



**HAL**  
open science

## **SAMSS : un système d'aide à la modélisation et à la simulation dans le domaine des sciences sociales**

Pierre Dumarchez

► **To cite this version:**

Pierre Dumarchez. SAMSS : un système d'aide à la modélisation et à la simulation dans le domaine des sciences sociales. Ordinateur et société [cs.CY]. 1982. dumas-00305802

**HAL Id: dumas-00305802**

**<https://dumas.ccsd.cnrs.fr/dumas-00305802v1>**

Submitted on 24 Jul 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.

# CONSERVATOIRE NATIONAL DES ARTS ET METIERS

## CENTRE AGREE DE GRENOBLE (C.U.E.F.A)

---

MEMOIRE

présenté en vue d'obtenir

1<sup>e</sup> DIPLOME D'INGENIEUR C.N.A.M.

en

INFORMATIQUE

par

*Pierre DUMARCHEZ*

---

S.A.M.S.S. : Un Système d'Aide à la Modélisation et à la  
Simulation dans le domaine des Sciences Sociales

---

SOUTENU LE : 26 Octobre 1982

### JURY

Président : L. BOLLNET, Professeur Université Grenoble II

Membres : RAYMOND, Professeur CNAM. Paris.  
C. BRUNET, Ingénieur Hewlett Packard Grenoble  
B. CHATEAU, IEJE ER n° 94 du CNRS Grenoble  
B. LEPAPE, Chargé de Recherche CRISS Grenoble



# CONSERVATOIRE NATIONAL DES ARTS ET METIERS

## CENTRE AGREE DE GRENOBLE (C.U.E.F.A)

---

MEMOIRE

présenté en vue d'obtenir

le DIPLÔME D'INGÉNIEUR C.N.A.M.

en

INFORMATIQUE

par

*Pierre DUMARCHEZ*

---

S.A.M.S.S. : Un Système d'Aide à la Modélisation et à la  
Simulation dans le domaine des Sciences Sociales

---

SOUTENU LE :            Octobre 1982

### JURY

Président : L. BOLLLET, Professeur Université Grenoble II

Membres :

C. BRUNET, Ingénieur Hewlett Packard Grenoble

B. CHATEAU, IEJE ER n° 94 du CNRS Grenoble

B. LEPAPE, Chargé de Recherche CRISS Grenoble





<u>CHAPITRE II : LE LOGICIEL S.A.M.S.S.</u> .....	17
II.1. PRÉSENTATION GÉNÉRALE .....	19
II.2. CARACTÉRISTIQUES GÉNÉRALES DU LOGICIEL .....	21
II.2.1. LES OBJECTIFS .....	21
II.2.2. TYPES DE MODELES SUPPORTES .....	22
II.2.3. S.A.M.S.S. : UN OUTIL D'AIDE A L'ECRITURE ET A LA MAITRISE DES MODELES .....	24
II.2.4. LE FONCTIONNEMENT DU LOGICIEL .....	27
II.2.5. ORGANISATION INFORMATIQUE DU LOGICIEL .....	29
II.3. LES FONCTIONS DISPONIBLES .....	31
II.3.1. CONVENTIONS DE REPRESENTATION DES ELEMENTS DE LA SYNTAXE .....	31
II.3.2. DESCRIPTION DES COMPOSANTS CARACTERISTIQUES DU PHENOMENE .....	33
II.3.2.1. Les entrées du module de création du dic- tionnaire .....	34
. les indices .....	34
. les opérandes .....	35
II.3.2.2. La syntaxe des entrées du module .....	35
. forme générale .....	35
. description d'un objet .....	37
. remarques et restrictions .....	41
II.3.2.3. Les informations fournies par le module .....	41
. les informations apparentes .....	41
. les informations transparentes .....	42
II.3.3. RESERVATION ET STRUCTURATION DE LA "BASE DES OPERANDES" .....	45
II.3.3.1. Evaluation et réservation de l'espace physique .....	45
II.3.3.2. Structuration de l'espace acquis .....	46
II.3.3.3. Création du lien entre la définition d'une variable et ses réalisations .....	47
II.3.3.4. Les entrées du module .....	48
II.3.3.5. Les sorties du module .....	48

II.3.4. DESCRIPTION DES RELATIONS D'UN MODELE .....	49
II.3.4.1. Le module de macrogénération .....	51
II.3.4.2. Les entrées du module de macrogénération .....	52
. forme générale .....	53
. description d'une relation .....	54
. le corps de relation .....	55
- l'affectation L.P.P. ....	56
- l'instruction conditionnelle L.P.P. ....	57
- l'instruction composite L.P.P. ....	59
- l'itération L.P.P. ....	60
- l'instruction de contrôle L.P.P. ....	63
- une expression L.P.P. ....	66
- une constante L.P.P. ....	68
- une liste d'expressions d'indices .....	69
- remarques et restrictions d'utilisation des expressions L.P.P. ....	69
II.3.4.3. Les informations fournies par le module .....	70
. les informations apparentes .....	70
. les informations transparentes .....	71
- le fichier-pivot .....	71
- le descripteur du fichier-pivot .....	72
- le fichier des entrées du module de génération du graphe d'évaluation des relations d'un modèle .....	73
- le fichier accès .....	74
II.3.5. GENERATION DU GRAPHE D'EVALUATION DES RELATIONS ....	75
II.3.5.1. Formalisation du problème .....	76
. les objets nécessaires à la construction du graphe .....	76
. les notations : définition des ensembles et des fonctions utilisées .....	78
. les règles générales de construction du graphe .	80
- définition du vecteur de connaissance .....	80
- l'insertion de la relation r .....	80
- les actions .....	82
- les primitives de construction du graphe .....	84
- détermination de l'ordre d'évaluation des relations .....	84



II.3.5.2. Les entrées du module de génération du graphe .....	87
II.3.5.3. Les informations délivrées par le module .....	88
. les informations apparentes .....	88
. les informations transparentes .....	89
- la matrice d'incidence du problème par rapport à ses opérandes .....	89
- la matrice d'incidence du problème par rapport à ses relations .....	89
II.3.6. CONTROLE DE LA COHERENCE "DESCRIPTION/UTILISATION" DES VARIABLES D'UN MODELE .....	91
II.3.6.1. Les entrées du module de contrôle de la cohérence.	91
II.3.6.2. Les incohérences probables détectées .....	92
. les erreurs fatales .....	92
. les ambiguïtés pouvant menées à des erreurs graves .....	94
. les "warnings" .....	95
. les redondances d'informations .....	96
II.3.6.3. Les informations fournies par le module .....	97
II.3.7. GENERATION DE LA FORME EXECUTABLE D'UN MODELE .....	99
II.3.7.1. Les entrées du module de génération de la forme exécutable .....	99
II.3.7.2. Les informations produites par le module .....	101
. la forme exécutable du modèle .....	101
. les erreurs détectées par le module .....	101
II.3.8. INITIALISATION DE LA "BASE DES OPERANDES" D'UN MODELE .....	103
II.3.8.1. Le module d'initialisation de la base des opérandes : INITBD .....	104
II.3.8.2. La syntaxe des entrées-utilisateurs .....	106
. la forme générale .....	106
. l'identificateur de l'opérande à initialiser ...	106
. les réalisations de l'opérande .....	107
. remarques à propos des entrées-utilisateur .....	108

II.3.8.3. Les autres entrées du module .....	109
II.3.8.4. Les informations fournies par le module .....	110
. les informations visualisées .....	110
. les informations transparentes .....	111
II.3.9. RESOLUTION NUMERIQUE D'UN MODELE .....	113
II.3.9.1. Les entrées du module RESNUM .....	113
II.3.9.2. Les informations délivrées par le module .....	114
II.3.10. INTERROGATION DE LA "BASE DES OPERANDES" D'UN MODELE .....	117
II.3.10.1. Les entrées du module INTBOP .....	117
. la syntaxe du langage d'interrogation .....	117
. les entrées qui ne nécessitent pas l'inter- vention de l'utilisateur .....	119
II.3.10.2. Les informations délivrées par le module .....	119
. les éléments statistiques .....	120
. la trace de la description des requêtes d'interrogation .....	120
. les résultats de la simulation .....	120
II.3.10.3. Remarques et restrictions à propos des va- riables interrogées .....	122
 <u>CHAPITRE III : RÉALISATION PROGRAMMÉE</u> .....	 123
III.1. TECHNIQUES UTILISÉES .....	125
III.2. RÈGLES DE CONSTITUTION D'UN MODULE S.A.M.S.S. ....	126
III.2.1. TABLEAU DE CONSTITUTION DES MODULES S.A.M.S.S. ....	127
III.3. UTILISATION DE LA "CLASS" DANS S.A.M.S.S. ....	128
III.3.1. TECHNIQUES DE GESTION DE LA CLASS .....	128
III.3.2. LIAISON PASCAL-SYSTEME D'EXPLOITATION .....	129
III.3.3. IMPLEMENTATION D'UN MODULE EN MEMOIRE .....	129
III.4. ENVIRONNEMENT S.A.M.S.S. ....	130

III.5. LA PHASE DE DÉFINITION COHÉRENTE D'UN MODÈLE .....	131
III.5.1. CREDIC .....	132
III.5.2. FMATBD .....	132
III.5.3. MACROG .....	133
III.5.4. GENGRA .....	133bis
III.5.5. CONTRO .....	134
III.5.6. REORDO .....	134
III.6. LA PHASE D'EXÉCUTION D'UN MODÈLE .....	137
III.6.1. INITBD .....	138
III.6.2. RESNUM .....	138
III.6.3. INTBOP .....	139
III.7. CARACTÉRISTIQUES DES FICHIERS GÉRÉS PAR S.A.M.S.S.,	141
III.7.1. INPUT .....	142
III.7.2. SAVECLAS .....	143
III.7.3. BASOPMOD .....	144
III.7.4. FACCES .....	145
III.7.5. FPIV .....	146
III.7.6. SAVREL2 .....	147
III.7.7. FGRAPHE .....	148
III.7.8. FSREL1 .....	149
III.7.9. FPIVREOR .....	150
III.7.10. FIMPR .....	151
III.7.11. FTRACE .....	152
III.7.12. OUTPUT .....	153
III.8. DESCRIPTION ALGORITHMIQUE : L'EXEMPLE DU MODULE GENGRA .....	155
III.9. TAILLE DES MODULES DE S.A.M.S.S. ....	159

<u>CHAPITRE IV : UN EXEMPLE D'UTILISATION</u> .....	161
IV.1. PRÉSENTATION GÉNÉRALE DU MODÈLE .....	163
IV.2. CONSTRUCTION DU MODÈLE .....	164
IV.2.1. LE PROBLEME VU PAR L'ECONOMISTE .....	164
IV.2.2. SA TRADUCTION EN LANGAGE S.A.M.S.S. ....	166
IV.2.2.1. Constitution des textes utilisateurs .....	167
. description des composants caractéris-	
tiques du phénomène .....	167
. description des relations entre composants ..	168
. description des initialisations de certains	
composants .....	169
. description des requêtes d'interrogation de	
la base des opérandes .....	171
IV.2.3. AUTOMATISATION PROPREMENT DITE .....	172
IV.3. INTERPRÉTATION DES RÉSULTATS - INTÉRÊTS DE LA	
SIMULATION .....	183
IV.4. MISE EN OEUVRE TECHNIQUE DU LOGICIEL S.A.M.S.S. ...	185
IV.4.1. GENERALITES .....	185
IV.4.2. PRINCIPES DE FONCTIONNEMENT .....	187
IV.4.3. LES POSSIBILITES DE REPRISES .....	189
<u>CONCLUSION</u> .....	191

ANNEXES : Syntaxes des différents textes de description

BIBLIOGRAPHIE

## INTRODUCTION

Jusqu'à une date relativement récente (1970), les techniques de SIMULATION étaient encore mal connues dans le domaine des Sciences Sociales. Si les termes de MODELISATION et de Simulation occupaient une place de choix dans la littérature socio-économique depuis 1930 (date d'apparition de la théorie du système ouvert [1]), l'absence de recherches pluridisciplinaires rassemblant des économistes, des mathématiciens et des informaticiens, condamnait les premiers à l'utilisation brutale de packages de recherche opérationnelle ou à l'écriture de programmes spécifiques. Or, d'une part ces packages étaient mal adaptés à la spécificité des problèmes qu'avaient à résoudre les économistes, d'autre part la transcription en langage de programmation classique des concepts économiques n'était pas chose aisée, (elle en fige la structure), enfin, la maîtrise de l'une et/ou l'autre de ces techniques exigeait un temps d'apprentissage non négligeable, ce qui en a longtemps contrarié l'utilisation et le développement. En apportant d'une part, des langages où il est possible d'exprimer les spécificités des problèmes étudiés en Sciences Sociales et, d'autre part, les moyens de simuler les modèles ainsi définis, les techniques informatiques en ont facilité l'accès.

En abordant SAMSS, notre propos n'était pas de définir Le logiciel idéal destiné à décrire et à automatiser les modèles socio-économiques conçus selon les principes de la dynamique de système [2], mais plutôt de réaliser un outil qui aiderait la Commission des Communautés Européennes (Direction Générale XII : Sciences, Recherches et Développement) à automatiser le "Modèle de Prévission à long terme de la Demande d'Energie : MEDEE", développé à l'Institut Economique et Juridique de l'Energie (IEJE : Equipe de Recherche n° 94 du CNRS).

Avant d'exposer en détail le fonctionnement du logiciel (ce qui fera l'objet du chapitre 2), il nous semble important de présenter rapidement le cadre initial de sa réalisation : l'Analyse de Système en Prospective. Nous préciserons alors la place de SAMSS dans ce processus. Dans un deuxième temps nous aborderons chacune des

fonctions présentes dans le logiciel SAMSS, ainsi que les techniques de réalisation. L'analyse d'un exemple simple et de son traitement via SAMSS nous permettra ensuite d'évoquer la mise en oeuvre du logiciel.

En guise de conclusion, nous nous pencherons sur les outils qu'il serait possible d'intégrer au logiciel SAMSS afin d'en étendre le champ d'application. Nous mettrons alors en valeur les originalités de SAMSS par rapport à d'autres logiciels de la même famille.



## CHAPITRE I : LE CADRE THÉORIQUE DU DÉVELOPPEMENT DE S.A.M.S.S.

I.1. L'ANALYSE DE SYSTEME

I.2. MODELISATION ET SIMULATION

I.3. LA MODELISATION DE LA DEMANDE D'ENERGIE

I.4. OBJECTIFS DE L'AUTOMATISATION D'UN MODELE :  
LA PLACE DE S.A.M.S.S.



Choisie par Yves BAREL (souvent cité comme référence en la matière) pour définir le travail qu'il réalisa en 1971 pour la D.A.T.A.R., l'expression "Prospective et Analyse de Systèmes" traduit bien l'indissociabilité de ces deux concepts dans le domaine des Sciences Sociales.

L'analyse de Système est généralement présentée comme une démarche intellectuelle, un esprit de travail, plutôt que comme une méthode. Son originalité réside dans le fait que les chercheurs qui s'en réclament n'isolent pas le problème étudié de son contexte originel, c'est-à-dire qu'il s'agit d'une approche intégrative et non fragmentée.

Dans le cadre de notre travail d'"Aide à l'Automatisation de Modèles en Sciences Sociales", la définition qui suit nous semble digne d'intérêt. Il s'agit de la définition, parue dans les actes du Congrès A.F.C.E.T. : "Modélisation et Maîtrise des Systèmes" (novembre 1977). Le Président du Comité de Programme, J.L. LEMOIGNE, y présente la Systémique ou Analyse de Système comme "la Science de la Modélisation pour la maîtrise par l'action, de l'intervention des hommes dans et sur les systèmes techniques économiques et sociaux qu'ils conçoivent, qu'ils animent, qu'ils transforment". En effet, les modèles de prévision classique ayant révélé leur inaptitude à représenter des phénomènes où les structures ne sont pas stables et où tout agit sur tout, l'Analyse de Système, qui se propose de prendre en considération toutes ces interdépendances, est de plus en plus appréciée comme élément d'une méthode "Prospective" dans ce sens qu'elle permet d'anticiper, de décider, de connaître et de comprendre.

### I.1. L'ANALYSE DE SYSTÈME

L'Analyse de Système semble particulièrement bien adaptée à l'étude des phénomènes sociaux. En effet tout le monde s'accorde pour affirmer qu'un système social ne peut pas être appréhendé comme une "collection d'objets liés entre eux, que l'on isole artificiellement du reste du monde" [ 3].

Il existe deux conceptions différentes de l'analyse de système. Yves BAREL les présente toutes les deux [3] en qualifiant l'une de "cognitive" et l'autre de "décisionnelle". Nous ne prétendons pas faire ici le tour de la question, ni même discuter des mérites comparés de l'une et l'autre. Nous essayons simplement de caractériser chacune de ces grandes familles.

### I.1.1. L'ANALYSE COGNITIVE

Il s'agit d'une approche méthodologique destinée à étudier, à expliquer et à identifier des problèmes complexes. Les économistes qui se rangent sous cette barrière cherchent, non pas à "définir les objectifs du système", mais plutôt à "comprendre le système en fonction de ses objectifs". L'analyse des relations qui existent entre les objets manipulés par le système est pour eux, primordial. On peut donc qualifier l'approche cognitive d'approche relationnelle. C'est un "outil de connaissance et d'interprétation de la réalité sociale".

### I.1.2. L'ANALYSE DECISIONNELLE

Conçue comme un instrument de préparation et de formalisation de la décision, elle s'oppose à la première dans ce sens qu'elle s'appuie sur une théorie pour aboutir à l'élaboration d'un modèle qui est à même de prévoir le comportement du système. C'est plutôt de cette deuxième approche que relève la démarche adoptée par Bertrand CHATEAU et Bruno LAPILLONNE pour "analyser et comprendre les mécanismes de formation et de croissance de la demande d'énergie dans un système sociétal", et qui les a conduit à l'élaboration d'un "modèle de prévision à long terme de la demande d'énergie : MEDEE" dont le logiciel S.A.M.S.S. devait être le support informatique.

## I.2. MODÉLISATION ET SIMULATION

Face à l'accroissement tant en taille qu'en complexité, des problèmes auxquels se trouvent confrontés les industries, les gouvernements et les sociétés en général, il a été nécessaire de recourir à l'utilisation de techniques qui permettraient de les résoudre. Parmi ces techniques, il en existe deux auxquelles nous nous intéresserons particulièrement : la MODELISATION et la SIMULATION.

### I.2.1. LA MODELISATION

Celle-ci peut être considérée comme un outil de description du phénomène étudié, comme une abstraction de ce phénomène. Vu sous l'angle de l'Analyse de Système, c'est donc la représentation d'"une entité complexe (eu égard à certaines finalités) traitée comme une totalité organisée, formée d'éléments et de relations entre ceux-ci, les uns et les autres étant différenciés et définis en fonction de la place qu'ils occupent dans cette totalité, et cela, de telle sorte que son identité soit maintenue face à certaines évolutions" [4].

A partir de concepts théoriques, de méthodes et de langages, on souhaite parvenir à une formulation rapide et correcte du problème. Cette formulation sera alors concrétisée par :

- la définition des composants caractéristiques du problème,
- un ou plusieurs blocs d'équations, éventuellement imbriqués les uns dans les autres, qui traduisent les rapports qui existent entre ces composants, et leur évolution dans le temps.

### I.2.2. LA SIMULATION

Il s'agit d'un ensemble d'outils qui permettent de résoudre numériquement le problème mathématique qui représente le phénomène tel qu'il a été décrit précédemment. Mais la Simulation, c'est aussi un ensemble d'outils destinés à construire, à tester et à valider un modèle.

Ceci nous paraît d'autant plus vrai dans le domaine des Sciences Sociales, qu'un modèle n'est jamais neutre. Il comporte en effet un certain nombre d'objectifs implicites, et la confrontation des résultats produits, avec la réalité (même si celle-ci est perçue subjectivement) est alors nécessaire pour que ces résultats puissent être exploités en "Prospective".

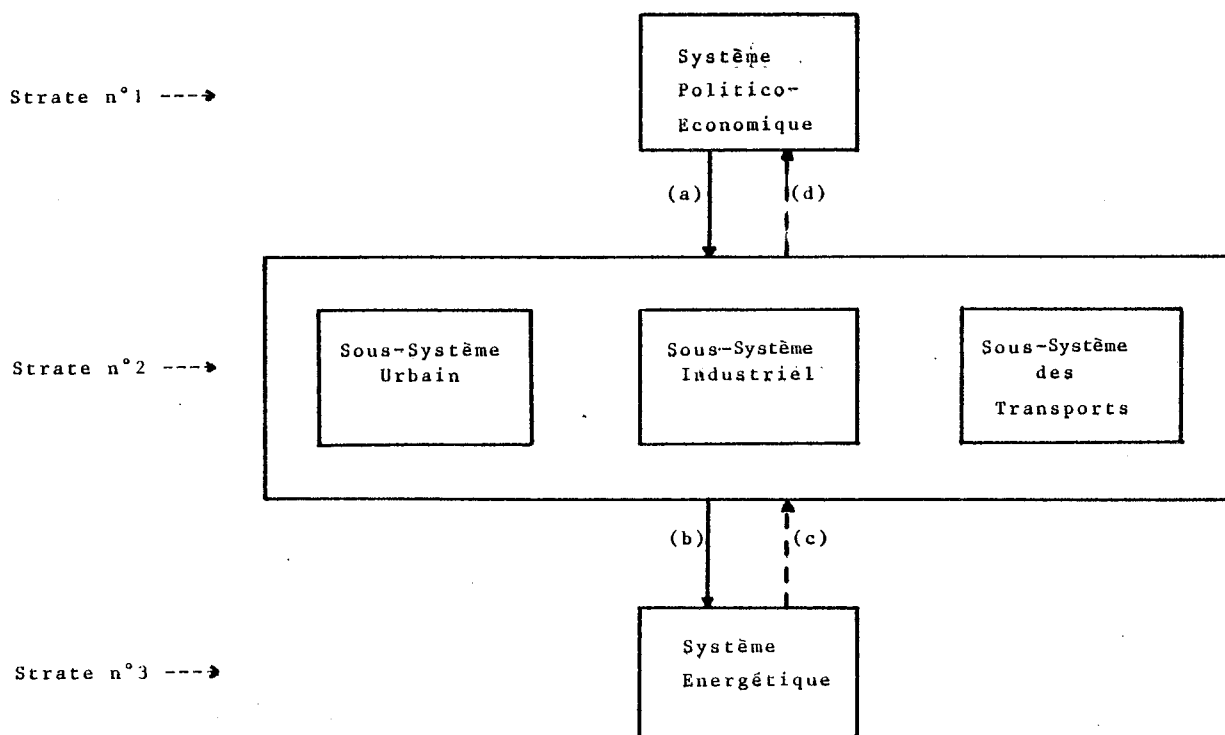
L'existence du rapport étroit qui lie ces deux techniques que sont la Modélisation et la Simulation explique que l'une et l'autre soient fréquemment confondues, ou utilisées dans des expressions impropres comme "Modèle de Simulation" ...

### I.3. LA MODÉLISATION DE LA DEMANDE D'ÉNERGIE : Approche Structurelle

Les outils informatiques à réaliser devaient servir de support à l'automatisation du Modèle de Prédiction à Long Terme de la Demande d'Énergie. Nous présentons ici quelques unes des caractéristiques structurelles de ce modèle.

Le système général de la Demande d'Énergie s'inscrit dans le cadre conceptuel de l'Analyse de Systèmes, et plus particulièrement dans celui des "Systèmes Hiérarchisés à plusieurs niveaux" [16]. Il peut s'analyser comme un arrangement vertical de sous-systèmes formant une hiérarchie dans laquelle un sous-système constitue une "strate" ou niveau. Une strate est en général directement influencée par celle qui la précède dans la hiérarchie, et influence celles du niveau suivant. Chaque strate intervient sur le niveau qu'elle commande, et répond éventuellement au niveau qui la commande.

Le système de la Demande d'Énergie peut se décomposer en trois strates

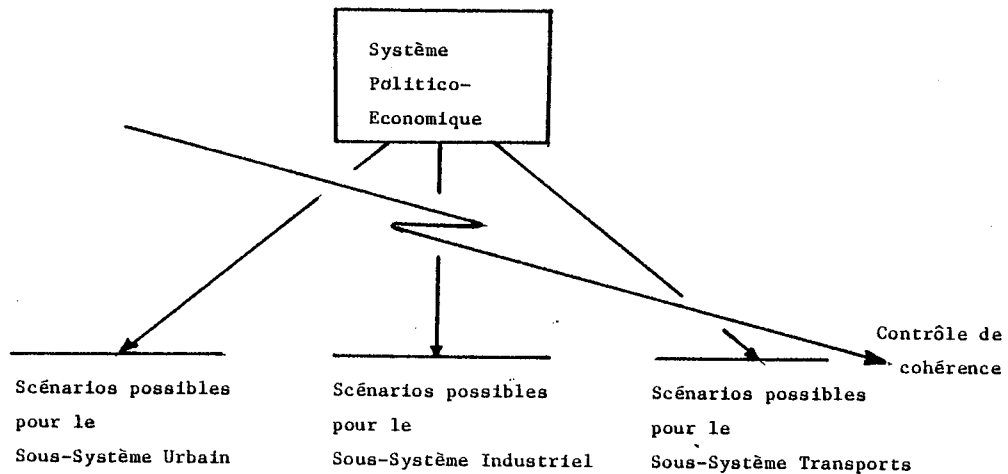


pour lesquelles les relations d'influence ont la signification suivante :

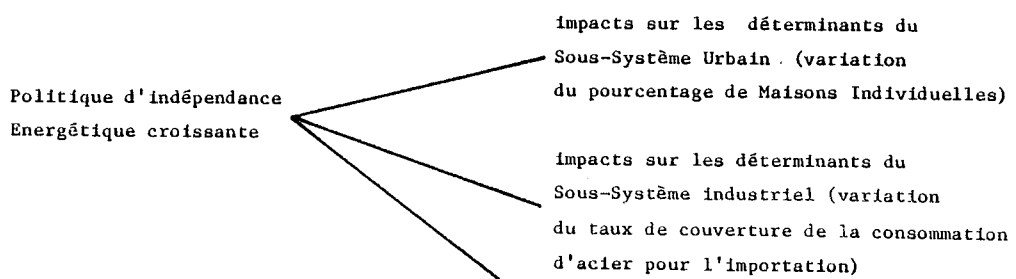
- . (a) indique la prépondérance du domaine politico-économique sur les sous-systèmes urbain, industriel et des transports.
- . (b) indique que les sous-systèmes urbain, industriel et des transports, déterminent la demande d'énergie, paramètre qui sera fourni au système énergétique.
- . (c) précise l'offre d'énergie.
- . (d) peut être considérée comme le résultat agrégé d'un scénario donné (variation de certains déterminants, ...).

### I.3.1. LA STRATE N° 1 : système politico-économique

Elle peut être assimilée à une hiérarchie décisionnelle à un niveau [16]. Elle permet de déterminer et de fournir des scénarios à la strate immédiatement inférieure. La principale difficulté, pour les économistes, semble être le contrôle de la cohérence des scénarios appliqué à chaque sous-système de la strate n° 2.



#### Exemple de scénario:



Remarque : Dans le cadre d'une politique et d'une économie décentralisée, la strate n° 1 pourrait être assimilée à une hiérarchie décisionnelle à deux niveaux, le premier étant formé du système politico-économique national (ou global) et le second, du système politico-économique régional.

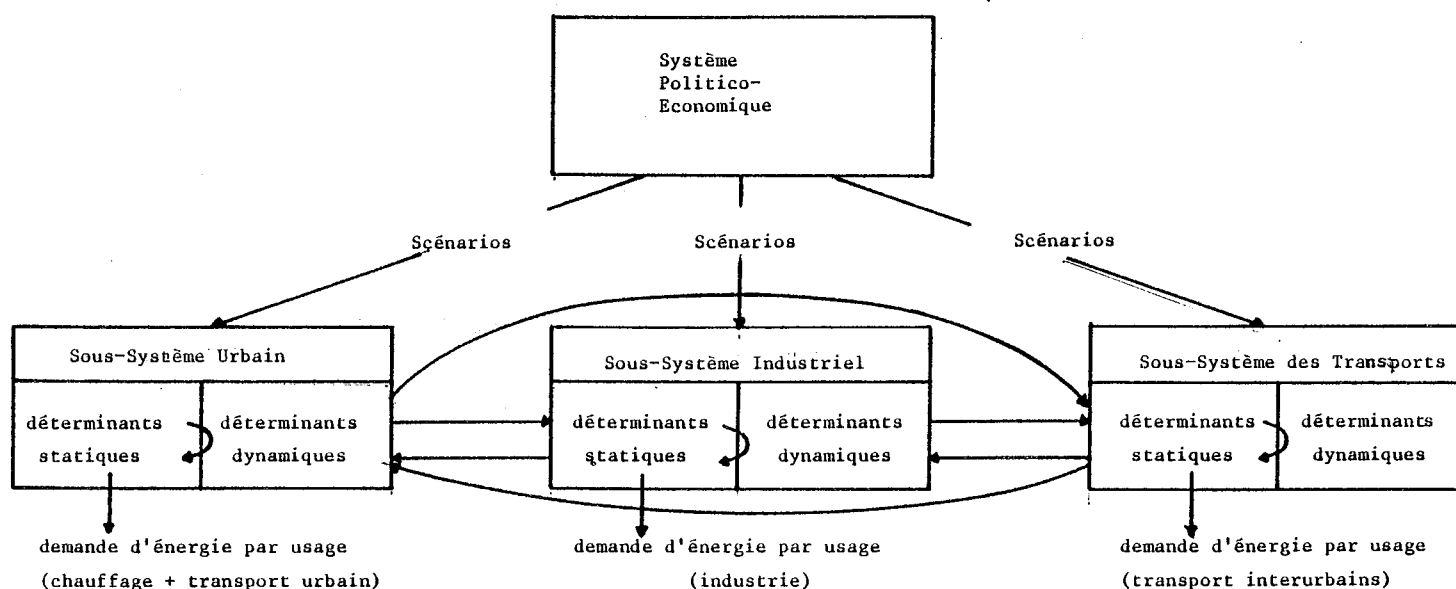
### I.3.2. LA STRATE N° 2 : sous-système urbain - sous-système industriel - sous-système des transports

C'est un ensemble de sous-systèmes que l'on peut caractériser en affirmant :

- . qu'ils évoluent dynamiquement à partir de conditions initiales
- . qu'ils sont interactifs les uns avec les autres.

