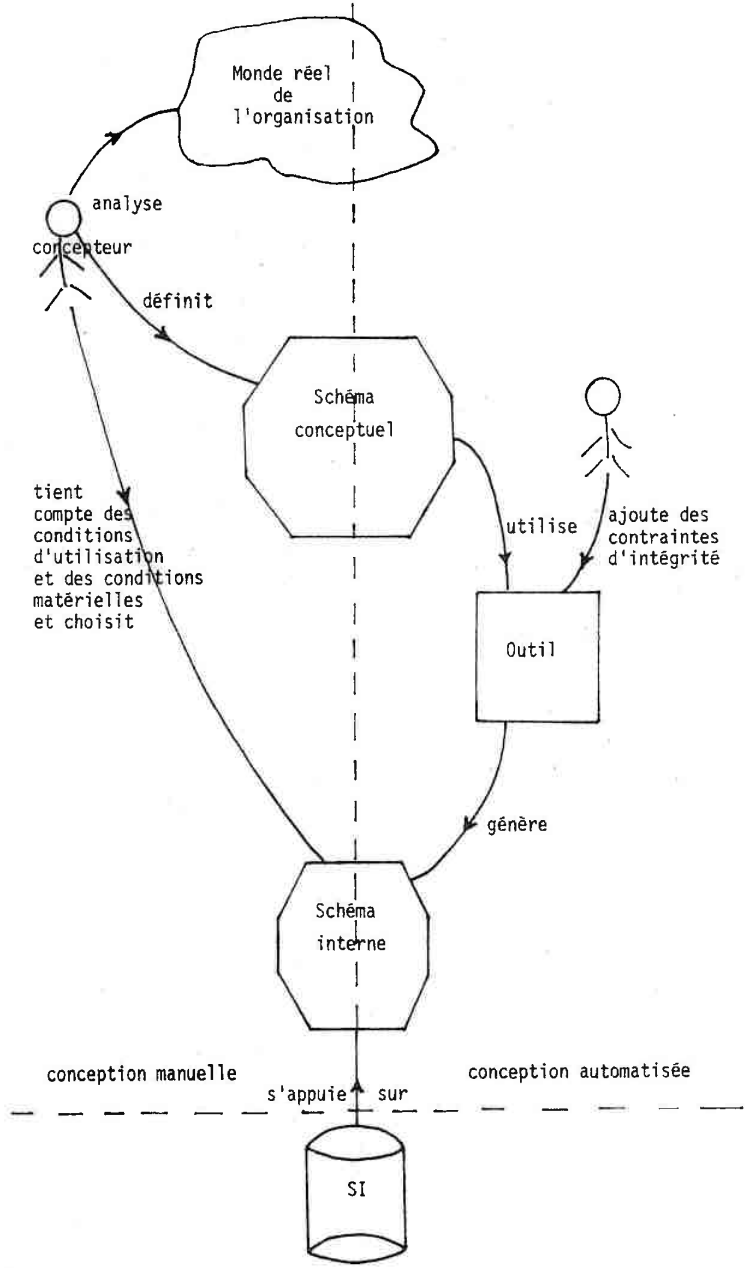
Dans le processus de conception d'un SI, le schéma conceptuel sert de point de départ à la recherche d'une solution technique d'implantation du SI sur un matériel. Nous appelons schéma interne la description de cette solution.

Ce schéma interne peut être défini de deux façons :

- Soit manuellement par un concepteur qui choisit, en plusieurs étapes, éventuellement, comme dans le cadre des méthodes de conception des bases de données (BEN 79), des solutions physiques d'implantation des données, des traitements et des commandes dynamiques, en tenant compte des conditions d'utilisation désirées et des caractéristiques techniques du matériel choisi.

- Soit automatiquement par un outil qui, comme le font les systèmes de gestion de bases de données relationnelles, choisit le schéma interne à partir du schéma conceptuel complété par des contraintes d'intégrité.

La figure ci-contre illustre ces deux méthodes de conception d'un SI.



Rôle du schéma conceptuel lors de l'utilisation et de l'évolution du SI

- Lors de l'utilisation du SI, le schéma conceptuel permet une interprétation sémantique de tout résultat fourni par le SI.

- Lors de l'évolution du SI, due à des modifications des caractéristiques techniques du matériel, il est réutilisable comme point de départ de choix d'une nouvelle solution technique.

- Enfin lors d'une évolution due à une modification de la réalité, extension ou modification de structure, le passage par le schéma conceptuel permet de déterminer plus facilement les impacts de l'évolution réelle sur le SI.

Pour assumer ces rôles, il est nécessaire, que, quelle que soit la méthode de conception du schéma interne employée, manuelle ou automatique, il existe une correspondance entre les structures abstraites du schéma conceptuel et les structures physiques du schéma interne.

L'établissement et surtout l'utilisation de cette correspondance sera plus facile dans une méthode de conception automatisée que dans une méthode manuelle où une très grande liberté et une très grande diversité seraient laissées au concepteur pour le choix du schéma interne.

C'est pourquoi dans l'équipe de recherche, la solution de l'outil a été choisie ($ROL_2$79), (BAN 79), (LEI 80).

En résumé le schéma conceptuel joue un triple rôle :

- lors de la conception du SI, il sert de point de départ pour le choix du schéma interne,
- lors de l'utilisation du SI, il sert de référence pour interpréter sémantiquement tout résultat,
- lors de l'évolution de structure de l'organisation, il sert de base pour définir l'évolution correspondante du SI.

II.3.1.3.- Qualités du schéma conceptuel

Pour assurer de façon efficace ces trois rôles il est nécessaire que le schéma conceptuel possède un certain nombre de qualités.

Dans le cadre du projet REMORA, O. THIERY (THI 76) a proposé une liste de qualités que nous rappelons ici.

- Le schéma conceptuel doit être minimal

La redondance, dans toute construction informatique, représentant un système réel évoluant au cours du temps, est une source potentielle d'incohérences dans la représentation à plus ou moins long terme.

C'est pourquoi l'élimination de la redondance nous apparaît comme un objectif prioritaire dans la conception des SI, comme elle l'est dans la conception des bases de données.

Nous appelons schéma conceptuel minimal une représentation qui permet de construire un SI non redondant.

- Le schéma conceptuel doit être fidèle, cohérent et complet

Afin que l'interprétation sémantique des représentations contenues dans le SI, ou produites par lui, soit possible et correcte, il doit posséder les qualités suivantes :

fidélité : Cette qualité est difficilement formalisable. Elle exige que la représentation ne comprenne ni biais, ni déformation de la réalité et que les échelles y soient respectées le plus exactement possible.

cohérence : Une représentation est cohérente si elle ne comprend pas de contradictions, ou s'il n'est pas possible d'en déduire des images contradictoires.

complétude : La complétude que nous assignons au schéma conceptuel, n'est pas synonyme d'exhaustivité c'est-à-dire que nous n'exigeons pas que toutes les caractéristiques de tous les phénomènes soient représentées dans le SI. Cependant, nous pensons que les différentes visions qu'ont des utilisateurs différents d'un même phénomène réel doivent être présentes dans le SI.
Le schéma conceptuel sera donc complet s'il comprend, ou peut restituer aux utilisateurs les différentes visions qu'ils ont des phénomènes réels qui sont utiles à la gestion de l'organisation.

- Le schéma conceptuel doit être souple

Afin de faciliter la prise en compte des évolutions de structure de l'organisation, le schéma conceptuel doit posséder une certaine souplesse c'est-à-dire qu'il doit être exprimé sous une forme aisément modifiable.

- Le schéma conceptuel doit être indépendant des usages

Rappelons qu'un des reproches que nous avons fait aux SI actuels est le suivant : la représentation des phénomènes qu'ils donent, en particulier au niveau de la dynamique, est déformée, car elle tient compte des conditions particulières d'utilisation du SI.

Afin de remédier à cet inconvénient, il faut rendre le schéma conceptuel indépendant des usages.

Ceci signifie, notamment au niveau de la représentation de la dynamique, que le schéma conceptuel doit donner une représentation des règles de gestion, dépouillée de toutes leurs modalités particulières d'application, liées aux conditions matérielles de fonctionnement de l'organisation.

Nous appelons schéma conceptuel canonique, un schéma conceptuel possédant l'ensemble de ces qualités.

II-3.2.- Modèle conceptuel de SI

II.3.2.1.- Définition

Un modèle conceptuel de SI est un ensemble de concepts permettant d'exprimer des schémas conceptuels de SI canoniques.

II.3.2.2.- Ce que doit permettre de représenter le modèle : les composantes de la modélisation

Compte tenue de l'analyse des SI que nous avons faite dans la première partie, il apparaît que le schéma conceptuel

doit comprendre :
- l'expression abstraite des données,
- l'expression abstraite des traitements,
- l'expression abstraite des commandes dynamiques.

doit permettre : - la prise en compte du temps au niveau des trois expressions,
- l'intégration des trois représentations dans un schéma unique.

Les composantes de la modèlisation sont donc
- les données
- les traitements
- les commandes dynamiques
- le temps

II.3.2.3.- Les caractéristiques du modèle

Afin d'assurer au schéma conceptuel les qualités que nous lui avons assignées, il est nécessaire que le modèle possède les deux caractéristiques suivantes : formel, orienté conception.

1.- Le modèle doit être formel

Pour assurer la minimalité de la représentation, pour vérifier sa cohérence ou sa complétude, il est nécessaire que les concepts du modèle soient définis formellement.

Pour ce faire deux possibilités s'offrent

- ou bien les concepts sont définis par rapport à un formalisme préexistant,
- ou bien les concepts sont définis par un ensemble de règles constituant un système formel.

L'usage d'un formalisme dans une modèlisation permet d'appliquer à la représentation abstraite les résultats théoriques qui lui sont associés et, par là, de démontrer certaines de ses propriétés.

Par exemple, dans un modèle de données, l'utilisation de concepts définis par référence au concept de dépendance fonctionnelle (COD 70), permet d'appliquer à la modèlisation correspondante les résultats théoriques associés à ce concept (DEL 73), (LEW 77).

De même l'utilisation de prédicats du premier ordre (SHO 67) dans un langage de spécification présente l'avantage de pouvoir appliquer aux spécifications ainsi obtenues les théorèmes du calcul des prédicats du premier ordre.

2.- Le modèle doit être "orienté conception"

Le caractère formel du modèle n'est pas suffisant pour garantir la qualité d'"indépendance par rapport à l'usage" du schéma conceptuel.

Par exemple, les réseaux de Petri (PET 66) constituent un modèle formel conçu pour décrire des systèmes en fonctionnement, c'est-à-dire des solutions à des problèmes de conception de systèmes dynamiques, et sont donc en conséquence mal adaptés à l'expression de schémas conceptuels aux sens où nous les avons définis.

Un modèle conceptuel doit donc à notre avis posséder une qualité particulière que nous appelons "orientation conception".

Nous entendons par là que les concepts du modèle, doivent être tels, qu'ils aident le concepteur, à extraire de l'analyse de la réalité ou de l'énoncé du problème à résoudre, le noyau minimum qui la ou le caractérise.

Un modèle peut posséder plus ou moins cette qualité.

Dans l'analyse des modèles que nous allons faire dans le paragraphe suivant, nous déduisons cette qualité de l'objectif assigné au modèle ou de l'utilisation la plus courante qui en est faite.

En conclusion notre politique de conception des SI nécessite un modèle formel, orienté conception, permettant l'expression de schémas conceptuels canoniques.

Ce modèle doit permettre de représenter, à un niveau abstrait, de façon indépendante des usages, - les données
- les traitements
- les commandes dynamiques,

de manière intégrée et par rapport au temps.

Dans le paragraphe suivant nous effectuons une analyse des modèles existants à l'heure actuelle pour concevoir des SI, de façon à mettre en évidence les problèmes de représentation résolus et ceux restant à résoudre.

III - ANALYSE DES MODELES EXISTANT ACTUELLEMENT POUR LA CONCEPTION DES SI

III-1.- Grille d'évaluation des modèles

Pour étudier et comparer les modèles proposés ou utilisés à l'heure actuelle pour la conception des SI, nous avons croisé dans un tableau

- d'une part les caractéristiques de la modélisation auquel nous avons ajouté la prise en compte du temps,
- d'autre part les composantes de la modélisation.

	Prise en compte du temps	formalisation	orientation conception
données			
traitements			
commandes dynamiques			
intégration des trois composantes			

Pour faciliter l'usage et l'intégration de la grille nous avons mis le temps en colonne.

Cette grille permet de multiples combinaisons.

Elle permet de définir des modèles qui prennent en compte une ou plusieurs composantes de façon différente.

Par exemple, un modèle de SI peut proposer une représentation des données formelle et une représentation des traitements qui ne l'est pas.

Cette grille définit en fait 10 critères précis pour étudier un modèle de SI sur lesquels nous allons nous appuyer pour situer les modèles les uns par rapport aux autres.

III-2.- Etude des modèles de SI

Nous avons classé les modèles en trois groupes suivant la composante qui y était représentée de façon originale, nous distinguons donc - les modèles de données,
- les modèles orientés traitement,
- les modèles orientés dynamique .

III-2.1.- Les modèles de données

Dans cette partie nous distinguons

- le modèle infologique de LANGEFORS (LAN 73) qui a son caractère propre et qui est antérieur à tous les autres.
- les modèles conceptuels de données développés depuis 1970 et maintenant largement utilisés.
- les approches utilisant les types abstraits.

III-2.1.1.- Le modèle infologique

L'approche infologique des SI, introduite par LANGEFORS (LAN 73) a eu une diffusion internationale et une influence sur le courant de pensée scandinave en matière de SI. Elle a servi de base à de nombreux travaux de son équipe, notamment ceux de SUNDGREEN (SUN 73) et par ailleurs a été utilisée dans les travaux d'autres chercheurs comme BUBENKO (BUB 77) ou LINDGREEN (LIN 74).

L'approche infologique est surtout une étude de l'information, qui dissocie l'aspect "infologique" du SI (celui de sa définition à partir des besoins des utilisateurs), de l'aspect "datalogique" (celui de la représentation par des données).

Cette étude de l'information a conduit :

- à faire une représentation de l'information dans le temps
- à faire une représentation élémentaire de l'information.

Le concept de base du modèle est le e-message

" un e-message" est un 3-uplet<x,y,t> ou

- x représente un objet d'une classe
- y représente une propriété de l'objet
- t représente un instant.

La sémantique d'un e-message est la suivante : c'est la représentation de la valeur d'une propriété d'un objet à un instant donné.

Ce modèle permet une représentation complète de l'historique des données dont la connaissance est, d'après Langefors, fondamentale pour les gestionnaires.

Sur la grille d'évaluation ce modèle satisfait tous les critères de la première ligne :

- Données - temps - formel - orientation conception

SUNDGREEN (SUN 73) a complété ce modèle par le concept de e-processus pour représenter les traitements :

"un e-processus est un processus qui à la fin d'une exécution couronnée de succès produit un e-message toujours du même type".

L'introduction de ce concept conduit à un modèle satisfaisant les critères de la seconde ligne de la grille.

Cependant il semble d'après les publications dont nous pouvons avoir connaissance que la description conceptuelle des traitements n'a pas connu la même diffusion et la même utilisation que celle des données.

En conclusion ce modèle présente deux particularités que nous ne trouvons pas dans les autres modèles de données :

- la prise en compte du temps,
- l'élémentarité des concepts.

III-2.1.2.- Les modèles conceptuels de données

Une importante communauté scientifique s'est intéressée à la modèlisation des données depuis 12 années environ, soit depuis le développement des premiers systèmes de gestion de base de données.

Les travaux concernant ce domaine ont été très nombreux, MOHAN (MOH 78) en a effectué une classification par thèmes : modèles, langages, système de gestion de base de données qui recense 415 publications entre les années 1974 et 1978.

Ici nous nous intéressons uniquement aux travaux concernant la modèlisation des données au niveau abstrait appelé conceptuel.

Parmi ceux-ci, beaucoup sont consacrés au modèle relationnel.

Le modèle relationnel, introduit par Codd en 70, dont nous rappelons les caractéristiques au chapitre 3, est un modèle qui satisfait aux critères de la première ligne de la grille d'évaluation à l'exception du critère "prise en compte du temps" (COD 70, 71, 74).

Parmi les autres modèles conceptuels de données, on distingue :

- les modèles dérivés du modèle relationnel comme le modèle entité-relation de CHEN (CHE 76), le modèle de FLORY (FLO 77), qui apportent des améliorations au niveau de l'interprétation sémantique de la modèlisation, par rapport au modèle relationnel, en particulier ils permettent de distinguer, dans le schéma conceptuel, les relations représentant des objets de celles représentant des associations d'objets.

- les modèles binaires qui consistent en une modèlisation ne comprenant que des associations binaires entre ensembles ABRIAL (ABR 74), BRACCHI (BRA 76), BUBENKO (BUB 77), SENKO (SEN 75), CREHANGE (CRE 75), FALKENBERG (FAL 75), ont proposé des modèles de ce type.

- les modèles reposant sur la logique et le calcul des prédicats du premier ordre.

La logique a d'abord été utilisée pour définir des langages d'interrogation et de manipulation de données. Une étude de ces langages a été effectuée en particulier par PIROTTE (PIR 76).

Certains auteurs proposent maintenant de l'utiliser pour exprimer le schéma conceptuel lui-même qui se présente alors sous la forme d'une collection de formules d'un système formel basé sur le

calcul des prédicats du premier ordre. Citons dans ce domaine les travaux de NICOLAS (NIC 79), de WARREN (WAR 81) et l'article synthétique de GALLAIRE (GAL 81) sur l'utilisation de la logique dans le domaine des bases de données.

Parmi tous ces modèles, certains prennent en compte le temps ou les traitements et nous intéressent donc plus particulièrement ici.

- Modèles prenant en compte le temps

Une étude de la prise en compte du temps dans la modélisation des données a été faite par BUBENKO (BUB 77). Dans celle-ci, BUBENKO propose, en outre, un modèle comprenant, en plus des concepts propres à la modélisation des données (entité, association, propriété) le concept d'événement défini de la façon suivante :

"un événement est une observation ou une décision, survenant dans le système objet au temps t, du commencement ou de la fin d'une occurrence d'une association".

Il définit également la notion de modèle abstrait de SI ainsi :

"un modèle abstrait est un ensemble d'assertions concernant des événements du système objet, un ensemble de règles de déduction et un ensemble de conclusions qui peuvent être tirées sur l'existence d'entités et d'associations à tous instants significatifs pour le système (historique, présent ou futur".

Bien que cette définition semble se rapprocher de notre notion de schéma conceptuel de SI, l'auteur ne propose, dans cet article, qu'un mode de représentation des états successifs pris par les données au cours du temps et ne propose pas de représentation abstraite et formelle des "règles de déduction" et des "conclusions" qui pourraient correspondre à une modélisation des traitements et de la dynamique à un niveau abstrait.

BUBENKO conclut ce papier en insistant sur l'intérêt de la prise en compte rigoureuse du temps dans la modélisation des données et sur les recherches qui restent à faire dans cette voie.

Notons que FALKENBERG (FAL 75) et BREUTMANN (BRE 79) intègrent le temps dans la modélisation des données d'une manière assez voisine de celle de BUBENKO en considérant que "toute association a un début et une fin considérés comme des événements sur l'axe des temps".

- Modèles intégrant les traitements

- Le modèle DATA-SEMANTICS D'ABRIAL (ABR 74) permet, grâce à un langage associé au modèle de données, la description des traitements à un niveau abstrait et satisfait ainsi aux critères de la seconde ligne, il s'agit cependant plus d'une approche langage que d'une approche modèle.

- DELOBEL (DEL 73) complète son modèle de données par quelques propositions qui permettent de représenter la dynamique d'un SI suivant les concepts du modèle de FORRESTER (FOR 62).

III-2.1.3.- Représentation des traitements et de la dynamique dans un schéma conceptuel exprimé à l'aide d'un modèle conceptuel de données

Dans un schéma conceptuel exprimé à l'aide d'un modèle conceptuel de données, comme le modèle relationnel ou le modèle entité-relation, une certaine représentation des traitements et de la dynamique est souvent effectuée par l'intermédiaire des contraintes d'intégrité.

Les contraintes d'intégrité sont des prédicats qui définissent des propriétés que doivent vérifier les données de la base pour assurer le maintien de sa cohérence lors de l'exploitation de la base de données.

Ces contraintes sont dites statiques si elles portent sur l'état courant de la base de données et dynamiques lorsqu'elles font référence à un changement d'état de la base.

Les contraintes statiques permettent de spécifier certains traitements en indiquant,par exemple,quelle doit être la formule qui doit toujours être vérifiée entre certains constituants.

Exemple

Au schéma de relation

facture (nofac,montanthtaxe,tva,montanttotal)

on pourra associer la contrainte d'intégrité

montanttotal = montanthtaxe+tva

Les contraintes dynamiques,quant à elles,permettent de définir quelques éléments de la dynamique du SI.

Elles peuvent exprimer par exemple :

- les changements d'état permis ou interdits pour un type d'élément,
- le sens de variation obligatoire d'une grandeur,
- le traitement à exécuter lorsqu'un certain type de changement d'état se produit.

Exemples

Au schéma de relation

Personne (nopers,indice, situation-familiale ...)

on peut associer

- une contrainte d'intégrité qui donne,sous forme d'un ensemble de couples de valeurs, quels sont les changements de valeur possibles pour le constituant situation-familiale.
- une contrainte qui indique que la valeur du constituant indice pour une personne donnée ne peut que croître.
- une contrainte qui définit le traitement à exécuter lorsqu'on insère ou supprime un nuplet dans la relation Personne.

Le langage SEQUEL associé au SGBD relationnel System R (CHA 74) permet l'expression de contraintes de ce type par l'intermédiaire des instructions ASSERT et TRIGGER.

Le langage UDL (unified data langage) proposé par DATE (DAT 80) comprend également des clauses permettant l'expression de ce genre de contraintes.

Notons aussi que,dans les modèles basés sur la logique, il est possible d'écrire sous forme de prédicats n'importe quelle contrainte et en particulier les contraintes dynamiques.

Ce mode d'expression des traitements et de la dynamique présente à notre avis plusieurs inconvénients :

- il ne comporte pas de concepts précis permettant d'appréhender l'ensemble des phénomènes réels à décrire par les contraintes,
- il conduit à une collection de contraintes juxtaposées qui ne donnent pas une vue d'ensemble de la dynamique du système représenté,
- il rend difficile les contrôles nécessaires concernant la complètude ou la cohérence de la représentation effectuée.

Pour ces raisons, nous considérons que les modèles de données ne satisfont pas au critère "formalisation" de la dynamique ni au critère "intégration des représentations des aspects statiques et dynamiques".

III-2.1.4.- Les approches utilisant les types abstraits

Les types abstraits, dont nous avons donné une définition au paragraphe II-2.2., sont utilisés par plusieurs auteurs pour spécifier un SI, c'est-à-dire dans notre terminologie, pour exprimer un schéma conceptuel de SI.

Citons, par exemple, les travaux de EHRIG et WEBER (EHR 78), de BRODIE (BRO 81), de SALES (SAL 80) et le projet TYP (CHA 80),(DER 81).

Les types abstraits apparaissent bien adaptés à la spécification des SI puisqu'ils permettent de définir à un niveau abstrait indépendant de toute considération technique :

- les données par une collection de types,
- les traitements par l'intermédiaire des opérations liées aux types,
- certains aspects de la dynamique comme par exemple les conditions d'application des traitements par l'intermédiaire des pré-conditions et des post-conditions associées aux opérations.

En revanche, ces approches ne permettent pas une spécification complète de la dynamique à un même niveau de formalisme que celui utilisé pour les données et les traitements. BRODIE,par exemple, la décrit par l'intermédiaire de TRANSACTIONS qui sont en fait la spécification de programmes utilisant les opérations définies dans les types. Il manque,dans cette spécification de la dynamique, une représentation des interactions pouvant exister entre les transactions.

En conclusion, les approches utilisant les types abstraits satisfont aux deux premières lignes de la grille d'évaluation et partiellement à la quatrième, c'est-à-dire qu'elles permettent une représentation intégrée et formelle des traitements et des données dans une orientation conception mais ne conduisent pas à une représentation équivalente de la dynamique dans l'état actuel de leur développement.

III-2.2.- Les modèles orientés traitement

Nous regroupons dans cette classe
- les modèles des méthodes d'analyse
- les modèles plus récents, dits de SI, comprenant un modèle conceptuel de données emprunté à d'autres propositions.

III-2.2.1.- Les modèles des méthodes d'analyse

Les méthodes d'analyse sont apparues avant les bases de données et les modèles conceptuels de données. Leur objectif était d'aider à la conception des SI, organisés en chaînes de traitement indépendantes utilisant un pool de fichiers.

Ces méthodes dont plusieurs études comparatives ont été effectuées (CXP 74), (HAB 75), proposent de suivre,pour concevoir tout ou partie d'un SI,un certain nombre d'étapes dans un ordre déterminé.

Les dénominations les plus courantes de ces étapes sont :
- étude de l'existant

- analyse fonctionnelle
- analyse organique

L'étude de l'existant est une description informelle de l'application à automatiser, tenant compte des conditions particulières de fonctionnement de l'organisation avant l'introduction de l'automatisation.

L'analyse fonctionnelle est une définition générale de l'application automatisée à réaliser, indépendante des caractéristiques techniques du matériel à utiliser, mais dépendante des besoins exprimés par ses utilisateurs futurs.

L'analyse organique est une définition détaillée de la solution proposée au niveau de l'analyse fonctionnelle, tenant compte des caractéristiques du matériel utilisé pour la mettre en oeuvre.

Remarquons que la description abstraite correspondant au schéma conceptuel du SI, correspond à un niveau de représentation intermédiaire entre l'étude de l'existant et l'analyse fonctionnelle.

Ceci a pour conséquence que les modèles inclus implicitement dans les méthodes d'analyse pour décrire, en particulier, les traitements et les commandes dynamiques, ne peuvent convenir comme modèle conceptuel de SI.

Au niveau de l'étude de l'existant, ils ne satisfont pas au critère "formalisation".

Au niveau de l'analyse fonctionnelle, ils ne satisfont pas au critère "orientation conception".

Notons cependant que,dans ce domaine, comme dans celui des bases de données, une évolution est apparue récemment qui consiste à incorporer dans les méthodes d'analyse les résultats obtenus dans le domaine de la modélisation des données et à chercher une représentation équivalente des traitements et de la dynamique. Le projet MERISE (HEC 80) dont nous étudions l'un des modèles au paragraphe suivant est un exemple de cette évolution.

III-2.2.2.- Les modèles de SI orientés traitement

Les insuffisances des méthodes d'analyse pour

résoudre efficacement les problèmes de la conception des SI, ont entraîné des recherches dans ce domaine qui ont été surtout orientées vers la proposition d'outils, c'est-à-dire de système permettant une certaine automatisation de l'analyse.

Parmi ces propositions, le projet MACSI (PEC 75) et le projet LAPAGE (CAV 79) incluent un modèle de SI qui se présente comme un modèle de données complété par un modèle de traitement original.

Dans MACSI, le modèle de données est très voisin du modèle DATA-SEMANTICS d'ABRIAL, dans LAPAGE, il s'agit du modèle de FLORY.

Le modèle de traitement de MACSI comprend les concepts de commande, action, opérateur, module, pour lesquels seule une définition sémantique est donnée. L'absence de définition formelle s'explique par l'orientation usage qui est faite de ce modèle pour la réalisation de prototypes de SI qui doivent prendre en compte les conditions d'utilisation futures du système réel à réaliser.

Le modèle de traitement de LAPAGE comprend les concepts de fonction logique, processeur norme de fonctionnement, événement, pour lesquels nous pouvons faire les mêmes remarques que pour les concepts de MACSI.

De plus ces deux modèles ne conduisent pas à une représentation intégrée des données et des traitements car il n'y existe pas de formalisation des liens qui existent entre ces deux composantes de la modélisation.

III-2.3.- Les modèles orientés dynamiques

Depuis 1977, la préoccupation de la communauté scientifique s'intéressant à la modélisation des données se déplace vers la recherche de modèles plus généraux intégrant la représentation des traitements et de la dynamique.

On peut même considérer maintenant que la nécessité de représenter conjointement dans le schéma conceptuel d'un SI les données et leur dynamique est admise par une grande partie de cette communauté. Les conclusions du rapport préliminaire du groupe WG 3 de l'ISO (ISO 81) sont significatives de cette tendance.

Ce groupe de travail, créé en 1977, dans l'objectif de définir des normes pour le domaine des bases de données, définit la dynamique comme étant "la façon dont les éléments changent et les raisons pour lesquelles ils changent" et considère que sa représentation fait partie intégrante d'un schéma conceptuel.

Remarquons cependant que le groupe ne propose pas de normes pour la représentation de la dynamique mais se contente de donner les caractéristiques de quelques approches existant à l'heure actuelle sans faire de comparaison entre elles, ni porter de jugement quant aux qualités de la représentation de la dynamique qu'elles permettent.

Les approches permettant une représentation de la dynamique y sont classées en :

- approches orientées état
- approches orientées commande
- approches orientées interaction.

Le premier type d'approche correspond à une expression de la dynamique sous la forme d'une collection de contraintes d'intégrité dynamiques du type de celles que nous avons présentées au paragraphe III-2.1.3.

Le second type d'approche correspond aux modèles orientés traitement que nous venons d'étudier.

Le troisième type d'approche concerne des modèles qui permettent de décrire les interactions entre l'organisation à modéliser et son environnement ainsi que les liens de causalité qui existent entre les changements d'état qu'elle subit et les traitements à exécuter.

Ce sont ces modèles que nous présentons dans ce paragraphe.

Le modèle proposé dans ce travail appartient à ce type d'approche et a été développé par l'équipe REMORA parallèlement à la plupart d'entre eux.

Nous présentons ici :

- la proprosition de BENCI et al (BEN 76) qui apparaît comme

précursive au développement des autres approches et qui a servi de base pour la plupart d'entre elles,

- quelques modèles ayant fait l'objet de publications détaillées et assez voisins de celui que nous présentons
  - la proposition de BRACCHI et aL (BRA 79)
  - le modèle conceptuel de MERISE (MER 79)
  - le modèle de DSL (BOD 79)
  - le modèle de MAESTRO. (DUF 80)

Nous citons ensuite plus rapidement quelques autres approches récentes ayant les mêmes objectifs ou des objectifs voisins de représentation de la dynamique et qui utilisent le formalisme des réseaux de Petri.

La proposition de BENCI, BODART, BOGAERT et CABANES (BEN 76)

Dans cette proposition, les auteurs définissent le schéma conceptuel comme l'expression dans un certain langage de l'organisation conceptuelle.

Cette organisation conceptuelle comprend :

- la structure conceptuelle, c'est-à-dire le schéma conceptuel de la base de données complété par des contraintes d'intégrité.
- les règles d'évolution qui décrivent la manière dont les données évoluent au cours du temps.

Le modèle utilisé pour exprimer la structure conceptuelle est un modèle de type "entité-relation".

Les règles d'évolution définissent les actions à exécuter sur les données lorsque certains événements se produisent, ainsi que les contraintes d'intégrité qui doivent être vérifiées pour que les règles puissent s'appliquer. Les concepts utilisés pour exprimer ces règles sont ceux d'événements, d'opérationset de contraintes d'intégrité.

Un événement est défini comme "une décision qui déclenche l'exécution d'opérations sur les données".

La liaison entre la modélisation des données et celle de la

dynamique est effectuée par l'intermédiaire de paramètres associés aux événements ; ces paramètres sont des relations de la structure conceptuelle.

Notons aussi que les auteurs insistent sur l'importance de la représentation du temps sans toutefois proposer un moyen précis pour le prendre en compte dans la modélisation.

En conclusion, cette proposition satisfait en grande partie ou presque à l'ensemble des critères de la grille d'évaluation : elle permet une représentation intégrée des données des traitements et de la dynamique et insiste sur la nécessité d'y incorporer le temps.

Il lui manque cependant une formalisation des concepts permettant la prise en compte des traitements et de la dynamique.

La proposition de BRACCHI, FURTADO, PELAGATTI (BRA 79)

Cette proposition est assez voisine dans son esprit de la précédente.

Pour ces auteurs le schéma conceptuel d'un SI comprend trois sous-schémas :

- le schéma des données défini classiquement par des entités, des relations, des attributs et des contraintes statiques.
- le schéma fonctionnel défini par des contraintes dynamiques.
- le schéma d'évolution défini par des événements et des règles de modification des données et du schéma fonctionnel.

Ces règles, comme les règles d'évolution de la proposition précédente, précisent les transactions à exécuter lorsque certains événements se produisent. Les transactions sont des combinaisons d'opérations correspondant à des types de modifications élémentaires des données.

Une analyse détaillée des contraintes dynamiques est effectuée dans cette proposition.

Les auteurs y soulignent, à ce propos, la nécessité de conserver dans le SI les états successifs pris par les données afin de pouvoir les vérifier.

Cette proposition, comme la précédente, répond à l'ensemble des critère d'un modèle conceptuel de SI à l'exception de celui de la formalisation des concepts représentant la dynamique critère nécessaire pour garantir et contrôler les qualités d'un schéma conceptuel du SI.

Le modèle conceptuel de MERISE (MER 79)

MERISE est une méthode de conception de systèmes d'information développée par le CETE d'AIX en PROVENCE et des sociétés de service pour le compte du Ministère de l'Industrie.

Cette méthode, comme les méthodes de construction de bases de données, propose de construire un SI en trois étapes : conceptuelle, logique et physique.

Elle comprend pour chaque étape un modèle de données, un modèle de traitement, une méthode et des outils.

Nous ne nous intéressons ici qu'aux modèles du niveau conceptuel.

Le modèle conceptuel de données est le modèle individuel qui est un cas particulier du modèle entité-relation. Ces concepts sont :

INDIVIDU
RELATION
PROPRIETE

Le modèle conceptuel de traitement comprend les concepts suivants :

EVENEMENT
RESULTAT
OPERATION
SYNCHRONISATION

qui s'articulent ainsi :

Un ensemble d'événements déclenche, après combinaison par une synchronisation, une opération qui produit un ensemble de résultats en sortie.

La définition de ces concepts dans (HEC 80) est la suivante :

EVENEMENT

Un événement est la représentation d'un fait nouveau pour le système d'information.

OPERATION

Une opération est la représentation d'un ensemble de traitements effectués dans le SI en réaction à un stimulus.

Le stimulus peut être un événement unique ou une synchronisation portant sur plusieurs événements.

RESULTAT

Un résultat est une représentation de la réponse codifiée du système d'information, produite par une opération.

SYNCHRONISATION

La synchronisation représente une précondition pour l'activation d'une opération à partir de plusieurs événements.

A ce modèle est associée une représentation graphique qui permet de visualiser les interactions entre opérations, événements et résultats.

De plus, il a été montré (HEC 80) qu'il était possible d'établir une correspondance entre le modèle de traitement et les réseaux de Petri, ce qui permet de le formaliser et de lui appliquer les résultats théoriques associés aux réseaux de Petri.

Vis à vis de nos critères d'évaluation, ce modèle permet une représentation formelle des données, des traitements et de la dynamique dans une orientation conception. Il ne prend pas en compte explicitement le temps mais surtout ne conduit pas à notre avis à une représentation véritablement intégrée des trois composantes de la modélisation dans la mesure où les définitions des concepts du modèle de traitement ne fait pas référence à ceux du modèle de données.

Le langage DSL

Le langage DSL (Dynamic specification language) est un langage de spécification dérivé du langage PSL (problem specification langage) et développé à l'Université de NAMUR par l'équipe de F. BODART (BOD 79).

PSL est un langage qui a été développé à l'Université de MICHIGAN dans le cadre du projet ISDOS dont l'objectif était la définition d'un système de conception et d'optimisation de systèmes d'information (TEI 73).

Les concepts du modèle conceptuel de SI inclus dans DSL sont très proches de ceux du modèle de MERISE, ils se partagent en :

- un modèle conceptuel de données qui est le modèle entité relation de CHEN.
- un modèle de traitement comprenant les concepts de : TRAITEMENT PROCESSUS MESSAGE EVENEMENT ET POINT DE SYNCHRONISATION.

Les définitions que nous donnons ici sont tirées de (BOD 81).

TRAITEMENT

Un traitement est un ensemble d'opérations, structurées selon des règles logiques et exécutées par des processeurs d'information.

PROCESSUS

Une occurrence d'exécution des opérations constituant un traitement est appelé PROCESSUS.

MESSAGE

Un MESSAGE contient les informations relatives à une occurrence d'un événement survenant, soit dans l'environnement du SI, soit à l'intérieur du SI.

EVENEMENT

Un événement est un changement d'état du système porté à la connaissance d'un observateur uniquement par l'intermédiaire d'un message.

POINT DE SYNCHRONISATION

Le point de synchronisation permet la spécification de deux types de structure de contrôle du flux de messages entre processus : la jointure et l'éclatement conditionnel. Ces deux structures peuvent être combinées dans un même point de synchronisation.

Jointure : la jointure correspond à une combinaison de changement d'état élémentaires (événement) du SI. Elle est définie par un ensemble d'événements et une condition de synchronisation qui explicitent la combinaison des événements contribuant à la synchronisation.

Eclatement conditionnel : Quand la réalisation d'une synchronisation s'est produite, en fonction de l'état d'une condition d'éclatement, un choix est effectué parmi deux actions dynamiques.

Les actions dynamiques concernent le déclenchement, l'interruption, le redémarrage, la destruction de processus.

Notons qu'à ce modèle sont associés un modèle de ressources et un système permettant d'évaluer et de simuler le fonctionnement du SI à construire.

A ce niveau figure une certaine prise en compte du temps qui permet de donner les fréquences de réalisation d'événements et de point de synchronisation.

En conclusion, ce modèle présente les mêmes qualités que le modèle de MERISE : la représentation des trois composantes de la modélisation dans une orientation conception. Malgré la présence d'un métamodèle explicitant les liens entre les concepts du modèle de traitement et du modèle de données, l'intégration des spécifications statiques et des spécifications dynamiques ne nous semble pas encore véritablement atteinte.

Le modèle de MAESTRO

L'objectif du projet MAESTRO développé par DUFOURD (DUF 80) est la construction de maquettes pour évaluer les systèmes d'information des organisations.

L'évaluation proposée concerne la structure du SI et son fonctionnement.

L'évaluation de la structure est de type syntaxique et ne diffère pas de l'évaluation associée à toute modèlisation.

L'évaluation du fonctionnement est effectuée par l'intermédiaire de simulations et peut être soit qualitative soit quantitative. Les simulations permettent d'obtenir un certain nombre de résultats concernant le comportement du système au cours du temps qui peuvent être très utiles aux gestionnaires.

Le modèle de SI inclus dans ce projet utilise pour la modèlisation des données, la formalisation proposée par PAIR (PAI 74) et REMY (REM 74) et utilisée également par CREHANGE (CRE 75).

Cette formalisation repose sur les résultats associés aux systèmes formels et à la logique.

Pour la modèlisation des traitements et de la dynamique, les concepts retenus sont tirés de ceux utilisés pour décrire le fonctionnement des systèmes d'exploitation (CRO 75).

Un système d'information est perçu comme un système de processus coopérants par l'intermédiaire de données communes partagées et concurrents vis à vis d'un ensemble de ressources.

Ce modèle permet une représentation formelle de chacune des trois composantes de la modèlisation et assure une prise en compte du temps au niveau de la dynamique qui lui permet de simuler le fonctionnement du système modèlisé.

Cependant, il ne possède pas, à notre avis, le critère "orientation conception" dans la mesure où il prend en compte dans la spécification les conditions techniques humaines ou financières de fonctionnement du SI par l'intermédiaire des "ressources".

Approches utilisant les réseaux de Petri

Pour représenter la dynamique d'un système d'information, plusieurs approches utilisent le formalisme des réseaux de Petri (PET 66).

Un réseau de Petri est un formalisme permettant de décrire

le fonctionnement d'un système.

Les concepts d'un réseau de Petri sont la place et la transition. Un réseau est un graphe biparti $G = (P,T,\Gamma)$ dont les sommets appartiennent soit à l'ensemble des places P, soit à l'ensemble des transitions T. Les arcs orientés sont définis par la relation binaire $\Gamma(P\times TUT\times P)$: ils relient une place à une transition ou une transition à une place.

Les réseaux de Petri ont fait l'objet de nombreuses recherches théoriques et ont été utilisés pour la modèlisation de divers types de systèmes.

Plusieurs auteurs ont eu l'idée de les utiliser pour modèliser la dynamique des SI. Citons :

- FLORY et KOULOUDMIDJAN (FLO 79) qui attachent une sémantique aux concepts de place et de transition dans le cadre d'un modèle de SI assez voisin de celui de la proposition de BENCI et aL (BEN 76).

- LEONARD (LEO 81) dont l'objectif n'est pas directement la conception des SI mais la recherche d'une structure de données qui autorise le maximum de parallèlisme.

- DE ANTELONELIS et ZONTA(DEA 81) qui,eux aussi, donnent une sémantique précise aux concepts de place et de transition dans le contexte d'un modèle de la dynamique basé sur le concept d'événement suivant :

> "un événement exprime un changement de conditions entre des conditions vérifiées "auparavant" (pré-conditions) et des conditions vérifiées "après" (post-conditions). Il est décrit par ses pré-conditions et les actions qui produisent le changement (les post-conditions sont la conséquence de ces actions) ".

Ce modèle conduit à une représentation formelle des traitements et de la dynamique mais qui ne s'intègre pas avec une représentation formelle des données.

CONCLUSION

De cette étude, qui s'est limitée aux résultats proposés dans le domaine des bases de données et des systèmes d'information, nous tirons les conclusions suivantes :

- Il existe beaucoup de modèles permettant une représentation formelle des données dans une orientation conception, parmi lesquels seuls quelques uns prennent en compte le temps.

- Les modèles permettant une représentation formelle des traitements et de la dynamique sont moins nombreux, ne sont pas tous "orientés conception" et ne prennent pas tous en compte le temps.

- Il existe très peu de modèles réalisant une véritable intégration des représentations des trois aspects (donnée, traitement, dynamique).

- La formalisation la plus poussée est rencontrée dans les approches utilisant les types abstraits.

Ainsi, parmi les quatre problèmes de représentation posés par la conception des SI, les problèmes de la prise en compte du temps et celui de l'intégration de la représentation des aspects statiques et des aspects dynamiques ne sont pas résolus par les modèles actuels.

Le modèle conceptuel de SI que nous proposons dans cet ouvrage a pour objectif d'apporter une solution à ces problèmes.

Cette solution, dans la forme où elle est présentée dans cette thèse, n'est qu'une étape dans la recherche de solutions au problème de la spécification d'un système dynamique en général.

Nous assistons actuellement à un rapprochement des préoccupations, en matière de spécification, de domaines de recherches variés tels que la programmation, l'intelligence artificielle, les bases de données.

Nous pensons que de ce rapprochement, permettra de dégager sur la base de la solution proposée ici, d'autres solutions, utilisant certains résultats de ces domaines, plus satisfaisantes, soit sur le plan formel, soit sur le plan sémantique.

DEUXIEME PARTIE

MODELE CONCEPTUEL DE SYSTEME D'INFORMATION

CHAPITRE 2 : MODELE DESCRIPTIF POUR L'ANALYSE D'UNE ORGANISATION

CHAPITRE 3 : MODELE CONCEPTUEL POUR LA STRUCTURATION DES SI

CHAPITRE 2 : MODELE DESCRIPTIF POUR L'ANALYSE D'UNE ORGANISATION

INTRODUCTION

Pour rechercher les concepts utiles à la modèlisation conceptuelle des SI, nous avons utilisé une démarche descendante qui a consisté

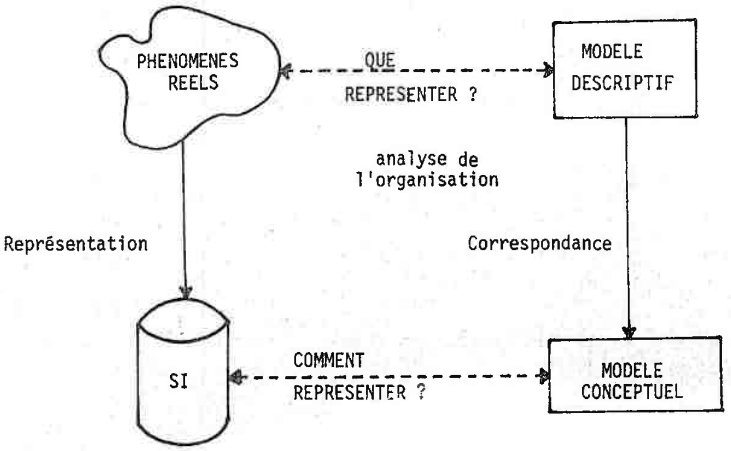
1 - à faire une analyse des organisations afin de définir les catégories de phénomènes réels à représenter et d'étudier leurs structures.

2 - à choisir un mode de représentation formel de ces catégories.

L'analyse du monde réel nous a conduit à définir un modèle que nous qualifions de descriptif, ce modèle sert en effet à décrire une organisation après qu'on l'ait analysée ; il répond à la question : QUE REPRESENTER ?

Le modèle conceptuel répond,quant à lui, à la question:COMMENT REPRESENTER ? compte tenu des objectifs que nous lui avons assignés au chapitre précédent.

La figure ci-dessous illustre les rôles respectifs des deux modèles.



Définition du schéma conceptuel

I - DEFINITION D'UNE ORGANISATION ET INTRODUCTION DES CATEGORIES DE PHENOMENES REELS

I-1.- Définition d'une organisation

A l'origine, en théorie des systèmes, l'organisation est introduite comme la qualité que doit posséder un système pour satisfaire des objectifs. Puis les spécialistes des systèmes ayant défini tout système comme un ensemble d'éléments en intéraction orientés vers la satisfaction d'un objectif, donc comme étant par nature organisé, le nom commun "organisation" a peu à peu remplacé celui de système organisé, lorsque ce système est un système réel du type entreprise.

Nous retenons donc cette définition de l'organisation :

Une organisation est un système réel organisé dans le but de satisfaire un objectif.

Les organisations sont souvent classées suivant la nature de leur objectif et en informatique de gestion nous nous intéressons plus spécialement

- aux entreprises dont l'objectif est de nature économique
- aux administrations pour lesquelles il est de nature sociale.

Parmi les nombreuses définitions de l'organisation vue comme un système nous avons retenu celle de MESAROVIC présentée par FORTET qui nous permet de définir exactement le sous-système de l'organisation auquel nous nous intéressons. (FOR 67)

Pour ces auteurs une organisation comporte deux sous-systèmes

- le sous-système causal qui est composé des unités causales, qui sont, par exemple, les divers processus technologiques, dont les réponses aux perturbations qu'elles subissent sont parfaitement définies par la transformation qui lie leurs entrées et leurs sorties ;
- le sous-système téléologique composé d'unités poursuivant des objectifs, telles des services, des comités ou des hommes de l'entreprise définies par les objectifs qu'elles sont chargées de

poursuivre et leurs moyens d'action sur le système causal.

Beaucoup de travaux sur les organisations portent sur le sous-système téléologique ; leur but est de proposer des modèles permettant de simuler le comportement de l'organisation et d'étudier ses possibilités d'adaptation ou d'auto-organisation lors d'une modification de ses objectifs ou de son environnement. Il apparaît alors que le sous-système causal définit l'organisation pour un objectif donné et un environnement donné. Et la définition même, donnée à ce sous-système, revient à dire que, tant que ces derniers éléments ne sont pas modifiés une organisation est un système dont le comportement est déterministe et dont les constituants évoluent suivant des règles précises, les règles de gestion de l'organisation.

Dans la suite de notre étude, nous nous intéressons au sous-système causal de l'organisation : les modèles de description et de représentation que nous proposons sont valables dans un contexte d'objectif et d'environnement stable.

I-2.- Le modèle descriptif et son rôle

I-2.1.- Définition d'un modèle descriptif

Un modèle est un moyen permettant de représenter une réalité.

Un modèle est dit descriptif si son but est de reproduire la réalité telle qu'elle est ; il est normatif s'il cherche à représenter ce qui devrait être.

Si la réalité que l'on étudie est complexe un modèle descriptif constitue toujours une simplification et même une idéalisation de cette réalité. Cependant il est souvent le seul moyen de comprendre cette réalité.

I-2.2.- Rôle du modèle descriptif de l'organisation dans la conception des systèmes d'information

Une organisation est, par essence même, une réalité complexe. Pour comprendre cette réalité, un modèle descriptif peut être utilisé.

Il s'impose si l'on veut construire des systèmes d'information "significatifs", c'est-à-dire tels que les utilisateurs du SI puissent raisonner avec le SI comme ils le font avec les phénomènes réels. En effet comment construire une représentation partielle d'une réalité si l'on ne possède pas de définition, de description de cette réalité ? Comment interpréter les informations produites par le système d'information si l'on ne possède pas de règles de correspondance entre ces informations et les phénomènes réels ?

Cette nécessité d'un modèle descriptif de l'organisation paraît évidente lorsqu'on la présente ainsi. Cependant il existe de nombreuses méthodes d'analyse qui ne reposent sur aucun modèle descriptif de l'organisation, c'est donc que le besoin d'un tel modèle n'est pas toujours apparu aussi clairement.

Les chercheurs en base de données ont, quant à eux, ressenti ce besoin, en particulier KENT (KEN 76) montre la nécessité de réfléchir et de comprendre la réalité avant d'utiliser un modèle de données pour la représenter, mais ils ont restreint leur analyse à un sous-ensemble de l'organisation.

Le rôle du modèle descriptif est de permettre une correspondance entre le système d'information et la réalité de l'organisation.

I-3.- Analyse dynamique de l'organisation : introduction des catégories

D'après la définition de l'organisation que nous avons donnée et dans le contexte dans lequel nous l'étudions, décrire l'organisation revient à décrire un système causal.

Pour décrire un système causal, il est nécessaire d'utiliser un modèle permettant de décrire l'évolution du système au cours du temps par les causes et les effets de tout changement y survenant. C'est-à-dire qu'il faut que le modèle descriptif soit un modèle dynamique, ainsi appelé par opposition à un modèle statique qui ne décrit que l'état du système à un instant donné.

Pour trouver ce modèle nous nous sommes posé trois questions :

- Par quoi est défini l'état de l'organisation à un instant donné ?
- Qu'est-ce qui modifie l'état de l'organisation ?
- Par quoi sont provoquées ces modifications ?

La réponse à la première question nous est donnée par la théorie des systèmes : un système est un ensemble d'éléments, et l'état du système est défini à tout instant par l'état de chacun de ses éléments.

I-3.1.- La catégorie objet

Les éléments de l'organisation, que nous regroupons sous le terme d'objets, sont de diverses natures ; nous trouvons

- des choses réelles qui y sont présentes,
  Ex : des produits, des machines
- des hommes qui y travaillent ou sont en relation avec elle ,
  Ex : des employés, des clients, des fournisseurs
- des objets abstraits, immatériels utilisés pour la gestion de l'entreprise
  Ex : les comptes, les salaires.

Le point commun entre tous ces objets est leur appartenance pour un certain laps de temps à l'organisation, ce qui en fait des éléments du système organisation.

L'état de l'organisation à un instant donné est donc défini par l'état de tous ses objets.

I-3.2.- La catégorie opération

Une autre manière d'analyser un système est de le décomposer en sous-systèmes, chaque sous-système possédant son propre objectif. Pour une organisation, les objectifs de ses sous-systèmes sont ce que l'on appelle couramment les fonctions de l'entreprise : produire, facturer, livrer, calculer les salaires, etc...

Pour réaliser ces fonctions,l'organisation utilise des moyens qui sont certains objets de l'organisation, des hommes, des machines, et fournit du travail c'est-à-dire exécute des opérations : opération de production, opération de livraison, etc.. Le résultat de l'exécution de ces opérations est, soit la création de nouveaux objets (des produits, des salaires), soit la modification d'objets existants déjà (des comptes).

Cette analyse donne alors une réponse à notre deuxième question : ce sont les opérations qui modifient l'état de l'organisation.

I-3.3.- La catégorie événement

Il nous faut maintenant chercher quelles sont les causes, ou les conditions de l'exécution de ces opérations. Au premier abord ces causes semblent différentes suivant les opérations :

- l'opération calcul du salaire est déclenchée à la fin du mois
- l'opération livraison est déclenchée lorsqu'une commande a été traitée
- l'opération réapprovisionnement est déclenchée lorsque le stock passe en dessous d'un certain seuil
- l'opération mise à jour des comptes est déclenchée tous les soirs.

Appelons événement tout ce qui survient ainsi dans l'organisation et qui déclenche l'exécution d'opérations.

A priori les exemples précédents font apparaître deux types d'événements :

- des événements dont la définition est liée au temps : la fin du mois, tous les jours ;
- des événements dont la définition est liée au changement d'état d'un objet : lorsque la commande a été traitée, lorsque le stock passe en dessous d'un seuil.

Si nous considérons l'écoulement du temps comme l'évolution d'un objet particulier, l'événement apparaît dans tous ces exemples comme la constatation d'un changement d'état particulier d'objet.

La réponse "intuitive" aux trois questions nous conduit à répertorier 3 catégories de phénomènes

OBJET - OPERATION - EVENEMENT

et à décrire la dynamique de l'organisation par 3 types d'interactions entre les catégories :

MODIFIE	(opération, objet)
CONSTATE	(objet, événement)
DECLENCHE	(événement, opération)

C'est notre hypothèse sur la dynamique de l'organisation et la base du modèle descriptif. Développons la par une étude de chacune des catégories et de ses intéractions avec les autres.

II - ANALYSE ET DESCRIPTION DES CATEGORIES

II-1.- Choix d'un mode de description

Nous inspirant de la description de la catégorie OBJET faite par les chercheurs en base de données (BEN 76), (KEN 76), (CHE 76), nous décrivons les catégories par leurs propriétés, leurs associations et leur décomposition en classes.

Définition de l'association

Une association est un ensemble de deux ou plusieurs phénomènes appartenant à la même catégorie ou à des catégories différentes où chacun d'eux joue un rôle spécifique.

Définition de la propriété

Une propriété est une qualité d'un phénomène ou d'une association de phénomènes réels qui permet de l'appréhender et de la distinguer des autres phénomènes ou des autres associations.

Définition de la classe

Une classe est un ensemble de phénomènes ou d'associations de phénomènes définis par les mêmes propriétés et jouant le même rôle dans l'organisation.

Les interrelations entre les catégories seront décrites également par des associations et des classes.

II-2.- Structure de la catégorie objet

Remarque préliminaire

Cette catégorie de phénomènes a été très largement décrite et discutée dans les travaux portant sur les modèles de données. Certains auteurs (FLO 77), (KEN 76) ont même fait une comparaison des différentes définitions proposées dans les différents modèles. Ainsi cette partie de notre exposé n'est pas entièrement originale, elle est, pour une bonne part, une synthèse de leurs conclusions.

II-2.1.- Définition de l'objet

La plupart des définitions données à l'objet, ou à l'entité qui est un terme souvent utilisé à la place d'objet sont statiques, c'est-à-dire qu'elles définissent l'objet indépendamment du temps.

Voici quelques unes de ces définitions :

- BENCI et al (BEN 76)
  Un objet est un constituant concret ou abstrait qu'un individu ou un groupe conçoit et perçoit comme ayant une existence intrinsèque.
- MIJARES (MIJ 76)
  Une entité est une personne, un endroit, un statut, un événement ou une chose spécifique. Chaque entité est caractérisée par les valeurs d'un ensemble spécifique d'attributs.
- LINDGREN (LIN 74)
  Une entité est une chose qui peut être distinguée dans son environnement soit physiquement soit par une description.

Comme l'ont remarqué KENT (KEN 76) et FLORY (FLO 77), ces définitions n'en sont pas puisqu'elles ramènent la définition de l'objet à celle d'un constituant ou d'une chose pour lequel les auteurs ne donnent pas de définition.

Nous appuyant sur la définition systémique de l'organisation, nous définissons tout d'abord l'objet comme un élément de l'organisation. Cependant, proposant trois catégories de phénomènes réels, il nous faut préciser ce qui différencie l'objet de l'opération ou de l'événement et pour cela il nous faut tenir compte de son comportement au cours du temps.

Nous rejoignons alors la définition de l'objet proposé par PEAUCELLE (PEA 77) qui est la seule à n'être pas une définition statique. Pour lui "un objet est un élément fixe durable dans sa spécificité" ; une réalité que l'on peut connaître, qu'il est possible de définir en plusieurs lieux et à plusieurs moments" ; "l'objet bouge dans l'espace, évolue dans le temps".

Ce qui caractérise l'objet, par rapport à l'opération et à l'événement, c'est son autonomie d'existence.

Un objet existe en effet dans l'organisation indépendamment des transformations qu'il peut subir, ou du rôle qu'il peut jouer dans des opérations ou des événements. Nous n'entrerons pas ici dans des considérations philosophiques sur la permanence des êtres et des choses qui consistent à se demander à partir de quel moment un objet ayant été transformé devient un autre objet. Nous sommes tout à fait conscient de l'ambiguïté qui subsistera toujours dans tout modèle descriptif d'une réalité complexe.

Nous retiendrons donc cette définition :

Un objet est un constituant abstrait ou concret de l'organisation durable dans sa spécificité.

II-2.2.- Propriété d'objet

Un objet se caractérise, se distingue des autres par des qualités que nous appelons propriétés. Pour décrire ces propriétés il est nécessaire de leur donner un nom, une référence. Cette

référence de la propriété est nécessaire pour distinguer la qualité de l'objet de la valeur prise par cette qualité à un instant donné. En effet, au cours du temps,un objet est décrit par un même ensemble de propriétés mais les valeurs prises par chacune de ces propriétés peuvent être différentes à des instants différents de la vie de l'objet.

Cette distinction entre le nom de la propriété et la valeur prise par la propriété a été faite par les concepteurs de modèles de données mais seul LANGEFORS (LAN 66),introduit le temps dans la définition des propriétés d'objets par l'intermédiaire du concept de e-message dont nous avons donné la définition au chapitre 1.

Identifiant

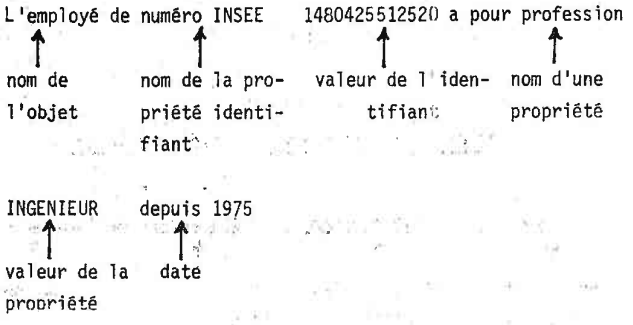
Pour définir précisément les objets, les propriétés sont insuffisantes puisque, comme nous venons de le dire, leurs valeurs peuvent être différentes au cours du temps. Pour reconnaître la permanence de l'objet malgré l'évolution de ses propriétés, il est nécessaire qu'il possède une propriété particulière qui garde la même valeur tout au long de sa vie. Cette nécessité a été démontrée également par les concepteurs de modèles de données (KEN 77) et cette propriété particulière est en général appelée identifiant.

Cet identifiant n'existe pas toujours dans la réalité de l'organisation. Un objet qui ne possède pas d'identifiant, est souvent un objet qui ne joue aucun rôle direct dans la gestion de l'organisation c'est-à-dire dans la réalisation de ses objectifs, car il y a longtemps que les gestionnaires ont reconnu la nécessité d'identifier les constituants de l'organisation afin de pouvoir contrôler leur gestion. Ainsi l'identifiant est souvent une propriété qui ne représente pas une qualité réelle de l'objet, mais une qualité abstraite attribuée à l'objet dans l'unique objectif de l'identifier. Ex: le numéro INSEE, le numéro de compte.

Remarquons qu'il est possible qu'un objet possède plusieurs identifiants. Il est de coutume alors d'en choisir un appelé identifiant principal pour repérer l'objet. Dans la suite de ce travail, lorsque nous parlerons d'identifiant, il s'agira de l'identifiant principal.

Exemples d'objets et de propriétés d'objets

- objets concrets



Le produit de numéro X524 a un prix unitaire de 5000 F depuis le 5/4/81.

- objets abstraits

Le compte numéro 610 a un solde de 3582 F à la date du 15 Mai 1981.

La facture numéro 4857 établie le 15/9/81 a un montant de 4520 F.

Le contrat d'assurance X45L54 signé le 25/4/81 est du type tout risque.

II-2.3.- Association d'objets

Définition

Une association est un ensemble de deux ou plusieurs objets, qui définit une situation réelle dans laquelle chacun d'eux joue un rôle particulier.

Comme l'objet, l'association est définie par un ensemble de propriétés qui lui sont propres.

Elle est identifiée par les objets qui la composent c'est-à-dire par l'ensemble de leurs identifiants.

Parfois l'association possède un identifiant propre, les identifiants des objets apparaissent alors comme des propriétés de l'association.

Exemples d'association

- la commande d'une bicyclette de type X25 par le client DUPONT effectuée le 15/2/82 ;
- l'appartenance de l'employé DURAND à un service comptabilité depuis 1976 ;
- la composition de la bicyclette X25 en composants élémentaires X22, X23, X24.

Nous distinguons dans ces exemples deux types d'associations :

- les associations de composition qui décrivent les objets de façon statique car elles existent tout au long de la vie des objets. Elles supposent la définition d'objets simples et d'objets composés ; la composition des objets composés en objets simples ou en d'autres objets s'exprimant par des associations.

- les associations de situation qui décrivent la situation des objets les uns par rapport aux autres à un instant donné ou pendant une certaine période. Ces associations, à la différence des précédentes, n'ont pas la même durée de vie que les objets qu'elles relient. Par exemple le client et la bicyclette X25 existent avant la commande qui les associe.

II-2.4.- Etat d'un objet

Définition : Situation de l'objet vis à vis de l'organisation à un instant donné.

Ayant défini les objets comme des êtres vivants, il est logique de leur associer deux états particuliers : état de vie et état de mort, ou encore état existant et état inexistant.

Un objet est dans l'état existant vis à vis de l'organisation, s'il existe réellement, c'est-à-dire s'il a subi une opération de création, et si l'organisation en a connaissance. La qualification de cet état ne peut se faire que par rapport au temps, soit, comme dans les deux derniers exemples d'objets : la facture et le contrat d'assurance, en associant à l'objet "leur date de naissance", soit,

comme nous le proposerons ensuite, en décrivant les autres catégories et en datant l'opération de création ou l'événement, qui a provoqué cette création. La date de création réelle de l'objet ne coïncide pas toujours avec sa date de naissance vis à vis de l'organisation. Par exemple, le bon de commande rempli chez lui par le client et non encore posté, existe, mais son existence n'est pas encore connue de l'organisation. Ce n'est que lorsque le bon de commande, acheminé par la poste, sera parvenu sur le bureau de la personne chargée de les dépouiller que l'objet "bon de commande" passera dans l'état existant vie à vis de l'organisation, ainsi la date de création, vis à vis de l'organisation de l'objet bon de commande n° 5420 sera la date à laquelle il a été lu dans l'organisation et non pas la date réelle à laquelle il a été confectionné, cette dernière date devenant une propriété de l'objet.

Inversement un objet est dans l'état inexistant vis à vis de l'organisation à un instant donné lorsque l'objet n'existe plus ou pas encore à l'instant où l'on se place, ou lorsqu'il existe mais que l'organisation n'a pas encore connaissance de son existence.

De même qu'une opération de création est nécessaire pour faire passer un objet de l'état inexistant à l'état existant, une opération de suppression est indispensable pour faire retourner à l'état inexistant un objet existant. La date d'exécution de cette opération est la date de mort de l'objet vis à vis de l'organisation. De même que pour la date de création, cette date ne coïncide pas toujours avec la date de mort réelle de l'objet. Par exemple un employé partant en retraite passe dans l'état inexistant vis à vis de l'organisation et, heureusement pour lui, cet instant n'est pas celui de sa mort.

Nous insistons sur ce point afin de bien mettre en évidence que nous définissons l'état des objets en tant qu'éléments du système organisation, et non pas en tant qu'objets indépendants de tout système.

