



**HAL**  
open science

## Modèle d'évaluation de systèmes d'organisation

Jean-Louis Cavarero

► **To cite this version:**

Jean-Louis Cavarero. Modèle d'évaluation de systèmes d'organisation. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 1972. Français. NNT: . tel-00284245

**HAL Id: tel-00284245**

**<https://theses.hal.science/tel-00284245>**

Submitted on 2 Jun 2008

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

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

N° d'ordre

# THESE

présentée à

L'UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE

pour obtenir

LE GRADE DE DOCTEUR-INGENIEUR

par

Jean-Louis Cavarero



## MODELE D' EVALUATION DE SYSTEMES D' ORGANISATION

Thèse soutenue le 25 Avril 1972, devant la Commission d'Examen :

Monsieur L. BOLLIET Président

Messieurs M. GRIFFITHS

B. LUSSATO Examineurs

F. PECCOUD



Président : Monsieur Michel SOUTIF  
Vice-Président : Monsieur Gabriel CAU

PROFESSEURS TITULAIRES

MM. ANGLES D'AURIAC Paul	Mécanique des fluides
ARNAUD Georges	Clinique des maladies infectieuses
ARNAUD Paul	Chimie
AYANT Yves	Physique approfondie
Mme BARBIER Marie-Jeanne	Electrochimie
MM. BARBIER Jean-Claude	Physique expérimentale
BARBIER Reynold	Géologie appliquée
BARJON Robert	Physique nucléaire
BARNOUD Fernand	Biosynthèse de la cellulose
BARRA Jean-René	Statistiques
BARRIE Joseph	Clinique chirurgicale
BENOIT Jean	Radioélectricité
BESSON Jean	Electrochimie
BEZES Henri	Chirurgie générale
BLAMBERT Maurice	Mathématiques Pures
BOLLIET Louis	Informatique (IUT B)
BONNET Georges	Electrotechnique
BONNET Jean-Louis	Clinique ophtalmologique
BONNET-EYMARD Joseph	Pathologie médicale
BONNIER Etienne	Electrochimie Electrometallurgie
BOUCHERLE André	Chimie et Toxicologie
BOUCHEZ Robert	Physique nucléaire
BRAVARD Yves	Géographie
BRISSENEAU Pierre	Physique du Solide
BUYLE-BODIN Maurice	Electronique
CABANAC Jean	Pathologie chirurgicale
CABANEL Guy	Clinique rhumatologique et hydrologie
CALAS François	Anatomie
CARRAZ Gilbert	Biologie animale et pharmacodynamie
CAU Gabriel	Médecine légale et Toxicologie
CAUQUIS Georges	Chimie organique
CHABAUTY Claude	Mathématiques Pures
CHARACHON Robert	Oto-Rhino-Laryngologie
CHATEAU Robert	Thérapeutique
CHENE Marcel	Chimie papetière
COEUR André	Pharmacie chimique
CONTAMIN Robert	Clinique gynécologique
COUDERC Pierre	Anatomie Pathologique
CRAYA Antoine	Mécanique
Mme DEBELMAS Anne-Marie	Matière médicale
MM. DEBELMAS Jacques	Géologie générale
DEGRANGE Charles	Zoologie
DESSAUX Georges	Physiologie animale
DODU Jacques	Mécanique appliquée
DREYFUS Bernard	Thermodynamique
DUCROS Pierre	Cristallographie
DUGOIS Pierre	Clinique de Dermatologie et Syphiligraphie
FAU René	Clinique neuro-psychiatrique
FELICI Noël	Electrostatique
GAGNAIRE Didier	Chimie physique
GALLISSOT François	Mathématiques Pures
GALVANI Octave	Mathématiques Pures

PROFESSEURS SANS CHAIRE

MM. AUBERT Guy	Physique
BEAUDOING André	Pédiatrie
BERTRANDIAS Jean-Paul	Mathématiques Appliquées
BIARES Jean-Pierre	Mécanique
BONNETAIN Lucien	Chimie minérale
Mme BONNIER Jane	Chimie générale
MM. CARLIER Georges	Biologie végétale
COHEN Joseph	Electrotechnique
COUMES André	Radioélectricité
DEPASSEL Roger	Mécanique des Fluides
DEPORTES Charles	Chimie minérale
DESRE Pierre	Métallurgie
DOLIQUE Jean-Michel	Physique des Plasmas
GAUTHIER Yves	Sciences biologiques
GEINDRE Michel	Electroradiologie
GIDON Paul	Géologie et Minéralogie
GLENAT René	Chimie organique
HACQUES Gérard	Calcul numérique
JANIN Bernard	Géographie
Mme KAHANE Josette	Physique
MM. MULLER Jean-Michel	Thérapeutique
PERRIAUX Jean-Jacques	Géologie et minéralogie
POULOUJADOFF Michel	Electrotechnique
REBECQ Jacques	Biologie (CUS)
REVOL Michel	Urologie
REYMOND Jean-Charles	Chirurgie générale
ROBERT André	Chimie papetière
SARRAZIN Roger	Anatomie et chirurgie
SARROT-REYNAULD Jean	Géologie
SIBILLE Robert	Construction Mécanique
SIROT Louis	Chirurgie générale
Mme SOUTIF Jeanne	Physique générale
M. VALENTIN Jacques	Physique nucléaire

MAITRES DE CONFERENCES ET MAITRES DE CONFERENCES AGREGES

Mlle AGNIUS-DELORE Claudine	Physique pharmaceutique
ALARY Josette	Chimie analytique
MM. AMBLARD Pierre	Dermatologie
AMBROISE-THOMAS Pierre	Parasitologie
ARMAND Yves	Chimie
BEGUIN Claude	Chimie organique
BELORIZKY Elie	Physique
BENZAKEN Claude	Mathématiques Appliquées
Mme BERTRANDIAS Françoise	Mathématiques Pures
MM. BLIMAN Samuel	Electronique (EIE)
BLOCH Daniel	Electrotechnique
Mme BOUCHE Liane	Mathématiques (CUS)
MM. BOUCHET Yves	Anatomie
BOUSSARD Jean-Claude	Mathématiques Appliquées
BOUVARD Maurice	Mécanique des Fluides
BRIERE Georges	Physique expérimentale
BRODEAU François	Mathématiques (IUT B)
BRUGEL Lucien	Energétique
BUISSON Roger	Physique
BUTEL Jean	Orthopédie
CHAMBAZ Edmond	Biochimie médicale
CHAMPETIER Jean	Anatomie et organogénèse

MM. GASTINEL Noël	Analyse numérique
GERBER Robert	Mathématiques Pures
GIRAUD Pierre	Géologie
KLEIN Joseph	Mathématiques Pures
Mme KOFLER Lucie	Botanique et Physiologie végétale
MM. KOSZUL Jean-Louis	Mathématiques Pures
KRAVTCHENKO Julien	Mécanique
KUNTZMANN Jean	Mathématiques Appliquées
LACAZE Albert	Thermodynamique
LACHARME Jean	Biologie végétale
LATREILLE René	Chirurgie générale
LATURAZE Jean	Biochimie pharmaceutique
LAURENT Pierre	Mathématiques Appliquées
LEDRU Jean	Clinique médicale B
LLIBOUTRY Louis	Géophysique
LOUP Jean	Géographie
Mlle LUTZ Elisabeth	Mathématiques Pures
MALGRANGE Bernard	Mathématiques Pures
MALINAS Yves	Clinique obstétricale
MARTIN-NOEL Pierre	Seméiologie médicale
MASSEPORT Jean	Géographie
MAZARE Yves	Clinique médicale A
MICHEL Robert	Minéralogie et Pétrographie
MOURIQUAND Claude	Histologie
MOUSSA André	Chimie nucléaire
NEEL Louis	Physique du Solide
OZENDA Paul	Botanique
PAUTHENET René	Electrotechnique
PAYAN Jean-Jacques	Mathématiques Pures
PEBAY-PEYROULA Jean-Claude	Physique
PERRET René	Servomécanismes
PILLET Emile	Physique industrielle
RASSAT André	Chimie systématique
RENARD Michel	Thermodynamique
REULOS René	Physique industrielle
RINALDI Renaud	Physique
ROGET Jean	Clinique de pédiatrie et de puériculture
SANTON Lucien	Mécanique
SEIGNEURIN Raymond	Microbiologie et Hygiène
SENGEL Philippe	Zoologie
SILBERT Robert	Mécanique des fluides
SOUTIF Michel	Physique générale
TANCHE Maurice	Physiologie
TRAYNARD Philippe	Chimie générale
VAILLAND François	Zoologie
VAUQUOIS Bernard	Calcul électronique
Mme VERAÏN Alice	Pharmacie galénique
M. VERAÏN André	Physique
Mme VEYRET Germaine	Géographie
MM. VEYRET Paul	Géographie
VIGNAIS Pierre	Biochimie médicale
YOCCOZ Jean	Physique nucléaire théorique

PROFESSEURS ASSOCIES

MM. BULLEMER Bernhard	Physique
RADHAKRISHNA Pidatala	Thermodynamique

MM. CHIAVERINA Jean	Biologie appliquée (EFP)
CHIBON Pierre	Biologie animale
COHEN-ADDAD Jean-Pierre	Spectrométrie physique
COLOMB Maurice	Spectrométrie médicale
CONTE René	Physique
CROUZET Guy	Radiologie
DURAND Francis	Métallurgie
DUSSAUD René	Mathématiques (CUS)
Mme ETERRADOSSI Jacqueline	Physiologie
MM. FAURE Jacques	Médecine légale
GAVEND Michel	Pharmacologie
GENSAC Pierre	Botanique
GERMAIN Jean-Pierre	Mécanique
GIDON Maurice	Géologie
GRIFFITHS Michaël	Mathématiques Appliquées
GROULADE Joseph	Biochimie médicale
HOLLARD Daniel	Hématologie
HUGONOT Robert	Hygiène et Médecine préventive
IDELMAN Simon	Physiologie animale
IVANES Marcel	Electricité
JALBERT Pierre	Histologie
JOLY Jean-René	Mathématiques Pures
JOUBERT Jean-Claude	Physique du Solide
JULLIEN Pierre	Mathématiques Pures
KAHANE André	Physique générale
KUHN Gérard	Physique
Mme LAJZEROWICZ Jeannine	Physique
MM. LAJZEROWICZ Joseph	Physique
LANCIA Roland	Physique atomique
LE JUNTER Noël	Electronique
LE ROY Philippe	Mathématiques
LOISEAUX Jean-Marie	Physique Nucléaire
LONGUEUE Jean-Pierre	Physique Nucléaire
LUU DUC Cuong	Chimie Organique
MACHE Régis	Physiologie végétale
MAGNIN Robert	Hygiène et Médecine préventive
MARECHAL Jean	Mécanique
MARTIN-BOUYER Michel	Chimie (CUS)
MAYNARD Roger	Physique du Solide
MICOUD Max	Maladies infectieuses
MOREAU René	Hydraulique (INP)
NEGRE Robert	Mécanique
PARAMELLE Bernard	Pneumologie
PECCOUD François	Analyse (IUT B)
PEFFEN René	Métallurgie
PELMONT Jean	Physiologie animale
PERRET Jean	Neurologie
PERRIN Louis	Pathologie expérimentale
PFISTER Jean-Claude	Physique du Solide
PHILIP Xavier	Rhumatologie
Mlle PIERY Yvette	Biologie animale
MM. RACHAIL Michel	Médecine interne
RACINET Claude	Gynécologie et obstétrique
RICHARD Lucien	Botanique
Mme RINAUDO Marguerite	Chimie macromoléculaire
MM. ROMIER Guy	Mathématiques (IUT B)
ROUGEMONT (DE) Jacques	Neuro-Chirurgie
STIEGLITZ Paul	Anesthésiologie

MM. STOEBNER Pierre	Anatomie pathologique
VAN CUTSEM Bernard	Mathématiques Appliquées
VEILLON Gérard	Mathématiques Appliquées (INP)
VIALON Pierre	Géologie
VOOG Robert	Médecine interne
VROUSSOS Constantin	Radiologie
ZADWORNÝ François	Electronique

MAITRES DE CONFERENCES ASSOCIES

MM. BOUDOURIS Georges	Radioélectricité
CHÈKE John	Thermodynamique
GOLDSCHMIDT Hubert	Mathématiques
YACOUD Mahmoud	Médecine légale

CHARGES DE FONCTIONS DE MATIRES DE CONFERENCES

Mme BERIEL Hélène	Physiologie
Mme RENAUDET Jacqueline	Microbiologie

Fait le 8 MARS 1972.



A ma femme  
et à mon grand-père



Je tiens à remercier Monsieur le Professeur LUSSATO du C.N.A.M. qui, non seulement est à l'origine de cette étude, mais qui m'a, en outre, beaucoup aidé et encouragé lors de sa réalisation effective.

Monsieur LUSSATO m'a également fourni toutes les bases nécessaires à la construction du modèle, et je tiens à le remercier plus particulièrement pour l'ensemble des contacts très enrichissants que j'ai eus auprès de lui.

Monsieur F. PECCOUD, de l'Université de Grenoble, qui m'a énormément apporté tant sur le plan humain que matériel au cours de ces trois dernières années. Il est certain que sans lui cette étude n'aurait jamais abouti.

Monsieur le Professeur L. BOLLIET et Monsieur M. GRIFFITHS, président et membre du Jury, qui ont bien voulu accepter cette charge, et l'ont fait avec beaucoup de sympathie.

Monsieur le Professeur J. KUNTZMANN et Monsieur C. BENZAKEN, pour m'avoir guidé efficacement et apporté les compléments mathématiques.

Monsieur M. LARCHER de l'I.U.T. de Grenoble, en stage à l'I.M.A.G., à qui j'avais confié certains travaux de programmation, me permettant à ce jour d'avoir un modèle opérationnel.

Monsieur GROUILLET du B.H.V., qui s'est intéressé à l'étude et m'a toujours reçu très cordialement.

Messieurs DELPUECH et M. MENAHEM, élèves en 3ème année de l'Institut Informatique du C.N.A.M., avec qui j'ai travaillé en partie, et qui m'ont fourni les données nécessaires aux tests du modèle.



## TABLE DES MATIERES

---

CHAPITRE 1. - PRESENTATION DU MODELE -----	1
1.1. Introduction -----	1
1.2. Définitions -----	6
CHAPITRE 2. - OBJECTIFS DU MODELE -----	10
2.1. Mécanisme de l'évaluation-adaptation -----	11
2.2. Résultats à obtenir -----	15
CHAPITRE 3. - LES DONNEES D'ENTREE DU MODELE -----	21
3.1. Définition du SF -----	23
3.2. Définition du SM -----	30
3.3. Définition des paramètres associés aux éléments de solution -----	38
CHAPITRE 4. - PRINCIPES DE FONCTIONNEMENT DU MODELE. DOSSIER TECHNIQUE -	44
4.1. Présentation générale des différentes phases -----	44
4.2. Module de pré-évaluation (PREVAL) -----	46
4.3. Module de notation (NOTGLO) -----	50
4.4. Module de recherche et de notation des solutions partielles (RECHSOL) -----	54
4.5. Module de recherche du système idéal (SYSID) -----	56
4.6. Dossier technique -----	62
CHAPITRE 5. - GENERALISATION. CONCLUSIONS -----	77
5.1. Quelques notions relatives à la théorie de la modularité -----	77
5.2. Présentation de la version 2 -----	79
5.3. Autres applications envisageables -----	81
5.4. Conclusions -----	84
CHAPITRE 6. - ANNEXES -----	85
6.1. Constitution du jeu d'essai -----	86
6.2. Présentation des résultats obtenus -----	87
6.3. Algorithme particulier de recherche des "MIE" et programme PL/1 correspondant -----	88
6.4. Bibliographie -----	91



## CHAPITRE 1. PRESENTATION DU MODELE

### 1.1. Introduction

Les travaux effectués par Monsieur le Professeur LUSSATO, depuis 1968, sur la théorie de la modularité, sont à la base même de cette étude.

Concrètement, mon objectif initial était de construire un modèle (ensemble de programmes) permettant de déterminer quelle était, pour les utilisateurs du Bazar de l'Hôtel de Ville à Paris (B.H.V.), la machine idéale en matière de caisses enregistreuses. Pour ce faire, le modèle prenait en compte, d'une part, les fonctions élémentaires à assurer au niveau de chaque rayon, et d'autre part, les caractéristiques des différentes machines existant sur le marché. L'utilisateur spécifiait ensuite le degré de "satisfaction" de chacune des fonctions par chacune des machines (au moyen d'une note) et grâce à une analyse combinatoire relativement simple, le modèle déterminait la machine qui satisfaisait au mieux l'ensemble des fonctions proposées par l'utilisateur.

L'idée de la conception d'un tel modèle pour résoudre ce genre de problème est tout naturellement basée sur la diversité des solutions envisageables et les délais souvent fort longs pour l'obtention de la solution optimale, lorsque la recherche de cette solution est faite manuellement. Ce problème est bien connu des utilisateurs qui ont eu à faire face au lancement d'un appel d'offre en matière de matériel informatique et au dépouillement des réponses des constructeurs qui en découlent ; le choix de l'ordinateur qui répond le mieux au cahier des charges pose alors de grosses difficultés, difficultés qui sont malheureusement résolues, dans la plupart des cas, par l'introduction de "paramètres nouveaux" n'ayant aucun rapport ni avec le cahier des charges (besoins réels de l'utilisateur), ni avec les caractéristiques techniques des matériels en présence. On en déduit facilement (et on le constate non moins facilement) que la plupart des ordinateurs sont mal adaptés aux besoins de l'entreprise.

En réalité, l'intérêt d'un tel modèle (qui se contente d'automatiser le dépouillement des réponses des constructeurs vis à vis d'un cahier des charges donné) n'est que très limité dans la mesure où, dans la plupart des cas, la solution retenue (optimale) satisfait encore très mal l'utilisateur.

Pour le cas du B.H.V., par exemple, il est tout à fait inconcevable, compte tenu de la diversité des fonctions à assurer au niveau de chaque type de rayon, de proposer une caisse enregistreuse particulière, car cette caisse, même si elle est la "meilleure" parmi celles disponibles sur le marché, est loin de satisfaire complètement l'ensemble des fonctions. (Pour que ce soit le cas, il eut fallu que le constructeur la construise en tenant compte des besoins du B.H.V., et de ces besoins uniquement). C'est donc à la suite de ces quelques remarques que les objectifs du modèle se sont considérablement élargis, et que, sous sa version actuelle, le modèle a pour but principal la détermination de matériel "sur mesure" particulièrement adapté aux besoins de l'utilisateur et aux fins de l'entreprise. L'automatisation de la recherche de matériel sur mesure se justifie bien sûr pour les mêmes raisons que précédemment, avec cependant un avantage supplémentaire, puisque dans le cas d'une recherche manuelle, la combinatoire est non seulement fort longue mais aussi fort complexe, voire dans certains cas inextricable. Un tel objectif est d'ailleurs l'aboutissement logique de travaux antérieurs qui avaient amené, en 1966, le Professeur LUSSATO à "faire construire" pour le B.H.V. un tracteur électronique téléprogrammé appelé "ZEBULON".

Par la suite, la théorie de la modularité nous a permis d'aller plus loin, et plus particulièrement de formaliser les données d'entrée et le mécanisme même du modèle. En ce sens, on peut déjà dire que, compte tenu de certaines restrictions, le modèle permet de rechercher "autre chose" qu'une caisse enregistreuse optimale. La théorie de la modularité, dont le but est de pouvoir **graduier** les investissements en fonction des besoins et de réduire les risques d'obsolescence du matériel, permet en effet de prendre en compte, au niveau du modèle, les éléments les plus intéressants, c'est-à-dire, en fait, de guider l'utilisateur dans la spécification des objectifs qu'il veut réaliser, et surtout des moyens dont il dispose pour satisfaire ces derniers. Ceci dit, la version actuelle du modèle, si elle **s'aide** de la théorie de la modularité en ce qui concerne la définition des données d'entrée, ne permet pas encore de rechercher le matériel sur mesure "le plus modulaire". Une seconde version est prévue à cet effet, version qui se présentera sous forme conversationnelle et dont nous présentons les grandes lignes dans le dernier chapitre.

En ce qui concerne le travail que j'ai effectué personnellement pour mener à bien cette étude, on peut schématiquement le diviser en deux parties :

- une partie théorique, en collaboration avec le Professeur LUSSATO, pour déterminer dans le cas général les "sorties" et les "entrées" du modèle (étude

des besoins de l'utilisateur), et établir une codification des diverses fonctions et moyens envisageables pour des applications de type "grands magasins", codification qui a le mérite d'être suffisamment générale pour pouvoir être utilisée ailleurs que dans les grands magasins (cas de saisie de données quelconques, par exemple). Cette étude théorique a duré un an environ.

- une partie pratique pour la réalisation effective du modèle, réalisation pour laquelle on peut distinguer deux phases distinctes :

- \* une phase d'analyse globale (étude du mécanisme de traitement et conception des documents d'entrée et de sortie du modèle). Environ neuf mois.
- \* une phase d'analyse détaillée et de programmation. Environ neuf mois également.

Concrètement, le modèle se présente actuellement comme un véritable package, ensemble de programmes paramétrés, écrits en PL/1 et exploitables sur des ordinateurs de taille supérieure ou égale à 128 K octets. Les résultats obtenus à partir de données concrètes et saisies au B.H.V. (présentées en annexe) nous semblent être la meilleure preuve de l'intérêt et de l'utilité d'un tel modèle, même si l'exemple choisi a été considérablement simplifié pour permettre au lecteur de mieux comprendre le déroulement des différentes phases de traitement. Le modèle était opérationnel le premier octobre 1971.

Remarque d'ordre pratique : Afin de faciliter la lecture, nous proposons au lecteur un lexique des différents termes qui seront introduits au cours de l'étude (cf. page suivante). Le besoin d'une terminologie particulière se justifiant par l'existence de multiples sens associés à des mots que nous voulions pouvoir utiliser pour définir les notions de base du modèle. Nous demandons donc au lecteur de bien vouloir admettre ces quelques définitions qui n'ont rien de théorique, mais dont le seul but est de définir un vocabulaire commun relativement facile à retenir.

## LEXIQUE

Terme	Notation	Définition
Moyen	M	Produit fini disponible sur le marché et capable de satisfaire un certain nombre de fonctions
Elément opérationnel de moyen	E.O.M.	Partie d'un moyen capable de satisfaire une ou plusieurs fonctions
Processus	P	Ensemble de fonctions élémentaires
Fonction	F	
Elément de solution	E.S.	Couple (E.O.M.-F) compatible (la fonction F appartient à l'ensemble des fonctions satisfaites par l'E.O.M.)
Solution	S	Ensemble d'éléments de solution
Solution partielle	SP	Solution attachée à un processus ou à un moyen donné
Solution globale	SG	Solution attachée à l'ensemble des processus (ou des moyens) de l'application (selon le type d'adaptation)
Espace général des solutions		Ensemble de tous les éléments de solution d'une application donnée
Adaptation		Mode suivant lequel l'évaluation est réalisée : EXTERNE, les moyens sont évalués par rapport aux processus, INTERNE cas inverse
Machine idéale	MI	Ensemble d'E.O.M.(F) en adaptation externe (interne) satisfaisant l'ensemble des processus (satisfait par l'ensemble des moyens)
Machine idéale existante	MIE	Ensemble de moyens contenant la MI et le plus réduit possible
Critère	C	Elément pour lequel l'évaluation des éléments de solution est effectuée
Note	N	Résultat de l'évaluation d'un élément de solution par rapport à un critère donné
Distance	D	Notion permettant d'apprécier le degré de compatibilité entre deux moyens quelconques
Incompatibilité		Notion permettant d'apprécier le degré d'incompatibilité de deux E.O.M. quelconques

Implication		Notion permettant d'apprécier le degré de compatibilité de deux E.O.M. quelconques
Système des moyens	SM	Ensemble structuré* des moyens de l'application
Système des fins	SF	Ensemble structuré* des processus de l'application

\* : "structuré", car il faut tenir compte de certaines relations existant entre les moyens et les processus (compatibilité pour les premiers, priorité pour les seconds).

## 1.2. Définitions

Ce rapport a pour but l'étude d'un modèle d'évaluation de systèmes d'organisation.

La première difficulté consiste à définir de façon précise ce que signifie pour nous "évaluation" et "système d'organisation". Difficulté, car notre but n'est pas, à ce stade, d'expliquer ce que peut faire concrètement le modèle, mais d'en dégager la philosophie.

Bien que le terme de système soit très difficile à définir, car employé trop souvent dans le langage courant, c'est le concept d'évaluation qui nous paraît, à priori, le plus complexe à cerner.

L'action d'évaluer possède, d'après le dictionnaire, deux sens distincts :

- porter un jugement sur la valeur de ...
- fixer approximativement, apprécier.

C'est au premier sens que nous nous intéresserons plus particulièrement, en essayant d'en déterminer le contenu exact.

Tout jugement doit être rapporté à la ~~personne~~ personne qui l'a émis, il n'existe pas de jugement absolu. Dans cette hypothèse, un jugement n'est pour nous que la projection d'un certain nombre de concepts sur l'objet que l'on juge. Cette notion de projection nous amène à découvrir le caractère relatif de toute évaluation : on ne peut évaluer un objet que par rapport à autre chose (généralement par rapport à soi-même), il n'y a pas d'évaluation absolue, et la valeur résultat ne peut être considérée comme valide que si l'évaluateur a su discerner correctement, dès le départ, non seulement ce qu'il évaluait, mais aussi en fonction de quoi il voulait faire son évaluation.