L'aspect dynamique : Il implique la notion de période. Au cours d'une période, le sous-système évolue essentiellement grâce aux déterminants dynamiques. A la fin de la période, l'état de l'évolution est constaté (aspect statique) et sert de base à la période suivante, etc.

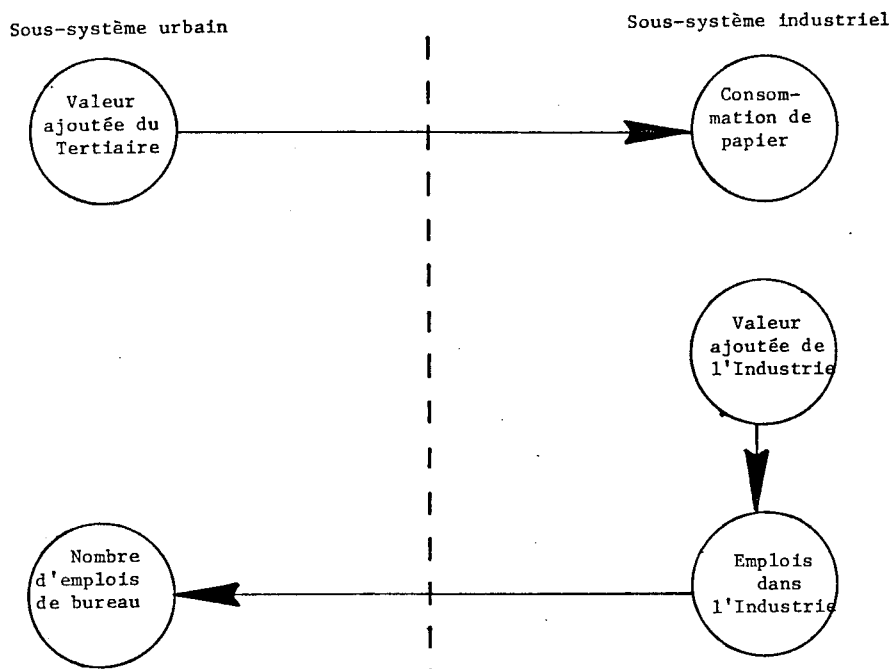
L'interactivité entre les différents sous-systèmes peut être prise en compte explicitement, de façon exogène pour chacun d'entre eux.



Exemple :. L'aspect statique et dynamique du sous-système urbain

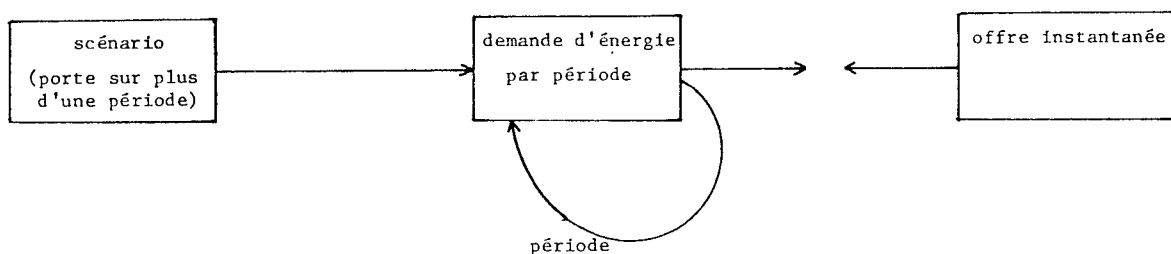
Le besoin en chaleur est déterminé de façon instantanée à partir des besoins en température de la durée du chauffage, et des paramètres physiques de la construction qui sont des déterminants statiques (ce qui ne signifie pas pour autant qu'ils sont constants).

L'évolution du type de logement, et de l'installation de chauffage, éléments qui vont influencer le besoin en chaleur de la période suivante, sont déterminées de façon dynamique.

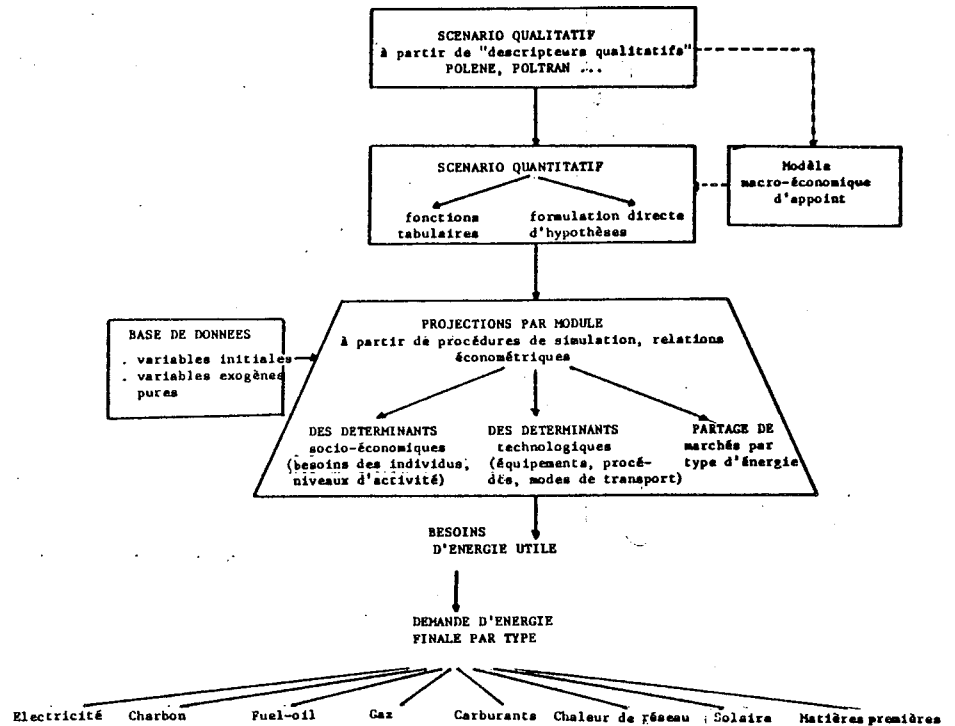
. L'interaction entre sous-système urbain et sous-système industrielI.3.3. LA STRATE N° 3 : système énergétique

Il s'agit de l'ensemble des agrégations qui permettent de déterminer l'approvisionnement énergétique.

Le fonctionnement du système de la demande d'énergie est théoriquement le suivant :



### I.3.4. SCHEMA DE MISE EN OEUVRE DE L'APPROCHE MEDEE POUR MODELISER LA DEMANDE D'ENERGIE



### I.4. LES OBJECTIFS DE L'AUTOMATISATION D'UN MODÈLE : LA PLACE DE S.A.M.S.S.

Si les progrès réalisés dans des domaines comme les statistiques, l'analyse de données, la théorie des mutations et les techniques informatiques sont des facteurs déterminants du développement important qu'a connu la modélisation des phénomènes socio-économiques, il n'est pas inutile de rappeler que la modélisation d'un système puis sa simulation ne doivent pas rester des exercices de style. Ces techniques s'inscrivent dans un processus qui doit permettre :

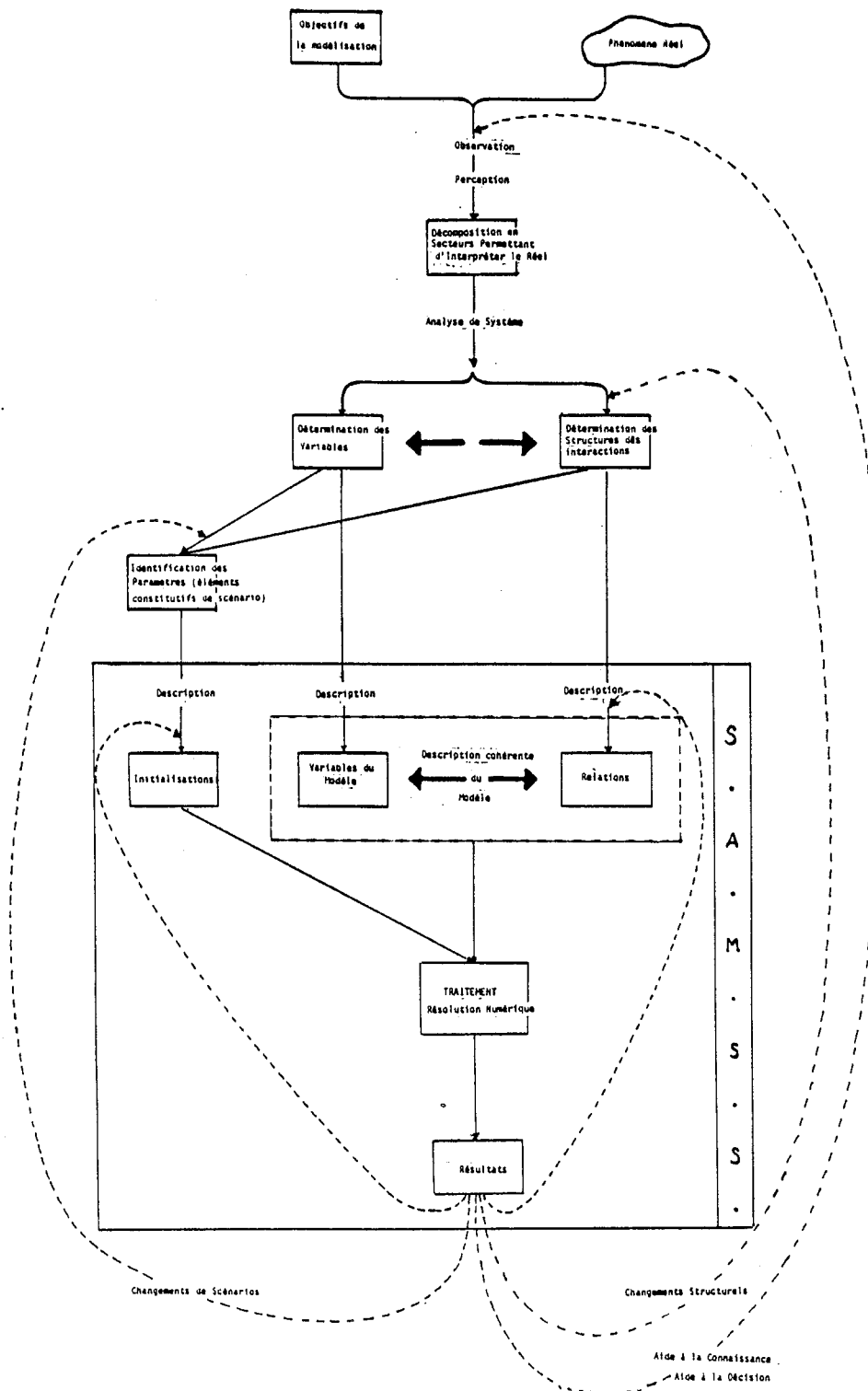
- l'exploration du système étudié,
- l'identification des éléments critiques (paramètres) de ce système,
- la synthétisation et l'évaluation de ses solutions,
- l'aide à la décision, ...

Mais attention, les modèles que l'on automatise ne sont pas des machines à fabriquer des décisions. Il ne faut pas perdre de vue le



fait qu'une analyse scientifique ne se réduit pas à la résolution numérique d'équations quantifiées qui seraient considérées comme intangibles.

Nous essayons de faire apparaître sur la figure 1, le parallélisme qui existe entre le processus d'analyse auquel se livrent les économistes et les fonctions qui sont disponibles dans le logiciel SAMSS.



On constatera ainsi, que ce processus d'analyse ne se limite pas à la modélisation et à la simulation. Il commence bien avant, et nous voulons insister sur le fait que ces techniques n'en sont que des éléments, et que toute "analyse prospective" doit faire intervenir une "analyse historique inverse" ou analyse a posteriori.

Il est assez malaisé de définir par des frontières nettes les différentes étapes de ce processus, et on aura beau jeu de nous faire remarquer que ni l'étape "modélisation" ni celle de "simulation" ne figurent sur ce schéma ; disons simplement ceci :

Après avoir posé clairement le problème qu'on se propose d'étudier, et après avoir défini les objectifs à atteindre, il est nécessaire d'observer "LE REEL". En fait, parler "du Réel" n'a pas tellement de sens ici puisque cela supposerait que l'observateur puisse être objectif, ce que nous avons fermement nié plus haut. D'autre part le modélisateur ne s'intéresse pas à la totalité de "ce Réel", ce qui serait bien trop complexe. En fonction des objectifs qu'il s'est assigné, il perçoit le phénomène observé d'une certaine façon, et pour parvenir à interpréter ce phénomène, il est amené à le décomposer en secteurs, de taille plus facilement maîtrisable. Ceci étant fait, aucune voie de recherche ne doit être exclue a priori, pour identifier, de la manière la plus exhaustive possible, les variables qui caractérisent le phénomène. Seules les limites de notre imagination doivent pouvoir nous arrêter, et l'approche "Analyse de Système" permettra alors de maintenir à cette quête d'informations, l'homogénéité requise, dans la mesure où, dans une vision systémique du monde, une variable n'existe que par ses relations. Après avoir isolé parmi les variables, celles qu'il sera nécessaire de quantifier (éléments constitutifs de scénarios) on pourra passer à la phase d'automatisation du modèle issu de cette première partie du processus d'analyse.

Cette automatisation comportera deux grandes phases. La première consistera à "formuler" dans un langage approprié, le problème que l'on souhaite résoudre, la Résolution Numérique et l'édition des résultats achèvent ce processus d'automatisation.

L'analyse des résultats fournis ne doit pas être considérée comme une fin en soi. En effet en admettant que les résultats correspondent à l'attente du modélisateur (ce qui ne sera pas souvent le cas), il semble important de procéder à d'autres essais, les uns portant uniquement sur des modifications de données (changements de scénarios), les autres sur des modifications structurelles, le tout permettant d'une part de "caler" le modèle, et d'autre part de le valider.

Enfin si on admet qu'"il n'y a pas de données, de connaissances, d'observation, de sciences qui ne modifient la réalité" [5], l'un des objectifs de l'analyse d'un système étant de le mieux connaître (aide à la connaissance, aide à la décision), les résultats de l'automatisation de celui-ci, sous sa forme modélisée sont à même de remettre en cause l'observation du Réel, voire même ses objectifs, et d'assurer ainsi la vie du système. Dans ce contexte, comment discerner ce qui appartient à la modélisation, ce qui est simulation, ce qui est analyse ? Si la modélisation est prise au sens de la description du phénomène étudié, alors la phase I (Description Cohérente du Modèle) assure cette fonction. Si, par contre, on lui fait englober toute une partie de la conceptualisation de la méthodologie d'élaboration du modèle, alors cette phase I est insuffisante. De la même manière, si la Simulation n'est interprétée que comme la Résolution Numérique du modèle, alors la phase II (Traitement - Edition des Résultats) y correspond, alors que si la Simulation s'entend comme outil de test, de validation et de construction de modèle, la phase II n'y suffit plus.

En conclusion, Analyse, Modélisation et Simulation sont les trois mots-clés qui caractérisent le cadre dans lequel s'est développé le logiciel S.A.M.S.S., support informatique pour l'automatisation de modèles en Sciences Sociales.

## CHAPITRE II : LE LOGICIEL S.A.M.S.S.

II.1. PRESENTATION GENERALE

II.2. CARACTERISTIQUES GENERALES DU LOGICIEL

II.3. LES FONCTIONS DISPONIBLES

## II.1. PRÉSENTATION GÉNÉRALE

Lorsque, en 1977, la Direction "Sciences, Recherches et Développement" de la Commission des Communautés Européennes, où l'axe "Energy Modeling" existe déjà, décide d'exploiter les travaux effectués au sein de l'I.E.J.E. par B. CHATEAU et B. LAPILLONNE, elle s'adresse au Département Informatique de l'Institut de Recherche Economique et de Planification (devenu depuis le C.R.I.S.S\*), et le charge de l'étude puis de la réalisation d'un ensemble d'outils informatiques destinés à automatiser le modèle technico-économique MEDEE-3.

Après avoir analysé la thèse qui définit le modèle conceptuel, et en tenant compte du fait que - bien que défini dans un premier temps pour le cas particulier de la France - celui-ci devait pouvoir être adapté pour prendre en compte les spécificités d'autres pays, nous avons proposé d'élargir le cadre de l'étude en réalisant un Système d'Aide à la Modélisation en Sciences Sociales : S.A.M.S.S. Les problèmes à résoudre pour automatiser le modèle MEDEE-3 étaient suffisamment nombreux, et d'ordre suffisamment généraux (au sens où ce sont des problèmes fréquents en modélisation) pour que nous ayons refusé de créer un logiciel destiné uniquement à simuler MEDEE-3.

Le logiciel S.A.M.S.S. met à la disposition de ses utilisateurs potentiels, un ensemble de fonctions qui permettent de définir les composants caractéristiques du phénomène modélisé, d'exprimer les rapports qui existent entre ces composants (relations), de gérer l'ensemble de valeurs qui seront exploitées pour initialiser la "Base des Opérandes" associée au modèle, de demander la résolution numérique du modèle et enfin d'éditer tout ou partie du contenu de la "Base des Opérandes" pour l'analyse des résultats.

La politique de la Direction "Sciences, Recherches et Développement" de la Commission des Communautés Européennes, en matière de "software", consiste à diffuser ceux-ci au maximum, et plus particu-

---

\* C.R.I.S.S. : *Centre de Recherche en Informatique appliquée aux Sciences Sociales.*

lièrement dans les centres nationaux avec lesquels existe une collaboration. Cela représentait donc pour nous, la nécessité de réaliser des outils qui soient aisément transportables. Le choix, au niveau de la programmation, d'un langage de haut niveau tel PASCAL (dont l'équipe a la maîtrise d'oeuvre sur des matériels aussi hétérogènes que IBM : série 360/370 ou 3033, ICL série 2900, SIEMENS série 7800, ou HONNEWELL BULL : HB68-Multics) a facilité notre tâche.

La réalisation du logiciel S.A.M.S.S. est le résultat d'un travail d'équipe qui représente un peu plus de 5 années/homme. Initialisé par une petite équipe de 2 personnes, le projet s'est ensuite adjoint les compétences de deux autres membres du CRISS, pour mieux cerner le problème, définir les concepts manipulés et concevoir l'architecture générale du logiciel. La réalisation du "prototype" [6] puis de la version automatisé (version actuelle) [7] a été confiée à une équipe plus restreinte.

Le travail a été réparti en fonction des domaines clés du système : Base des Equations, Génération et traitement de Graphe, Base de Données, Edition ...

L'utilisation progressive du logiciel S.A.M.S.S. - dans sa version "prototype" d'abord, dans sa "version automatisée" à partir d'avril 1980 - par les économistes de l'Institut Economique et Juridique de l'Energie, par les Experts de la Communauté Economique Européenne et par ses correspondants nationaux, a confronté l'équipe aux réalités parfois dures de la certification du logiciel, et de sa maintenance. Aussi, un important travail de réflexion et de réalisation a été accompli sur des thèmes comme la mise en oeuvre, la gestion du logiciel lui-même, la validation du logiciel, etc ..., préoccupations qui constituent l'un des axes de ce que l'on nomme aujourd'hui le "Génie Logiciel".

## II.2. CARACTÉRISTIQUES GÉNÉRALES DU LOGICIEL

### II.2.1. LES OBJECTIFS

La Modélisation et la Simulation en Sciences Sociales, et plus particulièrement dans le domaine socio-économique, sont des phénomènes très complexes. On pourra, à ce sujet, consulter les travaux de Stranch [8], Rechenman [9], et Barel [10]. Il s'agissait donc pour nous de définir et de réaliser un ensemble d'outils informatiques qui facilite la tâche des concepteurs de modèles, les aide à définir puis à simuler automatiquement leurs problèmes, en utilisant un formalisme qui ne soit pas trop éloigné de celui qu'ils adoptent couramment. Travaillant en étroite collaboration avec des économistes (notre équipe en compte 3), nous estimons en effet qu'il est inopportun d'ajouter des difficultés relevant exclusivement de l'informatique, à celles inhérentes à la définition des modèles socio-économiques.

Pour y parvenir, nous avons cherché à :

- minimiser les travaux d'analyse restant à la charge de l'utilisateur,
- simplifier l'écriture des applications par rapport aux programmations classiques,
- faciliter la prise en charge des applications par les utilisateurs eux-mêmes,
- libérer l'utilisateur de l'organisation des processus de traitement informatique.

La représentation d'un problème qu'il serait possible de traiter à l'aide du logiciel S.A.M.S.S. devait être essentiellement DESCRIPTIVE. La syntaxe préconisée devait permettre de spécifier les différents objets manipulés (indices, variables, relations ...) ainsi que les différents types d'action à entreprendre, sans se préoccuper de l'organisation des processus de traitement.

L'automatisation d'un modèle (construction, test, analyse, mise au point) est une opération longue, et les interactions homme-machine y sont très fréquentes. Le logiciel, support de cette automatisation devait donc être simple à utiliser, ce qui suppose l'existence d'un langage de commande approprié : étapes du travail de l'utilisateur bien définies, repérage à tout moment de ce qui a déjà été fait, ...

La modularité devait être importante car le logiciel est susceptible d'être modifié, amélioré. La transportabilité, comme nous l'avons vu plus haut, devait être assurée. Elle est d'autre part une nécessité en prospective en Sciences Sociales. En effet, la description d'un modèle, les expériences réalisées (simulation, tests de scénarios, analyse ...) doivent être répétables dans des lieux différents pour pouvoir être l'objet d'une critique incessante. Il était donc souhaitable que le logiciel d'automatisation puisse être installé sur des matériels informatiques différents mais de taille comparable.

L'une des originalités du logiciel S.A.M.S.S., en dehors de sa méthode de réalisation, devait donc résider dans ses modalités de mise en oeuvre.

### II.2.2. TYPES DE MODELES SUPPORTES

Les modèles dont le logiciel S.A.M.S.S. peut être le support informatique sont définis comme des ensembles d'EQUATIONS AUX DIFFERENCES (nous les appellerons RELATIONS). Dans le cas MEDEE-3, ces relations sont réparties en quatre grands groupes :

- . les équations de calcul de la demande d'énergie, qui se présentent sous la forme d'identités comptables
- . les relations de simulation technico-économique, qui sont, pour la plupart, linéaires, et permettent de faire évoluer les niveaux d'activités économiques d'une année de calcul sur l'autre



- . les relations de simulation des choix d'équipement énergétique, plus complexes et qui peuvent être formalisées de la façon suivante :

$$CF(h,k) = CI(h,k) * (\alpha + (1-\alpha) * P(h))$$

$$CFm(k) = \text{Minimum}(h) \text{ } CF(h,k)$$

$$h \text{ sera choisi si } CF(h,k) \leq CFm(k) * (1+\beta)$$

avec :

$CF(h,k)$  : coût fictif perçu par le décideur

$k$  pour l'équipement

$CI(h,k)$  : coût global implicite de l'équipement

$h$  pour décideur  $k$

$\alpha$  : pondération du critère économique

$P(h)$  : poids des externalités liés à l'équipement

$\beta$  : seuil de sensibilité

- . les relations de contrôle, dont la fonction est d'assurer la cohérence et la pertinence des données et des calculs.

La formulation des relations d'un modèle que l'on souhaite soumettre au logiciel S.A.M.S.S. doit donc permettre au minimum la prise en considération de ces 4 catégories, et par conséquent satisfaire aux contraintes suivantes :

- . Toute relation est soit de type calcul, soit de type contrôle.
- . Toute relation de type calcul doit comporter une partie gauche réduite à une variable, et une partie droite quelconque mettant en jeu, 0, 1, ... N variables (actuellement N=14).
- . Toute variable du modèle est une matrice d'ordre quelconque, la dimension temporelle (discrétisée et simulée) étant traitée de façon particulière.
- . La dimension temporelle (T) ne peut apparaître qu'en T (date de simulation courante) et/ou T-1 (date de simulation précédent la date de simulation courante), ce qui implique que la valeur maximum de la fonction retard correspond à une période.

- . Toute variable du modèle peut être fonction de la date de simulation, ou ne pas l'être.
- . L'expression d'évaluation d'une relation peut être simple ou complexe. Elle peut être linéaire ou non linéaire, conditionnelle ou inconditionnelle, itérative ou non itérative, et comporter une ou plusieurs variables intermédiaires.

Les modèles pris en compte par le logiciel S.A.M.S.S. se présentent donc sous une forme peu classique. Cependant, le logiciel impose actuellement une restriction importante. Le modèle ne doit pas posséder de variable de bouclage. En d'autres termes, la matrice d'incidence du problème à résoudre doit être triangularisable. Les économistes qui ont conçu MEDEE-3, ainsi qu'une part importante de ceux qui l'utilisent, estiment en effet que la notation d'équilibre en prospective à long terme n'a pas de sens et que l'on mesure bien mieux les écarts entre une situation réelle et une situation idéale, en utilisant des variables virtuelles.

### II.2.3. S.A.M.S.S. : UN OUTIL D'AIDE A L'ECRITURE ET A LA MAITRISE DES MODELES

Le problème posé, pour qu'il soit du ressort du logiciel SAMSS, consiste à résoudre numériquement, à partir de données plus ou moins brutes, un ensemble hiérarchisé d'équations aux différences, à présenter les résultats sous forme de tableaux permettant aux concepteurs de les analyser et d'en tirer les enseignements nécessaires à une meilleure connaissance du phénomène, et à la prise de décisions. Pour effectuer ce travail, il est nécessaire de fournir au logiciel SAMSS les quatre textes suivants :

- . Description des composants caractéristiques du phénomène modélisé.
- . Description des relations qui existent entre ces composants.
- . Description des réalisations connues (jeu de données) de certains de ces composants.
- . Description des requêtes d'interrogation de la "Base des Opérandes" du modèle.

En réalité, l'écriture de ces quatre textes ne se fait pas dans l'ordre indiqué ci-dessus, ni même de façon linéaire, mais plutôt par approximations successives, au prix de nombreux allers et retours entre ces différentes descriptions.

Le processus de construction d'un modèle implique, au départ, l'imperfection de sa représentation par rapport au phénomène réel observé. Cette inadéquation se retrouve par conséquent au niveau de sa description en *langage S.A.M.S.S.* Pour y remédier, en dehors de l'analyse syntaxique de la description des composants caractéristiques du modèle, et de celle des relations, nous avons introduit une fonction qui permet à l'utilisateur d'assurer leur cohérence : toute variable qui n'apparaît pas en partie gauche d'une relation doit être définie comme une donnée, ... etc. Le contrôle permet, entre autre, d'identifier les paramètres d'un modèle. Il contraint l'utilisateur à revenir sur la description de son modèle, soit pour s'assurer qu'il dispose des *données nécessaires*, soit pour s'apercevoir qu'il a omis de décrire une relation ... C'est en cela que le logiciel S.A.M.S.S. peut être considéré comme un outil d'aide à l'écriture, à la construction de modèle.

La description cohérente du modèle étant acquise, l'utilisateur devra fournir la description des réalisations connues de certains des composants caractéristiques du modèle. Là encore, le logiciel l'aidera en lui indiquant l'état de la "Base des Opérandes". Tant que tous les composants définis comme des données ne seront pas "initialisés", on pourra rejeter toute demande de résolution numérique.

Toute personne ou organisme qui conçoit un modèle décisionnel, le fait pour obtenir des informations qui l'aideront à intervenir sur le phénomène modélisé. Ainsi, toute connaissance nouvelle du phénomène peut être reliée aux connaissances antérieures, pour une meilleure compréhension de celui-ci, par voie de conséquence pour aider à la prise de décisions. On retrouve ce souci au niveau du logiciel puisqu'il permet, à partir des résultats de la résolution numérique, soit de modifier les données soit de modifier les relations, et ceci de façon simple. C'est dans ce sens que S.A.M.S.S. permet de mieux maîtriser (au sens de la connaissance) le phénomène étudié.