Remarque :

Ces définitions de l'état existant et de l'état inexistant peuvent être étendues aux associations avec la restriction suivante :

- Une association entre objets ne peut être dans l'état existant que si les objets qu'elle relie le sont.

II-2.5.- Classes d'objets ou d'associations d'objets

Définition

Une classe d'objets est un ensemble d'objets définis par des propriétés identiques par leur sens.

Une classe d'associations est un ensemble d'associations définies par des propriétés identiques par leur sens, reliant des objets appartenant chacun à une même classe et y jouant un rôle identique.

Cette notion de classe est naturelle à l'homme, elle correspond à sa faculté de conceptualisation c'est-à-dire d'abstraction et de généralisation qui le conduit à choisir des noms communs pour désigner les objets c'est-à-dire en fait à les répartir en classes.

Exemples de classes :

la classe produit
la classe employé
la classe client
la classe service
la classe commande

Remarquons que ce que nous appelions nom de l'objet ou nom de l'association dans les exemples précédents est, en fait, le nom de la classe à laquelle appartient cet objet ou cette association.

La répartition des objets en classes n'est pas unique, elle est liée aux objets et aux associations choisies pour définir l'organisation.

Résumé sur la catégorie objet

- Un objet est défini de façon statique par :
  - le nom et le sens de la classe à laquelle il appartient
  - le nom, le sens et la valeur de sa propriété identifiant
  - le nom, le sens et les valeurs de ses propriétés
  - le nom et le sens des classes d'associations où il intervient et le rôle qu'il y joue
  - le nom et le sens des propriétés de ces associations.
- L'état d'un objet à un instant donné est défini par :
  - son existence ou sa non-existence à cet instant.

Dans le cas où il existe son état est défini par :

  - les valeurs de ses propriétés à cet instant
  - l'existence ou la non-existence de ses associations avec d'autres objets
  - les valeurs des propriétés de ses associations existantes à cet instant.
- L'évolution de l'objet est définie par un ensemble de propriétés temporelles :
  - sa date de création
  - sa date de suppression
  - les dates de changement de valeur de ses propriétés
  - les dates de création, suppression ou modification des valeurs des propriétés de ses associations.
- La dynamique de l'objet, c'est-à-dire la description des causes de ses changements d'état, ne peut être décrite par la structure de la catégorie objet définie isolément, elle nécessite la définition de la structure complète de l'organisation, c'est-à-dire la définition des associations entre les trois catégories de phénomènes réels.

II-3.- Structure de la catégorie opération

II-3.1.- Définition de l'opération

Une opération est une action qui peut être exécutée isolément dans l'organisation.

Comme nous l'avons dit dans l'introduction des catégories, les

opérations définissent les phénomènes réels correspondant au travail fourni dans l'organisation.

Une opération est exécutée par un acteur, un homme ou une machine, dans un lieu donné, à un instant donné, elle dure un certain laps de temps.

Exemples d'opérations :

- la fabrication de l'atelier n° 5 le 15 janvier 1977
- la livraison effectuée le 25 mars 76 par le magasinier
- le calcul du salaire de l'agent Dupont effectué le 25 mars 1977 par le service de la paie
- la mise à jour des comptes effectuée par l'ordinateur le 15 avril 1978 à 15 h.

Son rôle consiste à participer de façon directe (comme les opérations de production) ou indirecte (comme les opérations de gestion), à la réalisation des objectifs de l'organisation. Cette participation se traduit, pour les premières, par la création d'objets concrets (des produits) et,pour les secondes, par la modification de l'état d'objets abstraits (des comptes, des salaires).

Il en résulte que la définition d'une opération ne peut être complète que si l'on précise son rôle dans l'organisation par ses effets sur les objets. Nous proposons de décrire ce rôle par des associations particulières entre opérations et objets.

II-3.2.- Propriétés des opérations

Une opération est définie par plusieurs types de propriétés :

- des propriétés spatio-temporelles qui permettent de la situer dans le temps et dans l'espace,comme par exemple :
  - la date d'exécution
  - le lieu d'exécution
  - la durée d'exécution
- des propriétés attachées à son rôle qui précisent la fonction, ou le sous-système de l'organisation auquel elle appartient.

Par exemple :

la fonction peut prendre les valeurs : production, gestion, prévision, etc ..

- des propriétés particulières définissant l'action en elle-même et les conditions nécessaires à son exécution comme :
  - la qualification exigée
  - les règles de sécurité à respecter
  - les grandeurs définissant les moyens qu'elle nécessite comme :
    - la quantité d'énergie
    - le nombre de personnes
    - le nombre de machines.

Pour les opérations portant sur des objets abstraits, comme les salaires, les comptes, l'action exécutée est décrite par un énoncé, ou texte de l'opération qui définit les modalités de valorisation des propriétés des objets modifiés par l'opération. Lorsque l'opération est exécutée par un ordinateur,la valeur de cette propriété est alors le texte du programme dont l'exécution correspond à celle de l'opération.

Cette propriété "énoncé" diffère des autres propriétés des opérations ou des objets par le domaine dans lequel se situent ses valeurs. En effet, pour les propriétés rencontrées, jusqu'ici les domaines de valeurs sont du type : liste finie de valeurs, ensemble des entiers, intervalle de réels.

Le texte a pour valeur une séquence de phrases exprimées dans un certain langage. Ce qui signifie que son domaine est l'ensemble des séquences de phrases engendrables par un langage.

Ayant fait l'inventaire des propriétés qui peuvent définir une opération, nous devons nous demander, comme cela a été fait pour les objets, si l'ensemble des valeurs prises par les propriétés d'une opération est suffisant pour la distinguer de façon certaine de toutes les opérations.

La réponse à cette question est négative, car il est tout à fait possible que deux opérations soient exécutées en même temps, au même endroit, et caractérisées par le même ensemble de propriétés. Par exemple, les opérations effectuées par l'ensemble du personnel d'un service de comptabilité ne peuvent pas toujours être différenciées par les valeurs de leurs propriétés.

Nous proposons donc dans ce cas de munir l'opération d'un identifiant, c'est-à-dire d'une propriété supplémentaire, destinée à la distinguer sans ambigüité possible des autres opérations.

II-3.3.- Associations entre opérations

Il existe,entre les opérations,des associations de composition qui expriment que l'exécution d'une opération est,en fait, une succession d'opérations élémentaires ou composées.

Par exemple, l'opération de calcul de salaire SAL2402 se compose de trois opérations : GAIN2402, RETENUE2402 et NET2402 correspondant aux calculs des gains,des retenues et du net à payer.

II-3.4.- Associations entre opérations et objets : Associations MODIFIE

Nous avons défini les opérations comme des phénomènes réels ayant pour effet la modification de l'état de l'organisation.

D'autre part, nous avons défini l'état de l'organisation par l'état de ses constituants, c'est-à-dire par l'état des objets et de leurs associations.

Donc décrire l'effet des opérations sur l'organisation revient à décrire l'effet des opérations sur l'état des objets ou des associations d'objets.

Nous proposons d'effectuer cette description par une association. Nous disons qu'il existe une association "modifie", entre une opération et des objets ou des associations d'objets, si l'exécution de l'opération a pour conséquence un changement d'état de ces objets ou de ces associations d'objets.

Par exemple, il existe une association"modifie"entre l'opération SAL2402 et l'objet Salaire du mois de janvier 1982 de l'employé de n° insee 1450708043013 car l'exécution de cette opération crée cet objet.

D'après la définition des différents états d'un objet, les changements provoqués par une opération sur un objet sont de trois types :

- création : l'opération provoque le passage de l'objet de l'état inexistant à l'état existant et a pour rôle d'attribuer des valeurs initiales à un certain nombre de propriétés de l'objet.
- suppression : l'opération provoque le passage de l'objet de l'état existant à l'état inexistant c'est-à-dire qu'elle le détruit réellement, par exemple : l'employé qui déchire une commande annulée, ou fictivement : l'employé qui réexpédie une commande erronée.
- mise à jour : l'opération provoque le passage de l'objet d'un état existant à un autre, en attribuant de nouvelles valeurs à certaines propriétés de l'objet.

Les types de changement d'état provoqués sur chaque objet de l'association "modifie" sont des propriétés de l'association.

II-3.5.- Classes d'opérations

Une classe d'opération est un ensemble d'opérations modifiant des objets ou des associations d'objetsappartenant respectivement à des mêmes classes et définies par des propriétés identiques par leur sens.

Exemple : la classe : calcul de salaire
la classe : fabrication de produit
la classe : gestion de commande.

Une classe est définie par un nom ; la répartition des opérations en classes n'est pas unique, elle dépend de la classification des objets et des propriétés utilisées pour décrire les opérations.

Notons qu'à la différence des objets, il est fréquent que les propriétés des opérations d'une même classe aient la même valeur pour toutes les opérations de la classe. Par exemple, la qualification, les règles de sécurité, le nombre de personnes ou même le texte peuvent avoir des valeurs identiques pour toutes les opérations de la classe. Remarquons cependant que la représentation de toutes les exécutions des opérations d'une classe peut être utile pour les gestionnaires de l'organisation. Par exemple,pour effectuer une opération de rappel de salaire pour un agent X sur une période déterminée,il est nécessaire de connaître exactement, d'une part la situation de l'agent X lors de chaque mois de la période de rappel, d'autre part les règles de calcul en vigueur à chacun de ces mois. Ces règles correspondent aux différentes valeurs de la propriété texte des opérations de la classe calcul de salaire, effectuées pendant cette période.

De même, effectuer un suivi de commande est très simple si les opérations sont décrites au moyen du modèle : il suffit de rechercher les opérations associées à cette commande et les propriétés date d'exécution de ces opérations, ce qui permet par exemple de répondre à un client émettant une réclamation :

- votre commande a été contrôlée le 15 janvier
- la fabrication des produits demandés a été lancée le 25 janvier et terminée le 30 janvier
- la facturation a été faite le 28 janvier
- l'expédition sera effectuée dans un délai minimum de 8 jours.

II-3.6.- Classes d'associations entre classes d'objets et classes d'opérations

Toutes les associations "MODIFIE" qui relient des objets (ou associations d'objets) appartenant à une ou plusieurs classes et des opérations (ou associations d'opérations) d'une même classe, définissent une classe d'association "MODIFIE" :

Ex : toute opération de la classe "calcul de salaire" provoque la création d'un objet "salaire d'agent" : il y a une classe d'association entre la classe d'opération "calcul de salaire" et la classe d'objet "salaire d'agent".

II-3.7.- Résumé sur la catégorie opération

- Une opération est définie de façon statique par :
  - le nom et le rôle de la classe à laquelle elle appartient
  - le nom, le sens et les valeurs de ses propriétés
  - les associations définissant sa composition en opérations
  - le nom et la valeur de son identifiant
  - les associations qui la lient aux objets et les propriétés de ces associations.
- Une opération ne peut avoir que trois états :
  - inexistant
  - en cours d'exécution
  - exécutée.
- L'évolution de l'opération est définie par une propriété temporelle
  - une date d'exécution qui correspond à son passage de l'état en cours d'exécution à l'état exécutée

Comme pour la catégorie objet,nous remarquons que la dynamique de l'opération n'est pas décrite par la seule structure de la catégorie OPERATION. En particulier, la propriété temporelle exprimant le passage de l'état inexistant à l'état en cours d'exécution est absente car elle appartient à l'association DECLENCHE exprimant l'origine de l'exécution de l'opération que nous décrivons dans la catégorie EVENEMENT.

II-4.- Structure de la catégorie événement

II-4.1.- Définition de l'événement

Un événement est ce qui survient dans l'organisation et y provoque un changement d'état.

L'événement arrive dans un lieu donné à un instant donné. Sa durée de vie est limitée à la période comprise entre l'instant où il survient et l'instant où ses conséquences sont effectives.

Exemples d'événements

- l'arrivée de la commande 2850 le 25 avril 1976
- la fin du mois de mai 1976
- la panne de la machine 2750 survenue le 15 janvier 1975 à 15 h
- la rupture de stock du produit X survenue le 18 février 1977
- le début de l'année 1977
- l'embauche de l'ouvrier Dupont le 1er juillet 1978
- l'accident survenu à l'employé Durand le 25 mars 1976 à 17 h.

On peut voir à priori plusieurs sortes d'événements :

- des événements temporels dont les définitions sont liées au temps et dont les arrivées sont causées par l'écoulement naturel du temps (fin de mois, début d'année).

- des événements aléatoires et imprévisibles dont les causes ne sont pas connues : la panne d'une machine, l'accident d'un employé, l'incendie d'un atelier. Ces événements, en fait, ne font pas partie du sous-système causal de l'organisation tel que nous l'avons défini, puisque leurs causes et surtout leurs effets ne sont pas déterminés.

- des événements liés aux changements d'état d'objets
  - l'arrivée de la commande 2850 : correspond au passage de cette commande de l'état inexistant à l'état existant
  - la rupture du stock du produit X : correspond au passage de la propriété stock du produit X d'une valeur quelconque à une valeur vérifiant un prédicat (stock< seuil)

  - l'embauche de l'ouvrier Dupont : correspond au passage de l'état inexistant à l'état existant, vis à vis de l'organisation, de l'objet Dupont.

Les périodes de temps, mois, année, semaine, jour... peuvent être assimilées à des classes d'objets particulières, chaque occurrence de période, le mois de mars, l'année 1912, étant des objets de la classe. Cette description du temps permet de rapprocher tout événement du sous-système causal de l'organisation d'un changement d'état d'un ou plusieurs objets.

II-4.2.- Propriétés des événements

Les événements possèdent des propriétés permettant de les situer dans le temps ou dans l'espace :

- la date ou l'instant où ils se produisent, que nous appelons DATE D'ARRIVEE
- le lieu où ils se passent.

Ils sont pauvres en propriétés descriptives ; ceux qui en possèdent le plus sont ceux que nous avons éliminés : les événements imprévisibles et aléatoires (par exemple un accident peut être décrit pas son taux de gravité, la nature des dommages causés, le temps qu'il faisait lorsqu'il s'est produit etc).

Pour les autres, auxquels, nous nous intéressons, nous pouvons citer :

Par exemple :

- le nom du jour de la semaine coïncidant avec la fin du mois de mars 1982.
- l'écart de la rupture de stock du produit 4527 survenue le 18 janvier 1982. Cet écart ayant pour valeur la différence entre la valeur du stock avant l'opération ayant provoqué la rupture et la valeur après l'opération.

Comme pour les objets et les opérations, ces propriétés peuvent être insuffisantes pour distinguer les événements les uns des autres.

Il faut identifier chaque événement par la valeur d'un identifiant.

II-4.3.- Associations d'événements

Les événements comme les autres catégories de phénomènes possèdent une structure propre que l'on peut décrire par des associations de composition.

- Une association de composition définit un événement comme étant une réunion d'événements simples ou composés se produisant en même temps. Ces associations résultent,en fait,de l'échelle de temps plus ou moins fine que l'on utilise pour les décrire. Peaucelle (PEA 77) donne un exemple de ce type d'association à propos d'un événement n'appartenant cependant pas à une organisation. Il montre qu'un événement comme une bataille peut être considéré soit comme un événement simple si l'échelle de temps que l'on prend est l'année, soit comme un événement composé d'un ensemble d'événements particuliers survenus au cours du déroulement de la bataille si l'échelle du temps est l'heure.

Dans le sous-système causal de l'organisation,la composition des événements est peu complexe dans la mesure où, lorsqu'on décrit le fonctionnement de l'organisation, on choisit une échelle de temps assez fine (le jour par exemple). Cependant, on peut trouver des exemples : l'association entre l'événement arrivée d'une commande et les n événements "arrivée d'une ligne de commande" et éventuellement l'événement "arrivée d'un client" exprime la concomitance de tous ces événements.

II-4.4.- Association entre événements et objets : association CONSTATE

Dans une organisation un événement est lié à un changement d'état d'objet ou d'association d'objets.

Nous disons que l'événement est la constatation d'un changement d'état de l'organisation par un détecteur et nous décrivons cette réalité par une association "CONSTATE" entre l'événement et les objets ou les associations d'objets dont il constate le changement d'état.

Par exemple il existe une association CONSTATE entre l'événement RUPT4525 et le produit X255, ou entre l'événement ARCOM5455 et la commande 4287.

Nous avons vu qu'un objet peut subir 3 types de changement d'état : création, suppression, mise à jour ; nous décrivons comme des propriétés de l'association CONSTATE les types de changement d'état des objets appartenant à l'association. Dans le cas où le changement d'état de l'objet est du type mise à jour, il est défini par deux états : son état dit état initial, avant le début de l'exécution de l'opération qui provoque sa mise à jour, et son état à l'instant de la fin de l'exécution de l'opération, dit état final. Ces états sont décrits par un prédicat portant sur les propriétés de l'objet associé à l'événement.

Ce prédicat est une propriété de l'association CONSTATE.

Il arrive souvent que seul un des deux états intervienne dans la définition de l'événement, l'autre pouvant être quelconque. Par exemple, la rupture de stock RUPT4525 constate le passage de l'objet produit X455 d'un état quelconque à un état final tel que sa quantité en stock soit inférieure à un certain seuil. En revanche, l'événement reconstitution du stock APP5235 constate le passage d'un état initial "quantité stock < seuil" à un état quelconque.

Un événement ne se produisant qu'à un instant donné n'apparait que dans une seule association CONSTATE. Alors qu'un objet ayant une durée de vie assez longue et pouvant subir de multiples changements d'état apparait dans plusieurs.

Remarquons à ce propos la distinction qui existe dans le modèle entre

- d'une part la notion d'état et celle de changement d'état
- d'autre part la notion de changement d'état et celle d'événement.

- Un état est durable, un changement d'état est instantané.

Par exemple,un produit peut rester dans l'état de rupture un certain temps dans l'attente de son réapprovisionnement, tandis

que le changement d'état correspondant au passage à l'état de rupture s'est produit à un instant précis.

- Tous les changements d'état ne sont pas des événements

Poursuivons l'exemple du produit ; le stock du produit subit au cours du temps de multiples changements d'état correspondant aux différentes livraisons effectuées ; parmi ces changements d'état, seuls quelques uns correspondent à des événements. Ce sont ceux qui satisfont aux prédicats des associations CONSTATE reliant des classes d'événements à la classe produit, c'est-à-dire les changements d'état correspondant à une rupture de stock ou à une reconstitution de stock.

Remarquons également que la définition de l'événement comme constatation d'un changement d'état de l'organisation permet d'exprimer le décalage temporel qui existe souvent entre l'instant où l'objet change d'état et l'instant où les conséquences de ce changement d'état, c'est-à-dire l'exécution d'opérations, se produisent. Par exemple ce n'est pas à l'instant où le produit est épuisé que l'opération de réapprovisionnement est lancée mais lorsqu'un employé a constaté la rupture de stock.

II-4.5.- Associations entre événements et opérations : Association DECLENCHE

Le rôle de l'événement est de déclencher des opérations, c'est-à-dire de les faire passer de l'état inexistant à l'état en cours d'exécution.

Comme précédemment nous décrivons cette réalité par une association, appelée "DECLENCHE", entre un événement et une ou plusieurs opérations, celles dont l'exécution est causée par l'événement.

Pour une opération donnée, il n'existe qu'une seule association la reliant à un événement. En effet, nous avons vu que l'opération est exécutée une seule fois, qu'elle existe dans l'organisation entre deux instants : l'instant de son début d'exécution et l'instant de sa fin d'exécution.

Une propriété temporelle de cette association est la date à laquelle le déclenchement a été effectué , nous l'appelons DATE DE DECLENCHEMENT.

Un événement peut déclencher plusieurs opérations.

Par exemple l'événement M0177, fin du mois de janvier 1977 déclenche l'opération SAL7701 de calcul de salaire et l'opération STAT37701 d'établissement des statistiques mensuelles.

Le déclenchement d'une opération peut être conditionnel.

Ex : un événement du type "rupture de stock" déclenche une opération de réapprovisionnement si la situation financière le permet.

La condition de déclenchement est l'expression d'un certain état que le système doit avoir pour que l'opération soit effectivement exécutée lorsque l'événement survient. Cette condition est une propriété de l'association DECLENCHE.

Le déclenchement peut être multiple

Ex : l'événement "fin de mois" déclenche une opération de calcul de salaire pour tous les employés.

Cela exprime le fait qu'il doit y avoir exécution répétée de n opérations de la même classe modifiant n objets de la même (ou de plusieurs) classe (s).

La multiplicité du déclenchement est une propriété de l'association déclenche que nous appelons facteur de déclenchement.

II-4.6.- Classes d'événements

Une classe d'événements est un ensemble d'événements constatant des changements d'état de même nature, d'objets ou d'associations d'objets appartenant respectivement à des mêmes classes, déclenchant des opérations appartenant à des mêmes classes d'opérations et définis par des propriétés identiques par leur sens.

Exemple : la classe rupture de stock est l'ensemble des événements constatant le changement d'état d'un objet de la classe produit,

définis par les propriétés : type de changement ="modification"et état final ="stock < seuil"et déclenchant des opérations de la classe réapprovisionnement.

La classe arrivée de la commande est l'ensemble des événements constatant la présence d'une nouvelle commande dans l'organisation et déclenchant des opérations de la classe "analyse des commandes".

II-4.7.- Classes d'associations entre classes d'objets et classes d'événements

Toutes les associations "CONSTATE" qui s'appliquent à des objets (ou des associations d'objets)appartenant à une ou plusieurs classes déterminées et à des événements d'une même classe définissent une classe d'associations"CONSTATE".

Ex : tous les événements "rupture de stock" constatent le changement d'état (passage en dessous d'un seuil) d'ob-jets de la classe "Produit " : il y a une classe d'associations CONSTATE entre la classe d'objets "Produit " et la classe d'événements "rupture de stock".

II-4.8.- Classes d'associations entre classes d'événements et classes d'opérations

Toutes les associations "DECLENCHE" qui s'appliquent à des événements d'une même classe et à des opérations appartenant à une ou plusieurs classes déterminées définissent une classe d'associations "DECLENCHE".

Ex : dans le cas précédent,tous les événements de la classe "rupture de stock" déclenchent des opérations appartenant toutes à la classe "réapprovisionnement de stock" : il y a une classe d'association "DECLENCHE" entre la classe d'événements "rupture de stock" et la classe d'opérations "réapprovisionnement de stock ".

II-4.9.- Résumé sur la catégorie EVENEMENT

Un événement est défini par

- le nom et le rôle de la classe à laquelle il appartient
- le nom, le sens et les valeurs de ses propriétés
- les associations définissant sa composition en événements.

Un événement a trois états possibles

- futur : l'événement n'a pas encore eu lieu
- présent : l'événement a lieu
- passé : l'événement a eu lieu

L'évolution de l'événement est définie par :

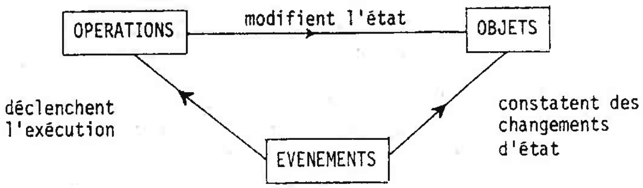
- sa propriété DATE D'ARRIVEE
- une association CONSTATE avec des objets spécifiant son origine c'est-à-direla raison de son passage de l'état futur à l'état présent
- les propriétés de cette association, en particulier la propriété PREDICAT
- une association DECLENCHE avec des opérations spécifiant ses conséquences sur l'organisation. Le déclenchement des opérations fait passer l'événement à l'état passé
- les propriétés de cette association notamment les propriétés : CONDITION DE DECLENCHEMENT
  FACTEUR DE DECLENCHEMENT
  DATE DE DECLENCHEMENT

RESUME

Le modèle descriptif d'une organisation que nous proposons permet de décrire de manière unifiée, grâce aux concepts simples de propriété, classe et association :

- les constituants de l'organisation par l'intermédiaire de la catégorie OBJET ,
- les règles de gestion qui y ont cours par l'intermédiaire de la catégorie OPERATION ,
- les conditions d'application de ces règles par l'intermédiaire de la catégorie EVENEMENT.

La dynamique du sous-système causal de l'organisation est décrite par des associations entre phénomènes de catégories différentes comme l'illustre ce schéma.



Les événements correspondent à des changements particuliers d'objets, ils déclenchent des opérations qui modifient l'état d'autres objets et provoquent parfois ainsi de nouveaux événements.

Ce modèle correspond à l'approche des modèles cybernétiques dans lesquels la dynamique d'un système est décrite en termes d'actions et de réactions (ASH 61)

- les actions ou stimuli externes sont ici les événements
- les réactions sont les opérations déclenchées par les événements
- les variables d'état du système sont les objets et leurs propriétés.

Pour compléter cet exposé du modèle descriptif d'une organisation nous en donnons dans le paragraphe suivant une formalisation.

Cette formalisation a un double rôle :

- elle permet de montrer la complétude du modèle vis à vis de la description de la dynamique de l'organisation.
- elle sert de référence, dans le chapitre suivant, pour la définition des concepts du modèle conceptuel associé au modèle descriptif.

III - FORMALISATION DU MODELE DESCRIPTIF

La formalisation du modèle descriptif est effectuée en deux étapes :

- définition formelle des notions d'état, de changement d'état et de dynamique d'une organisation,
- expression formelle de la dynamique du sous-système causal d'une organisation modélisée à l'aide des concepts du modèle descriptif.

III-1.- Définition générale de la dynamique d'une organisation

III-1.1.- Etat de l'organisation

- Soit O l'ensemble des phénomènes de la catégorie OBJET ayant existé ou pouvant exister dans l'organisation au cours du temps.

- Soit V un ensemble de valeurs contenant en particulier la valeur indéfinie notée u.

- Soit S l'ensemble des applications de profil : $O \rightarrowtail V$.

Etat de l'organisation

Une application $s \in S$ définit un état de l'organisation.

$\forall o \in O$ $\quad s(o) = u \Leftrightarrow o$ n'existe pas dans cet état

$\qquad\qquad s(o) \neq u \Leftrightarrow o$ existe et $s(o)$ est son état.

III-1.2.- Changement d'état de l'organisation

- Soit F l'ensemble des applications de profil : $P_0 \rightarrowtail V$ où $P_0 \in \mathcal{P}(O)$ contenant en particulier l'application de graphe vide : $\emptyset \rightarrowtail V$ notée n.

- Soit l'application CHGETAT de profil : $S \times F \rightarrowtail S$

$CHGETAT(s,f) = s'$ ou $f : O_1 \rightarrowtail V$ signifie que :

- les états s et s' différent par les états des objets appartenant à O_1
- $\forall o_1 \in O_1 \quad s'(o_1) = f(o)$ $\quad \forall o \in \complement_O^{O_1}$, $s'(o) = s(o)$.

Le changement d'état de tout $o_1 \in O_1$ peut être de trois types :

- $s(o_1) = u \Leftrightarrow$ une création
- $s'(o_1) = u \Leftrightarrow$ une suppression
- $s(o_1)$ et $s'(o_1) \neq u \Leftrightarrow$ une modification.

Un changement d'état est donc défini par un couple (s,s') où $s \in S$, $s' \in S$ et $s' = CHGETAT(s,f)$ $f \in F$ et $f \neq n$.

III-1.3.- Origine des changements d'état de l'organisation, introduction de la fonction DECL

Les changements d'état survenant dans l'organisation ont deux origines possibles : une origine externe et une origine interne.

- Origine externe

L'environnement de l'organisation peut à tout instant fournir des couples (o,v) où $o \in O$ et $v \in V$ dans lesquels v est la nouvelle valeur à donner à l'objet o.

Ceci signifie que l'environnement produit des changements d'état (s,s') avec $s' = CHGETAT(s,f)$ dans lesquels f est une donnée vis à vis de l'organisation.

Ces changements d'état correspondent

soit - à l'entrée dans l'organisation de nouveaux objets en provenance de son environnement.

Exemple : Arrivée de nouvelles commandes au courrier,

soit - à la communication à l'organisation de nouvelles valeurs prises par des objets avec lesquels elle est en relation.

Exemple : Communication de la nouvelle valeur du numéro de téléphone du client n° 5454.

- Origine interne

L'organisation réagit aux changements d'état qu'elle subit en provoquant de nouveaux changements d'état.

Cette réaction correspond à l'application de ses règles de gestion, la définir revient à définir ce que nous avons appelé la dynamique de l'organisation. Nous la définissons ici de manière très générale par une application.

Application DECL

Soit DECL une application de profil : $S \times S \longrightarrow F$

Soient $s \in S$, $s' \in S$, $f \in F$ $\quad f : O_1 \longrightarrow V$

$DECL(s,s') = f$ signifie :

- le changement d'état (s,s') a engendré un nouveau changement d'état (s',s'') où $s'' = CHGETAT(s',f)$
- si $f = n$ (fonction de graphe vide) le changement d'état (s,s') n'a pas eu de conséquence.

III.1.4.- Evolution de l'organisation au cours du temps

Les applications CHGETAT et DECL permettent de définir l'évolution de l'organisation au cours du temps.

Cette évolution est une suite $s_0, s_1 \ldots s_{i-1}, s_i \ldots s_n$ d'états définie par la formule de récurrence suivante :

- s_0 est une donnée, c'est l'état initial de l'organisation.
- $s_1 = CHGETAT(s_0,f_0)$ f_0 : une donnée : ensemble de couples (o,v) fournis par l'environnement.
- $s_{i+1} = CHGETAT\ (CHGETAT(s_i,f_i),\ DECL(s_{i-1},s_i))$

  f_i : ensemble de couples (o,v) fournis par l'environnement.
  $DECL(s_{i-1},s_i)$ ensemble de couples (o,v) fournis par l'organisation en réaction au changement d'état (s_{i-1},s_i).

Nous appelons une telle suite un déroulement.

Pour le situer dans le temps, nous définissons une application croissante $DATE : [o,n] \longrightarrow T$ qui à chaque état s_i $i \in [o,n]$

associe la date t_i à laquelle il a été observé.

t_i est appelé la date d'observation de s_i.

T est un ensemble totalement ordonné permettant de définir des dates vis à vis d'un certain calendrier.

Cette formalisation de la dynamique de l'organisation est en fait transposable à tout système pour lequel O est l'ensemble de ses composants.

Elle exprime que définir la dynamique d'un système revient à définir son application DECL, c'est-à-dire une application qui, à tout changement d'état survenant dans le système à un instant donné, associe un ensemble de modifications à faire subir au système et par là même un nouveau changement d'état.

Dans le paragraphe suivant nous donnons une formulation de l'application DECL du sous système causal d'une organisation dont la dynamique est décrite au moyen des concepts du modèle descriptif que nous venons de définir dans la première partie de ce chapitre.

III-2.- Expression formelle de la dynamique du sous-système causal d'une organisation

III-2.1.- Classes d'objets

Le modèle descriptif fait l'hypothèse que les phénomènes de la catégorie OBJET sont structurés en classes : classes d'objets et classes d'associations d'après les propriétés des objets.

Formalisons ces deux notions :

Propriété d'objet

Une propriété d'objet est représentée à un instant donné par une valeur appartenant à un certain ensemble appelé domaine de la propriété.

L'état d'un objet à un instant donné est représenté par l'ensemble des valeurs de ses propriétés à cet instant.

L'ensemble V défini précédemment est donc le produit $V_1 \times V_2 \ldots V_n$ où V_i correspond à un domaine d'une propriété d'objet

- les V_i ne sont pas tous nécessairement distincts.
- chaque V_i contient la valeur indéfinie notée u_i.

A tout état $s \in S$ on associe n applications notées $s^1, s^2 \ldots s^n$ telles que $\forall o \in O \quad s^k(o)$ = projection de $s(o)$ sur V^k.

- $s^k(o)$ est la valeur de la propriété k de domaine V_k de l'objet O dans l'état s.

Classes d'objets

Une classe d'objets est un sous-ensemble C de O.

A une classe d'objets est associé un ensemble de propriétés appelées les propriétés de la classe :

Ce sont les propriétés k telles que $\exists s \in S$ et $o \in C$ tel que $s^k(o) \neq u_k$.

Une propriété d'une classe est donc une application :

$P_k : S \times C \longrightarrow\!\!\!\!\!\rightarrow V_k$ avec $P_k(s,o) = s^k(o)$ pour $s \in S$ et $o \in O$.

Propriété identifiant d'une classe

Une propriété P_k est un identifiant d'une classe C si il existe une propriété $P_I : C \longrightarrow\!\!\!\!\!\rightarrow V_k$ telle que

- $\forall s \in S, \forall s' \in S, \forall o \in C \quad P_k(s,o) = P_k(s',o)$ notée $P_I(o)$
- $\forall o \in C$ et $\forall o' \in C \quad o \neq o' \Rightarrow P_I(o) \neq P_I(o')$

Rappelons que le modèle descriptif impose que toute classe possède au moins une propriété identifiant et parmi elles un identifiant principal.

Un objet est donc

- repéré au cours du temps par la valeur de son identifiant principal,
- représenté à un instant donné par le n-uplet des valeurs de ses autres propriétés.

III-2.2.- Classes d'événements

Evénement

Un événement est un changement d'état (s,s') tel que DECL(s,s') ≠ n.

Classes d'événements

Une classe d'événements CE est un ensemble d'événements qui correspondent à des changements d'état de même type, d'objets appartenant à des mêmes classes.

Prédicat d'une classe d'événements

Le type de changement d'état associé à une classe d'événements CE est défini par un prédicat noté $pred_{CE}$.

Ce prédicat est une application de profil : $S \times S \longrightarrow\!\!\!\!\!\rightarrow BOOL$ où BOOL = {vrai,faux}.

Il a la signification suivante :

Soit un événement (s,s') et une classe d'événements CE de prédicat $pred_{CE}$

si $pred_{CE}(s,s')$ alors $(s,s') \in CE$ sinon $(s,s') \notin CE$.

Exemples

- classe d'événements "arrivée de commande" ARRCOM

  Soit COM la classe des commandes

  $Pred_{ARRCOM}(s,s')$: $\exists com \in COM$, s(com) = u et s'(com) ≠ u

- classe d'événements "rupture de stock" RUPTSTOCK

  Soit PROD la classe des produits

  $pred_{RUPTSTOCK}(s,s')$: $\exists pro \in PROD$, s(pro) > 0 et s'(pro) = 0

Application PRED

Soit B l'ensemble des applications de profil : $S \times S \longrightarrow\!\!\!\!\!\rightarrow BOOL$

Soit CEV l'ensemble des classes d'événements d'une organisation

PRED : $CEV \longrightarrow\!\!\!\!\!\rightarrow B$ PRED(cev) est noté $pred_{cev}$

PRED associe à toute classe d'événements son prédicat.

Application CONST

Soit COB l'ensemble des classes d'objets ou d'associations d'objets d'une organisation.

$$CONST : CEV \longrightarrow\!\!\!\!\!\rightarrow \mathcal{P}(COB)$$

CONST associe à toute classe d'événements les classes d'objets qui interviennent dans son prédicat. Elle représente l'ensemble des associations CONSTATE entre classe d'événements et classes d'objets.

Exemple : CONST(ARRCOM) = COM.

Ceci signifie qu'un événement de la classe arrivée de commande correspond au changement d'état d'un objet de la classe COM.

III-2.3.- Classes d'opérations

Opération

Une opération est le moyen utilisé dans l'organisation, à un instant donné, pour engendrer tout ou partie d'un changement d'état d'origine interne.

Dans le cadre d'un système déterministe, dans lequel nous nous plaçons, ce moyen peut s'exprimer comme l'utilisation à un instant donné d'une certaine application qui fournit les nouvelles valeurs des propriétés des objets concernés par le changement d'état considéré.

Notation préliminaire

U_a : Union d'applications

Définition

Soient f_1 et $f_2 \in F$ $\quad f_1 : O_1 \longrightarrow\!\!\!\!\!\rightarrow V_1$, $\quad f_2 : O_2 \longrightarrow\!\!\!\!\!\rightarrow V_2$

tels que $O_1 \cap O_2 = \emptyset$

$f_1 U_a f_2 = f_3 \Rightarrow f_3 : O_1 \cup O_2 \longrightarrow\!\!\!\!\!\rightarrow V_1 \cup V_2$

et $\forall o_1 \in O_1 \quad f_3(o_1) = f_1(o_1)$

$\forall o_2 \in O_2 \quad f_3(o_2) = f_2(o_2)$

Soit un changement d'état (s,s') tel que

$DECL(s,s') = f \qquad f : O_f \longrightarrow\!\!\!\!\!\rightarrow V_f$

si on a $f = U_a(f_1,f_2 \ldots f_n)$

avec $f_1 = top_1(s',o_1^1,\ldots o_p^1) \qquad f_1 : O_1 \longrightarrow\!\!\!\!\!\rightarrow V_1$

$f_2 = top_2(s',o_1^2,\ldots o_q^2) \qquad f_2 : O_2 \longrightarrow\!\!\!\!\!\rightarrow V_2$

$f_n = top_n(s',o_1^n,\ldots o_r^n) \qquad f_n : O_n \longrightarrow\!\!\!\!\!\rightarrow V_n$

On dira que f a été engendré par n opérations $op_1, op_2 \ldots op_n$ correspondant chacune à l'utilisation d'une certaine application top_i.

top_i est le texte de l'opération

O_i est l'ensemble des objets modifiés par op_i.

les O_i^j sont les objets utilisés par les opérations.

Classes d'opérations

Une classe d'opérations est l'ensemble des opérations effectuées dans l'organisation correspondant à l'utilisation d'un même texte.

Profil du texte d'une classe d'opérations

Soit une classe d'opérations cop

Son texte est une application de profil :

$$text_{cop} : S \times P_1 \ldots P_n \longrightarrow\!\!\!\!\!\rightarrow F$$

où $P_i \in \mathcal{P}(C_i)$ et $C_i \in COB$ ensemble des classes d'objets.

Application TEXT

Soit COP l'ensemble des classes d'opérations d'une organisation.

Soit OP l'ensemble des applications de profil de type

$S \times P_1 \times \ldots P_n \longrightarrow\!\!\!\!\!\rightarrow F$ c'est-à-dire les applications dont la partie droite du profil contient S et un certain produit d'éléments P_i ou $P_i \in \mathcal{P}(C_i)$ et $C_i \in COB$.

$$TEXT : COP \longrightarrow\!\!\!\!\!\rightarrow OP$$

TEXT associe à chaque classe d'opérations son texte.

TEXT(op) est noté $TEXT_{op}$

Application MOD

$$MOD : COP \longrightarrow\!\!\!\!\!\rightarrow \mathcal{P}(COB)$$

MOD associe à chaque classe d'opérations, les classes d'objets auxquelles appartiennent les objets modifiés par les opérations de la classe.

Elle représente l'ensemble des associations MODIFIE entre classes d'opérations et classes d'objets.

Exemple

Soit la classe d'opérations MODSTOCK modification du stock d'un produit.

$text_{MODSTOCK} : S \times PRO \times COM \longrightarrow\!\!\!\!\!\rightarrow F$

PRO classe des produits

COM classe des commandes

Soient $pro \in PRO \qquad com \in COM$

$text_{MODSTOCK}(s,pro,com) = (pro,s'(pro))$

$s'(pro) = E(s(pro),s(com))$

E : une expression que nous ne détaillons pas ici.

MOD(MODSTOCK) = PRO

III.2.4.- Classes d'associations DECLENCHE

Rappelons que le modèle descriptif revient à exprimer

la dynamique de l'organisation par une structure entre classes de phénomènes de catégories différentes.

En particulier par des associations DECLENCHE entre classes d'événements et classes d'opérations.

Une association DECLENCHE associe une classe d'événements à toutes les classes d'opérations auxquelles doivent appartenir les opérations déclenchées par les événements de la classe.

Nous exprimons ces associations par l'application CDECL suivante :

Application CDECL

$$CDECL : CEV \longrightarrow\!\!\!\!\!\!\rightarrow \mathcal{P}(COP)$$

Exemple :

Soit la classe d'arrivée de commande ARRCOM
la classe d'opérations MODSTOCK

$$CDECL(ARRCOM) = MODSTOCK$$

signifie : la classe d'événements ARRCOM déclenche des opérations de la classe MODSTOCK.

Les propriétés condition de déclenchement et facteur de déclenchement des associations DECLENCHE sont exprimées également par des applications.

Soit G l'ensemble des applications de profil : $S \longrightarrow\!\!\!\!\!\!\rightarrow BOOL$.

Application COND

$$COND : CEV,COP \longrightarrow\!\!\!\!\!\!\rightarrow G$$

COND associe à tout couple $CE,CP \in CEV,COP$, lié par CDECL une fonction booléenne qui doit avoir la valeur vraie pour qu'un événement de la classe CE déclenche effectivement une opération de la classe CP.

On note $COND(CE,CP)$ $cond_{CE,CP}$

Exemple

$$cond_{ARRCOM,MODSTOCK} : S \longrightarrow\!\!\!\!\!\!\rightarrow BOOL$$

est un prédicat qui définit quel doit être l'état du système pour qu'une commande puisse être livrée.

Application FAC

Soit D un ensemble d'applications dont la partie gauche du profil est $S \times S$ et la partie droite l'ensemble des parties d'un produit cartésien de certains sous-ensembles de classes d'objets.

Par exemple $d \in D$ a pour profil : $S \times S \longrightarrow\!\!\!\!\!\!\rightarrow \mathcal{P}(P_1 \times \ldots P_n)$

$$P_i \in \mathcal{P}(C_i) \quad C_i \in COB$$

$$FAC : CEV,COP \longrightarrow\!\!\!\!\!\!\rightarrow D$$

FAC associe à tout couple $CE,CP \in CEV,COP$ et lié par CDECL une application, qui pour tout événement (s,s') détermine les paramètres de chacune des opérations de la classe CP à exécuter.

$FAC(CE,CP)$ est noté $fac_{CE,CP}$

Le profil de $fac_{CE,CP}$ est lié à celui du texte de CP

En effet si $text_{CP} : S \times P_1 \ldots P_q \longrightarrow\!\!\!\!\!\!\rightarrow F$

alors $fac_{CE,CP} : S \times S \longrightarrow\!\!\!\!\!\!\rightarrow \mathcal{P}(P_1 \times \ldots P_q)$

Exemple :

$$fac_{ARRCOM,MODSTOCK} : S \times S \longrightarrow\!\!\!\!\!\!\rightarrow PRO,COM$$

puisque $text_{MODSTOCK} : S \times PRO \times COM \longrightarrow\!\!\!\!\!\!\rightarrow F.$

III-2.5.- Expression de DECL

$$DECL(s,s') = \bigcup_{\substack{i\in I\\ j\in J\\ k\in K}} text_{op_i^j}(s', fac_i^j\ k)$$

avec $I : \{i /_{pred_{CE_i}}(s,s')\}$ où $pred_{CE_i} = PRED(CE_i)$

$\forall i \in I, J : \{j /_{op_i^j} \in CDECL(CE_i)$ et $cond_{CE_i,op_i^j}(s')\}$

où $cond_{CE_i,op_i^j} = COND(CE_i, op_i^j)$

$\forall i \in I, \forall j \in J, K : \{k /_{fac_i^j k} \in fac_{CE_i,op_i^j}(s,s')\}$

où $fac_{CE_i,op_i^j} = FAC(CE_i, op_i^j)$

Signification

- l'événement (s,s') est composé d'un ensemble d'événements appartenant aux classes CE_i telles que $pred_{CE_i}(s,s')$.
- la conséquence de (s,s') est la réunion des conséquences de chacun de ces événements.
- c'est l'exécution d'un certain nombre d'opérations.
- ces opérations appartiennent aux classes d'opérations OP_i^j liées aux classes d'événements CE_i par CDECL et telles que dans l'état final s' leur condition de déclenchement $cond_{CE_i\ OP_i^j}$ soit satisfaite.
- pour chaque classe OP_i^j le nombre d'opérations à exécuter ainsi que leurs paramètres est défini par $fac_{CE_i,\ OP_i^j}$.

CONCLUSION

Le modèle descriptif définit un ensemble de phénomènes réels à représenter dans un système d'information afin d'obtenir une représentation d'une organisation sous son double aspect statique et dynamique.

Ces phénomènes sont des classes appartenant aux trois catégories OBJET, OPERATION et EVENEMENT et des associations entre ces classes, CONSTATE, DECLENCHE et MODIFIE.

La formalisation associée à ce modèle montre sa complétude vis à vis de la modélisation de la dynamique.

En effet elle montre que la donnée de ces classes et de ces associations permet de définir la conséquence de tout changement d'état survenant dans l'organisation.

Cette formalisation met aussi en évidence la complexité des applications à écrire dans le cas général où aucune règle n'est imposée pour structurer les catégories en classes de phénomènes.

Cette complexité est un obstacle à la production de schémas conceptuels canoniques au sens où nous les avons définis au chapitre 1 car elle risque de rendre impossible leur contrôle.

Elle justifie la recherche d'un autre modèle, cas particulier du modèle descriptif, définissant de manière formelle les classes de phénomènes et leurs associations afin de parvenir à une représentation plus simple et plus aisément contrôlable de la réalité. Ce modèle est le modèle conceptuel que nous présentons au chapitre suivant.

CHAPITRE 3 : UN MODÈLE CONCEPTUEL POUR LA STRUCTURATION DES SI

INTRODUCTION

Le modèle conceptuel de SI est un modèle

- de type relationnel
- normatif
- intégrant le temps
- utilisant des types.

Cette définition signifie que :

- Le concept de base du modèle est la relation au sens du modèle relationnel de CODD (COD 70). C'est-à-dire : Tout phénomène réel, quelle que soit sa catégorie est représenté dans le modèle par tout ou partie d'une ou plusieurs relations.

- Le modèle impose une normalisation dans la représentation des phénomènes réels. Cette normalisation concerne la structuration en classes de chaque catégorie de phénomènes. Nous imposons une structuration en classes de phénomènes élémentaires qui a pour conséquence la simplification de l'expression de la dynamique par rapport à celle obtenue à l'aide du modèle descriptif.

- Le temps est intégré dans le modèle relationnel par la définition d'un nouveau type de dépendance fonctionnelle : la dépendance fonctionnelle permanente, et par une forme normalisée associée, la relation permanente en $3^{ème}$ forme normale. Cette nouvelle forme de relation permet une représentation historique, complète et minimale de toute classe de phénomènes réels.

- Le modèle inclut des types de relations et de constituants afin de prendre en compte les différences sémantiques existant entre les catégories de phénomènes et leurs propriétés.

- Dans la première partie du chapitre nous faisons un rappel sur les principaux résultats concernant le modèle relationnel en nous limitant volontairement à ceux qui nous sont utiles par la suite et nous introduisons les nouveaux concepts y permettant l'intégration du temps. Enfin nous introduisons les types de relation, et les types de constituants.

- Les trois parties suivantes sont consacrées respectivement à la définition des concepts utilisés dans le modèle pour la représentation de chacune des trois catégories de phénomènes OBJET, OPERATION et EVENEMENT.

- La cinquième partie donne la nouvelle expression de l'application DECL compte tenu des simplifications apportées par la normalisation des classes de phénomènes.

- Enfin la sixième partie définit les concepts permettant la représentation historique complète et minimale de cette application dans le SI.

I - INTRODUCTION DU TEMPS ET DES TYPES DANS LE MODELE CONCEPTUEL

I-1.- Rappels sur le modèle relationnel

I-1.1.- Les notions de relation, de schéma de relation, de contraintes d'intégrité, de clé

Définition mathématique d'une relation

Soient $D_1, D_2, \ldots D_n$ n ensembles.

Une relation R sur ces n ensembles est un sous-ensemble du produit cartésien $D_1 \times D_2 \ldots \times D_n$.

Chaque élément de R est donc un n-uplet $<d_1, d_2 \ldots d_n>$ tel que $d_1 \in D_1$, $d_2 \in D_2 \ldots d_n \in D_n$.

Schéma de relation

Un schéma de relation définit une classe de relations de même forme.

Plus précisément il s'agit de la donnée

- de n ensembles $D_1, D_2 \ldots D_n$
- de n noms distincts $A_1, A_2, \ldots A_n$ appelés constituants ou attributs de la relation, servant à désigner les ensembles

D_i de manière non ambigüe. D_i est appelé le domaine du constituant A_i

- de contraintes d'intégrité.

Intention et extension de relation

Le schéma de relation définit l'intention de la classe de relations, il en donne sa signification. Il est noté :

$R(A_1, \ldots A_n)$ R est le nom donné à toutes les relations de la classe

A_i est le nom d'un constituant.

L'ensemble des n-uplets d'une relation de la classe constitue son extension. Elle apparaît comme un tableau où une ligne correspond à un n-uplet et une colonne à un constituant.

Exemple

Le schéma de relation

COMMANDE(NOCOM,NOCLIENT,NOPRODUIT,QTECDEE,DATECOM)

définit une classe de relations représentant les commandes passées par les clients dans une organisation.

Une extension d'une relation de la classe peut être par exemple :

n-uplets

NOCOM	NOCLIENT	NOPRODUIT	QTECDEE	DATCOM
15	25	30	20	10.1.78
15	25	40	10	10.1.78
15	25	60	35	10.1.78
16	30	70	45	20.2.78
16	30	80	70	20.2.78
17	40	30	15	30.3.78
17	40	70	18	30.3.78
18	25	40	30	25.4.78

Intérêt de l'introduction des noms de constituants

L'intérêt d'introduire des noms des constituants distincts $A_1 \ldots A_n$ (qui dans la notation mathématique habituelle ne sont autres que les entiers 1,2 ... n) est d'ordre technique et d'ordre sémantique.

Sur le plan sémantique, ils permettent de désigner de façon identique une même propriété de phénomène réel apparaissant dans plusieurs schémas de relation, quel que soit son emplacement dans l'ensemble des constituants de chacun des schémas.

Par exemple dans les schémas

COMMANDE(NOCOM,NOCLIENT,NOPRODUIT,QTECDEE,DATCOM)

CLIENT(NOCLIENT,ADRCLIENT,TELCLIENT)

NOCLIENT représente dans les deux schémas la même propriété : le numéro d'un client de l'entreprise.

Sur le plan technique, ils permettent d'identifier un n-uplet à un ensemble de n couples formés d'un constituant A_i et d'une valeur a_i de D_i, noté

$$<A_1 : a_1, \ldots A_i : a_i, \ldots A_n : a_n>$$

sans qu'aucun ordre sur ces couples ne soit nécessaire.

Ceci permet de définir des ensembles de constituants et en particulier la projection d'un n-uplet sur un ensemble.

Projection d'un n-uplet sur un ensemble de constituants

Soit une relation de schéma $R(A_1 \ldots A_n)$

Soit X un ensemble de constituants $X = \{A_{i_1}, \ldots A_{i_p}\}$ de R

Soit un n-uplet de R $<A_1 : a_1 \ldots A_n : a_n>$

La projection sur X de ce n-uplet est le p-uplet $<A_{i_1} : a_{i_1}, \ldots A_{i_p} : a_{i_p}>$.

Union de deux ensembles de constituants

Soit une relation de schéma $R(A_1 \ldots A_n)$.

Soient X et Y deux ensembles disjoints de constituants de R

$$X = \{A_{i_1} \ldots A_{i_p}\} \qquad Y = \{A_{j_1} \ldots A_{j_q}\}$$

XUY noté souvent XY désigne l'ensemble de constituants

$$\{A_{i_1} \ldots A_{i_p}, A_{j_1} \ldots A_{j_q}\}$$

Si x est la projection sur X d'un n-uplet et y la projection sur Y du même n-uplet, on note $<x,y>$ la projection sur XY de ce n-uplet.

Contrainte d'intégrité

Une contrainte d'intégrité est une propriété qui doit être vérifiée par toute extension associée à un schéma de relation pour que la relation correspondante appartienne à la classe de relations définie par le schéma.

La classe de relations d'un schéma est donc formée des relations sur $D_1, \ldots D_n$ qui vérifient les contraintes d'intégrité, définies dans le schéma.

Les contraintes d'intégrité peuvent être exprimées :

- sous la forme de dépendances entre constituants

ou

- sous la forme de prédicats portant sur les valeurs d'un ou plusieurs constituants des n-uplets d'une relation exprimant des règles de cohérence ou de complètude de l'aspect de la réalité représentè par les relations de la classe.

Dans ce travail nous nous intéressons essentiellement au premier type de contraintes. Nous rappelons par la suite les différents types de dépendances qui ont été introduits dans le modèle relationnel.

Notion de superclé et de clé

Un ensemble de constituants X est une superclé d'une relation R si deux n-uplets distincts de R ont des projections différentes sur X.

X est une clé si elle est une superclé minimale c'est-à-dire si il n'existe pas d'ensemble $Y \subset X$ qui soit lui-même une superclé.

Remarque

Dans la littérature ces notions ont reçu des noms différents : DELOBEL (DEL 73) les appelle respectivement : ensemble de constituants discriminatoire et index.
CODD (COD 71) parle de clé primaire et de clé primaire non redondante.

Imposer que,dans toute relation appartenant à une classe définie par un certain schéma, un ensemble de constituants soit une clé est une contrainte d'intégrité.

Il se peut qu'un schéma de relation admette plusieurs clés, il est de coutume alors de demander au concepteur d'en choisir une particulière appelée clé principale qui servira pour repérer les n-uplets de la relation lors de leur manipulation.

Dans un schéma de relation la contrainte d'intégrité qui indique qu'un ensemble de constituants constitue la clé principale est exprimée en le soulignant dans la notation du schéma.

Exemple : Dans le schéma de relation

CLIENT(NOCLIENT,NOMCLIENT,ADRCLIENT)

le soulignement du constituant NOCLIENT indique que ce constituant doit être une clé pour toute relation obéissant à ce schéma, et qu'il en constitue la clé principale c'est-à-dire qu'il représente la propriété identifiant principal de la classe d'objets client.

I-1.2.- Opérations sur les relations

CODD (COD 71) a défini une algèbre relationnelle comprenant un ensemble d'opérations qui, appliquées à des relations, produisent d'autres relations.

La définition de ces opérations fait intervenir l'ensemble des constituants d'une relation.

On note R(A) une relation où A désigne l'ensemble de ses constituants.

Nous donnons ici la définition des deux opérations qui nous serviront par la suite : la projection et le produit.

Projection

Soient R(A) une relation et X un sous-ensemble non vide de A.

La projection de R suivant X notée R[X] est l'ensemble des projections sur X des n-uplets de R ; autrement dit

$$R[X] = \{<x>/\exists y : <x,y> \in R\}$$

Produit

Soient $R_1(A)$ et $R_2(B)$ deux relations.

Le produit de ces deux relations est la relation R(AUB) formée des n-uplets dont la projection suivant A appartient à R_1 et la projection suivant B appartient à R_2.

En notant $Y = A \cap B$, $X = A-Y$, $Z = B-Y$

$$R = R_1(X,Y) \otimes R_2(Y,Z) = \{<x,y,z>/<x,y> \in R_1 \text{ et } <y,z> \in R_2\}$$

Cette opération recouvre deux opérations de Codd :

- le produit cartésien si $Y = A \cap B = \emptyset$
  alors $R = \{<x,z>/<x> \in R_1 \text{ et } <z> \in R_2\}$
- le "join" si Y n'est pas vide.

I-1.3.- Décomposition d'une relation et types de dépendances

Le modèle relationnel étant utilisé pour construire des bases de données, les chercheurs de ce domaine se sont posés le problème de trouver une représentation des informations à stocker dans la base de données sous forme de relations, qui soit la meilleure possible.

La décomposition d'une relation en plusieurs de ses projections est apparue comme une solution à ce problème. En effet si une relation est décomposable en deux ou davantage de ses projections cela signifie qu'il est possible de stocker dans la base de données, à la place de la relation R initiale,uniquement ses projections. Ce qui correspond souvent à une diminution du volume d'information stocké et à une manipulation plus aisée des données, comme cela a été montré sur des exemples par CODD (COD 71).

Dans ce paragraphe, nous rappelons les résultats obtenus concernant la décomposition d'une relation en deux de ses projections, dans le suivant nous introduisons les formes particulières de relations dites "formes normales" liées aux différents types de dépendances.

Définition d'une relation décomposable (DEL 73)

Soit $R(X,Y,Z)$ une relation

R est décomposable s'il existe deux relations S et T telles que

a) S et T sont des projections de R $S = R[X,Y]$, $T = R[X,Z]$

b) le produit de S et T est R : $R = S \otimes T$.

Condition nécessaire et suffisante pour qu'une relation soit décomposable (DEL 73)

Soit une relation $R(X,Y,Z)$.

Pour que R soit décomposable en $R[X,Y]$ et $R[X,Z]$ il faut et il suffit que $\forall x \in R[X]$ et $\forall y \in R[Y]$ on ait

$$R[x,y,Z] = R[x,Z] \quad \text{où} \quad R[x,y,Z] = \{z \mid \langle x,y,z\rangle \in R\}$$
$$\text{et} \quad R[x,Z] = \{z \mid \langle x, z\rangle \in R[X,Z]\}$$

I-1.3.1.- Notion de dépendance fonctionnelle

Si $\forall x \in R[X]$, $R[x,Y]$ contient un seul élément alors la condition précédente est satisfaite.

On dit alors qu'il existe une dépendance fonctionnelle entre X et Y.

Définition de la dépendance fonctionnelle

Etant donnée une relation R(A) et deux sous-ensembles X et Y de A on dit que Y dépend fonctionnellement de X ou qu'il existe une dépendance fonctionnelle entre X et Y si et seulement si,lorsque deux n-uplets de R ont même projection sur X, ils ont également même projection sur Y.

Fonction associée à une dépendance fonctionnelle

Lorsqu'il existe dans une relation R(A) une dépendance fonctionnelle entre deux sous-ensembles X et Y de A, il existe entre les domaines E_X et E_Y de X et Y une fonction partielle qui,à tout élément $x \in (E_{X \cap R[X]})$, fait correspondre l'unique valeur $y \in E_Y$ telle que $\langle x,y\rangle \in R[X,Y]$.

Notation de la dépendance fonctionnelle

Une dépendance fonctionnelle entre X et Y dans une relation R est une contrainte d'intégrité associée au schéma de relation de R ; elle est notée $X \longrightarrow\!\!\!\!\!\rightarrow Y$.

Types de dépendances fonctionnelles

Soit R(A) une relation et X et Y deux sous-ensembles de A liés par une dépendance fonctionnelle $X \longrightarrow\!\!\!\!\!\rightarrow Y$

$X \longrightarrow\!\!\!\!\!\rightarrow Y$ peut appartenir aux types suivants :

- Dépendance fonctionnelle élémentaire

  $X \longrightarrow\!\!\!\!\!\rightarrow Y$ est élémentaire s'il n'existe pas $X' \subset X$ tel que $X' \longrightarrow\!\!\!\!\!\rightarrow Y$ dans R.

- Dépendance fonctionnelle directe

  $X \longrightarrow\!\!\!\!\!\rightarrow Y$ est directe s'il n'existe pas $Z \subset A$ tel que

$X \longrightarrow\!\!\!\!\!\rightarrow Z$ existe, $Z \longrightarrow\!\!\!\!\!\rightarrow Y$ existe et $Z \longrightarrow\!\!\!\!\!\rightarrow X$ n'existe pas dans R.

- Dépendance fonctionnelle canonique

$X \longrightarrow\!\!\!\!\!\rightarrow Y$ est canonique si Y est réduit à un seul élément de A.

- Dépendance fonctionnelle triviale

$X \longrightarrow\!\!\!\!\!\rightarrow Y$ est triviale si $Y \subset X$

Exemple

Dans la relation COMMANDE présentée au paragraphe I-1.1 de schéma COMMANDE(NOCOM,NOCLIENT,NOPRODUIT,QTECDEE,DATCOM) il existe les dépendances fonctionnelles

$$\text{NOCOM} \longrightarrow\!\!\!\!\!\rightarrow \text{NOCLIENT}$$
$$\text{NOCOM} \longrightarrow\!\!\!\!\!\rightarrow \text{DATCOM}$$
$$\text{NOCOM,NOPRODUIT} \longrightarrow\!\!\!\!\!\rightarrow \text{QTECDEE}$$

L'existence d'une dépendance fonctionnelle entre deux ensembles de constituants X et Y d'une relation R(A) est une condition suffisante pour que R soit décomposable en R[X,Y] et R[X,Z] où Z est le complémentaire de XUY dans A.

Une décomposition d'une relation s'appuyant sur les dépendances fonctionnelles existant entre ses ensembles de constituants est appelée décomposition fonctionnelle.

I-1.3.2.- Notion de dépendance multivaluée

Lorsque la condition de décomposition est satisfaite dans une relation R(X,Y,Z) on dit qu'il existe une dépendance multivaluée entre X et Y.

Il en existe aussi une entre X et Z.

Définition de la dépendance multivaluée (FAG 77)

Soient une relation R(A), X et Y deux sous-ensembles disjoints de A et Z le complément de XUY dans A.

Il existe une dépendance multivaluée entre X et Y dans R si et seulement si, lorsque deux n-uplets <x,y,z> et <x,y',z'> appartiennent à R, alors les n-uplets <x,y,z'> et <x,y',z> appartiennent également à R.

Notation

Une dépendance multivaluée entre deux sous-ensembles de constituants X et Y de la relation R(A) est une contrainte d'intégrité associée au schéma de la relation de R notée

$$X \longrightarrow\!\!\!\!\!\rightarrow Y.$$

Exemple

Soit le schéma de relation R (COURS,PROFESSEUR,LIVRE)
Il existe dans R la dépendance multivaluée

$$\text{COURS} \longrightarrow\!\!\!\!\!\rightarrow \text{PROFESSEUR}$$

si un cours peut être assuré par un ensemble déterminé de professeurs indépendamment du ou des livres illustrant le cours, c'est-à-dire si chaque professeur assurant un cours utilise tous les livres associés au cours.

L'existence d'une dépendance multivaluée entre deux sous-ensembles de constituants X et Y d'une relation R(X,Y,Z) constitue une condition nécessaire et suffisante pour que R soit décomposable en R[X,Y] et R[X,Z].

I-1.4.- Décomposition d'une relation et formes normales

Parallèlement ou conjointement à l'étude du problème de décomposition d'une relation, les chercheurs ont introduit des formes particulières de relation appelées formes normales, toujours dans l'objectif de parvenir à une meilleure représentation des informations.

L'amélioration apportée par ces formes a toujours été mise en évidence sur des exemples. Ceci explique qu'il y a eu successivement

plusieurs formes normales proposées par les auteurs, chaque nouvelle forme introduite ayant la particularité d'améliorer certaines représentations par rapport à celles obtenues à l'aide de formes précédentes. Les améliorations portent soit sur le volume d'information stocké, soit sur les conditions de manipulation des données.

CODD (COD 71) a introduit dans cet esprit trois formes normales appelées première, deuxième et troisième forme normale (1FN, 2FN, 3FN). Puis Boyce et Codd ont défini la forme normale dite de Boyce-Codd et Fagin (FAG 77) la quatrième forme normale (4FN).

Définitions des formes normales

Définition de la troisième forme normale (BEE 78)

Une relation est en troisième forme normale si aucun constituant n'appartenant à aucune clé de la relation, n'est transitivement dépendant d'aucune de ses clés.

Note : Dans une relation R(A) un constituant Z est transitivement dépendant d'un ensemble de constituants X s'il existe un ensemble Y de constituants tel que $X \longrightarrow Y$ existe dans R

$Y \longrightarrow X$ n'existe pas

et $Y \longrightarrow Z$ existe et $Z \notin Y$

Définition de la forme normale de Boyce Codd (BCFN) (BEE 78)

Une relation R(A) est en forme normale de Boyce-Codd si, pour toute dépendance fonctionnelle non triviale existant entre deux ensembles de constituants X et Y de R, X est une superclé de R.

Définition de la quatrième forme normale (4FN) (FAG 77)

Une relation R(A) est en quatrième forme normale si, pour toute dépendance multivaluée non triviale existant entre deux ensembles de constituants X et Y de R, X est une superclé de R.

Remarque

Ces trois formes sont liées par le fait que 4FN implique BCFN et que BCFN implique 3FN, les implications réciproques n'étant pas vraies.

Décomposition et formes normales

FAGIN (FAG 77) a montré la proposition suivante :

"Toute relation qui n'est pas en 4FN peut être décomposée en un produit de relations en 4FN".