Dans le cas particulier du modèle, l'idée de départ a été empruntée à MILES qui nous apprend (dans son ouvrage VALUE ANALYSIS) à distinguer dans tout objet, la fonction de l'objet -c'est-à-dire ce qui justifie son existence - de l'objet lui-même - c'est-à-dire des fonctions nécessaires à sa réalisation -. Cette distinction nous apparaît fondamentale, car elle prouve effectivement qu'évaluer un objet considéré comme un moyen n'a aucun sens si l'on n'a pas défini préalablement les fonctions pour lesquelles cette évaluation doit être faite.

En d'autres termes, on ne peut évaluer un moyen en tant que tel, mais un couple moyen-fonction, ou plus simplement : émettre un jugement sur la valeur d'un moyen est absurde si l'on ne précise pas quelle est la fonction de référence. Si je dis : "le stylo que je viens d'acheter est bon", cela n'a aucune signification, car la fonction pour laquelle "il est bon" n'est pas précisée ; il est bon pour la fonction "écrire", il est mauvais par contre pour la fonction "se laver les dents".

Il est donc clair qu'évaluer un moyen consiste à porter un jugement, non pas sur la valeur du moyen lui-même, mais sur la valeur d'un couple moyen-fonction. Nous appellerons ultérieurement un tel couple : élément de solution. Par extension, il est clair également que l'évaluation d'une fonction n'a de sens que si l'on définit préalablement les moyens de référence (ce sont, en principe, ceux qui sont capables de satisfaire la fonction). En ce qui concerne le jugement lui-même, ce dernier peut être émis dans plusieurs "domaines" pour un élément de solution donné. Nous appellerons ces domaines les critères d'évaluation. Le couple moyen-fonction : "stylo-écrire" peut être ainsi "évalué" par rapport à plusieurs critères : rapidité, qualité, sécurité ...

En conséquence, toute évaluation comporte deux phases bien distinctes :

- recensement et définition des éléments de solution à évaluer (phase de pré-évaluation),
- évaluation proprement dite de chaque élément de solution par rapport à l'ensemble des critères (phase de notation).

Le but de cette phase de notation est d'associer à chaque élément de solution, une note globale par rapport à un ensemble de critères hétérogènes (Electre - C.P.E.).

D'une telle conception de l'évaluation découle l'objectif premier de toute "Analyse de la valeur", à savoir l'identification des coûts superflus. En effet, d'après MILES, on peut associer à tout objet quatre types de valeur :

- 'Use value' (valeur d'utilisation), caractérisée par l'ensemble des propriétés et qualités de l'objet lui permettant d'accomplir certaines fonctions.
- 'Esteem value' (valeur d'estime), ensemble des propriétés qui rendent l'objet estimable (désirable).

- 'Cost value' (assimilable au prix de revient) caractérisée par l'ensemble des coûts nécessaires à la réalisation de l'objet.

- 'Exchange value' (valeur d'échange), ensemble des propriétés qui permettent de comparer l'objet à d'autres objets existants et éventuellement de l'échanger.

La "valeur" de l'objet peut être alors assimilée en première approximation aux coûts nécessaires à l'obtention d'une certaine "use value" et "esteem value" fixées à priori.

*Pour MILES, la question fondamentale est alors de savoir si l'ensemble des fonctions (caractérisant la use value et l'esteem value) est assuré au meilleur coût. Pour cela, il faut avoir avant toute chose identifié ces fonctions, faute de quoi le processus ne correspond plus à une "analyse de la valeur" mais tout simplement à une "analyse des coûts". Seule la cost value entre alors en jeu.*

*En conséquence, l'identification des fonctions et l'évaluation de couples moyens-fonctions qui en découle est la condition nécessaire à toute identification des coûts superflus.*

Il nous faut maintenant revenir en détail sur la notion de "système d'organisation".

Un système d'organisation se présente pour nous comme un ensemble structuré de moyens et d'objectifs. L'ensemble des moyens sera appelé système des moyens (SM), l'ensemble des objectifs, système des fins (SF).

Le terme de système a été choisi conformément à la terminologie structuraliste qui veut qu'un système soit défini non seulement par la liste des éléments qu'il contient (ceci suffit pour définir un ensemble), mais aussi par les relations qui existent entre ces éléments (un système est donc un ensemble muni de certaines relations, ou encore ensemble structuré).

Le terme organisation a été choisi dans son premier sens (1) : doter d'un mode de fonctionnement, ce qui signifie pour nous associer à certains objectifs certains moyens capables de les satisfaire (ou associer à certains moyens certains objectifs satisfaits par ces moyens).

---

(1) - Le second sens étant : ordonner, arranger. Le modèle présenté ici n'a aucun rapport avec les modèles de type ordonnancement.

Il y a donc deux types d'évaluation possibles d'un système d'organisation :

- L'évaluation du "système moyens" en fonction du "système fins", le résultat consistant à déterminer les moyens permettant de satisfaire les objectifs du SF.

- L'évaluation opposée (du SF en fonction du SM) qui a pour but de déterminer les objectifs satisfaits par les moyens du SM.

Dans les deux cas, le résultat de l'évaluation doit permettre de classer les couples moyens-objectifs par rapport aux différents critères et de réaliser une optimisation. En ce sens, le modèle qui fait l'objet de cette étude va plus loin qu'une simple évaluation : il permet de déterminer la solution idéale.

Mais avant d'étudier le modèle lui-même et ses différentes phases (cf. chapitre IV), il convient de définir l'ensemble des variables permettant de caractériser un système d'organisation et leur mode de saisie (cf. chapitre III, documents d'entrée du modèle), ainsi que l'intérêt et les objectifs concrets d'un tel modèle. Ceci nous amènera plus précisément à définir ce que représente un système d'organisation dans l'entreprise (cf. chapitre II).

En conclusion, le plan de l'étude aura la forme classique suivante :

- a) Etude des résultats à obtenir (objectifs).
- b) Etude des informations nécessaires à l'obtention des résultats (données d'entrée du modèle).
- c) Etude des moyens de traitement permettant d'obtenir les résultats à partir des données d'entrée.



## CHAPITRE II. OBJECTIFS DU MODELE

Ce que MILES a analysé au niveau de l'objet peut être étendu à celui des entreprises. En effet, l'intérêt de sa méthode réside dans la distinction qu'il établit entre objet et fonction. Cette distinction demeure valable au niveau de l'entreprise si l'on considère cette dernière comme un outil destiné à accomplir des opérations qui justifient son existence.

On retrouve ici le parallèle établi précédemment entre les fins et les moyens d'un système d'organisation : la fonction de l'objet étant dans cette perspective le SF, l'objet lui-même étant l'outil ou l'ensemble d'outils permettant d'atteindre (éventuellement) le SF, soit le SM. Un ordinateur, par exemple, est un moyen permettant de satisfaire un objectif : le traitement d'une information donnée.

Une difficulté apparaît à ce niveau. Elle est liée à la hiérarchisation et à l'imbrication des fins (objectifs primaires et secondaires). En particulier, lorsque l'on passe d'un système à un sous-système d'ordre inférieur, on peut être amené à considérer certains moyens comme des objectifs du nouveau système : c'est le changement classique des moyens en objectifs dû à l'augmentation du degré de finesse du système (problème bien connu de l'analyste, au niveau de la définition des objectifs de gestion et des objectifs de traitement, ces derniers n'étant que des moyens permettant de satisfaire les premiers). Cette difficulté sera résolue indirectement, en ce qui nous concerne, par la signification précise donnée aux systèmes-fins et systèmes-moyens que nous nous proposons d'évaluer.

Le système des fins est un ensemble d'objectifs, il s'exprime en termes d'exigences. Chaque objectif se présente comme une suite de fonctions à réaliser. Ces fonctions ne sont évidemment pas indépendantes, elles sont en particulier ordonnées dans le temps, et pour cette raison nous appellerons désormais une telle suite de fonctions : un processus. Un SF est donc, à ce stade, un ensemble structuré de processus (il est clair, en effet, qu'il existe des relations entre les processus, ne serait-ce que des relations de priorité).

Le système des moyens est un ensemble d'outils, simples ou complexes, à l'intérieur duquel chaque outil peut être représenté comme un ensemble d'outils

plus élémentaires, appelés "éléments opérationnels de moyens", ces derniers possédant la propriété de pouvoir assurer, dans un certain contexte, mais indépendamment des autres, certaines fonctions. Un élément opérationnel de moyens est, en quelque sorte, la plus petite partie d'un outil, se suffisant à elle-même pour assurer une fonction donnée (cette définition n'est pas tout à fait correcte, elle sera reprise et explicitée ultérieurement au chapitre V lorsque nous aborderons la théorie de la modularité.

De la rencontre d'un système-fins et d'un système-moyens, va résulter le système concret, "organisé", totalement défini.

*Remarque : Il est clair que la définition des "fonctions" pose pour l'utilisateur un énorme problème : celui de savoir quel est le degré de finesse du système dans lequel il se place. Il ne nous appartient pas de résoudre de façon théorique ce problème. En ce qui concerne l'utilisation effective du modèle, la solution consistera d'abord à définir les e.o.m. (par décomposition des moyens du SM). Le "degré de finesse" des fonctions en découle alors automatiquement.*

## 2.1. Mécanisme de l'évaluation-adaptation

Deux questions se posent maintenant au niveau de l'entreprise, en supposant que le système des fins soit clairement défini :

- quel système de moyens lui opposer ?
- lequel des deux systèmes doit avoir la priorité en cours d'évaluation ?

En d'autres termes, comme nous l'avons évoqué précédemment, doit-on évaluer les fins en fonction des moyens, ou les moyens en fonction des fins ?

Nous retrouvons ici le problème classique et fondamental de l'adaptation de l'un des deux systèmes à l'autre.

Nous parlerons d'adaptation externe lorsque les moyens s'adaptent aux objectifs de l'entreprise (cela revient à acheter du matériel sur mesure).

Nous parlerons d'adaptation interne lorsque les objectifs de l'entreprise s'adaptent aux moyens existants sur le marché. On conçoit que ce type d'adaptation est plus dangereux que le précédent. Dans la réalité, il est rare cependant de constater l'acquisition par une entreprise de matériel fait sur mesure, et, dans la plupart des cas, on peut déplorer qu'il y ait eu effectivement adaptation des

fins aux moyens ; le malheur étant que cette adaptation est souvent très difficile à percevoir. Dans le domaine de l'informatique, en particulier, il est courant de rencontrer des entreprises qui "s'adaptent" petit à petit à la "présence d'un ordinateur".

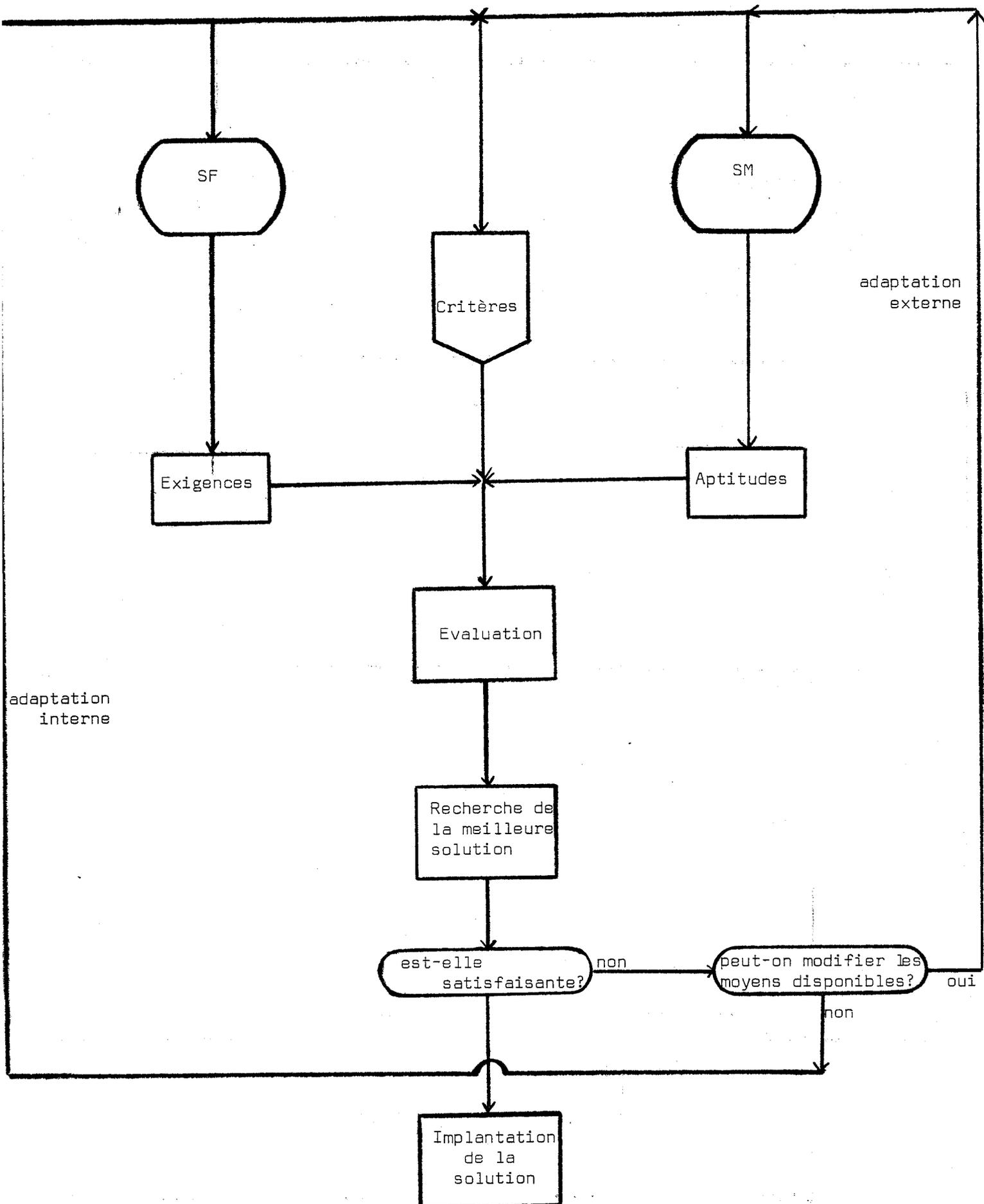
L'adaptation externe, quant à elle, rencontre souvent une farouche résistance au sein des entreprises, symbolisée par la réaction bien connue de réticence au changement, d'autant plus que c'est à l'entreprise elle-même qu'il incombe de définir les normes du matériel (sur mesure) dont elle devrait se doter. Elle représente néanmoins la tendance future vers laquelle se sont déjà orientés certains constructeurs. Ces efforts de leur part pour chercher à faire du "sur mesure" (sans pour autant négliger les avantages de la "série") débouchent sur la construction de matériel modulaire. C'est en fait l'établissement d'une véritable théorie de la modularité qui pourra permettre, à notre avis, la résolution effective de ce problème fondamental.

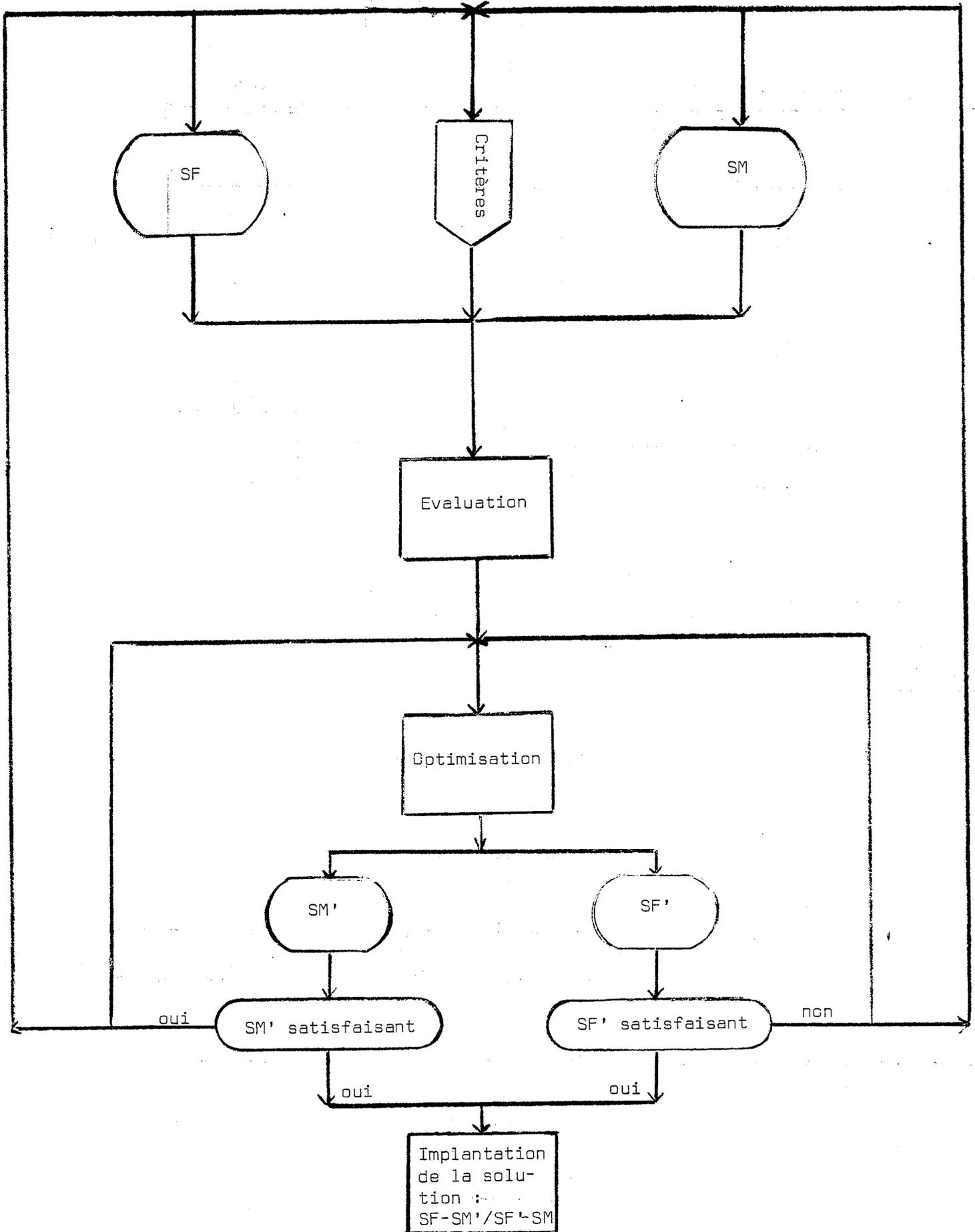
Quoi qu'il en soit, il semble, en ce qui nous concerne, que le fonctionnement du modèle doive répondre au schéma suivant (cf. schéma 1) : Sur ce schéma, le SF représente l'ensemble des objectifs à atteindre, le SM l'ensemble des moyens disponibles sur le marché et susceptibles de satisfaire tout ou partie des objectifs du SF.

Lorsque la "meilleure solution" fournie par le modèle ne satisfait pas l'utilisateur, celui-ci peut :

- soit rechercher de nouveaux moyens, c'est à dire faire une nouvelle évaluation avec un nouvel SM : nous parlerons alors d'adaptation externe ;
- soit modifier les objectifs du SF, c'est à dire faire une nouvelle évaluation avec un nouvel SF : nous parlerons alors d'adaptation interne.

Si l'on y regarde de plus près, le schéma 1 n'est pas tout à fait satisfaisant, car il introduit une dissymétrie entre les fins et les moyens. En effet, dans les deux cas envisagés, l'évaluation est faite "à partir" des moyens "en fonction" des fins. En d'autres termes, le type d'adaptation réalisé est connu de l'utilisateur mais n'intervient pas dans le modèle. En conséquence, le fonctionnement du modèle répondra au schéma 2 dans lequel la symétrie est totale.





Ce schéma établit clairement que, suivant le type d'adaptation souhaité, la réponse du modèle n'est pas la même :

- en adaptation externe, le modèle fournit comme solution un ensemble de moyens (noté SM') permettant de satisfaire, plus ou moins parfaitement, le SF de référence (ce dernier étant considéré comme rigide pendant le déroulement des opérations) ;
- en adaptation interne, le modèle fournit comme solution un ensemble de processus (noté SF') satisfaits par le SM de référence.

Il nous faut maintenant préciser de quelle façon sont construits SM' et SF' (respectivement à partir de SM et de SF), et étudier préalablement sous quelle forme ils peuvent être exprimés par le modèle. Malgré la symétrie du schéma 2, il nous faut étudier séparément les deux types d'adaptation puisque les "sorties" du modèle ne sont pas les mêmes.

## 2.2. Résultats à obtenir

### a) adaptation externe

Le but du modèle est de fournir un ensemble de moyens capables d'assurer toutes les fonctions de tous les processus du SF. Cet ensemble de moyens SM' peut s'exprimer de deux façons différentes :

- comme un ensemble d'outils Mi extraits du SM de référence, donc existants,
- comme un ensemble d'éléments opérationnels de moyens (e.o.m.) contenus dans les outils Mi du SM de référence.

A ce niveau la nuance est fondamentale. En effet, dans le premier cas, le SM' représente le sous-ensemble d'outils existants (extraits du SM) permettant de satisfaire au mieux le SF de référence. Dans le second cas, le SM' représente la combinaison idéale d'e.o.m. satisfaisant le SF de référence. Dans le cas général, cet "outil idéal" n'existe pas sur le marché, mais il est composé d'e.o.m. issus d'outils existants. Ceci sous-entend, en particulier, que l'outil idéal n'est défini que par rapport aux e.o.m. de référence. Nous insistons sur le fait que le but du modèle n'est pas de "créer" de nouveaux e.o.m. mais de définir des combinaisons d'e.o.m. n'existant pas sur le marché et capables de satisfaire totalement un ensemble de fonctions (représenté par le SF).

Dans la pratique, il est extrêmement intéressant pour l'utilisateur de connaître le SM' sous ses deux formes possibles. En effet, si l'outil idéal représente indubitablement la meilleure solution (en théorie), sa réalisation concrète peut s'avérer impossible (contraintes technologiques par exemple), et il faut laisser à l'utilisateur la possibilité de se rabattre sur le "meilleur" matériel existant (qui correspond à la première forme d'expression du SM') que nous appellerons désormais (par analogie avec ce qui précède) "l'outil idéal existant".

En ce qui concerne l'obtention pratique du SM', l'évaluation étant faite pour chaque couple "fonction-e.o.m.", c'est l'outil idéal qui est obtenu en premier. Le modèle calcule alors l'outil idéal existant comme étant l'ensemble des moyens  $M_i$  extraits du SM contenant tous les e.o.m. de l'outil idéal et le moins possible de gadgets, c'est à dire tel que le nombre d'e.o.m. appartenant à l'outil idéal existant et n'appartenant pas à l'outil idéal soit minimum (voir exemple en fin de chapitre).

Nous reviendrons ultérieurement en détail sur les problèmes posés par la recherche effective de l'outil idéal existant (à partir de l'outil idéal) avec un tel critère d'optimisation.

#### b) adaptation interne

Par analogie avec ce qui précède, le but du modèle est de fournir, dans une première étape, l'ensemble des fonctions satisfaites par tous les e.o.m. de tous les moyens du SM. Ceci constitue la première forme d'expression du SF'.

A partir de cet ensemble (noté F), le modèle détermine l'ensemble (P) des processus existants (issus du SF) tel que toute fonction de P appartienne à F (afin d'être satisfaite par le SM) et tel que le nombre des fonctions appartenant à F et pas à P soit minimum.

Sous sa seconde forme, le SF' représente l'ensemble des processus existants, "le plus grand possible", satisfaits par un SM donné.

Il va de soi que l'adaptation interne présente beaucoup moins d'intérêt que l'adaptation externe. Nous pensons que son utilisation devrait se limiter aux études demandées par les constructeurs.

Remarques :

1) Il est clair que quel que soit le type d'adaptation désiré, la solution (SF' ou SM') n'est pas obligatoirement unique. Ceci explique l'existence d'une éventuelle seconde boucle (cf. schéma 2) au niveau de la phase d'optimisation. Il est à noter cependant, qu'à notre avis, l'utilisateur n'a pas intérêt à utiliser cette boucle, c'est à dire à s'accommoder du premier SF' (ou SM') satisfaisant. L'intérêt du modèle n'est pas de fournir à l'utilisateur le rang de la solution qu'il peut implanter, mais bien de le pousser à choisir la meilleure (en recommençant l'évaluation, au besoin).

2) En ce qui concerne la première boucle (celle qui consiste à refaire une évaluation en ré-injectant dans le modèle un nouvel SF ou un nouvel SM), on conçoit aisément qu'elle aurait tout intérêt à se dérouler en conversationnel. A cet effet, une seconde version du modèle est prévue dans un mode conversationnel ; nous y reviendrons ultérieurement.

3) Avant de lancer une évaluation, il est parfois utile de réaliser une étude sommaire et grossière des systèmes fins et moyens en présence.

Ainsi, par exemple, si l'on constate que le système fins est très malléable (structure administrative et humaine, souple et disciplinée), et que, par ailleurs, les moyens existants sont peu malléables et répondent bien au système fins, il est inutile d'envisager une adaptation externe. Il suffit de choisir dans les gammes de matériels existants celle qui convient le mieux aux besoins et d'en surveiller soigneusement l'application, en apportant s'il le faut des modifications mineures. Cette remarque sous-entend, en particulier, que l'intérêt du modèle est d'autant plus réduit que la combinatoire entre les fins et les moyens est limitée, ce qui n'est pas évident a priori.

Exemple :

Afin de mieux faire comprendre au lecteur ce que représentent effectivement les sorties du modèle, nous allons étudier un exemple concret :

Soit un système fins composé de 3 processus P1, P2, P3 dont le découpage en fonctions s'établit comme suit :

P1 = f1, f2, f3

P2 = f2, f4, f5

P3 = f6, f7

Supposons que l'ensemble des moyens disponibles sur le marché, capables de satisfaire tout ou partie du SF, soit formé de 3 "machines" : M1, M2, M3 dont la décomposition en éléments opérationnels de moyens est la suivante :

M1 : m1, m2, m3

M2 : m4, m5, m6

M3 : m7, m8, m9

Il y a 72 éléments de solutions "possibles". La phase de pré-évaluation fait disparaître ceux d'entre eux qui sont "incompatibles" (ceci signifie : élément de solution pour lequel le moyen n'est pas du tout adapté à la fonction).