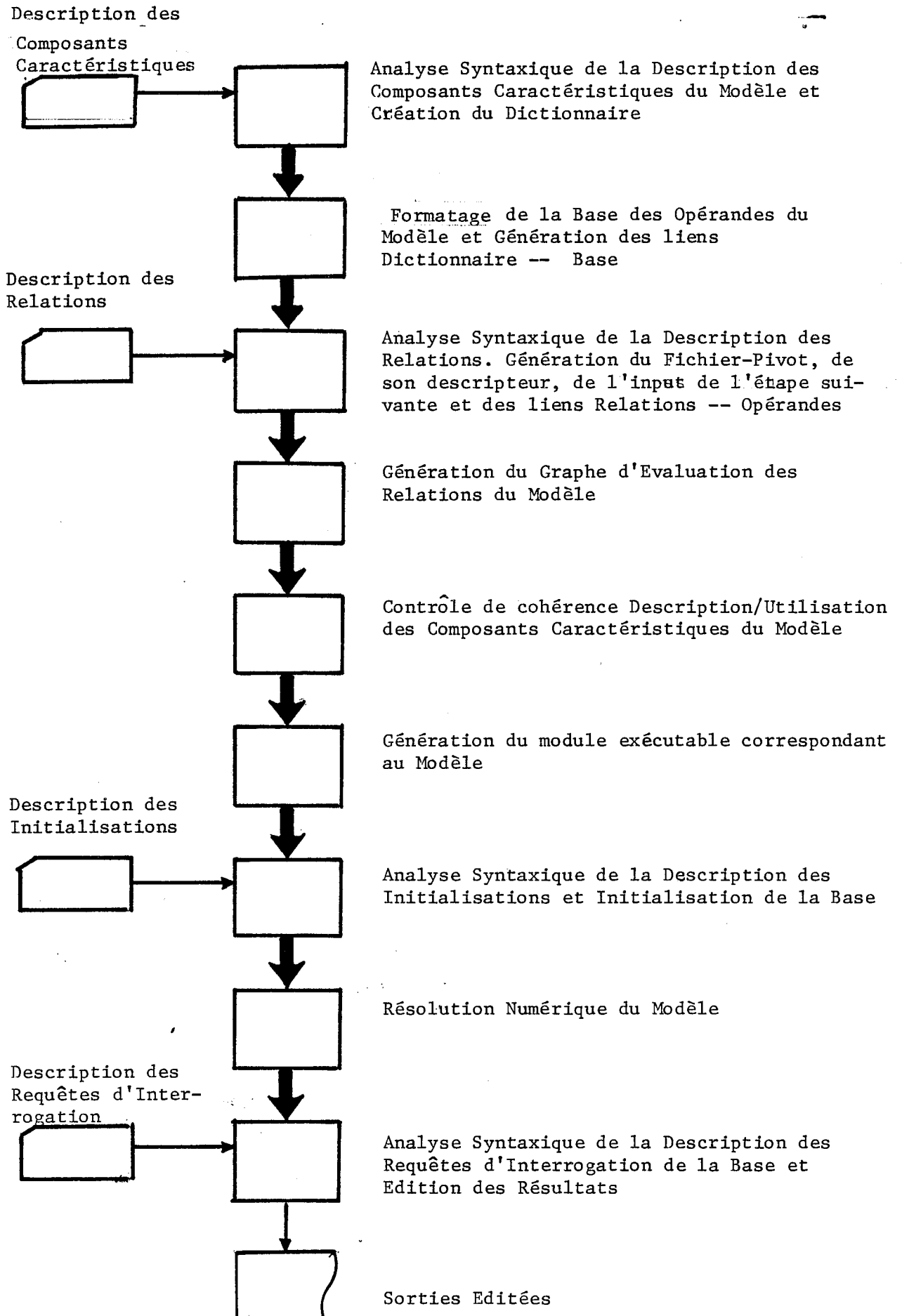


Fig. 2 - Les étapes d'une application SAMSS

#### II.2.4. LE FONCTIONNEMENT DU LOGICIEL

Le logiciel S.A.M.S.S. fonctionne par étapes successives (voir figure 2) qui exploitent tant les informations fournies par l'utilisateur, que celles qui en sont dérivées, générées par les différents modules du logiciel.

Dans un premier temps, le module de Création du Dictionnaire (CREDIC) du modèle, analyse le texte de la description des composants caractéristiques fourni par l'utilisateur. Si la syntaxe appropriée a été respectée, si on n'a rencontré aucune erreur, le module produit une description codée que nous appellerons *Dictionnaire du Modèle*.

A partir de certaines des informations mémorisées dans le *Dictionnaire du Modèle*, le module de formatage de la Base des Opérandes (FMATBD) réserve l'espace nécessaire au stockage de toutes les variables du modèle : la *Base des Opérandes du Modèle*, et génère les liens entre la définition des variables et leurs représentations dans la *Base* (il y a donc mise à jour du *Dictionnaire du Modèle*).

Le module de Macrogénération des Relations (MACROG) d'un modèle est alors chargé d'analyser la description des relations qui existent entre les composants caractéristiques, texte fourni par l'utilisateur. Si le langage d'expression des relations a été correctement utilisé, le module génère quatre types d'informations :

- . le fichier-pivot : texte PASCAL compilable des relations non réordonnées du modèle,
- . le descripteur de ce fichier-pivot,
- . le fichier des relations réduites à leurs opérandes,
- . le fichier de *Liaison* entre relations et ensemble des variables qu'elles utilisent.

A partir du fichier des relations réduites à leurs opérandes, le module de Génération du Graphe d'évaluation (GENGRA) des relations du modèle, analyse la faisabilité du problème. Si le modèle ne comporte pas

de variables de bouclage, le module génère l'ordre dans lequel les relations doivent être évaluées.

Avant de passer à la génération du "load-module" correspondant à la résolution numérique du modèle, on va comparer la nature des composants caractéristiques du phénomène telle qu'elle a été enregistrée dans le *Dictionnaire du Modèle*, et celle de ces mêmes composants telle que le logiciel l'a évaluée et mémorisée dans le "Graphe d'Evaluation des Relations".

Lorsque les contrôles de cohérence se révèlent satisfaisants, l'utilisateur peut demander la constitution du module exécutable correspondant au modèle. En exploitant les informations produites par les modules précédents, le module de Réordonnement des Relations (REORDO) produit le fichier "source du programme correspondant à la résolution numérique" qui sera ensuite compilé pour aboutir au module exécutable de *résolution numérique* du modèle.

L'étape suivante est chargée d'analyser la description des réalisations connues de certains composants caractéristiques du modèle. Si on a respecté la syntaxe de ce troisième texte utilisateur, l'initialisation de la *Base des Opérandes* (INITBD) du modèle est effectuée, et on contrôle que toute variable devant être initialisée (Donnée Initiale, Donnée de Scénario, Donnée Exogène) l'est réellement. L'utilisateur peut alors activer la résolution numérique (RESNUM) de son modèle en précisant les dates de début et de fin de simulation. Si l'exécution de ce module s'effectue sans incident, la *Base des Opérandes* du modèle contiendra les résultats de la simulation.

La description des requêtes d'interrogation de la *Base des Opérandes* du modèle, si elle ne comporte aucune erreur, permet alors, via le module d'interrogation de la *Base des Opérandes*, d'éditer le contenu (total ou partiel) de la *Base* pour analyse par l'utilisateur.

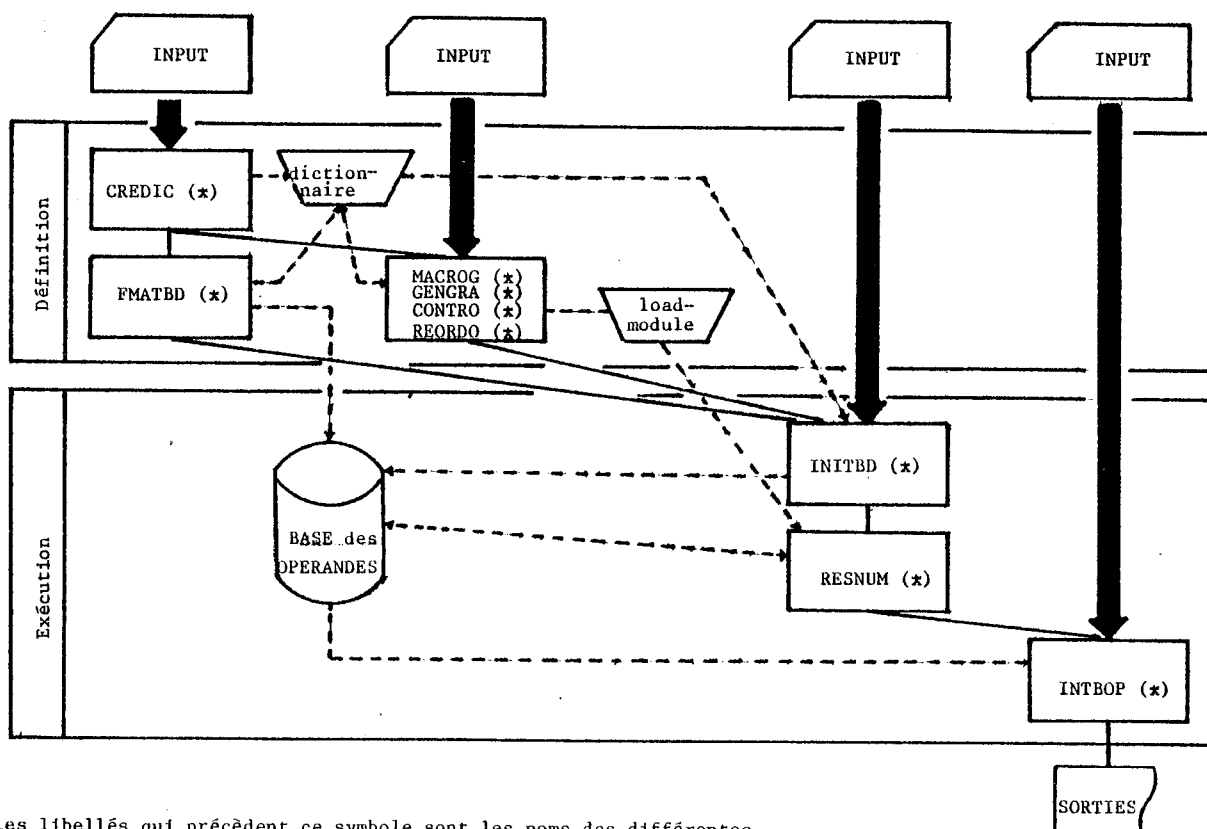
## II.2.5. ORGANISATION INFORMATIQUE DU LOGICIEL

Le système S.A.M.S.S. s'articule en deux grandes phases, une phase de définition du modèle (description des composants caractéristiques du phénomène, description des liens qui existent entre ces composants), et une phase d'exécution. La phase de définition d'un modèle va consister à :

- . analyser les textes de description,
- . générer la forme exécutable du modèle,
- . produire les fichiers nécessaires à l'exécution de l'application.

Dans la phase d'exécution on trouvera trois fonctions séparées :

- . initialisation des fichiers produits lors de la phase précédente,
- . résolution numérique du modèle,
- . édition des résultats.



\* les libellés qui précèdent ce symbole sont les noms des différentes fonctions du logiciel S.A.M.S.S.

## II.3. LES FONCTIONS DISPONIBLES

### II.3.1. CONVENTIONS DE REPRESENTATION DES ELEMENTS DE LA SYNTAXE

Il est bien connu que l'écriture d'un programme nécessite une certaine rigueur sous peine d'être inopérante. Les programmes S.A.M.S.S. (car tels peuvent être considérés les travaux demandés aux concepteurs de modèles), n'échappent pas à cette contrainte, compensée il est vrai par une relative simplicité des règles. Les quelques conventions qui suivent permettront de mieux préciser ces règles, et éviteront leur répétition à tous les stades de la présentation des fonctions activées par le logiciel S.A.M.S.S.

- MOT-CLE Les mots-clés du langage S.A.M.S.S. sont des mots dont la signification est fixée d'avance par et pour le logiciel. Ils doivent être écrits exactement comme nous l'indiquons et ne pas être utilisés avec une signification différente de celle qui leur est assignée. Dans toutes les descriptions de syntaxe S.A.M.S.S., nous utiliserons le symbole ○ pour les caractériser.

Exemple : INSERER

- Unité Syntaxique  
Complexe Tout langage de programmation peut se représenter sous forme d'un arbre syntaxique dont les feuilles seraient le vocabulaire du langage. Nous appellerons unité syntaxique simple, les éléments syntaxiques appartenant à ce vocabulaire : mots-clé, symboles ... (On les désigne souvent sous le terme d'unités lexicales). Une Unité Syntaxique Complexe se définit alors comme un regroupement d'unités syntaxiques simples ou d'un degré de complexité moindre. Ainsi, si nous considérons une unité syntaxique complexe, elle peut, par dérivations successives s'exprimer comme une suite d'unités syntaxiques simples.



Dans toutes les descriptions de syntaxe S.A.M.S.S., nous utiliserons le symbole  pour les caractériser.

Exemple : Description d'un Opérande

- Nom Générique : Partout où ce couple apparaît, éventuellement qualifié par *d'indice*, *d'opérande*, ..., il désigne de façon abstraite ce que l'utilisateur remplacera par un mot (combinaison de lettres et de chiffres) de son choix, lorsqu'il écrira son *programme S.A.M.S.S.*

Exemple : NLOG

Cette combinaison de quatre lettres désignera l'objet "Nombre de Logements".

- Commentaires : Des commentaires explicatifs peuvent être introduits dans toute utilisation de la syntaxe S.A.M.S.S., pour documenter le travail souhaité par l'utilisateur. Par exemple, si un Nom Générique n'est pas suffisamment explicite, la syntaxe S.A.M.S.S. permet d'en préciser la signification à l'aide de commentaires. Ces commentaires doivent commencer par l'unité lexicale : (\* et se terminer par l'unité lexicale complémentaire : \*). Tout texte écrit entre ces deux séparateurs sera sans effet (le logiciel n'en tiendra pas compte) à condition qu'il ne comporte aucune de ces deux unités lexicales.

Exemple : Dans la description des variables utilisées par un modèle, *PP(K,T) (\*PART DE LA POPULATION URBAINE DE REVENU K\*) REEL;* aura l'avantage d'être moins "imperméable" à la compréhension qu'une description non commentée : *PPU(K,T) REEL;*

### II.3.2. DESCRIPTION DES COMPOSANTS CARACTERISTIQUES DU PHENOMENE MODELISE

Dès le début de ses travaux, notre équipe a réfuté l'idée, communément admise jusque là, qui consistait à voir un modèle comme un programme au sens classique du terme, c'est-à-dire comme une séquence d'instructions compilables et exécutables. Nous considérons plutôt les modèles comme des données sur lesquelles doit opérer la séquence d'instructions. En effet, les "modèles-programmes" ont une caractéristique commune : leur plus ou moins grande illisibilité, celle-ci s'accompagnant invariablement de l'absence de maîtrise (compréhension) du problème ainsi modélisé.

L'action "automatisation d'un modèle socio-économique" consistera donc, pour nous, à déduire un programme exécutable à partir de l'énoncé du problème. Cette technique d'approche diverge légèrement de celle préconisée dans des systèmes plus généraux comme L.A.D.E.S.H. [11], qui assure l'indépendance entre le langage de modélisation et le système de simulation (ce qui permet de conserver le même système de simulation pour des langages de description différents).

Comme nous l'avons vu au chapitre précédent, l'énoncé du problème est concrétisé par une liste des composants caractéristiques du phénomène modélisé, et par l'ensemble des liens qui existent entre ces composants.

Dans un premier temps, nous allons donc demander à l'utilisateur de décrire en "*Langage S.A.M.S.S.*", chacun des composants caractéristiques (objets) du phénomène modélisé.

Nous en distinguerons trois catégories :

- . les variables, que le modèle devra évaluer,
- . les données, qui se distribuent elles-mêmes en trois groupes :
  - . données de scénario
  - . données exogènes
  - . données initiales
- . les fonctions tabulaires préenregistrées qui autorisent la prise en considération de plusieurs scénarios types.

Chacun de ces objets se présente, du point de vue du mathématicien, comme une matrice à 1, 2, 3 ... ou D dimensions (le maximum autorisé par la version actuelle du logiciel étant 8).

Leur description permettra d'élaborer le *Dictionnaire du Modèle* : traduction en une structure accessible au système S.A.M.S.S., de l'ensemble des éléments que l'utilisateur se propose de manipuler. Nous aurions pu éviter la construction d'un Dictionnaire et accepter de travailler uniquement à partir de la description des relations (d'où nous aurions pu extraire ces informations à condition de mettre en oeuvre des algorithmes d'une complexité et d'une ampleur bien supérieure), mais nous aurions alors perdu de vue l'un des aspects importants de la "philosophie" S.A.M.S.S. : l'aide à la définition cohérente d'un modèle. En effet les contrôles de cohérence entre la définition et l'utilisation des variables n'auraient pas eu de sens, puisqu'il n'y aurait pas eu de définition. D'autre part sa présence se justifie du point de vue technique :

- facilités de recherche d'un objet (que ce soit pour l'initialiser, le consulter ...)
- possibilités de mise à jour.

### II.3.2.1. Les entrées du module de création du dictionnaire

Nous venons d'évoquer la forme sous laquelle se présente les objets à manipuler : matrices, dimensions de matrice ... Dans la suite, nous utiliserons de préférence les termes d'opérandes et d'indices.

#### . Les indices

En nombre relativement restreint ( $\approx 15$  pour le modèle MEDEE-3), l'utilisateur leur associe une nomenclature. Par exemple, à l'*indice* de "nom générique" I, les économistes, qui ont conçu MEDEE-3, associent 6 classes d'agglomérations : agglomération parisienne (1), agglomérations de plus de 100 000 habitants (2), agglomérations de 30 000 à 100 000 habitants (3), agglomérations de 10 000 à 30 000 habitants (4), agglomérations de moins de 10 000 habitants (5), communes rurales (6).

Par définition, les *indices* sont de type entier. Ils ont une plage de variation bien définie et bornée qui correspond soit à l'ensemble des codes de nomenclature (pour I nous aurons 1, 2 ... 6), soit à la codification d'une liste de valeurs (cas de l'indice Temps).

Les *indices* permettent de sélectionner les différentes réalisations d'une variable, d'une donnée ou d'une fonction tabulaire.

. Les opérandes

En nombre nettement plus important que les indices ( $\approx 950$  pour le modèle MEDEE-3), ils désignent les variables (exogènes et endogènes) à manipuler. Ils peuvent être de *type entier ou réel* et on ne leur connaît pas a priori de *plage de variation* bornée.

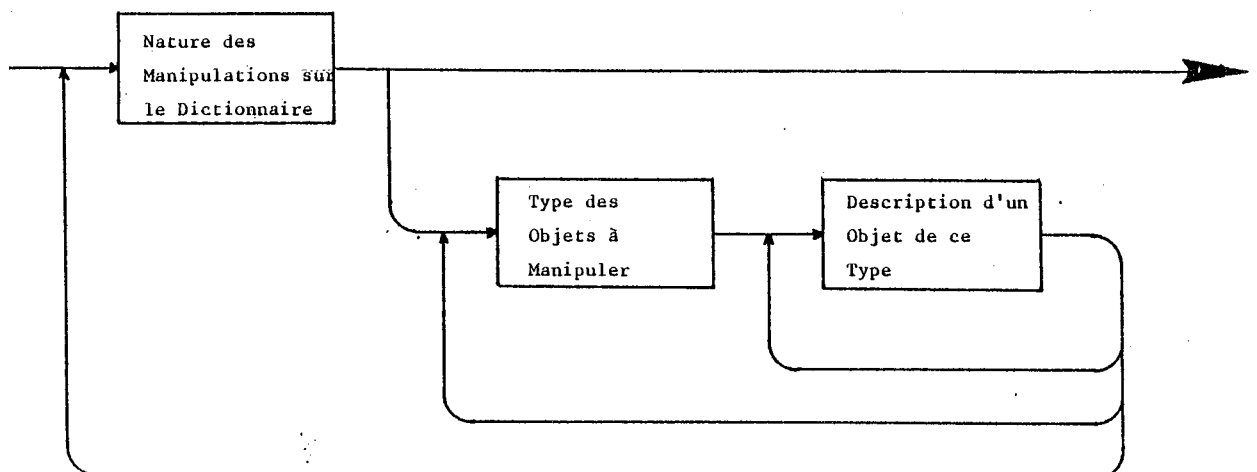
Certains *opérandes* peuvent être utilisés comme *indices* (c'est le cas des objets dits "données de scénario" qui interviennent dans la définition des fonctions tabulaires préenregistrées), et dans ces cas ils devront être décrits comme des objets de *type entier* pour lesquels la *plage de variation* sera nécessairement connue et précisée.

### II.3.2.2. La syntaxe des entrées du module

On trouvera en annexe la carte syntaxique complète du "langage de description" qui doit être respecté pour décrire les composants caractéristiques du phénomène modélisé.

. Forme générale

La forme générale de la Description des Composants Caractéristiques du modèle, peut se représenter de la façon suivante :



Exemple :

```

1  INSERER
2  INDICE
3      I DE 1 A 6;          (*CLASSES D'AGGLOMERATIONS*)
4      K DE 1 A 3;        (*CLASSES DE REVENUS DES MENAGES*)
5      T 1975, 1980, 1985, 1990, 1995, 2000; (*DATES DE SIMULATION*)
6
7  OPERANDE
8      PPU (I,T) REEL DONIN; (*PART POPULATION URBAINE/I*)
9      PPU (K,T) REEL;      (*PART POPULATION URBAINE/K*)
10     TCPPU (I) REEL;      (*TAUX DE CROISSANCE DE PPU (I)*)
11     NLOGDEM (T) REEL;    (*NB LOGEMENTS DEMOLIS DE T-1 A T*)
12     MENURB (K,T) REEL;   (*NB MENAGES URBAINS DE REVENU K*)
13     POLAME REEL DE 1 A 5 DONSC; (*POLITIQUE D'AMENAGEMENT DU TERRITOIRE*)
14     FH3101 (POLAME,I) REEL DONSC; (*SCENARIOS POUR TCPPU (1)*)
15     EMEN (K,T) REEL DONEX; (*PART DES MENAGES DE REVENU K*)
16     POLLOG REEL DE 1 A 5 DONSC; (*POLITIQUE DU LOGEMENT*)
17     FH3102 (POLLOG,T) REEL DONSC; (*SCENARIOS POUR NLOGDEM (T)*)
18     MENURB (T) REEL DONEX; (*NB MENAGES URBAINS*)
19  FINMAJ

```

- . Toute *Description des Composants Caractéristiques* (objets) d'un modèle doit débiter obligatoirement par un mot-clé qui précisera la nature des opérations à effectuer sur le *Dictionnaire du Modèle* :

- INSERER
- SUPPRIMER
- MODIFIER

D'autre part, tout ensemble (éventuellement réduit à un élément) de "*Description d'un Objet*" doit être précédé d'un mot-clé qui identifie le type des objets à introduire, supprimer ou modifier dans le *Dictionnaire du Modèle* :

- INDICE pour les dimensions des matrices (nomenclatures ...)
- OPERANDE pour les variables, qu'elles soient endogènes ou exogènes.

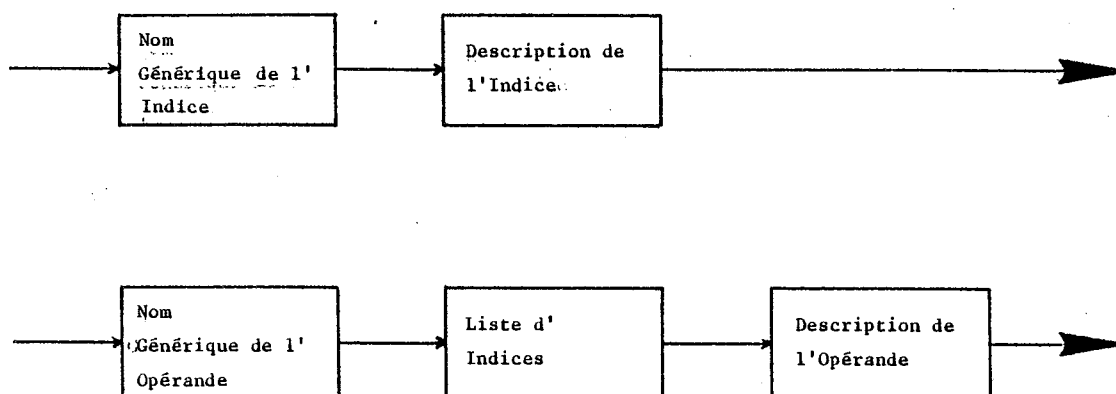
Enfin toute *Description des Composants Caractéristiques* du modèle se termine par un mot-clé "fin de manipulation" : FINMAJ.

Remarque : On notera que le programme d'entrée du module de création du dictionnaire d'un modèle comporte un aspect de documentation du modèle, pour peu que l'utilisateur prenne le temps d'y introduire des commentaires.

## . Description d'un Objet

Une description d'objet est constituée d'une suite de description de caractéristiques dont la présence peut être facultative.

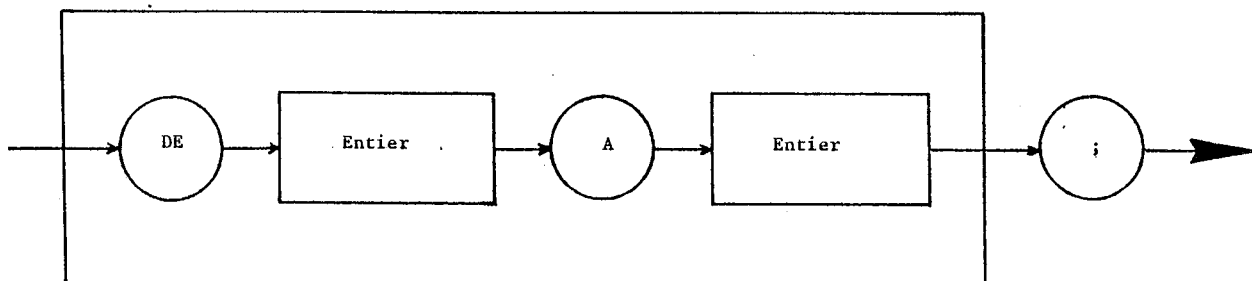
Elle prendra l'une des deux formes suivantes, en fonction du type d'objet décrit :

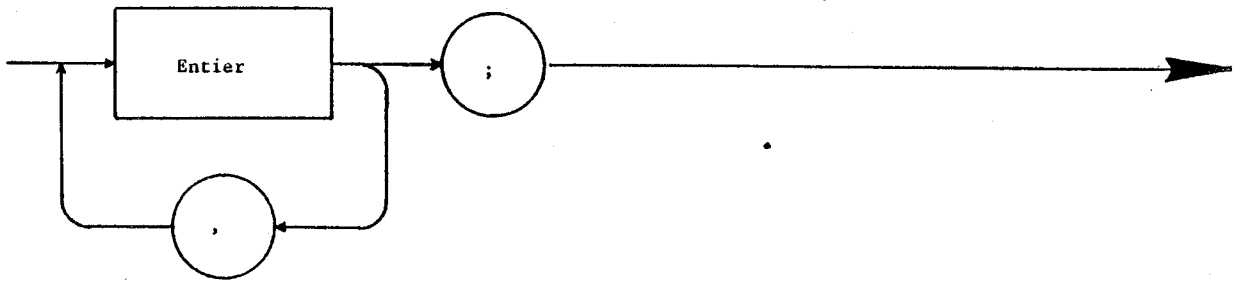


Le Nom Générique d'un indice est un libellé alphanumérique quelconque, mais doit être unique dans l'ensemble des indices utilisables par le modèle et de longueur au plus égale à huit (8) caractères alphanumériques.

Dans l'exemple citée plus haut, I est l'indice qui traduit la nomenclature retenue pour les types d'agglomérations, K traduit celle retenue pour les revenus des ménages ...

La Description de l'indice peut, elle aussi, se présenter sous deux formes différentes :



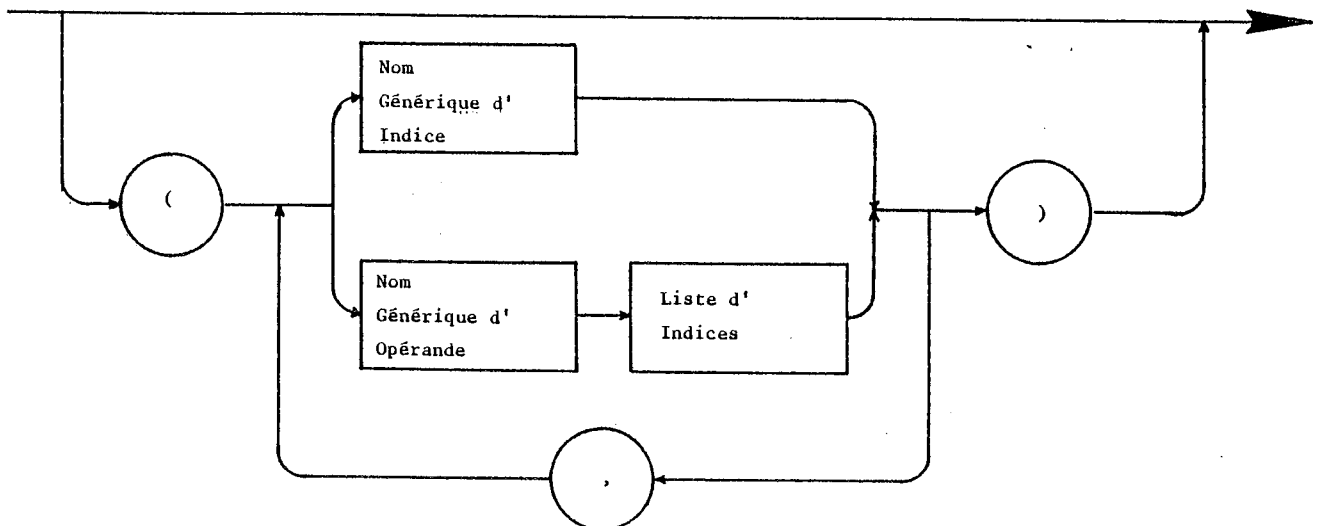


Cette alternative dans la description d'un indice permet la prise en compte des codes de nomenclature définie par l'utilisateur pour cet indice, ou la liste des valeurs à associer à cet indice.

Le Nom Générique de l'Opérande est un libellé alphanumérique quelconque. Contrairement à ce que nous avons dit pour les indices, ce libellé n'est pas nécessairement unique dans l'ensemble des opérandes utilisables par le modèle. En effet, pour identifier un opérande, nous nous servons du couple d'informations suivant : (Nom Générique de l'opérande, liste d'indices). Pour faciliter la (re) lecture (lien entre nom générique d'un opérande et sa signification économique), nous avons fixé la longueur maximum d'un nom générique d'opérande à 11 caractères alphanumériques.

Exemple : PPU, NLOGDEM, FH3101 ...

La liste d'indices associée à un nom générique d'opérande peut, quant à elle, se représenter de la façon suivante :



On notera au passage qu'un opérande peut servir d'indice, avec cependant une contrainte qui n'apparaît pas sur ce schéma : cet opérande ne doit pas, lui-même comporter d'opérande dans la liste d'indices qui le qualifie. En d'autres termes, la récurtivité apparente de la syntaxe d'une liste d'indices s'arrête au niveau 2.