Cette proposition, associée aux avantages pratiques des relations en formes normales, justifie que les concepteurs de bases de données aient fixé comme contrainte à un schéma conceptuel de base de données d'être exprimé à l'aide de relations en formes normales.

Le problème de la conception d'un tel schéma a reçu un certain nombre de solutions que nous rappelons au paragraphe suivant.

I-1.5.- Les méthodes de conception de schéma conceptuel de bases de données

Notion de schéma conceptuel

Le schéma conceptuel d'une base de données exprimé à l'aide du modèle relationnel, est la collection des schémas des relations de la base.

Hypothèse de la relation universelle

Les méthodes de conception de schéma conceptuel de bases de données font toutes l'hypothèse qu'il est possible de représenter la réalité concernée par la base de donnée par une unique relation appelée relation universelle.

Cette hypothèse est contreversée mais il a été montré (BER 76) que dans des cas particuliers où elle ne semblait pas s'appliquer il était possible, moyennant l'introduction de constituants synonymes, de la satisfaire.

Les deux classes de méthodes

Les méthodes proposées pour concevoir un schéma conceptuel de base de données, appelées aussi méthodes de normalisation, résolvent toutes le problème suivant :

- Passer d'un schéma conceptuel constitué d'un unique schéma de relation (celui de la relation universelle) à un schéma conceptuel constitué par une collection de schémas de relation en forme normale.

Elles se classent en méthodes synthétiques (DEL 73), (FLO 77), (BER 76), (BEE 78) et méthodes de décomposition (COD 70), (FAG 77).

Plusieurs comparaisons de ces méthodes ont été effectuées (FAG 77), (BEE 78). DELOBEL (DEL 80) en rappelle les caractéristiques et les différences.

Nous avons choisi dans ce travail d'utiliser une méthode de type synthétique pour une raison pratique :

- les algorithmes associés aux méthodes synthétiques sont plus simples et plus rapides que ceux associés aux méthodes de décomposition et conduisent à une solution unique dans la plupart des cas.

Nous donnons les grandes lignes de la méthode choisie dans le paragraphe suivant.

I-1.6.- Méthode synthétique de conception de schéma conceptuel

Principe de la méthode synthétique

La méthode proposée par BERNSTEIN, appelée méthode synthétique, permet de passer d'un schéma conceptuel S_0, contenant un unique schéma de relation, à un schéma conceptuel S_F contenant une collection de schémas de relation en troisième forme normale.

Notons

$S_0 = \{R_0(A), D, F_0\}$

A : ensemble des constituants de la relation R_0

D : ensemble des domaines des éléments de A.

F_0 : ensemble des dépendances fonctionnelles existant entre les éléments de A.

S_F : $\{(<R_i(A_i), D_i, F_i>) \mid i = 1,n\}$

avec $A_i \subset A$

$D_i \subset D$

F_i : ensemble des dépendances fonctionnelles existant entre les éléments de A_i dans R_i.

Conditions d'application

Tout constituant de R_0 apparaît dans F_0,
c'est-à-dire $\forall X \in A$ il existe une DF $\in F_0$ $\quad Y \longrightarrow\!\!\!\!\!\rightarrow Z$
telle que :

- soit $X \in Y$
- soit $Z = X$.

Cette condition d'application, présentée par BERNSTEIN comme une limite de la méthode, est en fait toujours vérifiée si on considère que F_0 contient pour tout constituant $X \in A$, ne satisfaisant pas à la condition, la DF $X \longrightarrow\!\!\!\!\!\rightarrow \emptyset$.

Caractéristiques de S_F

- S_F est minimal

La minimalité concerne :

- le nombre n de schémas de relation de S_F, elle signifie que les relations $R_i(A_i)$ sont maximales c'est-à-dire qu'il n'existe pas dans S_F deux relations R_i et R_j ayant une clé en commun.
- l'ensemble des DF de S_F, c'est-à-dire $\bigcup_{i=n} F_i = \hat{F}_0$ ou $\hat{F}_0$ est la couverture minimale du graphe associé à F_0. (Dans ce graphe, chaque constituant, ou ensemble de constituants, source ou but de DF est un noeud et chaque DF un arc).

- S_F est équivalent à S_0

L'équivalence porte sur - l'ensemble des constituants puisque $A = \bigcup_{i=1,n} A_i$

- l'ensemble des DF puisque $\bigcup_{i=1,n} F_i = \hat{F}_0$

Remarquons que, cependant, on n'a pas toujours l'équivalence au sens de la décomposition c'est-à-dire $R_0 = R_1 \otimes R_2 \otimes .. .. R_i \otimes R_n$.

Algorithme SYN_1 permettant de passer de S_0 à S_F

L'algorithme SYN_1 permettant de passer de S_0 à S_F utilise les propriétés de réflexivité, augmentation et pseudotransitivité des dépendances fonctionnelles formalisées par DELOBEL (DEL 73) et ARMSTRONG (ARM 74).

Il comprend 4 étapes :

1) La recherche d'une couverture minimale $\hat{F}_0$ du graphe associé à F_0.

2) La partition de $\hat{F}_0$ en sous-ensembles F_i, i = 1,... n formés des DF ayant la même partie gauche.

3) La fusion des clés équivalentes et l'élimination des dépendances transitives qui en résultent,qui conduisent à un nouvel ensemble de groupes F'_i, i = 1 ... n.

4) La construction des relations de S_F à raison d'une relation par groupe F'_i. La partie gauche du groupe F'_i est la clé de la relation ainsi générée.

Dans la suite de ce travail nous utiliserons également les résultats suivants obtenus par DELOBEL , ADIBA et LEONARD (DEL 76) :

1) L'ensemble des dépendances fonctionnelles élémentaires directes et canoniques de la relation R_0 constitue une couverture minimale $\hat{F}_0$ de F.

2) Si $\hat{F}_0$ est sans circuit pour R_0 alors la partition de $\hat{F}_0$ en sous-ensembles suivant le principe défini ci-dessus à l'étape 2 de l'algorithme est unique.

Ces résultats permettent d'utiliser l'algorithme SYN2 de synthèse de relations en 3ème forme normale suivant :

- soit un schéma S_0 $(R_0,\hat{F}_0)$ où $\hat{F}_0$ est l'ensemble des dépendances fonctionnelles élémentaires directes et canoniques existant entre les attributs de R_0 :

- partitionner $\hat{F}_0$ en sous-ensembles F'_i (comme dans les étapes 2 et 3 de l'algorithme SYN1).

- construire les schémas de relations R_i de S_F à raison d'une par groupe F'_i.

Si $\hat{F}_0$ est sans circuit l'étape n° 3 n'est pas nécessaire et l'algorithme SYN3 simplifié réduit à l'étape 2 conduit à un schéma conceptuel minimal.

Exemple d'application

Prenons comme schéma S_0 le schéma de relation suivant :

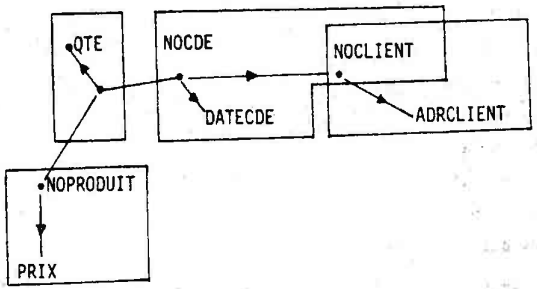
COMMANDE(NOCDE,NOCLIENT,ADRCLIENT,NOPRODUIT,PRIX,QTE,DATECDE)

et les contraintes NOCDE ⟶ NOCLIENT
NOCLIENT ⟶ ADRCLIENT
NOPRODUIT ⟶ PRIX
NOCDE ⟶ DATECDE
NOCDE,NOPRODUIT ⟶ QTE

constituant l'ensemble F_0.

Notons que F_0 ne contient que des DF, élémentaires, directes et canoniques.

Dessinons le graphe associé à F_0



Ce graphe est sans circuit.

La partition de F_0 en sous-ensembles F_i, i = 1,... n formés des DF ayant la même partie gauche conduit aux sous-graphes inclus dans les rectangles dessinés sur le graphe initial.

On en déduit la collection de relations suivantes constituant le schéma S_F.

R (NOCDE,DATECDE,NOCLIENT)

R (NOCLIENT,ADRCLIENT)

R (NOPRODUIT,PRIX)

R (NOCDE,NOPRODUIT,QTE)

I-2.- Introduction du temps dans le modèle relationnel

I-2.1.- Pourquoi introduire le temps dans le modèle relationnel

Rappelons que nous voulons utiliser le modèle relationnel pour exprimer le schéma conceptuel d'un système d'information d'une organisation.

Or,comme nous l'avons remarqué au chapitre 1, les organisations sont des systèmes évoluant au cours du temps dont la connaissance de l'histoire est indispensable aux gestionnaires.

Pour parvenir à une représentation de cette histoire ayant les qualités que nous avons assignées à un SI, il nous est apparu nécessaire d'introduire rigoureusement la notion de temps dans les concepts du modèle relationnel.

I-2.2.- L'utilisation courante du modèle relationnel correspond à une représentation instantanée des phénomènes réels

L'utilisation courante du modèle relationnel conduit à une représentation statique des phénomènes.

En effet, dans une base de données relationnelle, une relation représente l'état d'une classe de phénomènes à un instant donné, un n-uplet représentant l'état d'un phénomène particulier de la classe à cet instant.

L'évolution de la classe réelle entraîne au cours du temps des adjonctions, des modifications ou des suppressions de n-uplets dans la relation.

Aucune caractéristique de cette évolution, comme par exemple les dates auxquelles les modifications se sont produites, n'est en général représentée.

Remarquons que le modèle relationnel n'interdit pas cette représentation, mais elle est laissée à l'initiative du concepteur, en particulier aucune formalisation ne lui est proposée pour trouver une représentation de l'évolution des phénomènes au cours du temps qui soit minimale.

Remarquons de plus que toutes les définitions que nous avons données dans la partie précédente consacrée aux rappels concernant le modèle relationnel, ne font pas référence au temps.

Implicitement elles sont valables pour une relation définie à un instant donné.

I-2.3.- Nécessité de nouveaux concepts pour passer à une représentation dans le temps

Pour passer d'une représentation instantanée des phénomènes réels à une représentation de leur évolution au cours du

temps par des relations il faut :

- d'une part représenter les propriétés temporelles des phénomènes réels par des constituants de relation : ces propriétés sont les dates de changement d'état des phénomènes ; date de création, date de modification, etc...

- d'autre part conserver dans les relations toutes les valeurs successives prises par les propriétés des phénomènes au cours du temps et représentées par les constituants des relations.

A cette condition il sera possible de retrouver dans la collection de relations :

- soit l'état d'un phénomène ou d'une classe à une date donnée
- soit la succession des états pris par un phénomène sur une certaine période c'est-à-dire son histoire.

Pour parvenir à une telle représentation qui satisfasse encore au critère de minimalité que nous imposons au SI, nous avons introduit le temps dans le modèle relationnel par trois concepts :

- le concept de relation permanente
- les concepts de dépendance fonctionnelle permanente et de dépendance multivaluée permanente qui permettent de normaliser une représentation effectuée à l'aide de relations permanentes.

I-2.4.- Relation permanente et dépendance fonctionnelle permanente

I-2.4.1.- Relation dépendant du temps

Définition

Soit un schéma de relation définissant une classe S de relations par

- n constituants $A_1, A_2 \ldots A_n$
- n ensembles $D_1, D_2 \ldots D_n$ ou D_i est le domaine de A_i

- un ensemble Γ de contraintes d'intégrité.

Soit un ensemble T muni d'une relation d'ordre dont les éléments représentent des dates.

Une relation dépendant du temps est une fonction R de T dans l'ensemble des parties de $D_1 \times D_2 \ldots \times D_n$ telle que

$$\forall t \in T \quad R(t) \text{ notée } R_t \in S$$

R_t est l'extension de R à l'instant t

I-2.4.2.- Relation permanente

Définition

Une relation permanente est une relation dépendant du temps telle que $\forall t$ et $t' \in T : t' < t$ on ait

$$R_{t'} \subset R_t.$$

Signification

Dans une relation permanente la seule modification qui puisse apparaître entre deux extensions est l'adjonction de n-uplets.

I-2.4.3.- Dépendance fonctionnelle permanente DFP

Dépendance fonctionnelle dans une relation dépendant du temps

Soit une relation $R(A)$ dépendant du temps.
Soient X et Y deux sous-ensembles de A.
Soit $t \in T$ défini précédemment.
Soit R_t l'extension de R à l'instant t.
Il existe une DF entre X et Y dans R si
$\forall t \in T$ il existe dans R_t une DF entre X et Y.

Dépendance fonctionnelle permanente

Avec les mêmes notations que ci-dessus, notons

$$R_T = \bigcup_{t \in T} R_t[X,Y] : \{<x,y> : \exists t : <x,y> \in R_t[X,Y]\}$$

Une dépendance fonctionnelle entre X et Y dans R est une dépendance fonctionnelle permanente si il existe dans R_T une DF entre X et Y.

Cas particulier

Si, dans la relation R, les suppressions de n-uplets sont interdites d'une extension à une autre, alors une DF entre X et Y dans R est une DFP si la relation dépendante du temps R[X,Y] est permanente.

Exemple : Soit la relation CLIENT de schéma

client(noclient, adresse, notel, nomclient)

noclient ⟶ adresse

noclient ⟶ nomclient

noclient ⟶ notel

Soient les extensions de cette relation aux instants t_1, t_2, t_3

CLIENT t_1

noclient	adresse	notel	nomclient
1	PARIS	712	DUPONT
2	LYON	832	DURAND
3	NANCY	764	ALFRED

CLIENT t_2

noclient	adresse	notel	nomclient
1	NANTERRE	243	DUPONT
2	LYON	832	DURAND
3	MARSEILLE	195	ALFRED

CLIENT t_3

noclient	adresse	notel	nomclient
1	NANTERRE	325	DUPONT
2	LYON	832	DURAND
4	NANCY	241	DUPONT
5	NANCY	842	ALBERT

Il est immédiat de vérifier sur ces trois extensions que l'on a pour les trois relations CLIENT t_1 CLIENT t_2 et CLIENT t_3 les DF noclient ⟶ adresse

noclient ⟶ notel

noclient ⟶ nomclient

Parmi ces trois DF seule la DF noclient ⟶ nomclient est permanente pour la relation CLIENT.

Sémantique d'une DFP

L'existence d'une DFP entre deux constituants X et Y d'une relation dépendant du temps signifie qu'au cours du temps à une même valeur de $x \in Dom(X)$ est toujours associée la même valeur de $y \in Dom(Y)$. Si X et Y représentent deux propriétés d'une même classe de phénomènes réels cette DFP signifie que ces deux propriétés sont toujours valorisées ou modifiées simultanément, nous dirons qu'elles ont le même comportement dynamique.

Notation d'une DFP

Une DFP entre deux ensembles de constituants X et Y d'une relation R est notée $X \overset{\cdot}{\longrightarrow} Y$.

Relation permanente et dépendance fonctionnelle permanente

Dans une relation permanente toute DF est une DFP.

Soit R une relation permanente.

Soient X et Y deux sous-ensembles de R tels que $X \twoheadrightarrow Y$ existe dans R alors $\forall t,t' : t' < t \quad R_{t'} \subset R_t$ donc en particulier $R_{t'}[X,Y] \subset R_t[X,Y]$ donc la relation R[X,Y] est permanente donc,d'après le cas particulier de la définition de la dépendance fonctionnelle permanente, $X \twoheadrightarrow Y$ est permanente dans R.

I-2.4.4.- Dépendance multivaluée permanente

Dépendance multivaluée d'une relation dépendant du temps

Soit R une relation dépendant du temps définie sur T.
Soient X et Y deux sous-ensembles de constituants de R.
Il existe une DM entre X et Y dans R
si $\forall t \in T \quad \exists$ une DM entre X et Y dans R_t.

Dépendance multivaluée permanente (DMP)

Avec les mêmes notations, une DM entre X et Y dans R est une DMP si R[X,Y] est permanente.

Interprétation

Une DM entre X et Y dans R est permanente si au cours du temps l'ensemble des valeurs de Y associées à une valeur de X ne peut que croître.

Une dépendance fonctionnelle permanente est une dépendance multivaluée permanente.

Immédiat.

Dans une relation permanente toutes les dépendances multivaluées sont des dépendances multivaluées permanentes.

Ceci provient du fait que si R est permanente $\forall X$ et $\forall Y$ R[X,Y] est permanente.

I-2.4.5.- Normalisation des relations permanentes

Les résultats définis à propos des relations instantanées rappelés dans les paragraphes précédents consacrés au modèle relationnel, sont applicables aux relations dépendant du temps et aux relations permanentes en particulier.

Il est donc possible de définir des formes normales pour les relations permanentes reposant sur les notions de dépendance fonctionnelle permanente et de dépendance multivaluée permanente. Nous pouvons parler de :

- Relation permanente en troisième forme normale
- Relation permanente en forme de Boyce-Codd
- Relation permanente en quatrième forme normale

Les définitions de ces formes normales sont celles données précédemment dans lesquelles les notions de DF et de DM sont remplacées par celles de DFP et de DMP.

La normalisation d'une relation permanente conduit à une collection de relations permanentes

Rappelons que la normalisation consiste à remplacer une relation par plusieurs relations correspondant à certaines de ses projections.

Une projection quelconque d'une relation permanente sur un sous-ensemble de ses constituants est une relation permanente d'où la proposition.

La relation permanente en troisième forme normale est la forme que nous avons retenue dans le modèle conceptuel.

Dans le paragraphe II-2 nous montrons, à propos de la catégorie OBJET, comment passer d'une représentation instantanée exprimée par une relation dépendant du temps à une représentation historique normalisée exprimée par des relations permanentes en troisième forme normale.

I-3.- Les types de relation et les types de constituants

I-3.1.- Définition et utilisation des types de relation

L'analyse descriptive de l'organisation présentée au chapitre précédent montre que nous avons à représenter trois catégories de phénomènes ayant chacune des caractéristiques propres.

Le choix que nous avons fait : représenter toute classe de phénomènes ou d'associations de phénomènes, par des relations, indépendamment de la catégorie à laquelle elle appartient, conduit à une représentation sémantiquement incomplète si la catégorie du phénomène n'est pas représentée.

Cette appartenance à une catégorie ne saurait être représentée par des constituants particuliers de relation sans entraîner une très grande redondance d'information puisque,par définition,dans une même classe,tous les phénomènes appartiennent à la même catégorie.

C'est pourquoi nous avons introduit dans le modèle relationnel la notion de type de relation.

Définition

Un type de relation est un ensemble de relations représentant des phénomènes d'une même catégorie et dont les schémas respectent un certain nombre de contraintes.

Les contraintes associées à un type peuvent être par exemple :

- la forme de la relation
- la présence obligatoire de constituants d'un certain type
- le type de sa clé ou des constituants la composant.

Notation

Dans l'expression d'un schéma de relation le nom du type T précède le nom de la relation

T $R(A_1 \ldots A_n)$ désigne une relation R de type T.

Utilisation

Les types de relation ont tout d'abord un rôle sémantique, ils permettent d'exprimer l'appartenance d'une classe de phénomènes représentée à une catégorie.

Les contraintes qui leur sont associées permettent de vérifier une certaine complètude et une certaine cohérence d'un schéma conceptuel.

Enfin ils peuvent servir au moment du choix de la structure interne du SI, dans la mesure où les différences de modes d'utilisation des relations de types différents justifient des choix d'implémentation physique différents.

I-4.- Les types de constituants

L'analyse des propriétés des phénomènes réels effectuée au chapitre précédent met en évidence trois types de constituants :

Le type DATE, le type PROPRIETE, le type TEXTE.

Définition des types

- Le type PROPRIETE est l'ensemble des constituants représentant des propriétés non temporelles d'objets, d'opérations et d'événements dont les domaines sont des ensembles de valeurs numériques, alphanumériques ou alphabétiques.

Exemples : nocde, qtécdée, nomproduit.

- Le type DATE est l'ensemble des constituants représentant des propriétés temporelles de phénomènes c'est-à-dire des dates. Les domaines de ces constituants sont des sous-ensembles finis d'un même ensemble appelé ensemble TEMPS.

Un élément de l'ensemble TEMPS représente un instant de la vie de l'organisation. La valeur d'un élément de l'ensemble fait référence à un calendrier et se présente comme une concaténation de valeurs appartenant elles-mêmes chacune à un domaine particulier.

TEMPS = ANNEE, MOIS, JOUR, HEURE, etc ...
si on utilise le calendrier grégorien.

Exemples : date-arrivée-cde, date-exécution-livraison.

- Le type TEXTE est l'ensemble des constituants représentant certaines propriétés des OPERATIONS et des EVENEMENTS dont le domaine est l'ensemble des phrases engendrables par un langage particulier.

Ce sont les propriétés "texte" des opérations, les propriétés "prédicats" des évènements et les propriétés "condition" et "facteur" de déclenchement des associations DECLENCHE.

Exemples : textelivraison, prédicat-rupt, cond-réappro.

Ces types, comme les types de relation n'ont pas uniquement un rôle sémantique, ils sont utiles pour choisir le mode de représentation interne des constituants et pour définir les types d'opérations que l'on peut exécuter sur eux lors de l'utilisation du SI.

II - NORMALISATION ET REPRESENTATION DE LA CATEGORIE OBJET

II-1.- Objectifs de la représentation de la catégorie OBJET

II-1.1.- Représentation historique

Rappelons qu'une des finalités que nous avons assignées au SI dans le chapitre 1, est de restituer aux gestionnaires tout ou partie des états par lesquels est passée l'organisation au cours du temps.

Pour que le SI satisfasse à ce besoin, nous avons choisi de représenter ce que nous avons appelé, au chapitre 2, un déroulement complet, c'est-à-dire la suite des états pris par l'organisation au cours du temps.

II-1.2.- Représentation économique

La représentation d'un tel déroulement étant très volumineuse, nous nous sommes efforcés de l'effectuer de la manière la plus économique possible, c'est-à-dire en éliminant au maximum la redondance dans la représentation des états.

II-2.- Mode d'expression retenu : relation permanente en forme normalisée

Pour satisfaire les deux objectifs, nous avons choisi comme mode de représentation des phénomènes de la catégorie OBJET la relation permanente en forme normalisée pour les raisons suivantes :

- la relation permanente permet une représentation historique complète puisqu'elle ne peut admettre comme modification de ses extensions que l'adjonction de n-uplets.
- la relation permanente en forme normalisée présente les avantages de toute relation en forme normalisée et donc en particulier élimine certaines redondances dans la représentation.

II-3.- Méthode de représentation de la catégorie OBJET - Concepts de C-OBJET et de C-CLASSE.

II-3.1.- Points de départ et d'arrivée de la méthode : schéma descriptif et schéma conceptuel de données

La méthode retenue pour représenter la catégorie OBJET consiste à passer :

- d'un schéma descriptif de données exprimé sous la forme d'une collection de classes d'objets et de classes d'associations d'objets telles que nous les avons définies au chapitre 2.
- à un schéma conceptuel de données exprimé sous la forme d'une collection de schémas de relations permanentes en forme normalisée.

II-3.2.- Etapes de la méthode : représentation instantanée, représentation dans le temps

Le passage d'un schéma à l'autre est effectué en deux étapes :

- Etape n° 1 : représentation instantanée de la catégorie OBJET

Cette étape correspond à la recherche classique d'un schéma conceptuel de base de données.

Elle conduit à une collection de schémas de relations dépendant du temps en forme normalisée (la forme finale obtenue dépend de la méthode utilisée pour déterminer les schémas de relations).

Chaque schéma de relation est interprété alors comme la représentation d'une classe d'objets ou d'associations d'objets, n'apparaissant pas en général sous la même forme dans le schéma descriptif.

Nous appelons C-CLASSES (la lettre C signifiant conceptuel) ces nouvelles classes.

- Etape n° 2 : représentation dans le temps

La deuxième étape prend en compte les caractéristiques temporelles des C-CLASSES.

Elle conduit à une collection de schémas de relations permanentes en forme normalisée représentant chacune un aspect temporel d'une C-CLASSE. Nous appelons C-OBJET chacun de ces aspects temporels.

Détaillons ces deux étapes.

II-3.3.- Représentation instantanée de la catégorie OBJET

Dans cette étape nous passons du schéma descriptif de données à une collection de C-CLASSES en deux temps :

1) Passage du schéma descriptif à un unique schéma de relation RO.

2) Normalisation de RO.

II-3.3.1.- Construction du schéma de relation RO

Rappelons que, dans le chapitre 2, nous avons défini les classes de la catégorie OBJET de la façon suivante :

Définition d'une classe d'objets

Une classe d'objets est définie par :
- un nom de classe NCO
- une collection de propriétés PCO c'est-à-dire d'applications de profil P_k : NCO $\longrightarrow\!\!\!\!\!\rightarrow$ V_k V_k : domaine de la propriété P_k.
- un ensemble d'axiomes comprenant au moins l'axiome identifiant : $\exists P_I \in$ PCO tel que nco $\neq$ nco' $\Rightarrow$ $P_I(nco) \neq P_I(nco')$
  P_I est appelé la propriété identifiant de la classe.

Définition d'une classe d'associations d'objets

Une classe d'associations est définie par :
- une relation $r : NCO_1 \times NCO_2 \ldots NCO_p$ où NCO_i est un nom de classe d'objets
- une collection de propriétés P_k de profil $NCO_1 \times NCO_1 \ldots NCO_p \longrightarrow\!\!\!\!\!\rightarrow V_k$.

- un ensemble d'axiomes précisant la nature de la relation r.

Définition du schéma de relation RO

Soit RO(A),D,Γ le schéma suivant

- A : ensemble des constituants
  - chaque propriété de nom différent de chaque classe est représentée par un constituant A_k de A.
- D : ensemble des domaines des constituants
  - A chaque constituant A_k représentant une propriété P_k : $NCO_i \longrightarrow\!\!\!\!\!\rightarrow V_k$ est associé le domaine V_k.
- Γ : ensemble des contraintes d'intégrité

1) Pour chaque propriété P_k, appartenant à une classe NOC_ℓ et représentée par le constituant A_k, Γ contient une dépendance fonctionnelle $I_\ell \longrightarrow\!\!\!\!\!\rightarrow A_k$ où I_ℓ est le constituant représentant la propriété identifiant de la classe NCO_ℓ.

2) Pour les classes d'associations, suivant les axiomes associés à la relation r
   Γ contient
   - soit une ou plusieurs dépendances fonctionnelles entre les constituants représentant les identifiants des classes NCO_ℓ intervenant dans r.
   - soit des dépendances fonctionnelles de type $I_1 \times I_2 \ldots \times I_p \rightarrow\!\!\!\!\!\rightarrow A_i$ où A_i est un constituant représentant une propriété de r.

3) Les autres axiomes donnent lieu à des contraintes d'intégrité entre constituants non exprimées par des dépendances fonctionnelles.
   Nous ne nous intéressons pas à ces contraintes puisque nous avons choisi une méthode de normalisation n'utilisant que les DF.

II-3.3.2.- Normalisation de RO et définition des C-CLASSES

La normalisation de RO par la méthode synthétique de Bernstein, exposée dans le paragraphe I conduit à une collection de schémas de relation : $RO_1, RO_2 \ldots RO_n$ en troisième forme normale.

Chaque schéma RO_i représente une C-CLASSE
- un schéma dont la clé est un constituant simple représente une C-CLASSE d'objets.
- un schéma dont la clé est un constituant composé représente une C-CLASSE d'associations d'objets.

Dans la suite nous parlerons de C-CLASSE sans faire de distinction entre C-CLASSES d'objets et C-CLASSES d'associations d'objets.

II-3.4.- Représentation dans le temps

Cette étape se déroule également en deux phases :

1) Représentation de chaque C-CLASSE par une relation permanente.

2) Normalisation de chacune de ces relations permanentes.

II-3.4.1.- Représentation de chaque C-CLASSE par une relation permanente

Définition temporelle d'une C-CLASSE

- Soit une C-CLASSE de schéma RO_i $(I, A_1, A_2 \ldots A_n)$ représentant une classe réelle CO_i définie par

les propriétés P_I : $CO_i \longrightarrow\!\!\!\!\!\rightarrow V_I$ V_I domaine de I propriété identifiant

pour $k = 1,n$ P_k : $CO_i \longrightarrow\!\!\!\!\!\rightarrow VA_k$ VA_k domaine de A_k

- L'introduction du temps dans la définition de CO_i revient à en donner une nouvelle définition intégrant ce que nous avons appelé les propriétés temporelles des objets dans le chapitre 2.

- Compte tenu de l'hypothèse concernant la durée de vie des objets ou des associations d'objets que nous considérons limitée à un intervalle de temps la collection de propriétés de CO_i devient :

- Propriété date de création : $P_{DC} : CO_i \rightarrowtail T$ T domaine de valeurs des propriétés temporelles.
- Propriété date de suppression : $P_{DS} : CO_i \rightarrowtail T$
- Propriété identifiant : $P_I : CO_i \rightarrowtail D_I$ inchangée puisque, par définition, un identifiant garde une valeur constante au cours du temps.
- Une collection de propriétés constantes PC : $P_k : CO_i \rightarrowtail V_k$ pour toutes les propriétés de la classe qui gardent, pour un objet donné, une valeur constante au cours du temps et connue lors de la création de l'objet.
- Une collection de propriétés variables PV : $P'_k : CO_i \times T \rightarrowtail V_k$ pour toutes les propriétés de la classe qui, pour un objet donné, prennent des valeurs différentes au cours du temps.
  P'_k est définie pour $T \in DMAk$: ensemble des dates de changement de valeurs de la propriété.

Représentation par un schéma de relation permanente ROP

ROP(A),D,Γ est défini comme précédemment.

- A : $A_1 \ldots A_n, DC, DS, DM_i, \ldots DM_{i+\ell}$
  - DC est le constituant correspondant à la propriété date de création
  - DS est le constituant correspondant à la propriété date de suppression
  - DM_j est le constituant correspondant à T dans la propriété variable $CO_i \times T \rightarrowtail V_j$
- D : $V_I, V_1, \ldots, V_n, T$
- Γ : ensemble de dépendances fonctionnelles permanentes comprenant

  $I \dot{\rightarrow} DC$ $\quad I,DS \dot{\rightarrow} \theta$ $\quad \theta$ constituant fictif

  $I \dot{\rightarrow} A_k$ pour tout A_k correspondant à une propriété appartenant à PC

$I,DMA_j \dot{\rightarrow} A_j$ pour tout A_j correspondant à une propriété appartenant à PV.

Remarque : Il est possible que plusieurs propriétés variables d'une classe aient le même comportement dynamique c'est-à-dire qu'elles soient toujours modifiées simultanément, alors la définition temporelle de la C-CLASSE comprendra un axiome indiquant par exemple que DMA_k et DMA_ℓ ont les mêmes domaines de valeur et sont synonymes.

Cet axiome se traduira dans ROP par le remplacement de DMA_k et DMA_ℓ par un constituant unique $DMA_{k\ell}$ dans A et le remplacement dans Γ des DFP $I,DMA_k \dot{\rightarrow} A_k$ et $I,DMA_\ell \dot{\rightarrow} A_\ell$ par $I,DMA_{k\ell} \dot{\rightarrow} A_k$ et $I,DMA_{k\ell} \dot{\rightarrow} A_\ell$.

II-3.4.2.- Normalisation des relations permanentes associées aux C-CLASSES

Les relations permanentes obtenues par le processus ci-dessus ne sont pas en troisième forme normale.

Pour chacune d'elles nous appliquons la méthode synthétique de normalisation ce qui conduit à les remplacer chacune par une collection de relations permanentes en 3FN.

Soit avec les notations précédentes

ROP(A),D,Γ, schéma à normaliser,

est remplacé par $ROP_1, ROP_2, \ldots ROP_n$

avec $ROP_1(\underline{I},DC,A_1 \ldots A_\ell)$

$A_1 \ldots A_\ell$ ensemble des constituants A_k de A tels que dans Γ on ait $I \dot{\rightarrow} A_k$

$ROP_n(\underline{I,DS})$

$i = 2,n-1$, $ROP_i(\underline{I,DMA_i},A_{i_1},\ldots A_{i_k})$

$A_{i_1} \ldots A_{i_k}$ ensemble des constituants A_j de A tels que dans Γ on ait $I,DMA_i \dot{\rightarrow} A_j$

Chaque ROP_i ainsi obtenu est appelé un C-OBJET.

II-4.- Type de C-OBJETS et sémantique associée

Sémantique d'un C-OBJET

La méthode appliquée pour déterminer les C-OBJETS entraîne qu'un C-OBJET regroupe dans son schéma un sous-ensemble de propriétés d'une C-CLASSE ayant le même comportement dynamique.

Ce sous-ensemble est maximal, d'après l'algorithme appliqué, c'est-à-dire que toutes les propriétés d'une C-CLASSE ayant le même comportement dynamique appartiennent au même C-OBJET.

Appelons aspect temporel d'une C-CLASSE un tel ensemble de propriétés.

> Un C-OBJET est la représentation d'un aspect temporel d'une C-CLASSE

Types de C-OBJET

La composition des clès des C-OBJETS obtenus par normalisation de la relation permanente associée à une C-CLASSE au paragraphe précédent permet de distinguer trois types de C-OBJET :

- Type C-OBJET PERMANENT

  Un C-OBJET dont la clé est de type propriété.

- Type C-OBJET VARIABLE

  Un C-OBJET dont la clé est de type propriété - date c'est-à-dire composée d'un ou plusieurs constituants de type propriété et d'un constituant de type DATE.

- Type C-OBJET SUPPRESSION

  Un C-OBJET dont la clé est de type propriété - date, le constituant date représentant alors la propriété date de suppression de la C-CLASSE.

Exemple et sémantique des types

- Soit la C-CLASSE AGENT d'une organisation

Cette C-CLASSE est représentée dans une perspective instantanée par la relation dépendant du temps en 3FN AGENT 1 de schéma

AGENT 1(noagent,nomagent,noinsee,situation-familiale,noservice, indice)

Dans une perspective de représentation historique elle est représentée par la relation permanente AGENT 2 de schéma

AGENT 2(noagent,nomagent,noinsee,situation-familiale,noservice, indice,date-embauche,datechgtsitfami,datechgtservice, datechgindice,datedépart)

auquel sont associées les DFP

noagent $\dot{\rightarrow}$ nomagent,noinsee,date-embauche
noagent,datechgtservice $\dot{\rightarrow}$ noservice
noagent,datechgtsitfami $\dot{\rightarrow}$ situation familiale
noagent,datechgtindice $\dot{\rightarrow}$ indice
noagent,datedépart $\dot{\rightarrow}$ θ

La normalisation de AGENT 2 conduit aux C-OBJETS

AGENT 3, AGENT 4, AGENT 5, AGENT 6, AGENT 7 de schémas

AGENT 3(noagent,nomagent,noinsee,dateembauche)
AGENT 4(noagent,datechgtservice,noservice)
AGENT 5(noagent,datechgtsitfami,situationfamiliale)
AGENT 6(noagent,datechgtindice,indice)
AGENT 7(noagent,datedépart)

Le C-OBJET AGENT 3 est de type permanent il représente l'aspect permanent de la C-CLASSE.

Son extension contient à une date donnée t un n-uplet pour chaque agent ayant été embauché dans l'organisation à une date comprise entre le début de l'observation de l'organisation et t.

Les C-OBJETS AGENT 3, AGENT 4, AGENT 5, AGENT 6 sont de type variable, ils représentent un aspect variable de la C-CLASSE.

Par exemple, le C-OBJET AGENT 5 représente l'aspect situation de famille.

Son extension contient à un instant donné t, pour chaque agent autant de n-uplets qu'il a eu de situations familiales différentes depuis sa date d'embauche jusqu'à t.

Le C-OBJET AGENT 7 est de type SUPPRESSION, il représente l'aspect départ de la C-CLASSE

Son extension contient à une date donnée t autant de n-uplets que d'agents ayant quitté l'organisation à une date située entre la date de début d'observation de l'organisation et t.

II-5.- Expression du schéma conceptuel de données

Nous nous contentons dans ce paragraphe de donner une forme d'expression très simple d'un schéma conceptuel de données à l'aide des concepts de C-CLASSES et de C-OBJETS en utilisant la notation classique des schémas de relation.

Nous donnons dans le chapitre 4 un mode de représentation graphique de ce schéma et discutons du type de langage nécessaire pour l'exprimer complètement, c'est-à-dire permettre également l'expression des contraintes de domaine et des autres contraintes d'intégrité.

Le schéma conceptuel de données se présente comme une collection de schémas de relations de C-OBJETS de différents types regroupés en C-CLASSES.

Chaque C-CLASSE est définie de la façon suivante :

C-CLASSE nom de C-CLASSE : nom du constituant identifiant de la C-CLASSE [Liste de noms des C-OBJETS de la C-CLASSE]

Chaque C-OBJET est défini de la façon suivante :

C-OBJET {PERM / VAR / SUP} nom de C-OBJET (liste des noms des constituants du C-OBJET)

Les définitions des C-OBJETS d'une C-CLASSE suivent celles de la C-CLASSE, l'ensemble des constituants du C-OBJET correspondant à sa clé est souligné.

Exemple : Reprenons la C-CLASSE AGENT précédente. Sa définition dans le schéma conceptuel de données du SI est :

C-CLASSE agent : noagent(agent3,agent4,agent5,agent6,agent7)

C-OBJET PERM agent 3(noagent,noinsee,nomagent,dateembauche)
C-OBJET VAR agent 4(noagent,datechgtservice,service)
C-OBJET VAR agent 5(noagent,datechgtsitfami,situationfamiliale)
C-OBJET VAR agent 6(noagent,datechgtindice,indice)
C-OBJET SUP agent 7(noagent,datedépart)

II-6.- Représentation d'un état et d'un changement d'état dans un SI construit à l'aide du modèle conceptuel

Rappelons qu'au chapitre 2, dans le paragraphe donnant la formalisation du modèle descriptif, nous avons défini l'état de l'organisation à un instant donné comme une application $s : O \rightarrowtail V$ où O est l'ensemble des objets et associations d'objets et V un ensemble de valeurs.

Nous nous proposons ici de définir cette application s dans un SI construit à l'aide du modèle conceptuel de données exposé ci-dessus.

Ce SI est, à un instant donné, composé de l'ensemble des extensions des C-OBJETS définis dans son schéma conceptuel de données.

Soit COB l'ensemble des C-OBJETS du SI.

Représentation de O

O est représenté par l'ensemble des n-uplets des C-OBJETS de type PERMANENT.

Représentation de $o \in O$

o est représenté par un n-uplet d'un C-OBJET de type PERMANENT et est identifié par la projection de ce n-uplet sur la clé de ce C-OBJET.

Classe d'un objet $o \in O$

La C-CLASSE à laquelle appartient le C-OBJET contenant la re-

présentation de o représente la classe de o.

Etat d'un objet o, définition de s(o)

Soit s un état dont la date d'observation est t.
Soit un objet o appartenant à la C-CLASSE suivante

CO IOB[ROP,RO_1,...RO_n,ROS]

avec ROP de type permanent ROP(IOB,DC,A_1...A_n)
RO_1...RO_n de type variable RO_i(IOB,DM_i,A_i^1...A_i^p)
ROS de type suppression ROS(IOB,DS)

Soit iob la valeur de sa propriété identifiant iob ∈ ROP[IOB] alors

s(o) = si(ROP[iob,DC]>t ou (ROP[iob,DC]>t et ROS[iob,DS]>t))

alors u
"l'objet o n'a pas encore été créé à t ou bien a déjà été supprimé".
sinon (iob,$s^1(o)$,...$s^k(o)$,...$s^n(o)$)

- $s^k(o)$ valeur de la propriété P_k, représentée par le constituant A_k, à l'instant t.

- si A_k appartient au schéma ROP
alors $s^k(o)$ = ROP[iob,A_k]

- si A_k appartient au schéma RO_i
alors $s^k(o)$ = RO_i[iob,t_i,A_k]

où t_i est tel que $(t_i-t) = \min(t_j-t)$
$t_j \in RO_i[iob,DM_i]$ et $t_j < t$.

Représentation d'un déroulement

La collection des extensions de tous les C-OBJETS permet de reconstituer des suites d'états par lesquels est passée l'organisation au cours du temps, c'est-à-dire ce que nous avons appelé au chapitre 2 des déroulements.

En effet, s'il on se donne un ensemble de dates d'observations $t_1,t_2...t_n$, il est possible de déduire des extensions des C-OBJETS les états $s_1,s_2...s_n$ associés à ces dates (s_i état à la date t_i) en appliquant la formule précédente.

Représentation d'un changement d'état

Soient deux états s et s' successifs d'un déroulement, de dates d'observation t et t' t' > t,
le changement d'état (s,s') est représenté dans le SI par

$$CH = \bigcup_{ROB \in COB} (ROB_{t'} - ROB_t)$$

c'est-à-dire par tous les n-uplets de C-OBJETS qui ont été ajoutés au SI entre t et t'.

Changement d'état élémentaire

(s,s') est élémentaire si Card(CH) = 1 c'est-à-dire si les deux états ne diffèrent que par la valeur d'un unique aspect d'un unique objet.

Conséquence : Un changement d'état élémentaire est représenté par l'adjonction d'un nuplet dans un C-OBJET.

Remarque : Il s'agit bien d'un changement d'état élémentaire dans le sens d'indécomposable en un ensemble d'autres changements d'état, puisque un C-OBJET regroupe un ensemble de propriétés d'une classe d'objets ayant un même comportement dynamique.

Conclusion

La représentation des phénomènes de la catégorie OBJET est effectuée dans le modèle conceptuel à l'aide des concepts de C-CLASSE et de C-OBJET.

Une C-CLASSE est un ensemble de C-OBJETS, représentant une

classe d'objets ou d'associations d'objets, dans son évolution au cours du temps.

Un C-OBJET est une relation permanente en troisième forme normale représentant un aspect temporel d'une classe d'objets ou d'associations d'objets.

La partie donnée d'un SI construit à l'aide de ces concepts représente la succession des états pris par l'organisation au cours du temps.

L'adjonction d'un n-upletdans un C-OBJET correspond à un changement d'état élémentaire survenu dans l'organisation.

Cette élémentarité des C-OBJETS sert de point de départ pour la normalisation de la représentation des phénomènes des autres catégories.

III - NORMALISATION ET REPRESENTATION DES PHENOMENES DE LA CATEGORIE OPERATION

III-1.- Critères de normalisation des catégories OPERATION et EVENEMENT

Comme nous l'avons remarqué dans l'introduction, les structurations en classes des trois catégories de phénomènes sont dépendantes les unes des autres.

C'est pourquoi la définition des concepts permettant la représentation des phénomènes des catégories OPERATION et EVENEMENT est dépendante de ceux choisis pour représenter la catégorie OBJET.

Les critères appliqués pour choisir ces concepts sont les suivants :

- Permettre une représentation historique de la catégorie

Le SI doit conserver une trace de toute opération exécutée dans l'organisation et de tout événement qui y est survenu afin de pouvoir répondre à toute question concernant les règles de gestion appliquées dans l'organisation à une date donnée.

- Conserver l'élémentarité de représentation adoptée pour les objets

Nous avons montré que le concept de C-OBJET permet de définir l'évolution de l'organisation au cours du temps en termes de changements d'état élémentaires. Dans un souci d'uniformité des concepts du modèle, nous avons choisi de représenter également les phénomènes des catégories OPERATION et EVENEMENT en termes de classes de phénomènes élémentaires.

III-2.- Normalisation et représentation des classes d'opérations Concept de C-OPERATION

III-2.1.- Opération élémentaire

Définition

Une opération élémentaire est une opération qui a pour effet un changement d'état élémentaire dans l'organisation.

C'est donc une opération qui modifie un seul objet sous un unique aspect.

Représentation de l'effet d'une opération élémentaire dans le SI

Dans le SI l'exécution d'une opération élémentaire est représentée par l'adjonction d'un nuplet dans un C-OBJET.

III-2.2.- Classe d'opérations élémentaires C-OPERATION

Une classe d'opérations élémentaires est constituée de l'ensemble des opérations élémentaires effectuées dans l'organisation correspondant à l'utilisation d'un même texte.

Lien entre C-OPERATION et C-OBJET

Dans le SI toutes les occurrences d'une C-OPERATION ont pour effet d'ajouter un nuplet à une même relation de type C-OBJET et à une seule.

Toute C-OPERATION est donc liée à un seul C-OBJET appelé le C-OBJET "modifié" par la C-OPERATION.

Application CMOD

Soit COP l'ensemble des C-OPERATIONS
Soit COB l'ensemble des C-OBJETS
} d'un SI

Il existe entre COP et COB une application

$$CMOD : COP \rightarrowtail\!\!\!\!\!\!\rightarrow COB$$

qui, à toute C-OPERATION, associe un C-OBJET : le C-OBJET "modifié" par la C-OPERATION.

III-3.- Représentation des C-OPERATIONS

Définition d'une C-OPERATION

Une C-OPERATION est un ensemble OP muni d'un certain nombre de propriétés que nous définissons comme pour les objets par des applications :

Toute C-OPERATION possède les propriétés suivantes :

- propriété identifiant IOP : $OP \rightarrowtail\!\!\!\!\!\!\rightarrow N$
  propriété qui à toute opération d'une C-OPERATION associe un numéro d'ordre qui permet de la distinguer des autres.
- propriété date de fin d'exécution DEX : $OP \rightarrowtail\!\!\!\!\!\!\rightarrow T$
  propriété qui situe dans le temps une opération de la classe. Nous avons choisi ici de la situer par sa date de fin d'exécution, sa date de début d'exécution étant indiquée dans la représentation de l'association déclenche contenant la référence de l'événement qui a été à l'origine de son exécution.
- propriété objet modifié OM : $OP \rightarrowtail\!\!\!\!\!\!\rightarrow CMOD(OP)$
  CMOD(OP) désigne le C-OBJET "modifié" par la C-OPERATION.
  propriété qui, à toute opération, associe le n-uplet de C-OBJET qu'elle a créé. Cette propriété représente en fait la classe d'association MODIFIE introduite au chapitre 2 entre les éléments de OP et CMOD(OP).

Chaque C-OPERATION peut posséder en plus de ces trois propriétés des propriétés particulières de type $P_k : OP \rightarrowtail\!\!\!\!\!\!\rightarrow V_k$.

Représentation d'une C-OPERATION par une relation

En appliquant la même méthode que pour la représentation des objets, nous représentons chaque C-OPERATION par une relation de schéma

C-OP ROP(IOP, DEX, IOB, $A_1 \ldots A_n$)

$D : \{N, T, V_{IOB}, V_1 \ldots V_n\}$

Γ : $IOP \rightarrowtail\!\!\!\!\!\!\rightarrow DEX$

$$IOP \rightarrowtail IOB$$
$$k=1,n \quad IOP \rightarrowtail A_k$$

où IOB est le constituant clé du C-OBJET CMOD(OP) de domaine V_{IOB}
IOP est le constituant correspondant à la propriété identifiant
DEX est le constituant correspondant à la date de fin d'exécution
$A_1 \ldots A_n$ sont les constituants correspondant aux propriétés $P_1 \ldots P_n$.
C-OP indique que le type de la relation est C-OPERATION.
Cette relation est :

- permanente : une opération d'une classe ne peut plus être modifiée une fois qu'elle a été exécutée, les n-uplets de ROP sont donc constants au cours du temps ; de plus le choix de représentation de l'évolution complète de l'organisation au cours du temps, interdit de supprimer des n-uplets.
- en troisième forme normale : par construction.

Conclusion : Dans le modèle conceptuel, une C-OPERATION est une relation permanente en troisième forme normale qui représente une classe d'opérations élémentaires.

IV - NORMALISATION ET REPRESENTATION DES PHENOMENES DE LA CATEGORIE EVENEMENT

IV-1.- Normalisation des classes d'événements : Concept de C-EVENEMENT

IV-1.1.- Evénement élémentaire

Un événement élémentaire est un événement qui correspond à un changement d'état élémentaire.

Représentation d'un événement élémentaire dans le SI

Dans le SI un événement élémentaire correspond à l'adjonction d'un n-uplet dans un C-OBJET.

IV-1.2.- Classe d'événements élémentaires

Définition

Une classe d'événements élémentaires est un ensemble d'événements élémentaires constatant des adjonctions de même type de nuplets dans un même C-OBJET : On appelle C-EVENEMENT une telle classe.

Lien entre C-EVENEMENT et C-OBJET - Application CCONST

A tout C-EVENEMENT est associé un C-OBJET, le C-OBJET dont l'adjonction de certains nuplets ont été des événements de la classe représentée par le C-EVENEMENT.

On appelle ce C-OBJET le C-OBJET "constaté" par le C-EVENEMENT.

Application CCONST

Soit CEV l'ensemble des C-EVENEMENTS d'un SI
Soit COB l'ensemble des C-OBJETS

$$CCONST : CEV \rightarrowtail COB$$

CCONST associe à tout C-EVENEMENT un C-OBJET : le C-OBJET "constaté" par le C-EVENEMENT.

IV-2.- Représentation des C-EVENEMENTS

Définition d'un C-EVENEMENT

Un C-EVENEMENT est un ensemble CE défini par les propriétés suivantes :

Propriétés communes à tous les C-EVENEMENTS

- propriété identifiant IEV : $CE \rightarrowtail N$
  associe à tout événement de CE un numéro d'ordre qui permet de le distinguer des autres.
- propriété date d'arrivée DAR : $CE \rightarrow\!\!\!\!\!\rightarrowtail T$
  associe à tout événement la date à laquelle il a été constaté dans l'organisation.

- propriété objet constaté OC : CE ↣ CCONST(CE)
  associe à tout événement le n-upletde C-OBJET dont l'adjonction dans le SI a été à l'origine.
  Cette propriété représente la classe d'associations CONSTATE entre CE et des C-OBJETS.

Propriétés particulières : P_k : CE ↣ V_k k = 1,n.

Représentation par une relation

En appliquant toujours la même méthode, nous représentons chaque C-EVENEMENT par une relation de schéma :

C-EV REV(IEV,DAR,IOB,$A_1 \ldots A_n$)

D = N,T,V_{IOB},$V_1 \ldots V_n$

Γ : IEV ↣ DAR
IEV ↣ IOB
k=1,n IEV ↣ A_k

où IEV est le constituant correspondant à la propriété identifiant
DAR est le constituant correspondant à la propriété date d'arrivée
IOB clé du C-OBJET CCONST(CE), correspond à la propriété objet constaté.
k=1,n A_k est le constituant correspondant à la propriété P_k.
C-EV indique que le type de la relation est C-EVENEMENT.

Cette relation est :

- permanente : Un événement d'une classe ne peut plus être modifié une fois qu'il est passé, d'autre part,tout événement survenant dans l'organisation doit être conservé dans le SI. Donc la seule modification que peut subir REV au cours du temps est l'adjonction de nuplets.
- en troisième forme normale : par construction.

Conclusion : Dans le modèle conceptuel un C-EVENEMENT est une relation permanente en troisième forme normale qui représente une classe d'événements élémentaires.

IV-3.- Représentation des classes d'associations "DECLENCHE"
Concept de C-DECLENCHE

Les relations C-OPERATION contiennent la représentation des liens entre C-OPERATIONS et C-OBJETS c'est-à-dire des classes d'association "MODIFIE".

Les relations C-EVENEMENT contiennent la représentation des liens entre C-EVENEMENTS et C-OBJETS c'est-à-dire des classes d'association "CONSTATE".

Pour représenter complètement la structure intercatégorie, il faut représenter également les classes d'association DECLENCHE.

Définition d'une classe d'association DECLENCHE

Une classe d'association DECLENCHE est un ensemble d'associations entre des éléments d'un C-EVENEMENT CE et d'une C-OPERATION CP.

Cet ensemble est défini par les propriétés suivantes :

- propriété identifiant : CE×CP ↣ N×N
  associe à tout couple (ce,cp) la valeur (IEV(ce),IOP(cp)), IEV et IOP étant respectivement les propriétés identifiant de CE et CP.
- propriété date de déclenchement DEC : CE×CP ↣ T
  associe à tout couple (ce,cp) la date à laquelle "ce" a effectivement déclenché "cp".
  Cette date de déclenchement est aussi la date de début d'exécution de "cp".
- propriétés particulières à la classe : P_k : CE×CP ↣ V_k
  k = 1,n.

Représentation d'une classe d'association DECLENCHE par une relation

Toute classe d'association DECLENCHE est représentée par une relation de schéma

CDEC RDEC(IEV,IOP,DEC,A_1,A_2...A_n)

D : V_{IEV},V_{IOP},T,V_1,V_2...V_n

Γ : IEV,IOP $\rightarrowtail$ DEC

k=1,n IEV,IOP $\rightarrowtail A_k$

Cette relation est :

- permanente : une fois le déclenchement effectué et enregistré dans le SI, il n'est plus possible de le supprimer ni de le modifier.
- en troisième forme normale

CDEC indique que le type de la relation est : C-DECLENCHE.

Conclusion : Les phénomènes de la catégorie EVENEMENT sont représentés par deux types de relation : C-EVENEMENT et C-DECLENCHE.

Un C-EVENEMENT est une relation permanente en troisième forme normale représentant une classe d'événements élémentaires.

Un C-DECLENCHE est une relation permanente en troisième forme normale représentant une classe d'association DECLENCHE entre un C-EVENEMENT et une C-OPERATION.

V - FORMULATION DE L'APPLICATION DECL DANS LE MODELE CONCEPTUEL

Les choix effectués dans le modèle conceptuel :

- l'élémentarité des phénomènes et des classes représentées
- la conservation de l'historique des états de l'organisation

permettent de préciser et de simplifier :

- les applications entre ensembles de classes de phénomènes d'une même catégorie
- le profil des applications qui définissent la dynamique de l'organisation

et d'aboutir ainsi à une expression de l'application DECL plus simple que celle associée au modèle descriptif et donnée à la fin du chapitre 2.

V-1.- Applications entre ensembles de classes de phénomènes

Rappelons que nous avons défini entre les ensembles :

COB ensemble de classes d'objets
COP ensemble de classes d'opérations
CEV ensemble de classes d'événements
les applications MOD : COP $\rightarrowtail \mathcal{P}$(COB)
CONST : CEV $\rightarrowtail \mathcal{P}$(COB)
CDECL : CEV $\rightarrowtail \mathcal{P}$(COP)

Si les classes d'objets sont des C-OBJETS, les classes d'opérations des C-OPERATIONS et les classes d'événements, des C-EVENEMENTS alors l'élémentarité des classes implique que

$\forall$cop $\in$ COP CARD(MOD(cop)) = 1
$\forall$cev $\in$ CEV CARD(CONST(cev)) = 1

ce qui permet de remplacer les applications MOD et CONST par les applications CMOD et CCONST

CMOD : COP $\rightarrowtail$ COB
CCONST : CEV $\rightarrowtail$ COB

V-2.- Profil des textes des C-OPERATIONS

Rappelons que le profil général du texte d'une classe d'opérations cop est :

$text_{cop} : S \times P_{c_1} \times \ldots P_{c_n} \rightarrowtail F$

où $c_1 \ldots c_n$ sont des classes d'objets

$P_{c_i} \in \mathcal{P}(c_i)$ $\quad F : \{P_o \rightarrowtail V\}$ $\quad P_o \in \mathcal{P}(O)$

$S : \{O \rightarrowtail V\}$ $\quad O$ ensemble des objets

Si la classe d'opérations est une C-OPERATION alors :

- F peut être remplacée par cob = CMOD(cop)
- $S \times P_{c_1} \times \ldots P_{c_n}$ peut être remplacé par $P_{cob_1} \times \ldots P_{cob_n}$ où
  cob_i est un C-OBJET
  $P_{cob_i} \in \mathcal{P}(cob_i)$

et son profil devient

$P_{cob_1} \times \ldots P_{cob_n} \rightarrowtail cob$

Le texte d'une C-OPERATION est donc une application qui à un ensemble de n-uplets appartenant à certains C-OBJETS fait correspondre un n-uplet du C-OBJET qui lui est lié par l'application CMOD.

V-3.- Profil des prédicats des C-EVENEMENTS

Rappelons que le profil du prédicat associé à une classe d'événements "ce" est une application

$pred_{ce} : S \times S \rightarrowtail BOOL$ $\quad BOOL = \{vrai, faux\}$.

$pred_{ce}(s,s')$ = vrai signifie que le changement d'état (s,s') est un événement de la classe "ce".

Si "ce" est un C-EVENEMENT alors

si cob = CCONST(ce), un événement de "ce" correspond à un changement d'état représenté par l'adjonction d'un n-uplet dans cob et le profil de $pred_{ce}$ devient alors $pred_{ce} : cob \rightarrowtail BOOL$.

Le prédicat associé à un C-EVENEMENT "ce" est donc une application qui, pour tout n-uplet de C-OBJET "constaté" par le C-EVENEMENT, prend la valeur vrai ou faux et indique par là si l'adjonction de ce n-uplet est un événement appartenant à "ce".

V-4.- Profil des conditions de déclenchement

Rappelons que le profil général d'une condition de déclenchement d'une association DECLENCHE entre une classe d'événements "ce" et une classe d'opérations cop est :

$$cond_{ce,cop} : S \rightarrowtail BOOL$$

Cette condition porte sur l'état de l'organisation à l'instant où s'est produit un événement de la classe "ce".

Si "ce" est un C-EVENEMENT cet état est entièrement déterminé par la connaissance du n-uplet de C-OBJET qui correspond à l'événement.

Le profil de $cond_{ce,cop}$ devient donc : $cob \rightarrowtail BOOL$

avec cob = CCONST(ce)

V-5.- Profil des facteurs de déclenchement

Rappelons que le profil général d'un facteur de déclenchement d'une association DECLENCHE entre une classe d'événements "ce" et une classe d'opérations "op" est :

$$fac_{ce,op} : S \times S \rightarrowtail \mathcal{P}(P_{c_1} \times \ldots \times P_{c_q})$$

avec $text_{op} : S \times P_{c_1} \times \ldots P_{c_q} \rightarrowtail F$

Le facteur de déclenchement est une application qui à un changement d'état (s,s') appartenant à une certaine classe d'événements "ce", fait correspondre les paramètres de chaque opération de la classe "op" à exécuter en conséquence de (s,s').

Si "ce" est un C-EVENEMENT et "op" une C-OPERATION, alors $S \times S$ peut être remplacé par cob où cob = CCONST(ce) et $\mathcal{P}(P_{c_1} \times \ldots P_{c_q})$

est remplacé par la partie gauche du profil de "op".

Le profil de $fac_{ce,op}$ est donc

$$fac_{ce,op} : cob \rightarrowtail \mathcal{P}(P_{cob_1} \ldots P_{cob_n})$$

avec cob = CCONST(ce)

et $text_{op} : P_{cob_1} \ldots P_{cob_n} \rightarrowtail cob'$

cob' = CMOD(op)

Dans le cadre du modèle conceptuel, $fac_{ce,op}$ est une application qui, à un n-uplet de C-OBJET, fait correspondre un ensemble d'ensembles de n-uplets qui sont en fait les paramètres effectifs des opérations à exécuter à la suite de la création de ce n-uplet dans le SI.

V-6.- Formulation de DECL dans le modèle conceptuel

Soit une organisation représentée par

- COB l'ensemble de ses C-OBJETS
- COP l'ensemble de ses C-OPERATIONS
- CEV l'ensemble de ses C-EVENEMENTS

et les applications :

- CMOD : COP $\rightarrowtail$ COB
- CCONST : CEV $\rightarrowtail$ COB
- CDECL : CEV $\rightarrowtail \mathcal{P}$(COP)
- CTEXT : COP $\rightarrowtail$ A
- CPRED : CEV $\rightarrowtail$ B
- CCOND : CEV×COP $\rightarrowtail$ B
- CFAC : CEV×COP $\rightarrowtail$ C

A,B,C sont des ensembles d'applications.

A : ensemble d'applications de profil de type :

$$P_{cob_1} \ldots P_{cob_n} \rightarrowtail cob_{n+1}$$

où pour k = 1,n+1 $cob_k \in COB$, $P_{cob_k} \in \mathcal{P}(cob_k)$

B : ensemble des applications de profil de type :

$$cob \rightarrowtail BOOL \qquad BOOL = \{vrai, faux\}$$

où $cob \in COB$

plus les applications : CREATION : $\rightarrowtail$ vrai

et TOUJOURS : $\rightarrowtail$ vrai

qui rendent toujours la valeur vrai.

C : ensemble des applications de profil de type :

$$cob \rightarrowtail \mathcal{P}(P_{cob_1} \times \ldots P_{cob_n})$$

Soit $(s,s') = (ob_1 \ldots ob_n)$ un changement d'état de cette organisation

$\forall i = 1,n \quad ob_i \in RO_i \quad RO_i \in COB$

$$DECL(s,s') = \bigcup_{i=1,n} DECL(ob_i)$$

$$DECL(ob_i) = \cup\, text_{op_j^k}(fac_\ell)$$

$j : ce_j : CCONST(ce_j) = RO_i$ et $pred_{ce_j}(ob_i)$

$k : op_j^k \in CDECL(ce_j) : cond_{ce_j,op_j^k}(ob_i)$

$\ell : \{fac_\ell = fac_{ce_j,op_j^k(ob_i)}\}$

$pred_{ce_j} = CPRED(ce_j)$

$cond_{ce_j,op_j^k} = CCOND(ce_j,op_j^k)$

$fac_{ce_j,op_j^k} = CFAC(ce_j,op_j^k)$

$text_{op_j^k} = CTEXT(op_j^k)$

DECL devient donc au niveau du modèle conceptuel, une application qui, à un ensemble de n-uplets de C-OBJETS, fait correspondre un autre ensemble de n-uplets de C-OBJETS qui définissent le nouveau changement d'état engendré par le premier.

VI - REPRESENTATION DE L'APPLICATION DECL DANS LE MODELE CONCEPTUEL. METASTRUCTURE DE L'ORGANISATION

VI-1.- Définition de la métastructure de l'organisation

Le modèle conceptuel conduit à structurer l'organisation en quatre ensembles de classes de phénomènes : C-OBJETS, C-OPERATIONS, C-EVENEMENTS et C-DECLENCHES.

Le dynamique de l'organisation est alors définie par une collection d'applications sur ou entre ces ensembles que nous venons de détailler au paragraphe précédent.

D'autres applications ou relations peuvent être définies sur ou entre ces ensembles afin,en particulier,d'en préciser la sémantique.

Nous appelons METASTRUCTURE de l'organisation, l'ensemble de ces applications ou relations.

Pour que le SI soit une représentation complète de l'organisation, sous son double aspect statique et dynamique, il est nécessaire que cette métastructure y soit représentée.

Dans un souci d'uniformité, nous proposons d'effectuer cette représentation sous la forme d'une collection de relations permanentes en $3^{ème}$ forme normale.

Nous procèdons comme pour la catégorie OBJET,en donnant successivement

- une représentation instantanée de la métastructure
- une représentation historique permettant la prise en compte de l'évolution de la métastructure au cours du temps.

Enfin nous donnons l'expression de la fonction DECL à partir de la représentation des applications dans la métastructure.

VI-2.- Représentation instantanée de la métastructure

VI-2.1.- Schéma descriptif de la métastructure

Aux propriétés et associations définies par les applications introduites au paragraphe V, nous ajoutons pour chaque

ensemble de classes de phénomènes :

- une propriété "nom de relation" qui associe à chaque classe le nom de la relation du SI qui représente ses occurrences.
- une propriété "libellé explicatif" qui en précise le rôle et le sens.

Nous représentons aussi la structuration particulière de l'ensemble COB en C-CLASSES.

Définition de COB

COB est défini par les propriétés suivantes :

nom de relation	NCOB : COB → LN	LN :	ensemble des noms des relations du SI
libellé	LCOB : COB → L	L :	ensemble de chaînes de caractères
type	TYPEOB : COB → TO	TO :	PERM,VAR,SUP

Définition de COP

COP est défini par les propriétés suivantes :

nom de relation	NCOP : COP → LN		
libellé	LCOP : COP → L		
texte	CTEXT : COP → A	A :	Application définie précédemment.