La phase de recherche des solutions permet de déterminer pour chaque processus (si l'on est en adaptation externe) les ensembles d'e.o.m. capables de satisfaire chacun d'eux.

Supposons que les solutions soient les suivantes :

pour P1 S1 = m1, m2, m6

S2 = m1, m5, m6

S3 = m1, m5, m9

pour P2 S1 = m2, m6, m9

S2 = m5, m6, m8

pour P3 S1 = m5, m8

S2 = m5, m9

S3 = m7, m9

A chacune des solutions est associée une note calculée par le modèle et qui tient compte, en particulier, des notes élémentaires associées à chaque élément de solution (phase de notation).

Les solutions S sont appelées solutions partielles car elles sont associées à un processus déterminé.

Nous les noterons désormais  $S_{k,j}$  où k représente l'indice du processus, et j le rang de la solution pour le processus k.

A ce niveau, donc, le modèle travaille sur le tableau suivant :

S1,1	de note	N1	S2,2	de note	N5
S1,2	-----	N2	S3,1	-----	N6
S1,3	-----	N3	S3,2	-----	N7
S2,1	-----	N4	S3,3	-----	N8

Une solution générale du système (notée SG) est un ensemble d'éléments  $S_{k,j}$  tel que si  $p$  est le nombre de processus, pour tout entier  $l$  compris entre 1 et  $p$  (bornes incluses), il existe un  $S_{k,j}$  et un seul tel que  $k = l$ . (En d'autres termes, une solution générale est une combinaison quelconque de solutions partielles attachées à chaque processus du SF).

Supposons

$$\begin{aligned} N2 &= \text{MAX} (N1, N2, N3) \\ N5 &= \text{MAX} (N4, N5) \\ N6 &= \text{MAX} (N6, N7, N8) \end{aligned}$$

La meilleure des solutions générales est dans ce cas :

$$SG1 = \begin{pmatrix} S1,2 \\ S2,2 \\ S3,1 \end{pmatrix}$$

(de note globale  $N2 + N5 + N6$ ) en première approximation.

Pour satisfaire SG1 il faut disposer de :  $m1, m5, m6, m8$ . Dans la mesure où ces éléments opérationnels de moyens ne sont pas "incompatibles", ils constituent la "machine idéale" (associée aux processus P1, P2, P3).

Ceci constitue la première "sortie" du modèle. (Dans le cas où la configuration associée à SG1 est incompatible, le modèle calcule la solution générale d'ordre inférieur, c'est à dire la meilleure après SG1 - soit SG2 -, et ainsi de suite jusqu'à l'obtention d'une SG "compatible").

Le modèle calcule alors la "machine idéale existante". Celle-ci doit posséder tous les e.o.m. de la machine idéale et un nombre minimum de gadgets (c'est à dire le moins possible d'autres e.o.m.).

Dans notre exemple, la "M.I.E." est constituée des machines M1, M2, M3, ce qui signifie que la meilleure solution consiste à se doter des trois machines existant sur le marché (faute de pouvoir construire la M.I.) afin de pouvoir disposer des éléments opérationnels de moyens  $m1, m5, m6, m8$ . Dans cette hypothèse, les e.o.m. :  $m2, m3, m4, m7, m9$  sont des gadgets. Vu leur importance, l'utilisateur a alors tout intérêt, dans ce cas particulier, à faire en sorte que la construction de la M.I. soit possible.

Dans le but de donner à l'utilisateur un maximum d'informations, le modèle fournira en fait non seulement la M.I.E., mais toutes les combinaisons de moyens disponibles sur le marché permettant d'assurer les fonctions de la M.I. A chaque combinaison sera associé l'ensemble de ses gadgets (e.o.m. non nécessaires). Dans notre exemple, la seule combinaison possible est précisément la M.I.E.

En conclusion, pour ce cas particulier, le document de sortie du modèle serait le suivant :

1) La machine idéale est composée des éléments opérationnels de moyens suivants :

m1 (libellé)  
m5 -----  
m6 -----  
m8 -----

La note globale associée à cette solution par le modèle est : N

Cette solution consiste à satisfaire

le processus P1 par m1, m5, m6 Note : N2  
le processus P2 par m5, m6, m8 ----- N5  
le processus P3 par m5, m8 ----- N6

2) La machine idéale existante est composée des moyens M1, M2, M3 (libellé).

Les e.o.m. : m2, m3, m4, m7, m9 ne sont pas nécessaires.

3) Il n'existe pas d'autres combinaisons possibles.

Le mécanisme détaillé de l'évaluation n'a pas encore été abordé, et ceci volontairement. Néanmoins, comme nous l'avons dit au cours du chapitre I, il demeure la clé de voûte du modèle, puisqu'il en explique en particulier la dualité des entrées et la symétrie.

En conclusion, bien que les "entrées" du modèle n'aient pas encore été définies, nous espérons que le lecteur en a compris le sens profond et l'intérêt. Les conditions générales de fonctionnement et les objectifs du modèle étant posés, il nous faut maintenant rentrer réellement dans les détails, en commençant par étudier l'ensemble des informations nécessaires à la réalisation d'une telle évaluation.



### CHAPITRE III. LES DONNEES D'ENTREE DU MODELE

Comme nous l'avons vu précédemment, les deux notions fondamentales de toute application sont, d'une part, les fonctions (pour le SF), d'autre part, les éléments opérationnels de moyens (pour le SM). Au niveau de l'évaluation, une troisième notion intervient : les éléments de solution, qui résultent de l'association "fonction-e.o.m.". La détermination des éléments de solution est réalisée de façon automatique par le modèle, mais leur notation est faite par l'utilisateur, ce qui explique qu'il existe, en réalité, deux niveaux de saisie des données. L'existence de cette double saisie nous amène donc à parler succinctement du principe de fonctionnement, avant de définir plus précisément les données nécessaires à l'évaluation globale.

On peut considérer qu'il existe en fait deux modèles :

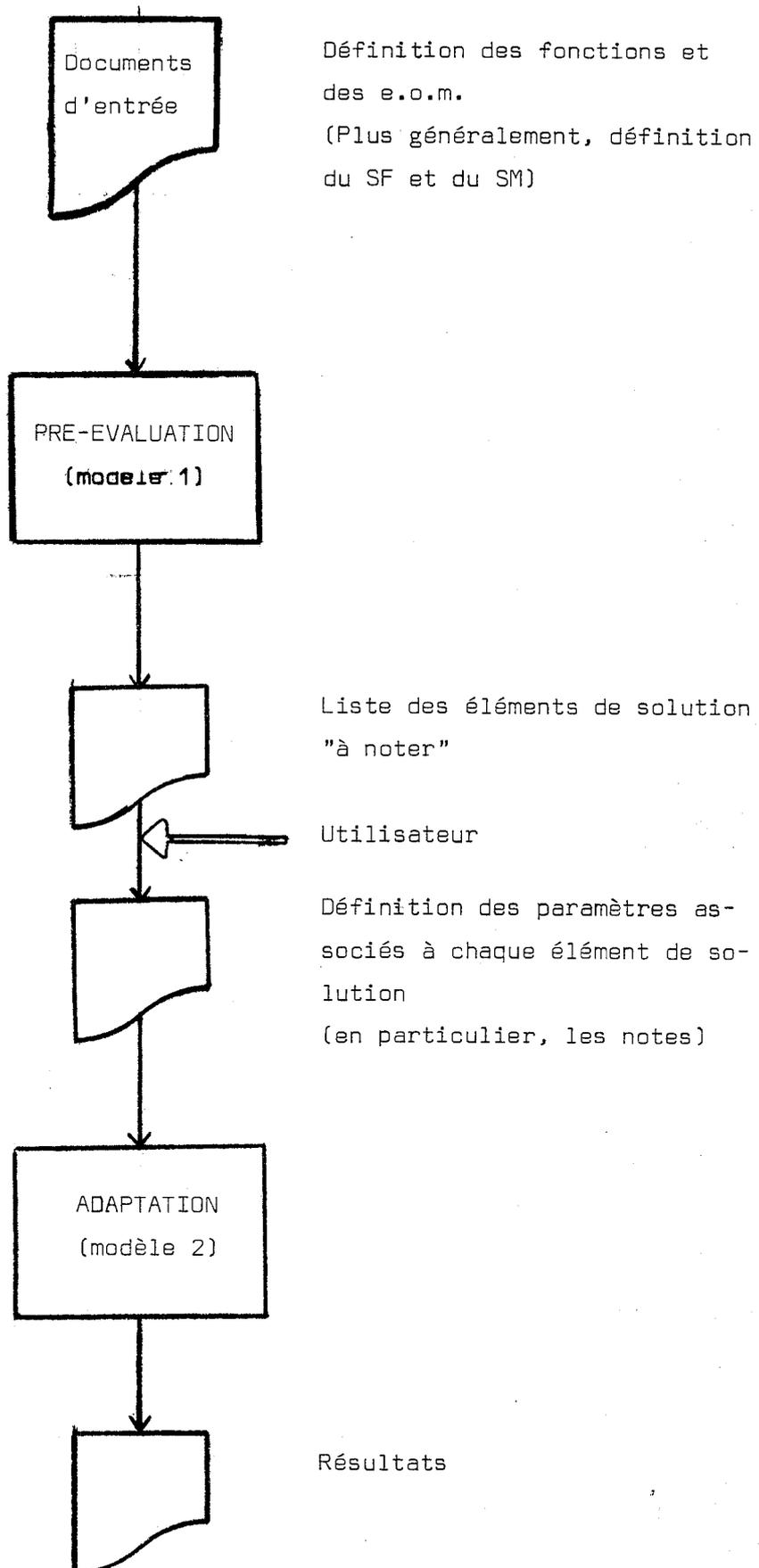
- Le modèle 1 chargé d'effectuer la "pré-évaluation", c'est à dire déterminer les éléments de solution non incompatibles.
- Le modèle 2 chargé d'effectuer l'adaptation, c'est à dire rechercher les différentes solutions existantes et délivrer la machine idéale.

Entre les deux, une intervention humaine est obligatoire, puisque l'utilisateur doit connaître les éléments de solution pour pouvoir les noter. A cet effet, le document de sortie du modèle 1, une fois annoté par l'utilisateur, servira de document d'entrée au modèle 2.

On peut résumer ce mode de fonctionnement (qui sera repris en détail au chapitre IV) par un schéma (voir page suivante).

Un tel schéma permet de mettre en évidence l'un des problèmes fondamentaux de cette phase de définition des données : celui de la codification des fonctions et des e.o.m. Cette codification, si elle est souhaitable au niveau de la formalisation, est nécessaire pour la détermination automatique des éléments de solution. Son importance est capitale. En effet, sans codification, la définition des données d'entrée devient rapidement inextricable pour l'utilisateur puisque ce dernier doit définir lui-même les éléments de solution non incompatibles. Le modèle perd alors une grande partie de son intérêt.

Nous pouvons maintenant aborder en détail la définition des données d'entrée.

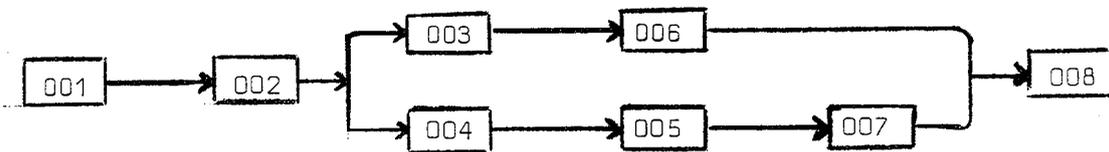


### 3.1. Définition du SF

#### 3.1.1. Définition du catalogue des processus

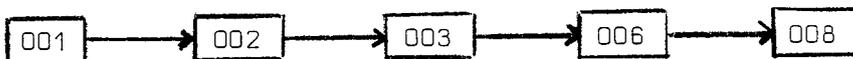
Le SF se présente au niveau macroscopique comme un ensemble de processus d'importance inégale vis-à-vis de l'application elle-même. Chaque processus est donc caractérisé par un libellé, un poids et un numéro d'ordre. Le processus lui-même est un ensemble structuré de fonctions et peut se représenter comme un graphe orienté de fonctions, dans lequel les noeuds sont les fonctions, les arcs traduisant la logique du processus. En réalité, il a été admis, pour des raisons de simplification, de ne pas travailler au niveau du graphe, mais directement sur l'ensemble des chemins qui le composent. Pour le modèle, un processus sera donc un chemin du graphe et non le graphe lui-même.

Ainsi, par exemple, le processus "utilisateur" suivant :

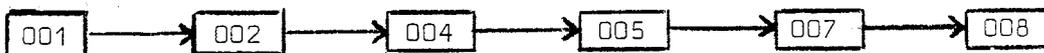


- dans lequel la fonction 001 est la fonction initiale, la fonction 008 la fonction terminale et la fonction 002 une fonction test qui permet d'aiguiller le traitement sur l'une des deux branches - sera décomposé en deux processus distincts :

le processus P1

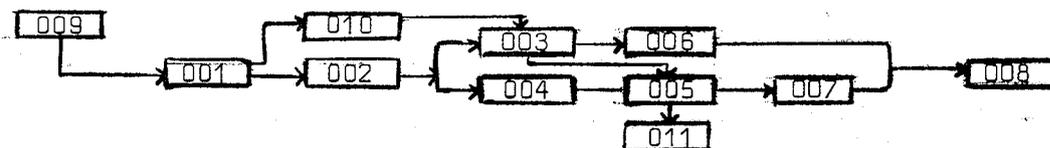


et le processus P2



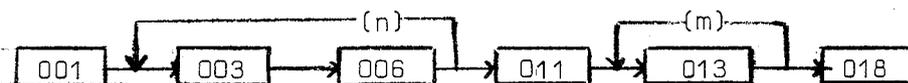
Cette décomposition, qui ne nuit en rien à la généralité du problème, ne demande pas, par ailleurs, de travail supplémentaire à l'utilisateur. Celui-ci, en effet, raisonne à un niveau global, surtout lorsque les processus sont complexes, et la méthode s'avère nettement avantageuse lorsqu'une (ou plusieurs) même fonction intervient dans plusieurs processus. Dans ce cas-là, effectivement, la décomposition du graphe global est nécessaire.

Ainsi, par exemple, le graphe suivant nécessite obligatoirement une décomposition (faute de quoi il subsiste une ambiguïté au niveau des chaînes de fonctions à "évaluer").



Dans ce graphe, la fonction 001 n'est pas une fonction test, elle demeure la fonction initiale du sous-graphe précédent et la deuxième fonction du chemin de fonction initiale 009.

La simplification annoncée précédemment permet enfin de considérer le processus au niveau du modèle comme une suite de fonctions, et l'intérêt pour l'utilisateur de travailler à un niveau purement séquentiel n'est pas négligeable. Les notions de fonctions initiales et terminales disparaissent alors. Un dernier problème restait à résoudre : celui des boucles. Il peut arriver, en effet, que, sur un chemin donné, une fonction (ou une suite de fonctions) doive être effectuée plusieurs fois, ce qui se représente par le schéma suivant :



où  $n$  et  $m$  sont les facteurs d'itération.

Afin de pouvoir traiter ce cas particulier, relativement fréquent pour des processus de type "grands magasins", il a été envisagé d'associer à chaque fonction, à l'intérieur d'un processus déterminé, un poids représentant de façon générale l'importance de la fonction dans le processus.

Dans l'exemple précédent, les fonctions 003 et 006 seraient affectées, par exemple, du poids  $n$ , et la fonction 013 du poids  $m$ . Mais on pourrait tout aussi bien associer à la fonction 011 un poids  $p$  éventuellement supérieur à  $m$  et  $n$ , si l'on estime que la fonction 011 est la fonction charnière du processus.

En conséquence, chaque processus sera caractérisé par une suite de fonctions (définies par leur numéro d'ordre) et une suite de poids attachés à chacune de ces fonctions.

Le premier document d'entrée du modèle aura donc la forme suivante :  
(ce document permet de constituer le fichier "catalogue des processus")



Exemple de processus :

Processus : "Elaboration d'un ticket"

Ce processus est constitué de six fonctions qui correspondent aux différentes opérations nécessaires à l'établissement d'un ticket de caisse.

- (1) Définition de l'en-tête du ticket : celle-ci contient plusieurs informations : la date, le code rayon, le numéro d'ordre, le numéro de poste et le nom du magasin.
- (2) Définition du document de transaction.
- (3) Définition de la modalité de paiement.
- (4) Prise en charge de la quantité de produit acheté et du prix unitaire.
- (5) Calcul des sous-totaux, du montant de la T.V.A. et du total.
- (6) Prise en charge de la monnaie reçue.

(Voir en annexe d'autres exemples).

3.1.2. Définition du catalogue des fonctions

Chaque fonction est définie indépendamment du (ou des) processus auquel elle appartient par trois paramètres : son numéro d'ordre (qui permet d'établir la liaison avec le catalogue des processus), son code et son libellé.

\* Problèmes posés par la codification des fonctions

Comme nous l'avons vu, l'intérêt de la codification réside non seulement dans le fait qu'elle permet de formaliser la définition des fonctions et d'établir une base pratique de comparaison (l'utilisateur peut, en particulier, définir des fonctions "a priori" sans tenir compte des processus actuels - ce qui est très avantageux dans l'hypothèse d'une adaptation interne -), mais aussi parce qu'elle permet de déterminer automatiquement les éléments de solution à évaluer.

Pour cela, la première contrainte à respecter était de rechercher un code commun (structure identique) valable pour les fonctions et les éléments opérationnels de moyens. La deuxième contrainte consistait à trouver un code suffisamment général pour être appliqué à d'autres problèmes que ceux du type "grands magasins - caisses enregistreuses". Faute d'avoir pu tester le modèle sur des applications tout à fait différentes, nous ne pouvons répondre à ce problème de façon catégorique. Il semble néanmoins souhaitable (voire nécessaire dans certains cas) de changer la codification et de l'adapter au type d'application envisagée, ne serait-ce que pour des raisons de performance de programmes.

En ce qui concerne la première contrainte, elle a été en partie satisfaite en adoptant effectivement une structure identique pour les codes fonctions et les codes e.o.m. En réalité, comme nous allons le voir, certaines zones ont une signification différente suivant qu'il s'agit d'une fonction ou d'un e.o.m.

\* Codification proposée

Le code utilisé est divisé en 5 zones, la dernière de ces zones étant elle-même composée de 3 sous-zones et pouvant se répéter plusieurs fois.

ZONE 1 - C'est la zone opérateur. Le code opérateur permet de spécifier le type de traitement correspondant à la fonction (ou à l'e.o.m.).

Le code traitement (ou code opérateur) peut prendre une valeur décimale comprise entre 1 et 99.

Ainsi, par exemple :

- 01 signifie entrée de données
- 02 -- sortie de données  
(dans le cas d'interfaces hommes/machines)
- 03 signifie entrée de données
- 04 -- sortie de données  
(dans le cas d'interfaces systèmes/terminal)
- 05, 06, etc... correspondent à des types de traitement particulier

ZONE 2 - permet de spécifier le degré d'automatisation de l'information.

4 valeurs sont possibles :

- 1 - information automatisée : l'opération correspondante exclut toute intervention humaine
- 2 - information mécanisée : il y a une intervention humaine assistée pour les travaux de routine
- 3 - information stable : information non-automatisée mais conservable (écrit, par exemple)
- 4 - information labile : intervention humaine sans trace matérielle (communication téléphonique non enregistrée, par exemple)

ZONE 3 - Cette zone possède deux significations différentes. Dans le cas d'une fonction, elle permet de spécifier le code du sous-système considéré :

- 1 - système des "notions neutres"
  - 2 - système "Argent"
  - 3 - système "Paiement différé"
  - 4 - système "Livraison"
  - 5 - système "Marchandises"
  - 6 - système "Contrôle productions"
- etc...

L'intérêt pour l'utilisateur est donc de pouvoir établir une liaison entre la notion rattachée à la fonction considérée et le sous-système correspondant.

Dans le cas d'un e.o.m., elle permet de spécifier le degré d'automatisation du support, et peut prendre 4 valeurs possibles :

- 1 - automatisée : manutention automatisée du support, pas d'intervention humaine
- 2 - manuelle : manutention manuelle du support, mais contrôle automatique d'erreurs
- 3 - manuelle et contrôle non automatique
- 4 - labile : pas de support matériel

Ainsi, par exemple, pour un e.o.m. permettant "d'entrer" une information automatisée, on peut avoir les trois cas suivants :

- 01.1.1. : Badge prélevé automatiquement à un endroit déterminé  
(entrée automatisée)
- 01.1.2. : Entrée manuelle du badge, mais guidée par une fente  
(contrôle d'erreurs automatique)
- 01.1.3. : Lecture par "tag reader" ou "monarch"  
(crayon lumineux que l'on passe sur l'étiquette, contrôle non automatique)

ZONE 4 - Elle permet de spécifier le nombre de zones 5 à prendre en considération.

ZONE 5 -

SOUS-ZONE 5.1 : indique le degré de formalisation de l'information. Les degrés sont codés de 1 à 9, comme suit (formalisation croissante) :

- 1 - information n'entraînant pas de décision
- 2 - information binaire
- 3 - information permettant plusieurs alternatives non ordonnées
- 4 - information décimale
- 5 - information alphabétique
- 6 - notion
- 7 - syntaxe
- 8 - information analogique bi-dimensionnelle
- 9 - sémantique

Dans le cas d'un e.o.m., c'est le degré de formalisation minimum accepté par l'e.o.m. qui est pris en compte.

SOUS-ZONE 5.2 : indique l'unité d'information :

- le bit,
- le digit décimal,
- la lettre, etc...

SOUS-ZONE 5.3 : indique la capacité (en nombre d'unités).

Il s'agit évidemment de la capacité requise par la notion rattachée à la fonction (dans le cas d'un code fonction) et de la capacité réelle offerte par l'e.o.m. (dans le cas d'un code e.o.m.).

*Remarque 1* : L'étude des différents contrôles permettant de déterminer la compatibilité entre les différentes zones d'un code fonction et d'un code e.o.m. sera envisagée au chapitre IV.

*Remarque 2* : L'utilisateur peut rendre son évaluation plus ou moins fine en jouant sur la zone 4 au niveau des codes fonctions. En effet, définir une seule zone 5 pour chaque fonction revient à associer à chaque fonction une seule notion et, en conséquence, générer autant de types d'éléments de solution que de notions. Par contre, regrouper en une seule fonction plusieurs notions revient à générer un seul type d'éléments de solution pour un ensemble de notions, et dans la solution finale, à choisir un seul e.o.m. pour satisfaire le traitement de plusieurs

notions. (Le traitement est identique pour l'ensemble des notions, puisque la zone 1 est unique).

Ceci peut être illustré par l'exemple suivant :

Soit la fonction : composition de l'en-tête du ticket (issue du processus "ticket" décrit précédemment).

Cette fonction qui a pour but d'entrer en machine les notions : date, code rayon, numéro d'ordre, numéro de poste, nom du magasin, est codifiée de la façon suivante :

01.1.1.05	}	4.10.06	date
		4.10.03	code rayon
		4.10.03	numéro d'ordre
		4.10.03	numéro de poste
		5.26.03	nom magasin (B.H.V.)

Si on la traite comme une fonction unique (c'est le cas de la codification proposée), elle sera satisfaite dans la solution délivrée par le modèle par un e.o.m. unique (celui qui permet à la fois de rentrer des informations décimales de 3 et 6 chiffres et des informations alphabétiques de 3 lettres). Si l'on éclate cette fonction en plusieurs fonctions (5 au maximum), le modèle recherchera autant d'e.o.m. que de fonctions issues de l'éclatement.

La solution obtenue sera donc beaucoup plus fine que la précédente.

En conclusion, le catalogue des fonctions aura la forme suivante (voir page suivante).

### 3.2. Définition du SM

La symétrie entre SF et SM n'est pas totale. La différence est due, en particulier, au fait qu'en général un e.o.m. donné ne peut appartenir à deux moyens différents. Par contre, on peut éventuellement trouver deux e.o.m. différents de codes identiques (cas, par exemple, d'un clavier de machine A qui possède exactement les mêmes caractéristiques que le clavier d'une machine B).

Par ailleurs, des paramètres nouveaux sont nécessaires à la définition du SM. Ceux-ci permettent de prendre en considération la notion d'incompatibilité entre e.o.m. - notion qui n'existe pas, de façon évidente, au niveau des fonctions -.



Enfin, il est de fait que, dans la plupart des cas, la définition du SM est réalisée par un utilisateur différent de celui qui a réalisé la définition du SF.

En effet, les profils requis ne sont pas les mêmes :

- Pour le SF, l'utilisateur doit avoir une très bonne connaissance des objectifs de l'entreprise et des différents circuits administratifs.
- Pour le SM, il suffira d'étudier directement à partir des catalogues proposés par les constructeurs et d'effectuer une décomposition systématique en e.o.m.

### 3.2.1. Définition du catalogue des moyens

Chaque moyen M est caractérisé par un numéro d'ordre, un libellé (qui correspond généralement à la marque), et par la liste des numéros d'e.o.m. qui le composent.

Le document correspondant (voir page suivante) permet de constituer le fichier "catalogue des moyens".

A ce stade intervient également une nouvelle notion : celle de distances entre les moyens du SM. Elle permet de prendre en considération le degré d'incompatibilité qui existe entre deux moyens quelconques du SM.

Son utilisation dans le modèle intervient au niveau du calcul de la note définitive affectée à la solution globale. Cette note est, en effet, égale à la somme des notes des solutions partielles moins la somme des distances entre les moyens intervenant dans la solution globale.