Ainsi la liste d'indice (POLAME, T) associée à la fonction tabulaire FH3101 qui permet, dans le modèle MEDEE-3, de mémoriser les différents scénarios possibles pour l'évolution dans le temps du taux de croissance de la part de la population urbaine, est une liste syntaxiquement correcte. Il en est de même pour la liste d'indice (I, ETACEN(I,T), POLLOG, T) qui définit la fonction tabulaire FH3106. Par contre, le module de création du dictionnaire refuserait la définition d'une variable dont l'une des dimensions serait la fonction tabulaire FH3101 :

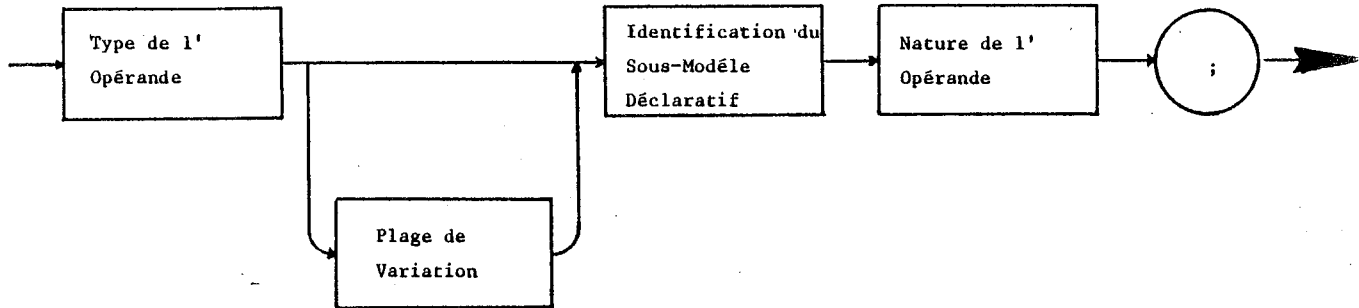
$V_1$  (FH3101(I,ETACEN(I,T),POLLOG,T),T) n'est pas acceptée.

Une deuxième restriction n'apparaît pas sur ce schéma. Elle concerne l'"ordre défini sur les indices". En effet, si  $I_1, I_2, I_3, \dots$  In sont les indices, décrits dans cet ordre par l'utilisateur, une variable  $NG_1(I_1, I_3)$  sera reconnue alors que la variable  $NG_2(I_3, I_1)$  sera rejetée comme ne faisant pas partie du domaine de définition du modèle. Si cette interprétation du domaine de définition d'un modèle peut paraître abusive, elle n'en est pas moins nécessaire pour autant. Nous admettons volontiers que pour le modélisateur les deux variables recouvrent le même concept, par contre l'utilisation de  $NG_1(I_3, I_1)$  en lieu et place de  $NG_1(I_1, I_3)$  dans une équation de calcul aboutirait à des erreurs lamentables au moment de l'évaluation, erreur qu'il serait très délicat de localiser pour un non informaticien.

Une dernière remarque encore, tout indice ou opérande servant d'indice, doit avoir été défini préalablement à son apparition dans une liste d'indices.



Une description d'opérande peut se formaliser de la façon suivante :

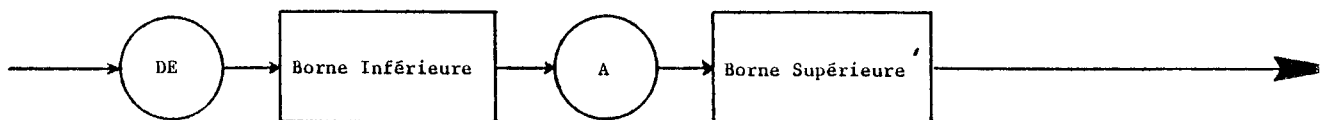


Le type de l'opérande est l'un des mots-clés ENTIER ou REEL. Il est obligatoire de le préciser puisqu'il indique au système que l'opérande (identifié par le Nom Générique et sa liste d'indices) qui précède sera représenté par une matrice d'entier (s) ou par une matrice de réel (s).

La plage de variations de l'opérande. La possibilité de préciser la borne inférieure et la borne supérieure de l'intervalle de définition de l'opérande a été introduite pour permettre l'utilisation d'opérandes comme indice. Sa présence n'est nécessaire que dans ce cas.

Il serait cependant intéressant d'exploiter cette information, lorsqu'elle a été fournie par l'utilisateur, à d'autres fins. On pourrait par exemple s'en servir pour contrôler la cohérence des calculs demandés, au cours de la résolution numérique.

On peut représenter sa règle d'écriture par le schéma suivant :



dans lequel, DE et A sont des mots-clés obligatoires, et Borne inférieure et Borne supérieure sont des valeurs numériques entières si l'opérande est de *type entier* (ou réel), ou des valeurs numériques réelles si l'opérande est de *type réel*.

L'identificateur du sous-modèle déclaratif permet, lorsque le modèle à traiter via le logiciel S.A.M.S.S. est subdivisé en plusieurs sous-modèles, d'identifier celui auquel se rattache l'opérande.

La nature de l'opérande caractérise le fait que l'opérande en cours de description est soit une donnée de scénario (DONSC) soit une donnée exogène (DONEX), soit encore une donnée initiale (DONIN). Lorsque aucun de ces trois mots-clés (DONSC, DONEX, DONIN) n'est précisé, le logiciel assimilera l'opérande à une variable endogène, à évaluer par le modèle.

#### . Remarques et restrictions

Si la carte syntaxique des entrées du module de création du dictionnaire d'un modèle traduit d'importantes facilités de manipulations du *Dictionnaire du Modèle*, l'état actuel de développement du logiciel S.A.M.S.S. est plus restrictif. En effet, bien que syntaxiquement acceptée, toute opération ponctuelle (insertion, suppression ou modification d'un ou plusieurs objets dans un dictionnaire existant) ne produira pas l'effet attendu.

Le module de création du dictionnaire d'un modèle ne fonctionne actuellement qu'en mode création.

### II.3.2.3. Les informations fournies par le module

Nous distinguerons deux catégories d'informations fournies par ce premier module : celles qui sont exploitables par l'utilisateur, donc apparentes, et celles, transparentes qui seront utilisées par les autres modules du logiciel S.A.M.S.S.

#### . Les informations apparentes

Dans cette catégorie, nous trouverons d'une part la trace de la *description des objets* du modèle, et d'autre part une série d'éléments statistiques concernant l'exécution du modèle.

Pour chaque description d'objet, la *trace* précisera éventuellement le ou les codes d'erreurs détectées au cours de son analyse.

Pour chaque exécution de ce module, nous produisons les éléments d'informations *statistiques* suivants :

- Nombre d'erreurs détectées au cours de l'analyse des descriptions.
- Nombre d'indices insérés dans le dictionnaire du modèle.
- Nombre d'indices supprimés dans le dictionnaire du modèle.
- Nombre d'indices dont les caractéristiques sont modifiées.
- Nombre total d'indices connus du modèle.
- Nombre d'opérandes insérés dans le dictionnaire du modèle.
- Nombre d'opérandes dont les caractéristiques sont modifiées.
- Nombre total d'opérandes connus du modèle.
- Taille de la zone mémoire (exprimée en octets) nécessaire pour élaborer ou charger le *dictionnaire du modèle*.

On trouvera au chapitre IV une illustration de ces différentes sorties.

#### Les informations transparentes

Il s'agit de la description codée des composants caractéristiques (objets) que le modèle pourra manipuler. Si l'analyse de la description de ces objets n'a révélé aucune erreur de syntaxe, sa traduction en code accessible à la machine (dictionnaire du modèle) est validée et sauvegardée. Les étapes suivantes de l'automatisation d'un modèle pourront alors s'y référer pour diverses opérations :

- contrôle de l'existence d'un objet,
- contrôle de la cohérence *Description-Utilisation* d'un objet,
- liaison entre l'identificateur d'un objet et sa représentation dans la Base des Opérandes du modèle,
- ...

Même si son existence n'est pas perçue par l'utilisateur, le Dictionnaire du Modèle est donc un élément essentiel du logiciel S.A.M.S.S.

Le problème fondamental, lorsqu'on aborde la création d'un dictionnaire, réside dans la façon dont il faut organiser les informations pour autoriser la mise à jour du dictionnaire au moment de son utilisation, pour "garantir un coût d'accès global minimal" [12], le tout en occupant un espace mémoire raisonnable. Pour des questions de délais de réalisation essentiellement, mais aussi parce que le sujet est bien connu [13], [14], [15], nous ne nous sommes pas livrés à des mesures d'efficacité des algorithmes de construction et de parcours du dictionnaire. Citons simplement les temps de création d'un dictionnaire de plus de 1000 variables (chaînes alphanumériques de 12 caractères en moyenne), soit 12 secondes sur un IBM 3033, pour affirmer que l'opération n'est pas prohibitive.

### II.3.3. RESERVATION ET STRUCTURATION DE LA "BASE DES OPERANDES"

Décrire les composants caractéristiques du phénomène étudié est une opération importante, nous venons de le voir. Mais pour pouvoir les manipuler, au travers des équations qui traduisent le comportement du phénomène, autrement que comme des concepts, il est nécessaire d'associer à chacun d'eux un espace physique dans lequel on pourra mémoriser ses réalisations.

L'objectif de cette seconde fonction consistera donc d'une part à évaluer et à réserver l'espace physique nécessaire à la représentation de tous les objets d'un modèle (décrits au cours de la première étape), et d'autre part à structurer l'espace acquis de telle sorte que les entrées/sorties soient minimisées au cours de la résolution numérique du modèle et de manière à créer le lien entre la description d'un objet et l'espace physique où seront stockées ses réalisations.

#### II.3.3.1. Evaluation et réservation de l'espace physique

Nous ne nous intéressons ici qu'aux objets de type OPERANDE. A partir de certains éléments descriptifs mémorisés dans le *Dictionnaire du Modèle*, tels que :

- . le type de l'opérande : ENTIER ou REEL,
- . la liste d'indices qui qualifie l'opérande,
- . les plages de variation de chacun des indices utilisés dans la liste d'indices,
- . . . .

et de certaines spécifications du système-hôte, telles que :

- . la taille de l'espace physique nécessaire au stockage d'un entier,
- . la taille de l'espace physique nécessaire au stockage d'un réel,
- . . . .

il est possible d'évaluer l'occupation mémoire (centrale ou annexe) de chacun des composants caractéristiques de type OPERANDE.

Le calcul de la taille de l'espace physique nécessaire au stockage de la "*Base des Opérandes*" du modèle se fera alors par sommation de la taille de chacun de ses éléments.

La *Base des Opérandes* d'un modèle peut atteindre des proportions importantes. Pour la version actuelle du modèle MEDEE-3 (le cas de la France), on en est à 2 Méga octets. Il était donc exclu que celle-ci soit présente, dans son intégralité au moment de son utilisation (création, consultation ou mise à jour). Il faut la stocker sur un support annexe, qui sera réservé via le langage de commandes du système hôte.

#### II.3.3.2. Structuration de l'espace acquis

Pour améliorer les temps d'entrées-sorties, que ce soit au cours de la Résolution Numérique du Modèle, au moment de l'initialisation de la *Base des Opérandes*, ou encore lors de l'interrogation de celle-ci, il est nécessaire d'organiser cet espace. Il faut que la *Base des Opérandes* soit utilisable en mode "accès direct", tout en regroupant les représentations des opérandes ayant le plus de chance d'être utilisés conjointement.

Nous allons donc découper la *Base des Opérandes* du modèle en "segments", chaque segment correspondant :

- . soit à l'espace physique où seront regroupées les réalisations, pour une date de simulation donnée, de l'ensemble des opérandes qui dépendent de la date de simulation (opérandes dont la liste d'indices comporte T),
- . soit à l'espace physique où on trouvera toutes les réalisations des opérandes indépendants de la date de simulation.

La *Base des Opérandes* du modèle comporte alors  $N+1$  segments,  $N$  étant le nombre de périodes de simulation retenues pour la résolution numérique.

Chacun des segments (encore trop important pour être chargé en mémoire centrale) est subdivisé en "pages" accessibles en mode "access-direct". La taille d'une page sera déterminée par le type de mémoire annexe, support de la *Base* (nombre d'octets d'une piste de disque par exemple).

### II.3.3.3. Création du lien entre la description d'une variable et ses réalisations

A partir des principes de structuration de l'espace acquis pour mémoriser la Base des Opérandes d'un modèle, nous définissons le lien description-réalisations par le sextuplet suivant :

- Numéro de segment : 0 si l'opérande ne dépend pas de la date de simulation.  
1 à N si l'opérande est fonction de la date de simulation.
- Numéro de page : Il s'agit du numéro de *page* (dans le segment correspondant) où sera stocké le début des réalisations de l'opérande (les réalisations d'un opérande peuvent s'étendre sur plusieurs pages). Pour un opérande dont les réalisations dépendent de la date de simulation, ce numéro est le même, quel que soit le numéro de segment.
- Adresse : Il s'agit de l'adresse (numéro d'ordre de l'octet), dans la page identifiée plus haut, du début de l'espace physique où sont stockées les réalisations de l'opérande.
- Taille de l'opérande : Exprimée en octets.

- Nombre de pages d'un segment : qui contient les opérandes dont les réalisations dépendent de la date de simulation : NEPAGSEGT.
- Nombre de pages du segment : qui contient les opérandes dont les réalisations sont indépendantes de la date de simulation : NEPAGSEGO.

Pour chaque opérande défini à l'étape précédente, il faudra donc évaluer les quatre premiers éléments de ce sextuplet, les deux derniers étant évalués une seule fois en fin d'exécution du module de formatage de la Base des Opérandes d'un modèle.

#### II.3.3.4. Les entrées du module

L'utilisateur n'intervient pas à ce niveau. En effet, toutes les informations utiles pour le Formatage de la Base des Opérandes d'un modèle sont disponibles dans le Dictionnaire du Modèle qui a été validé et sauvegardé en fin d'étape précédente.

#### II.3.3.5. Les sorties du module

Elles se résument d'une part en une mise à jour du Dictionnaire du Modèle, pour y insérer les informations qui permettront de localiser les réalisations de chacun des opérandes dans la *Base des Opérandes* du modèle, à partir de leur nom. D'autre part, toute exécution de ce module sera caractérisée par la création de la Base des Opérandes du Modèle. Enfin, l'utilisateur trouvera systématiquement un certain nombre d'éléments statistiques :

- Nombre de segments qui ne contiennent que des opérandes dont les réalisations dépendent de la date de simulation.
- Nombre de pages de l'un de ces segments.
- Nombre de pages du segment qui ne contient que des opérandes dont les réalisations sont indépendantes de la date de simulation,

qui permettent éventuellement de modifier les caractéristiques de la réservation d'espace physique nécessaire pour stocker la Base des Opérandes du Modèle.



#### II.3.4. DESCRIPTION DES RELATIONS D'UN MODELE

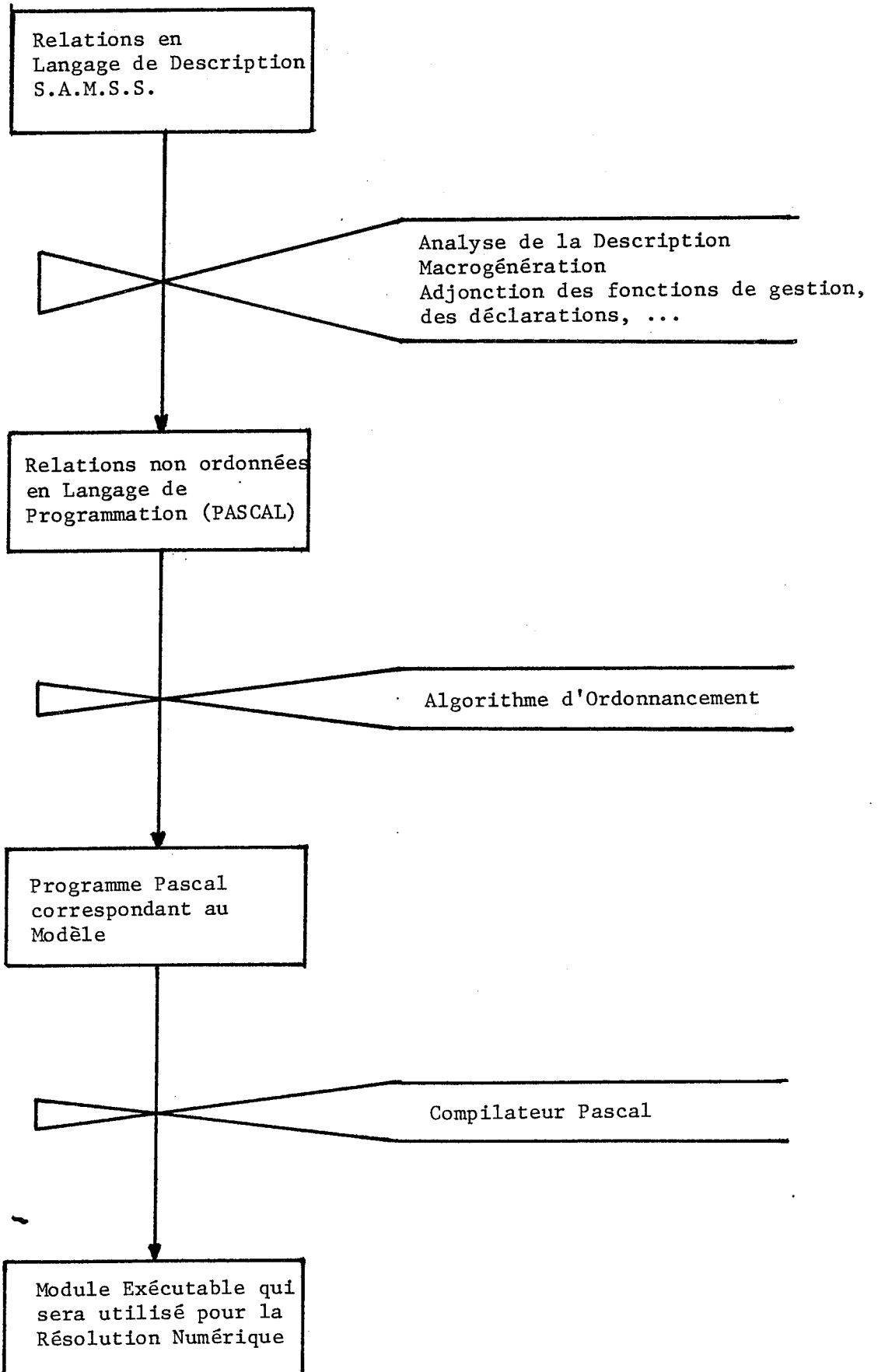
Si nous ne souhaitions pas nous contenter de la réalisation d'un "programme-modèle", nous n'envisagions pas non plus créer ce que l'on appelle un "système de simulation général" [20]. En effet, d'une part ce type de système est plus ardu à maîtriser par un non informaticien, les termes et les concepts utilisés n'y sont pas adaptés à tous les types de modèles et à leurs utilisateurs, et d'autre part l'aspect informatique de l'automatisation du modèle MEDEE-3 n'intéressait guère les décideurs de la Direction Générale XII de la Communauté Economique Européenne, ce qui ne les incitait pas au développement d'outils plus généraux.

Nous avons donc préféré nous ranger sous la bannière des "systèmes spécialisés" en essayant cependant de couvrir une gamme de problèmes suffisamment large, comme cela avait été fait pour TROLL [30] avec les applications de type économétriques.

Les modèles du type de MEDEE-3 sont dits "parallèles" [17], c'est-à-dire que leurs corps (équations) ne comportent pas, contrairement aux "modèles procéduraux", de séquence obligatoire qui détermine l'ordre dans lequel les composants doivent agir. Cela ne signifie pas pour autant que la résolution numérique des équations du modèle se fera en parallèle. Nous aurons recours à un algorithme d'ordonnement, ce qui présente un avantage évident pour l'utilisateur :

- . il n'a pas à se soucier de l'ordre d'écriture des relations (quel qu'en soit le type), et il pourra donc maintenir au niveau de la description des relations, la structure du phénomène modélisé (aspect documentaire).

On voit donc se profiler la séquence de travaux qu'il faudra effectuer à partir de la description des relations, pour aboutir à la forme exécutable du modèle (schéma page suivante).



### II.3.4.1. Le module de macrogénération

L'utilisateur va devoir fournir la description, en langage S.A.M.S.S., de l'ensemble des équations qui traduisent le comportement du phénomène qu'il étudie. Mais attention cela ne signifie pas, pour autant que le logiciel S.A.M.S.S. n'admet pas les descriptions partielles. Au contraire, il sera même judicieux d'exploiter cette fonction et les suivantes pour construire le modèle final par "touches" successives (voir l'exemple étudié au chapitre IV).

Les équations (ou relations) que l'utilisateur décrit, doivent respecter un certain nombre de conventions :

- Ce sont des relations qui manipulent les différents objets définis lors de la première étape du logiciel (opérandes, indices) ainsi que des variables intermédiaires.
- Ces relations présentent trois caractéristiques principales :
  - . elles doivent être statiques ou dynamiques
  - . elles sont soit de type calcul, soit de type contrôle
  - . elles sont hiérarchisables les unes par rapport aux autres.

Nous dirons qu'une relation est statique si elle ne fait intervenir que des variables (opérandes) dont les réalisations sont indépendantes de la date de simulation et/ou les réalisations, pour la date de simulation courante, des variables qui sont fonction de la date de simulation. Par opposition, nous affirmerons qu'une relation est dynamique si elle manipule les réalisations, pour une date de simulation différente de la date courante, d'au moins une variable.

Toute relation comporte une partie gauche et une partie droite, par rapport à un opérateur d'affectation ou de comparaison. Une relation de calcul est une relation dont la partie gauche représente la variable à évaluer et dont la partie droite est une expression quelconque, le séparateur entre ces deux parties étant alors l'opérateur d'affectation. Une relation de contrôle est une relation dont les parties

droite et gauche sont des expressions quelconques, le séparateur étant ici l'un des opérateurs de comparaison :  $<$ ,  $\leq$ ,  $>$ ,  $\geq$ ,  $=$  .

Les relations qui traduisent le comportement du phénomène modélisé seront hiérarchisables les unes par rapport aux autres, si la matrice d'incidence du problème à résoudre est triangulaire inférieure. Cette vérification sera faite dans une étape ultérieure.

Afin d'éviter à l'utilisateur l'apprentissage d'un langage de programmation, avec tout ce que cela comporte comme écueils, nous avons orienté notre travail vers la définition d'un langage d'expression des relations d'un modèle qui ne soit pas trop éloigné du formalisme propre à l'utilisateur, l'économiste en l'occurrence. La création du Langage Pseudo-Pascal (L.P.P.), qui, tout en conservant la puissance d'un langage évolué tel que PASCAL, laisse à l'utilisateur la faculté de désigner les variables manipulées comme il a coutume de le faire, n'atteint cet objectif que très partiellement. En effet, même si les utilisateurs de MEDEE-3 ont apprécié le chemin parcouru (ils étaient habitués à manipuler FORTRAN), nous estimons, à l'usage qu'il est encore insuffisant. La notion d'"instruction-composite" n'évoque pas grand chose pour eux, la manipulation des matrices leur pose encore bien des problèmes (quels indices varient en premier, ...), les contraintes syntaxiques sont encore trop nombreuses, et certaines opérations très appréciées (MIN, MAX, SUP, INF, PUISS ...) par les économistes n'y figurent pas.

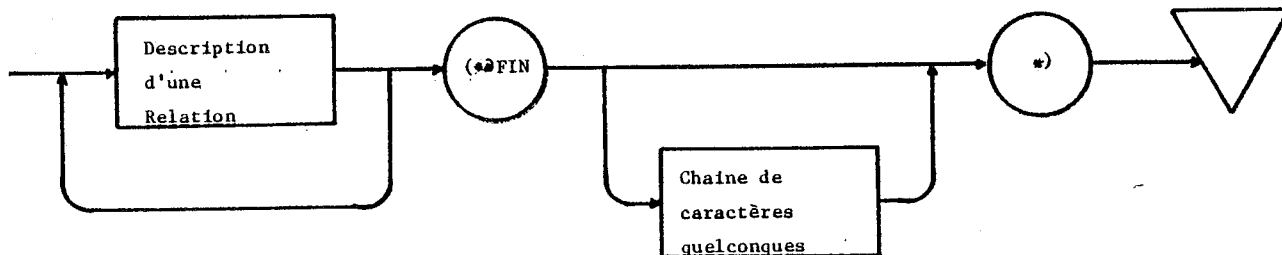
Le module de macrogénération des relations d'un modèle (MACROG) est alors chargé d'analyser le texte de description des relations (exprimée en L.P.P.) et d'effectuer les transformations nécessaires à son acceptation par le compilateur PASCAL.

#### II.3.4.2. Les entrées du module de macrogénération

On trouvera en annexe la carte syntaxique complète du "langage de description" que l'utilisateur devra respecter en formulant les relations qui traduisent le comportement du phénomène modélisé.

. Forme générale

La forme générale de la description des relations d'un modèle peut se schématiser de la façon suivante :



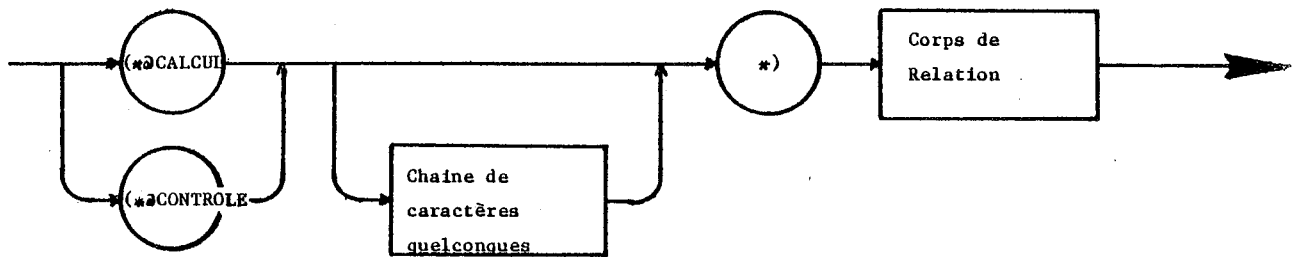
A partir de ce schéma, nous pouvons constater que le texte de la Description des Relations d'un modèle doit contenir au moins une relation et se terminer par un commentaire au sens PASCAL du terme, c'est-à-dire une suite de caractères alphanumériques comprise entre les unités lexicales '(\*' et '\*)', dont les quatre premiers symboles sont nécessairement ∂FIN .

Exemple : 1 (\*∂CALCUL DE LA PART DE LA POPULATION URBAINE DE REVENU K\*)  
 2 FOR K:= MINK TO MAXK DO  
 3 PPU (K, T) := PMEN (K, T)  
 4 (\*∂CALCUL DU NOMBRE DE MENAGES URBAINS DE REVENUS K\*)  
 5 FOR K:= MINK TO MAXK DO  
 6 MENURB (K, T) := PPU (K, T) \* MENURB (T);  
 7 (\*∂FIN DU MODELE QUI SE LIMITE AU CALCUL DE PPU (K, T) ET DE MENURB (K, T)\*)

Dans cet exemple, les lignes 1 à 3, et 4 à 6, correspondent à deux descriptions d'une relation, alors que la ligne 7 signale au logiciel que la description des relations du modèle est terminée.

### . Description d'une relation

Pour que la syntaxe d'une description d'une relation soit conforme à ce que le logiciel attend, elle devra respecter le schéma suivant :



La description d'une relation débute obligatoirement par un commentaire dont les premiers symboles sont ∂ suivi de l'un des mots-clés CALCUL ou CONTROLE. Cette entête de relation permet d'indiquer au système le type de la relation (nous en verrons l'utilité plus loin).

Pour une meilleure documentation du modèle, il est intéressant d'utiliser l'entête de relation pour exprimer en quelques mots, le contenu du corps de relation.

Exemple :

```

1 (*∂CONTROLE DE LA COHERENCE DU CONTENU DE LA VARIABLE CALCULEE
   PPU (I,T)*)
2 $VARINTR:=0;
3 FOR I:=MINI TO MAXI-1 DO
4   $VARINTR:=$VARINTR+PPU(I,T):
5   CONTROL (1-$VARINTR, EG, 0.001, CONTINUE, 2009);

```

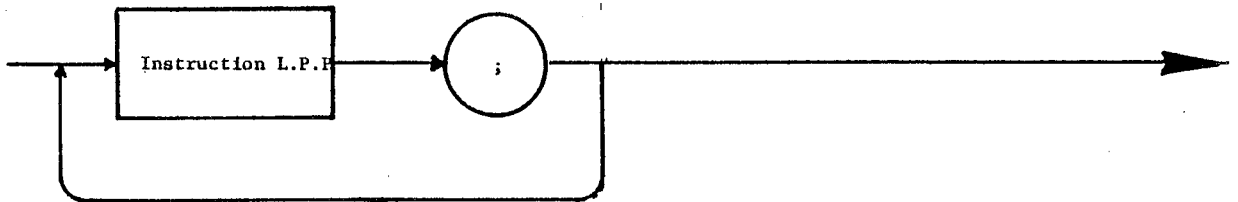
Dans cet exemple, les lignes 2 à 5 constituent le corps de relation.

La ligne 1, entête de relation, exprime qu'il s'agit d'une relation de contrôle et que l'utilisateur souhaite vérifier la cohérence des calculs précédents :  $\Sigma$  des partitions d'un ensemble = l'ensemble lui-même.

. Le corps de relation

Il s'agit de la formulation algorithmique, au moyen du *Langage Pseudo-Pascal*, de l'expression mathématique qui exprime le calcul des réalisations d'une variable endogène, ou le contrôle de cohérence des valeurs de variables préalablement calculées.