Définition de CEV

CEV est défini par les propriétés suivantes :

nom de relation	NCEV : CEV → LN		
libellé	LCEV : CEV → L		
prédicat	CPRED : CEV → B	B :	Application définie précédemment.

Applications entre COB,COP,CEV

CMOD : COP ↠ COB
CCONST : CEV ↠ COB
CDECL : CEV ↠ $\mathcal{P}$(COP)

} définies précédemment

Définition de CDEC

CDEC : sous ensemble de CEV×COP défini par CDECL possède les propriétés suivantes :

nom de relation NCDEC : CDEC ↠ LN

condition de déclenchement CCOND : CDEC ↠ B
facteur de déclenchement CFAC : CDEC ↠ C

} définies précédemment

Multiplicité de déclenchement MUL : CDEC ↠ {UN,MULT}

indique si le déclenchement est simple ou multiple c'est-à-dire si un événement de CEV déclenche une ou plusieurs opérations de COP.

Définition de CC

CC ensemble des C-CLASSES est définie par :

nom de C-CLASSE NCC : CC ↠ L

identifiant de C-CLASSE ICC : CC ↠ NI

NI ensemble des noms des clés des relations de type C-OBJET perm.

libellé LCC : CC ↠ L

Application APAR

APAR : COB ↠ CC

APAR relie tout C-OBJET à une C-CLASSE.

VI-2.2.- Schéma conceptuel de la métastructure

L'application de la méthode de représentation et de normalisation d'un schéma descriptif exposée au paragraphe II-3.2.

conduit à représenter la métastructure par le schéma conceptuel suivant :

CC(NCC,ICC,LCC) représente l'ensemble des C-CLASSES

COB(NCOB,LCOB,TYPECOB,NCC) représente l'ensemble des C-OBJETS et l'application APAR

COP(NCOP,LCOP,CTEXT,NCOB) représente l'ensemble des C-OPERATIONS et l'application CMOD

CEV(NCEV,LCEV,CPRED,NCOB) représente l'ensemble des C-EVENEMENTS et l'application CCONST.

CDEC(NCOP,NCEV,NCDEC,CCOND,CFAC,MUL) représente l'ensemble des C-DECLENCHES et l'application CDEC.

Chaque constituant représente la propriété de même nom.

Cette collection de relations donne une représentation instantanée de la métastructure de l'organisation.

Or cette métastructure évolue au cours du temps :

- certaines propriétés de certaines classes ou associations peuvent prendre des valeurs différentes au cours du temps, en particulier les prédicats, les textes des opérations, les conditions ou les facteurs de déclenchement.

- des classes peuvent être supprimées ou ajoutées dans la métastructure.

- L'évolution de cette métastructure doit être représentée dans le SI afin de satisfaire aux objectifs que nous lui avons assignés en particulier :

- donner des renseignements sur les conditions de fonctionnement de l'organisation dans le passé ou sur sa structure à un instant donné.

Pour représenter cette évolution, nous utilisons la méthode

de représentation historique d'une réalité exposée à propos des phénomènes de la catégorie OBJET au paragraphe II-3.4.

VI-3.- Représentation historique de la métastructure

Pour effectuer cette représentation historique, nous avons fait les hypothèses suivantes :

- La durée vie de toute classe de phénomènes réels est un intervalle de temps
- les propriétés, nom de relation, libellé explicatif sont,pour toutes les classes,constantes au cours du temps,ainsi que la propriété TYPEOB.
- les propriétés prédicat, texteopération, condition de déclenchement et facteur de déclenchement sont variables au cours du temps.
- Toute modification dans la liste des constituants d'un C-OBJET, d'une C-OPERATION ou d'un C-EVENEMENT est interdite. Si un changement dans la liste des propriétés de la classe réelle représentée intervient alors il y aura :
  - création d'un nouveau schéma de relation du type considéré dans le SI.
  - indication dans la métastructure de la suppression du schéma précédent et de la création d'un nouveau.
  - prise en compte des autres modifications que peuvent entraîner le remplacement d'une classe de phénomènes par une autre, au niveau de la représentation.

En appliquant la méthode développée à propos des objets, on passe de la représentation instantanée précédente à la représentation historique exprimée sous la forme d'une collection de relations permanentes en 3FN.

Nous appelons: METACLASSE l'ensemble des relations obtenues par normalisation d'une relation de la représentation instantanée de la métastructure

METAOBJET une relation d'une METACLASSE.

Les METAOBJETS comme les C-OBJETS sont de trois types : PERM, VAR et SUP.

En utilisant une notation des schémas de relation identique à celle utilisée pour les objets, la représentation historique de la métastructure devient :

METACLASSE C-CLASSE NCC[CC1,CC2]

METAOBJET PERM CC1(NCC ,ICC,LCC,DCC)
"DCC date de création d'une C-CLASSE"
METAOBJET SUP CC2(NCC,DSC)
"DSC date de suppression d'une C-CLASSE"

"La métaclasse C-CLASSE représente l'ensemble des C-CLASSES ayant existé au cours du temps dans l'organisation"

METACLASSE C-OBJET NCOB[COB1,COB2]

METAOBJET PERM COB1(NCOB,LCOB,TYPEOB,NCC,DCO)
"DCO date de création d'un C-OBJET"
METAOBJET SUP COB2(NCOB,DSO)
"DSO date de suppression d'un C-OBJET"

" La métaclasse C-OBJET représente l'ensemble des C-OBJETS ayant existé au cours du temps dans l'organisation"

METACLASSE C-OPERATION NCOP[COP1,COP2,COP3]
METAOBJET PERM COP1(NCOP,LCOP,NCOB,DCP)
"DCP date de création d'une C-OPERATION"
METAOBJET VAR COP2(NCOP, DMT,CTEXT)
"DMT date de modification des textes des opérations"
METAOBJET SUP COP3(NCOP,DSP)
"DSP date de suppression d'une C-OPERATION"

"La métaclasse C-OPERATION représente l'ensemble des C-OPERATIONS appliquées au cours du temps dans l'organisation et, pour chacune d'elles, l'ensemble des textes utilisés successivement"

METACLASSE C-EVENEMENT NCEV[CEV1,CEV2,CEV3]

METAOBJET PERM CEV1(NCEV,NCOB,LCEV,DCE)
"DCE date de création d'un C-EVENEMENT"

METAOBJET VAR CEV2(NCEV,DMP,CPRED)
"DMP date de modification d'un prédicat des C-EVENEMENTS"

METAOBJET SUP CEV3(NCEV,DSE)
"DSE date de suppression d'un C-EVENEMENT"

"La métaclasse C-EVENEMENT représente l'ensemble des C-EVENEMENTS définis dans les règles de gestion de l'organisation au cours du temps, ainsi que les différentes expressions prises par leurs prédicats".

METACLASSE C-DECLENCHE NCOP,NCEV[CDEC1,CDEC2,CDEC3,CDEC4]

METAOBJET PERM CDEC1(NCOP,NCEV,NCDEC,DCD,MUL)
"DCD date de création d'un C-DECLENCHE"

METAOBJET VAR CDEC2(NCOP,NCEV,DMCOND,CCOND)
"DMCOND date de modification d'une condition de déclenchement"

Convention

Si tout événement d'un C-EVENEMENT ncev déclenche inconditionnellement une opération d'une C-OPERATION ncop alors

$\forall t \in$ CDEC2[DMCOND] CDEC2[ncev,ncop,t,CCOND] = 'TOUJOURS'

METAOBJET VAR CDEC3(NCOP,NCEV,DMFAC,CFAC)
"DMFAC date de modification d'un facteur de déclenchement"

METAOBJET SUP CDEC4(NCOP,NCEV,DSD)
"DSD date de suppression d'un C-DECLENCHE"

"La métaclasse C-DECLENCHE représente l'ensemble des associations DECLENCHE définies dans les règles de gestion de l'organisation au cours du temps, ainsi que l'évolution de leurs caractéristiques (condition de déclenchement et facteur de déclenchement)".

VI-4.- Expression de DECL à partir de la métastructure

Soit un C-OBJET RO1(IOB,DOB,$A_1 \ldots A_n$)

Soit un n-uplet o de ce c-objet = <iob,t,$a_1 \ldots a_n$>

L'adjonction de o dans RO1 est un évenement si

CARD(CE) $\neq \emptyset$ avec

CE : {ncev $\in$ CEV1[IEV,RO1]:$pred_{ncev}(0)$ et CEV3[ncev,DSE] = $\emptyset$}

avec $pred_{ncev}$ = CEV2[ncev,t_r,CPRED]

$t_r \in$ CEV2[ncev,DMP]

et tel que $(t_r - t) = \min(t_j - t)$ $t_j \in$ CEV[ncev,DMP] et $t_j < t$

"valeur la plus récente du prédicat associé au C-EVENEMENT de clé ncev"

Si o est un événement alors

$DECL(o) = \cup\ top_j(fac_i)$

$\forall$ncev $\in$ CE,

$\forall op_j \in OP_{ncev}$

$\forall fac_i \in FAC_{op_j,ncev}(o)$

où OP_{ncev} = {nop $\in$ CDEC1[NOP,ncev]

tels que CDEC4[nop,ncev,DSD] = $\emptyset$

et $COND_{nop,ncev}(o)$}

$COND_{nop,ncev}$ = CDEC2[nop,ncev,t_r,CCOND]

t_r définie comme ci-dessus sur CDEC2

$FAC_{iop,icev}$ = CDEC3[nop,ncev,t_r,FAC]

"valeurs les plus récentes des conditions et des facteurs de déclenchement".

Cette définition de DECL à partir de la métastructure montre la complètude de la représentation en ce qui concerne la dynamique de l'organisation.

Résumé et conclusion sur le modèle conceptuel

Le modèle conceptuel de SI proposé est l'ensemble des concepts exposés dans ce chapitre.

Le concept de base est la relation permanente en 3ème forme normale qui permet une représentation historique et minimale de tout phénomène réel.

A partir de ce concept,sont définis quatre types de relation permanente en 3FN : C-OBJET, C-OPERATION, C-EVENEMENT et C-DECLENCHE, qui permettent une représentation dans le temps des trois catégories de phénomènes réels en termes de classes de phénomènes élementaires.

Le concept de C-CLASSE, ensemble de C-OBJETS représentant des aspects différents d'une même classe réelle,permet de donner une représentation plus globale de la catégorie OBJET et de définir l'état d'un OBJET à un instant donné.

Les concepts de METACLASSE et METAOBJET permettent la représentation de la métastructure de l'organisation et en particulier la représentation complète et historique de ses lois de fonctionnement c'est-à-dire de sa dynamique.

La normalisation imposée par tous ces concepts à la représentation des phénomènes réels conduit à une formulation de l'application DECL, définissant cette dynamique, beaucoup plus simple que celle donnée au chapitre 2.

Nous allons voir dans le chapitre suivant qu'elle permet également de définir un ensemble de règles de validité d'un schéma conceptuel de SI exprimé à l'aide du modèle conceptuel et contribue donc ainsi à la définition de schéma conceptuel canonique c'est-à-dire à la réalisation de l'objectif que nous nous étions fixé au chapitre 1.

TROISIEME PARTIE

LA MODELISATION

CHAPITRE 4 : ANALYSE STRUCTURELLE ET RÈGLES DE VALIDITÉ D'UN SCHÉMA CONCEPTUEL DE SI

CHAPITRE 5 : DÉFINITION CONCEPTUELLE D'UN SI ET TYPE DE LANGAGE ASSOCIÉ

INTRODUCTION

Dans cette partie, nous étudions le résultat de la modèlisation à laquelle conduit l'utilisation du modèle conceptuel.

Dans le chapitre 4, nous effectuons une analyse de la structure d'un schéma conceptuel de SI et nous en proposons un mode de représentation graphique facilitant son interprétation. Nous donnons une liste de règles que doit vérifier tout schéma conceptuel de SI canonique.

Dans le chapitre 5, nous introduisons les contraintes d'intégrité à associer à un schéma conceptuel canonique pour obtenir une description conceptuelle complète du SI, suffisante pour réaliser le SI. Nous discutons du type de langage nécessaire pour exprimer cette description conceptuelle.

CHAPITRE 4 : ANALYSE STRUCTURELLE ET RÈGLES DE VALIDITÉ D'UN SCHÉMA CONCEPTUEL

INTRODUCTION

Dans ce chapitre, nous donnons tout d'abord un exemple de schéma conceptuel de SI exprimé à l'aide du modèle conceptuel présenté au chapitre précédent.

Cette expression se présente sous la forme d'une collection de schémas de relation et de relations dont la présentation linéaire permet difficilement d'appréhender la structure du système réel représenté.

C'est pourquoi nous proposons de l'illustrer par deux représentations graphiques de deux sous-schémas particuliers :

- le sous-schéma des données dont le rôle est de donner une vue globale de la catégorie OBJET du système représenté.
- le sous-schéma de la dynamique qui met en évidence les lois de fonctionnement du système sous leur forme causale.

Nous terminons en donnant un ensemble de règles de validité que doit respecter un SC canonique c'est-à-dire possédant les qualités de fidélité, complétude et cohérence que nous lui avons assignées au chapitre 1.

I - EXEMPLE DE SCHEMA CONCEPTUEL

L'exemple que nous utilisons est tiré de celui présenté par ABRIAL dans la présentation du langage Z (ABR 78). Nous l'avons complété par des précisions concernant la dynamique de la gestion des prêts. Nous traitons un exemple plus complexe et tiré d'une organisation réelle en annexe.

I-1.- Enoncé : gestion des prêts-client

Une banque prête de l'argent à ses clients, les modalités d'attribution des prêts et de remboursement qu'elle applique sont les suivantes :

Attribution des prêts

Un client ne peut avoir plus d'un prêt à la fois.

Les prêts ne sont accordés qu'aux clients ayant un revenu mensuel suffisant et n'ayant pas eu de retard de remboursement taxé lors d'un prêt précédent. Dans le cas où le prêt est refusé, le motif du refus est signalé au client, dans le cas contraire le prêt est accordé. Le taux dépend de la durée demandée.

Remboursement

Le remboursement est mensuel, le montant mensuel à rembourser est calculé à l'attribution du prêt, ou lors d'une révision du prêt à partir du taux d'intérêt, du montant du prêt et de sa durée. On en déduit le montant total réel à rembourser.

Une certaine liberté est laissée aux clients pour effectuer leurs versements.

Chaque mois, à une date de son choix, le client effectue éventuellement, un versement qui peut être supérieur ou inférieur à la mensualité qu'il doit.

Les clients peuvent donc prendre de l'avance ou du retard sur leur remboursement.

Le contrôle de la situation d'un prêt est effectué par la banque, chaque mois, le jour d'exigibilité du prêt, choisi par le client lors de la demande de prêt.

Gestion de l'avance sur remboursement, révision de prêt

Lorsqu'un client a remboursé au moins trois mensualités en avance, le prêt est révisé. C'est-à-dire qu'on recalcule le montant total à rembourser et le nombre de mensualités restantes et tout se passe ensuite comme pour un nouveau prêt.

Gestion du retard sur remboursement

Dès qu'un client a un retard de remboursement supérieur ou égal à trois mensualités, une taxe lui est appliquée, égale à un pourcentage de la somme due. Cette taxe est appliquée tous les mois tant que le retard reste supérieur à 3 mensualités.

Le paiement de la taxe est prélevé sur les versements.

Situation des clients

Un client ayant un prêt en cours en situation régulière est actif.

Un client ayant un prêt en cours et en retard de remboursement est suspendu.

Un client n'ayant pas encore de prêt ou plus de prêt en cours est non actif.

Solde des prêts

Un prêt est soldé lorsque le client a fini de le rembourser.

I-2.- Schéma conceptuel

Nous donnons ici une expression instantanée du schéma conceptuel du SI associé à l'exemple précédent.

Elle comprend :

- les relations de la métastructure qui décrivent les ensembles de C-OBJETS, C-OPERATIONS et C-EVENEMENTS du SI et les applications qui définissent sa dynamique.
- les schémas de relation permettant de représenter dans le SI, les occurrences de ces différentes classes de phénomènes.

Les valeurs des constituants de la métastructure de type TEXTE c'est-à-dire représentant des applications correspondant aux textes des opérations ou des conditions de déclenchement et des facteurs de déclenchement ne sont pas données ici.

Leur écriture nécessite l'utilisation d'un langage dont nous discuterons les caractéristiques dans le chapitre suivant.

Métastructure

Extension de la relation CC(NCC,ICC,LCC) décrivant les c-classes et leur sémantique.

NCC	ICC	LCC
client	noclient	client de la banque
demande	nodemande	demande de prêt
prêt	noprêt	prêt accordé par la banque
versement	noversement	versement pour remboursement de prêt
taxe	notaxe	taxe pour retard de remboursement
jour	nojour	jour écoulé

Extension de la relation COB(NCOB,LCOB,TYPEOB,NCC) décrivant les C-OBJETS, leur appartenance aux c-classes et leur type.

NCOB	LCOB	TYPEOB	NCC
OB1	client-permanent	PERM	client
OB2	client-adresse	VAR	client
OB3	client-situation	VAR	client
OB4	client-supprimé	SUP	client
OB5	demande de prêt	PERM	demande
OB6	prêt-refusé	VAR	demande
OB7	aspect permanent d'un prêt	PERM	prêt
OB8	aspect remboursement de prêt	VAR	prêt
OB9	échéancier de prêt	VAR	prêt
OB10	versement pour remboursement de prêt	PERM	versement
OB11	taxe	PERM	taxe
OB12	jour	PERM	jour

NCC : nom de la C-CLASSE à laquelle appartient le C-OBJET.

Extension de la relation COP(NCOP,LCOP,CTEXT,NCOB) décrivant les C-OPERATIONS

NCOP	LCOP	CTEXT	NCOB
OP1	refus de prêt	TOP1	OB6
OP2	accord de prêt	TOP2	OB7
OP3	mise en activité d'un client	TOP3	OB3
OP4	suspension d'un client	TOP4	OB3
OP5	mise en non activité d'un client	TOP5	OB3
OP6	calcul du montant total à rembourser en début de prêt	TOP6	OB8
OP7	révision d'un prêt	TOP7	OB8
OP8	calcul d'une taxe	TOP8	OB11
OP9	création d'échéancier de prêt	TOP9	OB9

NCOB : nom du C-OBJET 'modifié' par la C-OPERATION.

Extension de la relation CEV décrivant les C-EVENEMENTS

CEV(NCEV,LCEV,CPRED,NCOB)

NCEV	LCEV	CPRED	NCOB
EV1	arrivée de demande de prêt	CREATION	OB5
EV2	attribution d'un prêt	CREATION	OB7
EV3	changement de jour	CREATION	OB12
EV4	exigibilité d'un prêt	CREATION	OB9
EV5	début de remboursement d'un prêt	PRED5	OB9

NCOB : nom du C-OBJET 'constaté' par le C-EVENEMENT.

Extension de la relation CDEC décrivant les liaisons entre C-EVENEMENTS et C-OPERATIONS

CDEC(NCEV,NCOP,NCDEC,CCOND,CFAC,MUL)

NCEV	NCOP	NCDEC	CCOND	CFAC	MUL
EV1	OP1	CDEC1	NONC1	FAC1	UN
EV1	OP2	CDEC2	C1	FAC2	UN
EV2	OP6	CDEC3	TOUJOURS	FAC3	UN
EV3	OP9	CDEC4	TOUJOURS	FAC4	MULT
EV4	OP7	CDEC5	C2	FAC5	UN
EV4	OP8	CDEC6	C3	FAC6	UN
EV4	OP3	CDEC7	C6	FAC7	UN
EV4	OP4	CDEC8	C5	FAC8	UN
EV4	OP5	CDEC9	C4	FAC9	UN
EV5	OP3	CDEC10	TOUJOURS	FAC10	UN

Commentaire

Cette extension s'interprète de la façon suivante : une ligne exprime une association DECLENCHE et ses propriétés entre un C-EVENEMENT et une C-OPERATION.

Par exemple :

ligne 2 : un événement de la classe EV1 déclenche une opération de la classe OP2 si une condition C1 est satisfaite. C1 est le nom d'un prédicat dont le texte sera fourni ultérieurement.

ligne 4 : un événement de la classe EV3 déclenche toujours plusieurs opérations de la classe OP9. Le nombre d'opérations à exécuter et leurs paramètres sont fournis par la fonction FAC4 dont le texte sera également fourni ultérieurement.

Schéma des relations de type C-OBJET

C-OB PERM OB1 (noclient,dcclient,nomclient)
C-OB VAR OB2 (noclient,dmadclient, adclient)

C-OB VAR OB3 (noclient,dmsclient,sitclient)
C-OB SUP OB4 (noclient,dsupclient)
C-OB PERM OB5 (nodemande,dadem,noclient,revenuclient, montantd,durêed,jourexigibilité,date-remisefond)
C-OB VAR OB6 (nodemande,drefus,motif)
C-OB PERM OB7 (noprêt,nodemande,taux,dcprêt)
C-OB VAR OB8 (noprêt,drévision,montanttotremb,nbmens, datedebutremb)
/montanttotremb : montant total à rembourser/
/nbmens : nombre de mensualités/
/datedebutremb : date début de remboursement/
C-OB VAR OB9 (noprêt,dateéchéance)
/dateéchéance : date d'exigibilité d'un remboursement sur un prêt/
C-OB PERM OB10 (noversement,noprêt,dversement,montant-versement)
C-OB PERM OB11 (notaxe,dtaxe,montanttaxe,noprêt)
C-OB PERM OB12 (noannée,nomois,nojour)

Pour tout C-OBJET OB_i ayant une clé composée on introduit un constituant synonyme de cette clé appelé nob_i.

Schéma des relations de type C-OPERATION

C-OP OP1 (nop1,dop1,nob6)
C-OP OP2 (nop2,dop2,noprêt)
C-OP OP3 (nop3,dop3,nob3)
C-OP OP4 (nop4,dop4,nob3)
C-OP OP5 (nop5,dop5,nob3)
C-OP OP6 (nop6,dop6,nob8)
C-OP OP7 (nop7,dop7,nob8)
C-OP OP8 (nop8,dop8,notaxe)
C-OP OP9 (nop9,dop9,nob9)

Schéma des relations de type C-EVENEMENT

C-EV EV1 (nev1,dev1,nodemande)
C-EV EV2 (nev2,dev2,noprêt)
C-EV EV3 (nev3,dev3,nob12)
C-EV EV4 (nev4,dev4,nob9)
C-EV EV5 (nev5,dev5,nob9)

Schéma des relations de type C-DECLENCHE

C-DEC CDEC1 (nev1,nop1,ddec1)
C-DEC CDEC2 (nev1,nop2,ddec2)
C-DEC CDEC3 (nev2,nop6,ddec3)
C-DEC CDEC4 (nev3,nop9,ddec4)
C-DEC CDEC5 (nev4,nop7,ddec5)
C-DEC CDEC6 (nev4,nop8,ddec6)
C-DEC CDEC7 (nev4,nop3,ddec7)
C-DEC CDEC8 (nev4,nop4,ddec8)
C-DEC CDEC9 (nev4,nop5,ddec9)
C-DEC CDEC10 (nev5,nop3,ddec10)

II - LE SOUS SCHEMA DES DONNEES

II-1.- Définition

Le sous-schéma des données d'un schéma conceptuel de SI est composé de :

- l'ensemble des schémas des relations C-OBJETS regroupées en C-CLASSES
- la relation CCN de la métastructure
- la relation COB de la métastructure.

La première partie de ce sous-schéma correspond au schéma conceptuel classique de la base de données de l'organisation représentée.

Les deux relations CCN et COB complètent ce schéma conceptuel en donnant par l'intermédiaire des libellés, la sémantique de chaque C-CLASSE ou C-OBJET.

II-2.- Structure du sous-schéma des données

Le sous-schéma des données peut être perçu comme un ensemble de C-CLASSES muni d'une structure. Cette structure n'est pas exprimée directement dans la collection des schémas des C-OBJETS.

Elle peut se déduire des dépendances existant entre les clés des C-OBJETS et de l'appartenance des C-OBJETS aux C-CLASSES.

Nous proposons de l'exprimer sous la forme d'un graphe de dépendances fonctionnelles entre clés de C-CLASSES.

Dépendance fonctionnelle entre deux C-CLASSES

Soient deux C-CLASSES

C_1 $I_1[A_0,A_1 \ldots A_n]$ A_0,B_0 sont de type PERM

C_2 $I_2[B_0,B_1 \ldots B_n]$ $A_1,\ldots A_{n-1},B_1 \ldots B_{n-1}$ sont de type VAR

A_n,B_n de type SUP

Il existe une DF entre C_1 et C_2 si il existe $i \in [0,n]$: $I_1 \longrightarrow\!\!\!\!\!\!\cdot\ \ I_2$ existe dans A_i.

Dépendance fonctionnelle permanente entre deux C-CLASSES

Si $i = 0$ la DF est permanente.

Dépendance fonctionnelle non permanente entre deux C-CLASSES

Si $i \neq 0$ la DF est non permanente.

Sémantique des DF entre C-CLASSES

Les DF entre C-CLASSES représentent des associations particulières entre classes d'objets réels.

Le caractère permanent ou non permanent des DF apporte une précision sur ces associations.

L'existence d'une DFP entre deux C-CLASSES C_1 et C_2 signifie : qu'un élément de la classe représentée par C_1 est tou-

jours lié au cours du temps à un et un seul même élément de classe C_2.

Par exemple les c-classes demande et client sont liées par une DFP. En effet le client auquel est affecté un prêt est toujours le même tout au long de la gestion du prêt depuis la demande de prêt.

En revanche, l'existence d'une DF non permanente entre C_1 et C_2 signifie qu'un élément de la classe représentée par C_1 n'est lié à chaque instant qu'à au plus un élément de C_2, mais que cet élément peut être différent au cours du temps.

Supposons qu'à chaque client de la banque soit associé un agent commercial de son choix chargé d'étudier ses problèmes financiers et de le conseiller. La dépendance fonctionnelle entre la C-CLASSE Client et la C-CLASSE AGENT est non permanente parce que, au cours du temps, le client est libre de changer d'agent, ou bien parce que la durée d'activité des agents commerciaux dans la banque ne coïncide pas avec la période pendant laquelle une personne est client de la banque.

Associations entre C-CLASSES non exprimées sous forme de DF

Remarquons que, d'autre part, toute C-CLASSE, dont la clé est composée de clés d'autres C-CLASSES, représente également une certaine association entre ces C-CLASSES.

Association permanente et non permanente

Soit une C-CLASSE C $I[A_0,\ldots A_n]$ avec $I = I_1\ldots I_p$.

L'association représentée par C est permanente si C ne comprend pas de C-OBJET de type suppression, elle est non permanente dans le cas contraire.

Une association permanente entre deux C-classes C_1 et C_2 par exemple, signifie qu'au cours du temps les éléments de C_2 associés à un élément de C_1 sont toujours les mêmes.

II-3.- Représentation graphique du sous schéma des données

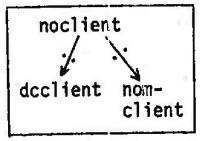
Cette représentation graphique a été proposée dans l'équipe REMORA par M. KRIEGUER (KRI 78), elle est assez proche de celle proposée par FLORY (FLO 77).

Représentation d'un C-OBJET

Un C-OBJET est représenté par son graphe de DF placé dans un rectangle.

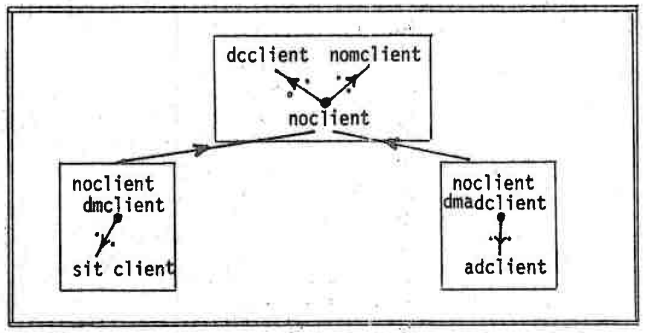
Exemple : Le C-OBJET

C-OB client (noclient, dcclient, nomclient) est représenté



Représentation d'une C-CLASSE

Une c-classe est représentée par l'ensemble de ses C-OBJETS regroupés dans un rectangle à double paroi.



Représentation des liens entre C-CLASSES

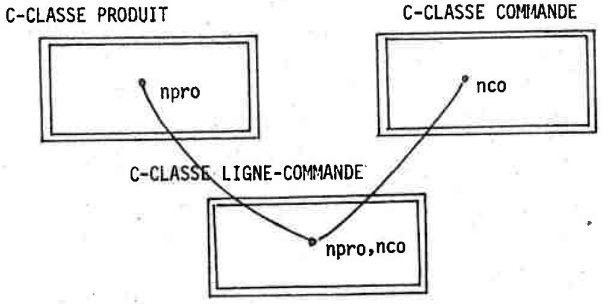
Les DF entre C-CLASSES sont représentées entre clés des C-OBJETS racines

→ représente une DF non permanente

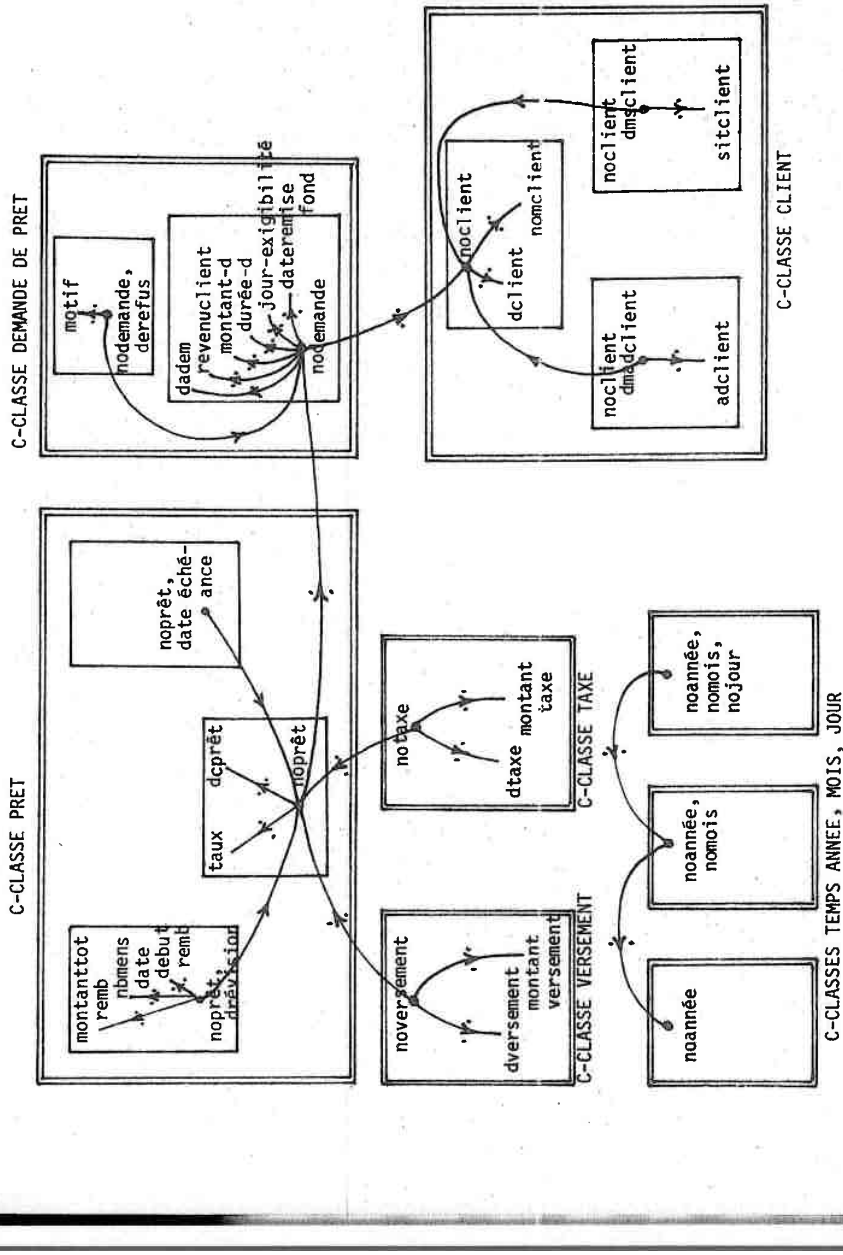
→ représente une DF permanente

Les associations entre C-CLASSES, non représentées par des DF sont représentées par des arcs non orientés reliant la clé de C-CLASSE association à celles des C-CLASSES qu'elle relie.

Exemple



II-4.- Sous-schéma des données de l'exemple gestion des prêts client



Commentaire

La représentation ci-contre met en évidence cinq C-CLASSES

- la C-CLASSE PRET
- la C-CLASSE CLIENT
- la C-CLASSE DEMANDE DE PRET
- la C-CLASSE VERSEMENT
- la C-CLASSE TAXE

correspondant aux cinq classes d'objets réels : prêt, client, demande de prêt, versement et taxe
et trois C-CLASSES temporelles ANNEE, MOIS et JOUR représentant trois classes d'objets temporels particuliers.

La représentation graphique visualise d'une part les différents aspects sous lesquels sont représentés les objets des différentes classes réelles et d'autre part les associations qui existent entre elles :

- un prêt correspond à une demande de prêt
- une demande de prêt est effectuée par un client
- une taxe porte sur un prêt
- un versement concerne un prêt.

III - LE SOUS-SCHEMA DE LA DYNAMIQUE

III-1.- Définition

Le sous-schéma de la dynamique d'un schéma conceptuel de SI est composé des relations suivantes de la métastructure :

- la relation COB
- la relation COP
- la relation CEV
- la relation CDEC

Ces relations représentent, comme nous l'avons montré au paragraphe 6.4 du chapitre 3, la fonction DECL du système réel représenté, c'est-à-dire ce que nous avons appelé sa dynamique.

III-2.- Structure du sous-schéma de la dynamique

Le sous-schéma de la dynamique peut être considéré comme un ensemble d'éléments appartenant à l'un des trois types C-OBJET, C-OPERATION et C-EVENEMENT et muni d'une structure.

Cette structure n'est autre que la collection des applications définies sur ou entre ces trois types appelés respectivement COB, COP et CEV.

Elle comprend :

- des liaisons entre C-EVENEMENTS et C-OBJETS correspondant à l'application CCONST,
- des liaisons entre C-OPERATIONS et C-OBJETS correspondant à l'application CMOD,
- des liaisons entre C-EVENEMENTS et C-OPERATIONS correspondant à l'application CDECL.

Ces liaisons expriment respectivement que :

- la création d'un nuplet d'un C-OBJET est suivie dans le temps d'un événement d'un C-EVENEMENT,
- la survenance d'un événement d'un C-EVENEMENT est suivie dans le temps de l'exécution d'opérations de certaines C-OPERATIONS,

- l'exécution d'une opération d'une C-OPERATION est suivie dans le temps de la création d'un nuplet d'un C-OBJET.

En raison de leur référence au temps nous appelons ces liaisons des dépendances chronologiques.

Nous donnons pour chacune d'elles sa définition dans le schéma conceptuel.

Nous introduisons des types de dépendances chronologiques permettant de rendre compte des applications CCOND, CFAC et CPRED et précisant ainsi la chronologie entre les classes de phénomènes.

III-2.1.- Dépendance chronologique entre C-EVENEMENTS et C-OBJETS

Définition de CCONST dans le schéma conceptuel

Rappelons que CCONST est de profil : CEV $\rightarrowtail$ COB

$\forall ev \in CEV$ CCONST(ev) = CEV[ev,NCOB]

Dépendance chronologique entre un C-EVENEMENT et un C-OBJET

Soient ev et ob un C-EVENEMENT et un C-OBJET.
Il existe une dépendance chronologique entre ev et ob si CCONST(ev) = ob.

Types de dépendance chronologique entre un C-EVENEMENT et un C-OBJET

- Dépendance chronologique forte

  La dépendance chronologique entre ev et ob est forte si CEV[ev,ob,PRED] = 'CREATION'

  Signification : Toute création d'un nuplet dans ob est un événement de ev.

  Exemple : EV1 et OB5 sont en dépendance chronologique forte.

- Dépendance chronologique faible

  La dépendance chronologique entre ev et ob est faible si CEV[ev,ob,PRED] ≠ 'CREATION'.

  Signification : Une création d'un nuplet de ob n'est un événement de ev que si un certain prédicat portant sur le changement d'état représenté par ce nuplet est vérifié.

  Exemple : EV5 et OB9 sont en dépendance chronologique faible.

  Le prédicat PRED5 exprime que la création d'une échéance pour un prêt n'est un évènement EV5 que s'il s'agit de la première créée pour ce prêt.

III-2.2.- Dépendance chronologique entre C-OPERATIONS et C-OBJETS

Définition de CMOD dans le schéma conceptuel

Rappelons que le profil de CMOD est : COP $\rightarrowtail$ COB

$\forall op \in COP$ CMOD(op) = COP[op,NCOB]

Dépendance chronologique entre une C-OPERATION et un C-OBJET

Soient op et ob une C-OPERATION et un C-OBJET, il existe une dépendance chronologique entre op et ob si CMOD(op) = ob.

Exemple : OP1 et OB6 sont en dépendance chronologique : l'exécution d'une opération de OP1 provoque la création d'un nuplet de OB6.

III-2.3.- Dépendances chronologiques entre C-EVENEMENTS et C-OPERATIONS

Définition de CDECL dans le schéma conceptuel

CDECL : CEV $\rightarrowtail \mathcal{P}$(COP)

$\forall ev \in CEV$ CDECL(ev) = CDEC[NCOP,ev]

Dépendance chronologique entre un C-EVENEMENT et une C-OPERATION

Soient ev et op un C-EVENEMENT et une C-OPERATION.
Il existe une dépendance chronologique entre ev et op si op ∈ CDECL(ev).

Types de dépendances entre un C-EVENEMENT et une C-OPERATION

- Dépendance chronologique forte

∃ une DCH forte entre ev et op si CDEC[ev,op,COND] = 'TOUJOURS'

Signification : Une DCH forte entre ev et op signifie qu'un événement de ev est toujours suivi d'opérations de op.

- Dépendance chronologique faible

∃ une DCH faible entre ev et op si CDEC[ev,op,COND] ≠ 'TOUJOURS'

Signification : Une DCH faible entre ev et op signifie qu'un événement de ev n'est suivi d'opérations de op que si,à l'instant où il se produit,l'état du système vérifie le prédicat défini par COND.

- Dépendance chronologique simple

∃ une DCH simple entre ev et op si CDEC[ev,op,MUL] = 'UN'

Signification : Une DCH simple entre ev et op signifie qu'un événement de ev ne peut être suivi que d'une seule opération de op.

- Dépendance chronologique multiple

∃ une DCH multiple entre ev et op si CDEC[ev,op,MUL] = 'MULT'

Signification : Une DCH multiple entre ev et op signifie qu'un événement de ev peut être suivi par plusieurs opérations de op.

En combinant ces définitions entre ev et op, nous aurons soit :

- une DCH forte et simple
- une DCH forte et multiple
- une DCH faible et simple
- une DCH faible et multiple

Exemples : EV2 et OP6 sont en dépendance chronologique forte et simple.
EV3 et OP9 sont en dépendance chronologique forte et multiple.
EV4 et OP7 sont en dépendance chronologique faible et simple.

Nous verrons dans le paragraphe suivant que cette typologie des dépendances chronologiques entre classes de phénomènes est très utile pour effectuer des contrôles de cohérence de la description de la dynamique du système réel effectuée dans le schéma conceptuel.

III-3.- Représentation graphique du sous schéma de la dynamique

La représentation graphique associée au sous-schéma de la dynamique comprend :

- les C-EVENEMENTS
- les C-OPERATIONS
- les C-OBJETS
- les dépendances chronologiques existant entre eux.

- Un C-OBJET est représenté par un cercle 
- Un C-EVENEMENT est représenté par un 
- Une C-OPERATION est représentée par une flèche

Les dépendances entre éléments sont représentées par la concaténation des symboles.

- Une dépendance chronologique faible est représentée par la présence d'un nom de condition sur la flèche correspondant à la C-OPERATION 

- Une dépendance chronologique multiple est représentée par la présence d'un chevron supplémentaire sur la flèche correspondant à la C-OPERATION

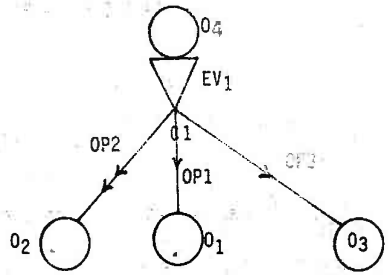
- Une dépendance chronologique faible entre un C-EVENEMENT et un C-OBJET est représentée en hachurant le triangle représentant le C-EVENEMENT 

Cycle dynamique

On appelle cycle dynamique l'ensemble constitué par un C-EVENEMENT, le C-OBJET qui lui est associé, les C-OPERATIONS en dépendance avec lui et les C-OBJETS associés à ces C-OPERATIONS.

Le terme cycle exprime le fait que dans le fonctionnement réel de l'organisation, la transaction décrite est exécutée de nombreuses fois, chaque fois qu'un évènement de la classe représentée survient.

Exemple théorique



Interprétation

O_1, O_2, O_3, O_4 sont des C-OBJETS

EV_1 un C-EVENEMENT

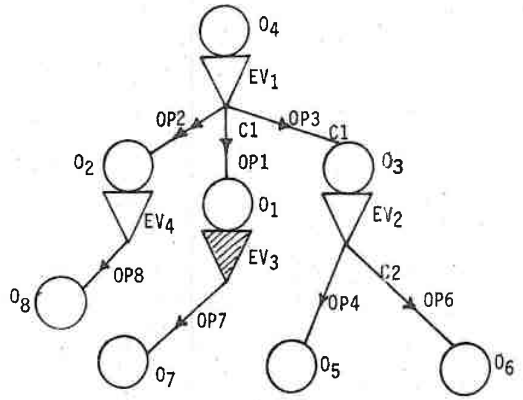
OP1, OP2, OP3 des C-OPERATIONS

Un événement EV_1 constate un changement d'état particulier d'un objet O_4.

Il déclenche - plusieurs opérations OP2 qui créent chacune un objet O_2
- si la condition C1 est vérifiée, une opération OP1 qui crée un objet O_1
- une opération OP3 qui crée un objet O_3.

Le schéma de la dynamique se présente comme une concaténation de cycles dynamiques qui permet de visualiser en particulier les dépendances directes et indirectes entre C-OPERATIONS et C-EVENEMENTS.

Exemple théorique



Ce schéma met en évidence quatre cycles dynamiques.

La concaténation des cycles par l'intermédiaire des C-OBJETS en dépendance à la fois avec des C-OPERATIONS et des C-EVENEMENTS met en évidence des dépendances entre C-EVENEMENTS et entre C-OPERATIONS :

Par exemple

Un événement de EV1 est suivi
- toujours par un événement de EV4 puisque EV1 est en DCH forte avec OP2 qui est en DCH avec O2 qui est en DCH forte avec EV4
- parfois par un événement de EV3 puisque EV1 est en DCH forte avec OP1 qui est en DCH avec O1 qui est en DCH faible avec EV3
- parfois par un événement de EV2 puisque EV1 est en DCH faible avec OP3 qui est en DCH avec O3 qui est en DCH forte avec EV2.

En appliquant un raisonnement de même type, on déduit que :

Une opération de OP2 est suivie
- toujours par une opération de OP8

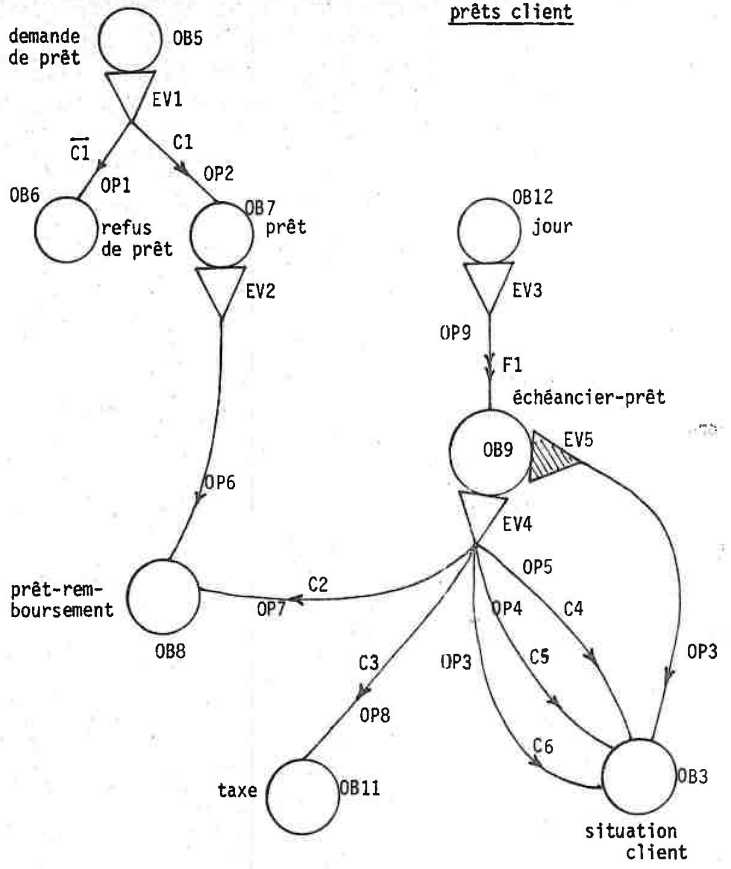
Une opération de OP3 est suivie
- toujours par une opération de OP4
- parfois par une opération de OP6

Dans l'équipe, RICHARD (RIC 79) a proposé une typologie de ces dépendances entre classes de même type qui permet de déduire du sous-schéma de la dynamique :
- un graphe des C-EVENEMENTS qui représente les liens de causalité entre événements
- un graphe des C-OPERATIONS qui représente les successions possibles d'opérations dans le SI.

Cette étude fait partie d'une évaluation du modèle conceptuel en tant que modèle permettant la représentation d'un problème de synchronisation, sur laquelle nous ne reviendrons pas dans ce travail.

Nous n'utilisons dans ce travail que les dépendances entre C-EVENEMENTS pour vérifier la cohérence de la dynamique d'un schéma conceptuel de SI. Nous en donnons la définition dans le paragraphe suivant.

III-4.- Sous-schéma de la dynamique de l'exemple gestion des prêts client



Commentaires

Ce schéma fait apparaître :

- 2 C-EVENEMENTS "externes" c'est-à-dire correspondant à des classes d'événements dont l'origine est une opération exécutée dans l'environnement du système représenté.

Ce sont :

- EV1 : arrivée d'une demande de prêt
- EV3 : changement de jour

- 3 C-EVENEMENTS "internes" c'est-à-dire correspondant à des classes d'événements dont l'origine est une opération exécutée à l'intérieur du système représenté.

Ce sont :

- EV2 : attribution d'un prêt
- EV4 : exigibilité d'un prêt
- EV5 : début de remboursement d'un prêt.

La représentation graphique met en évidence les conséquences directes et indirectes de chaque C-EVENEMENT.

Son interprétation est la suivante :

- A l'arrivée d'une demande de prêt (EV1), si le client satisfait une certaine condition (C1), un prêt est attribué (OP2), sinon le prêt est refusé (OP1).
- A l'attribution d'un prêt (EV2), on établit les caractéristiques du prêt concernant ses modalités de remboursement (OP6)
- Tous les jours (EV3), pour tous les prêts exigibles ce jour (F1) on crée une échéance (OP9).
- Si la création d'une échéance est la première pour ce prêt, il s'agit d'un début de remboursement de prêt (EV5) et le client correspondant est mis dans l'état actif (OP3).
- L'exigibilité d'un prêt (EV4) entraîne :
  - si à cette date, d'après les versements effectués par le client, il y a une certaine avance de remboursement (C2), la révision du prêt (OP7).

  - si, à cette date, d'après les versements effectués, le client a un certain retard de remboursement (C3), le calcul d'une taxe OP8
  - si le client a rattrapé son retard de remboursement depuis la dernière échéance (C6), le client est remis en situation actif (OP3)
  - si le client passe à cette échéance en retard de remboursement (C5), il est suspendu (OP4)
  - si le client a soldé son prêt à cette échéance (C4), il est mis en situation "non actif" (OP5).

IV - REGLES DE VALIDITE D'UN SCHEMA CONCEPTUEL

IV-1.- Schéma conceptuel valide

Un schéma conceptuel valide est un schéma conceptuel dont l'expression respecte les propriétés des concepts du modèle et qui de plus possède les qualités d'un schéma conceptuel canonique que nous avons introduites au chapitre 1 : complètude, fidélité, cohérence, non redondance.

Nous distinguons donc deux classes de règles de validité :

- des règles syntaxiques qui se déduisent du modèle conceptuel et dont la vérification sur un schéma conceptuel revient à contrôler que le modèle a été utilisé correctement.
- des règles sémantiques qui ont pour objectif de vérifier, dans la mesure du possible, les qualités du schéma conceptuel. Nous nous sommes essentiellement intéressés ici aux règles qui permettent d'effectuer un contrôle de la cohérence de la dynamique du système représenté.

Un schéma conceptuel de SI exprimé à l'aide du modèle comprend :

- la métastructure du SI : une collection d'extension de relations de schéma imposé par le modèle.
- la structure du SI : une collection de schémas de relations.

Un certain nombre de règles syntaxiques définissent les liens entre métastructure et structure.

Les autres règles sont des contraintes d'intégrité que doit vérifier toute métastructure de SI.

IV-2.- Rappel du schéma conceptuel de la métastructure d'un SI

Nous rappelons ici le schéma conceptuel de la métastructure d'un SI dans son expression instantanée.

C'est la collection de schémas de relations suivante :

CC(NCC,ICC,LCC)	représente l'ensemble des C-CLASSES NCC : nom d'une C-CLASSE ICC : nom du constituant, clé d'une C-CLASSE LCC : libellé explicatif
COB(NCOB,LCOB,TYPEOB,NCC)	représente l'ensemble des C-OBJETS NCOB : nom du C-OBJET LCOB : libellé explicatif TYPEOB : type du C-OBJET NCC : nom de la C-CLASSE du C-OBJET
CEV(NCEV,LCEV,CPRED,NCOB)	représente l'ensemble des C-EVENEMENTS NCEV : nom du C-EVENEMENT LCEV : libellé explicatif CPRED : nom prédicat associé au C-EVENEMENT NCOB : nom du C-OBJET constaté par le C-EVENEMENT
COP(NCOP,LCOP,CTEXT,NCOB)	représente l'ensemble des C-OPERATIONS NCOP : nom de la C-OPERATION LCOP : libellé explicatif CTEXT : nom du texte de la C-OPERATION NCOB : nom du C-OBJET modifié par la C-OPERATION
CDEC(NCOP,NCEV,NCDEC, CCOND,CFAC,MUL)	représente l'ensemble des associations C-DECLENCHE NCOP : nom de C-OPERATION NCEV : nom de C-EVENEMENT NCDEC : nom de l'association C-DECLENCHE

CCOND : nom de la condition de déclenchement
CFAC : nom du facteur de déclenchement
MUL : nature du déclenchement

Nous ne rappelons pas ici les domaines des constituants qui sont donnés au paragraphe 6-2.1. du chapitre 3.

Chaque fois que cela sera possible, nous exprimerons les règles de validité sur cette collection de relations à l'aide des opérateurs projection, produit et sélection définis sur les relations.

IV-3.- Règles de validité syntaxiques

Règle n° 1 : Complètude de la métastructure

- Tout C-OBJET 'modifié' par une C-OPERATION ou 'constatë' par un C-EVENEMENT est décrit.
- Toute C-OPERATION ou C-EVENEMENT aparaissant dans une association C-DECLENCHE est décrit.
- Tout C-EVENEMENT déclenche au moins une C-OPERATION.
- Toute C-OPERATION est déclenchée par au moins un C-EVENEMENT.

Expression sur la métastructure

CEV[NCOB] ⊂ COB[NCOB]
COP[NCOB] ⊂ COB[NCOB]
CDEC[NCOP] = COP[NCOP] et CDEC[NCEV] = CEV[NCEV]

Justification

Cette règle exprime simplement que les applications CMOD, CCONST et CDEC entre les ensembles COB, COP et CEV sont définies correctement.

Règle n° 2 : Complètude des C-CLASSES

Toute C-CLASSE possède un et un seul C-OBJET de type PERM.

Expression sur la métastructure

∀ncc ∈ CC[NCC] Card(COB[NCOB,'PERM',ncc]) = 1

Justification

Toute C-CLASSE possède au moins deux propriétés permanentes : sa propriété identifiant et sa propriété date de création, donc comprend toujours dans sa représentation un C-OBJET de type PERM.

Règle n° 3 : Cohérence entre métastructure et structure au niveau des noms de relation

Pour tout nom de C-OBJET, de C-EVENEMENT, de C-OPERATION ou de C-DECLENCHE apparaissant dans la métastructure, il existe dans la structure un schéma de relation portant ce nom et respectivement de type C-OB, C-EV, C-OP ou C-DEC.

Justification

Cette règle découle des deux niveaux de représentation
- le niveau des classes et de leurs interrelations : la métastructure,
- le niveau des éléments des classes et de leurs interrelations : la structure.

Règle n° 4 : Cohérence entre métastructure et structure au niveau des noms de constituants

- ∀op ∈ COP[NCOP] soit ob = COP[op,NCOB]
  alors la relation op de la structure admet un constituant qui est la clé de la relation ob.
- ∀ev ∈ CEV[NCEV] soit ob = CEV[ev,NCOB]
  alors la relation ev de la structure admet un constituant qui est la clé de la relation ob.

- ∀cdec ∈ CDEC[NCDEC] soit op = CDEC[cdec,NCOP]
  ev = CDEC[cdec,NCEV]
  alors la relation cdec de la structure a un constituant qui est la clé de la relation op et un autre qui est la clé de la relation ev.

Justification

Cette règle exprime simplement que les applications entre classes au niveau métastructure sont déduites des applications entre éléments de classes et réciproquement.

Règles de cohérence du sous-schéma des données

Règle n° 5 :

Les C-OBJETS d'une C-CLASSE sont indépendants les uns des autres.

Justification

Une dépendance fonctionnelle permanente entre deux C-OBJETS d'une même C-CLASSE correspondrait à une erreur d'utilisation du modèle : un non respect de la règle de maximalité des c-objets.

Règle n° 6 :

Seuls les C-OBJETS de type PERM peuvent être buts de dépendances fonctionnelles permanentes entre C-OBJETS.

Justification

Une dépendance fonctionnelle entre C-OBJETS représente une association entre classes d'objets réels. Dans une association les objets sont liés globalement et non par certains de leurs aspects.

Règle n° 7 : Cohérence du sous-schéma de la dynamique

Un C-OBJET apparaît une seule fois dans un cycle ou bien, s'il apparaît plusieurs fois, toutes les C-OPERATIONS qui lui sont associées sont en dépendance chronologique faible avec le C-EVENEMENT du cycle, et les conditions de déclenchement correspondantes sont telles qu'à un instant donné une seule peut prendre la valeur vraie.

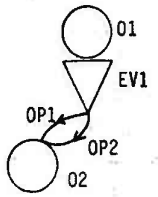
Expression sur la métastructure

∀ev ∈ CEV[NCEV]
∀op_i ∈ CDEC[ev,NCOP] soit ob_i = COP[op_i,NCOB]
i=1,n alors 1er cas : {ob_1...ob_n} ne contient pas de doubles
2ème cas : soit ob_k un c-objet apparaissant plusieurs fois dans {ob_1...ob_n}
soient op_{k_1}...op_{k_m} les opérations associées
telles que op_{k_i} ∈ CDEC[ev,NCOP]
et ob_k = COP[op_{k_i},NCOB]
soit $cond_i$ = CDECL[ev,op_{k_i},CCOND]
Alors ∀i ∈ [1,m] $cond_i$ ≠ TOUJOURS
et ∀o ∈ CEV[ev,NCOB]
si ∃i ∈ [1,m] : $cond_i$(o) = vrai alors
∀j ∈ [1,m] j ≠ i ⇒ $cond_j$(o) = faux.

Justification

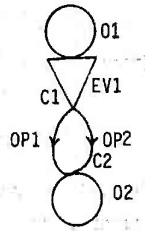
Cette règle provient de ce que, entre deux états successifs pris par le système, un objet ne peut subir qu'un seul changement d'état.

Elle interdit des cycles du type suivant :



dans lesquels il se pourrait que parfois OP1 et OP2 portent sur le même objet O2 ce qui conduirait à une indétermination de son état après l'exécution du cycle.

Elle exige que dans un cycle de ce type les conditions C1 et C2 s'excluent toujours.



IV-4.- Règles de validité sémantique

Règle n° 8 : Cohérence de la dynamique

Il n'existe pas dans le fonctionnement du système de circuits infinis.

Expression sur la métastructure

Dans l'équipe REMORA, RICHARD (RIC 79) a montré que cette règle peut s'exprimer à l'aide de dépendances chronologiques entre C-EVENEMENTS.

Nous donnons la définition de ces dépendances sur la métastructure et l'expression de la règle 8 qui en découle.

Dépendance chronologique directe entre deux C-EVENEMENTS

Soient ev_1 et ev_2 deux C-EVENEMENTS
il existe une DCH entre ev_1 et ev_2 si
$\exists op \in CDEC[NCOP,ev_1]$: ob = $CEV[ev_2,NCOB] = COP[op,NCOB]$

- DCH directe et forte

Cette DCH est forte si $CDEC(op,ev_1,CCOND)$ = 'TOUJOURS'
et $CEV(ev_2,CPRED)$ = 'CREATION'

- DCH directe et faible

Cette DCH est faible si $CDEC(op,ev_1,CCOND) \neq$ 'TOUJOURS'
ou $CEV(ev_2,CPRED) \neq$ 'CREATION'

Signification

Une DCH directe forte entre ev_1 et ev_2 signifie que tout événement de ev_1 est suivi dans le temps par un événement de ev_2.

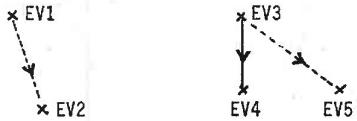
Une DCH directe faible entre ev_1 et ev_2 signifie que seuls certains événements de ev_1 sont suivis dans le temps par des événements de ev_2.

Graphe des événements

A partir des dépendances directes entre événements, on définit un graphe sur l'ensemble des C-EVENEMENTS. Pour rendre compte du type de la dépendance reliant deux événements, les arcs du graphe seront en traits pleins si elle est forte, sinon ils seront en pointillés.

Exemples

1) Graphe des événements associé au schéma conceptuel de l'exemple



Ce graphe est très simple et en particulier ne comporte

ni dépendances chronologiques indirectes entre événements ni circuit.

Nous prenons donc comme illustration de ce point un nouvel exemple de schéma conceptuel.

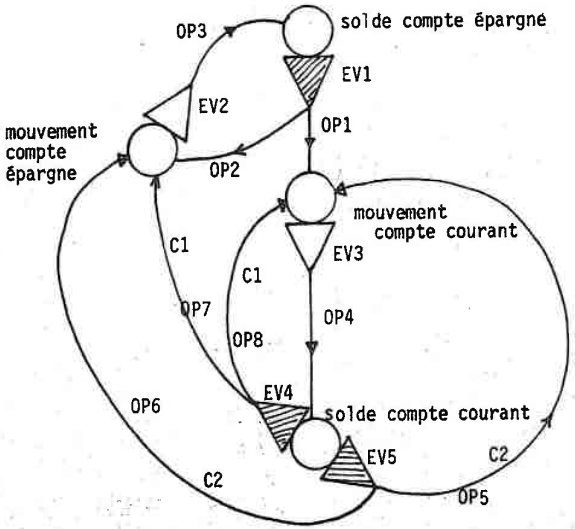
Il s'agit de la gestion couplée d'un compte courant et d'un compte épargne, définie par les règles suivantes :

- Lorsque le solde du compte épargne dépasse le plafond permis par le gouvernement, le surplus est versé sur le compte courant.

- Lorsque le solde du compte courant dépasse un plafond choisi par le titulaire du compte, le surplus est versé sur le compte épargne, à condition que celui-ci n'ait pas atteint le plafond permis.

- Lorsque le solde du compte courant devient négatif, le montant nécessaire à sa remise à zéro est prélevé, si cela est possible, sur le compte épargne.

Le sous-schéma de la dynamique correspondant est le suivant :



C1 : solde épargne > 0 C2 : solde épargne < plafond épargne

Il fait apparaître :

- les C-CLASSES EVENEMENT

EV1 : Passage du compte épargne au dessus du plafond épargne
EV2 : Arrivée d'un mouvement concernant le compte épargne
EV3 : Arrivée d'un mouvement concernant le compte courant
EV4 : Compte courant débiteur
EV5 : Passage du compte courant au-dessus du plafond-courant

- les C-CLASSES OPERATION

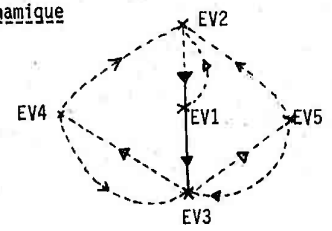
OP1 : création d'un versement sur le compte courant
OP2 : création d'un retrait sur le compte épargne
OP3 : mise à jour du solde du compte épargne
OP4 : mise à jour du solde du compte courant
OP5 : création d'un retrait sur le compte courant
OP6 : création d'un versement sur le compte épargne
OP7 : création d'un retrait sur le compte épargne
OP8 : création d'un versement sur le compte courant

- les C-OBJETS

solde-compte-épargne (nocpte-ép, date-mod-ep, solde-ep)
solde-compte-courant (nocpte-co, date-mod-co, solde-co)
mouvement-compte-épargne (noopep, nocpte-ep, sens-ep, montant-ep, date-ep) sens-ep='débit'ou'crédit'
mouvement-compte-courant (noopco, nocpte-co, sens-co, montant-co, date-co) sens-co='débit'ou'crédit'

C1 : le solde du compte-épargne est positif
C2 : le solde du compte-épargne est inférieur au plafond
$Pred_{EV1}$: le solde du compte-épargne est passé au dessus du plafond-épargne
$Pred_{EV4}$: le solde du compte-courant est devenu négatif
$Pred_{EV5}$: le solde du compte-courant est passé au dessus du plafond courant.

Graphe des événements associé au schéma conceptuel de la dynamique



Seuls EV1 et EV3 sont en dépendance chronologique directe.

Toutes les autres DCH sont faibles car soit les DCH entre C-EVENEMENT et C-OBJET sont faibles (existence d'un prédicat), soit les DCH entre C-EVENEMENT et C-OPERATION sont faibles (existence d'une condition de déclenchement).

Dépendance chronologique entre deux C-EVENEMENTS

Soient ev_1 et $ev_2 \in CEV[NCEV]$

ev_1 et ev_2 sont en dépendance chronologique s'il existe, dans le graphe des événements, un chemin allant de ev_1 à ev_2.

Dépendance chronologique forte

ev_1 et ev_2 sont en dépendance chronologique forte s'il existe un chemin allant de ev_1 à ev_2, dans le graphe des événements, tel que tout arc de ce chemin représente une dépendance chronologique directe et forte.

Dépendance chronologique faible

ev_1 et ev_2 sont en dépendance chronologique faible si, dans tous les chemins allant de ev_1 à ev_2 dans le graphe des événements, il existe au moins un arc représentant une dépendance chronologique directe et faible.

Expression de la règle 8

Aucun C-EVENEMENT n'est en dépendance chronologique forte avec lui-même.

Justification

Nous considérons a priori que toute description de la dynamique d'un système comprenant un circuit infini n'est pas valide car elle conduit à l'implantation de SI dont le fonctionnement ne peut être maîtrisable.

Application à l'exemple : Gestion couplée des comptes épargne et courant.

L'étude du graphe des événements montre que :

EV2, EV1, EV3, EV4, EV5 sont tous en dépendance chronologique avec eux-mêmes mais toutes ces dépendances sont faibles, donc la règle 8 est vérifiée.

Règle n° 9 : Non redondance de la dynamique

Il n'existe pas deux descriptions d'une même règle de gestion.