#### Exemple :

Soit la solution globale SG1 composée des 3 solutions partielles : S1, S2, S3 de notes N1, N2, N3.

L'ensemble des e.o.m. associés à SG1 est l'union des ensembles d'e.o.m. respectivement associés à S1, S2, S3 - soit : m1, m2, m3, m4.

Cet ensemble (si SG1 est la meilleure des solutions globales) n'est autre que la "machine idéale".

Supposons que	m1, m2	€	M1
	m3	€	M2
	m4	€	M3

4

Si chaque moyen a ses éléments propres indiquer '0' (zéro) dans " liste des e.o.m. "  
Le catalogue des e.o.m. doit alors être constitué par blocs, chaque bloc appartenant à un moyen donné.

CATALOGUE DES MOYENS

Société .....

Adresse .....

Date .....  
Responsable .....

PARTIE A REMPLIR PAR L'UTILISATEUR

1	M	NUMERO DU MOYEN	2	
1	M	NOMBRE D'E.O.M.	6	
1	M	LISTE DES E.O.M.	9	
1	M	LIBELLE DU MOYEN	6	

1	M	NUMERO DU MOYEN	2	
1	M	NOMBRE D'E.O.M.	6	
1	M	LISTE DES E.O.M.	9	
1	M	LIBELLE DU MOYEN	6	

1	M	NUMERO DU MOYEN	2	
1	M	NOMBRE D'E.O.M.	6	
1	M	LISTE DES E.O.M.	9	
1	M	LIBELLE DU MOYEN	6	

et            distance (M1, M2) = 1  
              distance (M1, M3) = 0  
              distance (M2, M3) = 2

la note définitive associée à SG1 sera :  $N1 + N2 + N3 - 3$   
c'est à dire :

$$\sum Ni - \sum \text{distances}$$

Le document associé aux distances a le format suivant (cf. page 35).

### 3.2.2. Définition du catalogue des e.o.m.

Par analogie avec le catalogue des fonctions, chaque e.o.m. est défini par son numéro d'ordre (en liaison avec le catalogue des moyens), son code et son libellé.

Le document correspondant est identique à celui utilisé pour définir les fonctions. On en extrait le fichier "catalogue des e.o.m."

Par contre, un second fichier est également créé, celui des incompatibilités entre e.o.m. Il est fréquent que deux e.o.m. n'appartenant pas au même moyen soient incompatibles (pour des raisons technologiques dans la plupart des cas). Le modèle en déduit alors qu'ils ne peuvent coexister au sein d'une même solution partielle. (Le fichier des incompatibilités est balayé lors de la phase "recherche des solutions partielles"). Par ailleurs, les incompatibilités sont prises en compte également lors de la recherche de la machine idéale (dans le cas d'adaptation externe). Lorsque l'union des e.o.m. relatifs aux solutions partielles est effectuée, il faut en effet vérifier que cette union demeure "compatible". (En d'autres termes, que la machine idéale n'est pas, a priori, impossible à construire).

Le document relatif aux incompatibilités a la forme suivante (voir page 36).

A ce stade, tous les éléments relatifs au SF et au SM sont complètement définis. Il reste alors à l'utilisateur à définir les paramètres caractérisant les éléments de solution (après que ceux-ci aient été déterminés par le modèle au cours de la phase de pré-évaluation).







### 3.3. Définition des paramètres associés aux éléments de solution

#### 3.3.1. Définition des notes

Chaque élément de solution est caractérisé par un ensemble de notes, affectées par l'utilisateur en fonction des deux notions qui ont généré l'élément de solution, c'est à dire d'un couple "fonction-e.o.m."

Il y a autant de notes que de critères d'évaluation. Le résultat de cette saisie est donc une matrice de n lignes et p colonnes, où n représente le nombre d'éléments de solution et p le nombre de critères d'évaluation. Nous reviendrons au cours du chapitre IV sur l'échelle de notation proposée à l'utilisateur puisque celle-ci est fonction de l'algorithme adopté pour le calcul de la note globale affectée à chaque élément de solution. Cette note globale constituera pour le modèle la donnée de base de l'évaluation proprement dite. (La notion de critère a alors totalement disparu).

#### 3.3.2. Définition des implications entre éléments de solution

La notion d'implication est assez délicate à saisir au premier abord, car elle fait intervenir le mécanisme de recherche des solutions partielles.

Pour l'expliquer, avant de la justifier, nous nous aiderons d'un exemple : Soit le processus P composé de 2 fonctions f1 et f2, soit le moyen M composé de 3 e.o.m. : m1, m2, m3.

(Dans notre hypothèse, le SF est réduit à un seul processus, le SM à un seul moyen). La matrice des éléments de solution (appelée espace général des éléments de solution) a la forme suivante :

	m1	m2	m3
f1	1		2
f2		3	4

(2 des 6 éléments de solution sont incompatibles).

Satisfaire le processus P revient à satisfaire les fonctions f1 et

f2, donc à trouver 2 e.o.m. satisfaisant successivement f1 et f2. Etant donné qu'il n'y a pas d'incompatibilités (puisque les e.o.m. d'un même moyen ne sont jamais incompatibles), tous les chemins sont possibles.

Il existe donc dans notre cas 4 solutions partielles (et globales par la même occasion) qui sont (1,3), (1,4), (2,3) et (2,4). En supposant que les poids des fonctions  $f_1$  et  $f_2$  sont tous deux égaux à 1, la note associée à chacune de ces solutions est la somme des notes associées à chaque élément de solution les composant. Cependant, il peut arriver qu'ayant choisi de satisfaire  $f_1$  par  $m_1$ , il soit tout à fait logique de satisfaire  $f_2$  par  $m_3$  (indépendamment des notes associées aux éléments de solution 3 et 4) et, a fortiori, même si l'élément de solution 3 a une meilleure note que le 4.

Pour prendre en considération un tel type de souhait, il a été décidé d'affecter à certains couples d'éléments de solution un "poids d'implication". Dans notre cas, la note de la solution (1,4) sera donc égale à la somme des notes associées aux éléments 1 et 4, augmentée du poids associé au couple (1,4).

De façon générale, la note d'une solution partielle sera égale à la somme des notes des éléments de solution la composant, augmentée de la somme des poids d'implication relatifs à tous les couples possibles d'éléments de solution de la solution partielle.

L'introduction d'une telle notion se justifie par le fait qu'elle permet d'obtenir des solutions partielles plus cohérentes, c'est à dire plus homogènes vis-à-vis de l'utilisation de certains e.o.m. Il est important de remarquer que cette notion intervient au niveau des éléments de solution et non pas à celui des e.o.m. Ce qui signifie que la fonction elle-même joue un rôle prépondérant. La notion d'implication a donc un sens tout à fait différent de celui rattaché à la notion d'incompatibilité. Un document particulier est prévu pour la saisie des implications entre éléments de solution (cf. page suivante). A partir de ce document est créé le fichier "implications".

A ce stade, toutes les données d'entrée du modèle ont été définies. Il n'en reste pas moins qu'avant de commencer le traitement proprement dit, le modèle a besoin d'un ensemble de données supplémentaires, à savoir l'ensemble des notes globales associées à chaque élément de solution. Ces notes globales sont, en réalité, des données "générées" (non fournies directement par l'utilisateur), c'est pourquoi nous ne parlerons de ces notes qu'au cours du chapitre IV.



Enfin, il faut noter qu'un document particulier est réservé pour la saisie des paramètres généraux de l'application. Ce document permet à l'utilisateur de spécifier le type d'adaptation souhaitée (interne ou externe) ainsi qu'un certain nombre d'informations nécessaires à l'allocation dynamique des différentes tables de données utilisées en cours de traitement.

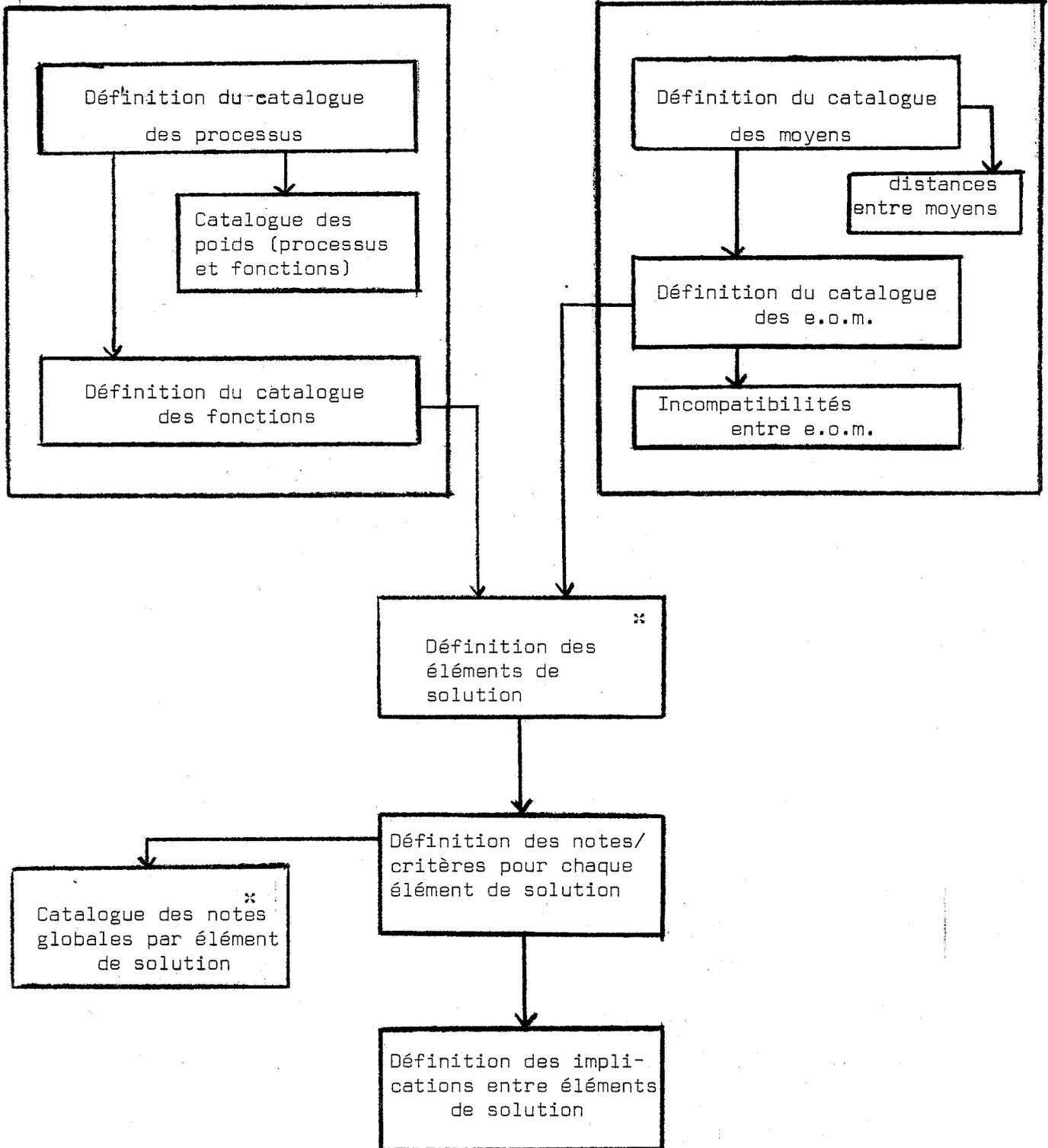
Nous reviendrons sur ce point particulier dans le paragraphe consacré au programme lui-même.

Pour conclure, nous nous proposons de récapituler sur un schéma l'ensemble des données nécessaires au bon fonctionnement du modèle, en mettant en évidence les liaisons logiques qui existent entre elles (voir page suivante).



Définition du SF

Définition du SM



\* Seuls les catalogues des éléments de solution et des notes globales sont établis automatiquement par le modèle.

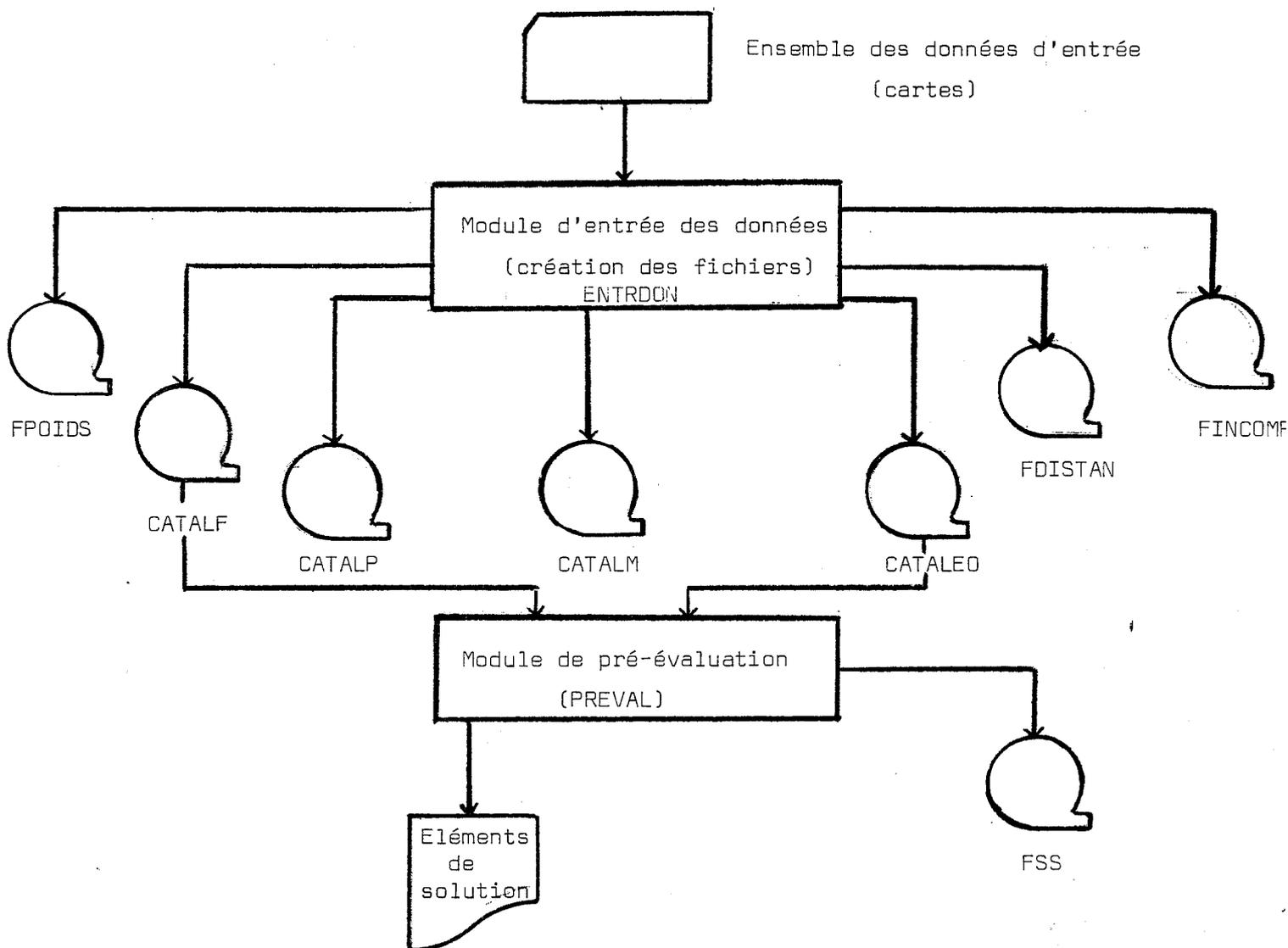


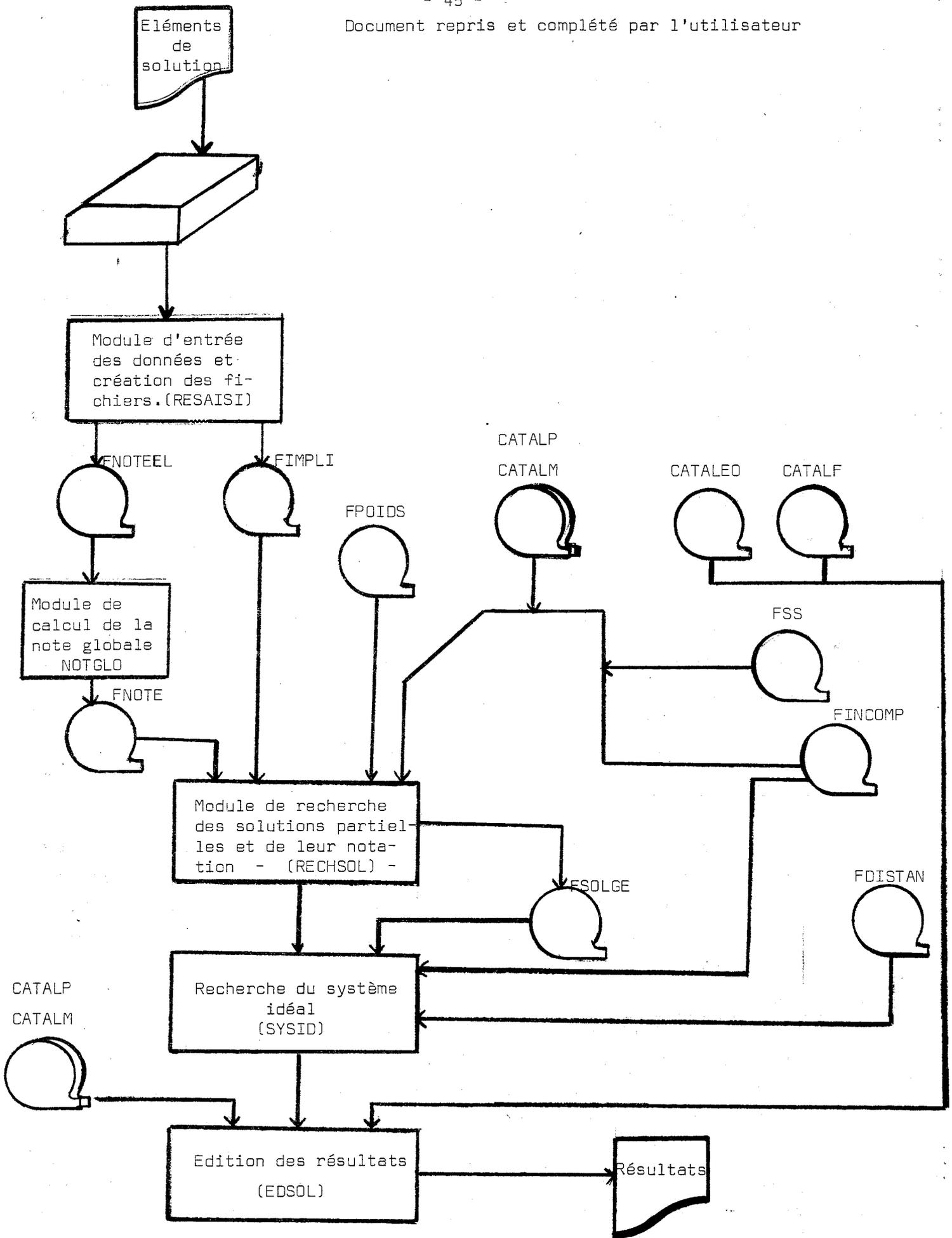
## CHAPITRE IV.. PRINCIPES DE FONCTIONNEMENT DU MODELE. DOSSIER TECHNIQUE

### 4.1. Présentation générale des différentes phases

L'ensemble des fichiers de données ayant été défini au chapitre précédent, il nous faut, maintenant, parler des unités de traitement et de leur organisation au sein de la chaîne générale de traitement. La caractéristique essentielle du modèle est, dans ce domaine, l'existence de deux programmes distincts liés par le fait que les documents de sortie du premier servent de documents d'entrée au second.

L'organigramme fonctionnel de l'ensemble de la chaîne se présente comme suit :





Remarque 1 - Le fichier FSS contient l'espace général des éléments de solution. Il renferme donc une partie des informations contenues dans le document de sortie du module PREVAL.

Remarque 2 - Le fichier FSOLGE contient toutes les solutions partielles et leurs notes (classées par processus ou par moyens, suivant le type d'adaptation demandée).

Remarque 3 - Un tel organigramme met nettement en évidence l'une des caractéristiques essentielles du modèle : le nombre important de fichiers. En réalité, aucun de ces fichiers n'est très volumineux et leur manipulation s'avère relativement aisée.

Remarque 4 - Le module SYSID se présente en fait comme deux modules distincts, l'un correspondant à l'adaptation externe (le plus important), l'autre à l'adaptation interne. A ce niveau du modèle la symétrie est rompue, et comme les problèmes posés par la recherche du système idéal diffèrent suivant le type d'adaptation, il faut avoir recours à deux modules distincts.

Enfin, parmi les sept modules proposés dans l'organigramme, quatre seulement présentent un intérêt certain. Nous laisserons donc de côté les deux modules d'entrée de données (ENTRDON et RESAISI) et le module d'édition (EDSOL).

#### 4.2. Module de pré-évaluation (PREVAL)

La pré-évaluation consiste à déterminer les éléments de solution compatibles. Le traitement est basé sur la comparaison des codes fonctions et des codes E.O.M. Cette comparaison est effectuée pour tous les couples "fonction-e.o.m." possibles, et lorsque pour un couple donné, les deux codes sont compatibles, il y a génération d'un élément de solution. Cet élément de solution possède les caractéristiques suivantes : un numéro d'ordre propre, égal au numéro de séquence de génération, les numéros d'ordre de la fonction et de l'e.o.m. desquels il est issu (ces trois informations génèrent le fichier FSS), et enfin une note calculée par le modèle, indépendante de tout critère (appelée note artificielle), fonction du degré de compatibilité des deux codes en présence. (Cette note varie de 0 à 3 - cf. paragraphe suivant). En effet, compatibilité entre deux codes ne signifie pas obligatoirement égalité. Dans le cas de l'égalité, la note sera maximum (3). Dans les autres cas, le modèle tiendra compte de la distance entre les deux codes. Celle-ci est calculée suivant des règles très précises exposées page suivante.

Ainsi, par exemple, il est évident qu'un e.o.m. de capacité supérieure à la fonction qui lui est opposée pourra satisfaire cette dernière ("qui peut le plus peut le moins"), à condition que les autres zones soient compatibles.

Il faut noter également qu'il existe une table T des couples d'opérateurs compatibles. Cette table est évidemment définie par l'utilisateur.

Enfin, il est clair que, pour déclarer l'élément de solution compatible, il faut qu'aucun des cinq contrôles (voir page suivante) ne soit défectueux. Dans ce cas, la note artificielle (NA) peut être calculée.

Remarque 1 - Lorsque le couple (f,m) à "contrôler" est tel que la fonction et l'e.o.m. possèdent chacun plusieurs zones 5, il faut que chaque zone 5 de la fonction soit satisfaite par l'une au moins des zones 5 de l'e.o.m. (on prend la meilleure s'il y en a plusieurs) pour que le couple soit déclaré compatible.

Remarque 2 - Au point de vue technique, la réalisation de ce module ne pose pas de grosses difficultés. Il a été choisi de rentrer en mémoire centrale l'un des deux fichiers - celui des e.o.m. - (puisque'il y a deux boucles imbriquées, on ne pouvait en effet défiler **n fois le même fichier** pour des raisons de performances). L'organigramme général de traitement est présenté ci-après.