On peut le schématiser de la façon suivante :



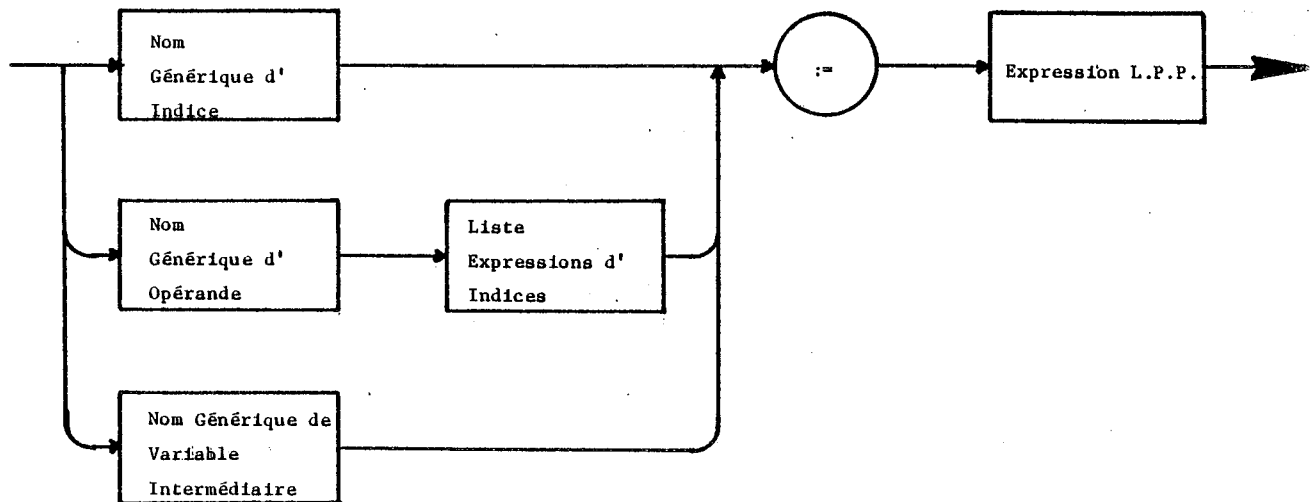
Le corps de relation est donc la description d'opérations sur les objets du modèle, description qui se traduit par une liste d'instructions L.P.P. séparées par des points virgules.

Nous allons maintenant examiner les diverses formes que peuvent prendre les instructions L.P.P., dans l'ordre :

- . l'affectation
- . l'instruction conditionnelle (ses 2 formes)
- . l'instruction composite
- . l'itération (ses 3 formes)
- . l'instruction de contrôle (ses 2 formes)

puis nous préciserons les notions . d'Expression L.P.P.,  
 . de Constante L.P.P.,  
 . de liste d'expression d'indices,

et nous aurons alors un panorama complet des possibilités qu'offre le Langage d'Expression des Relations que nous avons adopté.

L'affectation L.P.P.

Dans cette première forme d'une instruction L.P.P. (la plus usitée sans doute), les unités syntaxiques Nom Générique d'Indice et Nom Générique d'Opérande correspondent à celles que nous avons défini dans le module de création du dictionnaire d'un modèle.

Un Nom Générique de Variable Intermédiaire est un libellé alphanumérique commençant par le symbole \$ et dont le huitième caractère doit être I si la variable intermédiaire est de type Entier, ou R si elle est de type Réel. Ainsi, \$VARINTI1 sera considérée comme une variable intermédiaire entière, et \$VARINTRI comme une variable intermédiaire réelle. (Cette convention, adoptée pour le modèle MEDEE-3, peut être modifiée aisément, si elle se révèle d'un emploi pénible).

Les variables intermédiaires ne sont pas mémorisées dans la Base des Opérandes du modèle. Ce sont des variables de travail.

L'instruction d'affectation L.P.P. permet donc de transférer dans la variable endogène, la variable intermédiaire ou l'indice qui apparaît en partie gauche de l'affectation, le résultat de l'évaluation de l'expression qui se trouve en partie droite de cette affectation.



lorsque la partie gauche désigne une variable du modèle, il y aura mise à jour de son contenu dans la Base des Opérandes du Modèle, au cours de la résolution numérique.

Exemples :

- 1  $PPU (K, T) := PMEN (K, T) ;$
- 2  $MENURB (K, T) := PPU (K, T) * MENURB (T) ;$
- 3  $\$VARINTR := \$VARINTR + PPU (I, T) ;$
- 4  $I := MINI ;$

Dans les exemples 1 et 2, on exprime la modification du contenu de variables endogènes du modèle. L'expression qui apparaît en partie droite peut être plus ou moins complexe.

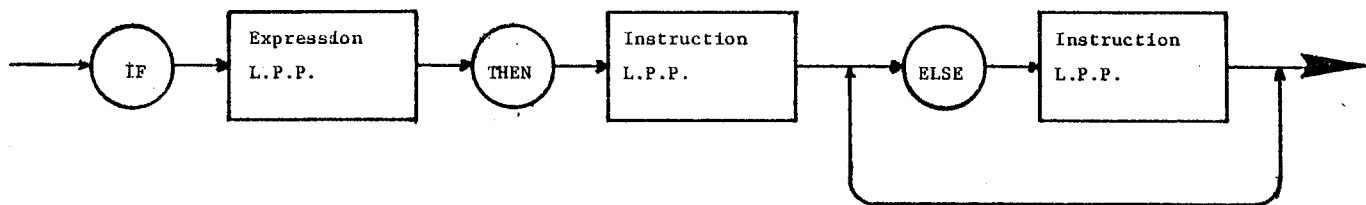
Dans l'exemple 3, c'est une variable intermédiaire (variable de travail) qui est affectée.

Enfin, dans le dernier exemple, on transfère la valeur de MINI dans l'indice I.

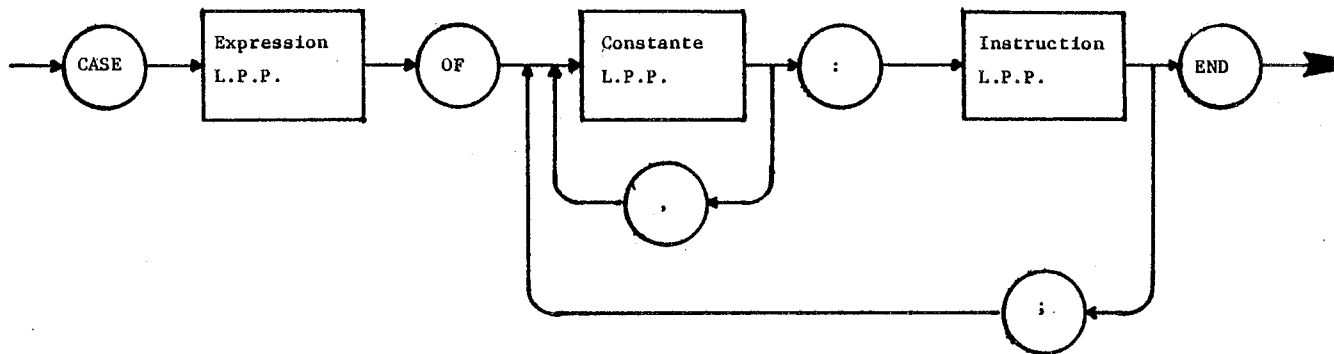
Nous reviendrons plus loin sur la syntaxe d'une Expression L.P.P. et sur celle d'une Liste d'Expressions d'Indices.

L'instruction conditionnelle L.P.P.

Première forme :



Deuxième forme :



Ces deux formes de l'instruction conditionnelle L.P.P. permettent l'expression, dans une même relation, de plusieurs chemins de calcul.

La première forme est particulièrement connue. Rappelons simplement qu'elle permet de se prémunir contre les risques de divisions par zéro. Il est en effet prudent, lorsque le calcul ou le contrôle des éléments d'une variable comporte une division, de distinguer le cas du dénominateur nul de celui où il est non nul. Cette remarque peut paraître inutile à un informaticien. A l'usage, on se rend compte que l'économiste ne raisonne pas tout à fait selon ce schéma. Il objecte fréquemment que l'affectation d'une valeur nulle à sa variable dans ce cas là devrait être naturel.

La seconde forme est moins classique. Lorsque le nombre de résultats différents que peut fournir l'évaluation de l'expression L.P.P. est tel que l'utilisation de la première forme est coûteuse et illisible, ou lorsque les calculs sont très différents pour aboutir aux différents éléments d'une matrice elle sera préférée. C'est le cas dans l'exemple que nous examinons ci-après. Il s'agit en effet d'évaluer la consommation d'énergie nécessaire à la fabrication des produits que l'on a coutume de regrouper sous le terme général d'industries grosses consommatrices d'électricité (acier brut, aluminium, ciment, verre, pâte à papier, chlore et éthylène). Or chacune de ces industries consomme une énergie que l'on calcule de façon différente, puisque les procédés de fabrication sont divergents. L'utilisation, pour cet exemple, de la première forme de l'instruction conditionnelle L.P.P. aurait été catastrophique, tant du point de vue de la compréhens-

sion du texte utilisateur, que du point de vue coût de son exécution au cours de la résolution numérique du modèle.

(\* $\partial$ CALCUL CIGCE(Y,T). CONSOMMATION D'ENERGIE DES IGCE\*)

FOR Y:=MINY TO MAXY DO

CASE Y OF

1:BEGIN (\*acier brut\*)

X:=2;

CIGCE(Y,T):=CTE(Y) + ELASIGCE1 (Y)\* VABTP(T)  
+ ELASIGCE2 (Y)\* VAIL(X,T);

END;

2:CIGCE(Y,T):=CTE(Y)\*EXP(ELASIGCE1(Y)\* LN (VAI(T)); (\*alu\*)

3:CIGCE(Y,T):=CTE(Y)\*CVP(T); (\*ciment\*)

4:CIGCE(Y,T):=CTE(Y)\*EXP(ELASIGCE1(Y)\* LN (VABTP(T))); (\*verre\*)

5:CIGCE(Y,T):=CTE(Y)\*EXP(ELASIGCE1(Y)\* LN (VATER(T))); (\*pâte  
à papier\*)

6,8:BEGIN (\*chlore et éthylène\*)

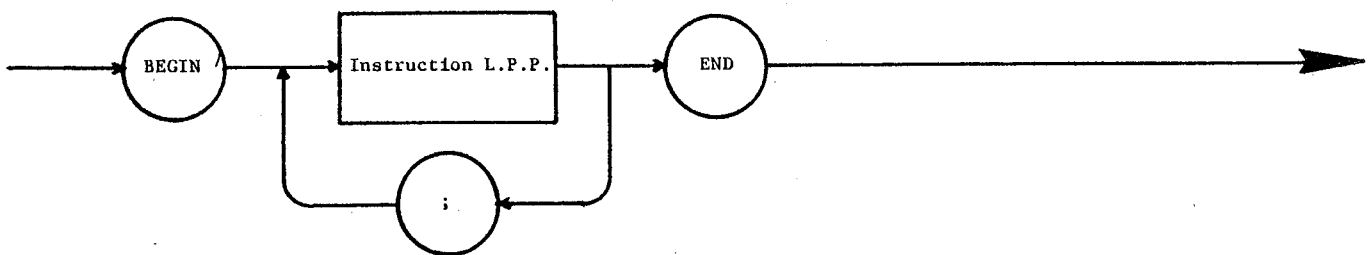
X:=3;

CIGCE(Y,T):=CTE(Y)\*EXP(ELASIGCE1(T)\*LN (VAIL(X,T)));

END;

END;

L'instruction composite L.P.P.



Lorsqu'une branche de l'algorithme qui traduit la relation de calcul ou de contrôle ne peut pas s'exprimer à l'aide de l'une des instructions L.P.P. de base, on aura recours à une instruction composite L.P.P. Il s'agit de regrouper plusieurs instructions de base et/ou instructions composites, de les encadrer par les mots-clés BEGIN et END et de considérer le tout comme une seule instruction.

```

Exemple : 1  (* $\partial$  CALCUL DU NOMBRE DE LOGEMENTS APPARTENANT A LA CLASSE J
           A L'INSTANT T*)
2  FOR J:=MINJ TO J(T)-1 DO
3  BEGIN
4  $VARINTR:=0;
5  IF T=TO+1 THEN
6  BEGIN
7  FOR I:=MINI TO MAXI-1 DO
8  $VARINTR:=$VARINTR + NLOG(I,J,T-1);
9  NLOGT1(J,T):=$VARINTR;
10 END
11 ELSE NLOGT1(J,T):=NLOG(J,T-1);
12 END;

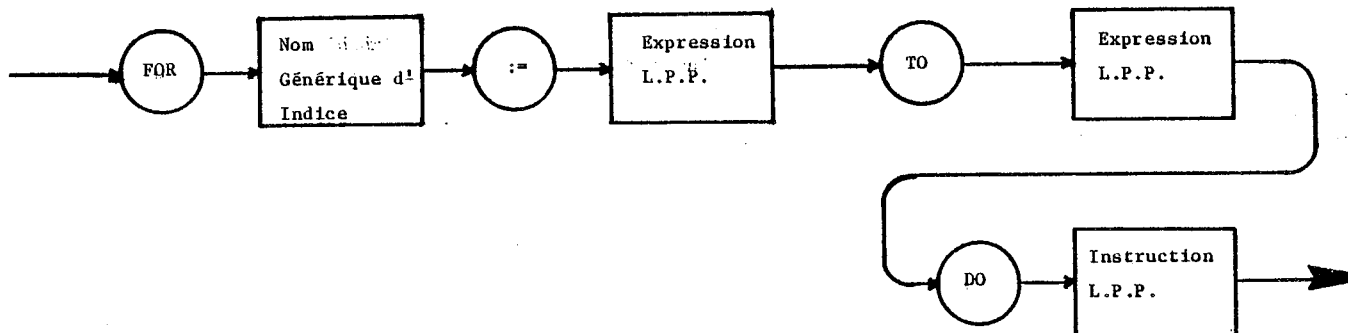
```

Dans cet exemple, les lignes 7 à 9 encadrées des lignes 6 et 10 constituent une instruction composite L.P.P. De même, en regroupant cette instruction composite et les lignes 4, 5 et 11 (instructions de base), l'utilisateur exprime, à l'aide d'une instruction composite (3 ... 12) le calcul, pour toute classe d'âge, du nombre de logements.

### L'itération L.P.P.

L'instruction itérative peut se présenter sous trois formes différentes dont l'une est une boucle au sens classique du terme : le contenu (instruction de base ou composite) de la boucle sera exécuté un nombre prédéterminé de fois (toujours le même). Les deux autres, moins habituelles, permettront d'arrêter l'exécution de l'itération par évaluation d'une expression plus ou moins complexe. L'itération L.P.P. est l'instruction de base pour la manipulation des matrices.

Nous présentons maintenant la syntaxe de chacune de ces trois formes d'itération L.P.P. en indiquant les avantages qu'elles apportent à l'utilisateur.

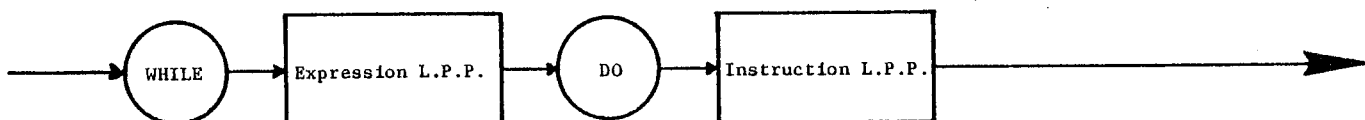
La forme classique :

Cette première forme itérative d'une instruction L.P.P. donne à l'utilisateur, la possibilité de préciser le champ d'application de la ou les opérations qui suivent. Ainsi, l'expression mathématique  $VV, IDPRIENE(V) = FH2301(POLENE, IDRIPE, V)$  qui exprime que la valeur actuelle de POLENE (scénario de politique de l'énergie), pour la valeur actuelle de IDRIPE (indice de la dérivée des prix du pétrole), on souhaite évaluer l'indice de la dérivée des prix de chacune des énergies  $V$ , cette expression mathématique sera traduite en *langage L.P.P.* de la façon suivante :

```

FOR V:=MINV TO MAXV DO
  IDPRIENE(V):=FH2301(POLENE, IDRIPE, V);
  
```

avec MINV qui représente la valeur de la borne inférieure de l'indice  $V$  et MAXV, la valeur de sa borne supérieure.

La deuxième forme :

Avec cette seconde forme de l'instruction itérative L.P.P., l'instruction L.P.P. (de base ou composite) qui suit le mot-clé DO sera effectuée tant que l'expression L.P.P. est vérifiée.

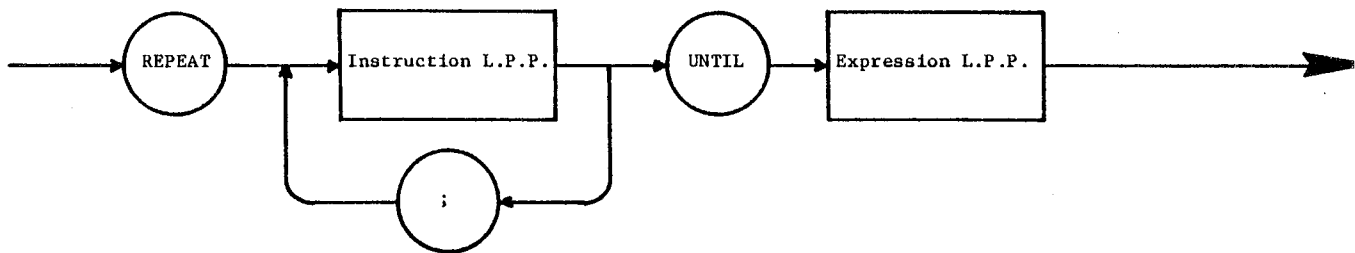
Exemple :

```

1  (*∂CALCUL DE LA CLASSE D'AGE DES LOGEMENTS AFFECTES
   PAR LES TRANSFERTS*)
2  K:=3;
3  J:=3;
4  $VARINTR:=PPU(K,T)* NLOG(J;T);
5  WHILE (MENURB(K,T)-$VARINTR-RESIDU(K,T)>MENURBT1 (J-1,K,T))
   & (J>MINJ+1) DO
6  BEGIN
7  J:=J-1;
8  $VARINTR:=$VARINTR+PPU(J,K,T-1)* NLOG(J,T);
9  END;
10 PPUJP(T):=J-1;

```

Dans cet exemple, tant que l'évaluation de l'expression booléenne (ligne 5) délivre un résultat vrai (la condition est vérifiée), l'instruction composite (lignes 6 à 9) sera exécutée.

La troisième forme :

La troisième forme de l'instruction itérative est, du point de vue de son fonctionnement, assez similaire à la seconde. En effet, le contrôle du nombre d'itérations à effectuer se fait par évaluation d'une expression L.P.P.

Elle en diffère cependant par le fait que l'expression L.P.P. étant évaluée en fin d'itération, on effectuera au moins une fois la liste d'instructions L.P.P. (de base et/ou composites) qui figurent entre les deux mots-clés REPEAT et UNTIL, quel que soit le résultat de l'évaluation de l'expression L.P.P. Si l'expression L.P.P. est vérifiée, l'itération s'arrête, sinon elle se poursuit.

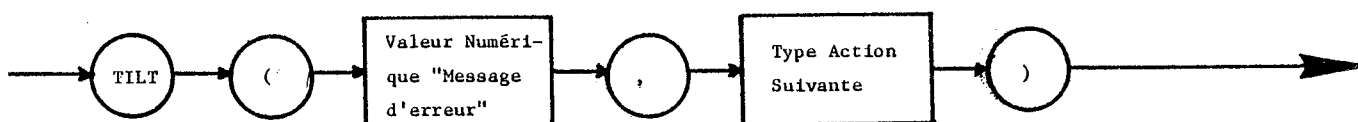
L'instruction de contrôle L.P.P.

Cette instruction n'a pas d'équivalent direct en Pascal. Il s'agit en fait d'un outil qui permet à l'utilisateur de valider le contenu de ses variables. Il peut en effet comparer le contenu de deux expressions dont la signification est identique et vérifier qu'ils sont équivalents.

Par exemple, si il existe dans le modèle une variable exogène qui recouvre, du point de vue de sa signification, une expression de une ou plusieurs autres variables (endogènes et/ou exogènes), alors l'utilisateur peut vérifier la cohérence de ses calculs par rapport à ses données.

Cette instruction de contrôle L.P.P. peut prendre deux formes différentes en fonction de l'importance que l'utilisateur accorde à la cohérence qui doit être respectée.

La première forme :



Cette première forme de l'instruction de contrôle L.P.P. est employée lorsque le modélisateur a une connaissance suffisamment précise du phénomène qu'il modélise, ou lorsqu'il estime que tout écart, si minime soit-il, entre des valeurs calculées et des valeurs de référence, amènerait à de sérieuses déconvenues dans la poursuite de la résolution numérique de son modèle.

Dans cette première forme de l'instruction de contrôle L.P.P., la valeur numérique message d'erreur est un entier qui permet à l'utilisateur, en cas de non satisfaction du contrôle, de repérer la relation qui l'émet. Il importe donc de choisir ici des valeurs qui soient discriminantes (numéro de relation par exemple).

Le type de l'action suivante exprime ce que le modélisateur souhaite faire en cas de non satisfaction du contrôle. Il a le choix entre deux options qui se matérialisent par deux mots-clés :

- . CONTINUE Dans ce cas la résolution numérique du modèle se poursuivra après émission d'un message qui permet, en principe, de localiser la relation en cause.
- . STOP La résolution numérique sera interrompue. L'utilisateur estime en effet que l'écart constaté entre les deux éléments comparés est trop important pour que l'on puisse tirer des informations cohérentes à partir des résultats produits par le modèle. Là encore il y a émission du code d'erreur : valeur numérique message d'erreur.

Exemple d'utilisation : Tout un chacun sait bien que lorsqu'on définit une partition d'un ensemble, l'union de toutes les parties ainsi déterminées est égale à l'ensemble lui-même. Ainsi, si PPU(K,T) représente la proportion de la population urbaine dont les revenus à l'instant T appartiennent à la classe K, alors la somme des valeurs calculées de PPU(K,T) pour un instant donné représente la population urbaine complète à l'instant T. Mathématiquement cela s'exprime par :

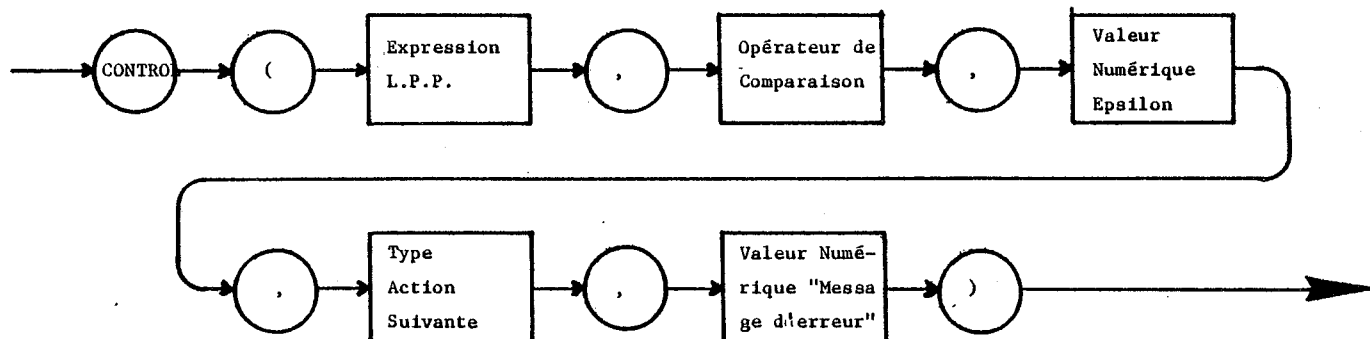
$$\sum_K \text{PPU}(K,T) = 1$$

En langage SAMSS l'utilisateur le traduira par la relation suivante :

```
(*∂CONTROLE SIGMA PPU(K,T) = 1*)
  $VARINTR:=0;
  FOR K:=MINK TO MAXK DO
    $VARINTR:=$VARINTR + PPU(K,T);
  IF $VARINTR>=1 THEN TILT (9999, STOP);
```



La deuxième forme :



Avec cette deuxième forme de l'instruction de contrôle L.P.P., le modélisateur peut se définir un intervalle de validité de ses variables calculées. En effet, l'expression L.P.P. exprime ce que l'utilisateur souhaite comparer à la Valeur Numérique Epsilon : rapport entre deux variables du modèle, résultats de calculs antérieurs, écart entre une variable exogène (variable de référence) et l'expression dont la signification est identique ...

L'opérateur de comparaison est un mot-clé qui exprime la nature du contrôle par rapport à la valeur numérique epsilon. Il doit être choisi parmi les mots-clés répertoriés dans le tableau ci-dessous dans lequel nous désignerons le résultat de l'évaluation de l'expression par EXP et la valeur numérique epsilon par le symbole  $\epsilon$ .

MOT-CLE	CONTROLE CORRESPONDANT
EG	$ EXP  \leq \epsilon$
NEG	$ EXP  > \epsilon$
INF	$(EXP < 0) \text{ et } (-EXP > \epsilon)$
SUP	$(EXP > 0) \text{ et } (EXP > \epsilon)$
INFEG	$(EXP < 0) \text{ et } (EXP \leq \epsilon)$
SUPEG	$(EXP > 0) \text{ et } (-EXP \leq \epsilon)$

Les unités syntaxiques complexes Valeur Numérique Message Erreur et Type Action Suivante ont le même rôle et la même signification que dans la première forme de l'instruction de contrôle L.P.P.

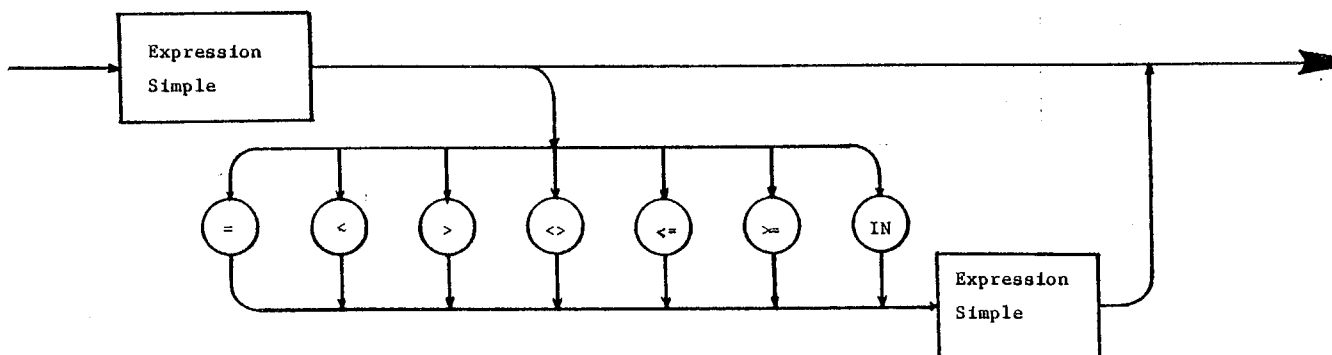
Exemple : En reprenant celui que nous venons de développer, on peut affirmer que l'égalité stricte, si elle est théoriquement justifiée, ne peut être obtenue du fait des arrondis effectués lors de l'évaluation des différentes proportions de population, arrondis que nous ne savons pas maîtriser, puisqu'ils sont pratiqués par le système-hôte de SAMSS. L'utilisation de la deuxième forme de l'instruction de contrôle L.P.P. permet alors de contrôler l'égalité à un epsilon près (epsilon très petit). Cela se traduit alors par :

```
(*∂CONTROLE SIGMA PPU(K,T) = 1 *)
  $VARINTR:=0;
  FOR K:=MINK TO MAXK DO
    $VARINTR:=$VARINTR+PPU(K,T);
  CONTROL (1-$VARINTR,EG,0.0001,STOP,9999);
```

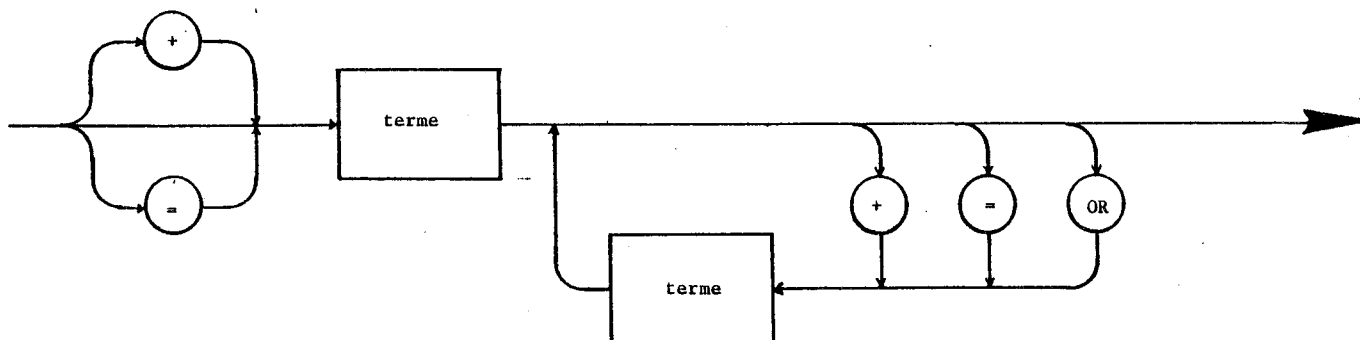
Si l'écart en valeur absolue entre  $\sum_K PPU(K,T)$ , représenté par la variable intermédiaire \$VARINTR, et la constante 1, est supérieur à 0.0001, l'exécution de cette relation provoquera l'arrêt de la résolution numérique après émission du code de message 9999.