Expression de la métastructure

Cette non redondance peut s'exprimer au niveau des cycles en interdisant la présence de deux cycles identiques.

Soit ev_1 et $ev_2 \in CEV[NCEV]$:

$$CEV[ev_1,NCOB] = CEV[ev_2,NCOB]$$

et $$CDECL[ev_1,NCOP,CCOND] = CDECL[ev_2,NCOP,CCOND]$$

Règles de fidélité

Définir la fidélité d'une représentation n'est possible que par comparaison avec la réalité dont elle est l'image.

Les règles de fidélité que l'on peut associer à un schéma

conceptuel sont très nombreuses, elles comprennent :

- des règles élémentaires qui consistent à imposer à tout schéma de relation du schéma conceptuel d'être une bonne représentation de l'aspect du phénomène réel dont il est l'image.
- des règles générales qui consistent à imposer que toute représentation déduite du schéma conceptuel soit elle-même une représentation fidèle.

Nous avons choisi ici :

- deux règles concernant la dynamique qui nous semblent indispensables de vérifier sur tout schéma conceptuel de SI.
- une règle concernant les objets dont le non respect peut correspondre à des oublis dans le schéma conceptuel.

Règle n° 10 : Fidélité des trappes

Toute trappe dans le sous-schéma de la dynamique correspond à une situation réelle.

Une trappe est un noeud terminal du graphe associé au sous-schéma de la dynamique. Il correspond à un C-OBJET qui n'est lié à aucun C-EVENEMENT, c'est-à-dire un C-OBJET dont aucun changement d'état n'a de conséquence.

Expression sur la métastructure

Soit $o_1 \in$ COB[NCOB] o_1 est une trappe si

$o_1 \in$ COP[NCOB] et $\notin$ CEV[NCOB]

Justification

La vérification par le concepteur de cette règle peut faire apparaître des oublis dans la description de la dynamique que l'application des règles syntaxiques de complétude ne peuvent permettre de détecter.

Règle n° 11 : Fidélité des circuits

Tout circuit dans le sous-schéma de la dynamique correspond à une situation réelle.

Expression sur la métastructure

Nous avons vu que pour tout ev $\in$ CEV[NCEV] en dépendance chronologique avec lui-même il existe un circuit.

Justification

Tous les circuits détectés à ce niveau sont théoriquement des circuits finis, puisque la vérification de la règle 8 élimine les circuits infinis.

Cependant il nous semble nécessaire de demander au concepteur de vérifier que ces circuits existent réellement et que, d'autre part, l'expression des conditions de déclenchement et des prédicats est correcte et en assure toujours une terminaison.

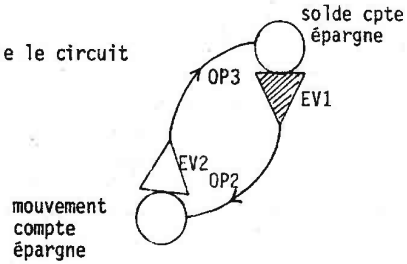
Exemple : Gestion couplée des comptes épargne et courant

- Sur cet exemple l'examen du graphe des événements met en évidence un certain nombre de circuits, 5 au total puisque chacun des C-EVENEMENTS est en dépendance chronologique faible avec lui-même.

- En fait, une étude approfondie des conditions de déclenchement, des prédicats et des textes des opérations montre que dans un fonctionnement d'un tel système ces circuits ne se produiraient pas.

- Prenons par exemple le circuit





Un événement EV1 n'a lieu que si $pred_{EV1}$ est vrai c'est-à-dire si le solde du compte épargne passe au dessus du plafond.

- OP2 a pour effet alors de créer un mouvement sur le compte épargne qui est un débit de ce compte correspondant à la somme dépassant le plafond.

- la création de ce mouvement est un événement EV2 qui déclenche une opération OP3 qui a pour effet de diminuer le solde du compte épargne.

- Cette opération en aucun cas ne pourra entraîner un événement EV1 puisqu'elle aura pour effet de faire passer le solde du compte d'une valeur supérieure au plafond à la valeur du plafond.

Donc il n'y aura pas de circuit : EV2 ⟶ EV1

Règle n° 12 : Rôle des C-OBJETS

Tout C-OBJET n'ayant aucun rôle dans le schéma conceptuel n'en a pas non plus dans le système réel et figure dans le SI à titre documentaire.

Expression sur la métastructure

Un C-OBJET OB joue un rôle dans le schéma conceptuel d'un SI si :

- il apparaît dans le sous-schéma de la dynamique
  soit si OB ∈ CEV[NCOB] ou OB ∈ COP[NCOB]

ou bien si

- il est utilisé dans l'expression d'un constituant de type TEXTE c'est-à-dire s'il est utilisé dans une condition de déclenchement, un prédicat, un texte de C-OPERATION ou un facteur de déclenchement.

- Cette deuxième partie de la règle ne peut être exprimée sur la métastructure que si cette dernière contient non pas

les noms de ces fonctions, comme nous l'avons supposé jusqu'à présent, mais leur expression dans un certain langage.

- Nous discutons dans le chapitre suivant du type de langage nécessaire pour exprimer ces fonctions.

Justification

Il nous semble nécessaire de faire vérifier au concepteur pour tout C-OBJET décrit dans le schéma conceptuel et n'y jouant aucun rôle, que cette situation correspond bien à la situation réelle afin de lui faire détecter d'éventuels oublis concernant la description de la dynamique de ces C-OBJETS.

CONCLUSION

Les règles que nous venons de présenter définissent un ensemble de contrôles à effectuer sur un schéma conceptuel de SI exprimé à l'aide du modèle proposé.

Tous ces contrôles, complétés par d'autres rendus possibles par l'intégration des contraintes d'intégrité dans la métastructure, sont effectués par l'outil d'aide à la conception d'un schéma conceptuel de SI que nous présentons au chapitre 6.

CHAPITRE 5 : DÉFINITION CONCEPTUELLE D'UN SI ET TYPE DE LANGAGE ASSOCIÉ

INTRODUCTION

Dans la première partie de ce chapitre nous montrons que pour obtenir une représentation abstraite du SI suffisante pour le réaliser, il est nécessaire de compléter le schéma conceptuel par un système d'intégrité. Nous appelons description conceptuelle du SI le schéma conceptuel complété par le système d'intégrité.

Nous donnons quelques exemples de contraintes d'intégrité spécifiques à un SI et nous montrons que le système d'intégrité d'un SI est réduit par rapport à celui d'une base de données.

Dans la seconde partie, nous donnons les caractéristiques que doit posséder un langage assurant la description conceptuelle d'un SI.

I - LE SYSTEME D'INTEGRITE

I-1.- Nécessité du système d'intégrité

La modèlisation proposée, comme toute modèlisation, implique des simplifications et des limites à la représentation. Il est évident, comme cela a été constaté à propos des modèles de données, qu'il subsistera toujours des aspects de la réalité que le modèle conceptuel ne pourra pas représenter intégralement et qu'il sera nécessaire d'adjoindre sous une autre forme au schéma conceptuel.

Nous supportons,en particulier, les limites de modèlisation du modèle relationnel qui ont été démontrées à propos de la modèlisation des données (SCH 77), (BAC 77).

Comme dans le contexte des données, nous appelons système d'intégrité ce complément d'ordre sémantique au schéma conceptuel décrit sous la forme d'un ensemble de contraintes d'intégrité (ESW 75), (HAM 76).

I-2.- Réduction du système d'intégrité d'un SI par rapport à celui d'une base de données

L'étude des exemples de contraintes d'intégrité associées à des schémas conceptuels de bases de données montre qu'une partie de ces contraintes portent sur les conditions de mise à jour des

données ou les transformations qu'elles peuvent subir et concernent en fait des phénomènes de la catégorie OPERATION ou EVENEMENT que la modèlisation limitée aux objets n'a pu prendre en compte.

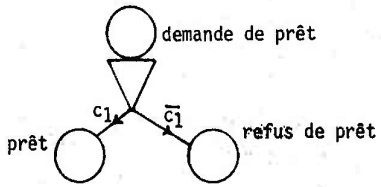
Ces contraintes deviennent donc inutiles dans le cadre d'un schéma conceptuel de SI représentant ces phénomènes.

Considérons par exemple quelques contraintes d'intégrité qui figureraient dans le système d'intégrité associé au schéma conceptuel de la base de données correspondant à l'exemple de gestion de prêts que nous avons traité.

Contrainte n° 1

Un client ne peut avoir plus d'un prêt à la fois.

Cette contrainte est décrite dans le schéma conceptuel du SI par le cycle dynamique



qui spécifie, par la condition c_1 qu'un prêt ne peut être accordé, en particulier, que si le client n'en a pas déjà un en cours.

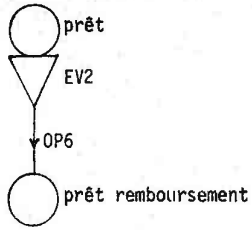
Contrainte n° 2 : Règle de validité de la mise à jour de la situation du client : la situation du client ne peut pas passer de la valeur nonactif à la valeur suspendu.

Cette contrainte est inutile puisque la gestion complète du c-objet représentant la situation du client est décrite dans le schéma conceptuel.

Notons que, dans une modèlisation limitée aux données, l'ensemble des opérations OP3, OP4, OP5 et leurs conditions respectives de déclenchement seraient décrites par des contraintes d'intégrité.

Contrainte n° 3 : On ne peut pas créer une occurrence de prêt-remboursement s'il n'existe pas au préalable une occurrence de prêt ayant même valeur de noprêt.

Cette contrainte est inutile, dans le schéma conceptuel du SI complet, puisque le cycle exprime le fait que la création d'une occurrence de prêt précède toujours celle d'une occurrence de prêt-remboursement.



Ainsi il apparaît sur ces exemples que le système d'intégrité d'un schéma conceptuel de SI est réduit par rapport à celui du schéma conceptuel de la base de données concernant la même organisation.

Il y subsiste cependant des contraintes que nous classons comme BRACCHI (BRA 79) en contraintes statiques et dynamiques.

Nous donnons quelques exemples de chacun de ces types de contraintes, spécifiques d'un schéma conceptuel de SI.

I-2.1.- Contraintes d'intégrité statiques

Définition

Une contrainte d'intégrité statique est une contrainte qui définit un état valide du SI sans faire référence à aucun autre état du SI.

Types de contraintes statiques

Ce sont :

- Les contraintes portant sur les constituants
  - domaine d'un constituant
  - contraintes entre domaines de plusieurs constituants d'une même relation.

- Les contraintes portant sur les relations
  - cardinalité d'une relation
  - contraintes de comparaison entre constituants de plusieurs relations.

I-2.1.1.- Exemples de contraintes portant sur les constituants

- Dans un schéma conceptuel de SI, l'expression des contraintes de domaine nécessite la définition de deux types de domaines particuliers correspondant aux types de constituants : DATE et TEXTE.

- Une contrainte de domaine associée à un constituant de type DATE définit l'unité de temps dans laquelle les valeurs prises par le constituant sont exprimées.

Exemple : OB7 (noprêt, nodemande, taux, dcprêt)

Contrainte : dcprêt type DATE unité JOUR.

- Une contrainte de domaine associée à un constituant de type TEXTE de la métastructure définit le langage dans lequel le texte du constituant doit être exprimé.

Exemple : COP2(NCOP,DMT,CTEXT)

Contrainte : CTEXT type TEXTE langage ISDEL.

Cette contrainte signifie que les textes des C-OPERATIONS seront exprimés dans le langage ISDEL, langage de définition d'un SI défini par LEIFERT (LEI 80) dans l'équipe REMORA.

- Une contrainte entre domaines de plusieurs constituants d'une même relation peut être nécessaire pour une relation correspondant à un C-OBJET externe, c'est-à-dire dont les occurrences ne sont pas créées par le SI.

Exemple : C-OB OB5 (nodemande, dadem, noclient, revenu client, montantd, duréed, jourexigibilité, date remise de fond)

Contrainte : date remise de fond > dadem + 30

exprime qu'un délai d'un mois est nécessaire pour l'attribution d'un prêt.

I-2.1.2.- Exemple de contraintes portant sur les relations

- Dans un schéma conceptuel de SI le rôle d'une contrainte de cardinalité est de limiter l'expansion du SI inévitable avec le mode de représentation choisi. Le cardinal de chaque relation devra être choisi suivant la fréquence de modification ou de création des éléments de la classe de phénomènes représentée et suivant la période pendant laquelle les gestionnaires ont besoin de connaître l'évolution de la classe sous cet aspect.

Exemple : C-OP OP2 (nop2,dop2,noprêt)

Contrainte : Cardinal = 600

Si la banque accorde en moyenne 100 prêts par mois, ce nombre signifie qu'on ne gardera trace des opérations d'accord de prêt qu'environ durant 6 mois.

- Une contrainte entre constituants de plusieurs relations peut être nécessaire entre relations représentant des C-OBJETS externes, c'est-à-dire modifiés uniquement par des opérations externes au SI.

Elle est inutile entre constituants de c-objets gérés par le SI, si l'on suppose que le SI suit exactement les règles de fonctionnement décrites dans le schéma conceptuel.

I-2.2.- Contraintes d'intégrité dynamiques

Définition

Une contrainte d'intégrité dynamique est une contrainte qui définit la validité des changements d'état se produisant dans le SI par référence aux états antérieurs pris par le système.

Ces contraintes sont définies sur les événements externes et définissent les conditions de validité des occurrences de C-OBJETS non créés par le SI.

On distingue :

- des contraintes portant sur des C-OBJETS appartenant à des C-CLASSES différentes :

  Exemple : Une demande de prêt ne peut être faite que par un client de la banque.

  Soit $\forall t$ $OB5_t[noclient] \subset OB1_t[noclient]$

- des contraintes portant sur des C-OBJETS appartenant à une même C-CLASSE

  Exemple : Il n'existe pas de contrainte de ce type dans l'exemple de la gestion des prêts.

Prenons la C-CLASSE suivante :

COMMANDE NCO[PREVI,DEGAGEE,ANNULEE,LIVREE]

C-OB PERM PREVI(NCO,DATECREA,NCLI,NPRO,QTE)
C-OB VAR DEGAGEE (NCO,DATEDEG,DELAILIV)
C-OB SUP ANNULEE (NCO,DATEANNUL)
C-OB VAR LIVREE (NCO,DATE-LIV,NOFACTURE)

Cette C-CLASSE représente les commandes d'une entreprise et chaque C-OBJET représente un type d'état possible de ces commandes.

Si on suppose que les changements d'état d'une commande ont une origine externe, il est nécessaire d'exprimer les contraintes suivantes :

- Une commande doit toujours être dégagée avant d'être livrée

$$\forall_t \ LIVREE_t[NCO] \subset DEGAGEE_t[NCO]$$

- Une commande dégagée ne peut être annulée

$$\forall_t \ ANNULEE_t[NCO] \cap DEGAGEE_t[NCO] = \emptyset$$

- Une commande annulée ne peut être livrée

$$\forall_t \ ANNULEE_t[NCO] \cap LIVREE_t[NCO] = \emptyset$$

- Dépendance chronologique entre C-OBJETS

Définition

Deux C-OBJETS O_1 et O_2 d'une même C-CLASSE de clé I sont en dépendance chronologique si $\forall_t$ on doit avoir $O2_t[I] \subset O1_t[I]$.

Signification

Un objet de la classe C ne peut passer dans l'état représenté par O_2 que si il est passé au préalable dans l'état représenté par O_1.

Dans l'exemple ci-dessus DEGAGEE et LIVREE sont en dépendance chronologique : une commande ne peut être livrée que si elle a été dégagée au préalable.

- Contrainte relative aux C-OBJETS de type VAR

Rappelons qu'un C-OBJET de type VAR peut représenter :

- soit un ensemble de propriétés d'une classe d'objets réels, variables au cours du temps,
- soit un ensemble de propriétés d'une classe d'objets réels, constants au cours du temps mais valorisés à une date ultérieure à la date de création des objets.

Cette distinction fondamentale de nature entre les propriétés n'est pas représentée dans le modèle. Il est donc nécessaire de la représenter par une contrainte d'intégrité.

Nous l'appelons contrainte de CONSTANCE.

Cette contrainte associée à un C-OBJET de type VAR de clé I,D, signifie que I,D n'est qu'une supercle et que la véritable clé du C-OBJET est I.

Exemple : les C-OBJETS DEGAGEE et LIVREE de l'exemple précédent sont munis de la contrainte de CONSTANCE. En effet une commande ne peut pas être dégagée ou livrée plusieurs fois.

Dans l'exemple de la gestion des prêts, seul le C-OBJET OB6 (refus de prêt) est muni de cette contrainte.

II - LANGAGE NECESSAIRE POUR EXPRIMER LA DESCRIPTION CONCEPTUELLE D'UN SI

Nous appelons description conceptuelle d'un SI son schéma conceptuel complété du système d'intégrité et nous nous interrogeons ici sur les caractéristiques que doit posséder un langage permettant son expression.

II-1.- Ce que doit permettre d'exprimer le langage

Le langage doit permettre l'expression de :

- la collection de schémas de relation de la structure du SI
- la collection de relations de la métastructure comprenant l'expression complète des valeurs des constituants de type TEXTE
- des contraintes d'intégrité.

Pour cela il doit permettre la définition :

- de types de schéma de relation
  Afin de définir les quatre types COB, COP, CEV et CDEC en précisant les constituants obligatoires qu'ils doivent comporter et leur type.
- de types de relation
  Afin de définir les relations de la métastructure.
- de types de constituants
  Afin de prendre en compte les trois types introduits : propriété, texte, date.
- de prédicats
  Afin d'exprimer les prédicats associés aux C-EVENEMENTS et les conditions de déclenchement . L'expression de ces prédicats nécessite les opérateurs classiques de manipulation des relations.
- d'algorithmes
  Afin d'exprimer les textes des C-OPERATIONS. L'écriture de ces textes nécessite des fonctions ou des procédures telles qu'on les rencontre dans les langages de programmation.

II-2.- Les langages existants

Parmi les langages existants à l'heure actuelle, deux classes de langages apparaissent répondre en partie aux exigences que nous venons de formuler.

Ce sont :

- les langages relationnels,
- les langages de haut niveau utilisant ou permettant la définition de types abstraits.

II-2.1.- Les langages relationnels

On appelle langage relationnel les langages conçus pour la définition et la manipulation des relations.

Ces langages utilisent soit l'algèbre relationnelle et le calcul des prédicats comme le langage ALPHA définit par CODD (COD 71), soit le calcul relationnel comme le langage SQUARE (BOY 74), soit des aspects de ces trois théories comme le langage SEQUEL (CHA 74).

Ces langages présentent deux caractéristiques :

- ils sont relationnellement complets.
- ils sont non procéduraux, ils permettent d'exprimer les questions concernant la base de données par la définition du résultat désiré, indépendamment de la manière réelle utilisée pour accéder aux n-uplets des relations permettant de l'obtenir.

PIROTTE (PIR 76), en particulier, a effectué une comparaison de ces langages et en a proposé une classification fondée sur les types de variables que l'on peut y utiliser.

Si l'on considère les caractéristiques que doit posséder le langage il apparaît :

- que les langages relationnels en possèdent deux :
  - la manipulation de relations
  - l'expression de prédicats
- ils ne permettent pas :
  - la définition de types de schémas de relation, de relations et de constituants.

- ils permettent difficilement l'écriture d'algorithmes, la facilité d'écrire les algorithmes dépend de la nature des variables de travail permises dans le langage.
- ils ne permettent pas la définition de fonctions sous forme de sous-programmes.

II-2.2.- Les langages de haut niveau utilisant les types abstraits

La nécessité d'utiliser des types dans le langage, types de schémas de relations, type de relations, types de constituants et d'associer des opérations à ces types : opérations de manipulation des relations, opérations de comparaison spécifiques sur le type DATE, nous a conduits à nous demander si les langages utilisant les types abstraits ne pouvaient pas servir de base à la définition d'un langage.

La notion de type abstrait a pour origine la notion de classe du langage SIMULA (DAH 70) qui permet de regrouper dans un même ensemble les fonctions de manipulation d'une même classe d'objets.

Elle a été définie par LISKOV (LIS 75) ; elle est maintenant utilisée dans de nombreuses recherches de nouveaux langages de programmation et de spécification (HAN 79), (MIN 79), (LES 79), (FIN 79).

Ces recherches ont pour objectif de permettre une définition du type par un ensemble d'opérations définies par des spécifications syntaxiques et sémantiques indépendantes de la "réalisation du type" c'est-à-dire de la structure physique d'implémentation choisie pour le type.

Certains de ces langages (WUL 75), (LAM 77), utilisent le calcul des prédicats pour exprimer ces spécifications.

D'autres langages comme SIMULA permettent l'expression d'algorithmes et la définition de fonctions, mais en revanche, ne permettent pas l'expression de prédicats.

En résumé, il apparaît qu'un langage utilisant ou permettant la définition de types abstraits, permet :

- la définition des types de relations et de constituants
- la définition des opérations de manipulation des relations
- parfois l'expression de prédicats
- l'expression d'algorithmes et de fonctions.

Un tel langage peut donc être utilisé pour exprimer une description conceptuelle de SI à condition d'y définir le type relation.

Solution retenue dans l'équipe REMORA

Dans le cadre du projet REMORA la solution retenue a été la première c'est-à-dire l'extension d'un langage relationnel.

LEIFERT (LEI 80) a défini un langage de description de SI, appelé ISDEL, dont les aspects relationnels sont proches de ceux du langage SEQUEL (CHA 74) et les autres aspects proches de ceux du langage PASCAL (HOA 72).

Solution retenue dans ce travail

Dans ce travail nous avons choisi une autre approche qui consiste à assister le concepteur dans sa tâche de description.

Nous avons donc construit un outil qui permet une entrée conversationnelle et progressive des éléments composant une description conceptuelle de SI et qui en effectue le contrôle.

Nous présentons cet outil au chapitre suivant.

QUATRIEME PARTIE

CHAPITRE 6 : SYSTÈME D'AIDE À LA CONCEPTION D'UN SYSTÈME D'INFORMATION

INTRODUCTION

L'outil que nous présentons ici s'intègre dans un système plus vaste, défini dans l'équipe de recherche REMORA appelé système de gestion de système d'information (SGSI en abrégé).

Un SGSI est au système d'information ce qu'est un SGBD à une base de données, c'est-à-dire un ensemble de logiciels qui en permet la création et l'utilisation tout en assurant sa sécurité.

Dans la première partie du chapitre, nous rappelons les rôles et l'architecture d'un SGSI tels qu'ils ont été définis dans l'équipe [ROL 79], [LEI 80], [ROL_2 81] et nous situons dans cet ensemble l'outil que nous avons réalisé.

Cet outil est un système d'aide à la conception d'un SI dont l'originalité est d'être utilisable tout au long de la vie du SI pour prendre en compte les modifications de structure se produisant dans l'organisation réelle au cours du temps.

Nous montrons tout d'abord l'intérêt de cet outil pour le concepteur de SI, puis nous présentons l'outil réalisé en donnant successivement :

- les fonctions qu'il assure,
- son architecture,
- la réalisation de chacune de ses fonctions.

On trouvera en annexe de la thèse :

- un exemple d'utilisation de cet outil,
- quelques textes des programmes qui le composent, l'ensemble des programmes se trouvant dans la notice d'utilisation de l'outil [FOU 82].

I - SYSTEME DE GESTION DE SYSTEMES D'INFORMATIONS (SGSI)

I-1.- Rôles du SGSI

Rappelons que nous avons défini au chapitre 1 le système d'information d'une organisation comme une représentation codée de la réalité, construite dans l'objectif de répondre aux besoins en information, variés et variables des gestionnaires.

Notre politique de conception d'un tel système consiste à faire une analyse, non pas des besoins des gestionnaires, mais de la structure de l'organisation, sous son double aspect statique et dynamique, et à construire une représentation normalisée de l'organisation à partir du résultat de cette analyse.

Cette représentation normalisée est le schéma conceptuel du SI.

Nous pensons que le travail du concepteur d'un SI doit se limiter à cette construction abstraite et qu'il est nécessaire de posséder un outil assurant le passage du schéma conceptuel à un SI concret construit sur un matériel informatique donné.

De plus, pour que cette représentation codée de la réalité qu'est le SI, garde au cours du temps sa qualité fondamentale de fidélité, il est nécessaire qu'un outil gère son évolution au cours du temps parallèlement et conformément à celle de la réalité.

Cette gestion est possible puisque le SI contient une représentation des règles de fonctionnement de l'organisation qui régissent l'évolution de ses constituants.

Nous proposons donc, dans l'équipe REMORA, d'associer à un SI un logiciel que nous appelons par analogie avec le domaine des bases de données, système de gestion de systèmesd'information dont les rôles sont les suivants :

1 - Réduire le travail du concepteur à celui de l'analyse et de la représentation des faits réels et aider le concepteur dans la formalisation de cette analyse.

2 - Faire en sorte que le SI soit en permanence une image fidèle de la réalité représentée.

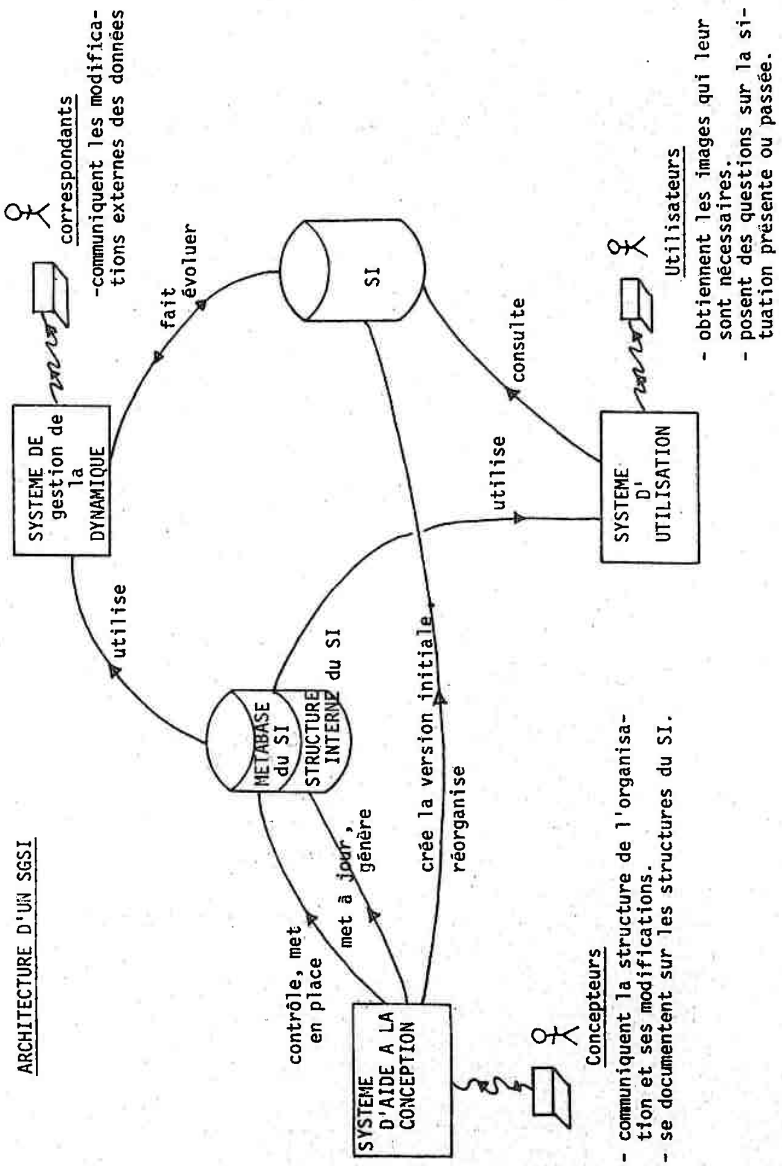
3 - Permettre une utilisation variée, souple, et simple des informations contenues dans le SI.

Ces rôles sont analogues à ceux d'un SGBD relationnel "idéal" tel qu'il est défini dans la littérature : (AST 76), (STO 76).

Remarquons cependant que, pour assurer le rôle n° 2, nous disposons dans le SI de plus de moyens que dans une base de données relationnelle puisque nous possédons la représentation complète des règles de fonctionnement de l'organisation réelle.

Cette représentation permet au SGSI de contrôler complètement l'exécution des traitements sur les données.

Un SGSI assure donc une fonction particulière correspondant à ce contrôle qui en constitue son originalité vis à vis des SGBD.



ARCHITECTURE D'UN SGSI

I-2.- Architecture du SGSI réalisé dans l'équipe REMORA

La figure ci-contre met en évidence l'architecture du prototype de SGSI réalisé dans l'équipe REMORA.

Ce prototype comprend trois sous-systèmes :

- un système d'aide à la conception
- un système de gestion de la dynamique
- un système d'utilisation

Ces trois sous-systèmes utilisent une base de données, interne au SGSI, appelée métabase du SI qui contient la description conceptuelle du SI et la description de sa structure interne.

- Le système d'aide à la conception

Ce système assure les fonctions classiques d'aide à la conception c'est-à-dire : le contrôle, l'intégration et la documentation de la spécification effectuée.

De plus, il effectue le choix d'une structure interne du SI et génère sa description.

Enfin, comme nous l'avons souligné dans l'introduction, il assiste le concepteur dans la maintenance du SI en lui offrant un ensemble de moyens pour communiquer les évolutions de structure se produisant dans l'organisation réelle et pour déterminer les conséquences de ces modifications sur le schéma conceptuel du SI et sur le SI lui-même.

La description de ce système fait l'objet de ce chapitre.

- Le système de gestion de la dynamique

Ce système, appelé encore processeur de la dynamique, a pour objectif de faire évoluer le contenu du SI conformément aux règles de fonctionnement décrites dans la description conceptuelle.

- Son principe de fonctionnement correspond à la mise en application de la fonction DECL telle qu'elle a été définie au chapitre 3 :

- Il reconnaît, parmi les modifications des données qui lui sont communiquées, les événements externes.

- Il recherche,dans la métastructure du SI stockée dans la métabase, les opérations à exécuter en conséquence de ces événements.

- Il lance et contrôle l'exécution de ces opérations et met à jour le SI.

- Il détermine si les changements d'états ainsi engendrés dans le SI sont des événements internes qu'il traite comme les précédents.

- Cet outil a été défini par LEIFERT (LEI 80) et réalisé par O. THIERY ($ROL_2$81). Il est une extension du SGBD SYNTEX (DEM 75).

- Le système d'utilisation du SI

Ce système a pour objectif de fournir aux utilisateurs un environnement leur permettant une consultation facile de l'ensemble du SI et de la métabase.

Il comprend un langage, prenant en compte explicitement le temps, qui leur permet de s'informer sur les points suivants :

- état présent de l'organisation
  - état de ses constituants, de sa structure ou de ses règles de gestion.
- état passé de l'organisation à une date quelconque
  - état passé de ses constituants,de sa structure ou de ses règles de gestion.
- évolution de l'organisation sur une période donnée.
- cause ou conséquence de tout changement d'état survenu à une date quelconque.

II - POURQUOI UN SYSTEME D'AIDE A LA CONCEPTION D'UN SYSTEME D'INFORMATION

L'expérimentation du modèle par les membres de l'équipe, sur des exemples d'organisations ou des organisations réelles a montré que son utilisation pose les problèmes classiques rencontrés dans la conception de logiciels importants c'est-à-dire les problèmes de documentation et de contrôle de validité de la solution élaborée.

En plus de ces problèmes, le domaine particulier des organisations auquel nous nous intéressons, a fait apparaître un problème supplémentaire, n'existant pas de façon aussi cruciale dans la conception d'autres types de logiciels : celui de la prise en compte de l'évolution de structure du monde réel dans le SI.

En effet, une organisation est un être vivant dont l'évolution est constante et rapide. Cette évolution concerne aussi bien la structure de ses éléments constitutifs que ses lois de fonctionnement. Il est même probable que de telles évolutions se produisent déjà pendant la période de temps consacrée à la conception du SI, ce qui exige de résoudre ce problème dès cette étape.

Le modèle proposé permet la représentation de ces évolutions de structure dans la métastructure du SI, mais laisse au concepteur la tâche très complexe de déterminer pour toute évolution de structure du monde réel, l'ensemble de ses conséquences sur le SI.

Nous avons donc ressenti,à ce niveau, la nécessité d'un outil aidant le concepteur dans la définition et la détermination des répercussions sur le SI de toute modification de structure se produisant dans l'organisation réelle.

La documentation et le contrôle sont bien évidemment des fonctions indispensables d'un tel outil.

- Le concepteur a besoin de se documenter sur le schéma conceptuel du SI avant et après la prise en compte d'une modification de structure : avant, afin de déterminer l'impact de la modification, après, afin de posséder une version à jour du schéma conceptuel.

- Toute modification étant une source potentielle d'erreurs, il est nécessaire, après toute modification d'un schéma conceptuel, d'en vérifier à nouveau la validité.

Ces fonctions de documentation et de contrôle de l'outil d'aide à la gestion de l'évolution de structure du SI sont en fait les mêmes que celles d'un outil d'aide à la conception d'un schéma conceptuel de SI dans un contexte non évolutif.

C'est pourquoi nous proposons un système d'aide à la conception d'un SI unique qui assiste le concepteur aussi bien pour la

conception initiale du SI que pour la prise en compte des évolutions de structure du monde réel lors de son utilisation.

L'originalité de ce système, par rapport à d'autres systèmes d'aide à la conception de base de données ou de SI, tels que MACSI (PEC 75), GALION (FLO 77) ou ISDOS (BOD 79) réside dans cette fonction de gestion de l'évolution de structure du SI.

III - FONCTIONS DE L'OUTIL

L'outil d'aide à la conception d'un SI assure trois fonctions principales :

- aide à la conception du SI initial
- documentation concernant le schéma conceptuel
- gestion de l'évolution de la structure du SI.

Pour assurer ces trois fonctions, l'outil utilise une base de données, appelée métabase du SI, qui contient sa description conceptuelle complète.

Détaillons ces trois fonctions.

III-1.- Aide à la conception du SI initial

L'aide à la conception du SI initial comprend :

- l'aide à la conception du schéma conceptuel initial du SI,
- la génération de la structure interne du SI,
- l'assistance à la création de la version initiale du SI.

III-1.1.- Aide à la conception du schéma conceptuel initial

L'aide à la conception du schéma conceptuel initial consiste en la mise en place de la métabase propre du SI, elle se compose :

- d'un langage permettant de définir soit en différé, soit dans un mode conversationnel, mais toujours de manière progressive et modulaire, le schéma conceptuel du SI,

- d'une méthode proposant au concepteur un ordre pour définir les différents éléments du schéma conceptuel,
- d'outils assurant
  - des contrôles des descriptions fournies
    - au fur et à mesure de leurs entrées
    - à la fin de la description du schéma conceptuel

    Ces contrôles correspondent à la vérification des règles de validité d'un schéma conceptuel présentées au chapitre 4.
  - une génération automatique de certains éléments de description.

III-1.2.- Génération de la structure interne du SI

Un outil génère, à partir de la description du schéma conceptuel stockée dans la métabase, la description de la structure interne du SI exprimée dans le langage de description de données du SGSI.

Dans le cas où le SGSI est une extension d'un SGBD, ce langage n'est autre que le langage de description de données du SGBD.

Un tel outil a été écrit par BANON (BAN 79) dans le cadre du SGBD SOCRATE. Il s'appuie sur un "mapping" général permettant de passer d'une collection de relations en 3ème forme normale à une structure de type réseau.

Un outil équivalent a été défini par LEIFERT (LEI 80) pour le SGBD SYNTEX.

Nous ne reviendrons pas sur cet outil dans ce travail.

III-1.3.- Création de la version initiale du SI

La création de la version initiale du SI consiste à mettre en place dans le SI

- d'une part les occurrences des C-OBJETS qui représentent l'état de l'organisation à l'instant du démarrage du fonctionnement du SI,
- d'autre part les valeurs initiales de tous les programmes

correspondants aux constituants de type TEXTE de la métabase (prédicats, conditions, facteurs et textes des opérations).

Cette création ne pose aucun problème sauf pour les C-OBJETS de type variable.

En effet, il est nécessaire de prévoir parallèlement à la création de ces C-OBJETS, une prise en compte de toutes les modifications intervenues dans l'organisation entre la date de création d'un C-OBJET et la date de début de l'exploitation effective du SI.

Cette fonction comprend donc :

En ce qui concerne la saisie des C-OBJETS

- Une méthode définissant un ordre de saisie des C-OBJETS à créer et un mode de prise en compte des modifications intervenant dans l'organisation pendant toute la période de création de la version initiale du SI.

- Un langage permettant une entrée, soit en mode différé, soit en mode conversationnel, des C-OBJETS.

- Des outils assurant :
  - le contrôle des informations entrées conformément aux contraintes d'intégrité déclarées dans le schéma conceptuel,
  - le passage de la forme externe des données à leur représentation interne.

Cette partie de la fonction "création de la version initiale du SI" est identique à la fonction création d'une base de données quelconque et nous ne la détaillerons pas ici. Nous noterons simplement qu'au niveau des SGBD classiques, il n'existe pas d'outil pour cette fonction, qui est donc à programmer, pour la création de chaque base de données particulière, en fonction du format externe des données.

En ce qui concerne la saisie des programmes

- Un langage pouvant être :

- Le langage du SGSI destiné à gérer le SI.

Par exemple dans le prototype réalisé par O. THIERY, il s'agit du langage SYNTEX muni d'un interface avec COBOL.

- Un langage de haut niveau (langage de spécification utilisant des types abstraits par exemple). Le SGSI doit alors comporter un générateur permettant de passer à des textes exprimés dans le langage du SGSI.

III-2.- Documentation du schéma conceptuel

La documentation concernant le schéma conceptuel est assurée par le système sous deux formes :

- une forme fixe correspondant à des éditions de tout ou partie d'un schéma conceptuel de SI selon un format prédéfini,
- une forme paramétrée permettant au concepteur de préciser les aspects du schéma conceptuel sur lesquels il désire se documenter.

Dans la forme fixe, la documentation fournie concerne toujours la structure du SI à l'instant de la demande.

Dans la forme paramétrée, le concepteur précise toujours la période à laquelle il s'intéresse et peut donc se documenter sur la structure du SI à un instant quelconque du passé.

III-3.- Gestion de l'évolution de structure du SI

Nous avons vu à la fin de chapitre 3 que le modèle conceptuel permet de représenter

- d'une part l'évolution des états pris par l'organisation au cours du temps,
- d'autre part l'évolution de sa structure.

La métabase contient donc une représentation dans le temps du schéma conceptuel du SI.

Cette représentation est construite en plusieurs étapes :

- une étape de création du schéma conceptuel initial qui fait partie de la fonction aide à la conception du SI initial que nous venons de présenter,

- des étapes de mise à jour du schéma conceptuel conformément aux modifications survenant dans la structure ou le fonctionnement de l'organisation réelle.

La fonction gestion de l'évolution de structure du SI se décompose en deux sous-fonctions :

- la mise à jour du schéma conceptuel citée ci-dessus,
- la mise à jour de la structure interne du SI.

La première sous-fonction nécessite de faire l'inventaire de chaque type de modification de structure pouvant intervenir dans l'organisation réelle et de déterminer pour chacune d'elle sa conséquence au niveau du schéma conceptuel du SI,tel qu'il est stocké dans la métabase du système.

Elle est assurée

- par un macrolangage permettant au concepteur de définir les modifications de structure du monde réel,
- par des outils assurant la mise à jour du schéma conceptuel dans la métabase du système et contrôlant sa validité après chaque mise à jour.

Cette mise à jour de la métabase n'est pas une mise à jour classique d'une base de données puisque l'historique des structures de l'organisation est conservée.

La seconde sous-fonction comprend :

a) la mise à jour de la description de la structure interne du SI.

Compte tenu du choix que nous avons fait, à savoir conserver l'histoire complète du SI, cette mise à jour ne consiste qu'en adjonctions de types de structures de données.

b) la création de la version initiale des nouvelles structures créées lorsque celles-ci sont des C-OBJETS.

Le fait de conserver l'histoire du SI au cours du temps simplifie les problèmes de réorganisation physique des données rencontrés dans les bases de données lors d'une modification de la structure de la base. En effet, cette réorganisation se bornera dans un SI à la création de nouvelles structures,soit à partir de données externes,soit à partir de données figurant déjà dans le SI.

Dans ce travail, nous n'avons réalisé que la première sous-fonction, la seconde nécessite d'avoir effectué un choix sur la structure interne du SI et sur l'outil qui le gère, elle est à l'étude dans l'équipe actuellement.

IV - ARCHITECTURE DU SYSTEME REALISE

Le système réalisé utilise le SGBD SOCRATE.

- La métabase du système est une base de données SOCRATE.
- Les outils sont des programmes SOCRATE écrits sur cette base et stockés dans la bibliothèque des programmes précompilés associée à la métabase.
- Les langages sont des macrolangages spécifiques définis à partir du langage de requête de SOCRATE.

Nous donnons successivement :

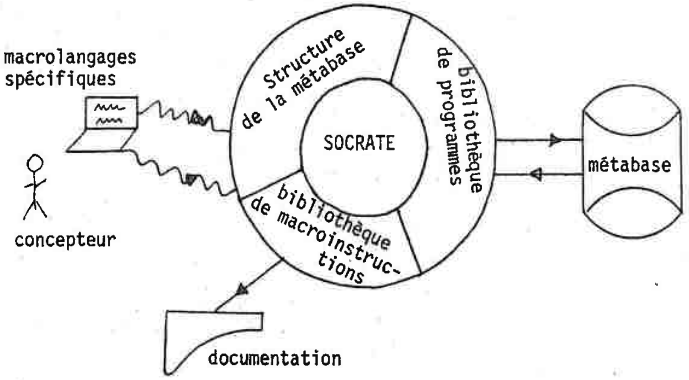
- Le schéma conceptuel de la métabase exprimé sous la forme d'une collection de relations en $3^{ème}$ forme normale.

Ce schéma est la métastructure du schéma conceptuel d'un SI présentéeà la fin du chapitre 3 complétée par des informations permettant la représentation de la structure du SI et celle des contraintes d'intégrité.

- La structure logique de la métabase implémentée ; cette structure logique contient quelques redondances au niveau des données ou au niveau des accès,qui ont été introduites volontairement afin d'optimiser certains programmes de contrôle ou d'édition.

Les principaux outils et macrolangages sont décrits dans les paragraphes suivants, relatifs à la mise en oeuvre de chaque fonction.

Le schéma ci-dessous illustre l'architecture de l'outil construit autour du noyau central qu'est le SGBD SOCRATE.



IV-1.- Schéma conceptuel de la métabase

La métabase d'un SI contient la description conceptuelle complète du SI au cours du temps c'est-à-dire :

- la métastructure du SI dans son expression temporelle,
- la structure du SI,
- les contraintes d'intégrité.

Le schéma conceptuel d'une métabase de SI est donc le schéma conceptuel de la métastructure d'un SI tel qu'il a été défini au chapitre 3 complété par des schémas de relations ou des constituants permettant la représentation de la structure du SI et des contraintes d'intégrité.

En ce qui concerne la représentation de la structure du SI, nous avons effectué les choix suivants : les seuls constituants de relations représentés sont les constituants des C-OBJETS et les constituants représentant des propriétés particulières des

C-OPERATIONS, des C-EVENEMENTS et des C-DECLENCHES (c'est-à-dire autres que les clés ou les propriétés de type date). Les noms des autres constituants des C-OPERATIONS et des C-EVENEMENTS sont choisis par le système.

Nous avons choisi de représenter comme contraintes d'intégrité :

- les contraintes de domaine des constituants
- la composition des clés des C-OBJETS
- les contraintes propres à chaque C-OBJET
- les dépendances chronologiques entre C-OBJETS
- les dépendances chronologiques directes entre C-EVENEMENTS.

Le schéma conceptuel de la métabase est présenté sous la forme d'une collection de METACLASSE, chaque METACLASSE est définie par l'ensemble des METAOBJETS qui la compose. Chaque METAOBJET est représenté par un schéma de relation permanente en 3ème forme normale.

- La sémantique de chaque METACLASSE ou METAOBJET est définie à sa suite entre //.
- La sémantique de chaque constituant de chaque METAOBJET est donné ensuite dans la liste des constituants de la métabase dans le paragraphe suivant.

```
METACLASSE CLASSE / Représente les C-CLASSES du SI/
      METAOBJET PERM CC1 (NCC, LCC, DCC)
      METAOBJET SUP  CC2 (NCC, DSC)
METACLASSE OBJET / Représente les C-OBJETS du SI/
      METAOBJET PERM COB1 (NCOB, LCOB, TYPOB, DCOP, NCC, CLE,
                           CONSTANCE)
                NCC nom de la C-CLASSE du C-OBJET
                CLE nom du constituant simple clé du C-OBJET
                Constance/Représente la contrainte d'intégrité de
                          "Constance" d'un C-OBJET de type VAR
                          contrainte définie au chapitre 5/
      METAOBJET SUP  COB2 (NCOB, DSOB)
```

METACLASSE EVENT / représente les C-EVENEMENTS du SI/
METAOBJET PERM CEV1 (NCEV, LCEV, DCE, NCOB)
METAOBJET VAR CEV2 (NCEV, DMP, PRED)
/représente les prédicats associés au cours du temps aux c-événements/
METAOBJET SUP CEV3 (NCEV, DSE)
METACLASSE OPER / représente les C-OPERATIONS du SI/
METAOBJET PERM COP1 (NCOP, LCOP, DCP, NCOB)
METAOBJET VAR COP2 (NCOP, DMT, TEXTOP)
/représente les textes des c-opérations au cours du temps/
METAOBJET SUP COP3 (NCOP, DSP)
METACLASSE DECL / représente les C-DECLENCHES du SI/
METAOBJET PERM CDEC1 (NCOP, NCEV, DCD)
METAOBJET VAR CDEC2 (NCOP, NCEV, DMCOND, CONDITION)
/représente les conditions de déclenchement au cours du temps/
METAOBJET VAR CDEC3 (NCOP, NCEV, DMFAC, FACTEUR, MUL)
/représente les facteurs de déclenchement au cours du temps/
METAOBJET SUP CDEC4 (NCOP, NCEV, DSD)
METACLASSE CONSTOB / Représente les constituants des C-OBJETS/
METAOBJET PERM CONSTOB1 (NCONSTOB, LCONSTOB, TYPCONST, DCCONS)
METAOBJET VAR CONSTOB2 (NCONSTOB, DMF, FORMAT)
/représente les contraintes de domaine/
METACLASSE COMPOB
METAOBJET PERM COMPOB1 (NCONSTOB, NCOB, DCOMPOB)
/représente l'appartenance des constituants aux c-objets/
METACLASSE COMPOSCLE
METAOBJET PERM COMPOSCLE1 (NCONSTOB1, NCONSTOB2, DCOMPOSCLE)
/représente la composition des clés des C-OBJETS en constituants élémentaires NCONSTOB1 et NCONSTOB2 sont des synonymes de NCONSTOB/

METACLASSE CONSTOP / Représente les constituants des C-OPERATIONS/
METAOBJET PERM CONSTOP1 (NCONSTOP, LCONSTOP, DCONSTOP, NCOP)
METACLASSE CONSTEV / Représente les constituants des C-EVENEMENTS/
METAOBJET PERM CONSTEV1 (NCONSTEV, LCONSTEV, DCONSTEV, NCEV)
METACLASSE CONTINT / Représente les contraintes d'intégrité relatives aux C-OBJETS/
METAOBJET PERM CONTINT1 (NCONTINT, LCONTINT, DCONTINT, NCOB)
METAOBJET SUP CONTINT2 (NCONTINT, DSCONTINT)
/Une contrainte d'intégrité ne peut être modifiée, elle est supprimée, et remplacée par une autre/
METACLASSE DEP-CHRON-OB / Représente les dépendances chronologiques directes entre C-OBJETS/
METAOBJET PERM DEP-CHRON-OB1 (NCOB1, NCOB2, DCHOB)
METAOBJET SUP DEP-CHRON-OB2 (NCOB1, NCOB2, DSHOB)
/NCOB1 et NCOB2 sont des synonymes de NCOB/
METACLASSE DEP-CHRON-EV / Représente les dépendances chronologiques directes entre C-EVENEMENTS/
METAOBJET PERM DEP-CHRON-EV1 (NCEV1, NCEV2, DCHEV, TYPDEP)
METAOBJET SUP DEP-CHRON-EV2 (NCEV1, NCEV2, DSEV)
/NCEV1, NCEV2, synonymes de NCEV/

Liste des constituants de la métabase

NCC	nom d'une c-classe - identifiant d'une c-classe
LCC	libellé explicatif d'une c-classe
DCC	date de création d'une c-classe
DSC	date de suppression d'une c-classe
NCOB	nom d'un c-objet - identifiant d'un c-objet
LCOB	libellé explicatif d'un c-objet
TYPOB	type du c-objet (permanent, variable ou suppression)
DCOB	date de création d'un c-objet
CLE	nom du constituant simple clé du c-objet
CONSTANCE	contrainte d'intégrité de "constance" d'un c-objet
DSOB	date de suppression d'un c-objet

NCEV	nom d'un c-événement - identifiant d'un c-événement
LCEV	libellé explicatif d'un c-événement
DCE	date de création d'un c-événement
DMP	date de modification du prédicat d'un c-événement
PRED	nom du prédicat d'un c-événement ou libellé 'création'
DSE	date de suppression d'un c-événement
NCOP	nom d'une c-opération - identifiant d'une c-opération
LCOP	libellé explicatif d'une c-opération
DCP	date de création d'une c-opération
DMT	date de modification d'un texte d'une c-opération
TEXTOP	nom du texte d'une c-opération
DSP	date de suppression d'une c-opération
DCD	date de création d'un c-déclenche
DMCOND	date de modification d'une condition de déclenchement
CONDITION	nom de la condition de déclenchement ou libellé 'TOUJOURS'
DMFAC	date de modification du facteur de déclenchement
FACTEUR	nom du facteur de déclenchement
MUL	multiplicité de déclenchement (UN ou MULT)
DSD	date de suppression d'un c-déclenche
NCONSTOB	nom d'un constituant de c-objet
LCONSTOB	libellé explicatif d'un constituant de c-objet
TYPCONST	type d'un constituant (date,propriété ou texte)
DCCONS	date de création d'un constituant
DMF	date de modification d'un format de constituant
FORMAT	domaine d'un constituant
DCOMPOB	date d'assignation d'un constituant à un c-objet
DCOMPOSCLE	date d'assignation d'un constituant à la clé d'un c-objet
NCONSTOP	nom d'un constituant d'une c-opération
LCONSTOP	libellé explicatif d'un constituant de c-opération
DCONSTOP	date de création d'un constituant de c-opération
NCONSTEV	nom d'un constituant d'un c-événement
DCONSTEV	date de création d'un constituant de c-événement
NCONTINT	numéro de contrainte d'intégrité
LCONTINT	libellé de contrainte d'intégrité
DCONTINT	date de création d'une contrainte d'intégrité

DSCONTINT	date de suppression d'une contrainte d'intégrité
DCHOB	date de création d'une dépendance chronologique entre c-objets
DSHOB	date de suppression d'une dépendance chronologique entre c-objets
DCHEV	date de création d'une dépendance chronologique entre c-événements
TYPDEP	type de dépendance chronologique entre c-événements (forte ou faible)
DSEV	date de suppression d'une dépendance chronologique entre c-événements.

IV-2.- Schéma logique de la métabase

IV-2.1.- Méthode d'obtention

Pour définir le schéma logique de la métabase, nous avons utilisé :

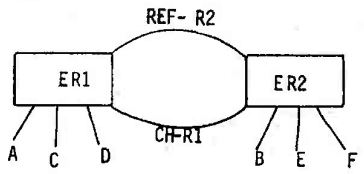
- la correspondance établie par BANON (BAN 79) entre un schéma conceptuel exprimé sous la forme d'une collection de relations en 3ème forme normale et une structure logique exprimée dans le langage de description de SOCRATE.
- une règle de simplification concernant la représentation des META-OBJET de type SUP.

Correspondance entre le modèle relationnel et le LDD SOCRATE

- Tout schéma de relation est représenté par une ENTITE (nom du type article en SOCRATE)
- Tout constituant de relation, non clé d'une autre relation, est représenté par une CARACTERISTIQUE de l'ENTITE représentant la relation.
- Tout constituant de relation, clé d'une autre relation ou synonyme d'une clé d'une autre relation, est représenté par une fonction d'accès de type REFERE.
- A toute fonction d'accès de type REFERE, est associée une fonction d'accès de type CHAINE

Soit le schéma conceptuel :

Exemple : R1 (A, B, C, D)
R2 (B, E, F)



Représentation graphique de la structure logique

Description dans le LDD SOCRATE

ENTITE ER1
DEBUT
A ——
C ——
D ——
REF-R2 REFERE UN ER2 PAR CH-R1
FIN
ENTITE ER2
DEBUT
B ——
E ——
F ——
CH-R1 CHAINE TOUT ER1 PAR REF-R2
FIN

ER1, ER2 sont des ENTITES.
A,C,D,B,E,F sont des CARACTERISTIQUES.
REF-R2 est une fonction d'accès qui permet à partir d'une occurrence de ER1 d'accéder à l'occurrence de ER2 qui lui est liée.
CH-R1 est une fonction d'accès qui permet à partir d'une occurrence de ER2 d'accéder à un ensemble d'occurrences de ER1 : toutes celles qui lui sont liées par REF-R2.

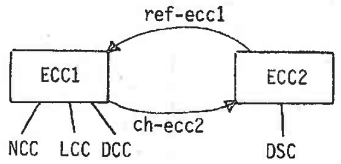
Règle de simplification concernant la représentation des META-OBJET de type SUP

Toute entité obtenue dans l'application de la correspondance précédente et représentant un métaobjet de type SUP est supprimée et il est ajouté,dans l'entité représentant le métaobjet de type Perm de la même métaclasse, une caractéristique représentant la date de suppression de la métaclasse.

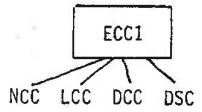
Exemple

Métaclasse CLASSE
Métaobjet perm CC1 (NCC, LCC, DCC, NCOBP)
Métaobjet sup CC2 (NCC, DSC)

Structure logique SOCRATE avant simplification



Structure logique SOCRATE après simplification



Cette simplification est possible puisque nous avons fait l'hypothèse au chapitre 2 qu'à tout objet réel, n'est associée qu'une seule date de suppression.

IV-2.3.- Choix concernant la représentation des valeurs des constituants de type texte

Nous avons choisi de faire figurer dans la métabase

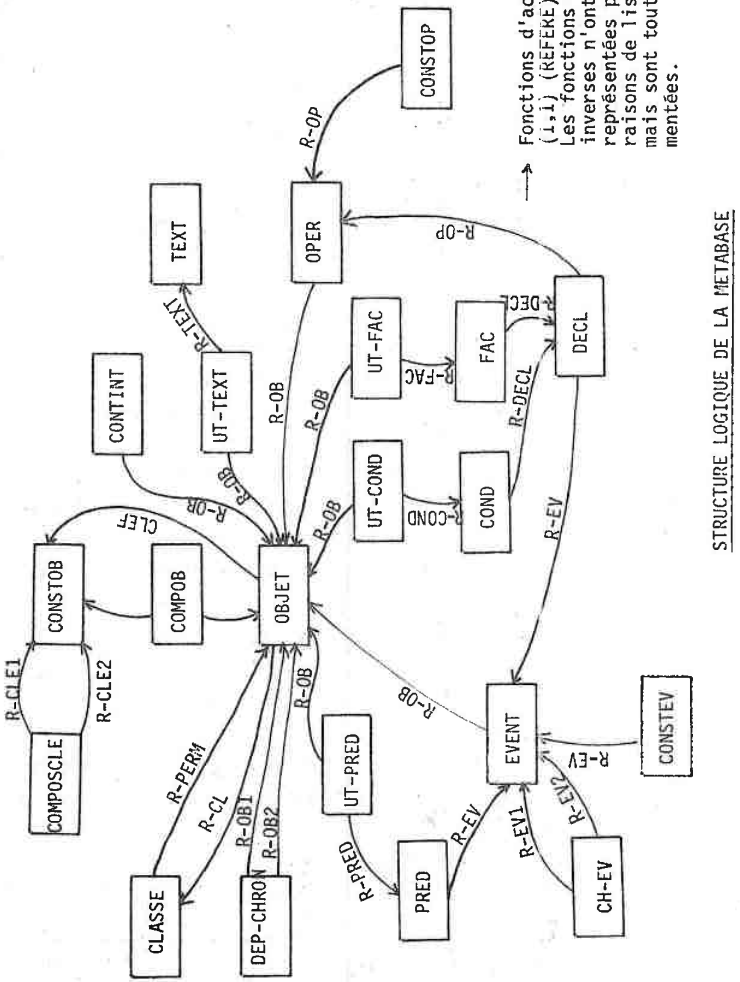
une spécification des constituants de type texte. Cette spécification peut être plus ou moins formelle suivant le désir du concepteur, elle n'est pas analysée par le système.

Cependant, pour aider le concepteur, lors de modification de structure du SI, à déterminer les conséquences de ces modifications au niveau des textes des programmes, nous avons mis en place dans la métabase, des entités représentant les associations "UTILISE" entre les textes et les C-OBJETS.

Il existe une association UTILISE entre un programme et un C-OBJET si ce C-OBJET apparait au moins une fois dans le texte du programme.

Ces associations sont mises en place :

- au niveau des conditions de déclenchement
  entité UT-COND
- au niveau des prédicats des événements
  entité UT-PRED
- au niveau des facteurs de déclenchement
  entité UT-FAC
- au niveau des textes des C-OPERATIONS
  entité UT-TEXT.



STRUCTURE LOGIQUE DE LA METABASE

V - FONCTION : AIDE A LA CONCEPTION DU SCHEMA CONCEPTUEL INITIAL

Rappelons que l'aide à la conception du schéma conceptuel initial d'un SI est assurée dans le système par

- une méthode
- un langage
- des outils de contrôle et de génération.

Développons ces trois éléments dans le système réalisé.

V-1.- Méthode de conception d'un schéma conceptuel de SI

L'expérimentation du modèle dans l'équipe Remora nous a conduit à proposer une méthode présentée par RICHARD (RIC 79) qui consiste à définir un schéma conceptuel de SI en trois étapes :

- conception du sous-schéma des données
- conception du sous-schéma de la dynamique
- définition des schémas de relation de la structure du SI.

Pour les deux premières étapes, il nous est apparu que la représentation graphique associée à chacun des sous-schémas était un instrument utile et agréable pour la mise au point de la métastructure.

Nous proposons donc dans l'outil, de communiquer au système les éléments d'un schéma conceptuel dans un ordre le plus proche possible de celui dans lequel ils apparaissent dans les représentations graphiques.

Le système demande donc au concepteur de lui communiquer

- successivement
  - les C-CLASSES
  - les C-OBJETS regroupés en C-CLASSES
  - les C-EVENEMENTS, les C-OPERATIONS et les C-DECLENCHES tels qu'ils se présentent sur la représentation graphique du sous-schéma de la dynamique,
- puis dans un ordre quelconque
  - les constituants des C-OBJETS, des C-OPERATIONS et des C-EVENEMENTS
  - les libellés explicatifs
  - les contraintes d'intégrité.

Notons que le système allège la définition de la structure, puisqu'il génère automatiquement les noms des constituants obligatoires des C-OPERATIONS, des C-EVENEMENTS et des C-DECLENCHES.

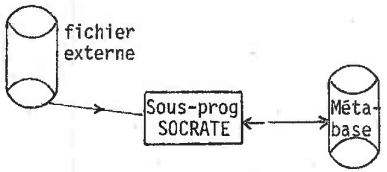
V-2.- Langage de définition d'un schéma conceptuel de SI

Le langage comprend deux sortes de primitives permettant la création des éléments d'un schéma conceptuel de SI

- des primitives utilisant des fichiers externes à SOCRATE et contenant les informations à entrer
- des primitives conversationnelles, permettant au concepteur de donner à la console directement les informations à entrer.

V-2.1.- Primitives utilisant des fichiers externes

Principe de fonctionnement



Les informations à entrer sont stockées dans un fichier classique sous une forme externe, aussi proche que possible de celle proposée pour exprimer un schéma conceptuel de SI au chapitre 4, c'est-à-dire sous forme de tableaux de valeurs.

Un sous-programme SOCRATE, stocké dans la bibliothèque des programmes précompilés de la métabase, contrôle ces informations et les transfère dans la métabase suivant le format interne de la structure logique.

Ces sous-programmes assurent en particulier, la mise en place de toutes les fonctions d'accès à partir des tableaux de valeurs correspondant aux relations de la métastructure.

Ce sont les primitives :

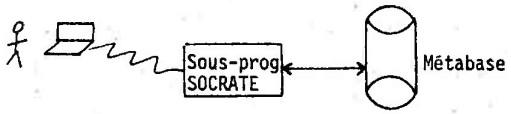
CCLASSE : création des c-classes

COBJET : création des c-objets
CONST : création des constituants des c-objets
CDYNLIB : création des libellés explicatifs des c-opérations et des c-événements.

Ces primitives sont bien adaptées à la création d'une version initiale d'un schéma conceptuel de SI assez important.

V-2.2.- Primitives conversationnelles

Principe de fonctionnement



- Le concepteur lance,dans une session conversationnelle,un sous-programme SOCRATE conçu sous la forme d'un dialogue homme machine.

- Le concepteur communique alors au système les informations dans l'ordre où celles-ci lui sont demandées.

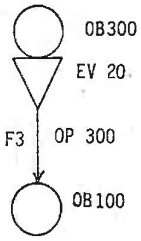
- Le programme contrôle leur validité et,en cas d'erreur,le signale au concepteur et propose un nouvel essai.

Liste des primitives :

- CRCLASSE - création d'une C-CLASSE
  - les différentes caractéristiques d'une C-CLASSE sont demandées les unesà la suite des autres au concepteur.
- CROBJET - création d'un C-OBJET et de ses constituants
  - le système demande successivement au concepteur
    - les caractéristiques du C-OBJET
    - son constituant clé
    - ses autres constituants
- CDYN - création de tout ou partie du sous-schéma de la dynamique.

Principe - Le sous-schéma de la dynamique est demandé au concepteur sous une forme la plus proche possible de sa représentation graphique,c'est-à-dire cycle par cycle.

Exemple : Le cycle



est communiqué au système comme suit :

```
NOM DE L EVENEMENT
?ev20
NOM DE L OBJET CONSTATE
?ob300
NOM DE PREDICAT
?creation
NOM DE L OPERATION
?op300
NOM DE L OBJET MODIFIE
?ob100
CONDITION TOUJOURS OU NOM PARTICULIER
?toujours
MULTIPLICITE UN OU MULT
?un
NOM DU FACTEUR
?f3
Y A-T-IL D AUTRES OPERATIONS DANS LE CYCLE O,N
?n
AVEZ VOUS D AUTRES CYCLES A ENTRER O,N
?o
```

- le programme effectue un certain nombre de vérifications et crée des occurrences d'entité OPER, EVENT, DECL, PRED, COND et FAC,ainsi que les fonctions d'accès les reliant.

- INTEGR 1 Entrée des contraintes d'intégrité de type CONSTANCE des C-objets
  - le programme édite le schéma de relation de chaque

C-OBJET ainsi que son libellé et demande s'il possède cette contrainte.

- INTEGR 2 — Entrée des dépendances chronologiques entre c-objets
  - le programme demande, c-classe par c-classe, d'entrer les couples de c-objets en dépendance chronologique.
- SCOBJET — Suppression d'un c-objet
  - le programme demande le nom du c-objet à supprimer et supprime toutes les entités CONSTOB, COMPOB et COMPOSCLE qui lui sont associées.
    Cette primitive doit être utilisée lorsque les programmes de contrôle ont détecté des erreurs dans le sous-schéma des données ou lorsque le concepteur s'aperçoit lui-même d'une erreur.
- MCYCLE — Modification d'un cycle
  - le programme demande le nom du c-événement du cycle à modifier et le type de la modification :
    - supprimer le cycle complètement
    - supprimer une c-opération
    - ajouter une c-opération
    - modifier un c-déclenche (facteur ou condition)

  et, suivant le type de la modification, les informations nécessaires, puis corrige la métabase en conséquence.