Contrôles permettant d'établir la compatibilité d'un couple (f,m)

Règles de calcul de la note artificielle

CONTROLE 1

zone opérateur

$op(m) = op(f)$   
ou  
 $(op(m), op(f)) \in T$

*oui*

si  $op(m) = op(f)$   
note 1 = 3  
sinon = 2

T est un tableau contenant les couples d'opérateurs compatibles

CONTROLE 2

zone degré  
{d'automatisation de l'information}

$DE1(f) \leq DE1(m)$

*oui*

$S1 = DE1(m) - DE1(f)$   
 $n2 = 3 - S1$

CONTROLE 3

zone degré  
{de formalisation de l'information}

$CO(f) \leq CO(m)$   
sauf  $\{CO\}f = 4$   
 $\{CO\}m = 5$

*oui*

$d = CO(m) - CO(f)$   
note 3 =  
 $(3 - d) * \text{sign}(3 - d)$

$\text{sign}(x) = 1$  si  $x > 0$   
= 0 sinon

CONTROLE 4

zone unité

$UN(f) = UN(m)$

*oui*

note 4 = 3

CONTROLE 5

zone capacité

$CA(f) \leq CA(m)$

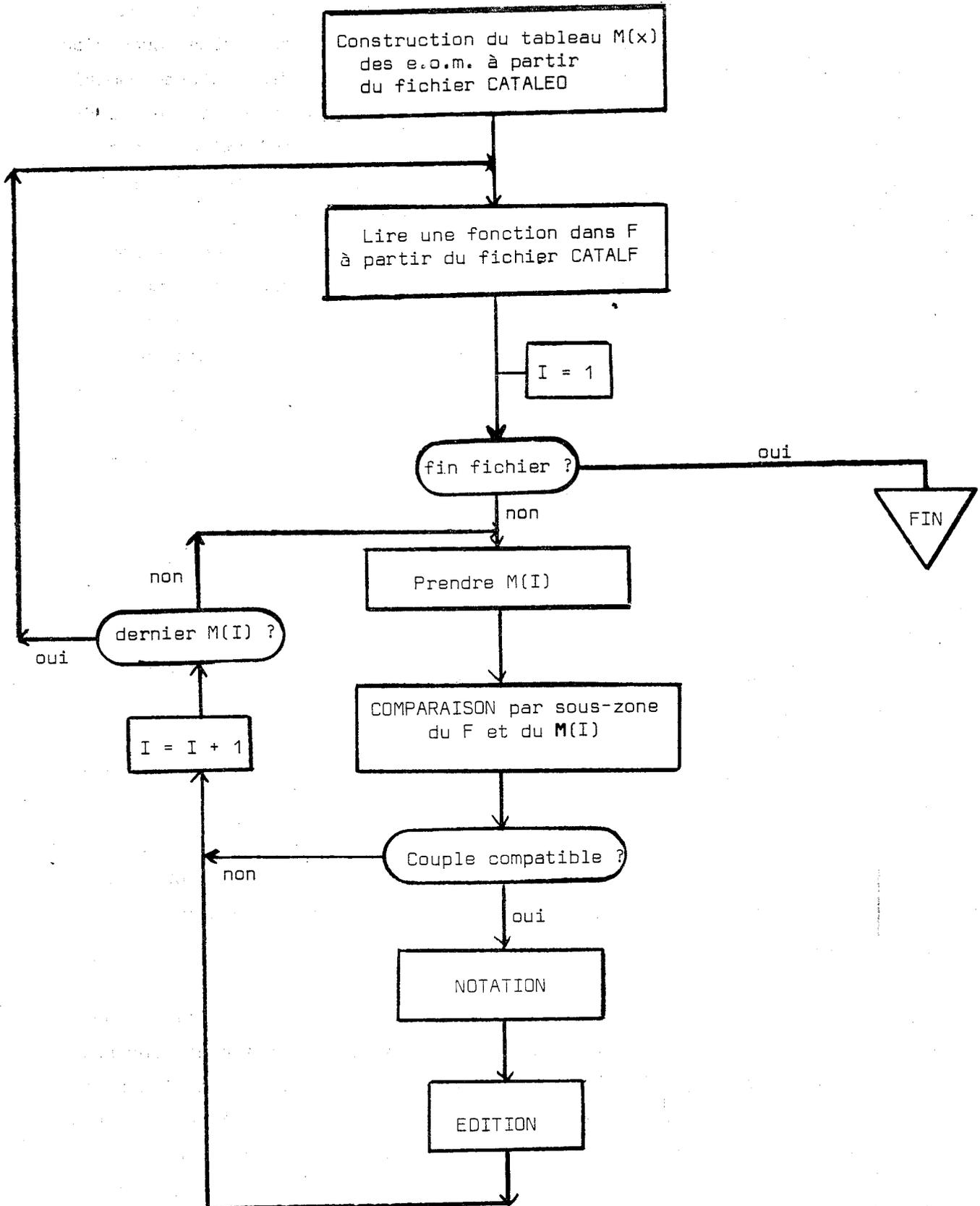
*oui*

$r = \frac{CA(m) - CA(f)}{CA(f)}$   
note 5 =  
 $(3 - r) * \text{sign}(3 - r)$

La note NA affectée au couple solution élémentaire ainsi trouvé est donnée par :

$$NA = \frac{1}{5} \sum_{i=1}^5 \text{note } i$$

Organigramme de traitement de la pré-évaluation



#### 4.3. Module de notation (NOTGLO)

L'obtention de notes globales à partir de notes élémentaires liées à chaque critère d'évaluation est un des points les plus délicats du modèle. De plus en plus, en effet, on prend conscience de l'irréalisme des méthodes classiques d'optimisation de choix. Ces méthodes évaluent les objets selon des critères homogènes, qui peuvent tous se ramener à un critère unique de nature économique (coût ou utilité), et le problème devient alors un problème de maximisation ou minimisation de fonctions. Ainsi, au fur et à mesure du perfectionnement des modèles, se révèle une disparité choquante entre la précision des résultats fournis et l'imprécision existant au niveau de la prise en compte des points de vue non quantifiables. Il en découle une perte d'information extrêmement dangereuse (puisqu'on ne peut la localiser) et un "manque de confiance" dans le modèle. La majeure partie de l'information fournie par les calculs d'optimisation est détruite, en effet, par des ajustements subjectifs (qui sont précisément sensés tenir compte des points de vue non quantifiables) et qui interviennent soit avant le calcul, au niveau de l'entrée des paramètres, soit, plus fréquemment, à la sortie des résultats, au cours de leur interprétation.

Afin de remédier à ce danger, trois méthodes ont été envisagées et seront exposées brièvement :

##### 4.3.1. Méthode à pondération globale des échelles et à réduction des noyaux

Un exemple d'application de cette méthode nous est fourni par le package ELECTRE de la SEMA.

Le but de la méthode est de ramener à un seul point de vue (ou critère) des critères multiples et hétérogènes, afin de traiter l'ensemble comme un problème unidimensionnel.

La méthode est basée sur la construction d'un graphe ( $G_i$ ) associé à chaque critère, mettant en évidence un ordre partiel des objets à évaluer par rapport à ce critère. (Cela suppose qu'une échelle de valeur a été construite pour chaque critère). Il s'agit alors de construire un graphe  $G$  de synthèse tel que, si pour tout graphe partiel  $G_i$  il existe un arc  $A \rightarrow B$ ,  $A \rightarrow B$  appartient à  $G$  (dans lequel  $A$  et  $B$  sont deux objets quelconques, la relation  $A \rightarrow B$  signifiant que  $A$  est préféré à  $B$ ).

Le graphe G permet d'éliminer tous les objets classés de la même façon quel que soit le critère. Il reste alors une partie irréductible, appelée noyau, qui contient les objets évalués différemment selon les critères. Le problème est alors de réduire ce noyau. Sans trop entrer dans le détail de la méthode, disons que la réduction est basée sur le calcul d'indicateurs de concordance (notés C) et de discordance (notés D) associés à chaque couple d'objets du noyau.

L'indicateur de concordance de deux objets A et B est le rapport entre la somme des poids des critères pour lesquels la relation  $A \rightarrow B$  est vraie et la somme des poids de tous les critères. Il varie entre 0 et 1, et lorsqu'il est égal à 1, l'objet B peut être retiré du noyau (réciproquement, l'objet A disparaît lorsque  $C = 0$ ). L'indicateur de discordance d de deux objets A et B est le rapport entre le plus grand écart de notation entre l'objet B et l'objet A (sur l'ensemble des échelles) et l'étendue de cette échelle. On peut modifier cet indicateur en changeant les pondérations entre échelles.

On peut prendre également le rapport entre la somme des écarts pondérés (pour tout critère dans lequel B est supérieur à A) et la somme des étendues des échelles correspondantes.

Exemple :	notes/20		Critère	Poids	Etendue de l'échelle
	A	B			
	12	9	1	5	100
	17	6	2	5	100
	6	12	3	1	20
	9	11	4	3	60
	15	13	5	2	40

$$C(AB) = \frac{5 + 5 + 2}{5 + 5 + 1 + 3 + 2} = \frac{12}{16} \quad (\text{il y a concordance pour les critères 1, 2, 5})$$

$$D(AB) = \frac{(6 \times 1) + (2 \times 3)}{20 + 60} = \frac{12}{80} \quad (\text{il y a discordance pour les critères 3 et 4})$$

Il est clair que plus d est élevé, plus il sera difficile d'accepter  $A \rightarrow B$ . Il est nécessaire alors de définir deux seuils p et q compris entre 0 et 1 tels que si :

$$C(AB) > p \quad \text{et} \quad D(AB) < q$$

$A \rightarrow B$  sera accepté.

Lorsque cette double condition n'est pas satisfaite, il faut manipuler certains paramètres pour qu'elle le soit, c'est à dire modifier  $p$  et/ou  $q$ , ou modifier la pondération. Ceci revient à introduire l'arbitraire à l'origine du système, au lieu de le faire intervenir au niveau de la sortie.

Au vrai, cette méthode possède un inconvénient majeur : obliger l'utilisateur à attribuer une note à chaque objet et pour chaque critère (faute de quoi, on ne peut plus construire l'indicateur de discordance). Or, en pratique, cette contrainte n'est absolument pas réaliste, car il est extrêmement difficile d'évaluer des écarts de notes entre critères hétérogènes.

#### 4.3.2. Méthodes combinatoires à pondération variable

(Du type C.P.E. - Centre de prévisions économiques). Ces méthodes classent des objets disparates en établissant des relations d'utilité entre eux. Ce sont ces relations qui seront le support des choix. Pour cela, on affecte une note à chaque relation (note incorporant un ensemble de notes relatives aux différents critères) et l'on choisit les relations qui ont obtenu les meilleures notes (correspondant à l'utilité maximale).

L'intérêt de la méthode est de mettre l'accent sur le qualitatif (détermination des relations) avant d'insister sur le quantitatif (notes).

#### Remarque :

*Le même problème que précédemment se pose lorsqu'il s'agit d'affecter une note unique à chaque relation (puisque les critères ne sont pas forcément homogènes).*

*La difficulté a été résolue en acceptant l'hétérogénéité et en incorporant "consciemment" la subjectivité sous forme d'échelles qualitatives : très favorables, favorables, peu favorables, pas favorables (cotées 0, 1, 2, 3). Il en découle alors une agrégation (sommation) possible, d'où une extrême simplicité des calculs.*

*L'intérêt de la méthode est de pouvoir choisir des échelles plus ou moins nuancées, tout en conservant les possibilités d'agrégation. (Une amélioration possible consisterait même à utiliser des échelles de raison variable (1, 2, 6, 20 par exemple).*

Cependant, bien que relativement simples, ces méthodes s'appliquent beaucoup plus à des schémas à long terme qu'à des choix opérationnels à court et moyen terme. Elles permettent d'établir un projet plus qu'elles ne fournissent des éléments d'aide à la décision élémentaire.

Dans notre cas particulier, très concret, une telle méthode ne pouvait donc pas s'appliquer. Il a été retenu en définitive la "méthode des commutations" exposée dans le projet LION II par Monsieur LUSSATO.

#### 4.3.3. La méthode des commutations

Cette méthode est basée sur l'idée que tout choix repose sur une "balance", un équilibre entre deux ou plusieurs critères, et sur la constatation que de nombreux choix s'effectuent au niveau de l'acte élémentaire (nécessité d'une situation concrète). C'est le psychologue américain THURSTONE qui en fut le précurseur en 1926. Elle a été appliquée au modèle en y apportant quelques simplifications.

Les échelles utilisées pour chaque critère ont été divisées en trois ordres :

- Premier ordre : (code 1). Il correspond à l'exigence de base, à la satisfaction minimale du critère. Tout objet qui n'est pas d'ordre 1 est exclu (car trop défavorable).
- Deuxième ordre : C'est l'ordre des "attributs favorables". Tout objet d'ordre 2 est satisfaisant. Il répond en particulier à l'essentiel (il est donc également d'ordre 1).
- Troisième ordre : Il correspond aux objets à propriété particulière (de type superflu). Tout objet d'ordre 3 est aussi d'ordre 2, donc d'ordre 1. La classe des objets d'ordre 3 correspond à la satisfaction maximale (mais non nécessaire) du critère. On peut les assimiler à des gadgets.

La notation s'effectue par addition pure et simple des ordres, en tenant compte cependant du nombre de gadgets. Ainsi, par exemple, deux objets A et B notés respectivement (par rapport à 5 critères) :

1	2	2	3	2	(A)
2	2	2	2	2	(B)

n'ont pas tout à fait la même note globale. (Celle associée à B est légèrement supérieure).

Remarque : Les notes globales sont toujours ramenées dans l'échelle (0 - 3).

L'intérêt de la méthode réside dans le fait que les ordres ont la même valeur quel que soit le critère et quel que soit l'objet (d'où une extrême simplicité), et que l'on peut distinguer l'essentiel du superflu (ce qui n'était pas le cas de la méthode précédente).

En outre, cette méthode est proche de la réalité, puisqu'elle recherche, par définition, l'équilibre entre plusieurs facteurs. Concrètement, elle a été appliquée dans le modèle en introduisant une pondération des critères, malgré tous les inconvénients que cette pondération entraîne (voir remarque précédente). L'utilisateur peut ainsi cerner plus facilement les critères qu'il estime essentiels.

*Remarque : La conséquence d'une telle notation est que la note maximale d'une solution partielle de longueur n (possédant n éléments de solution) est égale à :*

$$3 \sum_{i=1}^{i=n} p_i$$

*où  $p_i$  représente le poids de la  $i$ ème fonction (3 n dans le cas où les fonctions ne sont pas pondérées). Il faut donc, pour permettre à l'utilisateur de comparer des solutions partielles de longueur différente, diviser chaque note par la note maximale (qui correspond à la solution idéale).*

#### 4.4. Module de recherche et de notation des solutions partielles (RECHSOL)

Suivant le type d'adaptation demandé, le module fait éclater la matrice "espace général des solutions" en autant de sous-matrices qu'il y a de processus (adaptation externe) ou de moyens (adaptation interne). Chacune de ces sous-matrices est alors traitée indépendamment pour obtenir l'ensemble des solutions partielles (et des notes associées) qui lui correspondent.

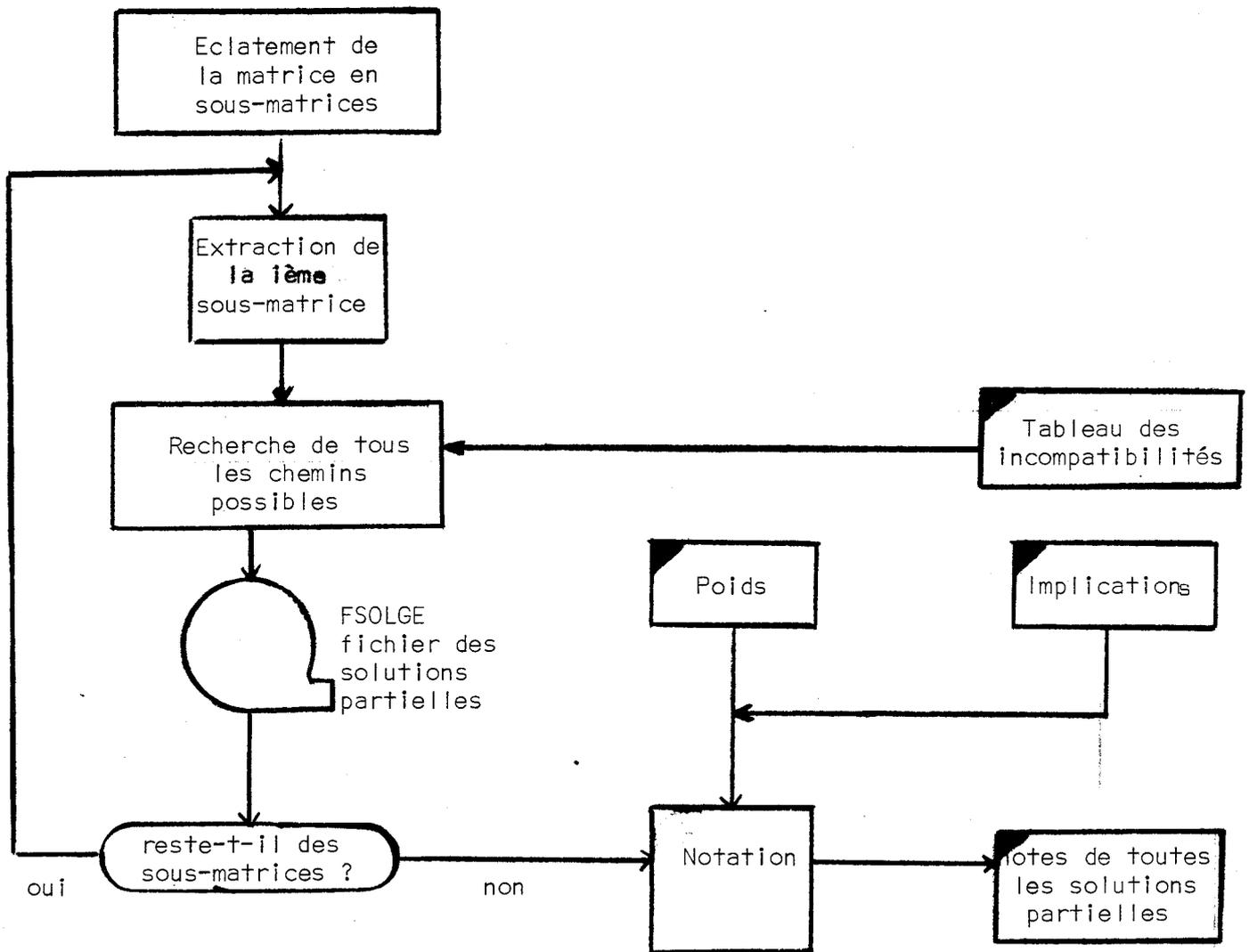
Pour cela, il suffit, dans une première phase, de trouver tous les chemins possibles (de la sous-matrice) permettant d'aller de la première à la dernière ligne (en tenant compte des incompatibilités), si l'on est en adaptation externe. Puis, il faut reprendre chacun de ces chemins et calculer la note qui lui est associée par la formule :

$$\text{Note solution partielle (de longueur n)} : \frac{\sum_{i=1}^{i=n} n_i \times p_i + k \sum \text{implications}}{3 \sum_{i=1}^{i=n} p_i}$$

où  $n_i$  représente la note du  $i$ ème élément de solution  
 $p_i$  " le poids de la  $i$ ème fonction  
 $k$  " un coefficient déterminé par l'utilisateur

*Remarque* : En adaptation interne, les poids associés aux fonctions (ainsi qu'aux processus) n'ont aucune signification (puisque ce sont les moyens qu'on évalue).

L'organigramme général de ce module se présente comme suit :



#### 4.5. Module de recherche du système idéal (SYSID)

Comme nous l'avons vu précédemment, il existe en réalité deux modules ayant une partie commune. Nous traiterons séparément les deux cas.

##### 4.5.1. Adaptation externe

C'est le cas le plus complexe et le plus fréquent. Tous les paramètres définis au chapitre III entrent alors en jeu. Le traitement consiste à reprendre le fichier FSOLGE des solutions partielles exprimées en termes d'éléments de solution, et à convertir chaque solution partielle en terme d'e.o.m. (puisque c'est un "moyen idéal" que l'on recherche). Il faut alors déterminer la meilleure des solutions générales (toujours exprimée en e.o.m.) et vérifier qu'elle est compatible, c'est à dire que l'ensemble des e.o.m. qui la composent le sont. Dans l'affirmative, la machine idéale est trouvée. Il reste à lui associer sa note globale, qui est égale à la différence entre la somme des notes de solutions partielles et la somme des distances entre moyens. Dans la négative, il faut déterminer la solution générale de rang immédiatement inférieur - on est alors ramené au cas précédent -.

*Remarque* : La notion de distance entre moyens n'intervient pas lors du calcul de la machine idéale, puisque celle-ci est composée des meilleurs e.o.m. possibles - indépendamment de toutes distances entre les moyens M de référence -.

La deuxième phase du traitement consiste à déterminer la machine idéale existante. Il n'y a pas de problème à ce niveau, puisque chaque e.o.m. n'appartient qu'à un seul moyen. Il suffit d'exprimer la machine idéale en terme de moyens pour obtenir la machine idéale existante.

*Remarque importante* : En réalité, contrairement à la théorie, il arrive fréquemment que certains e.o.m. soient communs à plusieurs moyens (ou tout au moins, c'est l'utilisateur qui en a décidé ainsi, l'intérêt étant d'augmenter le nombre lorsque certains d'entre eux sont très "ressemblants"). La détermination de la machine idéale existante pose alors certains problèmes que nous avons déjà abordés au cours du chapitre II.

Le problème effectif est le suivant :

Soit  $W$  l'ensemble des e.o.m. constituant la machine idéale. Soit  $M_1, M_2, M_3 \dots MN$ , les moyens existants. Soit  $I$  l'ensemble d'indices  $(1, N)$ , il faut trouver un ensemble d'indices  $J$  tel que :

$$J \subseteq I \quad (1)$$

$$W \subseteq \bigcup_{k \in J} M_k \quad (2)$$

$$\text{card.} \left( \bigcup_{k \in J} M_k - W \right) \text{ soit minimum} \quad (3)$$

Remarque : Il existe toujours une solution puisque  $W \subseteq \bigcup_{k \in J} M_k$

La contrainte (3) permet, comme nous l'avons vu précédemment, de prendre en compte un minimum d'e.o.m. superflus qui sont des "gadgets" pour l'application à résoudre.

En réalité, la contrainte (3) n'est pas suffisante pour résoudre totalement le problème, car il peut exister plusieurs familles  $M_J$  satisfaisant les trois contraintes, c'est à dire recouvrant  $W$ , et dans ce cas, il semble intéressant de déterminer le recouvrement minimum de  $W$  (déterminer la - ou les - familles  $M_J$  qui contiennent le moins d'éléments), ce qui nous donne une nouvelle contrainte (4). Au niveau du modèle, la recherche du recouvrement minimum est cependant inutile, car pour l'utilisateur il est très intéressant de connaître tous les recouvrements possibles de  $W$ , et pas uniquement le meilleur. Par ailleurs, les deux dernières contraintes (3) et (4) ne sont pas toujours suffisantes pour déterminer de façon certaine la meilleure des machines existantes, d'autres critères non connus du modèle (coûts, par exemple) entrant en jeu.

Il a donc été décidé de calculer toutes les combinaisons possibles de moyens existants satisfaisant la MI (c'est à dire tous les recouvrements possibles) en les classant au niveau de l'édition, de façon décroissante par rapport aux contraintes (3) et (4). Pour cela, le programme calcule toutes les combinaisons de moyens possibles en stockant le cardinal de l'ensemble résiduel et le nombre d'éléments de la combinaison lorsque la condition (2) est vérifiée (c'est à dire lorsque la combinaison est un recouvrement).

Il suffit alors de trier les solutions obtenues (en majeur sur le nombre d'e.o.m. superflus, en mineur sur le nombre d'éléments constituant la solution). La première de ces solutions constitue la MIE ; elle n'est pas obligatoirement unique.

A titre indicatif, il nous semble intéressant cependant de présenter, à ce niveau, la méthode généralement utilisée pour résoudre des problèmes de recouvrement minimum. Cette méthode est basée sur le calcul de la duale d'une fonction booléenne, c'est à dire le développement d'un produit de sommes, et, la procédure de dualisation étant une opération fondamentale en algèbre de Boole, il semble nécessaire de disposer d'algorithmes extrêmement performants. Nous nous contenterons de poser ici le principe de la dualisation et de présenter l'algorithme de base permettant de le résoudre. Monsieur BENZAKEN propose deux améliorations possibles de cet algorithme de référence (Cf. Bibliographie).

× Principe

Soit  $f$  une fonction booléenne.  $f = m_1 + m_2 + \dots + m_q$   
 et  $f^*$  sa duale.

Si  $g_i$  est la duale de  $m_1 + m_2 + \dots + m_i$   
 $g_{i+1} = g_i \cdot (m_{i+1}^*)$  est la duale de  $m_1 + m_2 + \dots + m_{i+1}$   
 $m_{i+1}^*$  étant la duale de  $m_{i+1}$   
 (soit  $m_{i+1} = a_1 \cdot a_2 \dots a_p$  alors  $m_{i+1}^* = a_1 + a_2 + \dots + a_p$ )

La méthode consiste à calculer  $g_q = f^*$  par itérations successives en tenant compte de la remarque précédente.

Le calcul de la duale d'une fonction écrite sous forme d'une somme de monômes consiste donc à répéter le calcul plus restreint suivant :

Soit  $g$  une fonction booléenne de  $k$  monômes irréductibles,  
 soit  $m = a_1 \cdot a_2 \dots a_p$  un monôme de  $p$  lettres, déterminer le polynôme irréductible de la fonction

$$h = g \cdot (m^*) = g(a_1 + a_2 + \dots + a_p)$$

avec  $g = \mu_1 + \mu_2 + \dots + \mu_k$

Pour effectuer ce calcul, la méthode triviale consiste évidemment à développer et former  $kp$  produits, après quoi, il faut éliminer les multiples en comparant les monômes deux à deux, ce qui donne un maximum de  $C_{kp}^2$  comparaisons (si  $m$  et  $g$  sont disjoints). Ainsi, par exemple, si  $k = 10$  et  $p = 5$ , on peut avoir  $C_{50}^2 = 1\ 225$  comparaisons inutiles dans le cas disjoint, puisqu'aucune élimination n'intervient.

\*: Algorithme

L'algorithme de référence correspondant est le suivant :

Les  $k$  monômes de  $g$  sont rangés dans  $k$  mémoires consécutives, le monôme  $m = a_1 a_2 \dots a_p$  dans une mémoire.

(1) Former le tableau  $m^*$  ayant  $p$  mémoires consécutives contenant  $a_1, a_2, \dots, a_p$ .

(2) Former  $\mu_1 a_1$ . Si le résultat est nul, continuer avec  $\mu_1 a_2$  etc... puis éventuellement  $\mu_2 a_1, \dots$  etc... Ranger le premier monôme non nul ainsi formé dans le tableau  $h$ . S'il n'y en a pas, le processus s'arrête et  $h = 0$ .

(3) Supposons avoir formé dans  $h$ , tous les monômes  $\mu_1 a_\ell$  irréductibles entre eux et non nuls avec :  $i < k, \ell < p$ .

On formera alors  $\rho = \mu_i a_{\ell+1}$  (si  $\ell < p$ ) ou  $\mu_{i+1} a_1$  (si  $\ell = p$ ).

Si ce monôme  $\rho$  est nul, on retourne à (3) avec  $\ell$  (ou  $i$ ) augmenté.

Si  $\rho$  est non nul, on le compare aux monômes de  $h$ .

S'il advient que  $\rho$  est un multiple d'un monôme  $h$ , on l'élimine et on repasse à la phase (3).

Si  $\rho$  est un diviseur d'un monôme  $\mu$  de  $h$ ,  $\mu$  est éliminé de  $h$ , et on continue ainsi à supprimer de  $h$  tous les multiples de  $\rho$ ;  $\rho$  est ensuite rangé dans  $h$ . On retourne à la phase (3).