### Une expression L.P.P.

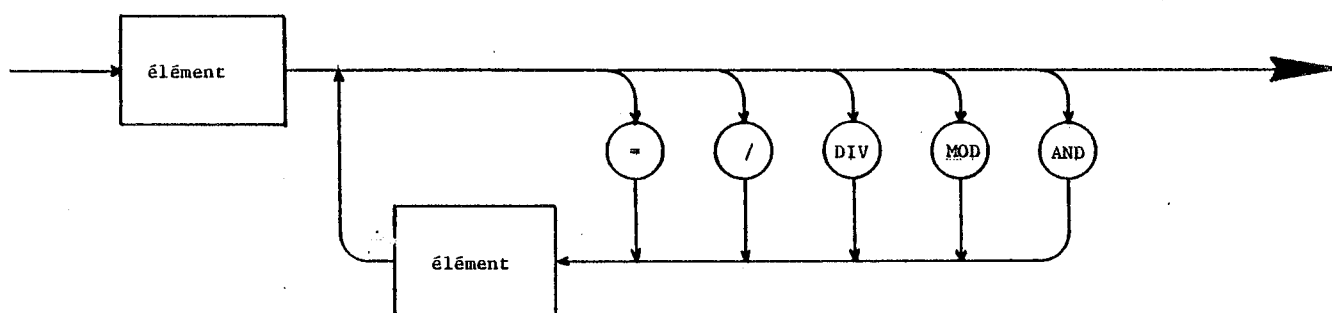
Quelle que soit sa forme, logique ou arithmétique, une expression L.P.P. peut être schématiser de la façon suivante :



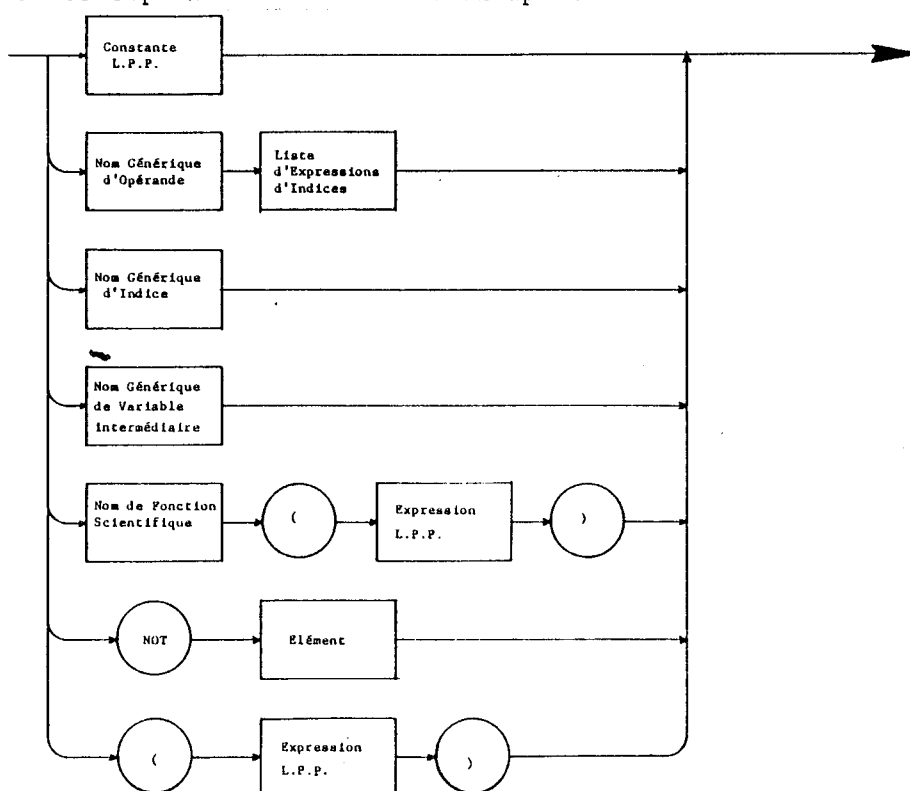
L'unité syntaxique complexe expression simple qui apparaît dans la forme générale d'une expression L.P.P. peut à son tour s'exprimer en fonction du schéma ci-dessous :



L'unité syntaxique complexe terme pouvant elle-même être schématisée par :



et dans laquelle, l'unité syntaxique complexe élément peut prendre l'une des sept formes du schéma ci-après :



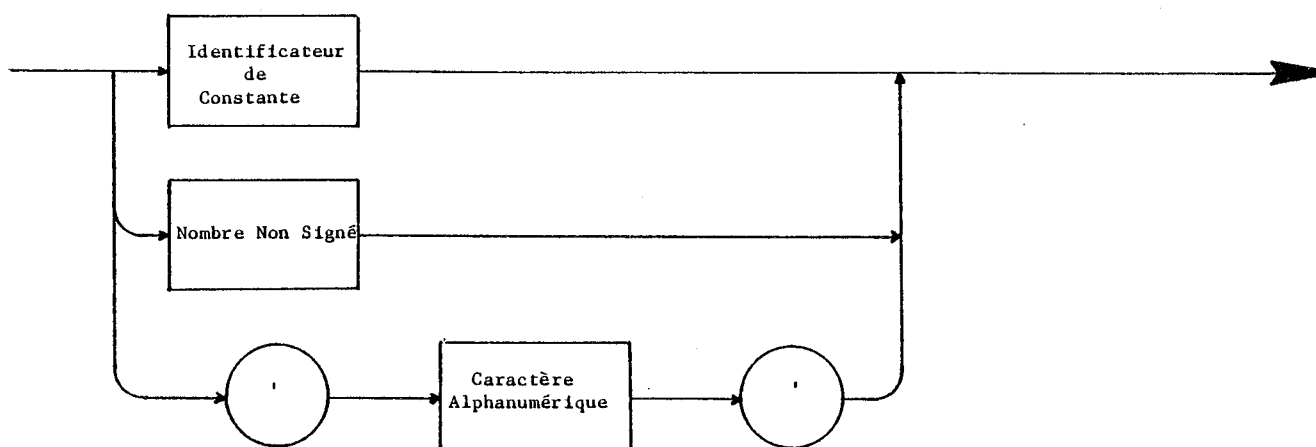
Nous verrons plus loin ce que recouvre, dans ce schéma les notions de Constante L.P.P. et de Liste d'Expressions d'Indices.

Les autres unités syntaxiques complexes ont déjà été explicitées dans les paragraphes précédents sauf Nom de fonction scientifique. Il s'agit de mots-clés qui identifient quelques unes des fonctions scientifiques les plus classiques. Actuellement le *langage S.A.M.S.S.* en admet trois (il est possible d'en rajouter d'autres, si le besoin s'en fait sentir) :

- ABS qui délivre la valeur absolue du résultat de l'évaluation de *l'expression L.P.P.* qui suit le mot-clé.
- LN qui délivre le logarithme népérien du résultat de l'évaluation de *l'expression L.P.P.* qui suit ce mot-clé.
- EXP qui délivre la valeur de l'exponentielle du résultat de l'évaluation de *l'expression L.P.P.* qui suit ce mot-clé.

#### Une constante L.P.P.

L'écriture de toute constante, dans une *instruction L.P.P.* doit respecter la syntaxe schématisée ci-dessous :



Les seuls identificateurs de constante que le logiciel SAMSS reconnaît sont ceux qui correspondent aux bornes des indices existants dans le *Dictionnaire du Modèle*. Ces identificateurs sont établis par concaténation du mot-clé MIN ou MAX avec un Nom Générique d'Indice.

Exemple : MINI est l'identificateur de la constante *borne inférieure* de la plage de variation de l'indice I du modèle.

MAXKPRIME est l'identificateur de la constante *borne supérieure* de la plage de variation de l'indice KPRIME du modèle.

#### Une liste d'expressions d'indices

Il s'agit d'une liste d'expressions L.P.P. séparées par des virgules. Une restriction cependant : dans une telle expression, le langage S.A.M.S.S. n'autorise la présence que d'un seul Nom Générique d'indice ou d'opérande. Ainsi, nous reconnaitrons l'expression I+1 dans une liste d'expressions d'indices. Par contre, si l'expression POLMUN(I) - I apparaît dans cette liste, alors elle sera rejetée.

Cette restriction s'est imposée pour répondre à notre souci de cohérence. En effet, si I+1 peut avoir un sens (classe d'agglomération suivante), il n'en est pas de même pour POLMUN(I) - I.

#### Remarques et restrictions d'utilisation des expressions L.P.P.

Pour qu'une affectation L.P.P. soit syntaxiquement et sémantiquement correcte, il est essentiel de veiller à ce que la variable désignée en partie gauche de l'affectation et le résultat de l'expression L.P.P. qui apparaît en partie droite soient de même type (ENTIER, REEL ...). La seule exception admise correspond à celle où le *type* de la partie gauche est REEL et celui de la partie droite est ENTIER : on admet qu'il y a alors compatibilité de type.

Cette règle demeure valable quels que soient les éléments d'une expression : on ne peut comparer que deux éléments de même type ...

Le mode de calcul d'une expression est défini par les règles suivantes :

- l'évaluation se fait de gauche à droite,
- il existe un ordre de précedence sur les opérateurs. Celui-ci attribué aux opérateurs multiplicatifs : \*, /, DIV (division entière), MOD (modulo), AND (et logique) la plus haute priorité. Viennent ensuite les opérateurs additifs \*, -, OR (ou logique). Enfin, la plus basse priorité est affectée aux opérateurs de relation : =, <, >, <> (non égal), <=, >=. Cet ordre peut être perturbé lorsque l'expression L.P.P. contient des sous-ensembles entre parenthèses.

Exemple :  $3 * 4 - 5 + 6 = 13$   
 $3 * (4-5) + 6 = 3$

#### II.3.4.3. Les informations fournies par le module

Là encore, il nous faut distinguer les informations apparentes et celles transparentes à l'utilisateur, les plus importantes pour la poursuite d'une *session S.A.M.S.S.* étant les secondes.

##### . Les informations apparentes

Dans cette première catégorie, nous trouverons une trace de la description, en langage L.P.P., des relations du modèle.

A cette trace, sont associés différents éléments d'information :

- numérotation des lignes du texte,
- numérotation des relations,
- codes des erreurs détectées en cours d'analyse.

Toujours dans le domaine des informations visualisées, nous trouverons certains éléments statistiques concernant l'exécution

du module :

- Nombre de relations traitées.
- Nombre total de relations macrogénérées qui décrivent le modèle.
- Nombre de variables intermédiaires.

L'exemple du chapitre **IV** montre bien ces différentes informations.

### . Les informations transparentes

A partir de la description des relations d'un modèle, le module de macrogénération (MACROG) génère quatre types d'informations qui, bien qu'essentiels pour la poursuite de l'automatisation, ne présentent pas d'intérêt pour l'utilisateur non informaticien. Il s'agit

- du Fichier-Pivot,
- du Descripteur du Fichier-Pivot,
- du Fichier des entrées du module de Génération du Graphe d'Evaluation des Relations,
- du Fichier-Accès.

### Le Fichier-Pivot

Pour chaque relation du modèle, nous produisons un pavé-relation et nous le mémorisons dans le Fichier-Pivot.

Un pavé-relation est compilable et éventuellement exécutable (à condition d'avoir respecté la syntaxe et la sémantique du *Langage L.P.P.*). Il est composé :

- de l'ensemble des ordres (instructions PASCAL) non directement liés à la formulation algorithmique de la relation, c'est-à-dire l'ensemble des ordres qui assurent la liaison entre la relation et les opérandes qu'elle manipule, ainsi que l'ensemble des ordres qui assurent la gestion des accès à la Base des Opérandes du Modèle ;

- du corps de relation modifié dans ce sens que la désignation des opérandes y a été transformée. On génère en effet les codes internes des opérandes à la place des couples (Nom Générique d'Opérande, Liste d'expressions d'indices).

Ainsi, une relation que l'utilisateur aurait décrit par :

```
(*∂CALCUL IDPRIENE(V) . RELATION E33*)
FOR V:=MINV TO MAXV DO
  IDPRIENE(V):=FH2301(POLENE, IDPRIPE, V)
```

pourrait être mémorisé dans le fichier-pivot, comme deuxième pavé-relation, sous la forme suivante :

```
1  (*∂CALCUL IDPRIENE(V) . RELATION E33*)
2  ACCESS (CALCUL, 2);
3  MVC (DESCOPAC(1).PTADOPMEM, IDPRIENE_10 , LGPOINT);
4  MVC (DESCOPAC(2).PTADOPMEM, POLENE_1 , LGPOINT);
5  MVC (DESCOPAC(3).PTADOPMEM, IDPRIPE_1 , LGPOINT);
6  MVC (DESCOPAC(4).PTADOPMEM, FH2301_23 , LGPOINT);
7  FOR V:=MINV TO MAXV DO
8  IDPRIENE_10"(V):=FH2301_23"(POLENE_1", IDPRIPE_1", V);
9  IMPOPREL(2);
```

où, les lignes 2 à 6 et la ligne 9 sont les ordres non directement liés à la formulation algorithmique de la relation (ils sont cependant déduits du texte L.P.P.), et les lignes 7 et 8 correspondent au corps de relation compilable et exécutable.

#### Le Descripteur du Fichier-Pivot

Pour que les relations d'un modèle, une fois macrogénérées, puissent être réordonnées en fonction de l'ordre produit par le module de génération du graphe d'évaluation des relations du modèle, il est impératif de pouvoir les acquérir et les manipuler individuellement.

Dans ce seul but, chaque fois que nous avons produit un pavé-relation, en même temps que nous l'introduisons dans le fichier-



pivot, nous mémorisons les quatre éléments suivants :

- Numéro de la relation.
- Numéro de la page du Fichier-Pivot où sera stocké le début du *pavé-relation*,
- Adresse dans cette page du premier caractère qui correspond à un *pavé-relation*,
- Taille (exprimée en nombre d'octets) du *pavé-relation*,

qui nous aideront à générer le modèle (relations) sous sa forme exécutable.

Le fichier des entrées du module de génération du graphe d'évaluation des relations d'un modèle

Générer le graphe d'évaluation des relations d'un modèle, c'est-à-dire, déterminer l'ordre dans lequel les relations doivent être exécutées pour que les résultats produits par le modèle soient cohérents, exige (nous verrons plus loin pourquoi) la présence d'éléments d'informations qu'il n'est possible de produire automatiquement qu'en cours de macrogénération.

Ces éléments d'informations consistent en :

- le type de la relation (CC pour Calcul, CT pour contrôle).
- La nature de la relation (ST pour Statique, DYN pour Dynamique).
- La liste des codes internes des opérandes utilisés dans la relation.

Ainsi, si l'on repart de l'exemple développé au § II.4.3.2.1., l'entrée correspondante pour le module de Génération du Graphe d'Evaluation des Relations du modèle se traduira par :

CC                    ST                    IDPRIENE\_10 : POLENE\_1 : IDPRIPE\_1 : FH2301\_23 ;

### Le fichier-Accès

Nous avons justifié l'existence de la Base des Opérandes d'un Modèle en montrant qu'il n'était pas pensable de disposer simultanément en mémoire centrale de la totalité des variables du modèle. Cependant, pour exécuter une relation, il sera utile d'avoir, en mémoire centrale, les réalisations des différents opérandes qui servent au calcul ou au contrôle.

Pour cela, nous devons d'une part, acquérir l'espace correspondant en mémoire centrale, et d'autre part, y charger les réalisations des opérandes déjà évalués. Enfin, il sera nécessaire d'établir le lien entre identificateurs d'opérandes de la relation et cet espace en mémoire centrale.

C'est ce qui permet de réaliser la présence du fichier-Accès, qui pour chaque relation d'un modèle contient un enregistrement où figurent la liste des codes internes de ses opérandes.

Remarque : Il serait possible, pour minimiser l'espace de stockage des informations nécessaires à l'exécution du logiciel S.A.M.S.S., de fusionner les deux types d'informations : entrées du module de génération du graphe d'évaluation des relations, et fichier-accès.

### II.3.5. GENERATION DU GRAPHE D'EVALUATION DES RELATIONS

Les relations qui traduisent le comportement du phénomène modélisé, mettent en jeu les différents composants caractéristiques (de base ou indirect) de chaque sous-système (urbain, industriel, des transports ... pour MEDEE-3) et du système global. Certaines sont dynamiques et d'autres sont statiques, mais la notion la plus importante du type de modèles auxquels nous nous intéressons, réside dans le fait que le "système" formé par ses équations doit être totalement "récursif" [39], c'est-à-dire qu'il est hiérarchisable. En effet toute modification structurelle d'un modèle de ce type implique une modification de la "hiérarchie" des relations.

Cette hiérarchie traduit une certaine cohérence d'exécution entre les relations :

*. Toute relation  $R_c$ , en cours d'évaluation doit être évaluable, c'est-à-dire que toutes ses variables (exceptée celle qui est calculée par  $R_c$ , si  $R_c$  est une relation de calcul) doivent avoir été évaluées au préalable.*

Pour connaître cette hiérarchie, il sera utile de construire un graphe d'évaluation des relations.

Avant d'aborder les spécifications de l'élaboration de ce graphe, il nous paraît important de redire ici qu'il est essentiel de pouvoir isoler un sous-système, de pouvoir le considérer indépendamment des autres (ne serait-ce que pour en mieux cerner les inputs extérieurs où les interactions avec les autres), même si cette notion de sous-système n'apparaît pas explicitement dans les relations.

### II.3.5.1. Formalisation du problème

Les objets qui apparaissent dans un modèle et qui sont nécessaires à la construction du graphe seront décrits et formalisés, puis nous préciserons les règles générales de construction et de modification du graphe.

#### . Les objets nécessaires à la construction du graphe

Nous disposons actuellement de deux catégories d'objets (générés au cours des étapes antérieures) :

- les opérandes, décrits dans le *Dictionnaire du Modèle* et stockés dans la *Base des Opérandes du Modèle*,
- les relations qui manipulent ces opérandes et qui sont mémorisées sous forme de *pavés-relations* dans le *Fichier-Pivot du Modèle*.

Les Opérandes peuvent être distribués en 6 classes différentes :

- les données initiales : en général utilisées pour le calcul de variables dynamiques, elles représentent les conditions initiales du modèle et sont évaluées pour  $t=t_0$  ( $t$ =indice du temps),
- les données statiques : données indépendantes du temps, ou dépendantes du temps mais évaluées pour toute valeur de  $t$ , elles correspondent à des séries chronologiques qui seront intégrées à la *Base des Opérandes* du modèle avant toute résolution numérique,
- les variables dynamiques : évaluées au cours de la résolution numérique du modèle, elles sont calculées par des relations de la forme  $R(t-1,t)$ ,

- les variables statiques : valuées elles aussi au cours de la résolution numérique du modèle, les relations qui les calculent ne font intervenir que la date de simulation courante,
- les données nécessaires : elles ne correspondent pas à un type d'opérande du modèle, mais sont utilisées pour la construction du graphe,
- les inconnues : opérandes qui n'ont pas encore été introduites dans le graphe.

Tout graphe d'évaluation des relations d'un modèle en cours de construction, tout au moins la partie "matrice d'incidence du problème par rapport à ses opérandes", peut comporter les cinq premiers types d'opérandes.

Tout graphe construit ne doit plus comporter de "données nécessaires".

Remarque : un opérande peut être local, c'est-à-dire qu'il est défini dans un sous-système et utilisé par lui seul, ou global c'est-à-dire qu'il peut être utilisé dans tous les sous-systèmes y compris celui auquel il est rattaché. Ce type d'information, non exploité dans la version actuelle du logiciel, nous semble la base d'une construction modulaire d'un modèle [ 26 ] .

Les Relations sont de deux types : calcul ou contrôle :

- les relations de calcul : équations à une inconnue, qui exploitent  $N$  opérandes préalablement évalués pour en calculer un  $N+1$ ème (cet opérande est alors une "variable calculée", dynamique ou statique),

- les relations de contrôle : qui permettent de valider les réalisations calculées d'un ou plusieurs opérandes (ces opérandes doivent donc avoir été évaluées avant le contrôle). Elles sont fonction de l'opérateur de contrôle :  $<, \leq, >, \geq, =$

Remarque : Dans le même ordre d'idée que pour les opérandes, on dira qu'une relation est locale à un sous-système si elle n'utilise que des variables définies dans ce sous-système. Elle est globale si elle n'utilise que des variables globales. Elle est mixte lorsqu'elle manipule les deux types de variables, locales et globales.

. Les notations : définition des ensembles et des fonctions utilisées

Nous présentons ici les notations mathématiques (symbolisme) qui vont être utilisées dans les paragraphes suivants. Il ne s'agit pas pour nous de sacrifier à une certaine mode qui affirme que "hors de la Mathématique, il n'y a pas de Salut pour les Sciences" (!), mais plutôt d'introduire un formalisme qui permet une plus grande rigueur, et qui pourrait servir de base pour développer le logiciel S.A.M.S.S.

Soit  $r$  une relation,

On note  $O_r = \{ x_1, x_2, \dots, x_n \}$  l'ensemble des opérandes qui interviennent dans la relation  $r$  (chacun n'étant nommé qu'une seule fois)

$\forall j \in [1, n] \quad j \neq j' \implies x_j \neq x_{j'}$

$\forall j' \in [1, n]$

A tout opérande  $x$ , connu du graphe, on associe

$F_x = \{ r_i / r_i \text{ utilise } x \}$  ensemble des relations qui utilisent l'opérande  $x$

Soit  $\{ R, \text{l'ensemble des Relations} \}$  On définit une fonction de  $O$  dans  $R$  :  
 $\{ O, \text{l'ensemble des Opérandes} \}$

$$g : O \longrightarrow R$$

$$g(x_i) = r \quad \text{si } r \text{ est la relation qui calcule } x_i$$

Il s'agit bien d'une fonction puisque nous voulons vérifier, au cours de l'élaboration du graphe, qu'une variable n'est calculée que par une seule relation. D'autre part, certains opérandes (les données par exemple) n'auront pas d'image par  $g$ .

On note  $A_r$ , l'ensemble des relations dont l'évaluation doit précéder celle de la relation  $r$ , ( $A_r \in EP(R)$ )

$A_r = \{ r_2, r_4, r_7 \}$  signifie que la relation  $r$  ne pourra être évaluée qu'après l'évaluation de  $r_2$ ,  $r_4$  et  $r_7$ .

C'est l'ensemble de ces éléments  $A_r$  qui nous permettra de déterminer l'ordre dans lequel les relations pourront être appelées pour la résolution numérique. Si  $A_r = \{ \phi \}$  alors la relation est évaluable sans préalable. Elle est de niveau 1.

L'ensemble  $O$  des opérandes est partitionné en quatre sous-ensembles :  
 $V$  = ensemble des variables (statiques ou dynamiques),  $D$  = ensemble des données (statiques ou initiales),  $DN$  = ensemble des données nécessaires, et  $I$  = ensemble des inconnues. Si  $\text{card}$  désigne le cardinal d'un ensemble, on peut écrire :

$$\text{card}(O) = \text{card}(V) + \text{card}(D) + \text{card}(DN) + \text{card}(I)$$

. Les règles générales de construction du graphe

L'élaboration du graphe consiste pour l'essentiel à insérer une nouvelle relation dans l'ensemble des relations existantes, à insérer les nouveaux opérandes (lorsqu'il y en a) de cette relation dans l'ensemble des opérandes déjà existants, et à mettre à jour la description des relations et des opérandes préexistants pour tenir compte de l'adjonction de la nouvelle relation.

Définition du vecteur de connaissance : Soit  $r$ , la relation à insérer dans le graphe, et  $O_r$ , l'ensemble des  $n$  opérandes  $x_1, x_2 \dots x_n$  qui y sont utilisés. On lui associe le  $n$ -uplet, noté  $e$ , défini par :

$\forall i \in [1, n] \quad e(x_i) = 0$  si  $x_i$  est une inconnue

$e(x_i) = 1$  si  $x_i$  est une variable, une donnée, ou une donnée nécessaire

et on notera  $e = \mathbf{1}$ , le  $n$ -uplet dont chacune des composantes  $e(x_i)$  vaut 1.

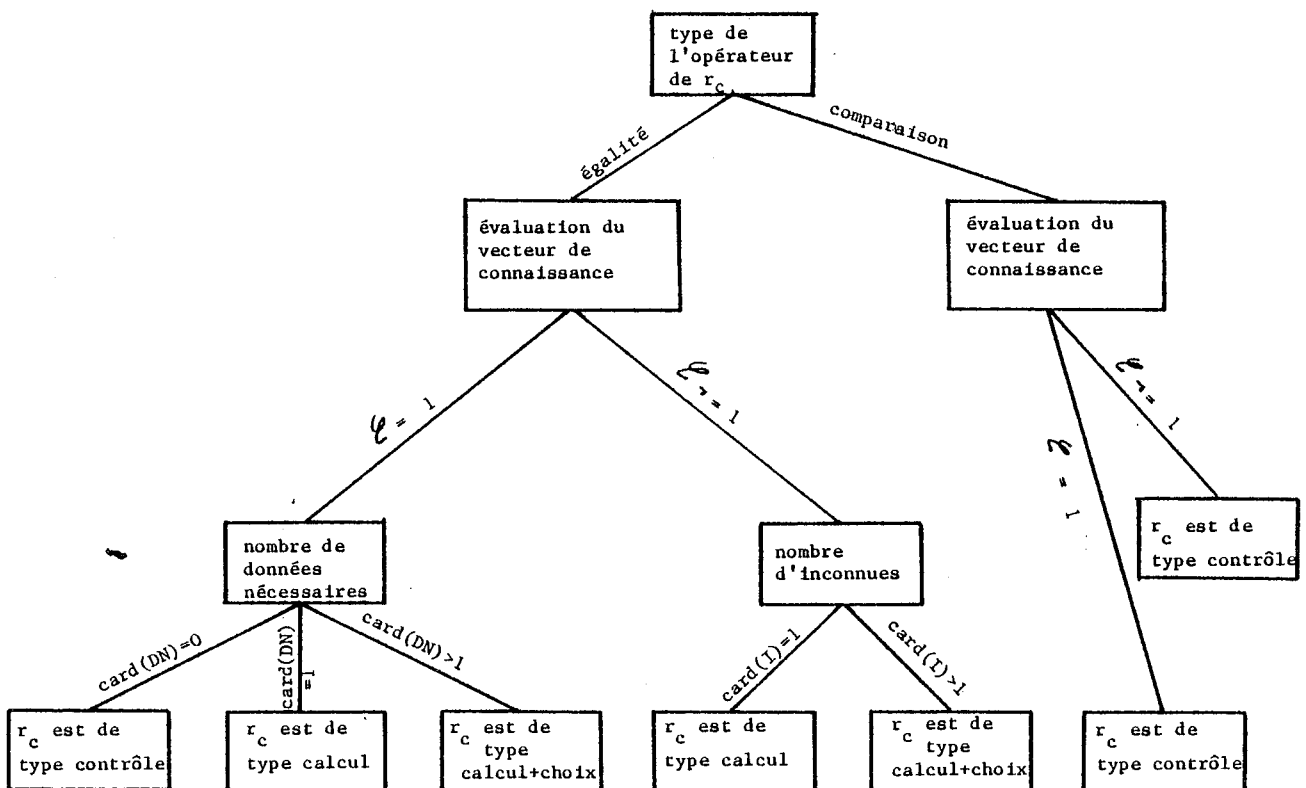
L'insertion de la relation  $r$  dans le graphe en cours d'élaboration dépend d'une part du type de la relation et d'autre part de l'état du vecteur de connaissance qui lui est associé :

- $r$  n'est pas du type égalité :  $r$  est une relation de contrôle. Tous ses opérandes doivent être évalués avant.  
Si  $e = \mathbf{1}$  alors tous les opérandes, de type "inconnue" sont transformés en opérandes de type "donnée nécessaire".



-  $r$  est du type égalité : nous distinguerons cinq cas :

1.  $e = 1$  et  $\text{card}(\text{DN}) = 0$   $r$  est une relation de contrôle.
2.  $e = 1$  et  $\text{card}(\text{DN}) = 1$   $r$  permet de calculer cette donnée nécessaire qui est alors transformée en variable calculée (statique ou dynamique).
3.  $e = 1$  et  $\text{card}(\text{DN}) > 1$   $r$  permet de calculer l'une des données nécessaires. Il faut donc déterminer celle qui sera transformée en variable calculée.
4.  $e = 1$  et  $\text{card}(\text{I}) = 1$   $r$  permet de calculer cette inconnue.
5.  $e = 1$  et  $\text{card}(\text{I}) > 1$   $r$  permet de calculer l'une des inconnues, les autres seront transformées en données nécessaires.



Les différents schémas d'insertion d'une relation dans le graphe d'évaluation.

Les actions : Nous donnons ici la liste des actions à entreprendre dans chacun des cas d'insertion d'une relation. Pour toute relation, nous commençons par :

(A1)

reconnaissance du type de la relation courante  $r_c$

(A2)

évaluation du vecteur de connaissance associé à  $r_c$   
celle-ci fournit les cardinaux des 4 types d'opérandes de  $r_c$

-  $r_c$  n'est pas de type égalité et  $e = 1$  : les opérations à effectuer consistent à marquer tous les opérandes utilisés par  $r_c$  et à évaluer l'élément  $A_{r_c}$

(A3)

$\forall i \in [1, n]$  tel que  $x_i \in O_{r_c}$  alors  $F_{x_i} \leftarrow F_{x_i} \cup \{r_c\}$

(A4)

$A_{r_c} = \bigcup_{i=1}^n \{g(x_i)/x_i \in O_{r_c}\}$

Remarque : Le symbole " $\leftarrow$ " qui apparaît en (A3) symbolise la mise à jour de la partie gauche par la partie droite.

Si  $g(x_i)$  n'est pas définie, nous faisons l'hypothèse qu'il existe une "relation blanche" à évaluer avant toute autre relation de R. Au lieu de définir une fonction de O dans R on définira une application de O dans  $R \cup \{\text{relation blanche}\}$  dans laquelle la "relation blanche" est l'image par défaut de tout  $x_i \in O$ . Cette remarque n'est pas uniquement destinée à satisfaire les puristes puisqu'elle correspond à l'hypothèse retenue pour la définition de l'ordre d'évaluation des relations.

- $r_c$  n'est pas de type égalité et  $e = 1$  : il faut introduire les "inconnues" dans le graphe (A5), puis effectuer les actions A3 et A4.

(A5)

$$\forall i \in [1, n] \text{ tel que } \begin{cases} x_i \in O_{r_c} \\ \text{et} \\ x_i \text{ est un opérande} \\ \text{de type "inconnue"} \end{cases} \text{ alors } g(x_i) = 0$$

- $r_c$  est de type égalité et  $e = 1$  avec  $\text{card}(\text{DN}) = 0$  : on va effectuer A3 et A4.

- $r_c$  est de type égalité et  $e = 1$  avec  $\text{card}(\text{DN}) = 1$  : la relation  $r_c$  calcule l'opérande de type "donnée nécessaire" ( $x_k$ ). Il faut marquer tous les opérandes utilisés par  $r_c$  (action A3), modifier le type de  $x_k$  (action A6), évaluer l'élément  $A_{r_c}$  (action A7), et mémoriser le fait que toute relation utilisant  $x_k$  doit être évaluée après  $r_c$  (action A8).