Remarques : Ces deux primitives ne sont à utiliser que dans la phase de mise au point du schéma conceptuel initial. Aucune trace des mises à jour correspondantes n'est conservée dans la métabase. Lors d'une modification du schéma conceptuel consécutive à une modification de la structure de l'organisation réelle, on utilisera les primitives associées au langage de définition de l'évolution de la structure et alors les versions successives du schéma conceptuel seront conservées.

Exemple :

```
$go mcycle
NOM DU C-EVENEMENT DONT LE CYCLE EST A MODIFIER
?evbid
OB1     <----EVBID    ---->OP45     ,OP5     ,
                             ↓          ↓
                            OB5       ,OB3     ,
POUR SUPPRIMER LE CYCLE TAPEZ 1
 POUR SUPPRIMER UNE OPERATION TAPEZ 2
POUR AJOUTER UNE OPERATION TAPEZ 3
 POUR MODIFIER UN DECLENCHEMENT TAPEZ 4
?2
NOM DE L OPERATION A SUPPRIMER
?op5
VOICI VOTRE NOUVEAU CYCLE
OB1     <----EVBID    ---->OP45     ,
                             ↓
                            OB5       ,
```

V-3.- Les programmes de contrôle d'un schéma conceptuel

Les programmes de contrôle ont pour objectif de vérifier que le schéma conceptuel stocké dans la métabase respecte les règles de validité définies au chapitre 4.

IV-3.1.- Contrôle des règles de validité syntaxiques

Les règles syntaxiques sont presque toutes, soit vérifiées dans les programmes de création définis précédemment, soit assurées automatiquement par le système lorsque celui-ci génère des éléments de description.

C'est le cas des règles :

- numéro 1 : concernant la complétude de la métastructure
  - l'ordre imposé pour entrer les éléments constitutifs du schéma conceptuel interdit ou signale les oublis qui pourraient se produire dans une conception manuelle d'un schéma conceptuel.
- numéros 3 et 4 : concernant la cohérence entre la métastructure et la structure

  Cette cohérence est assurée par le fait qu'une

grande partie des schémas de relation de la structure sont générés par le système.

La vérification des autres règles est assurée par les programmes suivants :

- CONTROL1 : contrôle de la complètude des C-CLASSES
  correspond à la règle n° 2.
- CONTROL2 : contrôle de la cohérence du sous-schéma des données
  correspond aux règles n° 5 et 6.
- CONTROL3 : contrôle de la cohérence du sous-schéma de la dynamique
  correspond à la règle n° 7 vérifiant qu'un même objet ne subit pas deux changements d'état différents après un événement donné.

V-3.2.- Contrôle des règles de validité sémantique

- CONTROL4 : Recherche des circuits dans le sous-schéma de la dynamique
  correspond à la règle n° 8 interdisant la présence de circuit inconditionnel et à la règle n° 11 demandant le contrôle des circuits conditionnels.

Ce programme comprend deux modules :

CONSGHEV : création du graphe des C-EVENEMENTS (création des entités GH-EV).

CIRCUIT : recherche des circuits dans le graphe GH-EV.

Exemple : L'application de ce programme à l'exemple de la gestion couplée d'un compte épargne et d'un compte courant présentée au chapitre 4 page 184 : donne le résultat suivant :
(Pour des raisons de compatibilité avec les autres c-événements déjà décrits dans le système, les noms des c-événements sont modifiés par rapport à ceux donnés sur le schéma page 184 : EV1 devient EV10 etc ...).


```
IL EXISTE LE CIRCUIT SUIVANT
EV10    --->EV30    ...>EV40    ...>EV30    ...>EV50    ...>EV20    ...>EV10
CIRCUIT CONDITIONNEL
IL EXISTE LE CIRCUIT SUIVANT
EV10    --->EV30    ...>EV40    ...>EV20    ...>EV10
CIRCUIT CONDITIONNEL
IL EXISTE LE CIRCUIT SUIVANT
EV10    --->EV30    ...>EV50    ...>EV30    ...>EV40    ...>EV20    ...>EV10
CIRCUIT CONDITIONNEL
IL EXISTE LE CIRCUIT SUIVANT
EV10    --->EV30    ...>EV50    ...>EV20    ...>EV10
CIRCUIT CONDITIONNEL
IL EXISTE LE CIRCUIT SUIVANT
EV10    --->EV20    ...>EV10
CIRCUIT CONDITIONNEL
IL EXISTE LE CIRCUIT SUIVANT
EV30    ...>EV40    ...>EV30
CIRCUIT CONDITIONNEL
IL EXISTE LE CIRCUIT SUIVANT
EV30    ...>EV40    ...>EV30    ...>EV50    ...>EV30
CIRCUIT CONDITIONNEL
IL EXISTE LE CIRCUIT SUIVANT
EV30    ...>EV50    ...>EV30
CIRCUIT CONDITIONNEL
```

- CONTROL5 : Non redondance du sous-schéma de la dynamique
  correspond à la règle n° 9
  Le programme compare deux à deux les cycles dynamiques associés à un même C-OBJET et vérifie qu'ils ne comprennent pas les mêmes opérations.
- CONTROL6 : Recherche des trappes dans le sous-schéma de la dynamique
  correspond à la règle n° 10
- CONTROL 7 : Recherche du rôle des C-OBJETS
  correspond à la règle n° 12
  Le programme recherche tous les C-OBJETS "indépendants" c'est-à-dire non liés à un C-EVENEMENT ou à

une C-OPERATION et non utilisés dans un texte de condition, de facteur, de prédicat ou d'opération et les signale aux concepteurs.

VI - FONCTION DOCUMENTATION

Rappelons que la documentation est assurée par le système sous deux formes :

- une forme fixe
- une forme paramétrée.

- La forme fixe se compose d'un certain nombre de programmes d'édition dont le concepteur peut demander l'exécution à tout moment.

- La forme paramétrée est un macrolangage spécifique, extensible au gré du concepteur, dont nous nous bornerons à donner quelques exemples de macro-instructions.

VI-1.- Programmes d'édition du système

Les programmes d'édition non paramétrés donnent des informations sur l'état du schéma conceptuel du SI à l'instant où ils sont lancés.

Pour obtenir une documentation concernant un état passé du schéma conceptuel ou du SI, il faut utiliser un programme paramétré dans lequel la date à laquelle on s'intéresse est passée en paramètre.

Le système comprend actuellement les programmes suivants (il est évident que bien d'autres pourraient être écrits) :

ESCHDONN : Edition du sous-schéma des données sous la forme d'une collection de schémas de C-OBJET regroupés en C-CLASSE.

Exemple : tiré de la gestion des prêts décrite au chapitre 4

```
CLASSE  PRET      CLE NOPRET      PRET ACCORDE PAR LA BANQUE

C-OBJET  PERM   OB7     (NOPRET  ,NODEMAND,TAUX     ,DCPRET  )
C-OBJET  VAR    OB8     (NO9     ,NOPRET  ,DREVISIO,MTTOTREB,NBMENSUA,DDEBREMB)
C-OBJET  VAR    OB9     (NO10    ,NOPRET  ,DTECHEAN)

CLASSE  VERST     CLE NOVERSEM    VERSEMENT POUR REMBOURSEMENT DU PRET

C-OBJET  PERM   OB10    (NOVERSEM,NOPRET  ,DTVERSEM,MONTVERS)

CLASSE  TAXE      CLE NOTAXE      TAXE POUR RETARD DE REMBOURSEMENT

C-OBJET  PERM   OB11    (NOTAXE  ,DTAXE   ,MONTTAXE,NOPRET  )
```

ESCHDYN : Edition du sous-schéma de la dynamique sous la forme d'une collection de schémas de relation regroupés par type.

Exemple : tiré de la gestion des prêts

```
C-EVENEMENTS
C-EV    EV1 (IEV1     ,DAREV1     ,NODEMAND)
C-EV    EV2 (IEV2     ,DAREV2     ,NOPRET  )
C-EV    EV3 (IEV3     ,DAREV3     ,NOANNEE )
C-EV    EV4 (IEV4     ,DAREV4     ,NO10    )
C-EV    EV5 (IEV5     ,DAREV5     ,NO10    )

C-OPERATIONS
C-OP    OP1     (IOP1     ,DOP1     ,NO7     )
C-OP    OP2     (IOP2     ,DOP2     ,NOPRET  )
C-OP    OP6     (IOP6     ,DOP6     ,NO9     )
C-OP    OP9     (IOP9     ,DOP9     ,NO10    )
C-OP    OP7     (IOP7     ,DOP7     ,NO9     )
C-OP    OP8     (IOP8     ,DOP8     ,NOTAXE  )
C-OP    OP3     (IOP3     ,DOP3     ,NO4     )
C-OP    OP4     (IOP4     ,DOP4     ,NO4     )
C-OP    OP5     (IOP5     ,DOP5     ,NO4     )

C-DECLENCHES
C-DECL   DECL1      (IEV1     ,IOP1     ,DDECL1      )
C-DECL   DECL2      (IEV1     ,IOP2     ,DDECL2      )
C-DECL   DECL3      (IEV2     ,IOP6     ,DDECL3      )
C-DECL   DECL4      (IEV3     ,IOP9     ,DDECL4      )
C-DECL   DECL9      (IEV4     ,IOP7     ,DDECL9      )
C-DECL   DECL10     (IEV4     ,IOP8     ,DDECL10     )
C-DECL   DECL11     (IEV4     ,IOP3     ,DDECL11     )
C-DECL   DECL12     (IEV4     ,IOP4     ,DDECL12     )
C-DECL   DECL13     (IEV4     ,IOP5     ,DDECL13     )
C-DECL   DECL14     (IEV5     ,IOP3     ,DDECL14     )
```

ECONTINT1 : Edition de la composition des clés des C-OBJETS

ECONTINT2 : Edition des dépendances chronologiques entre C-OBJETS

EXPLORDYN : Représentation graphique du sous-schéma de la dynamique.
Ce programme édite le sous-schéma de la dynamique, cycle par cycle, de manière linéaire.

Exemple : l'édition du sous-schéma de la dynamique de l'exemple de la gestion des prêts est la suivante :

```
$go explordyn
OB5     <-----EV1     ----->OP1     ,OP2
                          |           |
                        OB6         ,OB7
OB7     <-----EV2     ----->OP6     ,
                          |
                        OB8         ,
OB12    <-----EV3     ----->OP9     ,
                          |
                        OB9         ,
OB9     <-----EV5     ----->OP3     ,
                          |
                        OB3         ,
OB9     <-----EV4     ----->OP7     ,OP8      ,OP3      ,OP4      ,OP5
                          |           |         |         |         |
                        OB8         ,OB11     ,OB3      ,OB3      ,OB3
```

Il s'agit d'un programme de parcours de graphe particulier dont les arcs sont représentés dans la métabase par les fonctions d'accès entre les entités représentant les C-OBJETS, les C-OPERATIONS, les C-EVENEMENTS et les C-DECLENCHES.

Son algorithme est le suivant :

Pile ← vide TO ← vide TO : liste des objets déjà décrits

Tant que tous les objets auxquels est associé au moins un événement n'ont pas été décrits

- choisir un de ces objets X1
- pour tout événement X2 associé à X1
  empiler X2

fpour
mettre X1 dans TO
Tant que pile non vide faire
MX3 = sommet de la pile
éditer le cycle de X3
dépiler X3
pour toute c-opération déclenchée par X3
Si le c-objet X4 modifié par cette opération
∉ TO et si il lui est associé au moins un c-événement
  mettre X4 dans TO
  pour tout événement X5
    associé à X4
    empiler X5
  fpour
fsi
fpour
ftq

Le texte du programme correspondant est donné en annexe.

Nous étudions dans l'équipe REMORA, actuellement, un outil graphique (ROL 81) permettant à partir de ce programme d'obtenir la représentation graphique du sous-schéma de la dynamique telle qu'elle est présentée au chapitre 4.

VI-2.- Macrolangage de documentation

Le macrolangage de documentation permet au concepteur de se documenter sur un aspect particulier du schéma conceptuel du SI à une date de son choix.

Toutes les macroinstructions comprennent donc comme paramètre une date.

Un grand nombre de macroinstructions peuvent être proposées au concepteur, nous nous bornons ici à en donner quelques unes qui nous semblent les plus utiles.

Tout d'abord, nous proposons des macroinstructions permettant les éditions des deux sous-schémas à une date quelconque

- ESCHDONN A LA DATE DU :: :: emplacement de para-
- ESCHDYN A LA DATE DU :: mètre
- EXPLORDYN A LA DATE DU ::

Les programmes correspondants sont les précédents dans lesquels on sélectionne toutes les occurrences d'entité existantes à la date considérée c'est-à-dire telles que, si t est cette date

(date sup > t ou date sup = u) et (date création > t)

Nous proposons également des macroinstructions qui permettent de se documenter sur le sous-schéma de la dynamique.

- Conséguence directe d'un c-évènement :

CONSDIRECTE DE :: A LA DATE DU ::

Exemple

CONSDIRECTE DE UN EVENT 2 A LA DATE DU 811001

EV2 → OP1 OP2 OP3
↓ ↓ ↓
OB1 OB2 OB3

- Opérations portant sur un C-OBJET donné :

OPER MODIFIANT LE C-OBJET :: A LA DATE DU ::

Exemple

OPER MODIFIANT LE C-OBJET OB2 A LA DATE DU 811001

OP1, OP2,..... OP6, OP7

- Rôle d'un C-OBJET donné

ROLE DU C-OBJET :: A LA DATE DU ::

Ce programme donne successivement pour un c-objet les c-événements qui lui sont associés, les c-opérations qui le modifient, les prédicats, les conditions, les facteurs ou les textes dans lesquels il est utilisé à la date considérée.

Exemple

ROLE DU C-OBJET OB2 A LA DATE DU 811001
OB2 ← EV2, EV5
OB2 ← OP1, OP6, OP7
OB2 EST UTILISE DANS LE PREDICAT DE EV6
DANS LE TEXTE DE OP9
DANS LE FACTEUR DU DECL EV6 OP8

VII - GESTION DE L'EVOLUTION DE STRUCTURE DU SI

Rappelons que,dans le système réalisé, nous nous sommes limités à la prise en compte et à la représentation de l'évolution de la structure de l'organisation au niveau du schéma conceptuel.

Nous n'avons pas traité le problème général de la répercussion de ces modifications sur une structure interne quelconque du SI.

Nous nous sommes contentés de définir ces répercussions dans le cas où le SI serait,au niveau interne, une base de données relationnelle dans laquelle toute relation correspondrait à un schéma de relation du schéma conceptuel.

Nous donnons tout d'abord la liste des différents types d'évolution de structure qui peuvent se produire dans une organisation réelle et,pour chacun d'eux,sa conséquence au niveau du schéma conceptuel. Nous obtenons ainsi un ensemble de types de modifications élémentaires d'un schéma conceptuel de SI.

Nous proposons un macrolangage permettant une entrée desmodifications sous une forme aussi proche que possible de leur expres-

sion externe et signalant au concepteur toutes leurs conséquences sur le schéma conceptuel.

- Certaines de ces conséquences sont mises en place automatiquement par le système et sont alors signalées au concepteur.
- D'autres demandent son intervention et lui sont communiquées.

Enfin pour chaque type de modification du schéma conceptuel nous donnons sa conséquence sur la structure interne du SI, dans le cas cité ci-dessus.

VII-1.- Types de modifications de structure de l'organisation

Dans une organisation, les modifications de structure peuvent concerner

- soit la structure des éléments de l'organisation
- soit la structure de son fonctionnement, c'est-à-dire sa fonction DECL
- soit les deux sous-structures simultanément.

Pour des raisons méthodologiques, nous demandons au concepteur d'analyser toute modification de structure en deux étapes :

- étude de l'impact de la modification au niveau du sous-schéma des données
- étude de l'impact de la modification au niveau du sous-schéma de la dynamique

et de communiquer au système, dans cet ordre, les modifications déterminées au niveau de chacun de ces sous-schémas.

L'entrée des modifications dans cet ordre, permet au système d'assister le concepteur dans la recherche de l'impact d'une modification de la structure des données sur celle des traitements et de réduire dans la plupart des cas les modifications à entrer dans la seconde étape.

En effet, compte tenu des liens mis en place dans la métabase, entre C-OBJETS, C-OPERATIONS, C-EVENEMENTS, le système est capable de déterminer pour tout c-objet, son rôle et donc toutes les modifications induites sur le sous-schéma de la dynamique par sa suppression ou son remplacement par un autre.

VII-2.- Modifications concernant le sous-schéma des données

VII-2.1.- Modifications externes possibles

Les modifications de structure de données de l'organisation doivent être exprimées en termes de C-CLASSES. Une C-CLASSE représente une classe d'objets réels ou une classe d'associations entre classe d'objets, les types de modifications externes possibles sont :

ME1 : Création d'une nouvelle C-CLASSE
ME2 : Suppression d'une C-CLASSE
ME3 : Adjonction d'une propriété dans une C-CLASSE
ME4 : Suppression d'une propriété dans une C-CLASSE
ME5 : Modification de la dynamique d'une propriété

Nous entendons par là :

soit le fait qu'une propriété constante devienne variable

soit le fait qu'une propriété ayant le même comportement dynamique qu'une autre prenne un comportement dynamique propre ou identique à celui d'une troisième.

Exemple

Dans l'exemple de la gestion des prêts, nous avons supposé que le taux d'un prêt était constant au cours du temps. Il se peut qu'à la suite d'une décision quelconque, ce taux devienne variable, ce qui nécessite de remplacer le C-OBJET

COB OB7 (noprêt,nodemande,taux,dcprêt) par les C-OBJETS :

COB OB70 (noprêt,nodemande,dcprêt)
COB OB71 (noprêt,dmodtaux,taux)

VII-2.2.- Conséquences des modifications externes sur le schéma conceptuel

Nous exprimons les conséquences des modifications

externes sur le schéma conceptuel en termes de primitives de mise à jour de la métabase. Ces primitives correspondent toujours à un ensemble de créations de n-uplets dans des relations de la métabase puisque, par définition, toute trace de toute modification de structure intervenue au cours du temps est conservée dans la métabase.

Nous distinguons les conséquences vis à vis du sous-schéma des données et les conséquences vis à vis du sous-schéma de la dynamique.

VII-2.2.1.- Conséquences vis à vis du sous-schéma des données

ME1 : Création d'une nouvelle C-CLASSE

implique : MS1 : création d'une C-CLASSE
création d'un n-uplet de la relation CC1

k MS2 : création d'un C-OBJET
création d'un n-uplet de la relation COB1
plusieurs créations de n-uplets des relations CONSTOB, COMPOB et COMPOSCLE

(k : nombre de C-OBJETS de la C-CLASSE)

ME2 : Suppression d'une C-CLASSE

La suppression d'une C-CLASSE implique la suppression de toutes les associations dans lesquelles la classe réelle représentée par la C-CLASSE intervenait.

Les associations sont représentées soit
- par des C-CLASSES
- par la présence du constituant clé d'une C-CLASSE dans un C-OBJET d'une autre.

ME2 implique : - MS4 : suppression d'une C-CLASSE de la métabase
création d'un n-uplet de la relation CC2

- Pour tous les C-OBJETS de la C-CLASSE
  MS5 : suppression d'un C-OBJET
  création d'un n-uplet de la relation COB2
  fpour
- Pour tous les C-OBJETS dans lesquels la clé de la C-CLASSE figure
  Si cette clé est le seul constituant du C-OBJET n'appartenant pas à sa clé
  alors MS5
  sinon MS6 : "suppression" d'un constituant dans un C-OBJET.
  fsi
  fpour
- Pour toutes les C-CLASSES ayant dans la composition de leur clé, la clé de la C-CLASSE supprimée
  ME2 : suppression d'une C-CLASSE

Par exemple, dans la gestion des prêts, une modification importante des modalités d'attribution des prêts, peut entraîner la suppression de la C-CLASSE demande de prêt. Cette suppression entraîne bien évidemment la suppression du constituant n°demande dans le c-objet prêt.

ME3 : Adjonction d'une propriété dans une C-CLASSE

Cette modification peut se produire lorsqu'une nouvelle propriété d'une classe réelle ou d'une association réelle doit être représentée dans le SI.
Ses conséquences dépendent du comportement dynamique de cette nouvelle propriété.

ME3 implique : - Si la propriété a un comportement dynamique propre
MS2 : création d'un C-OBJET

- Si la propriété a un comportement dynamique identique à d'autres propriétés de la c-classe.

MS3 : "adjonction" d'un constituant dans un C-OBJET.

ME4 : Suppression d'une propriété dans une C-CLASSE

ME4 implique : - Si le constituant représentant la propriété est le seul constituant n'appartenant pas à la clé du C-OBJET où il figure.

alors MS5 : Suppression d'un C-OBJET

sinon

MS6 : "Suppression d'un constituant dans un C-OBJET".

ME5 : Modification de la dynamique d'une propriété

La conséquence de ME5 est différente suivant que la propriété prend un comportement propre ou identique à celui d'autres propriétés de la classe.

ME5 implique : toujours - MS6 "Suppression d'un constituant dans un C-OBJET"

a) si la propriété prend un comportement propre

- MS2 création d'un C-OBJET

b) si la propriété prend un comportement identique à d'autres

- MS3 adjonction d'un constituant dans un C-OBJET.

MS3 et MS6 respectivement "adjonction" et "suppression" d'un constituant dans un C-OBJET correspondent à :

- MS5 : Suppression du C-OBJET pour lequel la modification est demandée

- MS2 : création d'un C-OBJET à partir du précédent dans lequel un constituant est ajouté ou supprimé.

VII-2.2.- Conséquences vis à vis du sous-schéma de la dynamique

Toute suppression de C-OBJET a une conséquence immédiate vis à vis du sous-schéma de la dynamique :

- tous les rôles joués par le C-OBJET dans la fonction DECL du SI doivent être supprimés.

Dans le cas où cette suppression ne correspond pas à une suppression réelle d'un aspect de la réalité mais à une modification, c'est-à-dire quand elle est la conséquence d'une modification de type MS3 ou MS6, il est nécessaire de signaler au concepteur tous les rôles joués par le C-OBJET supprimé et de lui demander les nouveaux rôles du C-OBJET créé.

MS5 : implique en terme de modifications externes sur le sous schéma de la dynamique

- la suppression de tous les cycles associés à ce C-OBJET soit n (ME13) (n nombre de cycles concernés)
- la suppression des opérations qui modifient ce C-OBJET soit k (ME10) (k nombre d'opérations concernées).

De plus, pour tous les textes où le C-OBJET apparaît, il est demandé au concepteur de nouveaux textes d'où par la suite un ensemble de modifications externes de types.

ME6, ME7, ME8 ou ME9

MS3 et MS6 : impliquent, en plus des conséquences de MS5, la demande au concepteur de la dynamique du C-OBJET remplaçant celui supprimé c'est-à-dire la création de nouveaux cycles : ME12.

VII-3.- Modifications concernant le sous-schéma de la dynamique

VII-3.1.- Modifications externes possibles

Les modifications externes de la structure de fonctionnement d'une organisation concernent ses règles de gestion.

- Une règle de gestion peut être modifiée dans ses modalités d'application sans qu'il y ait pour autant une modification de la structure des données.
- Une règle de gestion peut être supprimée,
  - soit sans qu'il y ait modification de la structure des données,
  - soit à la suite d'une modification de la structure des données.
- Une règle de gestion peut être créée à la suite
  - soit d'une création de nouvelles structures de données,
  - soit en remplacement d'une autre.
- Une règle de gestion est représentée dans le schéma conceptuel par un cycle : c'est-à-dire un C-EVENEMENT associé à un C-OBJET et un ensemble de C-DECLENCHE.
- Nous exprimons donc les différents types de modifications externes vis à vis d'un cycle.
- Modifications possibles d'un cycle :

- ME6 : Modification d'un prédicat associé à un C-EVENEMENT
- ME7 : Modification d'une condition de déclenchement
- ME8 : Modification d'un facteur de déclenchement
- ME9 : Modification du texte d'une C-OPERATION
- ME10 : Suppression d'une C-OPERATION dans un cycle
- ME11 : Adjonction d'une C-OPERATION dans un cycle
- ME12 : Création d'un cycle
- ME13 : Suppression d'un cycle.

VII-3.2.- Conséquences des modifications de la structure de fonctionnement sur le schéma conceptuel

Pour chaque type de modification externe, nous indiquons sa conséquence sur le schéma conceptuel.

ME6 : Modification d'un prédicat associé à un C-EVENEMENT
implique - MS7 : création d'un prédicat
création d'un n-uplet de CEV2.

ME7 : Modification d'une condition de déclenchement
implique - MS8 : création d'une condition
création d'un n-uplet de CDEC2.

ME8 : Modification d'un facteur de déclenchement
implique - MS9 : création d'un facteur
création d'un n-uplet de CDEC3.

ME9 : Modification d'un texte d'une C-OPERATION
implique - MS10 : création d'un texte
création d'un n-uplet de COP2.

ME12 : Création d'un cycle
implique - MS11 : création d'un C-EVENEMENT
(n-uplet de la relation CEV1)
- MS 7 : création d'un prédicat
pour toute C-OPERATION du cycle
- MS12 : création d'un C-DECLENCHE
(n-uplet de la relation CDEC1)
- MS8
- MS9
Si la C-OPERATION n'existe pas (dans COP1)
- MS13 : création d'une C-OPERATION
fpour (n-uplet de la relation COP1)

ME13 : Suppression d'un cycle
implique - MS14 : suppression d'un C-EVENEMENT
(création d'un n-uplet de la relation CEV3)
Pour toutes les C-OPERATIONS du cycle
- MS15 : suppression d'un C-DECLENCHE
(création d'un n-uplet de la relation CDEC4)
Si la C-OPERATION appartient à aucun autre cycle
- MS16 : suppression d'une C-OPERATION
(création d'un n-uplet de la relation COP3)
fsi
fpour

ME10 : Suppression d'une C-OPERATION dans un cycle
implique
si la C-OPERATION n'apparaît dans aucun autre cycle
- MS16 : suppression d'une C-OPERATION
fsi
- MS15 : suppression d'un C-DECLENCHE
si la C-OPERATION était la seule du cycle
- MS14 : suppression d'un C-EVENEMENT

ME11 : Adjonction d'une C-OPERATION dans un cycle
implique - MS12 : création d'un C-DECLENCHE
- MS8 : création d'une condition de déclenchement
- MS9 : création d'un facteur de déclenchement
si la C-OPERATION n'appartient pas à d'autres cycles
- MS13 : création d'une C-OPERATION

VII-4.- Macrolangage de définition d'évolution de structure

Le macrolangage de définition d'évolution de structure comprend autant de macroinstructions que de types de modifications externes soit 13 macroinstructions.

La fréquence d'utilisation de ces macroinstructions étant faible, nous avons créé une macroinstruction supplémentaire appelée EVOLUTION DE STRUCTURE, qui donne au concepteur la liste des noms des macroinstructions qu'il peut utiliser avec leur rôle.

Chacune de ces macroinstructions se déroule sous la forme d'un dialogue.

Le système :

- demande au concepteur progressivement les différentes caractéristiques des éléments à créer, à supprimer ou à modifier,
- contrôle d'un point de vue syntaxique chacune des réponses fournies par le concepteur,
- signale au concepteur les conséquences de la modification demandée,
- indique au concepteur les autres mises à jour du schéma conceptuel qu'il devra effectuer par la suite,
- demande au concepteur si la modification effectuée est la dernière.

En cas de réponse positive, le système lance les programmes de contrôles globaux d'un schéma conceptuel.

Exemple

Reprenons le cas de la gestion des prêts, le taux d'un prêt devient une propriété variable au cours du temps, initialisée par le SI à la création d'un prêt et susceptible de modifications externes par la suite.

Nous avions

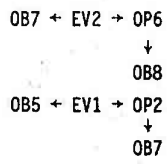
COB PERM OB7 (noprêt, nodemande, taux, dcprêt)
COB VAR OB8 (noprêt, drevision, montanstremb, nbmens, datedebutremb)

Le concepteur demande l'exécution de MODDYNPROP (correspond à ME5).

Le dialogue suivant s'établit :

nom de la C-CLASSE ?
PRET
nom du constituant ?
taux
Voici les C-OBJETS de la C-CLASSE
COB PERM OB7 (noprêt,nodemande,taux,dcprêt)
COB VAR OB8 (noprêt,drevision,montantotremb,nbmens,datedebutremb)
COB VAR OB9 (noprêt,dateéchéance)
le constituant taux doit-il appartenir à l'un d'entre eux O/N ?
N
nom du c-objet à créer ?
OB13
nom du constituant date ?
datemodtaux
la nouvelle version de la c-classe prêt est la suivante :
COB PERM OB7-1(noprêt,nodemande,dcprêt)
COB VAR OB8 (noprêt,drevision,montanttoremb,nbmens,datedebutremb)
COB VAR OB9 (noprêt,dateéchéance)
COB VAR OB13 (noprêt,datemodtaux,taux)

Dans le schéma de la dynamique vous aviez :



Le C-EVENEMENT EV2 est supprimé faut-il le remplacer par un C-EVENEMENT identique associé à OB7-1 et appelé EV2-1 O/N ?
O

Vous aurez à donner le prédicat de EV2-1
La C-OPERATION OP2 est supprimée,faut-il la remplacer par une C-OPERATION modifiant OB7-1 appelée OP2-1 O/N ?
O
Vous aurez à donner le texte de OP2-1
le C-OBJET OB7 intervient dans les textes suivants :
- texte de OP6
- texte de F1

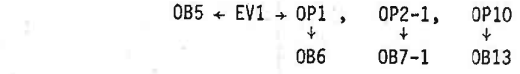
Vous devez les modifier.

Il vous reste maintenant à entrer la dynamique du C-OBJET OB13

Le concepteur lance alors la macroinstruction suivante :
- ADJOPE (adjonction d'une opération dans un cycle correspond à ME11

Le dialogue engagé est le suivant :

nom du C-EVENEMENT ?
EV1
nom de la C-OPERATION à ajouter ?
OP10
nom du C-OBJET modifié ?
OB13
condition de déclenchement ?
TOUJOURS
multiplicité de déclenchement ?
UN
facteur de déclenchement ?
FAC10
le nouveau cycle de EV1 est le suivant :



Cette adjonction de OP10 correspond au calcul de la valeur initiale du taux du prêt lors de l'établissement du prêt.

Lors d'une modification du taux signalée au SI, il est nécessaire de recalculer le montant total à rembourser c'est-à-dire de créer une nouvelle occurrence de prêt-remboursement.

D'où la création d'un nouveau cycle

Le concepteur lance la macroinstruction CRECYCLE (correspond à ME12)

Le dialogue engagé est le même que celui de la macroinstruction CDYN décrite dans la fonction aide à la conception d'un SC.



A la fin du dialogue, le système demande :

avez-vous terminé vos modifications du schéma conceptuel O/N ?

O

EXPLORDYN donne alors :

OB5 ← EV1 → OP1, OP2-2 , OP10
↓ ↓ ↓
OB6 OB7-1 OB13

OB7-1 ← EV2-1 → OP6
↓
OB8

OB13 ← EV5 → OP11
↓
OB8

le reste est inchangé.

Remarques

Dans le cas où le concepteur fait une erreur, non détectée par le système, en entrant une modification de structure, il peut toujours la corriger en utilisant les primitives d'aide à la mise au point d'un schéma conceptuel : SCOBJET et MCYCLE définies au paragraphe 4.2.2.

Pour aider de façon encore plus efficace le concepteur dans sa mise à jour du schéma conceptuel, il serait utile que le système

gère une liste de toutes les modifications que le concepteur doit effectuer en conséquence des autres ; ainsi le concepteur pourrait à tout moment savoir ce qui lui reste à faire et le système pourrait lui signaler des oublis éventuels lorsque le concepteur donnerait le signal de fin de modification.

Dans l'exemple précédent, la liste de modifications contiendrait à la fin de l'appel de MODDYNPROP

- Texte de prédicat de EV2-1
- Texte de OP2-1
- Nouveau texte de OP6
- Nouveau texte de F1
- Dynamique de OB10

Après l'exécution de CRECYCLE la liste contiendrait

- Texte de prédicat de EV2-1
  ¦
  ¦
- Nouveau texte de F1
- Texte de OP11

Au fur et à mesure de l'entrée des textes par le concepteur, la liste serait mise à jour.

Dans le cadre d'une étude plus générale concernant l'évolution de structure des bases de données et des SI, que nous pensons entreprendre à la suite de ce travail, nous envisageons la gestion automatique d'une telle liste par l'outil d'aide à la conception.

VII-5.- Répercussion des modifications du schéma conceptuel sur la structure interne du SI dans le cas de l'implantation sous la forme d'une base de données relationnelle

Rappelons que le choix de conserver dans le SI tout état pris par l'organisation au cours du temps a pour conséquence que toute modification de structure dans l'organisation se traduit par la création de nouvelles structures dans le SI.

Dans le cas où la structure interne du SI est une base de données relationnelle dans laquelle toute relation correspond à un schéma de relation du schéma conceptuel, complétée par une biblio-

thèque de procédures représentant les valeurs des textes de la métastructure (condition, textes des C-OPERATIONS, ect...) les modifications de structure de l'organisation ont :

- soit aucune conséquence, lorsqu'il s'agit de suppression de structures,
- soit pour effet d'ajouter de nouvelles relations dans le SI,
- soit pour effet d'ajouter de nouvelles procédures dans la bibliothèque.

Soit :

Modification du schéma conceptuel	Modification de la structure interne
MS1 : création d'une C-CLASSE	→ aucune conséquence
MS2 : création d'un C-OBJET	→ création d'une relation
MS4 : suppression d'une C-CLASSE	→ aucune conséquence
MS5 : suppression d'un C-OBJET	→ aucune conséquence
MS6 : suppression d'un constituant dans un C-OBJET	→ création d'une relation à partir de la relation représentant le C-OBJET modifié
MS3 : adjonction d'un constituant dans un C-OBJET	
MS7 : création d'un prédicat	→ création d'une procédure
MS8 : création d'une condition	→ création d'une procédure
MS9 : création d'un facteur	→ création d'une procédure
MS10 : création d'un texte	→ création d'une procédure
MS11 : création d'un C-EVENEMENT	→ création d'une relation
MS12 : création d'un C-DECLENCHE	→ création d'une relation
MS13 : création d'une C-OPERATION	→ création d'une relation
MS14 : suppression d'un C-EVENEMENT	→ aucune conséquence
MS15 : suppression d'un C-DECLENCHE	→ aucune conséquence
MS16 : suppression d'une C-OPERATION	→ aucune conséquence

Le seul cas complexe correspond à la création d'une relation en remplacement d'une autre.

Le système doit trouver la valeur initiale de la nouvelle relation à partir de la précédente.

Dans l'exemple précédent de la modification de la dynamique du taux des prêts

le C-OBJET OB7 (noprêt,nodemande, taux,dcprêt)

est remplacé par les C-OBJETS

OB7-1 (noprêt,nodemande,dcprêt)
OB13 (noprêt,dmtaux,taux)

Il faut donc ajouter au SI deux relations de schéma OB7-1 et OB13 et initialiser ces relations à partir de la relation OB7, c'est-à-dire pour toutn-uplet de OB7 créer un n-upletde OB7-1 et un n-upletde OB13.

Cette création de nouvelles relations permet à partir de cette date d'utiliser les textes des nouvelles C-OPERATIONS créées qui ne font plus référence à OB7 mais à OB7-1 et à OB13.

En conclusion, nous pouvons dire que la restructuration d'un SI consécutive à des modifications de la structure de l'organisation réelle représentée dans le SI est très simple s'il existe une correspondance biunivoque entre les structures définies dans le schéma conceptuel et les structures internes du SI.

Si cette correspondance est complexe, c'est-à-dire si pour des raisons d'efficacité on a, lors du choix de la structure interne du SI, représenté plusieurs C-OBJETS par une même structure de donnéees, ou bien fusionné plusieurs C-OPERATIONS d'un cycle dans une même procédure, alors la détermination des répercussions des modifications externes sur la structure interne est beaucoup plus difficile.

Elle nécessite de conserver dans la métabase, toutes les correspondances entre structure conceptuelle et structure interne et d'incorporer au système des fonctions permettant d'exploiter cette correspondance.

Conclusion concernant la gestion de l'évolution de structure

L'outil réalisé, associé au modèle conceptuel de SI que nous proposons, apporte une solution originale aux problèmes de maintenance des SI, dont l'origine est une modification du monde réel représenté.

Les modifications de structure de l'organisation réelle n'entraînent pas de restructuration complexe des informations du SI, mais uniquement la création de nouvelles structures utilisées à partir d'une certaine date, à la place d'autres qui, à partir de cette même date, ne sont plus utilisées que lors d'interrogations du SI, concernant une période passée.

L'outil mis en place permet au concepteur, par l'intermédiaire de la métabase, de ne rechercher les conséquences des modifications externes qu'au niveau du schéma conceptuel et de ne pas se préoccuper de leurs répercussions sur la structure interne du SI comme il est nécessaire de le faire avec les SGBD classiques.

Enfin l'outil aide au maximum le concepteur dans ce travail, en lui permettant d'entrer les modifications sous une forme assez proche de leur forme externe et en lui signalant toutes les conséquences directes et indirectes des modifications introduites.

CONCLUSION

L'outil que nous venons de présenter est un complément indispensable au modèle conceptuel de SI que nous proposons.

Il permet au concepteur de SI de se dégager du travail fastidieux de tenue à jour de la documentation concernant le SI en cours de conception ou en cours de restructuration.

Les contrôles effectués par le système permettent de détecter certaines erreurs d'analyse ou certains oublis, que le concepteur mettrait plus longtemps à découvrir dans une conception non assistée.

Mais l'assistance la plus efficace concerne la prise en compte de l'évolution de structure de l'organisation.

En effet, c'est à ce niveau, que les risques d'erreurs ou d'oublis sont les plus grands, car la modification peut intervenir longtemps après la conception initiale du SI et alors être à la charge d'un concepteur différent de celui ayant créé le SI.

Cependant, nous sommes tout à fait conscient des limites de cet outil sur deux plans en particulier :

- le plan de la méthode de conception d'un schéma conceptuel. En effet, l'aide apportée à ce niveau se limite à conseiller de commencer par la définition de la structure des données et n'apporte pas de solution au problème du passage de l'analyse du monde réel, à sa représentation à l'aide des concepts du modèle.
- le plan de la définition des constituants de type texte. Il serait intéressant, pour le concepteur, de pouvoir, dès la conception du schéma conceptuel, exprimer les constituants de type texte dans un langage de spécification permettant le contrôle de leur validité et d'avoir un outil complémentaire permettant, avant la création du SI réel de simuler son fonctionnement à partir de ces textes.

CONCLUSION

Le modèle et l'outil que nous venons de présenter s'intègrent dans l'ensemble des résultats du projet REMORA qui fournit un environnement pour concevoir et utiliser des systèmes d'information dans les organisations.

Ceci explique que certains problèmes fondamentaux concernant la conception des SI, comme les problèmes méthodologiques liés à l'usage du modèle ou les problèmes d'implantation physique du SI, ne sont pas traités dans ce travail.

Le bilan que nous effectuons ici porte donc uniquement sur le modèle et l'outil proposés. Il comporte une tentative d'évaluation de notre proposition et l'indication de quelques directions de recherche permettant d'envisager une amélioration ou une extension de la solution proposée.

L'évaluation d'une solution à un problème aussi complexe que la modèlisation des organisations est très difficile à effectuer compte tenu du grand nombre de critères que l'on pourrait utiliser.

Nous nous contentons donc ici d'effectuer cette évaluation suivant trois points :

- évaluation par rapport aux objectifs que nous nous étions assignés,
- évaluation par rapport aux critères proposés dans le rapport ISO (ISO 81),
- évaluation par comparaison avec d'autres approches voisines.

Evaluation du modèle par rapport à ses objectifs

Au chapitre 1, nous nous étions donné comme objectif, de définir un modèle formel, "orienté conception", permettant de représenter, de manière intégrée et par rapport au temps, l'ensemble des aspects statiques et dynamiques de l'organisation et conduisant à des schémas conceptuels canoniques.

Résumons, pour chacune de ces caractéristiques, la solution adoptée.

- La représentation des aspects statiques est assurée par les concepts de C-CLASSE et de C-OBJET. Une C-CLASSE est la représentation d'une classe d'objets ou d'associations d'objets du monde réel. Un C-OBJET est la représentation d'un aspect temporel d'une classe, c'est-à-dire la représentation d'un ensemble de propriétés de la classe réelle, ayant le même comportement dynamique.

- La représentation des aspects dynamiques est assurée par les concepts de C-OPERATION, C-EVENEMENT et C-DECLENCHE et par la métastructure du SI.

Une C-OPERATION est la représentation d'une classe d'opérations élémentaires du monde réel, un C-EVENEMENT est la représentation d'une classe d'événements élémentaires du monde réel. Un C-DECLENCHE est la représentation d'une association DECLENCHE entre une classe d'événements élémentaires et une classe d'opérations élémentaires. La métastructure permet la représentation des lois de fonctionnement de l'organisation, exprimées en termes de classes de phénomènes élémentaires.

Rappelons que les concepts proposés ne permettent pas de rendre compte de tous les aspects statiques et dynamiques puisque certaines caractéristiques du monde réel doivent être exprimées sous forme de contraintes d'intégrité. Notons cependant que le nombre de contraintes à exprimer est beaucoup plus faible dans notre approche que dans celles utilisant uniquement un modèle conceptuel de données.

- L'intégration des représentations des aspects statiques et dynamiques est assurée par les liens de causalité existant entre les concepts associés aux deux aspects

  - Une C-OPERATION "modifie" un C-OBJET c'est-à-dire que toutes les opérations d'une classe d'opérations élémentaires modifient selon le même aspect des objets appartenant à une même classe.

  - Un C-EVENEMENT "constate le changement d'état" d'un C-OBJET c'est-à-dire que tous les événements d'une classe d'événements élémentaires correspondent à des changements d'état élémentaires de même type subis par des objets appartenant à une même classe.

L'intégration est encore accentuée par l'utilisation d'un même formalisme pour exprimer les C-OBJETS, les C-OPERATIONS, les C-EVENEMENTS et les C-DECLENCHES : la relation permanente en troisième forme normale.

- Le caractère "orientation conception" du modèle provient de la normalisation incluse dans le modèle qui conduit à représenter l'organisation et ses lois de fonctionnement en termes de classes de phénomènes élémentaires et d'associations entre ces classes, donc d'une manière totalement indépendante des modalités d'application pratique de ces lois à un instant donné.

- Le caractère formel du modèle est assuré par l'usage du formalisme du modèle relationnel dans lequel nous avons introduit le temps par l'intermédiaire des concepts de dépendance fonctionnelle permanente et de relation permanente. De plus, la formalisation effectuée à la fin du chapitre 3 permet de définir les textes des opérations, les facteurs de déclenchement, les conditions de déclenchement et les prédicats des C-EVENEMENTS comme des fonctions ayant un certain type de profil. Peut être cette formalisation gagnerait-elle à être davantage exploitée dans le développement de ce travail.

- La prise en compte du temps est assurée comme nous venons de le dire, par son introduction dans le modèle relationnel. Elle conduit à la construction de SI historiques, dans lesquels l'évolution complète de l'organisation peut être représentée : l'évolution de sa structure et de ses lois de fonctionnement aussi bien que l'évolution de ses constituants.

- Les schémas conceptuels de SI construits à l'aide de ce modèle possèdent un certain nombre de qualités :

- L'utilisation de la 3ème forme normale de relation garantit, comme dans les schémas conceptuels de données, l'élimination d'une certaine redondance et de certains inconvénients de manipulation.

- La forme permanente élimine la redondance dans la représentation de l'histoire de l'organisation : en effet, toute valeur prise par une propriété de phénomène réel au cours du temps est représentée une fois et une seule dans le SI.

- L'élémentarité imposée pour la représentation des opérations permet également d'éviter la redondance dans l'expression de leurs textes, redondance qui pourrait se produire si le choix de la taille des opérations était laissée au concepteur.

- Enfin la collection de règles de validité associées au modèle et présentées au chapitre 4 permet de vérifier la complètude et la cohérence d'un schéma conceptuel.

Evaluation du modèle et de l'outil associé, par rapport aux critères du rapport ISO (ISO 81)

Le groupe de travail ayant élaboré le rapport ISO définit quatre exigences auxquelles doit satisfaire, à son avis, toute proposition concernant l'expression d'un schéma conceptuel de SI.

Ces exigences sont :

1) Fournir des moyens de modèlisation qui prennent en compte à la fois les aspects statiques et dynamiques de l'univers du discours.

2) Fournir un langage pour communiquer le schéma conceptuel à un ordinateur et plusieurs langages pour communiquer le schéma conceptuel à des utilisateurs différents, ces langages pouvant inclure des formalismes graphiques.

3) Fournir des possibilités pour adapter facilement le schéma conceptuel aux changements se produisant dans la réalité.

4) Etre facile à comprendre ou à utiliser.

Le modèle proposé, complété par le système d'aide à la conception d'un schéma conceptuel de SI, satisfait aux trois premières exigences :

1) Le modèle permet la modèlisation des aspects statiques et dynamiques de l'organisation.

2) Le système inclut un langage conversationnel permettant de lui communiquer les éléments constitutifs d'un schéma conceptuel de SI. Les outils de documentation du système permettent aux concepteurs du SI ou aux utilisateurs futurs de prendre connaissance de tout ou partie de la description conceptuelle. Enfin les représentations graphiques, associées aux deux sous-schémas, présentées au chapitre 4, sont des moyens de visualisation pratiques d'un schéma conceptuel de SI. Il serait intéressant de les incorporer au système et ce travail est à l'étude dans l'équipe REMORA.

3) La prise en compte des évolutions de structure se produisant dans l'organisation est rendue possible au niveau de la modèlisation par l'introduction du temps dans la méta-structure. Elle est facilitée par le sous-système de gestion de l'évolution de structure de l'outil réalisé, qui permet une expression des changements de structure sous leur forme externe et déduit toutes les conséquences de ces changements sur le schéma conceptuel.

Le quatrième critère est plus difficile à évaluer. Il nous semble que le nombre réduit de concepts de base : événement, opération et objet, ainsi que l'approche causale de la dynamique, sont des facteurs de simplicité. L'expérience

pédagogique que nous pouvons avoir de l'enseignement du modèle nous a montré que les concepts sont facilement compris, mais que leur utilisation est plus délicate. En effet, elle nécessite un très gros effort d'abstraction que seule l'expérience du modèle permet d'effectuer rapidement et facilement.

Evaluation du modèle par comparaison avec d'autres approches

Comme nous l'avons indiqué au chapitre 1, ce modèle a été développé parallèlement à un certain nombre d'autres modèles ayant le même objectif : modéliser simultanément les aspects statiques et dynamiques d'un SI. Ces modèles ont été développés par des équipes de recherche travaillant à l'origine dans le domaine des bases de données et ayant fréquemment l'occasion d'échanger et de confronter leurs résultats.

Cette situation explique l'existence d'un certain nombre de points communs entre ces modèles et celui que nous proposons.

Nous mettons en évidence ici quelques points communs et quelques différences fondamentales entre ces approches et notre proposition.

Les modèles auxquels nous nous référons sont ceux présentés au paragraphe III-2.3. du chapitre 1.

Points communs entre les modèles dynamiques de SI et notre proposition

- Utilisation du concept d'événement avec une signification commune : changement d'état de l'organisation.
- Expression causale de la dynamique : les événements sont des stimuli qui, directement ou indirectement, provoquent l'exécution de traitements.
- Utilisation d'un modèle conceptuel de données qui est souvent de type entité-relation.

Différences entre notre proposition et ces approches

- Existence d'une normalisation pour la représentation des aspects dynamiques

Dans toutes les approches que nous avons étudiées, aucune règle n'est imposée pour la définition des types d'événements et des types de traitements.

Cette absence de normalisation s'explique par le fait que dans ces approches, le niveau de formalisme utilisé pour décrire les aspects dynamiques, est moins élevé que celui utilisé pour décrire les aspects statiques.

La normalisation que nous imposons présente des avantages et des inconvénients.

Pour ce qui est des avantages, rappelons les qualités qu'elle permet d'assurer ou de vérifier sur un schéma conceptuel de SI.

Un autre avantage, très important à notre avis, provient de l'élémentarité de représentation des traitements. Cette élémentarité permet, dans la deuxième étape de la conception d'un SI qui correspond au choix de sa structure interne, de regrouper les opérations élémentaires en unités physiques de traitement selon de multiples façons.

Pour ce qui est des inconvénients, notons l'importance du nombre d'éléments à décrire dans un schéma conceptuel.

- Inexistence de la notion de synchronisation dans notre approche

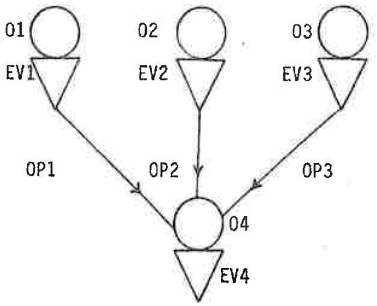
Parmi les modèles dynamiques de SI, présentés au chapitre I, le modèle de MERISE (MER 79) et celui inclus dans le langage DSL (BOD 79) apparaissent comme les plus proches de notre proposition.

La différence essentielle, entre ces deux approches et la nôtre, réside dans l'expression de la dynamique.

Dans notre approche ce sont les événements qui déclenchent directement l'exécution des opérations. Dans les deux modèles cités ci-dessus, l'origine du déclenchement des opérations est: soit un événement, soit une combinaison d'événements, appelée SYNCHRONISATION dans MERISE et POINT DE SYNCHRONISATION dans DSL.

Les situations dynamiques réelles représentées par cette notion de SYNCHRONISATION sont représentées dans notre modèlisation par l'intermédiaire de C-OBJETS particuliers. Ces C-OBJETS représentent des états d'objets réels qui peuvent être modifiés par des C-OPERATIONS déclenchées par des C-EVENEMENTS différents. Certains changements d'état de ces C-OBJETS correspondent à des C-EVENEMENTS.

Par exemple les cycles dynamiques suivants :



permettent de représenter une situation de synchronisation entre des événements de type EV1, EV2 et EV3.

Cette absence de la notion de synchronisation dans notre approche présente un avantage et un inconvénient.

Pour ce qui est de l'avantage, l'expérience du modèle nous a montré que l'absence de ce concept,oblige le concepteur, dans chaque situation dynamique réelle,à décrire, à rechercher de façon précise le changement d'état qui est à l'origine de l'exécution d'une opération.

Nous nous sommes aperçus alors que dans la plupart des cas le C-OBJET introduit correspondait à un état particulier effectif d'un objet réel qu'il était intéressant de représenter dans le SI.

Pour ce qui est de l'inconvénient,il peut paraître regrettable que ces situations de synchronisation ne soient pas mises en évidence dans le schéma conceptuel. En effet pour les retrouver il est nécessaire d'analyser certains textes d'opérations et de prédicats.

Directions dans lesquelles il serait intéressant de faire évoluer le modèle

L'évaluation que nous venons d'effectuer a mis en évidence certaines faiblesses de l'approche proposée qui suggèrent des directions de recherche dans lesquelles nous pourrions nous engager pour l'améliorer.

Nous retenons pour l'instant :

- L'augmentation du champ couvert par la modèlisation

Nous entendons par là l'adjonction de nouveaux concepts ou de nouvelles caractéristiques à des concepts existant dans le modèle, permettant une intégration dans le schéma conceptuel de certaines contraintes d'intégrité.

- La recherche d'un nouveau formalisme d'expression du modèle

Sur la base de la formalisation effectuée à la fin du chapitre 1, nous pensons utiliser le formalisme des types abstraits algébriques et des systèmes formels (FIN 79) pour définir tout d'abord les concepts du modèle puis,ensuite,pour exprimer le schéma conceptuel.

Cette recherche rejoint, d'une part les préoccupations des autres membres de l'équipe REMORA qui travaillent sur le langage de spécification de SI,associé au modèle et d'autre part les préoccupations des équipes de recherche travaillant dans le domaine de la définition de langage de spécification en général.

Utilisation du modèle et de l'outil en génie logiciel

L'approche causale de la modèlisation de la dynamique que nous avons choisie est très générale et nous permet donc d'envisager d'utiliser le modèle pour la conception d'autres types de système que les systèmes d'information.

Nous pensons en particulier à la spécification de logiciel.

Un exemple de ce type d'application a déjà été effectué dans le projet REMORA par LEIFERT (LEI 80) pour spécifier le processeur de la dynamique d'un SGSI.

D'autre part l'outil d'aide à la conception d'un SI remplit partiellement un certain nombre de fonctions assurées par des outils développés en génie logiciel : gestion d'une bibliothèque de textes, documentation sur un ensemble de programmes, aide à la maintenance. Nous pensons donc qu'il serait intéressant d'étendre cet outil dans l'esprit des recherches effectuées dans le domaine du génie logiciel.

Ces quelques exemples de direction de recherches, dans lesquelles nous pourrions poursuivre ce travail, montrent les interrelations qui existent entre les différents domaines de recherche de l'informatique et permettent d'envisager, pour les années à venir, de nouvelles solutions, pour le problème de la conception des systèmes d'information dans les organisations, bénéficiant des résultats de ces différents domaines.

ANNEXE I

EXEMPLE DE SCHEMA CONCEPTUEL DE SI ET D'UTILISATION DE L'OUTIL

INTRODUCTION

L'exemple que nous traitons ici nous a été fourni par Monsieur PELLAUMAIL d'IBM France par l'intermédiaire de Monsieur TARDIEU du CETE d'AIX EN PROVENCE.

Il s'agit d'une étude de cas tirée de l'analyse d'une entreprise réelle. (*)

Nous l'avons choisie pour illustrer ce travail, car cette étude a été communiquée à plusieurs équipes de recherche travaillant sur la conception des systèmes d'information. Nous espérons ainsi que la connaissance commune du même problème facilitera les discussions et les échanges au niveau des solutions de représentation proposées à l'aide des différents modèles, outils ou langages développés dans les équipes.

Nous présentons successivement :

- Une définition de la partie de l'organisation à laquelle nous nous intéressons

Cette définition est très largement extraite de celle qui nous a été fournie, nous avons reformulé certains paragraphes dans un souci de clarté et de concision. Elle constitue l'énoncé informel du problème de conception à résoudre.

- Le schéma conceptuel du SI correspondant

Nous présentons ce schéma tel qu'il est obtenu par les outils de documentation du système d'aide à la conception.

Nous donnons successivement :

- la métastructure
- le sous-schéma des données
- le sous-schéma de la dynamique
- les contraintes d'intégrité.

Nous complétons cette présentation par la spécification des constituants de type TEXTE.

(*) IBM "Etude de cas HYPOM" 1979

- Des exemples d'utilisation possibles du SI implanté sous la forme d'une base de données à partir de ce schéma conceptuel.

Ces exemples montrent des types d'informations que l'on peut obtenir facilement à partir de ce SI et qui sont difficiles ou même impossibles à retrouver dans les SI construits actuellement dans les organisations. Ce sont des interrogations concernant :

- l'état passé d'objets ou de règles de gestion
- l'origine de certains changements d'état
- l'histoire d'une classe d'objets

qui permettent en particulier de vérifier l'adéquation entre le fonctionnement réel de l'organisation et les objectifs qu'elle s'est fixés.

I - DEFINITION DE L'ORGANISATION REELLE A MODELISER

I-1.- Présentation de l'organisation et de ses objectifs

I-1.1.- L'organisation

La Société HYPOM a comme activité principale la construction et la distribution de véhicules "hypo-mobiles".

Afin de parvenir à ces deux objectifs, elle a implanté un réseau de vente et d'après-vente composé de :

- succursales
- concessionnaires
- agents

Les succursales appartiennent à la Société mère et sont gérées directement par elle.

Les concessionnaires sont en général des Sociétés indépendantes, dont les capitaux d'origine sont privés. Ils travaillent exclusivement pour la Société HYPOM. Ils servent, comme les succursales, de distributeurs de pièces détachées et de véhicules neufs.

Enfin, les agents sont des garages indépendants qui peuvent vendre ou réparer des voitures de la marque HYPOM.

Ils s'approvisionnent en pièces détachées ou en véhicules neufs auprès des succursales ou des concessionnaires.

Cette étude concerne les relations entre Siège et Concessionnaires pour les activités suivantes :

1 - Suivi des commandes (au Siège)

2 - Réseau de ventes (Concessionnaire). Il y a 40 concessionnaires.

I-1.2.- Les objectifs majeurs de l'organisation

On peut les résumer ainsi :

1 - Pour un article donné, chaque concessionnaire doit assurer une couverture en stock de 4 mois. Afin de l'aider dans ce sens, le Siège entretient un historique de commandes de réapprovisionnement par concessionnaire.

2 - Cet historique est utilisé par le Siège pour constituer un portefeuille de commandes prévisionnelles par concessionnaire.
Ces commandes sont livrables dans un délai de 4 mois.

3 - Ce portefeuille constitue une "entrée" pour les prévisions de production. Le délai maximum de fabrication est évidemment inférieur à 4 mois.

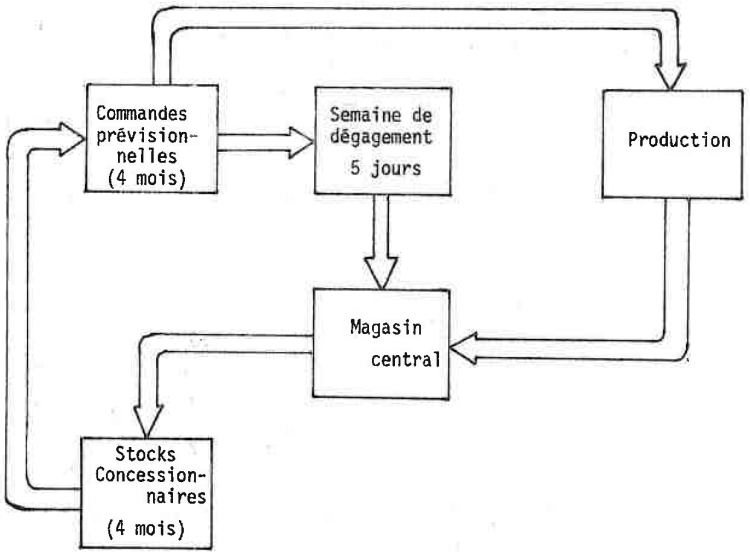
4 - Les articles sortis de production ne doivent transiter dans le magasin central que pendant une semaine.

En résumé, l'entreprise se fixe un triple objectif.

- régulation de la production en entretenant une prévision de 4 mois.
- réduire le plus possible le stock central.
- maintenir la rotation du stock concessionnaire à 4 mois.

L'entretien du portefeuille de commande et la valeur des informations qu'il contient constituent la clé de cet objectif.

Les concessionnaires auront la possibilité de passer des commandes (urgentes) ne passant pas par le circuit des commandes prévisionnelles.



I-1.3.- Description générale du fonctionnement de la société

I.1.3.1.- Les commandes

Le Siège approvisionne les concessionnaires en pièces détachées :

1 - par une procédure normale = les commandes prévisionnelles

2 - par une procédure d'exception = les commandes urgentes

Les commandes prévisionnelles sont lancées par le STI (Siège) sur la base d'une prévision.

Les commandes urgentes sont lancées directement par les concessionnaires.

Le stock central est alimenté directement par la production. Les articles ne doivent y séjourner que pendant une période très courte (une semaine).

Le contrôle de ce stock est sous la responsabilité de 60 "gestionnaires d'articles".

Ce contrôle comprend :

1 - approvisionnement du stock central

2 - contrôle du niveau des stocks

3 - suivi des commandes concessionnaires (prévisionnelles et urgentes). Chaque gestionnaire d'articles (GA) contrôle un groupe d'articles.

I-1.3.2.- L'emballage et l'expédition

Le magasin comprend le stock central, la zone de transit et les stalles. Ce magasin est sous la responsabilité des gestionnaires d'articles et du chef magasinier.

L'emballage comprend la zone d'emballage (préparation des colis) et la zone caisse (préparation des caisses). Cette partie est sous la responsabilité du chef magasinier et du chef d'emballage.

Les expéditions sont sous la responsabilité du chef des expéditions.

Circuits des articles - Commandes prévisionnelles.

- Les articles viennent de la production vers le stock central (par groupe d'articles).
- Les commandes sont servies sur stock par les GA. Cette sortie s'appelle une réquisition.
- Les réquisitions, sorties par article sont entreposées dans la zone de transit par article.
- Les magasiniers "éclatent" ces réquisitions dans les stalles, par concessionnaire. Il y a une stalle par concessionnaire.

- Les manutentionnaires de l'emballage enlèvent le contenu des stalles pour confectionner des colis par concessionnaire.
- L'emballage groupe les colis dans des caisses pour expédition. Dès qu'une caisse est pleine, la facturation est déclenchée et la caisse est expédiée.

N.B. : Le terme caisse désigne un envoi (caisse, container, camion, péniche, wagon).

I-2.- Description détaillée de la gestion des commandes prévisionnelles

I-2.1.- Emission

Les commandes prévisionnelles sont émises, chaque jour ouvrable, en trois exemplaires, à partir d'un historique des commandes, au service de la gestion des commandes (SGC).

Une commande prévisionnelle est livrable dans un délai de 4 mois, elle concerne un ou plusieurs articles (3 en moyenne) appartenant à un même groupe.

Un exemplaire est envoyé au service de la production, un autre au concessionnaire ; l'original est rangé dans le portefeuille des commandes (PC) géré par le SGC.

I-2.2.- Allocation

I.2.2.1.- Semaine de dégagement

5 jours avant la date de livraison, chaque commande est extraite du PC : elle entre dans la "semaine de dégagement".

I.2.2.2.- Eléments d'allocation

Tant qu'une commande n'est pas encore entrée dans la semaine de dégagement, la quantité commandée pour un article donné est exprimée avec une seule valeur. Or, pour des raisons de gestion interne (régulation de la production, optimisation du stock, meilleure répartition de la charge du magasinier, etc...) cette quantité

commandée sera servie en moyenne en trois fois. Chaque fraction de la quantité commandée s'appelle un élément d'allocation. Tant que la quantité commandée n'est pas soldée, l'article commandé reste dans le PC. La commande sera supposée totalement servie quand elle sortira de la semaine de dégagement.

I.2.2.3.- Statuts de l'élément d'allocation

Pour caractériser l'état d'un élément d'allocation au cours de son existence, on lui attribue un code statut mis à jour à chacune des étapes du cycle.

0 = élément d'allocation (EA) créé
1 = EA mis à servir
2 = EA servi (mis en stalle)
3 = EA emballé
4 = EA expédié et facturé
5 = EA reçu par le concessionnaire et réglé

I.2.2.4.- Règle d'allocation

Pour chaque groupe d'articles, il existe en général une règle d'allocation qui définit la ventilation de la ligne de commande en éléments d'allocation. Cette règle se ramène le plus souvent à un nombre maximum d'articles à livrer simultanément.

Pour les groupes d'articles pour lesquels il n'existe pas de règle d'allocation, c'est le GA qui choisit, d'après l'état du stock et les réapprovisionnements attendus, le nombre d'éléments d'allocation et la quantité correspondante.

I-2.2.5.- Date de mise à servir

A chaque élément d'allocation est associée une date de mise à servir appartenant à la semaine de dégagement de la commande correspondante.

I.2.2.6.- Fonctionnement de l'allocation

- Pour les groupes d'articles possèdant une règle d'allocation, l'allocation est effectuée au service gestion des commandes.

- Pour les autres, elle est effectuée au magasin par les GA et communiquée au SGC.

- Dans les deux cas les éléments d'allocation, comprenant une quantité à servir et une date de mise à servir, sont portés sur la commande prévisionnelle qui passe dans l'état "allouée".

I-2.2.7.- Signalement aux concessionnaires des commandes entrant dans la semaine de dégagement

Chaque fin de semaine, la liste de ses commandes, entrant dans la semaine de dégagement, la semaine suivante, est envoyée au concessionnaire par le SGC.