(4) Le procédé s'arrête lorsque  $i = k, \ell = p$ .

\*: Evaluation du coût théorique de cet algorithme

(le critère étant le nombre de comparaisons en moyenne).

Sur les  $kp$  produits formés, nous désignerons par :

$n$  le nombre de produits non nuls,

$r$  le nombre de monômes du résultat. . . .

Le nombre de monômes éliminés est donc de  $(n-k)$ .

Bien entendu, il est difficile d'évaluer rigoureusement le nombre total de comparaisons. Si l'on fait une évaluation simple, on peut noter que les  $k$  monômes du résultat ont été comparés deux à deux, ce qui fait  $k(k-1)/2$  comparaisons, et que chaque monôme éliminé a été comparé en moyenne  $k/2$  fois.

Le coût approximatif est donc  $k(k-1)/2 + (n-k)k/2 = (n-1)k/2$

Exemple :

$$g = ab + bd + bf + ace + cd + ad$$

$$m = aef$$

$$k = 6$$

$$p = 3$$

$$n = 18$$

On obtient :

$$h = g(m^*) = ab + bf + ace + ad + bde + cde + cdf$$

$$k = 7$$

Le coût théorique est donc  $17 \times 7/2 \approx 60$  comparaisons.

(On trouve en fait 61 comparaisons effectives en appliquant l'algorithme).

:: Application de la méthode à la recherche des recouvrements minima

Exemple : On désire rechercher les recouvrements minima de l'ensemble  $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$  par les parties suivantes :

$$a = \{0, 3, 5, 7, 8\}$$

$$b = \{0, 1, 2, 3, 4, 6\}$$

$$c = \{1, 2, 5, 6, 9\}$$

$$d = \{0, 2, 4, 6, 7\}$$

$$e = \{1, 7, 8, 9\}$$

$$f = \{2, 5, 7, 8, 9\}$$

$$g = \{1, 3, 4, 5, 8\}$$

On établit la fonction de recouvrement  $u$  dont les monômes sont associés aux parties recouvrant successivement chaque sommet, soit :

$$u = abd + bceg + cdf + abg + bdg + acfg + bcd + adef + aefg + cef$$

(L'élément 3 est "recouvert" par  $a$ ,  $b$  et  $g$ , ce qui donne le monôme  $abg$ ).

Le problème consiste alors à calculer les monômes de  $u^*$  de degré minimum.

$$u^* = (a + b + d) (b + c + e + g) (c + d + f) (a + b + g) (b + d + g) \\ (a + c + f + g) (b + c + d) (a + d + e + f) (a + e + f + g) (c + e + f)$$

Le problème est d'ailleurs encore simplifié si l'on se fixe par avance le degré maximum souhaité, ainsi, si l'on cherche des recouvrements de degré au plus égal à 2, il est extrêmement facile de procéder par étapes successives.

Exemple :

Les recouvrements de degré 2, recouvrant 0, 1, 2 sont :

$$ac, cd, de, dg, bc, bd, bf$$

en rajoutant le sommet 3 recouvert par  $abg$  il ne restera plus, comme monômes de degré 2 recouvrant 0, 1, 2, 3, que ceux qui ont au moins une lettre commune avec le monôme  $abg$ , soit :

$$ac, dg, bc, bd, bf$$

et ainsi de suite. Dans ce cas, il n'existe qu'un seul recouvrement de degré 2, le monôme  $bf$ .

#### 4.5.2. Adaptation interne

L'interprétation des résultats fournis par une évaluation de type adaptation interne reste très délicate. L'intérêt essentiel de l'adaptation interne est, en effet, de déterminer pour un moyen (ou un ensemble de moyens donnés) quelles sont les fonctions les mieux satisfaites par les e.o.m. composant ce moyen, et, partant, essayer de découvrir si ces fonctions représentent un ensemble cohérent de processus existants.

Pour cela, le modèle convertit les solutions partielles en termes de fonctions. La meilleure des solutions générales fournit alors directement l'ensemble des "fonctions idéales" (processus idéal) - puisqu'il n'y a pas d'incompatibilité - en réalisant simplement l'union des fonctions associées à chaque solution partielle.

Le processus idéal est calculé comme précédemment. On retrouve en effet le problème dual du précédent.

Si  $V$  est le processus idéal, il faut trouver un ensemble d'indices  $J$  tel que :

$$J \subseteq I \quad \text{avec } I = (1, n) \quad (1)$$

$$\bigcup_{k \in J} P_k \subseteq V \quad (2)$$

$$\text{card. } (V - \bigcup_k P_k) \text{ soit minimal} \quad (3)$$

L'édition des résultats ne pose aucun problème particulier. Elle se fait à deux niveaux :

- édition de la machine idéale (ou du processus idéal)
- édition de la machine idéale existante et des autres combinaisons possibles (même chose pour les processus).

Dans les deux cas, le traitement se borne à une recherche de libellés dans les fichiers correspondants, recherche qui est effectuée en accès direct.

Une attention particulière a été accordée à la présentation des résultats, car leur interprétation peut s'avérer délicate.

#### 4.6. Dossier technique

Le but de ce paragraphe n'est pas de présenter les dossiers d'analyse organique de l'ensemble des programmes du modèle, mais d'en justifier les options de base, prises au niveau de la programmation.

Nous parlerons donc successivement du choix du langage (PL/1), de l'utilisation de l'overlay et des conséquences qui en découlent, des problèmes rencontrés lors de la mise au point, et des performances du modèle.

##### 4.6.1. Choix du langage de programmation

PL/1 a été choisi comme langage unique, et ce pour plusieurs raisons. Le modèle se présente comme un outil de gestion, il fallait donc un langage de gestion pour prendre en compte avec suffisamment de souplesse tous les problèmes relatifs au traitement des entrées/sorties, à la gestion des fichiers et aux éditions multiples et relativement complexes qui en conditionnent directement l'efficacité.

Cependant, certains algorithmes (recherche des solutions dans la matrice, recherche de la MIE) pouvaient difficilement être programmés en COBOL. Nous proposons d'ailleurs au paragraphe 6.3, à titre d'exemple, le programme correspondant à la recherche de la MIE.

En conséquence, le choix du langage de programmation était à faire entre COBOL, FORTRAN et PL/1. PL/1 a été retenu en définitive, pour les facilités qu'il offre au niveau des déclarations de variables. En effet, l'un des points essentiels du modèle était de pouvoir limiter l'emplacement mémoire des différents tableaux de données correspondant à une application particulière. Ceci ne pouvait être réalisé que par le biais de déclarations dynamiques (impossibles en COBOL). Par ailleurs, il semble que l'assemblage de modules FORTRAN et COBOL, malgré leurs compatibilités, aurait présenté quelques difficultés pratiques qu'il nous semblait bon d'éviter.

*Remarque : L'allocation dynamique des tableaux ne peut s'effectuer que grâce à un document de saisie particulier contenant les paramètres généraux de l'application et rempli, soit par l'utilisateur, soit par la personne responsable de l'exploitation du modèle.*

Ce document (D1), présenté précédemment, contient les informations suivantes :

ADAPT	permet à l'utilisateur de préciser quel genre d'adaptation il désire. Cette chaîne de caractères prend deux valeurs : 'EXTERNE' pour une adaptation des moyens par rapport aux fins. 'INTERNE' pour une adaptation des fins par rapport aux moyens.
NPROC	donne le nombre de processus du système des fins.
NBF	le nombre de fonctions différentes qui composent ces processus.
MAXFCT	le nombre maximum de fonctions pouvant être associées à un processus.
NMOY	nombre de moyens.
NBEOM	nombre d'e.o.m.
MAXEOM	nombre maximum d'e.o.m. associés à un moyen.
LINCOMP	nombre de cartes décrivant les incompatibilités.
NINCOMP	nombre maximum d'incompatibilités pouvant exister pour un e.o.m.
LDISTANC	nombre maximum de cartes décrivant les distances.
NDISTANC	nombre maximum de distances associées à un moyen.
NBCRIT	nombre de critères d'évaluation.
NBES	nombre d'éléments de solution.
NBSG	nombre de solutions globales à prendre en considération.
NBSP	nombre de solutions partielles à prendre en considération.
ICOEF } ICOEFF }	coefficients utiles à la notation des solutions partielles. ... ..

#### 4.6.2. Utilisation de l'overlay

La principale contrainte des packages de gestion étant de limiter au maximum l'encombrement mémoire de l'ensemble des programmes et des données, il nous fallait trouver une méthode pour que le modèle puisse fonctionner sur des ordinateurs de taille relativement réduite (128 K octets) sans trop perturber les performances de l'ensemble des programmes.

Parmi les méthodes possibles, l'emploi des facilités offertes par l'overlay pouvait être choisi, à condition que la taille des programmes ne soit pas trop réduite par rapport à la taille des données (ce qui est souvent le cas des packages de gestion), et que l'arbre overlay proposé - tout en respectant une certaine logique - apporte effectivement un gain au niveau de l'encombrement mémoire.

En conséquence, la justification de l'overlay est basée sur les constatations suivantes :

- Pour une application moyenne, le nombre des fonctions est environ de 100, le nombre d'e.o.m. de 100, et le nombre d'éléments de solution de 1 000. La taille des tableaux de données correspondant à l'application est dans ce cas de 50 K octets. (Voir calcul détaillé page suivante).
- La taille de l'ensemble des programmes du second modèle est d'environ 90 K, ce qui implique qu'avec une procédure unique, le modèle ne peut s'exécuter dans une partition inférieure à 150 K.
- Par ailleurs, certains tableaux ne sont pas alloués en permanence, et il était possible - étant donné la logique du programme - de construire un arbre overlay qui tienne compte de cette particularité. Ainsi, par exemple, la matrice "espace général des solutions" (20 K octets avec les hypothèses précédentes) est utilisée uniquement par une procédure (RECHSOL) que l'on a pu isoler sur une branche relativement "courte" de l'arbre, le résultat étant que les 20 K octets de cette matrice n'interviennent plus sur l'encombrement global du modèle, puisque RECHSOL n'est pas située sur la branche maximale de l'arbre. (Voir pages suivantes).

Remarque : Calcul de l'encombrement mémoire des données du modèle (pour l'application traitée en annexe).

NOM DE LA DONNEE	BORNES	VALEURS EFFECTIVES	TAILLE
MOYEN	(NMOY, MAXEOM + 1)	(5,7)	70
INVMOY	(NBEOM, NMOY + 1)	(10,6)	120
PROCESS	(NPROC, MAXFCT + 1)	(5,9)	90
INVPROC	(NBF, NPROC + 1)	(10,6)	120
POIPROC	(NBADAPT)	(5)	10
VPOIDS	(NPROC, MAXFCT)	(5,8)	80
DISTANC	(NMOY, 2 × NDISTAN + 1)	(5,7)	70
IMPLI	(NBES, 2 × NIMPLI + 1)	(21,1)	42
INCOMP	(NBEOM, NINCOMP + 1)	(10,5)	100
SS	(NBF, NBEOM)	(10,10)	200
INVSS	(NBES, 2)	(21,2)	84
EQUCLEO	(NBEOM)	(10)	80
EQUCLEF	(NBF)	(10)	10
NBSOL	(NBADAPT)	(5)	10
LONG	(NBADAPT)	(5)	10
NOTEES	(NBES)	(21)	63
NOTE	ε(NBSOL)	(~ 1500)	~ 4500
NOTEPAR	ε(NBSOL)	(~ 1500)	~ 4500
MI	(NBEOM NBF)	(10)	20
NCS	(NBES)	(21)	42
INDNP	(NBADAPT)	(5)	10
SOLUT	(100, NBADAPT)	(100,5)	1000
NOYAUC	(0 : NBADAPT)	(0 : 5)	12
<i>TOTAL</i> .....			11.360

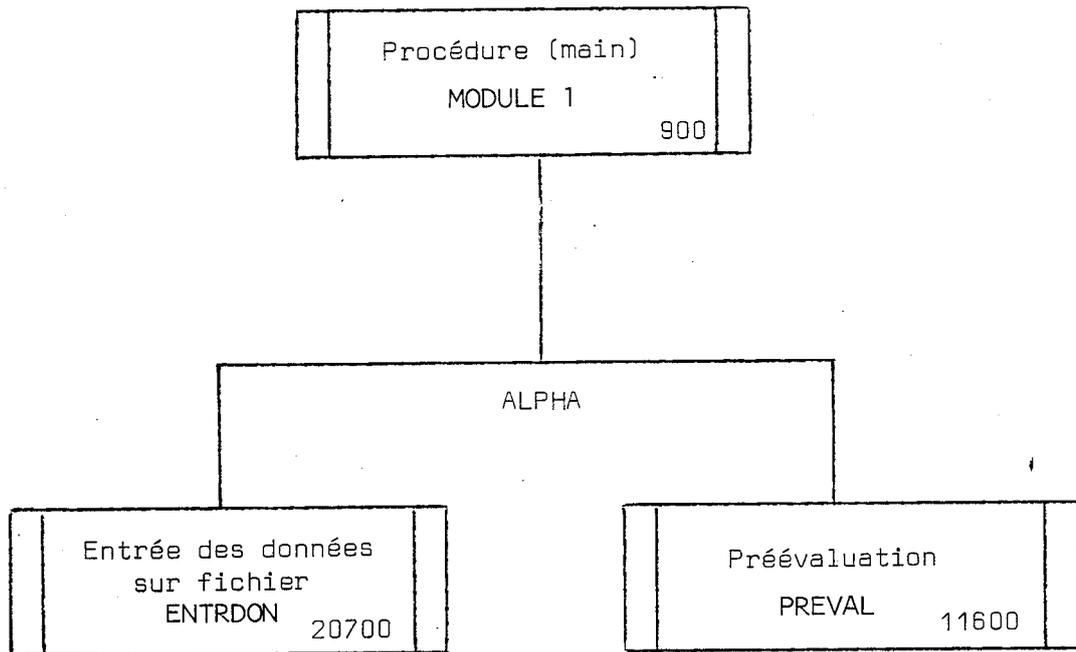
L'arbre overlay du modèle s'établit comme suit.

Deux remarques préalables sont à faire au niveau de ces schémas :

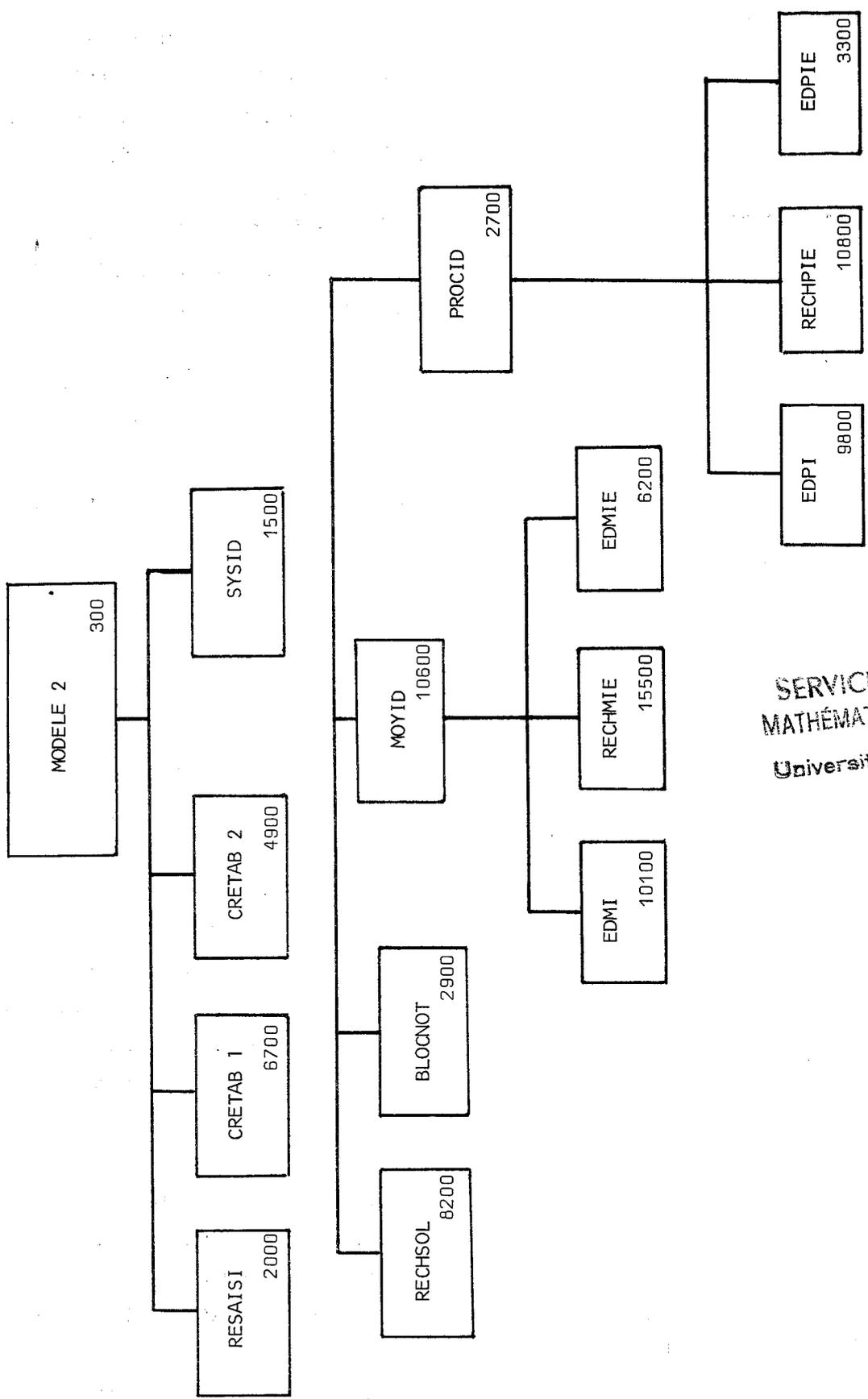
- Il faut ajouter à la taille de la branche maximale (qui est la somme des tailles des procédures intervenant dans cette branche) la taille des données et 30 K octets environ de procédures PL/1 rajoutées au link-edit, pour avoir le nombre de K octets nécessaires à la bonne exécution du programme.

- Dans l'exemple présenté en annexe, le volume des données était d'environ 19 K octets, et l'exécution s'est faite dans une partition de 104 K, avec un encombrement maximum effectif de 68 K (programme) plus environ 18 K (données), soit 86 K. Le gain dû à l'overlay a donc été dans ce cas de 40 % environ puisqu'une procédure unique aurait nécessité 140 K (90 K de programmes, 19 K de données, et 30 K rajoutés au link-edit). Cette remarque s'applique au second modèle.

Premier programme :



Deuxième programme :



SERVICE POLYCOPIE  
MATHÉMATIQUES APPLIQUÉES  
Université de GRENOBLE

*La branche la plus longue est d'environ 28 K octets*

Comme nous l'avons vu au paragraphe 4.6.1, il nous est apparu essentiel de pouvoir effectuer des déclarations dynamiques, et en PL/1, ceci peut se faire grâce à l'utilisation des structures de blocs ou à l'attribut CONTROLLED. Or, l'emploi de l'overlay implique nécessairement l'existence de procédures externes, et le passage de données ("dynamiques") entre deux procédures externes ne peut s'effectuer que si ces dernières ne sont pas déclarées à l'intérieur d'un bloc contenu dans l'une des deux procédures. D'où l'utilisation de l'attribut CONTROLLED pour toutes les déclarations dynamiques du programme.

*Remarque : Il est de fait, par ailleurs, que l'utilisation du BEGIN coûte légèrement plus cher en encombrement mémoire que l'utilisation de variables controlled.*

*Ceci est illustré par les tests suivants (pages suivantes). La différence est d'environ 400 octets.*

#### 4.6.3. Remarques relatives à la mise au point et aux performances du modèle

Comme nous l'avons signalé précédemment, pour que l'overlay soit réellement efficace, il fallait non seulement qu'il apporte un gain au niveau de l'encombrement global du programme, mais aussi que l'arbre overlay lui-même présente une certaine logique. En définitive, l'overlay décrit dans les pages précédentes tient compte de ces deux contraintes, ce qui signifie en particulier que chacune des procédures externes a pu être testée indépendamment des autres, avec un jeu d'essai qui lui était propre. Une telle conception nous a donc permis d'améliorer de façon assez nette les délais de mise au point du modèle, la programmation et les tests des différentes procédures ayant été réalisés dans la plupart des cas en parallèle.

Par ailleurs, une telle conception permettra, parce qu'elle implique une structure modulaire du programme, de résoudre assez facilement tous les problèmes de maintenance (ou d'amélioration du package), qui risquent de se poser pour des applications autres que celles des terminaux de gestion. Il est certain, par exemple, que pour des applications dans lesquelles la notion d'e.o.m. n'intervient pas (du type "Voyage en Angleterre" - Cf. chapitre 5), certains segments ne servent à rien. Il serait, par contre, souhaitable d'y introduire d'autres segments mieux adaptés au problème. Pour des applications d'évaluation de

TESTS: PROC OPTICS(MAIN);

TESTS: PROC OPTICS(MAIN);  
CCL N BIN FIXED(31);  
N=10000;  
BEGIN;  
DCL T(N) FIXED(3);  
T=C;  
END;  
END;

TEST AVEC BEGIN

1  
2  
3  
4  
5  
6  
7  
8

DEFAULT OPTIONS(S) USED

144-LEVEL LINKAGE EDITOR OPTICALS SPECIFIED MAP,LIST  
 VARIABLE OPTIONS (USED - SIZE=(52160,16384))

- 1EM0461 IME0000
- 1EM0461 IMEM51A
- 1EM0461 IMEM51B
- 1EM0461 IMEM51C
- 1EM0461 IMET51A

MODULE MAP

CONTROL SECTION			ENTRY					
NAME	ORIGIN	LENGTH	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
TESTS	00	182	IMESAPC	31A	IFESAPD	322	IMESAPA	32A
**TEST5A	1B8	130	IFESADF	33A	IMESADB	342	IMESADE	34A
IFEMAIN	2E8	C	IFESAFB	35A	IFESAFB	362	IMESAFB	36A
IFENTRY	2F0	C	IMESAFQ	37A	IFESARC	848	IMESADD	C40
IFESAP *	300	ADE	CTA					
IFESTR *	0DE	634	IMESTRE	CC8	IFESTRC	DDE		
IFEBEG *	1417	90	IFEBEGN	1410	IFEBEGA	145C		
IFEERR *	145C	725	IMEERRD	1490	IMEERRB	14A4	IMEERRA	14AE
			IFEERRE	1826				
IFEOLC *	1BCC	554	IMEOCLA	1EC0	IMEOCLR	18C2	IMEOCLD	18C6
IFEPRT *	211E	2CB	IMEPRTA	2118	IFEPRTE	211A		
IFESIZ *	23EC	C	IFESIZE	23E0				
IFETAB *	23FC	C	IFETAES	23F0				
IFEIOF *	24FC	2DC	IFEIOFA	2400	IMEITAZ	265E	IMEITAX	26AA
			IMEITPA	26BE				
IFESPRT *	24EC	21						

PSEUDO REGISTERS

NAME	ORIGIN	LENGTH	NAME	ORIGIN	LENGTH	NAME	ORIGIN	LENGTH
IMEQINV	CC	4	IMELEEM	4	4	IMEQLNF	C	4
IMEQSLA	IC	4	IMEQLDQ	4	4	**TEST5C	IC	4
IMEQLSA	ZC	4	IMEQLM1	24	4	IMEQLM3	ZC	4
			IMEQLM2	28	4			
			IMECTIC	8	4			
			**TEST5B	16	4			
			IMEQLM3	28	4			

NAME	CRIGIN LENGTH						
IMEQLWA	4	IMEQLNE	34	IMEQLCA	38	IMEQVDA	30
IMEQFVC	4	IMEGFPC	44	IMECLFL	48	IMEQADC	50
IMEQLPA	4	IMEGSAR	58	IMEORTC	50	IMEQSFC	60
IMEQSPR	4	IMEGXLY	68	IMEQVTV	70		

TOTAL LENGTH OF PSEUDOC REGISTERS 78  
 ENTRY ADDRESS 41C  
 TOTAL LENGTH 271C

\*\*\*\*\* DOES NOT EXIST BUT HAS BEEN ADDED TO DATA SET

DIAGNOSTIC MESSAGE DIRECTORY

TEM04E1 WARNING - SYMBOL PRINTED IS AN UNRESERVED EXTERNAL REFERENCE, NCAL WAS SPECIFIED.