(A6)

$$g(x_k) = r_c$$

(A7)

$$A_{r_c} = \bigcup_{\substack{i=1 \\ i \neq k}}^n \{ g(x_i) / x_i \in O_{r_c} \}$$

(A8)

$$\forall j / \begin{cases} r_j \in F_{x_k} \\ \text{avec } r_j \neq r_c \end{cases} \text{ alors } A_{r_j} \leftarrow A_{r_j} \cup \{r_c\}$$

- $r_c$  est de type égalité et  $e = 1$  avec  $\text{card}(\text{DN}) > 1$  : Avant d'effectuer les opérations A3, A6, A7 et A8 comme dans le cas précédent, il est nécessaire de choisir l'opérande de type "donnée nécessaire" que l'on va transformer en opérande de type "variable calculée".

(A9)

choix de l'opérande dont on va modifier le type

- $r_c$  est de type égalité et  $e = 1$  avec  $\text{card}(I) \geq 1$  : On va utiliser, dans l'ordre, les actions A5, A9, A3, A6 et A7, définies plus haut.

Les primitives de construction du graphe : Chacune des actions spécifiées précédemment a été traduite en une primitive de construction du graphe. Nous en avons ajouté une dernière, notée P10, qui consiste à repérer, une fois le graphe construit, les opérandes de type "donnée nécessaire" et à les transformer en opérande de type "donnée" (initiale ou statique). Nous supposons en effet a priori, que le modèle conceptuel est correctement formulé c'est-à-dire qu'il y a adéquation parfaite entre les données dont dispose l'utilisateur et les opérandes définis comme des données (initiales ou statiques) lors de la construction du graphe. Cette hypothèse nous permettra alors de vérifier la cohérence entre la description et l'utilisation d'un composant caractéristique du phénomène étudié (voir la fonction remplie par l'étape suivante du logiciel S.A.M.S.S.).

Détermination de l'ordre d'évaluation des relations : Nous disposons maintenant d'un graphe où sont mémorisées l'ensemble des informations qui vont nous permettre de déterminer l'ordre dans lequel il faudra exécuter les relations du modèle.

Rappelons une fois encore, que nous voulons exclure les systèmes d'équations ou d'inéquations simultanées à  $n$  inconnues ( $n > 1$ ). Si le besoin s'en faisait sentir il serait sans doute possible de revenir sur cette option tout en conservant l'acquis actuel.

L'apparition d'un système d'équations simultanées d'ordre  $n$  ( $n > 1$ ) aura pour conséquence de bloquer l'algorithme chargé de déterminer le niveau d'évaluation des relations. Considérons l'exemple suivant :

Soient  $r_i$  et  $r_j$ , deux relations du modèle  
 $C$  et  $D$  des opérands de type "donnée"  
 $A$  et  $B$  des opérands de type "inconnue"

si les relations  $r_i$  et  $r_j$  sont définies par :  $(r_i) \quad A = B + C$   
 $(r_j) \quad B = A/D$

alors  $r_j \in A_{r_i}$  et  $r_i \in A_{r_j}$

Nous sommes en présence d'un système d'équations simultanées d'ordre 2 (des systèmes d'ordre quelconque peuvent apparaître) et il n'est pas possible de choisir l'une ou l'autre des relations pour casser le cycle.

- La notion de niveau d'évaluation peut se définir de façon récurrente comme suit :

$$NIV_1 = \{ r_i / A_{r_i} = \phi \}$$

$$NIV_2 = \{ r_i / \forall j, \text{ si } r_j \in A_{r_i} \text{ alors } r_j \in NIV_1 \}$$

$$NIV_N = \{ r_i / \forall j, \text{ si } r_j \in A_{r_i} \text{ alors } \left\{ \begin{array}{l} r_j \in NIV_1 | NIV_2 \dots | NIV_{N-1} \\ \text{] au moins une } r_j \in NIV_{N-1} \end{array} \right\} \}$$



Marquage des  $r_j \in A_{r_i}$  évaluées au niveau p :

$$\left\{ \begin{array}{l} \forall i \in [1, n] / V(r_i) = p \\ \forall j \in [1, n] / j \neq i V(r_k) = 0 \\ p \in [1, m] \end{array} \right\} \text{ si } r_i \in A_{r_k} \text{ alors } W(r_i) = p$$

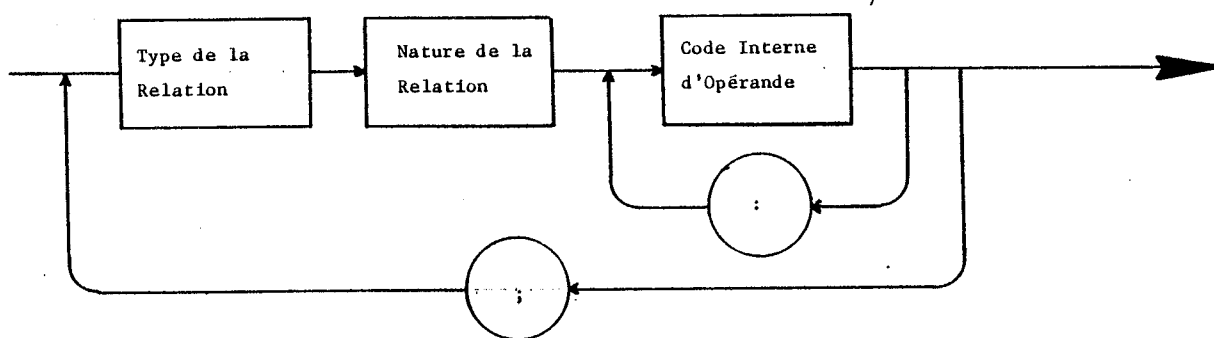
Repérage des systèmes d'équations d'ordre supérieur à 1 :

$$\left\{ \begin{array}{l} \forall i \in [1, n] / V(r_i) = 0 \\ p = \text{niveau d'évaluation} \\ \text{courant} \end{array} \right\} \text{ si } \left\{ \begin{array}{l} p=1 \text{ et } A_{r_i} \neq \emptyset \\ p \in [2, m] \text{ et } \exists r_j \in A_{r_i} / W(r_j) = 0 \end{array} \right.$$

### II.3.5.2. Les entrées du module de Génération du Graphe d'Evaluation

Nous ne demandons pas d'intervention directe de l'utilisateur, ici. En effet, les entrées du module de Génération du Graphe d'Evaluation des Relations d'un modèle ont été générées et mémorisées lors de l'étape précédente (module de macrogénération des relations).

Nous rappelons leur structure :



en précisant que si, au niveau des entrées du module de macrogénération des relations d'un modèle, nous exigeons de l'utilisateur qu'il précise le type de ses relations, c'est en fait le module de génération du graphe d'évaluation des relations qui en est le seul bénéficiaire. En effet, si cette information est ignorée par ailleurs, elle est ici utilisée pour déterminer l'ordre d'évaluation des relations.

Toute relation de CALCUL d'un opérande doit être exécutée avant l'ensemble des relations qui utilise cet opérande.

Toute relation de CONTROLE doit être exécutée après l'ensemble des relations qui évaluent les opérandes qui y sont utilisés.

Remarque : Il aurait été possible, par une analyse sémantique du corps de relation (dans le module MACROG), de générer automatiquement l'information : *Type de la Relation*. Nous avons choisi une solution, il est vrai plus contraignante pour l'utilisateur, mais beaucoup plus rapide.

### II.3.5.3. Les informations délivrées par le module

Nous distinguons une fois encore les deux types d'informations (apparentes et transparentes à l'utilisateur) produites par ce module.

#### Les informations apparentes

Ce sont celles qui permettent à l'utilisateur d'intervenir, si nécessaire, pour modifier l'écriture de son modèle, de telle sorte qu'il puisse être exécutable via SAMSS. Actuellement, elles se résument à :

- . Une trace des inputs (relations réduites à leurs opérandes) sur laquelle, en face de chaque relation, on trouvera un numéro d'identification et/ou éventuellement les codes des erreurs détectées en cours de traitement.
- . Un message terminal qui indique si le logiciel a pu déterminer l'ordre d'évaluation de toutes les relations du modèle. Lorsque ce message est positif, nous précisons la profondeur du graphe, c'est-à-dire le nombre de niveaux d'évaluation. Dans le cas inverse, c'est le niveau de blocage qui sera précisé.



### Les informations transparentes

Informations codées, leur visualisation ne présente que peu d'intérêt pour l'utilisateur. Il s'agit en effet des deux matrices d'incidences du modèle, par rapport à ses opérands et par rapport à ses relations.

#### La matrice d'incidence du modèle par rapport à ses opérands

Cette matrice élaborée au fur et à mesure de l'insertion des relations dans le graphe, fournit pour chaque opérande les éléments suivants :

- identificateur de la relation qui le calcule (dans la mesure où elle existe) ;
- ensemble des relations qui l'utilisent ;
- identificateur du sous-modèle déclaratif ;
- type de l'opérande : donnée nécessaire ou variable calculée.

Chacun de ces éléments sont déduits du texte de *description des relations*. Il sera donc possible de procéder à des contrôles de cohérence portant sur la déclaration d'un objet et son utilisation.

#### La matrice d'incidence du modèle par rapport à ses relations

Si l'évaluation du graphe s'est achevée correctement, c'est-à-dire si le modèle ne comporte pas de système d'équations simultanées, l'examen de cette matrice nous permettra de manipuler les "*pavés-relations*" générés par le module de macro-génération des relations. Nous pourrons en effet les ordonner de façon à ce que leur hiérarchie d'évaluation soit respectée, et partant de là, générer le module de Résolution Numérique du modèle, codification en langage Pascal de ce modèle.

Avec l'hypothèse inverse (modèle contenant des variables de bouclage), la matrice d'incidence du modèle par rapport à ses relations, sera d'une aide précieuse pour localiser le ou les systèmes d'équations simultanées. En effet, pour chaque relation cette matrice contient les

éléments d'informations suivants :

- son niveau d'évaluation (dans la mesure où il a pu être déterminé)
- son type (calcul ou contrôle)
- sa nature (statique ou dynamique)
- l'identificateur du sous-modèle auquel elle est rattachée (quand il a été mentionné)
- l'ensemble des relations qui doivent être évaluées avant elle
- l'ensemble des variables qu'elle utilise.

### II.3.6. CONTROLE DE LA COHERENCE DESCRIPTION-UTILISATION DES VARIABLES D'UN MODELE

Tout effort de modélisation implique au départ une certaine imperfection. Il n'est, par conséquent, nullement surprenant de constater que ce phénomène se répercute au niveau final du processus : l'automatisation du modèle. En effet, si l'utilisateur connaît le phénomène qu'il modélise, il n'est pas exceptionnel pour autant, que dans l'ensemble des relations qui traduisent son comportement, il en omette (dans ses premières formulations) une, voire plusieurs. De même, un identificateur d'opérande peut se libeller incorrectement, un opérande décrit comme une donnée peut en fait être une variable calculée (et inversement) ... Ces quelques remarques seront d'autant plus vraies que le nombre de relations du modèle est important.

Le module de contrôle de la cohérence "description-utilisation" des variables d'un modèle, qui, en rapprochant le Dictionnaire du Modèle et la Matrice d'incidence du modèle par rapport à ses opérandes, établit une liste des incohérences probables, permet à l'utilisateur d'intervenir sur l'un ou l'autre, et d'assurer ainsi la cohérence de son modèle avant la résolution numérique.

#### II.3.6.1. Les entrées du module de contrôle de la cohérence (CONTRO)

L'utilisateur, de nouveau, n'a pas à intervenir à ce niveau. En effet, les éléments nécessaires aux contrôles de cohérence figurent sous forme codée, d'une part dans le Dictionnaire du Modèle (généré par le module CREDIC), et d'autre part dans le Graphe d'Evaluation des Relations ou plus précisément dans la Matrice d'Incidence du Modèle par rapport à ses opérandes.

Rappelons que le Dictionnaire du Modèle comporte pour tout opérande déclaré, la codification des éléments d'information suivants :

- code interne de l'opérande
- nature déclarée de l'opérande (Donnée Initiale, Donnée Exogène, Donnée de Scénario, Variable Calculée).

Rappelons d'autre part que la *Matrice d'Incidence du modèle par rapport à ses opérandes* fait apparaître, pour chacun des opérandes utilisés par les relations, les éléments suivants :

- code interne de l'opérande, éventuellement postfixé par l'une ou l'autre des deux chaînes de caractères TP ou TC
- nature calculée de l'opérande (Donnée, Variable Calculée), obtenue par construction du *Graphe d'Evaluation des relations*.

### II.3.6.2. Les incohérences probables détectées

Elles portent sur la nature des opérandes. Or, nous venons de voir que les ensembles de qualificatifs utilisés pour indiquer la nature des opérandes, suivant qu'elle est décrite ou calculée, ne sont pas identiques. Pour préciser les différentes classes de Données (nature calculée), il aurait été nécessaire d'activer une analyse sémantique du *corps de relation*, ce que nous avons refusé. D'autre part, les informations qui se rapportent à la nature d'un opérande dont les réalisations sont fonction de la date de simulation peuvent être éclatées en deux endroits dans le Graphe d'Evaluation des Relations (cas des opérandes utilisés à la fois à l'instant T et à l'instant T-1), d'où l'apparition d'un "postfixe" accolé au code interne. Il sera nécessaire dans ce cas de concaténer ces informations avant de les comparer à celles contenues dans le *Dictionnaire du Modèle*.

En utilisant le degré d'incohérence comme critère de classification, nous avons pu définir quatre catégories d'anomalies : les erreurs fatales, les ambiguïtés qui peuvent entraîner des erreurs graves, les warnings et enfin les redondances d'informations.

#### . Les erreurs fatales

Si l'utilisateur n'intervient pas, soit au niveau de la description de l'opérande mis en cause, soit au niveau de la description des relations qui l'utilisent, la résolution numérique du modèle produira à coup sûr, des résultats erronés. C'est le cas lorsqu'un opérande est décrit comme une variable calculée et que la description des relations du modèle ne comporte pas de relation de calcul corres-

pondante. De même, un opérande décrit comme une Donnée Initiale peut, en fonction de son utilisation, provoquer ce type d'erreur.

Exemple 1 : En reprenant la description des composants caractéristiques de la page 36, mais en modifiant la description du nombre de ménages urbains de la manière suivante :

```
MENURB(T) REEL; (*NB MENAGES URBAINS*)
```

et en lui associant la description des relations ci-dessous :

```
(*SYSTEME URBAIN : CARACTERISTIQUES SOCIO-ECONOMIQUES*)

(*@CALCUL PPU (I,T). RELATION U1*)
  FOR I:=MINI TO MAXI-1 DO
    PPU(I,T):=PPU(I,T-1)+TCPFU(I);

(*@CALCUL TCPFU(I). RELATION U2*)
  FOR I:=MINI TO MAXI-1 DO
    TCPFU(I):=FH3101(POLAME,I);

(*@CALCUL PPU(K,T). RELATION U3*)
  FOR K:=MINK TO MAXK DO
    PPU(K,T):=PMEN(K,T);

(*@CALCUL NLOGDEM(T). RELATION U8*)
  NLOGDEM(T):=FH3102(POLLOG,T);

(*@CONTROLE PPU(I,T). RELATION U8-1*)
  $VARINTR:=0;
  FOR I:=MINI TO MAXI-1 DO
    $VARINTR:=$VARINTR + PPU(I,T);
  CONTROL ($VARINTR-1, EG, 0.001, CONTINUE, 2008);

(*@CONTROLE PPU(K,T). RELATION U9*)
  $VARINTR:=0;
  FOR K:=MINK TO MAXK DO
    $VARINTR:=$VARINTR + PPU(K,T);
  CONTROL($VARINTR-1, EG, 0.005, CONTINUE, 2009);

(*@CALCUL MENURB(K,T). RELATION U10*)
  FOR K:=MINK TO MAXK DO
    MENURB(K,T):=PPU(K,T) MENURB(T);

(*@FIN*)
```

Le module CONTRO détectera une erreur fatale : il est clair que le calcul du nombre de ménages urbains par classe de revenus ne pourrait que délivrer des résultats folkloriques.

Exemple 2 : En reprenant le même exemple de base, si l'utilisateur modifie la description du nombre de ménages urbains de la façon suivante :

MENURB(T) REEL DONIN; (\*NB MENAGES URBAINS\*)

sans modifier la description des relations, le contrôle de cohérence annoncera une erreur fatale puisque MENURB(T) est utilisé quelle que soit la date de simulation.

Par contre nous verrons plus loin que cette description peut être ambiguë si la description des relations du modèle est modifiée. Elle peut en effet soit être cohérente, soit être insuffisante en fonction de la formulation des relations qui utilisent MENURB(T).

. Les ambiguïtés pouvant mener à des erreurs graves

Le degré de finesse de l'analyse d'une relation du modèle ne nous permet pas, dans ce cas, de déterminer si la résolution numérique du modèle peut se dérouler correctement. Le modélisateur, après contrôle des *corps de relations* qui utilisent l'opérande incriminé décidera d'autoriser ou non l'exécution de la suite du logiciel.

Ainsi, lorsqu'un opérande, fonction de la date de simulation, est décrit comme une Donnée Initiale et que la description des relations du modèle ne comporte pas de relation de calcul correspondante, seul l'utilisateur pourra indiquer si l'opérande devait être décrit comme une Donnée de Scénario ou comme une Donnée Exogène, si, plus simplement, la relation de calcul a été omise, ou encore si l'opérande doit être décrit et utilisé comme un objet indépendant de la date de simulation (seul cas où l'exécution du logiciel peut se poursuivre sans problème).

De même si un opérande dont les réalisations dépendent de la date de simulation est décrit comme une variable calculée, que la relation de calcul correspondante existe dans le modèle, mais qu'une relation au moins utilise les réalisations de cet opérande à la date de simulation qui précède la date courante, alors nous ne savons pas distinguer si cette utilisation appelle une description de l'opérande comme Donnée Initiale ou si cela est inutile.

Exemple 1 : Dans le deuxième exemple (§ erreurs fatales) nous disions que la description de  $MENURB(T)$  pourrait être :

$MENURB(T)$  REEL DONIN;

Si la description de la relation de calcul de  $MENURB(K,T)$

est transformée en faisant apparaître uniquement  $MENURB(T-1)$

alors, il ne nous est pas possible de déterminer s'il y a erreur ou non, sans analyser les *corps de relations* qui utilisent  $MENURB(T-1)$ . Ainsi, si le texte de la seule relation qui utilise le nombre de ménages urbains est le suivant :

```

1   (*@CALCUL MENURB(K,T).*)
2       FOR K:=MINK TO MAXK DO
3           IF T=TO+1 THEN MENURB(K,T):=PPU(K,T)*MENURB(T-1)
4           ELSE MENURB(K,T):=MENURB(K,T-1);

```

alors la résolution numérique peut se faire sans problème.

Par contre, si on remplace les lignes 3 et 4 par :

$MENURB(K,T) = PPU(K,T) * MENURB(T-1)$

alors la résolution numérique ne pourra s'effectuer sans problème.

#### . Les warnings

Il s'agit d'ambiguités sans conséquences graves sur la résolution numérique du modèle. On les signalera par exemple lorsqu'un opérande, fonction de la date de simulation, a été décrit comme une Donnée Initiale, qu'il existe une relation de calcul de cet opérande et qu'au moins une relation du modèle exploite les réalisations de cet opérande à la date de simulation qui précède la date de simulation courante. On ne peut pas dire, en effet, si la connaissance des réalisations à la date de début de simulation est nécessaire ou pas cela dépend de la formulation de l'ensemble des relations qui utilisent l'opérande.

Exemple : Si nous reprenons l'exemple développé au § précédent et que nous modifions la description de l'opérande nombre de ménages urbains par classe de revenu en :

MENURB(K,T) REEL DONIN;

nous pouvons affirmer que cette précision n'était pas nécessaire puisque pour la date de début de simulation nous n'utilisons pas les réalisations de l'opérande. Si par contre nous remplaçons les lignes 3 et 4 de la relation de calcul par :

MENURB(K,T):=MENURB(K,T-1);

alors il est nécessaire de connaître les réalisations de l'opérande pour la date de début de simulation, sous peine de résolution numérique folklorique.

. Les redondances d'informations

Nous regroupons sous ce vocable, l'ensemble des cas où l'utilisateur fournit des informations précises sur les opérandes, mais où ces informations ne sont pas utiles pour la résolution numérique du modèle. C'est le cas lorsque l'opérande est décrit comme une Donnée Exogène ou comme une Donnée de Scénario, et qu'il existe une relation qui calcule cet opérande dans le modèle (seules les réalisations à la date de début de simulation pourraient être utiles, dans certains cas).

De même, un opérande décrit comme une Donnée Initiale peut très bien être utilisé sans que l'on exploite ses réalisations initiales.

Exemple : Reprenons l'exemple 1 du § erreurs fatales, en ajoutant la description des variables suivantes :

TMEN(T) REEL DONEX; (\*POURCENTAGE DES MENAGES/POPULATION TOTALE\*)

POPTOT(T) REEL DONEX; (\*POPULATION TOTALE\*)

TURB(T) REEL DONEX; (\*TAUX D'URBANISATION\*)

et la relation suivante :

(\*CALCUL MENURB(T). NOMBRE DE MENAGES URBAINS\*)

MENURB(T):=POPTOT(T)\*TMEN(T)\*TURB(T);



La description de MENURB(T) aurait pu alors se réduire à :  
 MENURB(T) REEL;  
 mais le maintien en Donnée Exogène n'empêcherait pas une  
 résolution numérique correcte.

### II.3.6.3. Les informations fournies par le module

Les anomalies présumées, que le module de contrôle de la cohérence *Description-utilisation* des variables d'un modèle, détecte en comparant le *Dictionnaire du Modèle* et la *matrice d'incidence du modèle par rapport à ses opérands* sont éditées. Pour tout opérande incriminé nous précisons :

- son identificateur (suivi de la liste d'indices qui le qualifie),
- le type de l'incohérence présumée,
- le code de l'erreur
- la liste de l'ensemble des relations qui l'utilisent.

Remarque : Lorsque le module constate plus d'une anomalie pour un même opérande nous n'éditions que la plus grave.

L'exemple présenté et développé au chapitre IV permet de bien visualiser les différentes formes d'informations fournies par le module de contrôle.

### II.3.7. GENERATION DE LA FORME EXECUTABLE D'UN MODELE

A partir des différents éléments d'information, élaborés au cours des modules précédents du logiciel SAMSS, nous sommes en mesure de générer la forme exécutable du modèle. Pour cela, il faut encore activer les fonctions suivantes :

- génération automatique des éléments qui permettront les contrôles sémantiques,
- génération de la fonction qui, à un niveau d'évaluation donné, associe la liste des relations de contrôle et/ou des relations de calcul, exécutables à ce niveau,
- réordonnancement du Fichier-Pivot, c'est-à-dire manipulation des *pavés-relations* de façon à les présenter à l'exécution, dans le bon ordre,
- compilation du texte réordonné, qui permet entre autres, de détecter les éventuelles erreurs syntaxiques et/ou les erreurs sémantiques.

Si aucune erreur n'est signalée, l'ensemble de ces opérations permettent d'aboutir à la forme codée et exécutable du modèle.

#### II.3.7.1. Les entrées du module de génération de la forme exécutable (REORDO)

Le modèle, dans sa forme macrogénérée et réordonnée est un programme en langage PASCAL, que l'on soumet au compilateur. Celui-ci vérifie alors la syntaxe et la sémantique du modèle, et génère son code exécutable.

Pour être en mesure de procéder aux contrôles sémantiques, le langage PASCAL exige qu'un objet soit déclaré selon des normes précises, dans le module qui l'utilise.

Ces déclarations consistent en :

- une définition de classes d'objets appelés TYPE
- une définition de lien objet-classe d'objets assurée dans la section VAR.

Elles permettront par exemple de vérifier que dans toute affectation, la partie droite est de type compatible avec celui de la partie gauche, que toute variable de contrôle d'une instruction itérative FOR est bien de type ENTIER, ... (cf. § II.3.4.).

La présence du Dictionnaire du Modèle en entrée du module de *Génération de la forme Exécutable* d'un modèle est donc justifiée par le fait que son parcours permettra de générer automatiquement les deux éléments TYPE et VAR indispensables à ce genre de contrôle.

Générer le programme PASCAL qui représente le modèle, consiste à créer un fichier dans lequel seront concaténés :

- les déclarations (au sens défini ci-dessus),
- les pavés-relations réorganisés selon leur ordre d'exécution,
- les fonctions qui assurent la gestion de la mémoire dynamique et l'accès aux réalisations des opérandes. Ces fonctions font partie intégrante du logiciel SAMSS et sont indépendantes du modèle.

Après avoir calculé, dans le module de génération du graphe d'évaluation des relations d'un modèle, le niveau auquel chacune des relations doit être évaluée, il nous reste à extraire, un à un, les pavés-relations mémorisés dans le Fichier-Pivot, pour les introduire dans ce nouveau fichier. Pour réaliser ce transfert, nous devons disposer d'une part du Fichier-Pivot et de son Descripteur, et d'autre part du Graphe d'Evaluation des relations.

En résumé, l'ensemble des informations nécessaires pour activer le module de génération de la forme exécutable d'un modèle, ne demande aucune intervention de l'utilisateur.

### II.3.7.2. Les informations produites par le module

Deux cas de figure doivent être envisagées. En effet, si le programme PASCAL qui correspond au modèle (nous venons de le générer) a pu être compilé sans problème, la quasi-totalité des informations produites ne seront pas apparentes. Dans le cas contraire, des informations permettant de corriger le texte de Description des Relations du Modèle seront éditées.

#### . La forme exécutable du modèle

L'analyse par le compilateur PASCAL, du fichier décrit en II.3.7.1., n'a révélé ni erreur de syntaxe, ni erreur de sémantique. Le modèle sous sa forme codée et exécutable a été généré et mémorisé. Aucune modification n'est nécessaire.

Les informations produites sont donc :

- Le texte codé du modèle (load module), non visualisé.
- Le message indiquant que le module de Génération de la forme Exécutable d'un Modèle s'est achevé sans incident.

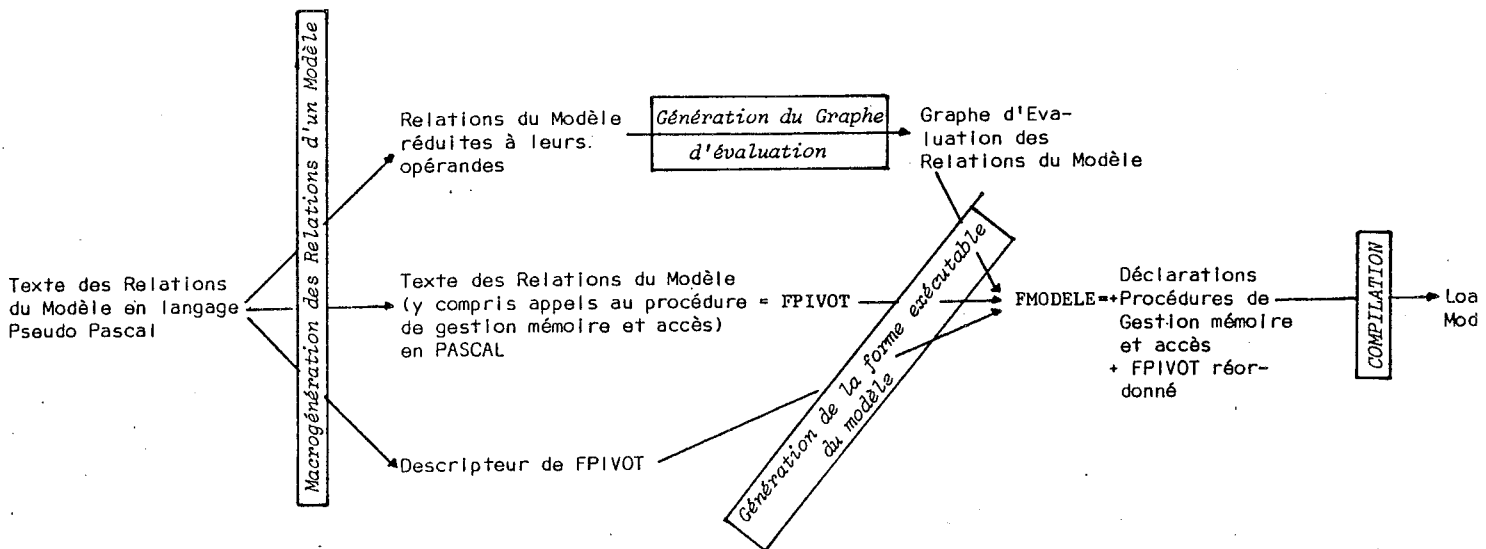
#### . Les erreurs détectées par le module

L'analyse, par le compilateur PASCAL, du texte contenu dans le fichier que nous avons généré, a permis de détecter une ou plusieurs erreurs de syntaxe et/ou de sémantique. Nous n'avons pas pu passer à la forme codée exécutable du modèle. Nous éditons alors le contenu du fichier, accompagné des messages d'erreurs correspondants.

Remarque : Si la lecture d'un texte L.P.P. est aisée pour l'utilisateur, il en va tout autrement du texte soumis au compilateur PASCAL. L'utilisateur sera alors contraint d'avoir recours à l'informaticien (formé aux techniques de programmation utilisées) pour pouvoir exploiter les informations qui lui sont délivrées sous la forme de messages d'erreurs du compilateur PASCAL.

Dans une étape ultérieure du développement du logiciel SAMSS, nous pourrions envisager d'exprimer ces mêmes informations par rapport au texte L.P.P. des relations.

En résumé, le module de génération de la forme exécutable d'un modèle génère et contrôle le programme PASCAL correspondant au modèle. Le texte codé au cours de la compilation est alors exécutable. Par le schéma ci-dessous, nous exprimons les différentes modifications que subit le texte des relations de l'utilisateur.



### II.3.8. INITIALISATION DE LA BASE DES OPERANDES D'UN MODELE

A différentes reprises nous avons eu l'occasion de parler des opérandes d'un modèle et de leur nature : données initiales, données statiques (exogènes, de scénario, fonctions tabulaires); variables dynamiques, variables statiques ... Nous avons évoqué leur description (*Dictionnaire du Modèle*), leur utilisation (*les relations du Modèle*), mais nous n'avons pas encore abordé la gestion de leurs "réalisations", si ce n'est sous l'angle spécifique de l'organisation physique de l'espace où elles seront stockées (*Base des Opérandes du Modèle*). Ce qui nous intéresse à ce stade concerne uniquement les opérandes de type "donnée". Les modèles du type de MEDEE-3 (technico-économiques très désagrégés) font appel à un nombre relativement important de données techniques et économiques sur lesquelles il est nécessaire de stipuler des hypothèses.