EDITIONS

Tous les jours le service de la gestion des commandes détermine les éléments d'allocation qui doivent être mis à servir le lendemain et éditent :

- une liste de réquisition
- des étiquettes
- des bons d'accompagnement

Liste de réquisition

La liste de réquisition est présentée par groupes d'articles et par articles ; elle donne le nombre total de chaque article à servir le lendemain.

Elle est destinée aux GA qui feront ainsi déplacer les articles vers la zone de transit.

Etiquette

Les étiquettes sont divisibles en deux parties, portant les

mêmes renseignements. Elles sont classées par article et concessionnaire. Il y en a une par élément d'allocation. Elles sont transmises au chef magasinier qui s'en sert pour répartir les articles dans les stalles des concessionnaires. Une partie est collée sur les articles, l'autre est conservée par le chef magasinier pour rapprochement avec le bon d'accompagnement.

Bon d'accompagnement

Le bon d'accompagnement est une liste par concessionnaire, commandes et articles. Il est transmis au chef magasinier. Il sert à ce dernier, à pointer, à partir des talons des étiquettes, les articles effectivement mis dans les stalles.

Quand tous les articles de cette liste sont pointés, le bon d'accompagnement est retourné au service de la gestion des commandes qui note sur les commandes prévisionnelles correspondantes, le fait que certains éléments d'allocation sont passés dans l'état "servi".

EMBALLAGE

Le bon d'accompagnement pointé sert au service de la gestion des commandes à éditer "une liste des articles à emballer" présentée par concessionnaire.

Cette liste est transmise au chef d'emballage. Elle lui permet d'organiser le travail des manutentionnaires de l'emballage qui confectionnent les colis.

Le numéro de colis est porté sur la liste des articles à emballer ainsi que le numéro de caisse dans laquelle il a trouvé place.

Le contenu d'une caisse concerne toujours un seul concessionnaire.

Une liste des caisses partiellement remplies et disponibles est éditée également chaque jour par le service de la gestion des commandes et est transmise également au chef d'emballage. Elle donne pour chaque caisse, ses caractéristiques et les colis qu'elle contient déjà.

La liste des articles à emballer, complétée par les numéros de colis et de caisse, est transférée, le soir, au service de la gestion des commandes.

Elle permet à ce service, de modifier l'état des éléments d'allocation correspondants qui passent dans l'état "emballé" et de produire une nouvelle liste de caisses pour le lendemain.

EXPEDITION

Les caisses sont expédiées quand elles sont pleines. Quand le chef d'emballage constate qu'une caisse est pleine, il le signale au service de la gestion des commandes sur la liste des caisses qu'il lui retourne.

Ce service édite alors la facture des articles contenus dans la caisse et le bon de transport.

La facture est envoyée au chef d'emballage qui la joint à la caisse.

Le bon de transport est envoyé au chef des expéditions, il est pointé par ce dernier quand la caisse correspondante a été effectivement expédiée.

Le bon de transport pointé et signé est retransmis au SGC qui note que les éléments d'allocation de la caisse sont passés dans l'état "expédié" et que la caisse est "indisponible".

RECEPTION

A l'arrivée de la caisse, le concessionnaire envoie un accusé de réception et le règlement de la facture au SGC qui note que les EA correspondants sont "reçus et réglés".

La caisse est retournée au siège central. A son arrivée le chef d'emballage le signale au SGC qui note qu'elle est à nouveau "disponible".

MODIFICATION OU ANNULATION DES COMMANDES PREVISIONNELLES

Tout concessionnaire peut demander l'annulation ou la modification d'une commande prévisionnelle qui lui est destinée, tant qu'elle n'est pas entrée dans la semaine de dégagement.

La modification ne peut porter que sur les quantités commandées.

Les modifications ou les annulations sont transmises en général par téléphone. Lorsqu'elles sont acceptées, un avis de modification ou d'annulation est envoyé par le SGC au concessionnaire les jours suivants.

PURGE DU PORTEFEUILLE DES COMMANDES

Au fur et à mesure de l'arrivée des accusés de réception, les employés du SGC notent sur les commandes prévisionnelles la réception des EA.

Quand tous les articles d'une commande sont reçus, la commande est dite soldée et est retirée du portefeuille des commandes et placée dans "l'historique".

QUELQUES CHIFFRES

Chaque jour environ 280 commandes sont émises et entrent dans la semaine de dégagement.

20 caisses, en moyenne, sont expédiées par jour et sont facturées ;
10 commandes, en moyenne, sont modifiées ou annulées par jour.

I-3.- Gestion des commandes urgentes

Chaque jour les concessionnaires envoient des commandes urgentes pour répondre à des demandes particulières (ruptures de stock, articles non suivis etc...). En moyenne 1080 commandes de ce type sont passées chaque jour 540 seulement seront honorées. Chaque commande comporte un seul article qui sera emballé séparément dans un colis et une caisse.

Le circuit des urgences est légèrement différent du circuit des commandes prévisionnelles afin de permettre deux livraisons journalières vers les concessionnaires. Ces commandes sont cependant enregistrées dans le portefeuille de commandes : la période de dégagement est égale à 1 jour et il n'y aura qu'un seul élément d'allocation par commande.

Saisie et validation d'une commande urgente d'un concessionnaire

Cette tâche permettra d'éditer un bon de réquisition en 3 exemplaires destiné au GA. Après vérification de l'existant en stock la liasse est pointée et transmise au GA.

Confection des caisses urgentes

Dans l'ensemble des 3000 caisses utilisables (prévisionnelles et urgentes) le GA choisira une caisse disponible par article commandé. Il reportera le numéro de caisse sur le bon de réquisition pointé qui s'appellera donc le bon de réquisition complété. Les 3 exemplaires seront séparés : un exemplaire destiné au stock, le deuxième au SGC pour la prise en compte des caisses pleines, le troisième au chef d'emballage pour lui permettre de créer les étiquettes destinées à être collées sur le colis et la caisse.

Après la création des étiquettes, on procèdera à l'emballage des articles commandés, collage des étiquettes sur les colis, mise en caisse des colis dans les caisses (1 colis = 1 caisse dans ce cas).

L'expédition, la facturation, la réception suivent les mêmes règles que pour les commandes prévisionnelles.

Les commandes urgentes soldées sont placées également dans l'historique des commandes.

II - LE SCHEMA CONCEPTUEL DU SI

II-1.- Métastructure du SI

Nous donnons, pour chaque relation de la métastructure, son extension telle qu'elle est produite par le système d'aide à la conception.

RELATION CC représentant les C-CLASSES

C-CLASSES DU SCHEMA CONCEPTUEL

NOM	LIBELLE
COM-PREV	COMMANDE PREVISIONNELLE
LCOMPREV	LIGNE DE COMM PREVISIONNELLE
EL-ALLOC	ELEMENT D ALLOCATION
CAISSE	CAISSE D EMBALLAGE
ARTICLE	ARTICLE FABRIQUE OU DISTRIBUE
COM-URG	COMMANDE URGENTE
CONCES	CONCESSIONNAIRE
GROUPE	GROUPE D ARTICLE
GEST	GESTIONNAIRE D ARTICLES
T-CAISSE	TYPE DE CAISSE
M-TRANS	MODE DE TRANSPORT
JOUR	JOUR OUVRABLE

RELATION CEV représentant les C-EVENEMENTS

C-EVENEMENTS DU SCHEMA CONCEPTUEL

NOM	LIBELLE	PREDICAT	OBJET CONSTATE
EV1	CHANGEMENT DE JOUR OUVRABLE	CREATION	JOUR
EV2	ENTREE COM EN SEMAINE DEGAGT	CREATION	COMDEG
EV3	MISE EN STALLE D UN ELT ALLOC	CREATION	ELMSTAL
EV4	EMBALLAGE D UN ELT ALLOC	CREATION	COLIS
EV5	CAISSE PLEINE	PEV5	ETAT-CAI
EV6	ARV DEMANDE MODIF LIG-COM	CREATION	DEMMLIG
EV7	ARV DEMANDE ANNULATION LIG-COM	CREATION	DEMANN
EV8	ANNULATION LIG-COM	PEV8	LIGQTE
EV9	ARRIVEE DE COMMANDE URGENTE	CREATION	DEMURG
EV10	EMBALLAGE DE COMMANDE URGENTE	CREATION	COLISURG
EV11	ENTREE DE PRODUIT EN MAGASIN	CREATION	ENTR-MAG

RELATION COB représentant les C-OBJETS

C-OBJETS DU SCHEMA CONCEPTUEL

NOM	LIBELLE	TYPE	CCLASSE
COMPREV	COM PREVISIONNELLE INITIALE	PERM	COM-PREV
COMDEG	COM PREVI EN DEGAGEMENT	VAR	COM-PREV
COMMAN	COM PREVI ANNULEE	SUP	COM-PREV
DEMANN	DEMANDE ANNULATION DE COM PREV	VAR	COM-PREV
REFANM	REFUS DE ANNULATION DE COMPREV	VAR	COM-PREV
LIGPREV	LIGNE DE COM PREV INITIALE	PERM	LCOMPREV
LIGQTE	QUANTITE ARTICLE DE LIGNE PREV	VAR	LCOMPREV
DEMMLIG	DEMANDE DE MODIF DE LIGNE PREV	VAR	LCOMPREV
REFMLIG	REFUS DE MODIF DE LIGNE PREV	VAR	LCOMPREV
ALLOCMAN	DEMANDE D ALLOCATION MANUELLE	VAR	LCOMPREV
EL-ALLOC	ELEMENT D ALLOCATION INITIAL	PERM	EL-ALLOC
ELMSERV	ELEMENT ALLOC MIS A SERVIR	VAR	EL-ALLOC
ELMSTAL	ELEMENT ALLOC MIS EN STALLE	VAR	EL-ALLOC
COLIS	COLIS DE UN ELEMENT	VAR	EL-ALLOC
ELFAC	ELEMENT D ALLOCATION FACTURE	VAR	EL-ALLOC
CAISSE	CAISSE INITIALE	PERM	CAISSE
ETAT-CAI	ETAT D UNE CAISSE	VAR	CAISSE
ATT-CAI	CONCESSIONNAIRE D UNE CAISSE	VAR	CAISSE
ARTICLE	ARTICLE INITIAL	PERM	ARTICLE
PRIX-ART	PRIX D UN ARTICLE	VAR	ARTICLE
STOCKMAG	QUANTITE EN STOCK EN MAGASIN	VAR	ARTICLE
SEUIL	SEUIL DE REFUS DE COM-URG	VAR	ARTICLE
ENTR-MAG	ENTREE D ARTICLE EN MAGASIN	VAR	ARTICLE
DEMURG	DEMANDE DE COMMANDE URGENTE	PERM	COM-URG
REFURG	REFUS DE COMMANDE URGENTE	VAR	COM-URG
URGMSERV	COMMANDE URGENTE MIS A SERVIR	VAR	COM-URG
COLISURG	COLIS D UNE COMMANDE URGENTE	VAR	COM-URG
FACURG	FACTURE D UNE COMMANDE URGENTE	VAR	COM-URG
CONS	CONCESSIONNAIRE	PERM	CONCES
CONSRAIS	RAISON SOCIALE CONCESSIONNAIRE	VAR	CONCES
CONSADR	ADRESSE CONCESSIONNAIRE	VAR	CONCES
CONSTEL	TELEPHONE CONCESSIONNAIRE	VAR	CONCES
CONSSTAL	STALLE D UN CONCESSIONNAIRE	VAR	CONCES
TYPCAISS	TYPE DE CAISSE	PERM	T-CAISSE
TYPTRANS	MODE DE TRANSPORT TYPE CAISSE	VAR	T-CAISSE
MODTRANS	MODE DE TRANSPORT	PERM	M-TRANS
TXTRANS	TAUX DE TRANSPORT	VAR	M-TRANS
GROUPE	GROUPE D ARTICLE INITIAL	PERM	GROUPE
GRGEST	GESTIONNAIRE D UN GROUPE	VAR	GROUPE
GRALLOC	REGLE D ALLOCATION D UN GROUPE	VAR	GROUPE
GEST	GESTIONNAIRE D ARTICLES	PERM	GEST
GESTPAR	GESTIONNAIRE PARTI	SUP	GEST
JOUR	JOUR OUVRABLE	PERM	JOUR

Relation COP représentant les C-OPERATIONS

C-OPERATIONS DU SCHEMA CONCEPTUEL

NOM	LIBELLE	TEXTE	OBJET MODIFIE
OP1	DEGAGEMENT D UNE COMMANDE	TOP1	COMDEG
OP3	CREATION DE LIG-COM PREVI	TOP3	LIGPREV
OP4	CALCUL DE LA QUANTITE A LIVRER	TOP4	LIGQTE
OP5	CREATION DE COM PREVI	TOP5	COMPREV
OP13	DIMINUTION DE STOCK	TOP13	STOCKMAG
OP15	REQUISITION D UNE CAISSE	TOP15	ETAT-CAI
OP16	ATTRIBUTION D UNE CAISSE	TOP16	ATT-CAI
OP20	FACTURATION DE COM PREVI	TOP20	ELFAC
OP8	MODIFICATION D UNE LIGN-COM	TOP8	LIGQTE
OP9	REFUS DE MODIF DE LIGN-COM	TOP9	REFMLIG
OP10	ANNULATION D UNE COMMANDE	TOP10	COMMAN
OP12	REFUS ANNULATION DE COMMANDE	TOP12	REFANM
OP21	ACCEPTATION DE COMMANDE URGENT	TOP21	URGMSERV
OP22	REFUS DE COMMANDE URGENTE	TOP22	REFURG
OP19	FACTURATION DE COMMANDE URGENT	TOP19	FACURG
OP18	ATTRIBUTION DE CAISSE URGENTE	TOP18	ATT-CAI
OP17	EXPEDITION DE CAISSE URGENTE	TOP17	ETAT-CAI
OP23	DIMINUTION DE STOCK	TOP23	STOCKMAG
OP14	AUGMENTATION DU STOCK	TOP14	STOCKMAG
OP2	MISE A SERVIR D UN ELT-ALLOC	TOP2	ELMSERV
OP6	ALLOCATION	TOP6	EL-ALLOC
OP7	DEMANDE D ALLOCATION MANUELLE	TOP7	ALLOCMAN

Relation CDECL représentant les C-DECLENCHE

C-DECLENCHES DU SCHEMA CONCEPTUEL

NOM	EVENEMENT	OPERATION	CONDITION	FACTEUR	MULTIPLICITE
DECL1	EV1	OP1	TOUJOURS	F1	MULT
DECL2	EV2	OP3	TOUJOURS	F3	MULT
DECL3	EV2	OP4	TOUJOURS	F4	MULT
DECL4	EV2	OP5	TOUJOURS	F50	UN
DECL5	EV3	OP13	TOUJOURS	F13	UN
DECL6	EV4	OP15	C5	F15	UN
DECL7	EV4	OP16	C5	F16	UN
DECL8	EV5	OP20	TOUJOURS	F8	MULT
DECL9	EV6	OP8	C1	F80	UN
DECL10	EV6	OP9	NONC1	F90	UN
DECL11	EV7	OP10	C2	F10	UN
DECL12	EV7	OP12	NONC2	F12	UN
DECL13	EV8	OP10	C3	F101	UN
DECL14	EV9	OP21	C6	F21	UN
DECL15	EV9	OP22	NONC6	F22	UN
DECL16	EV10	OP19	TOUJOURS	F19	UN
DECL17	EV10	OP18	TOUJOURS	F18	UN
DECL18	EV10	OP17	TOUJOURS	F17	UN
DECL19	EV10	OP23	TOUJOURS	F23	UN
DECL20	EV11	OP14	TOUJOURS	F14	UN
DECL21	EV1	OP2	TOUJOURS	F2	MULT
DECL22	EV2	OP6	TOUJOURS	F6	MULT
DECL23	EV2	OP7	TOUJOURS	F7	MULT

II-2.- La structure du SI

II-2.1.- Le sous-schéma des données

II-2.1.1.- Collection des schémas de relation

EDITION DU SOUS SCHEMA DES DONNEES

```
CLASSE  COM-PREV  CLE NCP          COMMANDE PREVISIONNELLE

C-OBJET  PERM   COMPREV (NCP     ,DEMICP   ,NCONS    ,NGROUPE ,DLIVR   )
C-OBJET  VAR    COMDEG  (N02     ,NCP      ,DDEGAG   )
C-OBJET  SUP    COMMAN  (N03     ,NCP      ,DANNC    )
C-OBJET  VAR    DEMANN  (N04     ,NCP      ,DDANNC   )
C-OBJET  VAR    REFANM  (N05     ,NCP      ,DREFUS   ,MOTREFC )

CLASSE  LCOMPREV  CLE N06        LIGNE DE COMM PREVISIONNELLE

C-OBJET  PERM   LIGPREV (N06     ,DEMISLCP,NCP       ,NART     )
C-OBJET  VAR    LIGQTE  (N07     ,N06      ,DMODQTE  ,QTE      )
C-OBJET  VAR    DEMMLIG (N08     ,N06      ,DDEMMOD  ,NQTE     )
C-OBJET  VAR    REFMLIG (N09     ,N06      ,DRMOD    ,MOTREFL  )
C-OBJET  VAR    ALLOCMAN(N010    ,N06      ,DDEMALLM,MOTALLOC)

CLASSE  EL-ALLOC  CLE NELT       ELEMENT D ALLOCATION

C-OBJET  PERM   EL-ALLOC(NELT    ,DCELT    ,N06      ,QTELT    ,DMSERVIP)
C-OBJET  VAR    ELMSERV (N012    ,NELT     ,DMSERVIR)
C-OBJET  VAR    ELMSTAL (N013    ,NELT     ,DMENSTAL)
C-OBJET  VAR    COLIS   (N014    ,NELT     ,DEMBALLE,NOCOLIS ,POIDS    ,NOCAISSE)
C-OBJET  VAR    ELFAC   (N015    ,NELT     ,DFAC     ,NFAC     )

CLASSE  CAISSE    CLE NOCAISSE   CAISSE D EMBALLAGE

C-OBJET  PERM   CAISSE  (NOCAISSE,DACHCAIS,NTYPE    ,CAPACITE,POIDSCAI)
C-OBJET  VAR    ETAT-CAI(N017    ,NOCAISSE,DCHGETCA,ETAT     )
C-OBJET  VAR    ATT-CAI (N018    ,NOCAISSE,DATT     ,NCONS    )

CLASSE  ARTICLE   CLE NART       ARTICLE FABRIQUE OU DISTRIBUE

C-OBJET  PERM   ARTICLE (NART    ,DCART    ,DESART   ,NGROUPE  ,PDART    )
C-OBJET  VAR    PRIX-ART(N020    ,NART     ,DGTPRIX  ,PRIX     )
C-OBJET  VAR    STOCKMAG(N021    ,NART     ,DMSTMAG  ,STOCKMAG)
C-OBJET  VAR    SEUIL   (N022    ,NART     ,DMSEUIL  ,SEUIL    )
C-OBJET  VAR    ENTR-MAG(N023    ,NART     ,DENTMAG  ,QTE-ENTR)

CLASSE  COM-URG   CLE NCDEURG    COMMANDE URGENTE

C-OBJET  PERM   DEMURG  (NCDEURG ,DAMURG   ,NART     ,QTURG    ,NCONS    )
C-OBJET  VAR    REFURG  (N025    ,NCDEURG  ,DREFURG  ,MOTURG   )
C-OBJET  VAR    URGMSERV(N026    ,NCDEURG  ,DMSERVUR)
C-OBJET  VAR    COLISURG(N027    ,NCDEURG  ,DEMBURG  ,COLISURG,PDSURG  ,NOCAISSE)
C-OBJET  VAR    FACURG  (N028    ,NCDEURG  ,DFACURG  ,NFAC     )

CLASSE  CONCES    CLE NCONS      CONCESSIONNAIRE

C-OBJET  PERM   CONS    (NCONS   ,DENTCONS)
C-OBJET  VAR    CONSRAIS(N030    ,NCONS    ,DMRAISOC,RAISON  )
C-OBJET  VAR    CONSADR (N031    ,NCONS    ,DMADR    ,ADRESSE )
C-OBJET  VAR    CONSTEL (N032    ,NCONS    ,DMTEL    ,TELEPHON)
C-OBJET  VAR    CONSSTAL(N033    ,NCONS    ,DGTSTAL  ,NSTALLE )
```


Sous-schéma des données (suite)

```
CLASSE  GROUPE    CLE NGROUPE    GROUPE D ARTICLE

C-OBJET  PERM   GROUPE  (NGROUPE ,DCGROUPE)
C-OBJET  VAR    GRGEST  (N039    ,NGROUPE ,DCHGEST,NGEST   )
C-OBJET  VAR    GRALLOC (N040    ,NGROUPE ,DCHGALLO,REGLALLO)

CLASSE  GEST      CLE NGEST      GESTIONNAIRE D ARTICLES

C-OBJET  PERM   GEST    (NGEST   ,DENTFONC)
C-OBJET  SUP    GESTPAR (N042    ,NGEST    ,DDEPGEST)

CLASSE  T-CAISSE  CLE NTYPE      TYPE DE CAISSE

C-OBJET  PERM   TYPCAISS(NTYPE   ,DESTYPE  ,DCTYPE   )
C-OBJET  VAR    TYPTRANS(N035    ,NTYPE    ,DMTRANS  ,NTRANS  )

CLASSE  M-TRANS   CLE NTRANS     MODE DE TRANSPORT

C-OBJET  PERM   MODTRANS(NTRANS  ,DCTRANS  ,DESTRANS)
C-OBJET  VAR    TXTRANS (N037    ,NTRANS   ,DMODTAUX,TAUXTRAN)

CLASSE  JOUR      CLE NOJOUR     JOUR OUVRABLE

C-OBJET  PERM   JOUR    (NOJOUR  ,DJOUR    )
```

II-2.1.2.- Signification des constituants

Les constituants sont listés dans leur ordre d'apparition dans les C-OBJETS de la liste précédente.

CONSTITUANTS DES C-OBJETS

NCP	NUMERO DE COMMANDE PREVISIONNELLE
DEMICP	DATE D EMISSION DE COMMANDE PREVISIONNELLE
NCONS	NUMERO DE CONCESSIONNAIRE
NGROUPE	NUMERO DE GROUPE
DLIVR	DATE DE LIVRAISON
N02	CLE SYNONYME DE COMDEG
DDEGAG	DATE D ENTREE EN SEMAINE DE DEGAGEMENT
N03	CLE SYNONYME DE COMMAN
DANNC	DATE ANNULATION DE COMMANDE PREVISIONNELLE
N04	CLE SYNONYME DE DEMANN
DDANNC	DATE DE DEMANDE ANNULATION DE COM PREV
N05	CLE SYNONYME DE REFANM
DREFUS	DATE DE REFUS ANNULATION DE COM PREVI
MOTREFC	MOTIF DE REFUS ANNULATION DE COM PREVI
N06	CLE SYNONYME DE LIGPREV
DEMISLCP	DATE D EMISSION DE LIGNE PREVISIONNELLE
NART	NUMERO D ARTICLE
N07	CLE SYNONYME DE LIGQTE
DMODQTE	DATE DE MODIF DE LIGNE PREVISIONNELLE
QTE	QUANTITE D ARTICLE DE LIGNE PREVI
N08	CLE SYNONYME DE DEMMLIG
DDEMMOD	DATE DE DEMANDE DE MODIF DE LIGNE PREVI
NQTE	NOUVELLE QUANTITE DEMANDEE
N09	CLE SYNONYME DE REFMLIG

Signification des constituants (suite)

DRMOD	DATE DE REFUS DE MODIF DE LIGNE PREVI
MOTREFL	MOTIF DE REFUS DE MODIF DE LIGNE PREVI
NO10	CLE SYNONYME DE ALLOCMAN
DDEMALLM	DATE DE DEMANDE D ALLOCATION MANUELLE
MOTALLOC	MOTIF DE L ALLOCATION MANUELLE
NELT	NUMERO D ELEMENT D ALLOCATION
DCELT	DATE DE CREATION D ELEMENT D ALLOCATION
QTELT	QUANTITE D ARTICLES D UN ELEMENT D ALLOCATION
DMSERVIP	DATE DE MISE A SERVIR PREVISIONNELLE D UN ELEMENT D ALLOCATI
NO12	CLE SYNONYME DE ELMSERV
DMSERVIR	DATE DE MISE A SERVIR EFFECTIVE D UN ELEMENT D ALLOCATION
NO13	CLE SYNONYME DE ELMSTAL
DMENSTAL	DATE DE MISE EN STALLE D UN ELEMENT D ALLOCATION
NO14	CLE SYNONYME DE COLIS
DEMBALLE	DATE D EMBALLAGE DE L ELEMENT D ALLOCATION
NOCOLIS	NUMERO DE COLIS
POIDS	POIDS D UN COLIS
NOCAISSE	NUMERO DE CAISSE
NO15	CLE SYNONYME DE ELFAC
DFAC	DATE DE FACTURATION D UN ELEMENT D ALLOCATION
NFAC	NUMERO DE FACTURE
DACHCAIS	DATE D ACHAT D UNE CAISSE
NTYPE	NUMERO DE TYPE DE CAISSE
CAPACITE	CAPACITE D UNE CAISSE
POIDSCAI	POIDS D UNE CAISSE
NO17	CLE SYNONYME DE ETAT-CAI
DCHGETCA	DATE DE CHANGEMENT D ETAT D UNE CAISSE
ETAT	ETAT D UNE CAISSE
NO18	CLE SYNONYME DE ATT-CAI
DATT	DATE D ATTRIBUTION D UNE CAISSE A UN CONCESSIONNAIRE
DCART	DATE D INTRODUCTION DE L ARTICLE AU CATALOGUE
DESART	DESIGNATION D UN ARTICLE
PDART	POIDS D UN ARTICLE
NO20	CLE SYNONYME DE PRIX-ART
DGTPRIX	DATE DE MODIFICATION DE PRIX D ARTICLE
PRIX	PRIX D UN ARTICLE
NO21	CLE SYNONYME DE STOCKMAG
DMSTMAG	DATE DE MODIFICATION DU STOCK D UN ARTICLE
STOCKMAG	QUANTITE EN STOCK D UN ARTICLE EN MAGASIN
NO22	CLE SYNONYME DE SEUIL
DMSEUIL	DATE DE MISE A JOUR DU SEUIL D UN ARTICLE
SEUIL	SEUIL DE REFUS DES COMMANDES URGENTES ET DE L ALLOCATION MAN
NO23	CLE SYNONYME DE ENTR-MAG
DENTMAG	DATE D ENTREE D ARTICLE EN MAGASIN
QTE-ENTR	QUANTITE D ARTICLE ENTREE
NCDEURG	NUMERO DE COMMANDE URGENTE
DARURG	DATE D ARRIVEE D UNE COMMANDE URGENTE
QTURG	QUANTITE D ARTICLE D UNE COMMANDE URGENTE
NO25	CLE SYNONYME DE REFURG
DREFURG	DATE DE REFUS DE COMMANDE URGENTE
MOTURG	MOTIF DE REFUS D UNE COMMANDE URGENTE
NO26	CLE SYNONYME DE URGNSERV
DMSERVUR	DATE DE MISE A SERVIR D UNE COMMANDE URGENTE
NO27	CLE SYNONYME DE COLISURG
DEMBURG	DATE D EMBALLAGE D UNE COMMANDE URGENTE
COLISURG	NUMERO DE COLIS D UNE COMMANDE URGENTE

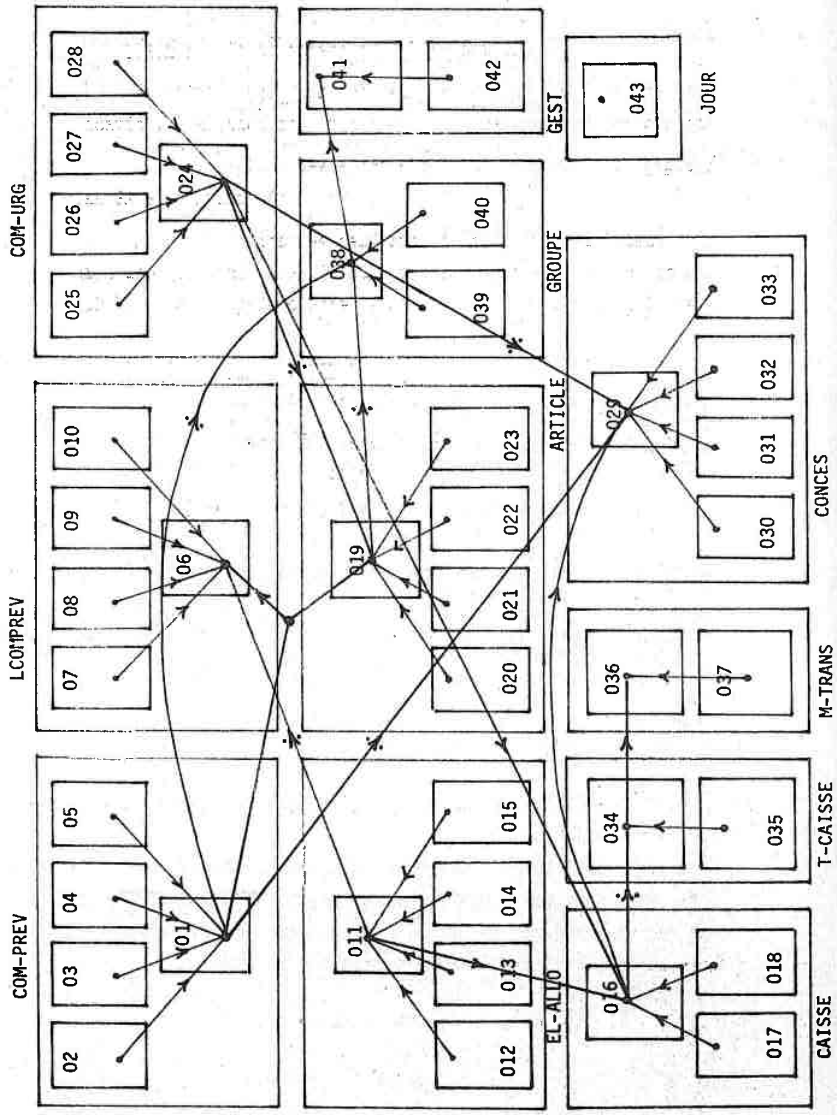
Signification des constituants (suite)

NCAISSE	
PDSURG	POIDS D UNE COMMANDE URGENTE
NO28	CLE SYNONYME DE FACURG
DFACURG	DATE DE FACTURATION D UNE COMMANDE URGENTE
DENTCONS	DATE D ENTREE D UN CONCESSIONNAIRE DANS LA SOCIETE
NO30	CLE SYNONYME DE CONSRAIS
DMRAISOC	DATE DE MODIFICATION DE LA RAISON SOCIALE D UN CONCESSIONNAI
RAISON	RAISON SOCIALE D UN CONCESSIONNAIRE
NO31	CLE SYNONYME DE CONSADR
DMADR	DATE DE CHGT D ADRESSE D UN CONCESSIONNAIRE
ADRESSE	ADRESSE D UN CONCESSIONNAIRE
NO32	CLE SYNONYME DE CONSTEL
DMTEL	DATE DE MODIFICATION D ADRESSE D UN CONCESSIONNAIRE
TELEPHON	NUMERO DE TELEPHONE D UN CONCESSIONNAIRE
NO33	CLE SYNONYME DE CONSSTAL
DGTSTAL	DATE DE CHANGT DE STALLE D UN CONCESSIONNAIRE
NSTALLE	NUMERO DE STALLE
DESTYPE	DESIGNATION DU TYPE DE CAISSE
DCTYPE	DATE DE CREATION D UN TYPE DE CAISSE
NO35	CLE SYNONYME DE TYPTRANS
DMTRANS	DATE DE CHGT DE MODE DE TRANSPORT D UN TYPE DE CAISSE
NTRANS	NUMERO DE MODE DE TRANSPORT
DCTRANS	DATE DE CREATION D UN MODE DE TRANSPORT
DESTRANS	DESIGNATION D UN MODE DE TRANSPORT
NO37	CLE SYNONYME DE TXTRANS
DMODTAUX	DATE DE MODIFICATION DU TAUX DE TRANSPORT
TAUXTRAN	TAUX DE TRANSPORT D UN TYPE DE CAISSE
DCGROUPE	DATE DE CREATION D UN GROUPE D ARTICLE
NO39	CLE SYNONYME DE GRGEST
DCGTGEST	DATE DE CHANGT DE GESTIONNAIRE D UN GROUPE D ARTICLE
NGEST	NUMERO DE GESTIONNAIRE
NO40	CLE SYNONYME DE GRALLOC
DCHGALLO	DATE DE CHANGT DE LA REGLE D ALLOCATION D UN GROUPE D ARTICL
REGLALLO	REGLE D ALLOCATION D UN GROUPE D ARTICLE
DENTFONC	DATE D ENTREE EN FONCTION D UN GESTIONNAIRE
NO42	CLE SYNONYME DE GESTPAR
DDEPGEST	DATE DE DEPART D UN GESTOIONNAIRE
NOJOUR	NUMERO DE JOUR OUVRABLE DANS L ANNEE
DJOUR	DATE D UN JOUR OUVRABLE

II-2.1.3.- Représentation graphique du sous-schéma des données

Remarque : Dans le représentation graphique du sous-schéma des données nous avons pour des raisons d'encombrement remplacé le nom d'un C-OBJET par un synonyme O_i où i est le numéro d'ordre d'apparition du C-OBJET dans l'édition du sous-schéma des données.

II-1.2.- La représentation graphique du sous-schéma des données



Liaisons entre C-CLASSES

- DFP entre la C-CLASSE COMPREV et la C-CLASSE CONCES
  Exprime que,dès sa création,une commande prévisionnelle est attribuée à un concessionnaire unique pour toute la durée de sa gestion.

- DFP entre la C-CLASSE COMPREV et la C-CLASSE GROUPE
  Exprime que,dès sa création,une commande prévisionnelle concerne un groupe d'articles unique pour toute la durée de sa gestion.

- DFP entre la C-CLASSE COM-URG et la C-CLASSE CONCES
  Signification identique à celle concernant la C-CLASSE COMPREV

- DFP entre la C-CLASSE COM-URG et la C-CLASSE ARTICLE
  Exprime le fait qu'une commande urgente ne concerne qu'un seul type d'article connu dès la création de la commande.

- DF entre la C-CLASSE COM-URG et la C-CLASSE CAISSE
  Exprime le fait qu'une commande-urgente est placée dans une seule caisse dont le numéro n'est pas connu à la création de la commande.

- DFP entre la C-CLASSE EL-ALLOC et la C-CLASSE LCOMPREV

  Exprime le fait qu'un élément d'allocation correspond à une seule ligne de commande prévisionnelle connue à sa création.

- DF entre la C-CLASSE EL-ALLOC et la C-CLASSE CAISSE
  Exprime le fait qu'un élément d'allocation est placé dans une seule caisse, à un instant ultérieur à sa création.

- DF entre la C-CLASSE CAISSE et la C-CLASSE CONCES
  Exprime le fait qu'une caisse à un instant donné, est attribuée à un seul concessionnaire, mais qu'au cours du temps, peut être attribuée à des concessionnaires différents.

- DFP entre la C-CLASSE CAISSE et la C-CLASSE T-CAISSE
  Exprime le fait qu'une caisse est d'un certain type, identique au cours du temps.

- DF entre la C-CLASSE T-CAISSE et la C-CLASSE M-TRANS

  Exprime le fait qu'à un type de caisse est associé, à un instant donné, un seul type de transport qui peut changer au cours du temps.

- L'identifiant de la C-CLASSE LCOMPREV est composé à partir des identifiants des C-CLASSES COMPREV et ARTICLE

  Exprime le fait qu'une ligne de commande prévisionnelle concerne une seule commande et un seul article.

- DFP entre la C-CLASSE ARTICLE et la C-CLASSE GROUPE

  Exprime le fait qu'un article appartient toujours au cours du temps à un seul et même groupe d'articles.

- DF entre la C-CLASSE GROUPE et la C-CLASSE GEST

  Exprime le fait qu'à un instant donné, il existe un seul gestionnaire d'articles responsable d'un groupe d'articles, le gestionnaire responsable d'un groupe pouvant être différent au cours du temps.

II-2.2.- Le sous-schéma de la dynamique

II-2.2.1.- Collection des schémas de relation

Les noms des constituants des relations du sous-schéma de la dynamique sont générés par le système à l'exception de ceux qui correspondent à des clés des C-OBJETS.

Leur signification est évidente

IEVN (respectivement IOPN) est le nom du constituant représentant l'identifiant du C-EVENEMENT (respectivement de la C-OPERATION) ayant pour nom EVN (respectivement OPN).

DAREVN est le nom du constituant représentant la propriété date d'arrivée du C-EVENEMENT EVN.

DOPN est le nom du constituant représentant la propriété date de fin d'exécution de la C-OPERATION OPN.

DDECLN est le nom du constituant représentant la propriété date de déclenchement de la C-OPERATION appartenant à l'association DECLN.

EDITION DU SOUS SCHEMA DE LA DYNAMIQUE

C-OPERATIONS

```
C-OP   OP1    (IOP1    ,DOP1    ,NO2    )
C-OP   OP3    (IOP3    ,DOP3    ,NO6    )
C-OP   OP4    (IOP4    ,DOP4    ,NO7    )
C-OP   OP5    (IOP5    ,DOP5    ,NCP    )
C-OP   OP13   (IOP13   ,DOP13   ,NO21   )
C-OP   OP15   (IOP15   ,DOP15   ,NO17   )
C-OP   OP16   (IOP16   ,DOP16   ,NO18   )
C-OP   OP20   (IOP20   ,DOP20   ,NO15   )
C-OP   OP8    (IOP8    ,DOP8    ,NO7    )
C-OP   OP9    (IOP9    ,DOP9    ,NO9    )
C-OP   OP10   (IOP10   ,DOP10   ,NO3    )
C-OP   OP12   (IOP12   ,DOP12   ,NO5    )
C-OP   OP21   (IOP21   ,DOP21   ,NO26   )
C-OP   OP22   (IOP22   ,DOP22   ,NO25   )
C-OP   OP19   (IOP19   ,DOP19   ,NO23   )
C-OP   OP18   (IOP18   ,DOP18   ,NO18   )
C-OP   OP17   (IOP17   ,DOP17   ,NO17   )
C-OP   OP23   (IOP23   ,DOP23   ,NO21   )
C-OP   OP14   (IOP14   ,DOP14   ,NO21   )
C-OP   OP2    (IOP2    ,DOP2    ,NO12   )
C-OP   OP6    (IOP6    ,DOP6    ,NELT   )
C-OP   OP7    (IOP7    ,DOP7    ,NO10   )
```

C-EVENEMENTS

```
C-EV   EV1 (IEV1    ,DAREV1    ,NOJOUR )
C-EV   EV2 (IEV2    ,DAREV2    ,NO2    )
C-EV   EV3 (IEV3    ,DAREV3    ,NO13   )
C-EV   EV4 (IEV4    ,DAREV4    ,NO14   )
C-EV   EV5 (IEV5    ,DAREV5    ,NO17   )
C-EV   EV6 (IEV6    ,DAREV6    ,NO8    )
C-EV   EV7 (IEV7    ,DAREV7    ,NO4    )
C-EV   EV8 (IEV8    ,DAREV8    ,NO7    )
C-EV   EV9 (IEV9    ,DAREV9    ,NCDEURG )
C-EV   EV10(IEV10   ,DAREV10   ,NO27   )
C-EV   EV11(IEV11   ,DAREV11   ,NO23   )
```

C-DECLENCHES

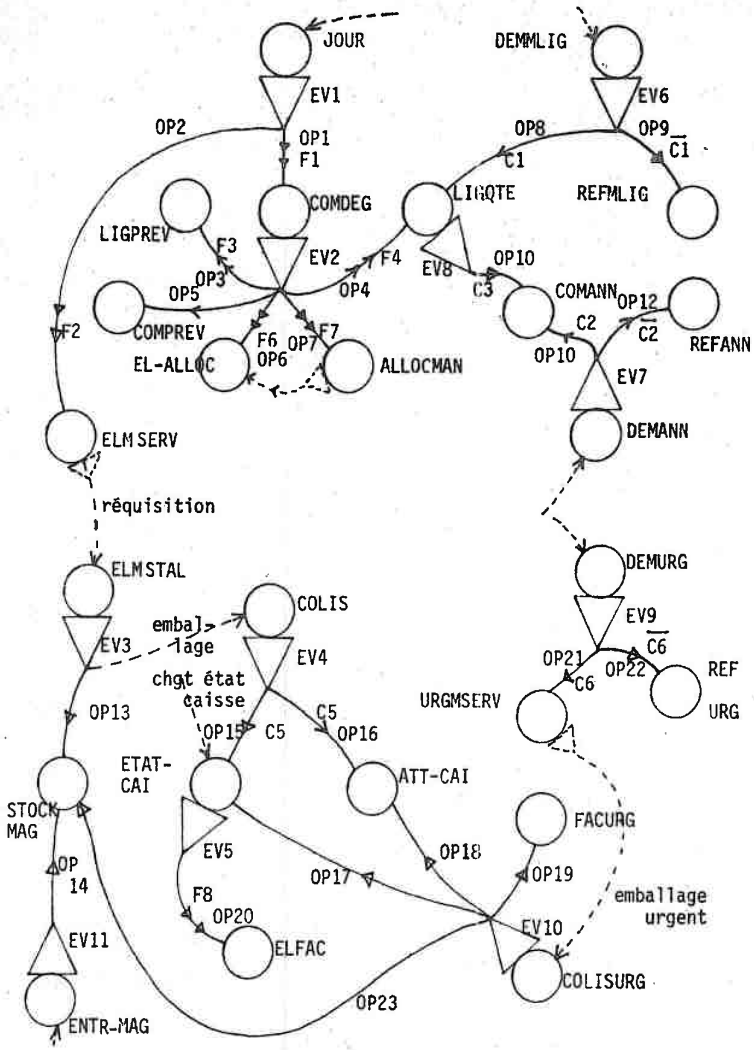
C-DECL	DECL1	(IEV1	,IOP1	,DDECL1)
C-DECL	DECL2	(IEV2	,IOP3	,DDECL2)
C-DECL	DECL3	(IEV2	,IOP4	,DDECL3)
C-DECL	DECL4	(IEV2	,IOP5	,DDECL4)
C-DECL	DECL5	(IEV3	,IOP13	,DDECL5)
C-DECL	DECL6	(IEV4	,IOP15	,DDECL6)
C-DECL	DECL7	(IEV4	,IOP16	,DDECL7)
C-DECL	DECL8	(IEV5	,IOP20	,DDECL8)
C-DECL	DECL9	(IEV6	,IOP8	,DDECL9)
C-DECL	DECL10	(IEV6	,IOP9	,DDECL10)
C-DECL	DECL11	(IEV7	,IOP10	,DDECL11)
C-DECL	DECL12	(IEV7	,IOP12	,DDECL12)
C-DECL	DECL13	(IEV8	,IOP10	,DDECL13)
C-DECL	DECL14	(IEV9	,IOP21	,DDECL14)
C-DECL	DECL15	(IEV9	,IOP22	,DDECL15)
C-DECL	DECL16	(IEV10	,IOP19	,DDECL16)
C-DECL	DECL17	(IEV10	,IOP18	,DDECL17)
C-DECL	DECL18	(IEV10	,IOP17	,DDECL18)
C-DECL	DECL19	(IEV10	,IOP23	,DDECL19)
C-DECL	DECL20	(IEV11	,IOP14	,DDECL20)
C-DECL	DECL21	(IEV1	,IOP2	,DDECL21)
C-DECL	DECL22	(IEV2	,IOP6	,DDECL22)
C-DECL	DECL23	(IEV2	,IOP7	,DDECL23)

II-2.2.2.- Représentation graphique du sous-schéma de la dynamique

Dans la représentation graphique du sous-schéma nous avons fait apparaître en pointillé les opérations et les événements qui n'appartiennent pas au sous-système causal du système représenté ou qui appartiennent à son environnement.

Par exemple : - L'opération d'emballage n'appartient pas au sous-système causal du système représenté car cette opération est effectuée suivant la décision du chef magasinier, selon un processus non défini, de même, le choix d'une caisse qu'elle nécessite n'obéit pas à une règle déterminée.

Le signalement de ces opérations, et parfois de l'événement qui les provoquent met en évidence les limites de la représentation effectuée.



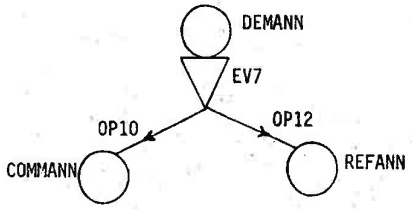
Sous-schéma de la dynamique

II-2.2.3.- Exploration et édition du sous-schéma de la dynamique par le système

Signification

```
DEMANN ← EV7 → OP10  ,  OP12
                 ↓       ↓
              COMMANN , REFANN
```

correspond au cycle




```
$go explordyn
DEMANN  <----EV7     ---->OP10     ,OP12     ,
                          ↓          ↓
                        COMMAN    ,REFANM   ,
DEMMLIG <----EV6     ---->OP8      ,OP9      ,
                          ↓          ↓
                        LIGQTE    ,REFMLIG  ,
LIGQTE  <----EV8     ---->OP10     ,
                          ↓
                        COMMAN    ,
ELMSTAL <----EV3     ---->OP13     ,
                          ↓
                        STOCKMAG  ,
COLIS   <----EV4     ---->OP15     ,OP16     ,
                          ↓          ↓
                        ETAT-CAI  ,ATT-CAI  ,
ETAT-CAI<----EV5     ---->OP20     ,
                          ↓
                        ELFAC     ,
ENTR-MAG<----EV11    ---->OP14     ,
                          ↓
                        STOCKMAG  ,
DEMURG  <----EV9     ---->OP21     ,OP22     ,
                          ↓          ↓
                        URGMSERV  ,REFURG   ,
COLISURG<----EV10    ---->OP19     ,OP18     ,OP17     ,OP23     ,
                          ↓          ↓         ↓         ↓
                        FACURG    ,ATT-CAI  ,ETAT-CAI ,STOCKMAG ,
JOUR    <----EV1     ---->OP1      ,OP2      ,
                          ↓          ↓
                        COMDEG    ,ELMSERV  ,
COMDEG  <----EV2     ---->OP3      ,OP4      ,OP5      ,OP6      ,OP7      ,
                          ↓          ↓         ↓         ↓         ↓
                        LIGPREV   ,LIGQTE   ,COMPREV  ,EL-ALLOC ,ALLOCMAN,
```


II-2.2.4.- Contraintes d'intégrité dynamique

- Contraintes de "constance" des C-OBJETS de type variable

Rappelons que cette contrainte associée à un C-OBJET O_1 a la signification suivante :

O_1 représente un état d'une classe d'objets réels, dans lequel un élément de la classe ne peut passer qu'une fois au cours du temps.

Les C-OBJETS suivants possèdent cette contrainte :

CONTRAINTES DE CONSTANCE DES C-OBJETS

COMDEG	COM PREVI EN DEGAGEMENT
ELMSERV	ELEMENT ALLOC MIS A SERVIR
ELMSTAL	ELEMENT ALLOC MIS EN STALLE
COLIS	COLIS DE UN ELEMENT
ELFAC	ELEMENT D ALLOCATION FACTURE
REFURG	REFUS DE COMMANDE URGENTE
URGMSERV	COMMANDE URGENTE MIS A SERVIR
COLISURG	COLIS D UNE COMMANDE URGENTE
FACURG	FACTURE D UNE COMMANDE URGENTE

- Dépendances chronologiques entre C-OBJETS

Rappelons que deux C-OBJETS O_1 et O_2 d'une même C-CLASSE de clé I sont en dépendance chronologique si $\forall t$ on doit avoir $O_{2_t}[I] \subset O_{1_t}[I]$.

Si O_1 et O_2 représentent des états particuliers pris au cours du temps par les objets de la classe représentée par la C-CLASSE, cette contrainte signifie qu'un élément de la classe ne peut passer dans l'état représenté par O_1 que s'il est passé au préalable dans l'état représenté par O_2.

Dans l'application il existe les dépendances chronologiques suivantes :

DEPENDANCES CHRONOLOGIQUES ENTRE C-OBJETS

COMPREV	PRECEDE	DEMANN
LIGPREV	PRECEDE	DEMMLIG
EL-ALLOC	PRECEDE	ELMSERV
ELMSERV	PRECEDE	ELMSTAL
ELMSTAL	PRECEDE	COLIS
ARTICLE	PRECEDE	ENTR-MAG
URGMSERV	PRECEDE	COLISURG

III - TESTS DE VALIDITE

- les programmes CONTROL 1 : vérification de la complètude des C-CLASSES
  CONTROL 2 : vérification de la cohérence du sous-schéma de la dynamique
  CONTROL 5 : contrôle de la non redondance du sous-schéma de la dynamique

ne sortent pas de message d'erreurs.

- CONTROL 2 : vérification de la cohérence du sous-schéma des données sort le message suivant :

```
$go control2
LE CONSTITUANT   NFAC    DEVRAIT ETRE UNE CLE
IL APPARAIT DANS  ELFAC     FACURG
```

Ce message fait apparaître l'oubli de la description de l'objet FACTURE.

- CONTROL 4 : Recherche des circuits dans le sous-schéma de la dynamique.
  Produit l'édition des dépendances chronologiques directes entre C-EVENEMENTS et ne signale pas de circuits.

```
EV1     ---->EV2
EV2     ....>EV8
EV4     ....>EV5
EV6     ....>EV8
EV10    ....>EV5
```

- CONTROL 6 : Recherche des trappes dans le sous-schéma de la dynamique donne la liste suivante :

```
$go control6
LE COBJET COMPREV    EST UNE TRAPPE EST CE NORMAL  ?
LE COBJET COMMAN     EST UNE TRAPPE EST CE NORMAL  ?
LE COBJET REFANM     EST UNE TRAPPE EST CE NORMAL  ?
LE COBJET LIGPREV    EST UNE TRAPPE EST CE NORMAL  ?
LE COBJET REFMLIG    EST UNE TRAPPE EST CE NORMAL  ?
LE COBJET ALLOCMAN   EST UNE TRAPPE EST CE NORMAL  ?
LE COBJET EL-ALLOC   EST UNE TRAPPE EST CE NORMAL  ?
LE COBJET ELMSERV    EST UNE TRAPPE EST CE NORMAL  ?
LE COBJET ELFAC      EST UNE TRAPPE EST CE NORMAL  ?
LE COBJET ATT-CAI    EST UNE TRAPPE EST CE NORMAL  ?
LE COBJET STOCKMAG   EST UNE TRAPPE EST CE NORMAL  ?
LE COBJET REFURG     EST UNE TRAPPE EST CE NORMAL  ?
LE COBJET URGMSERV   EST UNE TRAPPE EST CE NORMAL  ?
LE COBJET FACURG     EST UNE TRAPPE EST CE NORMAL  ?
```


- CONTROL 7 : Recherche du rôle des C-OBJETS

Ce programme a été exécuté avant que les relations "UTILISE" entre les textes des opérations des conditions etc ne soient mises en place.

Il permet de vérifier si l'on n'a pas oublié dans le sous-schéma de la dynamique de décrire des opérations portant sur ces C-OBJETS.

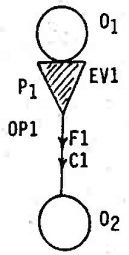
```
LE COBJET CAISSE     NE JOUE AUCUN ROLE    EST CE NORMAL?
LE COBJET ARTICLE    NE JOUE AUCUN ROLE    EST CE NORMAL?
LE COBJET PRIX-ART   NE JOUE AUCUN ROLE    EST CE NORMAL?
LE COBJET SEUIL      NE JOUE AUCUN ROLE    EST CE NORMAL?
LE COBJET CONS       NE JOUE AUCUN ROLE    EST CE NORMAL?
LE COBJET CONSRAIS   NE JOUE AUCUN ROLE    EST CE NORMAL?
LE COBJET CONSADR    NE JOUE AUCUN ROLE    EST CE NORMAL?
LE COBJET CONSTEL    NE JOUE AUCUN ROLE    EST CE NORMAL?
LE COBJET CONSSTAL   NE JOUE AUCUN ROLE    EST CE NORMAL?
LE COBJET TYPCAISS   NE JOUE AUCUN ROLE    EST CE NORMAL?
LE COBJET TYPTRANS   NE JOUE AUCUN ROLE    EST CE NORMAL?
LE COBJET MODTRANS   NE JOUE AUCUN ROLE    EST CE NORMAL?
LE COBJET TXTRANS    NE JOUE AUCUN ROLE    EST CE NORMAL?
LE COBJET GROUPE     NE JOUE AUCUN ROLE    EST CE NORMAL?
LE COBJET GRGEST     NE JOUE AUCUN ROLE    EST CE NORMAL?
LE COBJET GRALLOC    NE JOUE AUCUN ROLE    EST CE NORMAL?
LE COBJET GEST       NE JOUE AUCUN ROLE    EST CE NORMAL?
LE COBJET GESTPAR    NE JOUE AUCUN ROLE    EST CE NORMAL?
```

IV - SPECIFICATION DES CONSTITUANTS DE TYPE TEXTE

Nous donnons ici une spécification des constituants de type TEXTE s'appuyant sur la formalisation effectuée au chapitre 2.

Rappelons que cette formalisation conduit à définir ces constituants comme des applications de profil déterminé.

Par exemple dans le cycle théorique suivant :



Nous avons

$P_1 : O_1 \longrightarrow$ BOOL BOOL{vrai,faux}

$F1 : O_1 \longrightarrow (\mathcal{P}(O_{i_1}) \ldots (\mathcal{P}(O_{i_p}))^* \; O_{ij}$ C-OBJETS utilisés dans OP1

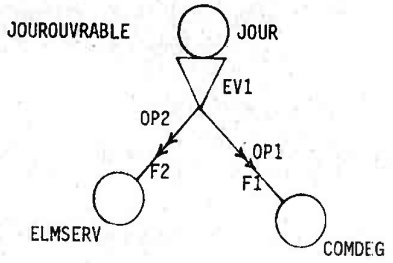
$TOP1 : \mathcal{P}(O_{i_1}) \ldots \mathcal{P}(O_{i_p}) \longrightarrow O_2$

$C1 : O_1 \longrightarrow$ BOOL

Nous présentons ces spécifications cycle dynamique par cycle dynamique.

Nous ne donnons une spécification formelle que pour le premier cycle, pour les autres nous nous contentons de donner le profil des constituants et leur signification.

CHANGEMENT DE JOUR OUVRABLE



F1 : JOUR $\longrightarrow$ $\mathcal{P}$(COMPREV)
TOP1 : COMPREV $\longrightarrow$ COMDEG
F2 : JOUR $\longrightarrow$ $\mathcal{P}$(EL-ALLOC)
TOP2 : EL-ALLOC $\longrightarrow$ ELMSERV

F1 : Pour toutes les commandes prévisionnelles, non annulées, dont la date de livraison correspond à au moins 7 jours au-delà de la date du jour ouvrable.
Soit j ∈ JOUR
F1(j) = {c ∈ COMPREV : DEFMANN[NCP = c.NCP] = ∅ et DIFFDATE(c.DLIVR,j.DJOUR) ⩽ 7}

DIFFDATE : D,D $\longrightarrow$ N
Opération qui calcule le nombre de jours séparant deux dates exprimées de façon classique (an, mois, jour).

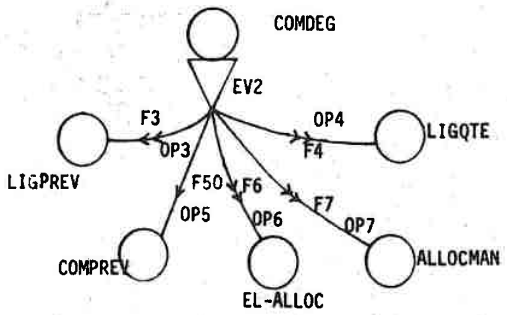
TOP1 : Création d'un n-uplet de COMDEG
TOP1(c) = (n,d)
où n = c.NCP
d = h h: instant d'exécution de l'opération.

F2 : Pour tous les éléments d'allocation, non encore servis, dont la date de mise à servir prévue, correspond à la date du jour ouvrable ou à une date antérieure.

F2(j) : {e ∈ EL-ALLOC:ELMSERV[NELT=e.NELT] = ∅ et
((e.DMSERVIP=j.DJOUR) ou (ANT(e.DMSERVIP,j.DJOUR)))}
ANT : D×D ⟶ BOOL
ANT(d_1,d_2) = vrai si d_1 est une date antérieure à d_2

TOP2 : Création d'un n-uplet de ELMSERV.
TOP2(e) = (n,d)
où | n = e.NELT
| d = h

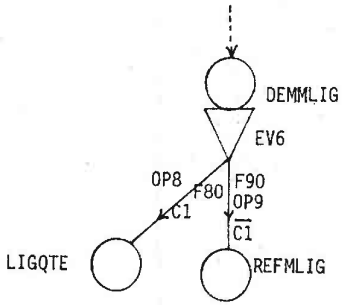
ENTREE D'UNE COMMANDE EN SEMAINE DE DEGAGEMENT



F50 : COMDEG ⟶ COMDEG application identité notée I
TOP5 : COMDEG ⟶ COMPREV
"Création d'une commande prévisionnelle, identique à celle qui vient d'entrer en semaine de dégagement mais livrable dans quatre mois".
F3 : COMDEG ⟶ $\mathcal{P}$(LIGPREV)
TOP3 : LIGPREV ⟶ LIGPREV
"Pour toutes les lignes de commande, non annulées, de la commande qui vient d'entrer en semaine de dégagement, création d'une ligne de commande de la commande prévisionnelle, associée, à livrer dans quatre mois".
F4 : identique à F3
TOP4 : LIGPREV ⟶ LIGQTE
"Calcul de la quantité à livrer initiale d'une ligne de commande prévisionnelle".

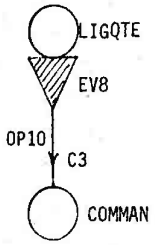
F6 : COMDEG ⟶ $\mathcal{P}$(EL-ALLOC)
TOP6 : EL-ALLOC ⟶ EL-ALLOC
"Pour chaque ligne de commande, non annulée, portant sur un article muni d'une règle d'allocation et dont la quantité en stock est supérieure au seuil toléré et pour chaque élément d'allocation que l'on peut faire dans la ligne, création d'un élément d'allocation".
F7 : COMDEG ⟶ $\mathcal{P}$(LIGPREV)
TOP7 : LIGPREV ⟶ ALLOCMAN
"Pour toutes les lignes de commande, non annulées, portant sur un article non muni d'une règle d'allocation ou tel que sa quantité en stock est inférieure au seuil toléré, création d'une demande d'allocation manuelle".

ARRIVEE D'UNE DEMANDE DE MODIFICATION DE LIGNE DE COMMANDE



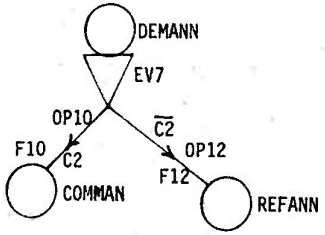
F80, F90 : application identité
C1 : DEMMLIG ⟶ BOOL
"La commande à laquelle appartient la ligne concernée par la demande de modification n'est pas entrée dans la semaine de dégagement".
OP8 : DEMMLIG ⟶ LIGQTE
"Calcul de la nouvelle quantité à livrer".
OP9 : DEMMLIG ⟶ REFMLIG
"Refus de la demande de modification".

ANNULATION D'UNE LIGNE DE COMMANDE



PEV8 : LIGQTE ⟶ BOOL
"La nouvelle quantité à livrer est nulle".
F101 : LIGQTE ⟶ COMPREV
C3 : LIGQTE ⟶ BOOL
"Toutes les quantités à livrer des autres lignes de commande de la commande à laquelle appartient la ligne annulée sont nulles également".
OP10 : COMPREV ⟶ COMMAN
"Annulation de la commande".

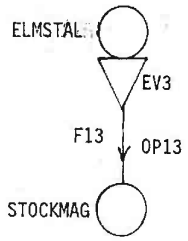
DEMANDE D'ANNULATION D'UNE COMMANDE



F10 : DEMANN ⟶ COMPREV
F12 : DEMANN ⟶ COMPREV
C2 : DEMANN ⟶ BOOL
"La commande concernée par la demande d'annulation n'est pas encore entrée dans la semaine de dégagement".

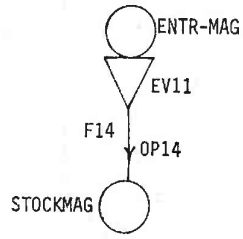
OP10 : COMPREV ⟶ COMMAN
"Annulation de la commande"
OP12 : COMPREV ⟶ REFANN
"Refus de demande d'annulation de commande".

MISE EN STALLE D'UN ELEMENT D'ALLOCATION



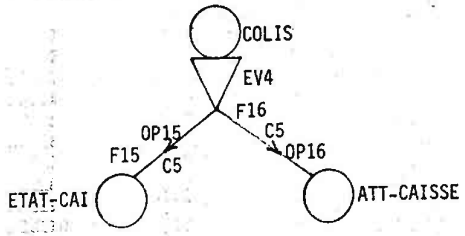
F13 : ELMSTAL ⟶ EL-ALLOC,ARTICLE
OP13 : EL-ALLOC,ARTICLE ⟶ STOCKMAG
"Mise à jour de la quantité en stock de l'article concerné par l'élément d'allocation mis en stalle".

ENTREE DE PRODUIT EN MAGASIN



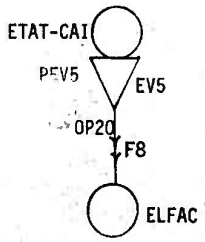
F14 : application identité
OP14 : ENTR-MAG ⟶ STOCKMAG
"Mise à jour de la quantité en stock de l'article pour lequel une entrée en magasin vient d'avoir lieu".

EMBALLAGE D'UN ELEMENT D'ALLOCATION



C5 : COLIS ⟶ BOOL
"La caisse dans laquelle est placé le colis est dans l'état 'disponible' ".
F15 : COLIS ⟶ CAISSE
OP15 : CAISSE ⟶ ETAT-CAI
"Modification de l'état de la caisse dans laquelle est placé le colis - Passage à l'état 'en cours de remplissage' ".
F16 : COLIS ⟶ CAISSE,CONS
OP16 : CAISSE,CONS ⟶ ATT-CAISSE
"Attribution de la caisse, dans laquelle est placé le colis, au concessionnaire auquel est destiné la commande à laquelle appartient le colis".

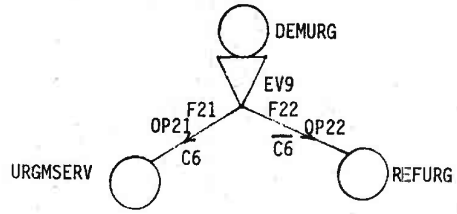
CAISSE PLEINE



PEV5 : ETAT-CAI ⟶ BOOL
" Passage d'une caisse dans l'état plein".

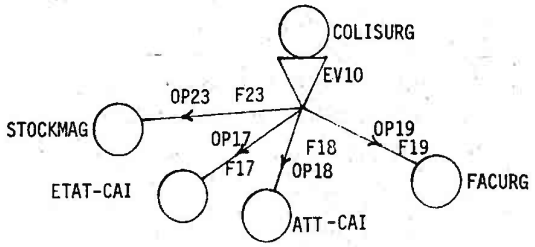
PEV5(e) : e.ETAT ='plein'
F8 : ETAT-CAI ⟶ $\mathcal{P}$(EL-ALLOC)
OP20 : EL-ALLOC ⟶ ELFAC
"Pour tous les éléments d'allocation correspondant aux colis placés dans la caisse, passage à l'état facturé".

ARRIVEE DE DEMANDE DE COMMANDE URGENTE



C6 : DEMURG ⟶ BOOL
"La quantité en stock de l'article concerné par la demande est supérieure au seuil toléré et il n'existe pas de commandes prévisionnelles comportant cet article livrables dans les huit jours au concessionnaire passant la commande urgente".
F21 : application identité
OP21 : DEMURG ⟶ URGMSERV
"Acceptation de la commande urgente et mise à servir de cette commande".
F22 : application identité
OP22 : DEMURG ⟶ REFURG
"Refus de la demande de commande urgente".