TEST AVEC VARIABLES CONTROLLED

TEST6: PROC OPTIONS(MAIN);

TEST6: PROC OPTIONS(MAIN);  
DCL N BIN FIXED(31);  
N=10000;  
DCL T(\*) FIXED(3) CTL;  
ALLOCATE T(N);  
T=0;  
END;

1  
2  
3  
4  
5  
6  
7

F44-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED MAP, LIST  
 VARIABLE OPTIONS USED - SIZE=(92160,16384)

DEFAULT OPTION(S) USED

- MEM0461 IPE0000
- MEM0461 IMEM91A
- MEM0461 IPEM91B
- MEM0461 IPEM91C
- MEM0461 IMETERA

MODULE MAP

CONTROL SECTION			ENTRY			MODULE MAP		
NAME	ORIGIN	LENGTH	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
TEST6	C0	67	THESADA	178	THESAPC	192	THESAPA	192
TEST6A	FB	67	THESAPB	1AA	THESADP	19A	THESAPD	19A
IMEMAIN	160	C	THESAPC	1CA	THESADQ	1B2	THESADE	19A
IPEM91	168	C	THESAPD	1EA	THESAFB	1D2	THESAFD	1DA
IPEM91A	178	ADB	THESAPF	AF2	THESAFQ	1F2	THESADD	9C0
IPEM91B			IMESTRA	C50	HESTRB	C52	HESTRC	C56
IPEM91C			HEBEGN	1288	HEDECA	12C8	HEERRA	1324
IMETERA			HEERRD	1308	HEERRC	1312	HEERRB	131C
			HEERRR	199E				
			HEOCLA	1A38	HEOCLB	1A3A	HEOCLC	1A3C
			HEPRTA	1F90	HEPRTB	1F92		
			HESIZE	2258				
			HEATABS	2268				
			HEIOFB	2278	HEIOFA	227A	HEITAZ	2516
			HEITAA	2536				
			HESPRT	2558				

PSEUDO REGISTERS

NAME	ORIGIN	LENGTH	NAME	ORIGIN	LENGTH	NAME	ORIGIN	LENGTH
HEINVT	00	4	HEQTTG	8	4	HETEST63	C	4
HEST6C	10	4	HEQLWD	18	4	HEQLWI	1C	4
HESTW2	20	4	HEQLW4	28	4	HEQLWE	2E	4

NAME	ORIGIN LENGTH						
IHEQLCA	30	IHEQVDA	34	IHEQFVD	38	IHEQFDP	3C
IHEQCFI	40	IHEQADC	48	IHEQLPR	4C	IHEQSLA	50
IHEQSAR	54	IHEQLWF	58	IHEQRYC	5C	IHEQSFC	60
IHEQSPR	64	IHEQLV	68	IHEQVT	70		

TOTAL LENGTH OF PSEUDO REGISTERS 78  
 ENTRY ADDRESS 168  
 TOTAL LENGTH 2580

\*\*\*GO DOES NOT EXIST BUT HAS BEEN ADDED TO DATA SET

DIAGNOSTIC MESSAGE DIRECTORY

IEWC461 WARNING - SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE; NCAL WAS SPECIFIED.

matériel informatique, en particulier, il faudrait notamment revoir la codification proposée précédemment, c'est à dire, en fait, refaire le second module de la "préévaluation". Nous estimons que ce genre de modifications pourra être réalisé très facilement, dans la mesure où les difficultés qui en découlent ont été envisagées dès la phase de spécification (et, en particulier, la structure overlay du programme en tient compte).

En ce qui concerne la réalisation effective du modèle, il faut signaler qu'elle n'a pas posé de problèmes techniques particuliers au niveau de la programmation, une fois les options de base définies (overlay, variables CONTROLLED). En réalité, nous estimons que la difficulté essentielle a été de manipuler correctement les différents fichiers et tableaux associés à une application donnée. Cette difficulté provient du fait que le nombre de tableaux est relativement élevé (30), et que la structure de ces tableaux n'est pas unique (malgré un effort de standardisation).

A titre indicatif, notons que le nombre de segments est de 17, le nombre de procédures PL/1 (externes et internes) de 28, et le nombre d'instructions de l'ensemble des programmes de 1 570. Les temps de compilation étant respectivement de 28.74 et de 1.13.37 pour chacun des modules (4.74 et 7.48 pour le link-edit).

Pour conclure, il nous faut donner quelques indications sur les performances du programme ou, tout au moins, sur les temps d'exécution des tests qui ont été réalisés. Avec le jeu d'essai proposé en annexe, la préévaluation a duré 6.31 sec., et l'évaluation-adaptation 8.19 sec. (Ce jeu est établi à partir de données concrètes recueillies au B.H.V.). Avec le paramètre "INTERNE", les temps d'exécution ont été respectivement de 6.31 et 4.08, puisque les éléments de solution sont indépendants du type d'adaptation. De façon évidente, les deux paramètres de base qui conditionnent directement les temps C.P.U. du modèle sont le nombre de fonctions et le nombre d'e.o.m. Le nombre d'éléments de solution intervient indirectement, puisque l'on estime qu'il représente, dans le cas général, 10 % du produit (nombre de fonctions, nombre d'e.o.m.). Pour la préévaluation, d'autres essais ont été effectués, en rajoutant ou en supprimant certains e.o.m. (ou fonctions), et on a pu constater ainsi, par exemple, qu'avec le même nombre de fonction et trois fois plus d'e.o.m. (30 au lieu de 10) que dans le cas proposé en annexe, le temps d'exécution augmentait de 2 secondes environ, soit 25 % (par ailleurs, le nombre d'éléments de solution était multiplié par 4).

Nous ne pouvons malheureusement pas en déduire une formule permettant de calculer les temps d'exécution en fonction du nombre d'e.o.m. et du nombre de fonctions (les essais effectués n'étant pas assez nombreux). Cependant, les quelques chiffres obtenus nous autorisent à penser que le modèle est "rentable", et nous estimons qu'avec un jeu d'essai complet l'ensemble des opérations ne devrait pas dépasser 3 minutes de temps C.P.U. (pour une application "normale").

En réalité, c'est surtout au niveau de la saisie des données qu'il faut se pencher sur le problème des délais et de la rentabilité du modèle. Pour l'ensemble des données sur lesquelles nous avons travaillé, l'opération de saisie peut être évaluée à 1 homme/mois (30 e.o.m. et 15 fonctions). Pour l'application B.H.V. elle-même, dont les données sont en cours de saisie, nous avons évalué les délais à 3 hommes/mois, en tenant compte du fait qu'il y aurait, en réalité, 2 personnes à mi-temps pendant 3 mois (puisque, comme nous l'avons vu précédemment, les profils requis pour la définition des moyens et des processus sont différents). Ces temps peuvent paraître longs, mais ils sont, à notre avis, indispensables pour une bonne application de la méthode (analyse détaillée des systèmes fins et moyens de façon indépendante). En conséquence, nous pensons, d'une part, que ces temps seront encore plus longs pour des entreprises "non préparées", et d'autre part, que dans la mesure où ces temps ne sont pas respectés, le problème du choix des moyens nécessaires à la satisfaction d'un ensemble d'objectifs ne sera jamais que partiellement résolu (avec ou sans modèle).

Remarque : Les tous derniers essais (complets) donnent les résultats suivants  
(pour le modèle 2)

nombre de processus → 5  
nombre de moyens → 12  
nombre de e.o.m → 59  
nombre de fonctions → 34  
nombre d'éléments  
de solution ~ 120

Temps CPU (exécution) 1.00.01



## Chapitre 5. GENERALISATION - CONCLUSIONS

---

### 5.1. Quelques notions relatives à la théorie de la modularité

Il est extrêmement difficile de donner une définition précise de la notion de "modularité". La meilleure preuve en est qu'elle n'a pas la même signification pour l'ensemble des constructeurs, ce qui n'a pas grande importance après tout, puisque les appareils dits "modulaires" se vendent mieux que les autres !

Pour notre part, nous pensons que la modularité est un concept qui recouvre un ensemble de propriétés, le problème étant bien sûr de déterminer ces propriétés. Nous nous contenterons de présenter succinctement les propriétés fondamentales, notre but n'étant pas de reprendre point par point les éléments de la théorie de la modularité présentée par M. LUSSATO dans LION II.

Nous appellerons  $S$  le système de référence (machine)  
 $m_i$  ses éléments

#### a) Cloisonnement

A cette propriété est lié le degré d'indépendance des éléments du système. Si les éléments sont cloisonnés, lorsque l'un d'eux subit un dommage, il peut être isolé des autres sans empêcher leur fonctionnement (exemple : les modules d'Apollo XI).

Dans notre cas particulier, le cloisonnement est une propriété très intéressante. Supposons, en effet, que l'une des fonctions du SF vienne à disparaître : si la machine idéale fournie par le modèle est cloisonnée, il est inutile de refaire l'évaluation, il suffit de retirer de la machine idéale l'e.o.m. qui satisfaisait la fonction disparue (à condition évidemment qu'il ne satisfasse pas d'autres fonctions).

#### b) Fractionnement

Cette propriété permet d'évaluer le rapport entre le nombre de fonctions satisfaites par  $S$  ( $N$ ) et le nombre d'éléments de  $S$  ( $M$ ).

c) Echange standard

Propriété caractérisant la possibilité de remplacer certains éléments défectueux, remplacement ne nécessitant pas de compétence technique particulière, donc accessible à des non-spécialistes (cette propriété est assimilable au "plug-in" : connection immédiate).

*Remarque : Il est clair que n'importe quel élément ne peut remplacer un élément donné. On arrive ainsi à définir des classes d'équivalence d'éléments (vis à vis de l'opération de remplacement), appelées "fonctions modulaires". Les éléments d'une même classe ne sont pas obligatoirement compatibles.*

d) Compatibilité

Elle se définit comme la propriété d'un élément à être "accepté" par un système donné. Par extension, elle permet de définir une notion de distance entre les systèmes. Lorsque le nombre d'éléments ( $m_i$ ) compatibles entre deux systèmes S1 et S2 est élevé, les deux systèmes sont proches l'un de l'autre.

e) Flexibilité

Propriété mesurant l'aisance avec laquelle les extensions ou les réductions de systèmes peuvent être effectuées. Généralement, elle est exprimée par le rapport de coût suivant :

$$\frac{C(S + m_i)}{C(S) + C(m_i) + C'}$$

où  $C(S + m_i)$  représente le coût du système comprenant dès sa conception l'élément  $m_i$

$C(S)$  le coût du système (sans  $m_i$ )

$C(m_i)$  le coût de l'élément  $m_i$

$C'$  le coût de la connection

Enfin, une dernière notion est essentielle, celle de noyau d'un système. Il correspond au sous-système le plus réduit que l'on puisse imaginer. Le noyau comprend donc l'ensemble des éléments de base du système, chacun de ces éléments étant nécessaire à la réalisation de l'une des quelconques fonctions que l'on se propose de satisfaire. On en déduit que le noyau ne peut se définir que par rapport à un ensemble de fonctions de référence, et l'on retombe de façon évidente sur les problèmes exposés au cours du chapitre II. (La définition préalable des fonctions est nécessaire à toute évaluation de système, modulaire ou non). Par ailleurs, les éléments du système S qui n'appartiennent pas au noyau sont appelés modules dans la mesure où ils possèdent alors obligatoirement l'une des

propriétés exposées précédemment. Il ne faut pas en déduire pour autant que le système global est modulaire (abus très fréquent chez les constructeurs), car tout système l'est plus ou moins à la limite.

*Remarque : Les éléments opérationnels de moyens sont des modules potentiels. Ils correspondent, comme nous l'avons dit au chapitre II, à la plus petite partie d'un outil, se suffisant à elle-même mais associée au noyau pour assurer une fonction donnée.*

Le modèle, pour sa part, n'utilise pas les notions présentées ici. Il se borne à rechercher le système idéal, et non pas le système idéal le plus "modulaire". L'introduction de ces notions ne pourra se faire, en fait, que dans un modèle conversationnel, car seul le dialogue utilisateur-modèle peut permettre la construction d'un système de haute modularité (construction pas-à-pas). Il n'en reste pas moins vrai que ces notions interviennent au niveau de la conception même du modèle, et les sorties fournies par lui sont tout à fait compatibles avec une étude (a posteriori) de la modularité. L'utilisateur peut, en particulier, déterminer aisément les fonctions modulaires.

L'ensemble de ces idées nous permet de présenter les objectifs de la version 2.

## 5.2. Présentation de la version 2

Comme nous l'avons évoqué précédemment, l'intérêt de la version 2 est de permettre à l'utilisateur de dialoguer avec le modèle, en cours d'évaluation. En effet, l'inconvénient majeur de la version actuelle est de figer en début d'évaluation l'ensemble des paramètres, ce qui n'est pas réaliste, si l'on se réfère aux méthodes utilisées dans la pratique.

Une telle version conversationnelle devra permettre :

- Une modification de la pondération appliquée aux critères d'évaluation, de façon à comparer les résultats et orienter son choix vers telle ou telle solution. Ceci implique, en fait, que l'utilisateur pourra déterminer avec précision l'influence de chaque critère sur la solution définitive. Par extension, le modèle fournira la pondération "à appliquer" pour l'obtention d'une solution choisie a priori.

Bien que cet aspect soit en contradiction avec l'objectif initial du modèle, il présente un avantage non négligeable : permettre l'analyse d'un système existant et découvrir quel a été le rôle de chaque critère lors de l'élaboration de ce système, ce qui n'est pas toujours connu de façon explicite.

- Une modification des notes partielles affectées à chaque élément de solution, de façon à mettre en évidence l'importance d'un e.o.m. (ou d'une fonction) dans la solution générale. Il ne faut pas perdre de vue, en effet, que la phase de notation dépend entièrement de l'utilisateur (et de ses souhaits particuliers).

- Une modification des incompatibilités entre éléments opérationnels de moyen et des implications entre éléments de solution. Ceci revient à faire abstraction des contraintes technologiques (ou purement logiques) qui existent entre ces divers éléments. L'objectif est de pouvoir regrouper certains e.o.m. (incompatibilités négatives) afin de les traiter en bloc, c'est à dire de créer des fonctions modulaires potentielles. En d'autres termes, cette opération pourrait éventuellement permettre de travailler sur une matrice des éléments de solution diagonale par blocs (en procédant à une renumérotation des colonnes). Chaque bloc constituerait alors un domaine de modularité (fonction modulaire), c'est à dire un ensemble d'e.o.m. compatibles et capables de satisfaire un ensemble donné de fonctions.

- Une modification de la pondération associée aux fonctions composant les différents processus, afin de mettre en évidence les "fonctions-clés" de l'application et de poser éventuellement le problème de la création de nouveaux e.o.m. Ceci peut arriver lorsqu'il n'existe pas d'e.o.m. (dans le système de référence) satisfaisant pleinement une fonction de poids élevé.

- Enfin, une évaluation pas-à-pas des différentes solutions, pour donner à l'utilisateur la possibilité d'arrêter le déroulement du programme lorsqu'il le désire, et d'effectuer certains regroupements d'e.o.m. (ou de fonctions) ou certains calculs relatifs aux facteurs de modularité. L'intérêt évident étant de pouvoir ré-injecter dans le modèle, en cours d'évaluation, un nouvel SF ou un nouvel SM en fonction des solutions partielles déjà obtenues. Ce mécanisme contribuerait à améliorer nettement le degré de modularité du système idéal final.

Sur le plan pratique, une telle version supprimerait l'existence des deux programmes (actuellement indispensables), et l'ensemble des opérations pourrait s'effectuer dans des délais beaucoup plus courts ; ce qui obligerait, par contre-coup, l'utilisateur à avoir une très bonne connaissance de son système (la phase

de notation des éléments restant le point délicat d'une telle manipulation).

Concrètement, il a été prévu d'implanter par étapes successives le modèle conversationnel sur le modèle actuel (puisque les modules de base du traitement restent valables). La durée de réalisation est évaluée à un homme/année.

### 5.3. Autres applications envisageables

Il va de soi que si le but premier du modèle est de rechercher la "machine idéale" (ou le processus idéal en adaptation interne), son objectif second est bien d'associer à un ensemble de fonctions (ou de processus) l'ensemble des moyens existant sur le marché et satisfaisant ces fonctions de façon optimale. Cet objectif devient notamment l'objectif unique du modèle lorsque, dans la définition des données d'entrée, l'utilisateur fait abstraction des éléments opérationnels de moyen attachés à chaque moyen M. (Ceci est d'ailleurs obligatoirement le cas des applications pour lesquelles la décomposition n'offre aucun intérêt, la notion d'incompatibilité entre e.o.m. disparaît alors totalement, elle est remplacée par celle d'incompatibilité entre moyens). Le modèle se contente, dans ce cas, de calculer la meilleure solution, c'est à dire la meilleure association "processus-moyens existants", ce qui est illustré par l'exemple qui va suivre.

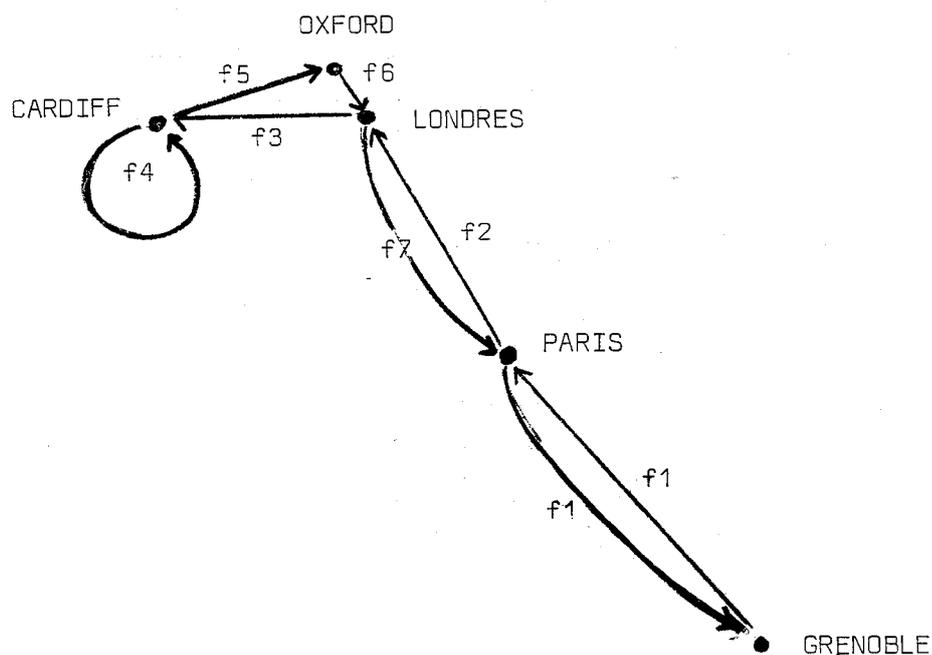
*Remarque : Pour ce type d'évaluation, l'adaptation interne n'a absolument aucun intérêt car elle ne fournit pas des informations suffisamment précises : on connaît toujours a priori l'ensemble des processus que peut satisfaire un ensemble de moyens. Ceci corrobore la remarque faite au cours du chapitre II : l'adaptation interne n'a d'intérêt que pour le constructeur, et que si l'on travaille au niveau des e.o.m.*

#### Exemple :

Nous nous proposons d'étudier l'organisation d'un voyage en Angleterre, du point de vue des modes de transport, sachant que l'itinéraire est établi (ce qui détermine les processus et les fonctions), et que les moyens envisagés sont : le train (M1), le bateau (M2), l'avion (M3), le car (M4), la voiture (M5), la voiture de location (M6), l'auto-stop (M7).

Trois critères d'évaluation seront pris en considération : le prix, l'agrément, les délais.

L'itinéraire est le suivant :



Sept fonctions (liées à des déplacements) sont prévues. L'ensemble se divise en trois processus :

- P1 : Voyage aller (f1, f2), de poids 1.
- P2 : Séjour en Angleterre (f3, f4, f5, f6), de poids 2.
- P3 : Voyage retour (f7, f1), de poids 1.

Toutes les fonctions sont de poids 1, sauf la fonction 4 (poids 2). Bien que les trajets aller et retour soient identiques, nous supposons que le déplacement PARIS-LONDRES diffère selon le sens suivant lequel il est effectué. Il y a donc deux fonctions distinctes associées (f2 et f7). (On peut, par exemple, supposer qu'au retour les objectifs ne sont pas les mêmes car le nombre de personnes a diminué). Nous n'explicitons pas les objectifs de chaque fonction.

Nous supposons également que l'évaluation est effectuée en 1980, date à laquelle le tunnel sous la Manche sera construit ! (Faute de quoi, f2 et f7 ne peuvent être satisfaits que par M2 et M3).

La matrice des éléments de solution se présente comme suit :

	M1	M2	M3	M4	M5	M6	M7
f1	1		2	3	4	5	6
f7	7	8	9		10		
f3				11	12	13	
f4				14	15	16	
f5	17				18	19	20
f6	21				22	23	24
f2	25	26			27	28	

Remarque 1 : Les éléments de solution incompatibles le sont :

- soit de par leur nature : (f1, M2) par exemple,
- soit parce que l'on estime a priori qu'ils ne sont même pas d'ordre 1 (donc tout à fait défavorables).

Remarque 2 : La colonne 5 est incompatible avec toutes les autres. Ce qui signifie que l'on ne peut avoir de solution partielle dans laquelle un élément issu de la colonne 5 suit un élément issu d'une autre colonne. La réciproque n'est pas vraie. Ainsi, pour le processus P1, la solution (4, 9) est possible : elle consiste à laisser sa voiture à Paris.

Il faut ensuite noter chaque élément de solution (suivant les 3 critères retenus) et définir les implications qui existent entre eux. Par exemple, si les trajets f5 et f6 ont été effectués en auto-stop, on aura tout intérêt à effectuer f1 en avion (pour le retour), afin d'arriver à Grenoble en forme et dans les délais prévus : ainsi 20 et 24 impliquent 2.

Le modèle peut alors réaliser l'évaluation et fournir une solution de la forme :

- P1 (4. 25)
- P2 (13. 16. 17. 21)
- P3 (8. 2)

L'exemple choisi est évidemment extrêmement simple, mais nous espérons qu'il a convaincu le lecteur des possibilités d'utilisation du modèle dans d'autres domaines que celui des terminaux de gestion.

Au vrai, le point délicat reste celui de la codification qui doit s'adapter à l'application traitée. L'exemple ci-dessus ne peut, en effet, bénéficier de la préévaluation, et la matrice des éléments de solution doit être calculée manuellement.

#### 5.4. Conclusions

Le modèle présenté dans ce rapport se veut simple, facilement compréhensible et facilement utilisable. Ce fut notre but premier. L'ensemble des recherches effectuées aussi bien dans le domaine de l'organisation que dans le domaine informatique, pour réaliser effectivement le modèle et le mettre au point fait que le résultat a de loin dépassé cet objectif, et l'optique dans laquelle doit se situer une éventuelle version 2 est déjà nettement définie. Néanmoins, nous pensons que l'efficacité de tels modèles ne peut être réelle que s'ils sont établis en étroite collaboration avec des personnes ayant une bonne connaissance de l'entreprise. Dans ce sens, on peut affirmer sans risque que l'ère des "organiseurs-informaticiens" est ouverte.

Dans ce domaine, l'objectif à long terme est alors de construire un modèle capable d'évaluer des systèmes informatiques. On rejoint par là l'un des problèmes fondamentaux posés par l'analyse de gestion, au niveau de l'analyse fonctionnelle : celui de la détermination des moyens informatiques nécessaires à l'entière satisfaction d'un certain nombre d'objectifs de gestion. La plupart des packages (ou des méthodes) dits "d'aide à l'analyse" semblent prendre cette direction, mais, à l'heure actuelle le problème n'est pas résolu pour autant, par manque, semble-t-il, d'organiseurs compétents et de collaboration réelle. Seule, à notre avis, une véritable théorie des systèmes d'organisation pourra résoudre le problème dans toute sa généralité. Cette théorie devra être comprise et appliquée par des informaticiens. Il est donc temps que ces derniers prennent conscience de leur véritable attribution qui n'est pas de réaliser des softwares, mais plutôt, dans un premier temps, de comprendre pourquoi il faut les réaliser.

## CHAPITRE VI. ANNEXES

---

Le chapitre est divisé en quatre parties :

6.1. Présentation des données ayant servi à la constitution du jeu d'essai.

Le lecteur y trouvera successivement les rubriques suivantes :

- liste des processus
- liste des moyens
- liste des fonctions
- liste des e.o.m.
- liste des éléments de solution
- autres paramètres (en particulier, liste des paramètres généraux de l'application)

6.2. Présentation des résultats obtenus

6.3. Exemple de programme. Algorithme de recherche de la MIE

6.4. Bibliographie

### 6.1. Constitution du jeu d'essai

Une remarque préalable est à faire en ce qui concerne ce jeu d'essai : toutes les données qui sont présentées ici sont réelles et extraites du B.H.V. Mais en réalité, le jeu lui-même a été simplifié afin de pouvoir mieux mettre en évidence le fonctionnement même du modèle. Le lecteur pourra, en particulier, refaire manuellement une partie de l'évaluation s'il le désire, la matrice des éléments de solution étant relativement réduite.

\*\*\*\*\*  
 \*\*LISTE DES PARAMETRES GENERAUX DE L'APPLICATION\*\*  
 \*\*\*\*\*

ENTREES & SORTIES  
DU MODELE 1

TYPE D'ADAPTATION DESIREE :	EXTERNE		
NOMBRE DE PROCESSUS	5		
NOMBRE TOTAL DE FONCTIONS	10		
MAXIMUM DE FONCTIONS ASSOCIEES A UN PROCESSUS	8		
NOMBRE DE MOYENS	5		
NOMBRE TOTAL D'E.C.M.	10		
MAXIMUM D'E.C.M. ASSOCIES A UN MOYEN	6		
NOMBRE DE CARTES D'INCOMPATIBILITE	4		
MAXIMUM D'INCOMPATIBILITE POUR UN ECM	1		
NOMBRE DE CARTES DE DISTANCES	4		
MAXIMUM DE DISTANCES ASSOCIEES A UN MOYEN	3		
NOMBRE DE CRITERES	3		
POIDS DES CRITERES	2	1	1
NOMBRE DE SOLUTIONS GLOBALES A CONSIDERER	25		
NOMBRE DE SOLUTIONS PARTIELLES A CONSIDERER	5		
COEFFICIENT DE PONDERATION DES IMPLICATIONS	2		

\*\*\*\*\*  
 \*\*LISTE DES PROCESSUS\*\*  
 \*\*\*\*\*

NUMERO DU PROCESSUS	1				
LIBELLE DU PROCESSUS	TICKET				
POIDS DU PROCESSUS	1				
NOMBRE DE FONCTIONS	5				
LISTE DES FONCTIONS	1	3	5	6	7
POIDS DES FONCTIONS	1	2	2	2	1

2							
ENCAISSE EXTERIEUR							
1							
8							
1	3	5	8	7	2	9	10
1	2	2	1	1	2	1	1

3							
CARNET D'ACHAT							
2							
8							
1	4	3	8	7	9	5	10
1	2	2	1	1	1	2	1

4						
APPLICATION						
1						
6						
1	2	8	9	10	5	
1	1	2	2	1	2	

5				
EMFORTE				
1				
5				
1	3	10	8	5
1	2	1	2	1

\*\*\*\*\*  
 \*\*LISTE DES FONCTIONS\*\*  
 \*\*\*\*\*

NUMERO	CODAGES	LIBELLE	
1	0243 09 41006 41003 41003 41003 52603 52602 41004 41003 57710	CONSTANTES ENTETE	DATE CODE RAYON NUMERO D'ORDRE NUMERO DE POSTE CARTOUCHE B.H.V. LETTRE RAYON NUMERO VENDEUR NUMERO DOC. D'ENTREE AUTRES
2	0243 6 30806 52610 41003 41003 52610 30803	CONST. LIVRAISON	TYPE CLIENT ZONE SECTEUR INDICATEURS DATE
4	0243 8 52610 57720 41006 41006 41010 47010	IDENT. MARCHANDISE	DESIGNATION DESCRIPTION CODE NOMENCLATURE CODE MARCHANDISE QUANTITE PRIX UNITAIRE
5	0633 3 41010 41004 41010	MONTANT	SOUS-TOTAL T.V.A. TOTAL
6	4643 1	MONTAIE RECUE	
7	9043 3 1 72610 78801	ERREUR	ANNULATION COMMANDE EFFACEMENT LETTRES EFFACEMENT LIGNE
8	0243 3 30802 30804 30803	MODALITES ENCAISSE	APPLICATION ENCAISSEMENT EMPORTE
9	6043 41015	ESCOMPTE	
3	0311 2 52630 30805	DOCUMENT TRANSACTION	
10	0243 30805	MODALITES PAIEMENT	

\*\*\*\*\*  
 \*\*LISTE DE MOYENS\*\*  
 \*\*\*\*\*

NUMERO DU MOYEN  
 LIBELLE DU MOYEN  
 NOMBRE D'E.O.M.  
 LISTE DES E.O.M.

\*\*1  
 MDTS FRIDEN  
 3  
 1 3 10

\*\*2  
 NCR 250  
 4  
 1 7 10 2

\*\*3  
 SINTRA TE 4000  
 6  
 8 5 10 2 7 5

\*\*4  
 SAGE S 8000  
 4  
 8 10 7 5

\*\*5  
 UNIVCTE  
 6 9 4 8

\*\*\*\*\*  
 \*\*LISTE DES E.C.M.\*\*  
 \*\*\*\*\*

NUMERO	CODAGES	LIBELLE	
1	0243 2 41010 52647	CLAVIER ALPHANUM.	NUMERIQUE ALPHABETIQUE
2	0311 6 41010 52631 41010 52631 41010 52631	IMPRIMANTES	TICKET JOURNAL LIASSE
3	0243 30806	FONCTIONS	
4	6041 41016	CALCUL	
5	9043 3 78801 72010 1	ERREUR	EFFACEMENT LIGNE EFFACEMENT LETTRES REENTER
6	0333 67730	CLAVIER TRANSACTIONS	
7	0243 67713	FONCTIONS	
8	0243 C2 41010 52656	CLAV. ALPHANUMERIQUE	NUMERIQUE ALPHABETIQUE
9	4643	TIRCEIR CAISSE	
10	0633 3 41013 41004 41007	SUBTOTAL, TAX, TOTAL	SUBTOTAL TAX TOTAL

\*\*\*\*\*  
\*\*INCOMPATIBILITES ENTRE E.G.M.\*\*  
\*\*\*\*\*

NUMERO D'E.G.M.	NCMBRE D'INCCMPATIBILITES	LISTE DES E.G.M. INCGPATIBLES
1	1	8
3	1	7
4	1	10
2	1	6

\*\*\*\*\*  
\*\*CISTANCES ENTRE MOYENS\*\*  
\*\*\*\*\*

NUMERO DE MOYEN	NCMBRE DE CCLFLES	CCLFLES	DISTANCE/MOYEN
1	2	2**3	2**4
2	3	1**3	2**4 1**5
3	2	2**4	1**5
5	2	1**4	2**3

N1: NUMEROTATION PROGRESSE PAR LE MODELE  
 N2: NUMEROTATION EVENTUELLE DE L'UTILISATEUR  
 N3: NUMERO D'ORDRE DE LA FONCTION  
 N4: NUMERO D'ORDRE DES E.C.M.  
 NA: NOTE ALPHABETIQUE

N1	N2	N3/N4	LIBELLE DE LA FONCTION	LIBELLE DE L'E.C.M.	LIBELLE DU MOYEN	NA	NOTES DES CRITERES
*	1	1	CONSTANTES ENTETE	/	CLAVIER ALPHANUM.	***	2.37
*	2	1	CONSTANTES ENTETE	/	FNCTIONS	***	1.60
*	3	1	CONSTANTES ENTETE	/	CLAV. ALPHANUMERIQUE	***	2.37
*	4	2	CONST. LIVRAISON	/	CLAVIER ALPHANUM.	***	2.10
*	5	2	CONST. LIVRAISON	/	FNCTIONS	***	1.56
*	6	2	CONST. LIVRAISON	/	CLAV. ALPHANUMERIQUE	***	2.18
*	7	3	ECCLEMENT TRANSACTION	/	IMPRIMANTES	***	2.39
*	8	3	DOCUMENT TRANSACTION	/	CLAVIER TRANSACTIONS	UNIVOTE	1.30
*	9	4	IDENT. MARCHANDISE	/	CLAVIER ALPHANUM.	***	2.62
*	10	4	IDENT. MARCHANDISE	/	CLAV. ALPHANUMERIQUE	***	2.61
*	11	5	MONTANT	/	CALCUL	UNIVOTE	2.24
*	12	5	MONTANT	/	SUSTOTIL, TAX, TOTAL	***	2.92
*	13	6	PECUNIAIRE RECUE	/	TIRAGE CAISSE	***	3.00
*	14	7	ERREUR	/	ERREUR	***	3.00
*	15	6	LOCALITES ENCAISSE	/	CLAVIER ALPHANUM.	***	1.60
*	16	6	LOCALITES ENCAISSE	/	FNCTIONS	MDTS FRIDEN	2.60
*	17	6	LOCALITES ENCAISSE	/	FNCTIONS	***	1.20
*	18	6	LOCALITES ENCAISSE	/	CLAV. ALPHANUMERIQUE	***	1.60
*	19	5	ESCOMPTE	/	CALCUL	UNIVOTE	2.57
*	20	10	LOCALITES PAIEMENT	/	CLAVIER ALPHANUM.	***	1.80
*	21	10	LOCALITES PAIEMENT	/	FNCTIONS	MDTS FRIDEN	2.92
*	22	10	LOCALITES PAIEMENT	/	FNCTIONS	***	1.20
*	23	10	LOCALITES PAIEMENT	/	CLAV. ALPHANUMERIQUE	***	1.80

6.2.

PRESENTATION DES RESULTATS OBTENUS

SORTIES DU GROUPE 2

LA MACHINE IDEALE EST COMPOSEE DES F.O.M. SUIVANTS

```
*****
* NUMERO 4 * LIBELLE DE L'F.O.M. * LIBELLE DU MOYEN *
*****
* 1 * *CLAVIER ALPHANUM. * MOTS FRISEY *
* * * * * MCR 280 *
*****
* 2 * *TYPRIIMANTES * NCR 280 *
* * * * * SYMTRA TE 4000 *
*****
* 4 * *CALCUL * UNIVOTE *
*****
* 9 * *TIPIC CAISSE * SYMTRA TE 4000 *
* * * * * UNIVOTE *
*****
* 5 * *PERFEUR * SYMTRA TE 4000 *
* * * * * SAGEM S 8000 *
*****
* 3 * *FUNCTIONS * MOTS FRISEY *
*****
```

LA DCAI TATION D'UNE TIRE SOLUTION EST ETABLIE A PARTIR DES SOLUTIONS PARTIELLES ET DES NOTES SUIVANTES :

PROCESSIONS

NUMERO	POIDS	LIBELLE	NOTE	MAXIMUM	RAPPORT
1	1	TICKET	17.98	24	0.74
2	1	ENCAISSE EXTEPIEUR	22.78	33	0.69
3	2	CARNET D'ACHAT	45.92	66	0.69
4	1	APPLICATION	19.47	27	0.72
5	1	EMPORTE	13.04	21	0.66

CETTE MACHINE SATISFAIT LES FONCTIONS DES PROCESSUS DENTREE SUIVANT LA DECOMPOSITION :

N1 : NUMERO DE LA FONCTION  
N2 : NUMERO DE L'OPERATION

N1	CODES	FONCTION	LIBELLE	FONCTION	N2	CODES	F. N. M.	NOTE
1	214131410161	CONSTANTES ENTEE		1	214131410101	2.17		
2	3111115126101	ORDONNEMENT TRANSACTION		2	3111115126101	1.37		
3	115126121							
4	115126121							
5	115126121							
6	115126121							
7	115126121							
8	115126121							
9	115126121							
10	115126121							
11	115126121							
12	115126121							
13	115126121							
14	115126121							
15	115126121							
16	115126121							
17	115126121							
18	115126121							
19	115126121							
20	115126121							
21	115126121							
22	115126121							
23	115126121							
24	115126121							
25	115126121							
26	115126121							
27	115126121							
28	115126121							
29	115126121							
30	115126121							
31	115126121							
32	115126121							
33	115126121							
34	115126121							
35	115126121							
36	115126121							
37	115126121							
38	115126121							
39	115126121							
40	115126121							
41	115126121							
42	115126121							
43	115126121							
44	115126121							
45	115126121							
46	115126121							
47	115126121							
48	115126121							
49	115126121							
50	115126121							
51	115126121							
52	115126121							
53	115126121							
54	115126121							
55	115126121							
56	115126121							
57	115126121							
58	115126121							
59	115126121							
60	115126121							
61	115126121							
62	115126121							
63	115126121							
64	115126121							
65	115126121							
66	115126121							
67	115126121							
68	115126121							
69	115126121							
70	115126121							
71	115126121							
72	115126121							
73	115126121							
74	115126121							
75	115126121							
76	115126121							
77	115126121							
78	115126121							
79	115126121							
80	115126121							
81	115126121							
82	115126121							
83	115126121							
84	115126121							
85	115126121							
86	115126121							
87	115126121							
88	115126121							
89	115126121							
90	115126121							
91	115126121							
92	115126121							
93	115126121							
94	115126121							
95	115126121							
96	115126121							
97	115126121							
98	115126121							
99	115126121							
100	115126121							

1	1	1	2	3	4	5
1	1	*	1	*	1	1
1	2	1	*	1	1	1
1	4	1	1	1	1	*
1	9	1	1	1	1	*
1	5	1	1	1	*	1
1	3	*	1	1	1	1
1	6	1	1	1	1	*
1	7	1	*	1	*	1
1	9	1	1	*	1	*
1	0	*	1	*	1	*

(cf exemple - 6.3)

LE NOYAU DE LA M.J.E. EST COMPOSE DES MOYENS SUIVANTS:

```

*****
* NUMERO *
*****
LIBELLE
*****
* 5 * INVOLTE *
*****
* 1 * MOTS FRIDEN *
*****

```

LE NOYAU COMPRET 3 C.O.M. SUPERIEUS  
LA DISTANCE EST DE: 0

LA LISTE DES SOLUTIONS SUIVANT COMME SUIIT :

SOLUTION NUMERO 1

```

*****
* NUMERO *
*****
LIBELLE
*****
* 3 * CONTRA TE 4000 *
*****

```

NUMBRE DE C.O.M. SUPERIEUS: 3  
LA DISTANCE EST DE: 0



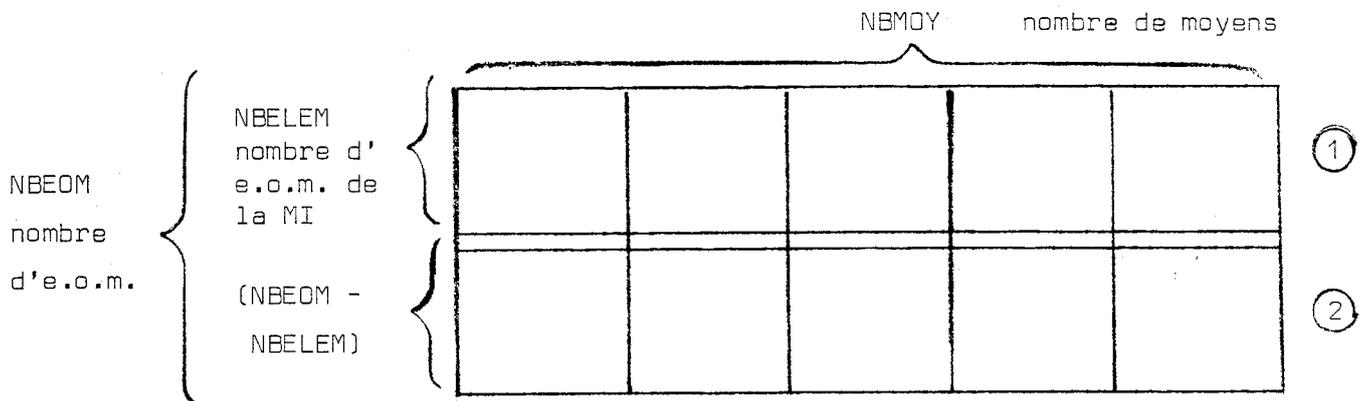


SERVICE POLYCOPIE  
MATHÉMATIQUES APPLIQUÉES  
Programme PL/T  
Université de GRENOBLE

6.3. Algorithme particulier de recherche des "MIE" et correspondant

Rappel : La machine idéale étant déterminée (elle s'exprime comme un ensemble d'e.o.m.), il faut définir toutes les combinaisons de moyens  $M_i$  contenant la machine idéale. La MIE (machine idéale existante) étant alors la meilleure de ces combinaisons, c'est à dire celle qui contient le moins d'e.o.m. superflus.

L'algorithme travaille sur la matrice booléenne MA suivante :



Cette matrice définit, pour chaque e.o.m.  $m_i$ , les moyens  $M_j$  qui le contiennent :

$$MA(i, j) = 1 \iff m_i \in M_j$$

mais, préalablement, les lignes sont ré-arrangées de telle façon que tous les e.o.m. de la MI soit regroupés vers le haut de la matrice.

1) Recherche du noyau de la MIE. Dans une première phase, on s'intéresse uniquement à la partie ① de MA, celle qui concerne les e.o.m. de la MI. Si une ligne de la partie ① de MA ne contient qu'un seul 1, la colonne correspondante appartient au noyau de la MIE. En effet, lorsqu'un e.o.m. de la MI n'est satisfait que par un seul  $M_j$ , ce  $M_j$  appartient obligatoirement à la MIE.

La première phase consiste donc à rechercher tous les  $M_i$  indispensables de la MIE.

Chaque fois qu'un  $M_i$  est retenu dans le noyau, nous stockons dans un vecteur V l'ensemble des e.o.m. qu'il apporte effectivement à la MIE.

V est un vecteur booléen de NBELEM éléments.

$V_i = 1 \iff$  le  $i$ ème e.o.m. de la MI est satisfait par l'un des M du noyau.

2) La deuxième phase consiste alors à travailler sur une matrice  $MA'$ , obtenue à partir de  $MA$  en enlevant toutes les colonnes du noyau et les lignes (de la partie ①) qui ont servi à la constitution du noyau.

L'algorithme calcule alors toutes les combinaisons possibles des colonnes restantes.

À chaque combinaison sont associés deux vecteurs booléens  $V_1$  et  $V_2$  :  
 $V_1$  représente l'union des colonnes sur les lignes de la partie ①  
 $V_2$  représente l'union des colonnes sur les lignes de la partie ②

Si  $V_1$  union  $V$  est un vecteur qui ne contient plus de 0, la solution est retenue puisque cela signifie que les e.o.m. du noyau, plus les e.o.m. de la combinaison, représentent tous les e.o.m. de la MI.

Le nombre de 1 de  $V_2$  est alors appelé le résidu relatif à la combinaison traitée. La combinaison de colonnes (associée au noyau) qui a le plus petit résidu représente la MIE. (À résidu égal, la sélection est faite sur la distance globale de la combinaison, qui est la somme des distances entre les moyens).

Exemple :

Soit la matrice  $MA$  du paragraphe 6.3 :

		MOYENS				
		1	2	3	4	5
e.o.m.	1	*	*			
	2		*	*		
	4					*
	9			*		*
	5			*	*	
	3	*				
	6					*
	7		*	*	*	
	8			*	*	*
	10	*	*	*	*	

Le noyau est constitué des moyens (1, 5)

Le vecteur  $V = (1 \ 0 \ 1 \ 1 \ 0 \ 1)$

Nous retiendrons comme solutions toutes les combinaisons possibles entre les colonnes 2, 3, 4, telles que  $V_1$  union  $V = (1 \ 1 \ 1 \ 1 \ 1 \ 1)$ .

Parmi ces combinaisons, celle qui a le plus petit résidu constitué (associée aux  $M$  du noyau) la MIE, soit (1, 3, 5) - résidu 3, distance nulle puisque la combinaison ne contient qu'un seul moyen  $M3$  -.

```

1  RECHMTE:RECHMTE: 0000;
2  DCI 1 CACAM EXT
   2 ADAPT CHAP(7);
   2 (NBOCC,NPF,MAXFCT) FIXED(3);
   2 (NBOV,NPROM,MAXFOM) FIXED(3);
   2 (ITNCCID,NINCQMD) FIXED(3);
   2 (IDISTANC,NOSTANC) FIXED(3);
   2 NBOCCT FIXED(3), 2 NBOCCT(11) FIXED(3) INIT(1,(10)0),
   2 (NPSG,NRSD) FIXED(3);
   2 XCOCC FIXED(3);
   2 NRES FIXED(2) INIT(0);
   DCI NRADAPT FIXED(3) EXT;
   DCI NBFLEW FIXED(3) EXT;
   DCI NIT(*) FIXED(3) EXT CTL;
   DCI NRELOOM FIXED(3);
   DCI MOCMD(*) FIXED(3) CTL;
   DCI DRESS (*,*) FIXED(3) EXT CTL;
   DCI MOYEN (*,*) FIXED(3) EXT CTL;
   DCI MATAP (*,*) BIT(1) CTL;
   DCI VECT(*) BIT(1) CTL;
   DCI VECTE (*) BIT(1) CTL;
   DCI VECT(*) BIT(1) CTL;
   DCI SOLUT (*,*) FIXED(3) CTL EXT;
   DCI RESID (0:100) FIXED(3) EXT;
   DCI RESIDU FIXED(3) EXT;
   DCI NOYAIN(*) FIXED(3) CTL EXT;
   DCI NOYAINL(*) FIXED(3) CTL;
   DCI COLMPS(*) FIXED(3) CTL;
   DCI YTOUP FIXED(3);
   DCI W(7) FIXED(1);
   DCI PBT ENTRY (*,FIXED(3),*);
   DCI PAC ENTRY (BIN FIXED(15)) RETURNS (BIN FIXED(15));
   DCI COMPIN FILE FAV (5(7)) RECORD;
/* ***** */
PUT PAGE;
OPEN FILE (COMBIN) INPUT;
IF ADAPT=VXTEPNE1 THEN YTOUR=NREOM;
ELSE YTOUR=NRF;
NBFLOM=YTOUR-NBFLEW;
ALLOCATE COLONES (NBADAPT);
ALLOCATE NOYAIN(0:NBADAPT);
ALLOCATE SOLUT (100,NBADAPT);
ALLOCATE MATAP (YTOUR,NBADAPT);
ALLOCATE VECT (YTOUR);
ALLOCATE VECTE (YTOUR);
ALLOCATE VECTS(YTOUR);

```



\*\*\*\*\*

```

95 DO JJ=1 TO MPADAPT;
96   DO I=1 TO IPRM+1 TO ITOUR;
97     IF MATAP(I,J) = 1 OR THEN GOTO NONOK;
98   END;
99   JJ=JJ+1;
100   COLONES(JJ)=J;
101   NONOK:END;
102   END;
103   *****

```

\*\*\*\*\*

```

104   KK=0;
105   DO I=1 TO MPADAPT-JJ-1;
106     DO J=1 TO I;
107       K=FAC(I)/(FAC(I-J)*FAC(J));
108       KK=KK+K;
109     END;END;
110     DO I=1 TO KK;
111       READ FILE (COMBIN) INTO (W);
112     END;
113     I=0;
114     I=JJ;
115     *****

```

\*\*\*\*\*

```

116     IF ADAPT=EXTREME THEN DO;
117       DO I=1 TO MPADAPT-JI;
118         K=FAC(MPADAPT-I)/(FAC(MPADAPT-I-I)*FAC(I));
119       DO JJ=1 TO K;
120         READ FILE (COMBIN) INTO (W);
121         VECT=1018;
122         DO JJ=1 TO I;
123           VECT=POW(VECT,MATAP(*,COLONES(W(JJ))),'0111');
124         END;
125         VECT=ROU(VECT,VECT,'0111');
126       DO JJ=1 TO IPRM;
127         IF VECT(JJ)=1018 THEN GOTO NONOK2;
128       END;
129       I=0;
130       DO JJ=IPRME+1 TO ITOUR;
131         IF VECT(JJ)=1018 THEN I=I+1;
132       END;
133       I=I+1;
134       RESO(L)=I;
135       SOLUT(L,I)=I;
136     DO JJ=1 TO I;
137       SOLUT(I,JJ+1)=COLONES(W(JJ));
138     END;
139     NONOK2:END;
140     END;END;
141     *****
142     *****
143     *****

```

\*\*\*\*\*



\*\*\*\*\*

```

194      PUT EDIT ('-' ) (A);
195      END;
/* ***** */
196      PUT :DROC (NR,MY,MC);
197      DO L NR FIXED(3);
198      DO L MY (*) FIXED(3);
199      DO L MC FIXED(3);
200      DO I=1 TO NR;
201      PUT SKIP;
202      PUT EDIT (MY(I)) (X(1),D,Z(0));
203      PUT SKIP(0);
204      PUT EDIT ('-' ) (A);
205      DO J=1 TO NRADAPT;
206      PUT EDIT ('-' ) (X(3),A);
207      END;
208      PUT SKIP(0);
209      PUT EDIT ('-' ) (A);
210      DO J=1 TO NRADAPT;
211      IF MATAP(I+NC,J)=1118 THEN
212      PUT EDIT ('-' ) (X(3),A);
213      ELSE PUT EDIT ('-' ) (X(3),A);
214      END;
215      END EDIT;
/* ***** */
216      RECHCOM : DROC;
217      K=0;
218      DO I=1 TO ITOUR;
219      DO J=1 TO NRELFM;
220      IF MY(J)=T THEN GOTO OK1;
221      END;
222      K=K+1;
223      MYCOMP(K)=I;
224      OK1:FN;
225      END OFF-COM;
/* ***** */
226      RECHCOM : DROC;
227      RECHCOM : DROC;
228      DO L NR1,NR2) FIXED(3);
229      DO L MY(*) FIXED(3);
230      DO L MC(*) FIXED(3);
231      DO I=1 TO NR2;
232      DO J=1 TO NRADAPT;
233      K=PRCMMY(J,I);
234      DO K=1 TO K;
235      IF PRCMMY(J,K+1)=MY(I) THEN DO;
236      MATAP(I+NR1,J)=1118;
237      GOTO OK2;
238      END;
239      END;
240      END;

```

RECHUTE: PROC;

```

241      MATAP(I+NR1, J) = 101R;
242      END;
243      PR2:END;
244      END;
245      END RECHUTE;
246      /* ***** */
247      FAC: PROC(N1) RECURSIVE BIN FIXED(15);
248      DO I = 1 BY 1 TO N1;
249      IF N1 = 0 THEN RETURN(I);
250      ELSE RETURN(N1 * FAC(N1-1));
251      END FAC;
252      /* ***** */
      END;

```



6.4.

B I B L I O G R A P H I E

- [1] M. BENAYOUM, BROY, B. SUSSMAN  
Electre  
*Note de travail - n° 49 - Juin 1966*
- [2] C. BENZAKEN  
Contribution des structures algébriques ordonnées à la théorie  
des réseaux  
*Thèse de Docteur-Ingénieur - Faculté des Sciences de Grenoble  
Mars 1968*
- [3] H. CHESNUT  
System and Engineering Methods  
*J. WILING & Sons - 1967*
- [4] C. DUVAL et J. LEPANOT  
L'organisation de l'entreprise et l'imprimé  
*Dunod - 1960 (211 pages)*
- [5] M.H. de L'ESTOILE  
La programmation de la recherche et du développement au niveau  
d'un département ministériel ou d'une grande firme.  
*CERI - Cahier 9810 - 1960*
- [6] GAGE  
Value Analysis  
*Mac Graw Hill - 1962*
- [7] R. JOHNSON, F. KAST et J. ROSENZWEIG  
The theory and management of systems  
*Mac Graw Hill - 1963*
- [8] B. LUSSATO  
Méthodes d'évaluation des fins et des moyens selon des critères  
hétérogènes  
*1970*

- [9] B. LUSSATO  
Une méthode d'évaluation des solutions modulaires : LION 2  
1970
- [10] J.G. MARCH et H.A. SIMON  
Les organisations  
Dunod - 1969
- [11] C. MARTZLOFF  
Les ordinateurs, l'analyse et l'organisation  
Dunod - 1966 (283 pages)
- [12] J. MELESE  
La gestion par les systèmes  
Editions Hommes et Techniques - 1968 (235 pages)
- [13] L.D. MILES  
Technics of Value Analysis and Engineering  
Mac Graw Hill - 1961
- [14] S.L. OPTNER  
L'analyse des systèmes et les problèmes de gestion  
Dunod - 1968 (131 pages).
- [15] J. ORLICKY  
The successful computer system  
Mac Graw Hill - 1969.