EMBALLAGE D'UNE COMMANDE URGENTE



F23 : COLISURG ⟶ DEMURG,ARTICLE

OP23 : DEMURG,ARTICLE ⟶ STOCKMAG

"Mise à jour de la quantité en stock de l'article concerné par la commande urgente".

F17 : COLISURG ⟶ CAISSE

OP17 : CAISSE ⟶ ETAT-CAI

"Passage de l'état de la caisse dans laquelle est placé le colis à l'état 'expédié' ".

F18 : COLISURG ⟶ CAISSE,CONS

OP18 : CAISSE,CONS ⟶ ATT-CAI

"Attribution de la caisse, dans laquelle est placé le colis, au concessionnaire auquel il est destiné".

F19 : application identité

OP19 : COLISURG ⟶ FACURG

"Facturation de la commande correspondant au colis".

V - EXEMPLES DE RENSEIGNEMENTS QUE L'ON PEUT OBTENIR A PARTIR DU SCHEMA CONCEPTUEL ET DU SI

Dans cette partie nous nous plaçons dans l'hypothèse d'une implantation du SI sous la forme d'une base de données à partir du schéma conceptuel et nous donnons quelques exemples d'interrogations qu'il serait possible d'effectuer sur cette base de données.

Nous les avons classés en

- questions concernant le fonctionnement passé ou présent de l'organisation
- questions concernant la situation présente de l'organisation
- questions concernant le passé de l'organisation et présentant un intérêt pour le contrôle de son fonctionnement et de ses objectifs
- questions concernant l'origine d'une situation passée
- questions concernant le futur.

V-1.- Questions sur le fonctionnement posées sur la métabase

La représentation des lois de fonctionnement de l'organisation dans la métabase permet de poser des questions du type suivant :

- Conséquence d'un événement

Ex : Conséquence de l'entrée d'une commande en semaine de dégagement.

```
NOM DU C-EVENEMENT
?ev2

        EV2     ENTREE COM EN SEMAINE DEGAGT
                DECLENCHE :

        OP3     CREATION DE LIG-COM PREVI
        OP4     CALCUL DE LA QUANTITE A LIVRER
        OP5     CREATION DE COM PREVI
        OP6     ALLOCATION
        OP7     DEMANDE D ALLOCATION MANUELLE
```

- Rôle d'un C-OBJET

Ex : rôle du C-OBJET ETAT-CAI

```
NOM DU C-OBJET
?etat-cai

EST CONSTATE PAR :
        EV5       CAISSE PLEINE
EST MODIFIE PAR :
        OP15      REQUISITION D UNE CAISSE
        OP17      EXPEDITION DE CAISSE URGENTE
```

- Valeur actuelle ou passée des constituants de type texte

V-2.- Questions sur la situation présente

- Quelle est la situation de la commande prévisionnelle n° XXX ?

/ Recherche des occurrences des relations COMDEG, EL-ALLOC, ELMSERV , ELMSTAL , COLIS et ELFAC ayant pour valeur de NCP XXX
Suivant la présence ou l'absence d'occurrences dans ces relations on déduit la situation de la commande prévisionnelle - non entrée dans la semaine de dégagement
- allouée
- servie totalement ou partiellement
- livrée totalement ou partiellement /

- Quel est l'état de la caisse n° YY ? Si elle est expédiée, chez quel concessionnaire devrait-elle être ?

/ Recherche de la valeur de ETAT dans la dernière occurrence créée de la relation ETAT-CAI ayant pour valeur de NOCAISSE YY. Si état = expédiée, recherche de la valeur de NCONS dans la dernière occurrence créée de la relation ATT-CAI ayant pour valeur de NOCAISSE YY /

V-3.- Questions sur le passé

Questions permettant de contrôler la prévision effectuée en matière d'approvisionnement des concessionnaires

- Quel est le nombre de commandes urgentes passées par un concessionnaire X pendant une période ?

/ Recherche des occurrences de la relation DEMURG où NCONS = X et où DARURG appartient à la période /

- Quel est le nombre de commandes urgentes passées sur un article Y pendant une période ?

/ Recherche des occurrences de la relation DEMURG où NART = Y et ou DARURG appartient à la période /

- Quel est le nombre de demandes de modifications de commandes prévisionnelles passées par un concessionnaire X ou pour un article donné ?

/ Recherche des occurrences des relations DEMMLIG et DEMANN) où le constituant date a une valeur appartenant à la période et concernant, des lignes de commandes ou des commandes passées par ce concessionnaire ou l'article donné./

Question permettant de contrôler la livraison

Quel est le délai moyen mis par les caisses d'un certain type pour parvenir à un concessionnaire ?

/ Recherche et calcul à partir des relations ETAT-CAI et ATT-CAI /

Questions permettant de contrôler la gestion des commandes au siège lui-même

- Toute commande est-elle effectivement livrée dans sa semaine de dégagement ?

  / Comparaison de la date de facturation dans la relation ELFAC à la date d'entrée dans la semaine de dégagement de la relation COMDEG, pour tous les éléments d'allocation d'une même commande /

- Quel est le nombre de colis confectionnés en moyenne sur une journée ?

  / Calcul à partir de la relation COLIS /

- Quel est le temps pris en moyenne par le gestionnaire d'articles pour réaliser l'allocation manuelle ?

  / Comparaison entre la date de création d'une occurrence d'ALLOC-MAN et celles des occurrences concernant la même ligne de commande de la relation EL-ALLOC /

V-4.- Questions sur l'origine d'une situation passée

- La caisse Y a-t-elle été attribuée à la date X pour une commande urgente ou pour une commande prévisionnelle ?

  / Recherche si la valeur de la clé YX de la relation ATT-CAI apparaît dans une occurrence de la relation OP18 ou dans une occurrence de la relation OP16 /

- Est-ce que le passage en-dessous d'un seuil d'un produit Y à la date X a été causé par la livraison d'une commande urgente ?

  / Recherche s'il existe dans la relation OP23 une occurrence où NO21 = YX /

- Quels ont été tous les éléments d'allocation créés à une date X par suite d'une allocation manuelle ?

  / Recherche des occurrences de la relation EL-ALLOC ayant leur date de création égale à X et n'apparaissant pas dans la relation OP6 /

- Qu'elle a été sur une certaine période les causes de l'évolution de la quantité en stock en magasin d'un certain produit ?

  / Ordonnancement sur le constituant date des occurrences des relations OP13 , OP14 et OP23 concernant ce produit et recherche pour chaque opération de son événement déclencheur dans les relations DECL5, DECL20 et DECL19.

V -5.- Questions concernant le futur

- Quelles sont les livraisons prévues pour un certain article à une certaine date X ?

  / Recherche des occurrences de la relation LIGPREV concernant cet article et concernant des commandes (Relation COMDEG) dont la date de livraison est égale à X /

- Si la production d'un certain article est de n unités par semaine, quel sera le nombre d'articles que l'on pourra livrer chaque semaine pour des commandes urgentes ?

  / Calcul à partir de la quantité en stock actuel (relation STOCKMAG) et des occurrences de la relation LIGPREV concernant cet article et livrables la même semaine /

CONCLUSION

Les exemples de question que nous venons de présenter montrent l'intérêt pour les gestionnaires de représenter dans le SI :

- les règles de gestion sous la forme de relations particulières, pouvant être consultées comme n'importe quelles autres relations représentant des phénomènes réels
- l'application de ces règles au cours du temps, pour retrouver l'origine d'une situation donnée
- l'histoire complète de l'organisation afin de contrôler sur une large période son fonctionnement et prendre en toute connaissance des décisions concernant la modification de règles de gestion.

STRUCTURE LOGIQUE DE LA METABASE (1) : La signification des constituants est donnée au paragraphe IV-1 du chapitre 6.

```
00001 $BASE 50
00002       ENTITE 30 CLASSE
00003       DEBUT
00004          NCC MOT 8
00005          LCC MOT 48
00006          DCC DECIMAL 6  DE 800000 A 991231
00007          DSC DECIMAL 6  DE 800000 A 991231
00008           R-PERM REFERE UN OBJET
00009          CH-OB CHAINE 4 TOUT OBJET  PAR R-CL
00010       FIN
00011       ENTITE 150 OBJET
00012       DEBUT
00013          NCOB MOT 8
00014          LCOB MOT 48
00015          TYPOB CODEE (3 4) (PERM VAR SUP)
00016          DCOB DECIMAL 6 DE 800000 A 991231
00017          DSOB DECIMAL 6 DE 800000 A 991231
00018          CONSTANCE CODEE (2 1) (0 N)
00019          CLEF REFERE UN CONSTOB
00020          R-CL REFERE UNE CLASSE PAR CH-OB
00021          CH-COMP CHAINE 4 TOUT  COMPOB PAR R-OB
00022          CH-OP CHAINE 4 TOUTE OPER PAR R-OB
00023          CH-EV CHAINE 4 TOUT EVENT PAR R-OB
00024          CH-UT-PRED CHAINE 4 TOUT UT-PRED PAR R-OB
00025          CH-UT-COND CHAINE 4 TOUT UT-COND PAR R-OB
00026          CH-UT-FAC  CHAINE 4 TOUT UT-FAC  PAR R-OB
00027          CH-UT-TEXT CHAINE 4 TOUT UT-TEXT PAR R-OB
00028          CH-CHRON2  CHAINE 4 TOUT DEP-CHRON PAR R-OB1
00029          CH-CHRON1  CHAINE 4 TOUT DEP-CHRON PAR R-OB2
00030          CH-CONTINT CHAINE 8 TOUT CONTINT PAR R-OB
00031    FIN
00032       ENTITE  50 EVENT
00033       DEBUT
00034         NCEV MOT 8
00035         LCEV MOT 48
00036         DCE DECIMAL 6 DE 800000 A 991231
00037         DSE DECIMAL 6 DE 800000 A 991231
00038         CH-DECL CHAINE 4 TOUT DECL PAR R-EV
00039         CH-PRED CHAINE 4 TOUT PRED PAR R-EV
00040         R-OB REFERE UN OBJET PAR CH-EV
00041         CH-GHEV2 CHAINE 4 TOUT GH-EV PAR R-EV1
00042         CH-GHEV1 CHAINE 4 TOUT GH-EV PAR R-EV2
00043           CH-CONSTEV CHAINE 4 TOUT CONSTEV PAR R-EV
00044           CH-CONTINT CHAINE 4 TOUT CONTINT PAR R-EV
00045       FIN
00046       ENTITE 150 PRED
00047       DEBUT
00048         DMP DECIMAL 6 DE 800000 A 991231
00049         PRED MOT 20
00050         SPRED TEXTE 200
00051         R-EV  REFERE UN EVENT PAR CH-PRED
00052         CH-UT-PRED CHAINE 4 TOUT UT-PRED PAR R-PRED
00053       FIN
```

STRUCTURE LOGIQUE DE LA METABASE (2)

```
00054        ENTITE 200 OPER
00055        DEBUT
00056          NCOP MOT 8
00057          LCOP MOT 48
00058          DCP DECIMAL 6 DE 800000 A 991231
00059          DSP DECIMAL 6 DE 800000 A 991231
00060          R-OB REFERE UN OBJET PAR CH-OP
00061          CH-DECL CHAINE 4 TOUT DECL PAR R-OP
00062          CH-TEXT CHAINE 4 TOUT TEXT PAR R-OP
00063          CH-CONSTOP CHAINE 4 TOUT CONSTOP PAR R-OP
00064        FIN
00065        ENTITE 200 DECL
00066        DEBUT
00067          DCD DECIMAL 6 DE 800000 A 991231
00068          DSD DECIMAL 6 DE 800000 A 991231
00069          R-OP REFERE UN OPER PAR CH-DECL
00070          R-EV REFERE UN EVENT PAR CH-DECL
00071          CH-COND CHAINE 4 TOUT COND PAR  R-DECL
00072          CH-FAC CHAINE 4 TOUT FAC PAR R-DECL
00073        FIN
00074        ENTITE 400 COND
00075        DEBUT
00076          CONDITION  MOT 20
00077          DMCOND DECIMAL 6 DE 800000 A 991231
00078          SCOND TEXTE 200
00079          R-DECL REFERE UN DECL PAR CH-COND
00080          CH-UT-COND CHAINE 4 TOUT UT-COND PAR R-COND
00081        FIN
00082        ENTITE 400 FAC
00083        DEBUT
00084          FACTEUR  MOT 20
00085          DMFAC DECIMAL 6 DE 800000 A 991231
00086          MUL CODEE (2 4) (UN MULT)
00087          SFAC TEXTE 300
00088          R-DECL REFERE UN DECL PAR CH-FAC
00089          CH-UT-FAC CHAINE 4 TOUT UT-FAC PAR R-FAC
00090        FIN
00091        ENTITE  300 TEXT
00092        DEBUT
00093          TEXT-OP MOT 20
00094          DMT DECIMAL 6 DE 800000 A 991231
00095          STEXT TEXTE 500
00096          R-OP REFERE UNE OPER PAR CH-TEXT
00097          CH-UT-TEXT CHAINE 4 TOUT UT-TEXT PAR R-TEXT
00098        FIN
00099        ENTITE  900  UT-TEXT
00100        DEBUT
00101          R-TEXT REFERE UN TEXT PAR CH-UT-TEXT
00102          R-OB REFERE UN OBJET PAR CH-UT-TEXT
00103        FIN
00104        ENTITE 800 UT-FAC
00105        DEBUT
00106          R-FAC REFERE UN FAC PAR CH-UT-FAC
00107          R-OB  REFERE UN OBJET PAR CH-UT-FAC
00108        FIN
```


STRUCTURE LOGIQUE DE LA METABASE (3)

```
00109        ENTITE 800 UT-COND
00110        DEBUT
00111          R-COND REFERE UN COND PAR CH-UT-COND
00112          R-OB   REFERE UN OBJET  PAR CH-UT-COND
00113        FIN
00114        ENTITE 800 UT-PRED
00115        DEBUT
00116          R-PRED REFERE UN PRED PAR CH-UT-PRED
00117          R-OB REFERE UN OBJET PAR CH-UT-PRED
00118        FIN
00119        ENTITE  300 DEP-CHRON
00120        DEBUT
00121          DCHOB DECIMAL 6 DE 800000 A 999999
00122          DSHOB DECIMAL 6 DE 800000 A 999999
00123          R-OB1 REFERE UN OBJET PAR CH-CHRON2
00124          R-OB2 REFERE UN OBJET PAR CH-CHRON1
00125        FIN
00126        ENTITE 250 CH-EV
00127        DEBUT
00128           DCHEV DECIMAL 6 DE 800000 A 999999
00129            DSHEV DECIMAL 6 DE 800000 A 999999
00130           TYPDEP CODEE (2 6) (FORTE FAIBLE)
00131          R-EV1 REFERE UN EVENT PAR CH-CHEV2
00132          R-EV2 REFERE UN EVENT PAR CH-CHEV1
00133        FIN
00134        ENTITE  600  CONSTOB
00135        DEBUT
00136           NCONSTOB  MOT  8
00137           FORMAT   TEXTE  60
00138           LCONSTOB1  MOT 30
00139            LCONSTOB2 MOT 30
00140           TYPCONST CODEE (3 5) (DATE PROPR TEXTE)
00141           DCCONS DECIMAL 6 DE 800000  A 991231
00142           DSCONS DECIMAL 6 DE 800000  A 991231
00143           CH-COMPOB CHAINE 4 TOUT COMPOB PAR R-CONSTOB
00144           CH-CLE1 CHAINE 4 TOUT COMPOSCLE PAR R-CLE2
00145           CH-CLE2 CHAINE 4 TOUT COMPOSCLE PAR R-CLE1
00146           FIN
00147        ENTITE  800 COMPOB
00148        DEBUT
00149           R-OB REFERE UN OBJET PAR CH-COMP
00150           R-CONSTOB REFERE UN CONSTOB PAR  CH-COMPOB
00151        FIN
00152        ENTITE 300 COMPOSCLE
00153        DEBUT
00154           R-CLE1 REFERE UN CONSTOB PAR CH-CLE2
00155           R-CLE2 REFERE UN CONSTOB PAR CH-CLE1
00156        FIN
00157        ENTITE 30 CONTINT
00158         DEBUT
00159           NCONTINT DECIMAL 2 DE 1 A 30
00160           LCONTINT TEXTE 500
00161            DCONTINT DECIMAL 6 DE  800000 A 999999
00162           DSCONTINT DECIMAL 6 DE 800000 A 999999
00163           R-OB REFERE UN OBJET PAR CH-CONTINT
00164           R-EV REFERE UN EVENT PAR CH-CONTINT
00165          FIN
```

STRUCTURE LOGIQUE DE LA METABASE (fin)

```
00166        ENTITE 50 CONSTOP
00167         DEBUT
00168           NCONSTOP MOT 8
00169           LCONSTOP MOT 48
00170           CONSTOP DECIMAL 6 DE 800000 A 999999
00171            R-OP REFERE UN OPER PAR CH-CONSTOP
00172          FIN
00173         ENTITE 50 CONSTEV
00174           DEBUT
00175           NCONSTEV MOT 8
00176           LCONSTEV MOT 48
00177            DCONSTEV DECIMAL 6 DE 800000 A 999999
00178            R-EV  REFERE UN EVENT PAR CH-CONSTEV
00179            FIN
```


EXEMPLES DE PROGRAMMES DE CREATION DU SCHEMA CONCEPTUEL

CCLASSE : création des C-CLASSES
COBJET : création des C-OBJETS
CONST   : création des constituants des C-OBJETS

```
*f/cclasse/
00001*1,$ 1
00001 D BUFFER = DEBUT
00002 N1 MOT 8
00003 N3 MOT 6
00004 N2 MOT 48
00005 FIN
00006 FAIRE M YZO = U
00007 LIRE $DON1 DANS BUFFER
00008 I N3 DE BUFFER
00009 M Y1 = N3 DE BUFFER
00010 SI Y1 = 0 ALORS SORTIE SINON
00011 G UNE CLASSE X1
00012 M Y2 = N3 DE BUFFER
00013 M NCC DE X1 = N1 DE BUFFER
00014 M LCC DE X1 = N2 DE BUFFER
00015 M DCC DE X1 = Y2
00016 REFAIRE
00017 FIN
00018 FIN
FIN DE FICHIER
*f/cobjet/
00001*1,$ 1
00001 D BUFFER = DEBUT
00002 N1 MOT 8
00003 N3 MOT 6
00004 N4 MOT 4
00005 N5 MOT 8
00006 N2 MOT 48
00007 FIN
00008 FAIRE M YZO = U
00009 LIRE $DON2 DANS BUFFER
00010 M Y1 = N3 DE BUFFER
00011 SI Y1 = 0 ALORS SORTIE SINON
00012 G UN OBJET X1
00013 M Y2 = N3 DE BUFFER
00014 M NCOB DE X1 = N1 DE BUFFER
00015 M LCOB DE X1 = N2 DE BUFFER
00016 M DCOB DE X1 = Y2
00017 M TYPOB DE X1 = N4 DE BUFFER
00018 M R-CL DE X1 = UNE CLASSE X2  AYANT NCC = N5 DE BUFFER
00019 SI TYPOB DE X1  = 'PERM' ALORS
00020                   M R-PERM DE X2 = X1 FIN
00021 REFAIRE
00022 FIN
00023 FIN
FIN DE FICHIER
*f/const/
00001*1,$ 1
00001 D BUFFER = DEBUT
00002 N1 MOT 8
00003 N3 MOT 6
00004 N4 MOT 1
00005 N5 MOT 1
00006 N2 MOT 30
00007 N6 MOT 34
00008 FIN
```

PROGRAMME CONST (suite)

```
00009 FAIRE M YZO = U
00010 LIRE $DON3 DANS BUFFER
00011 M Y1 = N3 DE BUFFER
00012 SI Y1 = 0 ALORS SORTIE SINON
00013 M X1 = UN OBJET AYANT NCOB = N1 DE BUFFER
00014   SI X1 = U ALORS
00015                I N1 DE BUFFER
00016 I 'OBJET INEXISTANT'
00017                FAIRE LIRE $DON3 DANS BUFFER
00018                 M Y1 = N3 DE BUFFER SI Y1 = 0 ALORS
00019                                        SORTIE SINON REFAIRE
00020                  FIN FIN
00021        REFAIRE
00022   FIN
00023 M YZO = U
00024 LIRE $DON3 DANS BUFFER M Y1 = N3 DE BUFFER
00025 SI TYPOB DE X1 = 'PERM' ALORS
00026                   M X3 = UN CONSTOB AYANT NCONSTOB = N1 DE BUFFER
00027                     SI X3 = U ALORS GCONSOB FIN  M CLEF DE X1 = X3
00028                   M X9 = X3
00029                     G UN COMPOB X4 M R-OB DE X4 = X1
00030                     M R-CONSTOB DE X4 = X3
00031                     SINON  CLEVAR   GCONSOB
00032                   G UN COMPOB X4
00033                     M R-OB DE X4 = X1
00034                     M R-CONSTOB DE X4 = X3
00035 G UN COMPOSCLE X5
00036 M R-CLE1 DE X5 = X9
00037 M R-CLE2 DE X5 = X3
00038 FIN
00039      FAIRE M YZO = U
00040       LIRE $DON3 DANS BUFFER
00041       M Y1  = N3 DE BUFFER
00042        SI Y1 = 0 ALORS SORTIE SINON
00043            M X3 = UN CONSTOB AYANT NCONSTOB = N1 DE BUFFER
00044             SI X3 = U ALORS GCONSOB
00045             SINON
00046              FIN
00047            G UN COMPOB X4
00048            M R-OB DE X4 = X1
00049            M R-CONSTOB DE X4 = X3
00050            M Y2 = N5 DE BUFFER
00051             SI Y2 = 1 ALORS
00052                 G UN COMPOSCLE X5
00053                  M R-CLE1 DE X5 = X9
00054                  M R-CLE2 DE X5 = X3
00055              SINON
00056              FIN
00057       REFAIRE
00058     FIN
00059   FIN
00060 REFAIRE
00061   FIN
00062 FIN
FIN DE FICHIER
*end
```


PROGRAMME CONST (fin)

```
*f/gconsob/
00001*1,$ 1
00001 :EXP
00002 G UN CONSTOB X3
00003 M NCONSTOB DE X3 = N1 DE BUFFER
00004  M LCONSTOB1 DE X3 = N2 DE BUFFER
00005 M LCONSTOB2 DE X3 = N6 DE BUFFER
00006 M Y1 = N3 DE BUFFER
00007 M DCCONS DE X3 = Y1
00008 SI N4 DE BUFFER = '0' ALORS
00009                    M TYPCONST DE X3 = 'PROPR'
00010                    SINON M TYPCONST DE X3 = 'DATE'
00011                FIN
FIN DE FICHIER
*f/clevar/
00001*1,$ 1
00001 :EXP
00002 D Z1 = DEBUT
00003 Z01   MOT 2
00004 Z02 MOT 6
00005 FIN
00006 D Z2 = DEBUT
00007 Z03 MOT 16
00008 Z04 MOT 8
00009 FIN
00010 M X6 = R-CL DE X1
00011 POUR TOUT CH-OB X8  DE X6
00012 SI TYPOB DE X8 = 'PERM' ALORS M X7 = X8 FIN
00013 FIN
00014 M X8 = CLEF DE X7
00015 G UN COMPOB X4
00016 M R-OB DE X4 = X1
00017 M R-CONSTOB DE X4 = X8
00018 G UN CONSTOB X9
00019 M Z01   = 'NO'
00020 M Y2 = NUMDE X1
00021 M Z02   = Y2
00022 M NCONSTOB DE X9 = Z1
00023 M Z03  = 'CLE SYNONYME DE '
00024 M Z04  = NCOB DE X1
00025 M LCONSTOB1 DE X9 = Z2
00026 M DCCONS DE X9 = Y1
00027 M CLEF DE X1 = X9
00028 G UN COMPOSCLE X5
00029 M R-CLE1 DE X5 = X9
00030 M R-CLE2 DE X5 = X8
FIN DE FICHIER
```

EXEMPLES DE PROGRAMMES DE DOCUMENTATION

ESCHDONN : édition du sous-schéma des données

EXPLORDYN : édition du sous-schéma de la dynamique

```
00001*f/eschdonn/
00001*1,$ l
00001 I ´       EDITION DU SOUS SCHEMA DES DONNEES´
00002 M BUFFER = U  ECRIRE
00003 POUR TOUTE CLASSE X1 AYANT DSC = U
00004   M BUFFER = U ECRIRE
00005 M Z1 = NCONSTOB DE CLEF DE R-PERM DE X1
00006   I (2) ´CLASSE  ´
00007   I (13) NCC I (22) ´  CLE  ´ I (28) Z1
00008 I (36)  ´  ´ I  (40) LCC
00009 ECRIRE
00010 M BUFFER = U ECRIRE
00011      POUR TOUT CH-OB X4 AYANT DSOB = U
00012        M BUFFER = U
00013       I (2) ´C-OBJET  ´ I (+2) TYPOB I (+0) ´   ´
00014        I (+2) NCOB I (+2) ´(´
00015 M X3 = CLEF DE X4 I (+0) NCONSTOB DE X3
00016 M Y6 = 0
00017           POUR TOUT CH-COMP X5
00018            SI R-CONSTOB DE X5 = X3 ALORS SINON
00019           I(+0) ´,´
00020             I (+0) NCONSTOB DE R-CONSTOB DE X5
00021 M Y6 = Y6 + 1
00022          SI Y6 > 5 ALORS ECRIRE I (0 131) ´ ´
00023                      I(34) ´,´ M Y6 = 0 FIN
00024            FIN
00025           FIN
00026       I (+0) ´)´
00027              ECRIRE
00028 FIN
00029 FIN
FIN DE FICHIER
```


PROGRAMME EXPLORDYN : Parcours du graphe du sous-schéma de la dynamique

```
*F/EXPLORDYN/
00001*1,$ L
00001    INIT
00002 FAIRE
00003      CHOIXOB
00004    SI X1 = U ALORS SORTIE
00005              SINON M Z-O (Y8) = NCOB DE X1
00006                  M Y8 = Y8 + 1
00007            POUR TOUT CH-EV X2 DE X1
00008            M Y9 = Y9 + 1
00009                M Z-E (Y9) = NCEV DE X2
00010            FIN
00011         CONS
00012   REFAIRE
00013 FIN
00014 FIN
FIN DE FICHIER
*END
$MACRO
*F/INIT/
00001*1,$ L
00001 :EXP
00002 M YZO = U
00003 D YZO = DEBUT
00004 ESPACE 80
00005     TABLEAU 20 Z-O MOT 8
00006    TABLEAU 10 Z-E MOT 8
00007    FIN
00008 M Y8 = 1
00009 M Y9 = 0
00010 M Y1 = 0
FIN DE FICHIER
*F/CHOIXOB/
00001*1,$ L
00001 :EXP
00002 SI Y1 = 0 ALORS M X1 = UN OBJET AYANT EXISTE
00003          UN CH-EV ET PAS UN CH-OP ET PAS
00004      UN CH-CHRON2
00005         M Y1 = Y1 + 1
00006         SI X1 = U ALORS  M X1 = UN OBJET AYANT EXISTE
00007             UN CH-EV ET PAS UN CH-OP
00008    FIN
00009 SI X1 = U ALORS M X1 = UN OBJET AYANT EXISTE
00010                         UN CH-EV FIN
00011 SINON
00012 POUR TOUT OBJET X1   AYANT EXISTE UN CH-EV
00013            ET PAS UN CH-OP
00014    RECHTO DE X1 F
00015    SI Y2 = 0 ALORS   SUIVANT SINON SORTIE
00016 FIN
00017 FIN
00018 SI X1 = U ALORS  POUR TOUT OBJET X1 AYANT EXISTE
00019             UN CH-EV   RECHTO DE X1 F
00020               SI Y2 = 0 ALORS SUIVANT SINON SORTIE
00021               FIN
00022             FIN
00023    FIN
00024 FIN
```

PROGRAMME EXPLORDYN (suite)

```
*F/CONS/
00001*1,$ L
00001 :EXP
00002 FAIRE
00003  SI Y9 = 0 ALORS SORTIE
00004       SINON M X9 = UN EVENT AYANT NCEV = Z-E(Y9)
00005       CONSDIRECTE DE X9 F
00006 M Z-E (Y9) = U
00007   M Y9 = Y9 - 1
00008    POUR TOUT CH-DECL X5 DE X9
00009   M X4 = R-OB DE R-OP DE X5
00010 SI EXISTE UN CH-EV DE X4 ALORS
00011  RECHTO DE X4 F
00012  SI Y2 = 1
00013       ALORS M Z-O (Y8) = NCOB DE X4
00014            M Y8 = Y8 + 1
00015            POUR TOUT CH-EV X6 DE X4
00016              M Y9 = Y9 + 1
00017              M Z-E (Y9) = NCEV DE X6
00018            FIN
00019     FIN
00020  FIN
00021 FIN
00022 REFAIRE
00023 FIN
00024 FIN
FIN DE FICHIER
*F/RECHTO/
00001*1,$ L
00001 DE : F
00002 :EXP
00003 M Y3 = 1  M Y2 = 1
00004 FAIRE SI Z-O (Y3) = U
00005   ALORS SORTIE
00006  SINON  SI Z-O (Y3) = NCOB DE :1:
00007    ALORS   M Y2 = 0 SORTIE
00008   SINON M Y3 = Y3 + 1
00009   REFAIRE
00010   FIN
00011 FIN
00012 FIN
FIN DE FICHIER
*F/CONSDIRECTE/
00001*1,$ L
00001   DE :  F
00002 :EXP
00003 M BUFFER = U
00004 I(1) NCOB DE R-OB DE :1:
00005 I(+0) '<----'
00006 I (+0) NCEV DE :1:
00007 I (+0) '---->'
00008 POUR TOUT CH-DECL X3 DE :1:
00009 I (+0) NCOP DE R-OP DE X3
00010 I (+0) ','
00011 FIN
00012 ECRIRE M BUFFER = U ECRIRE
00013 I (0) '
00014 POUR TOUT CH-DECL X3 DE :1:
00015 I (+0) NCOB DE R-OB DE R-OP DE X3
00016 I (+0) ','
```


PROGRAMME CONTROL 4 : Recherche et édition des circuits dans le sous-schéma de la dynamique - Exemple de programme de contrôle

```
00001*F/CONTROL4/
00001*1,$ L
00001 D YZO  = DEBUT
00002         ESPACE 80
00003          TABLEAU 50 Y-P DECIMAL 2
00004          TABLEAU 50 Y-E DECIMAL 1
00005         FIN
00006 EXEC CONSGHEV
00007 EXEC INITCIR
00008 POUR TOUT GH-EV X1
00009   M Y2 = 1
00010   M Y1 = NUMDE X1
00011    M Y-P (Y2) = Y1
00012 M Y-E (Y1) = 1
00013    M Y3 = Y1
00014      FAIRE
00015          SI Y2 = 0 ALORS SORTIE
00016                  SINON EXEC RECHCIRCUIT
00017                          REFAIRE
00018           FIN
00019     FIN
00020  FIN
FIN DE FICHIER
*F/CONSGHEV/
00001*1,$ L
00001  S TOUT GH-EV
00002  I 'DATE DU JOUR AN MOIS JOUR'
00003 M Z1 = U M Z1 = EXT M Y1 = Z1
00004 POUR TOUT EVENT X1 AYANT DSE = U
00005    POUR TOUT CH-DECL X2 AYANT DSD = U ; DE X1
00006      M X3 = R-OP DE X2
00007      M X4 = R-OB DE X3
00008      POUR TOUT CH-EV X5 DE X4
00009         G UN GH-EV X6 DE XO
00010         SI CONDITION DE UN CH-COND DE X2 = 'TOUJOURS'
00011               ET PRED   DE UN CH-PRED DE X5 = 'CREATION'
00012              ALORS M TYPDEP DE X6 = 'FORTE'
00013               SINON M TYPDEP DE X6 = 'FAIBLE'
00014        FIN
00015        M DCHEV DE X6 = Y1
00016        M R-EV1 DE X6 = X1
00017        M R-EV2 DE X6 = X5
00018       FIN
00019   FIN
00020 FIN
FIN DE FICHIER
*F/INITCIR/
00001*1,$ L
00001 M YZO = U
00002 D YZO = DEBUT
00003        ESPACE 80
00004        TABLEAU 50 Y-P DECIMAL 2
00005        TABLEAU 50 Y-E DECIMAL 1
00006       FIN
00007 M Y1 = 1
00008 FAIRE SI Y1 = 51 ALORS SORTIE
00009             SINON M Y-E (Y1) = 0
00010                 M Y1 = Y1 + 1 REFAIRE
00011     FIN
00012 FIN
```

PROGRAMME CONTROL4 (suite)

```
*F/RECHSUCCESSEUR/
00001*1,$ L
00001 D YZ0 = DEBUT
00002       ESPACE 80
00003        TABLEAU 50 Y-P DECIMAL 2
00004         TABLEAU 50 Y-E DECIMAL 1
00005      FIN
00006 D X2 = UN GH-EV
00007 POUR TOUT GH-EV X3 AYANT R-EV1 DE X3 = R-EV2 DE X2
00008     M Y10 = NUMDE X3
00009      SI Y-E (Y10) = 0 ET Y10 > Y3 ALORS M Y8 = 1
00010                                      SORTIE
00011      FIN
00012 FIN
FIN DE FICHIER
*F/TESTCIRCUIT/
00001*1,$ L
00001 D X3 = UN GH-EV D X1 = UN GH-EV
00002 D YZ0 = DEBUT
00003       ESPACE 80
00004        TABLEAU 50 Y-P DECIMAL 2
00005        TABLEAU 50 Y-E DECIMAL 1
00006        FIN
00007 SI R-EV2 DE X3 = R-EV1 DE X1
00008       ALORS I 'IL EXISTE LE CIRCUIT SUIVANT'
00009             M Y5 = 1 M Y4 = 1
00010           M Y7 = Y2 + 1
00011 M BUFFER = U M Y8 = 1
00012             FAIRE  SI Y5 = Y7 ALORS SORTIE
00013                         SINON M Y6 = Y-P (Y5) M X4 = UN GH-EV Y6
00014                               I (+0)   NCEV DE R-EV1 DE X4
00015                               SI TYPDEP DE X4 = 'FORTE'
00016                                   ALORS I(+0)   '---->'
00017                                   SINON I(+0)  '...>'
00018                                         M Y4 = 0
00019                                FIN
00020                            M Y9 = 6 * Y8 SI Y5 > Y9 ALORS
00021                                  ECRIRE M BUFFER = U
00022                                                M Y8 = Y8 + 1 FIN
00023                               M Y5 = Y5 + 1
00024                               REFAIRE
00025                     FIN
00026             FIN
00027       I(+0)  NCEV DE R-EV1 DE X1 ECRIRE
00028 M BUFFER = U
00029 SI Y4 = 1 ALORS I 'CIRCUIT INCONDITIONNEL'
00030             SINON I 'CIRCUIT CONDITIONNEL'
00031 FIN
00032 FIN
```


PROGRAMME CONTROL4 (fin)

```
*F/RECHCIRCUIT/
00001*1,$ L
00001 D X3 = UN GH-EV
00002 D YZ0 = DEBUT
00003       ESPACE 80
00004        TABLEAU 50 Y-P DECIMAL 2
00005        TABLEAU 50 Y-E DECIMAL 1
00006       FIN
00007 M Y4 = Y-P (Y2)
00008 M X2 = UN GH-EV Y4
00009 M Y8 = 0 EXEC RECHSUCCESSEUR
00010 SI Y8 = 1
00011          ALORS M Y2 = Y2 + 1
00012                M Y9 = NUMDE X3
00013                M Y-P (Y2) = Y9
00014                M Y-E (Y9) = 1
00015                M Y3 = Y1
00016                 EXEC TESTCIRCUIT
00017           SINON
00018                M Y3 = NUMDE X2
00019                M Y-E ( Y3) = 0
00020                M Y2 = Y2 - 1
00021  FIN
FIN DE FICHIER
```

EXEMPLE DE PROGRAMME DE MISE A JOUR DU SCHEMA CONCEPTUEL

MDYPROP : Modification de la dynamique d'une propriété

```
*f/mdyprop/
00001*1,$ 1
00001  I´      MODIFICATION DE LA DYNAMIQUE    ´
00002  I´      D UNE PROPRIETE                 ´
00003  D YZO = DEBUT
00004        ESPACE 80
00005        TABLEAU 2 Z-OBS MOT 8
00006        TABLEAU 2 Z-OBN MOT 8
00007       FIN
00008  M Y5 = 1
00009  EXEC INITIAL
00010  I´DONNER LE NOM DU C-OBJET OU SE TROUVE´
00011  I´    LA PROPRIETE A MODIFIER´
00012  M Z1=U  M Z1=EXT
00013  M X1=UN OBJET AYANT NCOB=Z1
00014  SI X1=U ALORS I´OBJET INEXISTANT CONTROLER SVP´
00015    SINON
00016    M X2=R-CL DE X1
00017    I´VOICI LES C-OBJETS DE LA CLASSE´
00018     ESCHD DE X2 F
00019  I´DONNER LE NOM DU CONSTITUANT´
00020  I´REPRESENTANT LA PROPRIETE A MODIFIER´
00021  M Z1=U  M Z1=EXT
00022    POUR TOUT CH-COMP X9 DE X1
00023    SI NCONSTOB DE R-CONSTOB DE X9=Z1
00024    ALORS M Z9=U  M Z9=Z1 FIN FIN
00025    M BUFFER=U ECRIRE  M BUFFER=U
00026    I(2)´LE CONSTITUANT  ´ I(+2)Z9
00027    I(+2)´  DOIT IL APPARTENIR A L UN D ENTRES EUX O/N?´
00028    ECRIRE  M BUFFER=U
00029    M Z7=U  M Z7=EXT
00030    SI Z7=´N´ ALORS
00031    M DSOB DE X1=Y25   M X10=X1
00032     EXEC CREATION-OBJET
00033     EXEC CREATION-CONST
00034     G UN COMPOB X8
00035     M R-OB DE X8=UN OBJET AYANT NCOB=Z8
00036     M R-CONSTOB DE X8= UN CONSTOB AYANT NCONSTOB=Z9
00037     M X1=X10
00038     I(2)´LE C-OBJET  ´ I(+2) NCOB DE X1
00039     I(+2)´  EST SUPPRIME´ ECRIRE  M BUFFER=U
00040     I(2) ´IL DOIT ETRE REMPLACE PAR UN AUTRE C-OBJET´ ECRIRE
00041     M BUFFER=U  EXEC COPY-CONST
00042     I´VOICI LA NOUVELLE VERSION DE LA CLASSE´
00043     M X2=U M X3=U M X4=U M X5=U
00044     M X6=U M X7=U M X8=U M X9=U M X10=U
```


MDYPROP (fin)

(les sous-programmes figurent dans la notice d'utilisation (FOU 82)).

```
00045     M BUFFER=U  ECRIRE  M BUFFER=U
00046     I(2)´VOUS AUREZ A ENTRER LA DYNAMIQUE ´
00047     I(+2)´DU NOUVEAU C-OBJET CREE´   ECRIRE
00048      SINON
00049      M DSOB DE X1=Y25
00050      M Z-OBS (1) = NCOB DE X1
00051      M BUFFER=U
00052      I(2)´LE C-OBJET  ´ I(+2) NCOB DE X1
00053      I(+2)´  EST SUPPRIME ´ ECRIRE  M BUFFER=U
00054      I(2)´IL DOIT ETRE REMPLACE PAR UN AUTRE´ ECRIRE
00055      EXEC COPY-CONST  M Z-OBN (2) = Z6 M BUFFER=U
00056       I(2)´DONNER LE NOM DU C-OBJET DANS´
00057       I(+2)´ LEQUEL IL FAUT INSERER LE CONSTITUANT  ´
00058       I(+2) Z9 ECRIRE  M BUFFER=U
00059       M Z1=U  M Z1=EXT
00060       M X1=U  M X1=UN OBJET AYANT NCOB=Z1
00061       SI X1=U ALORS I´OBJET INEXISTANT CONTROLER SVP´
00062        SINON
00063        M DSOB DE X1=Y25
00064        M Z-OBS (2) = NCOB DE X1
00065        I(2)´LE C-OBJET  ´ I(+2) NCOB DE X1
00066        I(+2)´ EST SUPPRIME´ ECRIRE
00067        EXEC COPY-CONST
00068        M Z-OBN (2) = Z6
00069        G UN COMPOB X8
00070        M R-OB DE X8=UN OBJET AYANT NCOB=Z1
00071        M R-CONSTOB DE X8= UN CONSTOB AYANT NCONSTOB=Z9
00072        I´VOICI LA NOUVELLE VERSION DE LA CLASSE´
00073        ESCHD DE X2 F
00074         M X1=U M X2=U M X3=U M X4=U M X5=U
00075        M X6=U M X7=U M X8=U M X9=U M X10=U
00076        FAIRE
00077            SI Y5 > 2 ALORS SORTIE SINON
00078              M X1=UN OBJET AYANT NCOB=Z-OBS (Y5)
00079              EXEC REPERCUT
00080              M Y5 = Y5 + 1
00081              REFAIRE
00082            FIN
00083        FIN
00084        M BUFFER=U  ECRIRE  M BUFFER=U
00085        I(2)´VOUS AUREZ A ENTRER LA DYNAMIQUE ´
00086        I(+2)´DES NOUVEAUX C-OBJETS CREES´  ECRIRE  M BUFFER=U
00087        I(2)´ EN PARTICULER:     ´
00088        I(+2) Z-OBN (1) I(+2)´  ET  ´  I(+2) Z-OBN (2)  ECRIRE
00089      FIN
00090     FIN
00091  FIN
```

BIBLIOGRAPHIE

ABR 74 ABRIAL J.R. - "Data semantics" Proc IFIP TC2 WC Cargèse 1974.

ABR 78 ABRIAL J.R. - "Z : A specification language" Proc of IFIP Tokyo 1978.

ANS 75 ANSI/X3/SPARC - "Interim report from the study group on data base management systems" Bulletin of ACM Vol 7 n° 2 1975.

ARM 74 ARMSTRONG W.W. - "Dependency structures of data base relation-ships" Proc IFIP 74 North Holland publ.

ASH 61 ASHBY W.R. - "An introduction to Cybernetics" Chapman and Hall - Londres 1961.

AST 76 ASTRAHAN M.M. et Al - "System R : a relational approach to data base management" ACM TODS Vol 1 n° 2 1976.

BAC 77 BACHMAN C.W., DAYA M. - "The role concept in data models" Proc 3ème conf on VLDB 1977.

BEE 78 BEERI C., BERNSTEIN P.A., GOODMAN N. - "A sophisticat's introduction to data base normalization theory" Proc of 4ème Conf VLDB Berlin 1978.

BEN 76 BENCI G., BODART F., BOGAERT H., CABANES A. - "Concepts for the design of a conceptual schema" Proc of IFIP TC2 WC Freundenstadt 1976.

BEN 79 BENCI G., ROLLAND C. - "Bases de données conception canonique pour une réalisation extensible" SCM Paris 1979.

BER 76 BERNSTEIN P.A. - "Synthesizing third normal form relations from functional dependencies" ACM TODS Vol 1 n° 4 1976.

BOD 79 BODART F., PIGNEUR Y. - "A model and a language for functional specifications and evaluation of information system dynamics" IFIPTC8WG on "formal models and pratical tools for information system design" Oxford 1979.

BOD 81 BODARD F. - "Elements de conception et d'analyse des systèmes d'information des organisations" Notes de cours - Ecole d'été d'informatique de l'AFCET 1981.

BOY 74 BOYCE R.F., CHAMBERLIN W.F., HAMMER M.M. - "Specifying queries as relational expressions : SQUARE" Proc IFIP TC2 WC North Holland 1974.

BRA 76 BRACCHI G. et AL - "Binary logical associations in data modelling" Proc IFIP TC2 WC Freundenstadt 1976.

BRA 79 BRACCHI G., FURTADO A., PELAGATTI G. - "Constraint specification in evolutionary data base design" IFIP TC8 WC on "formal models and pratical tools for information system design" Oxford 1979.

BRE 79 BREUTMAN B., FALKENBERG E., MAUER R. - "CSL : a language for defining conceptual schemas" IFIPTC2 WC on data base architecture Venise 1979.

BRO 81 BRODIE N.L. - "On Modelling behavioural semantics of data bases" Proc of the seventh international conference on VLDB Cannes 1981.

BUB 77 BUBENKO J.A. - "The temporal dimension in information modelling" Proc of IFIP TC2 WC Nice 1977.

CAV 79 CAVARERO J.L. - "LAPAGE, un modèle et un outil d'aide à la conception de systèmes d'information" Thèse d'état Nice 1979.

CHA 80 CHABRIER J.J., HENRY P. - "TYP un système de programmation pour la construction et la manipulation de système de bases de données" AFCET bulletin GROPLAN n° 11 1980.

CHA 74 CHAMBERLIN D.D., BOYCE R.F. - "SEQUEL : a structured english query language" Proc of ACM SIGMOD Worhshop on data description, access and control 1974.

CHE 76 CHEN P. - "The entity-relationship model toward a unifed view of data" ACM Transactions on data base systems Vol 1 mars 1976, pp. 9-36.

COD_A 71 CODASYL DBTG - "Report on the codasyl data base task group" ACM New York 1971.

COD 70 CODD E.F. - "A relational model of data for large shared data banks" CACM vol 13 n° 6 1970.

COD 71 CODD E.F. - "A data base sublanguage formed on the relational calculus" Proc ACMSIGDIDET Workshop on data description, access and control 1971.

COD 74 CODD E.F. - "Recent investigations into relational data base systems" Proc IFIP congress Stockholm 1974.

CRE 75 CREHANGE M. - "Description formelle, représentation, interrogation des informations complexes - Système PIVOINES" Thèse d'état Nancy 1975.

CRO 75 CROCUS - "Systèmes d'exploitation des ordinateurs" Dunod 1975.

CXP 74 CXP - "Etude comparative des méthodes d'analyse" Centre d'expérimentation des packages étude n° 8 1974.

DAH 70 DAHL O.J., MYHRHAUG B., NYGAARD K. - "SIMULA 67 common base language" pub S 22 Norwegian computer center 1970.

DATE 77 DATE C.J. - "An introduction to data base systems" Second edition Addison Wesley 1977.

DAT 80 DATE C.J. - "An introduction to the unified data base language UDL" Proc 6th international conference on VLDB 1980.

DEA 81 DE-ANTONELLIS V., ZONTA B. - "Modelling events in data base applications design" Proc of the seventh international conference on VLDB Cannes 1981.

DEL 73 DELOBEL C. - "Contributions théoriques à la conception et à l'évaluation d'un système d'informations appliqué à la gestion" Thèse d'état Université de Grenoble 1973.

DEL 76 DELOBEL C., ADIBA M., LEONARD M. - "A unified approach for modelling data in logical data base design" Proc IFIP TC2 WC Freundenstadt 1976.

DEL 80 DELOBEL C. - "An overview of the relational data theory" Prof of the IFIP conference Tokyo 1980.

DEM 75 DEMOLOMBE R., LEMAITRE A., NICOLAS J.M. - "SYNTEX 2" Rapport DRME - n° 74222 CERT Toulouse 1975.

DER 79 DERNIAME J.C., FINANCE J.P. - "Types abstraits de données spécification, utilisation, réalisation" Ecole d'été de l'AFCET Monastier 1979.

DER 81 DERNIAME J.C., CHABRIER J.J., GODARD C., TOUNSI N., AIT-HADDOU A. - "Un système de gestion multibase" Rapport CRIN 81-R-53.

DUF 80 DUFOURD J.F. - "Maquettes pour évaluer les systèmes d'information des organisations" Thèse d'état Nancy 1980.

EHR 78 EHRIG H., WEBER - "Algebraic specification - schemas for data base systems" Proc of the Fourth international conference on VLDB Berlin 1978.

ESW 75 ESWARAN K.P., CHAMBERLIN D.D. - "Functional specifications of a subsystem for data base integrity" Proc on first conf VLDB 1975.

FAG 77 FAGIN R. - "Multivalued dependencies and a new normal forme for relational data bases" ACM TODS vol 2 nb 3 1977.

FAL 75 FALKENBERG E. - "Design and application of a natural Language oriented data base Language" Adv course on data base Languages and natural Language processing, Freundenstadt, Germany, Aug 1975.

FIN 79 FINANCE J.P. - "Etude de la construction des programmes : méthodes et langages de spécification et de résolution de problèmes" - Thèse d'état Nancy 1979.

FLO 77 FLORY A. - "Un modèle et une méthode pour la conception logique d'une base de données" Thèse d'état Lyon 1977.

FLO 79 FLORY A., KOULOUDMIDJAN J. - "La représentation des règles d'évolution d'un système d'information" Colloque les bases de données modèles, mise en oeuvre, évaluation Paris 1979.

FOR 62 FORRESTER J.W. - "Industrial dynamics" Wiley and Sons 1962.

FOR 67 FORTET R., LE BOULANGER H. - "Eléments pour une synthèse sur les systèmes à auto-organisation" Revue METRA n° 12 1967.

GAL 81 GALLAIRE - "Impacts of logic on data base" Proc of the seventh international conference on VLDB Cannes 1981.

GUT 75 GUTTAG J.V. - "The specification and application to programming of abstract data types" PhD. Thesis, CSRG TR 59, University of Toronto 1975.

HAM 76 HAMMER M.M., MCLEOD D.J. - "A frame work semantic integrity" Proc II Int Conf on software Eng, Oct 1976.

HAN 79 HANSON S., JACKSON P., JULLIG R., LEVY P. - "Summary of the characteristics of several modern programming languages" SIGPLAN Notices 1979.

HEC 80 HECKENROTH H., TARDIEU H., ESPINASSE B. - "Niveaux de modèles formalismes et outils pour la dynamique dans les systèmes d'informations" Actes du colloque INFORSID Aix en Provence 1980.

HOA 72 HOARE C.A.R., WIRTH N., - "An axiomatic definition of the programming language PASCAL" Acta informatica vol 2, ed Springer Verlag 1972.

ICH 79 ICHBIAH J.D. et al - "Preliminary ADA reference manual" Siplan notices vol 14 n° 6A juin 1979.

IFI 76 IFIP TC2 Working conférence "Modelling in data base management systems" Freundenstadt 1976.

IFI 75 IFIP TC2 Special Working conférence - "A technical in-depth evaluation of the DDL" Namur 1975.

$INF_1$75 INFORSID Groupe II - "Séminaire sur le langage d'analyse et de conception des systèmes d'information automatisés" Publ IRIA Pont-à-Mousson 1975.

$INF_2$75 INFORSID Groupe II - "Représentation des systèmes d'information : maquette, modèle et prototype - Mesures et simulation. Publ. IRIA Saint-Pierre de Chartreuse 1975.

INF 76 INFORSID - "Colloque sur la recherche en informatique des organisations" Publ IRIA Caen 1976.

INF_1 78 INFORSID - "Colloque bilan des ATP CNRS IRIA en informatique d'organisation" Publ. IRIA Rennes 1978.

INF_2 78 INFORSID Groupe II - "La dynamique dans les systèmes d'information" Publ IRIA CERGY Pontoise 1978.

INF 80 INFORSID Groupe II - "Implémentation de la dynamique dans les systèmes d'information" Publ Cete Aix en Provence 1980.

ISO 81 ISO TC 97/SC5/WG3 - "Preliminary report concepts and terminology for the conceptual schema" Ed JJ Van Griethuysen Fevrier 1981.

KEN 76 KENT W. - "Describing information (Not data reality)" Technical Report TR03012 1976.

KEN 77 KENT W. - "Entities and relationships in information" Proc IFIP TC2 WC Nice 1977.

KLE 68 KLEENE S.C. - "Mathematical Logic" John Wiley and Sons Inc 1968.

LAM 77 LAMPSON B.W. et AL - "Report on the programming language EUCLID" SIGPLAN notices Vol 12, n° 2 pp. 2-77.

LAN 66 LANGEFORS B. - "Theoretical analysis of information systems" Lund 1966, Fourth edition 1973.

LAN 74 LANGEFORS B. - "Theoretical aspects of information systems for management" IFIP Congress Stockholm 1974.

LEO 81 LEONARD M., LUONG B.T. - "Information systems design integrating data and transactions" Proc of the seventh international conference on VLDB Cannes 1981.

LES 79 LESCANNE P. - "Etude algèbrique et relationnelle des types abstraits et de leur représentation" Thèse d'état, Université Nancy I 1979.

LEW 77 LEWIS E.A. et Al - "A canonical representation for the relational schema and logical data independence" Proc COMPSAC'77 1977.

LIN 74 LINDGREN P. - "Basic operations on informations as a basis for data base design" IFIP Congress Stockholm 1974.

LIS 75 LISKOV B. - "An introduction to CLU in new directions in algorithmic languages" Schuman ed. 1975.

MER 79 MERISE - "Méthode de définition d'un SI" - CII Ministère de l'Industrie Juin 1979.

MIN 79 MINOT R. "ATM : un système de fabrication de programmes basé sur les consepts de modularité et de type abstrait" Thèse 3ème cycle Nancy 1979.

MOH 78 MOHAN C. - "An overview of recent data base research" DATA BASE vol 10 n° 2 Fall 1978.

MIJ 76 MIJARES I., PEEBLES R., - "Structural semantics of data bases" Modelling in data base management systems. Proceedings of the IFIP TC2 WC Freundenstadt 1976.

MYL 75 MYLOPOULOS J. et AL - "A multilevel relational System" Proc NCC 1975.

NIC 79 NICOLAS J.M. - "Contributions à l'étude théorique des bases de données - Apports de la logique mathématique" Thèse d'état Université Paul Sabatier. Toulouse 1979.

NIJ 77 NIJSSEN G.M. - "Current issues in conceptual schema concepts" Proc of IFIP TC2 WC Nice 1977.

PAI 74 PAIR C. - "Some theorical aspects of the notions of data, information and information structure" Data base management systems. Klimbie-Koffman North Holland 1974.

PAI 79 PAIR C. - "La construction des programmes" RAIRO informatique Vol 13, 2 1979, p. 113-118.

PEA 77 PEAUCELLE J.L. - "Les besoins en information dans les systèmes informatiques de gestion" Thèse d'état Paris 1977.

PEC 75 PECCOUD "MACSI" - Thèse d'état, Grenoble 1975.

PET 66 PETRI C.A. - "Communication with automata" Supplement to technical report RADC-TR-65-377, Vol 1, New-York, 1966.

PIR 76 PIROTTE A. - "Explicit description of entities and their manipulation in languages for the relational data base model" Thesis, Université libre de Bruxelles Belgique 1976.

REM 74 REMY J.L. - "Structures d'information - Formalisation des notions d'accès et de modification d'une donnée" Thèse de spécialité Nancy 1974.

RIS 77 RISSANNEN J. - "Independent components of relations" ACM TODS Vol 2, n° 4 1977.

SAL 80 SALES A. - "Utilisation de la notion de type abstrait générique en bases de données relationnelles" Actes du congrès AFCET Nancy 1980.

SCH 77 SCHMID H.A. - "An analysis of some constructs for conceptual model" Proc of IFIP TC2 WC Nice 1977.

SEN 75 SENKO M.E. - "Spécification of stored data structures and desired output results in DIAM II With FORAL" Proc First conf VLDB 1975.

SHO 67 SHONFIELD J.R. - "Mathematical Logic" Addison-Wesley 1967.

SOC 77 SOCRATE II - "Manuel d'utilisation" Ref F 24742 CII Honeywellbull 1977.

STO 76 STONEBRAKER M.R. et AL - "The design and implementation of INGRES" ACM TODS Vol 1 n° 3, September 1976.

SUN 73 SUNDGREN B. - "An infological approach to data bases" PHD Thesis Stockholm 1973.

SUN 75 SUNDGREN B. - "Theory of data bases" Mason/Charter New-York 1975.

TAR 74 TARDIEU H. - "The individual Model" IFIP TC2 WC Namur 1974.

TEI 73 TEICHROEW D., RATAJ W., HERSHEY E. - "An introduction to computer aided documentation of user requirments for computer bases information processing systems" ISDOS, Working paper, n° 72, 1973.

WAR 81 WARREN D.H.D., "Efficient processing of interactive relational data base queries expressed in logic", Proc of the seventh international conference on VLDB Cannes 1981.

WUL 75 WULF W.A., LONDON R.L., SHAW W. - "Abstraction and verification in ALPHARD in new directions in algorithmic languages" SASCHUMAN ed IRIA 1975.

ZIL 75 ZILLES S.N. - "Abstract specifications for data types" IBM research Laboratory, San José, California 1975.

ZLO 75 ZLOOF M.M. - "Query by example : the invocation and definition of tables and forms" Proc VLDB 1975.

REFERENCES BIBLIOGRAPHIQUES RELATIVES AU PROJET REMORA

	N° CRIN	
PER 76	76-R-009	PEREA VILLACORTA W. - "Méthode d'analyse dans le projet REMORA" Thèse de 3ème cycle, Nancy, 1976.
THI 76	76-R-008	THIERY O. - "L'aide à la conception dans le projet REMORA" Thèse de 3ème cycle, Nancy, 1976.
FOU 76	76-R-038	FOUCAUT O., ROLLAND C. - "Une méthode et un système de conception des systèmes d'informations dans les organisations" Congrès AFCET TTI Paris, 1976.
ROL 76	76-R-059	ROLLAND C. - "Bilan de deux années de recherche sur le projet REMORA" Actes du Congrès INFORSID, publication IRIA, Caen 1976.
ROL 77	77-R-022	ROLLAND C. - "Un système de pilotage de la conception des systèmes d'information" Informatique et gestion n° 85, mars 1977.
FOU 76	77-R-023	FOUCAUT O., LAMY D., ROLLAND C., THIERY O. "Rapport final des travaux de recherche menés sur le projet REMORA dans le cadre du contrat ATP-CNRS-IRIA n° 76-124, octobre 1974 à décembre 1976.
LAM 77	77-R-024	LAMY D. - "Les traitements : leurs structures et leur génération dans le projet REMORA" Thèse de 3ème cycle Nancy, 1977.
FOU 77	77-R-038	FOUCAUT O. - "Modèle pour la représentation des systèmes d'information" Deuxième rapport intermédiaire du contrat Sesori 76-168, 1977.
CHE 78	78-T-047	CHESSERON M. - "Modèle de représentation et méthode de construction des composantes dynamiques d'un système d'information dans le projet REMORA" Thèse de 3ème cycle, Nancy, 1978.

KRI 78 78-T-048 KRIEGUER M. - "Modèle de représentation et méthode de construction des composantes d'un SI : Projet REMORA" Thèse de 3ème cycle, Nancy, 1978.

FOU 78 78-P-014 FOUCAUT O., ROLLAND C. - "Concepts for design of an information system conceptual schema and its utilization in the Remora project" 4ème conférence internationale Very large data base, Berlin, 1978.

$ROL_1$78 78-P-062 ROLLAND C., THIERY O. - "Les fonctions d'un système d'aide à la conception des SI". Convention informatique, Paris, 1978.

$ROL_2$78 78-P-071 ROLLAND C. - "Le projet REMORA" Actes du colloque AIP CNRS IRIA, Rennes, 1978.

FOU 78 78-P-055 FOUCAUT O. - "Modèle conceptuel dynamique de système d'information" Actes du colloque INFORSID, Publ. IRIA, Cergy Pontoise, 1978.

FOU 79 79-P-039 FOUCAUT O., LANY D., ROLLAND C. - "Propositions pour la conception et la réalisation structurée des traitements d'un système d'information" Informatique et Gestion, Mars 1979.

$ROL_1$79 79-R-026 ROLLAND C., RICHARD C., "Concepts for a distributed computer system design" Seminar distributed data sharing systems Working groupe C, Aix en Provence, mai 1979.

$ROL_2$79 79-P-027 ROLLAND C. - "Un modèle pour la conception des systèmes de bureau" Actes du Congrès bureautique, Grenoble Mai 1979.

FOU 79 79-P-032 FOUCAUT O., ROLLAND C. - "Dimension dynamique dans un modèle de données" Actes du Colloque "les bases de données, modèles, évaluation" Paris, Juin 1979.

ROL 79 79-P-041 ROLLAND C., BENCI G. - "Management of information system projects Actes du Congrès Internet, Garnich-Partenkischen Septembre 1979.

$ROL_1$79 79-P-016 ROLLAND C., FOUCAUT O., RICHARD C., THIERY O. "Information system design and information system computer aided design" Proc. EUROIFIP 79, London, Septembre 1979.

$ROL_2$79 79-P-038 ROLLAND C., LEIFERT S., RICHARD C. - "Tools for information system dynamics management" Proc. 5ème Conf. VLDB Rio-Janeiro, Octobre 1979.

BAN 79 79-T-046 BANON D. - "Projet Remora : un outil pour la gestion des systèmes d'information" Thèse de 3ème cycle, Nancy Mai 1979.

RIC 79 79-T-075 RICHARD C. - "Une approche conceptuelle des problèmes de synchronisation" Thèse de 3ème cycle Nancy 1979.

$ROL_1$80 80-P-017 ROLLAND C., LEIFERT S., RICHARD C. - "Le pilotage de l'évolution des SI" Actes du colloque INFORSID Aix en Provence Janvier 1980.

$ROL_2$80 80-P-028 ROLLAND C., THIERY O. - "Proposition for an Information system computer aided design software" Proc of ADM congress SORENTO Avril 1980.

LEI 80 80-T-114 LEIFERT S. - "Langage et outil pour la définition et la gestion de systèmes d'informations" Thèse docteur ingénieur Nancy, 1980.

$ROL_1$81 81-P-032 ROLLAND C., THIERY O. "A graphic tool for information system design". Conférence CIPS, Waterloo, Canada juin 1981.

$ROL_2$81 81-P-033 ROLLAND C., THIERRY O. - Une machine-système d'information - Convention informatique latine Barcelone 1981.

$ROL_3$81 81-P-034 ROLLAND C., THIERY O. - Constructions de base d'un langage de spécification de système d'information. Congrès base de données Tunis août 1981.

ROL$_1$82 ROLLAND C., RICHARD C. -"The REMORA Methodology for information systems design and Management" IFIP TC8 Conference on Comparative Review of information systems design and methodologies. Amsterdam 1982.

ROL$_2$82 ROLLAND C., THIERY O. -"Basis constructions of a specification language for information system" 1982 - Information systems à paraître.

ROL$_3$82 ROLLAND C., RICHARD C. - "Transactions Modelling" ACM SIGMOD Conference 1982, Orlando.

FOU 82 FOUCAUT O., KEBAILI A., - Projet REMORA - "Notice d'utilisation du système d'aide à la conception d'un SI" Rapport interne CRIN (à paraître).

UNIVERSITE * BIBLIOTHEQUE-SCIENCES * NANCY

NOM DE L'ETUDIANT : *Madame FOUCAUT Odile née GUILLOT*

NATURE DE LA THESE : *Doctorat ès sciences mathématiques*

VU, APPROUVE

ET PERMIS D'IMPRIMER

NANCY, le 9 JUIN 1982 1984

LE PRESIDENT DE L'UNIVERSITE DE NANCY I,

R. MAINARD

R E S U M E

Cette thèse présente un modèle conceptuel permettant une expression abstraite de l'ensemble des phénomènes réels (aspects statiques et aspects dynamiques) à représenter dans le système d'information (SI) d'une organisation. Ce modèle est une extension du modèle relationnel de CODD dans lequel sont introduits le temps et la notion de type de relation.

Il conduit à la construction de SI historiques permettant de retracer l'évolution complète de l'organisation au cours du temps (évolution de la structure, évolution des constituants, traitements effectués, évènements survenus).

Un logiciel associé au modèle apporte une assistance complète au concepteur (modularité, contrôle, documentation, generation partielle de la définition) aussi bien lors de la conception initiale du SI que lors de sa maintenance.

Mots clés :
Informatique - Informatique d'organisation - Systèmes d'information - Modèle conceptuel - Modèle relationnel - Bases de données - Temps - Dynamique - Logiciel d'aide à la conception.