Parmi ces données, nous considérerons les exogènes pures dont le champ de variation est relativement étroit sur l'horizon considéré, soit parce qu'elles ont une très grande inertie, soit parce qu'elles correspondent à des tendances de très longue période. Les autres, dont le champ de variation est relativement ouvert, et pour lesquelles les hypothèses stipulées sont susceptibles de changer à chaque nouvelle application du modèle, seront appelées données de scénario. Pour élaborer les hypothèses, on (c'est-à-dire les économistes) aura recours à la fois aux variables quantitatives (séries chronologiques) et aux descripteurs qualitatifs qui spécifient les grandes orientations (choix de politiques ...).

Certains logiciels généraux d'aide à la modélisation, comme MODULECO [40] [41] ou TROLL, possèdent une fonction "manipulation de données" qui est en fait un véritable Système de Gestion de Base de Données (S.G.B.D.). Pour nous, même si certains problèmes sont identiques : stockage de documentation sur les données (à peine abordé dans le logiciel S.A.M.S.S.), protection de l'accès à certaines données (non abordé dans S.A.M.S.S.), éditions ..., dans le cas de la gestion d'un très vaste ensemble de données et dans celui qui nous préoccupe, à savoir, la gestion de séries chronologiques nécessaires à l'"amorçage"

de la Résolution Numérique d'un Modèle, la différence d'ampleur et même de nature, des problèmes qui se posent dans les deux cas ne justifie pas que l'on en arrive aux S.G.B.D. Nous avons préféré la solution classique de gestion d'un fichier d'Initialisation.

### II.3.8.1. Le module d'Initialisation de la Base des Opérandes : module INITBD

Nous avons pu voir, au cours de l'élaboration du *Dictionnaire du Modèle*, qu'un objet de *type opérande* pouvait être décrit :

soit comme une variable calculée,  
 soit comme une donnée initiale,  
 soit comme une donnée de scénario,  
 soit encore comme une donnée exogène.

D'autre part, après analyse du texte de *Description des Relations* du modèle (texte L.P.P.), nous avons pu déduire la nature de ces opérandes et assurer la cohérence *description-utilisation*.

Enfin, l'espace en mémoire annexe, suffisant pour stocker toutes les réalisations de tous les opérandes du modèle est disponible. La *Base des Opérandes du Modèle* a été formatée par exécution du module de format de la Base des Opérandes d'un Modèle.

Il est donc possible d'intégrer les données dans la *Base des Opérandes* du Modèle et de vérifier que l'utilisateur a bien fourni l'intégralité des informations nécessaires à la résolution numérique. C'est le module d'Initialisation de la Base des Opérandes d'un modèle qui est chargé de cette fonction.

#### Rappels

Un opérande de nature DONIN (donnée initiale) est un opérande dont l'utilisateur doit fournir les réalisations pour la date de début de simulation et dont le calcul et/ou l'utilisation au delà de cette date nécessite la connaissance de celles-ci.

Un opérande dont la nature déclarée est DONEX (donnée exogène) ou DONSC (donnée de scénario), est un opérande non calculé dont l'utilisateur doit fournir toutes les réalisations pour toutes les dates de simulation.

Exemple :

(\*SYSTEME URBAIN : INDICATEURS SOCIO-ECONOMIQUES\*)

(\*SCENARIO DE CROISSANCE\*)

POLAME : 2;

POLLOG : 2;

(\*VALEURS INITIALES\*)

PFU (I,T)

POUR 1975 : 0.248, 0.113, 0.157, 0.111, 0.371, 0;

(\*FONCTIONS TABULAIRES PREENREGISTREES\*)

FH3101 (POLAME,I):

(\*TOPPU(I)\*)

-0.006, +0.031, +0.009, +0.000, -0.034, \*

-0.010, +0.007, +0.029, +0.008, -0.034, \*

-0.010, -0.003, +0.009, +0.038, -0.034, \*

-0.018, -0.003, -0.003, +0.080, +0.016, \*

\*, \*, \*, \*, \*, \*

FH3102 (POLLOG,T)

(\*NLOGDEM(T)\*)

POUR 1975 : \*, \*, \*, \*, \*

POUR 1980 : 1.5, 1.6, 1.7, \*, \*

POUR 1985 : 1.5, 1.7, 2.0, \*, \*

POUR 1990 : 1.5, 1.7, 1.5, \*, \*

POUR 1995 : 1.4, 1.3, 1.2, \*, \*

POUR 2000 : 1.3, 0.9, 0.8, \*, \*;

(\*VARIABLES EXOGENES\*)

PMEN(K,T)

POUR 1975 : 0.35, 0.42, 0.25

POUR 1980 : 0.33, 0.41, 0.26

POUR 1985 : 0.32, 0.41, 0.27

POUR 1990 : 0.30, 0.42, 0.28

POUR 1995 : 0.29, 0.40, 0.31

POUR 2000 : 0.30, 0.41, 0.29;

MENURB(T)

POUR 1975 : 13.324

POUR 1980 : 14.178

POUR 1985 : 15.372

POUR 1990 : 16.250

POUR 1995 : 17.120

POUR 2000 : 18.015;

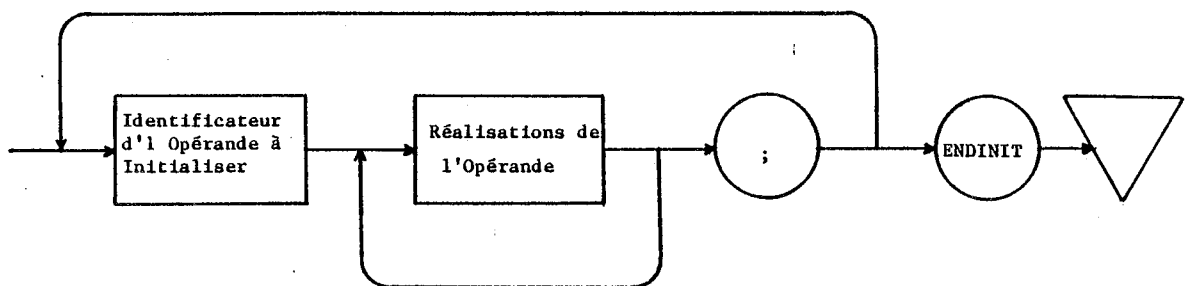
ENDINIT



### II.3.8.2. La syntaxe des entrées utilisateur

#### . La forme générale

La forme générale de la Description des initialisations des opérandes à initialiser d'un modèle, se définit en *langage S.A.M.S.S.* de la façon suivante :

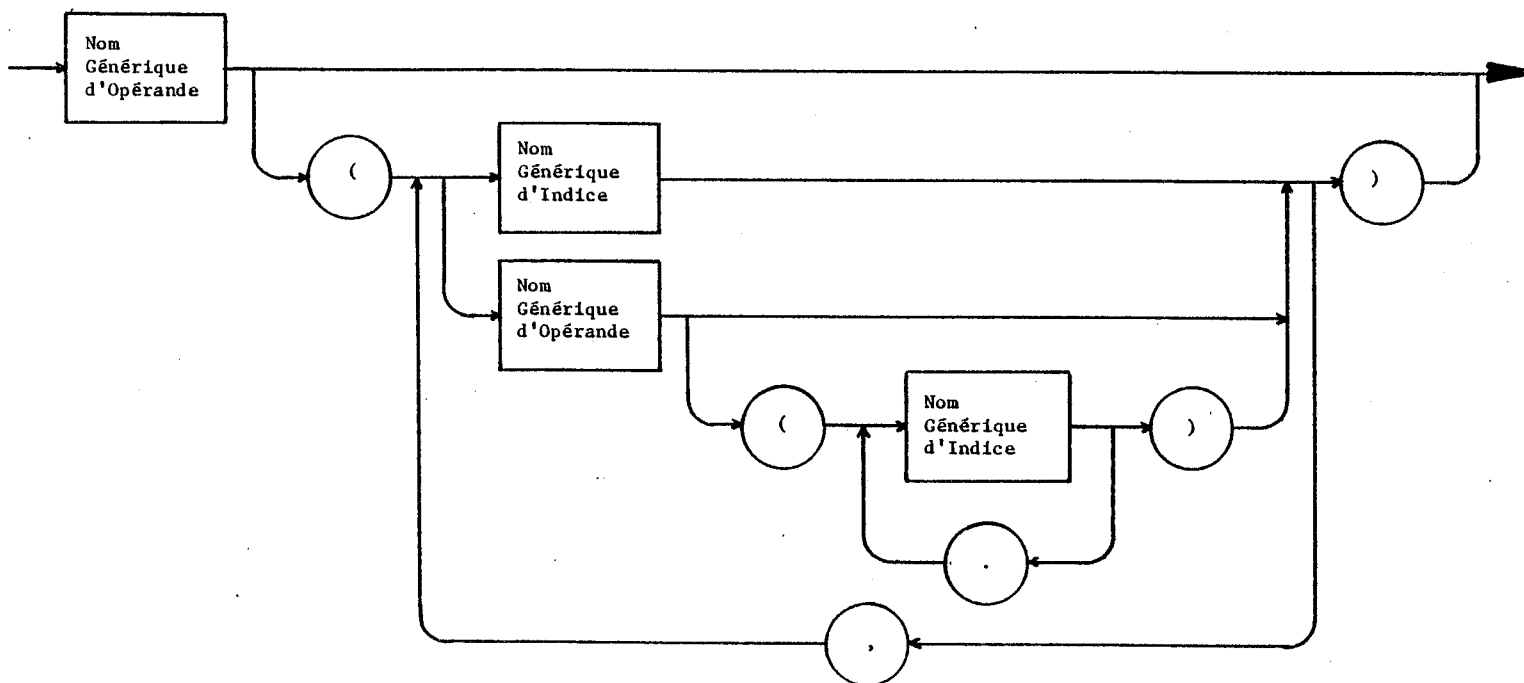


Cette formulation montre bien que toute description des initialisations des opérandes d'un modèle comporte obligatoirement :

- l'identificateur d'au moins un opérande,
- la (ou les) réalisation (s) correspondante (s),
- le symbole ";" qui annonce la fin de l'initialisation de l'opérande courant,
- le mot-clé ENDINIT qui précise que la phase d'initialisation est terminée et que l'on peut contrôler que tout opérande signalé comme devant être valué, l'est effectivement.

#### . L'identificateur de l'opérande à initialiser

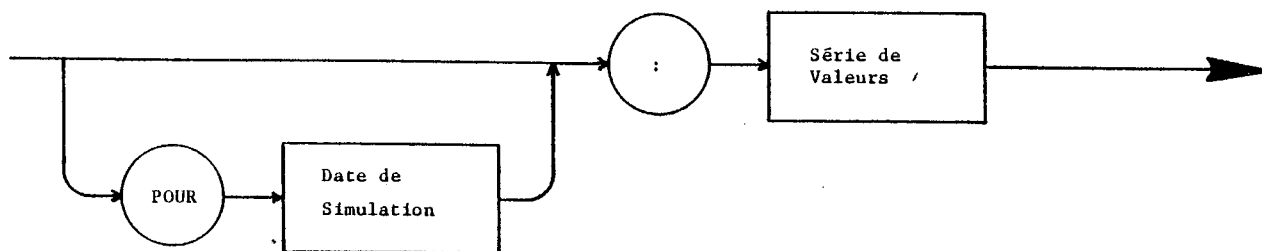
Comme pour le module de description des objets d'un modèle, cet identificateur d'opérande sera décrit comme le couple : *Nom Générique d'Opérande, Liste d'Indices qui le qualifie*, et que l'on représente ainsi :



. Les réalisations de l'opérande

Suivant la nature de l'opérande concerné et en tenant compte de sa dépendance ou de son indépendance par rapport à la date de simulation (présence ou absence de l'indice T dans la liste d'indices qui le qualifie), ses réalisations s'expriment sous des formes différentes.

Nous les représenterons globalement de la manière suivante :

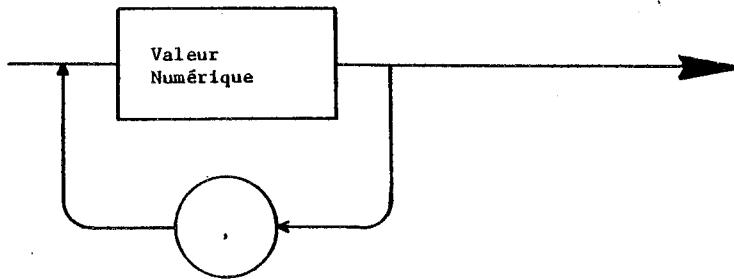


A partir de ce schéma, nous pouvons préciser :

- que, pour les réalisations d'un opérande qui dépend de la date de simulation, il est impératif de préciser la date concernée par la série de valeurs. Cette date, exprimée après le mot-clé POUR par un entier, doit figurer dans la liste descriptive que l'utilisateur a fourni pour l'indice T ;

- qu'à un opérande, fonction de la date de simulation et dont la *nature* déclarée est DONEX (donnée exogène) ou DONSC (donnée de scénario), doit correspondre une liste de réalisations où toutes les dates retenues pour la simulation apparaissent ;
- qu'à un opérande, fonction de la date de simulation, et dont la *nature* déclarée est DONIN (donnée initiale), doit correspondre une liste de réalisation où seule la date de début de simulation du modèle apparaît.

Une série de valeurs peut, quant à elle, être schématisée comme suit :



en précisant que :

- pour un opérande dont la description mentionne que ses éléments sont de type entier, les valeurs numériques de la série de valeurs doivent être entières,

- pour un opérande dont la description mentionne que ses éléments sont de type réel, les valeurs numériques fournies doivent être entières ou réelles,

- le symbole spécial \*, utilisé comme élément d'une série de valeurs, exprime le fait que l'utilisateur ne se préoccupe pas de la valeur de l'élément correspondant de l'opérande en cours d'initialisation.

. Remarques à propos des entrées-utilisateur

La version actuelle du module d'Initialisation des variables d'un modèle ne prévoit pas de possibilité d'interventions ponctuelles. Pour l'utilisateur, il serait parfois intéressant de pouvoir désigner

un sous-ensemble d'une variable et de n'initialiser que celui-ci. En effet, il n'est pas rare de constater que seuls quelques éléments d'une matrice sont utilisés par un modèle, pour un scénario donné (matrices creuses dont la gestion, tant au niveau de la Résolution Numérique qu'au niveau de son stockage, posent des problèmes complexes). De même, pour éviter de fastidieuses répétitions de valeurs, il serait intéressant de disposer d'un mécanisme du type facteur multiplicatif.

Au niveau des contrôles de cohérence, on notera que toute vérification des valeurs fournies par l'utilisateur, quant à leur appartenance aux plages de variations éventuellement définies au cours de la description des objets d'un modèle, est absente dans cette version du logiciel.

### II.3.8.3. Les autres entrées du module

Elles ne nécessitent pas d'intervention de la part de l'utilisateur. Pour pouvoir analyser correctement la Description des Initialisations, à savoir :

- contrôler l'existence des identificateurs d'opérandes dans le *Dictionnaire du Modèle*,
- vérifier que le nombre de réalisations, ainsi que le nombre et le type (entier ou réel) des valeurs fournies par l'utilisateur, correspondent bien à ceux que l'on a pu déduire des éléments stockés dans le *Dictionnaire du Modèle*,
- localiser l'espace correspondant dans la *Base des Opérandes du Modèle* pour y mémoriser les valeurs fournies,
- enfin, vérifier, en parcourant le *Dictionnaire du Modèle*, que toute variable définie comme une donnée (DONIN, DONEX, DONSC) est effectivement initialisée,

Le module d'initialisation des variables d'un modèle doit pouvoir accéder au *Dictionnaire du Modèle* et à la *Base des Opérandes du Modèle*.

#### II.3.8.4. Les informations fournies par le module

Il nous faut distinguer les informations que l'utilisateur pourra exploiter et donc visualisées, de celles qui, essentielles pour la poursuite d'une session S.A.M.S.S., sont transparentes à l'utilisateur.

##### . Les informations visualisées

Toute activation du module d'Initialisation des Variables d'un modèle donne lieu à l'édition d'un listing de contrôle. L'utilisateur y trouvera la *trace de la Description des Initialisations*. Cette trace précise, si c'est le cas, les erreurs détectées en cours d'analyse.

A la suite de cette trace, on édite le résultat des contrôles qui portent sur le fait qu'une variable définie comme donnée doit être initialisée. Ce résultat se présente sous la forme d'une liste des variables qui ne sont pas encore initialisées.

Enfin, les informations visualisées comportent les éléments statistiques qui concernent l'exécution du modèle :

- Nombre de variables initialisées au cours de la présente exécution du module.
- Nombre total de variables déjà initialisées.
- Nombre de variables qui doivent être initialisées.
- Espace physique minimum, nécessaire pour l'initialisation, ...

Dans la présentation de l'automatisation du modèle "didactique" du chapitre IV, on trouvera un exemple de ces diverses sorties du module d'Initialisation.

. Les informations transparentes

Il s'agit uniquement de la modification de l'état de la Base des Opérandes du modèle. Tout espace correspondant à une variable initialisée contient maintenant les valeurs validées fournies par l'utilisateur.

En résumé, le module d'initialisation des variables d'un modèle, analyse la *description des initialisations*, valide ou invalide cette description, mémorise les valeurs fournies dans la *Base des Opérandes* du Modèle et autorise ou pas la poursuite de la session S.A.M.S.S. en fonction du fait que l'on dispose ou non de la totalité des informations nécessaires à la résolution numérique.

### II.3.9. RESOLUTION NUMERIQUE D'UN MODELE

A ce stade de l'automatisation d'un modèle, on dispose :

- d'une description des objets du modèle : le Dictionnaire du Modèle,
- d'un espace physique où sont stockés tous les opérandes du modèle : la Base des Opérandes, initialisée au cours de l'exécution du module précédent,
- de la forme exécutable des équations du modèle.

La résolution numérique du modèle, c'est-à-dire l'évaluation de ses variables endogènes et éventuellement les contrôles de cohérence portant sur les réalisations des variables endogènes et/ou exogènes pour toutes les dates de simulation, peut donc être envisagée.

#### II.3.9.1. Les entrées du module RESNUM

Toutes les entrées de ce module sont transparentes à l'utilisateur :

- Le Dictionnaire du Modèle, élaboré au cours de la première étape de l'automatisation du modèle, mis à jour quant au lien entre description et représentation des variables, et dont la cohérence description/utilisation d'une variable a été assurée,
- La Base des Opérandes du Modèle, formatée lors de l'exécution de la deuxième étape de l'automatisation du modèle initialisée et dont la description existe dans le Dictionnaire du Modèle,
- Le Fichier Accès, généré à partir du texte en langage Pseudo-Pascal des équations du modèle, et qui permet d'accéder à l'ensemble des variables d'une relation avant exécution de celle-ci,

- La Forme Exécutable du Modèle, obtenue par compilation du texte macrogénéré, puis réordonné des équations du modèle.

L'utilisateur devra cependant spécifier, au niveau de la commande d'exécution du module, les dates de début et fin de la simulation (voir au chapitre IV la mise en oeuvre du logiciel S.A.M.S.S.).

### II.3.9.2. Les informations délivrées par le module

Au fur et à mesure de son exécution, le module de résolution numérique d'un modèle modifie le contenu de sa *Base des Opérandes*, et transmet un certain nombre d'indications sur le comportement du phénomène modélisé. Ces indications apparaissent sous forme de messages plus ou moins graves.

Si le contenu de la *Base des Opérandes* du modèle n'est visualisé que lors de l'interrogation de la base (cf. fonction suivante), l'émission de ces messages doit cependant permettre au concepteur du modèle de "caler" son modèle.

Les informations visualisées que produit le module de résolution numérique d'un modèle sont de trois ordres :

- les warnings : émis lors de l'exécution d'une relation de type Contrôle, ils indiquent que l'écart entre la valeur calculée d'une variable ou d'une expression, et la valeur attendue par le modélisateur n'est pas satisfaisant. Il convient alors de revoir :
  - . soit la formulation de la relation,
  - . soit les données,
  - . soit encore l'écart considéré comme acceptable,
 avant de tirer des enseignements valables de la simulation. La Résolution Numérique du Modèle n'est pas bloquée par ces warnings.
- les stops : s'ils émanent aussi des relations de type Contrôle, ils correspondent à un degré d'incohérence plus grave. Le modélisateur estime dans ce cas qu'il n'y a pas lieu de poursuivre la simulation tant que le contrôle n'est pas satisfait.



La Résolution Numérique du Modèle est arrêtée sur la relation en cause.

- un message : indiquant que le module de Résolution Numérique s'est achevé normalement, soit en fin de simulation, soit par arrêt dû à un contrôle non satisfait.

Si ce message n'apparaît pas, l'intervention de l'informaticien chargé de la gestion du logiciel S.A.M.S.S. est souhaitable. En effet, le motif de cette absence peut être une division dont le dénominateur est nul. Dans ce cas, un "code condition" au niveau du langage de contrôle du système hôte du logiciel S.A.M.S.S. le précisera, et il sera nécessaire de relancer l'exécution du module avec une demande de "vidage mémoire".

L'absence de ce message peut aussi être causée par une insuffisance du temps alloué pour la Résolution Numérique du Modèle. Là encore, un "code condition" permettra à l'informaticien de formuler un diagnostic et une parade.

On trouvera au chapitre IV une illustration de ces différentes sorties.

En résumé, l'exécution du module de Résolution Numérique d'un modèle, met à jour la *Base des Opérandes* et émet des informations qui permettent de "caler" (ou valider) le modèle.

### II.3.10. INTERROGATION DE LA BASE DES OPERANDES D'UN MODELE

L'Interrogation de la Base des Opérandes d'un modèle permet de présenter les résultats d'une simulation.

A partir de la Description des requêtes d'interrogation, on va éditer le contenu de la Base des Opérandes du modèle.

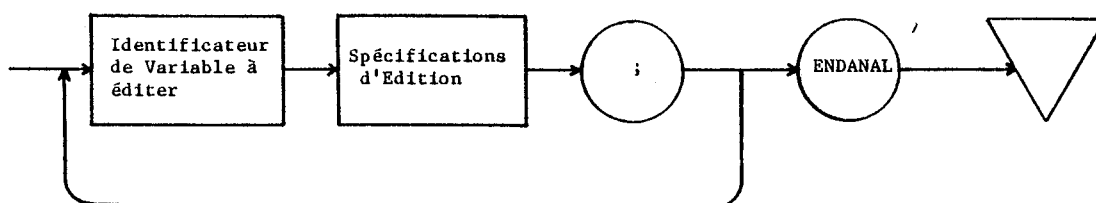
#### II.3.10.1. Les entrées du module INTBOP

Nous donnons en annexe, la carte syntaxique complète que l'utilisateur doit respecter avant d'activer ce module.

##### . La syntaxe du langage d'interrogation

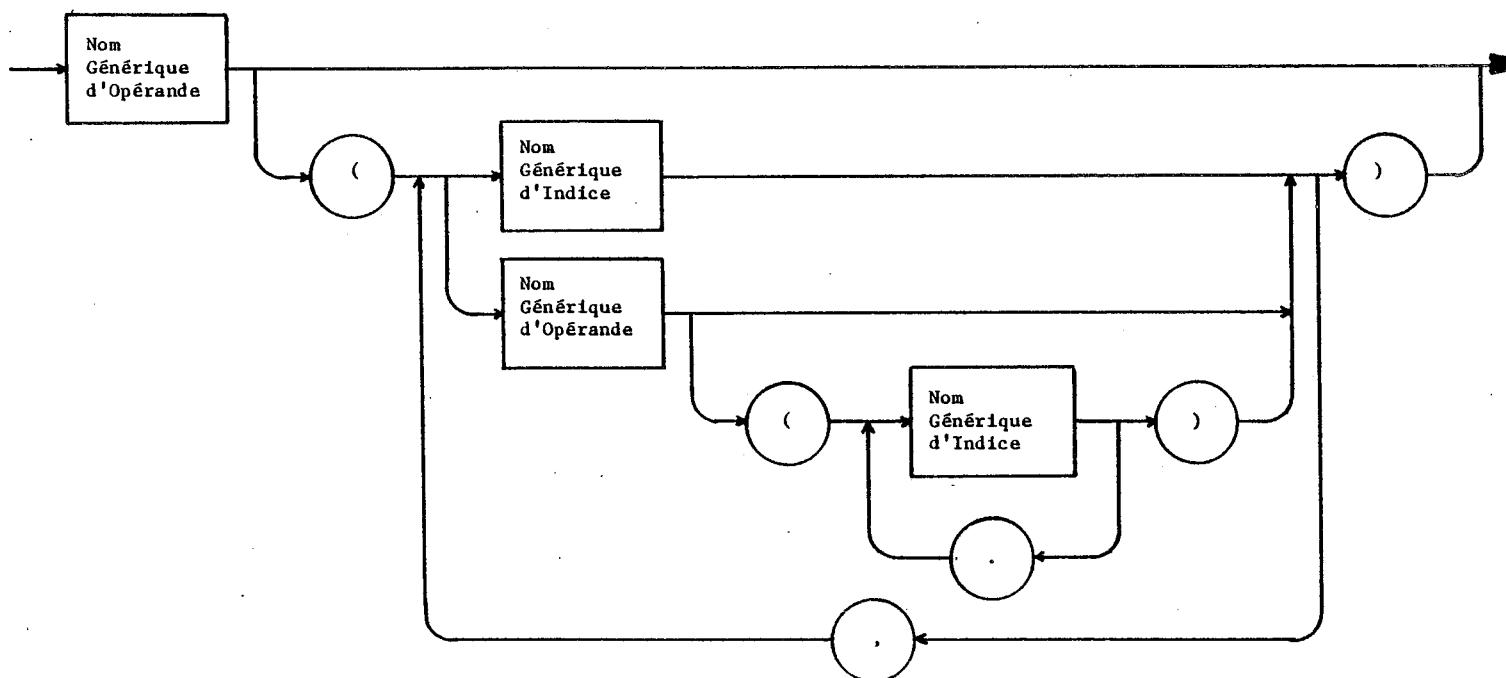
Pour être reconnue par le module d'Interrogation de la Base des Opérandes d'un modèle, la Description des requêtes d'interrogation doit respecter la syntaxe ci-dessous définie :

La forme générale : celle-ci peut, schématiquement, se définir de la façon suivante :

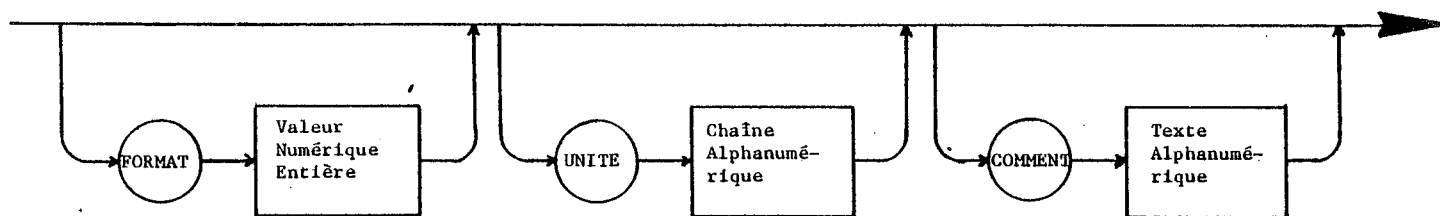


La Description des requêtes d'interrogation comporte donc obligatoirement au moins un identificateur de variable à éditer, suivi de la description des spécifications d'édition retenues. Le séparateur entre deux requêtes d'interrogation est le point virgule. Le mot-clé ENDANAL, obligatoire dans toute Description des requêtes d'interrogation, indique au logiciel que l'interrogation est terminée.

L'identificateur de variable à éditer : identifier une variable à éditer, c'est fournir le couple : Nom Générique d'Opérande, liste d'indices qui le qualifie. Nous reprenons ci-après le schéma descriptif de la syntaxe que doit respecter tout identificateur de variable.



La description des spécifications d'édition : pour présenter, de façon lisible, les résultats d'une simulation, nous mettons à la disposition de l'utilisateur trois options d'édition. Celles-ci peuvent être schématisées de la manière suivante :



Dans ce schéma, l'unité syntaxique complexe Valeur Numérique Entière qui suit le mot-clé FORMAT précise le nombre de positions décimales significatives que devra respecter le format d'édition des réalisations de la variable. Cette valeur numérique doit être comprise entre 1 et 8. Si l'option FORMAT est absente, la valeur de référence est 4.

L'unité syntaxique complexe Chaîne Alphanumérique qui suit le mot-clé UNITE identifie l'unité dans laquelle sont exprimées les réalisations de la variable. Le nombre maximum de caractères que cet identificateur peut comporter est 20 (aucun caractère blanc n'est admis).

L'unité syntaxique complexe Texte Alphanumérique qui suit le mot-clé COMMENT est un commentaire qui peut, par exemple, expliciter le libellé de la variable. Ce texte explicatif ne peut comporter plus de 74 caractères, blancs compris.

Exemple :

```

1 PIB(T);
2 EXPORT(T) UNITE MILLIONS_DINARS;
3 IMPORT(T) UNITE MILLIONS_DINARS;
4 CONSOM(T) UNITE MILLIONS_DINARS;
5 INVEST(T) UNITE MILLIONS_DINARS;
6 RESGAP(T) COMMENT POURCENTAGE DE COUVERTURE DES IMPORTATIONS;
7 QGAZPROD(T) UNITE MILLIONS_M3 COMMENT PRODUCTION DE GAZ;
8 QPETCONS(T) UNITE MILLIERS_DE_TONNES COMMENT CONSOMMATION INTERIEURE DE PETROLE;
9 QUANTGNL(T) UNITE MILLIERS_DE_TONNES COMMENT EXPORTATION DE GNL;
10 QUANT(TYPHC,T) UNITE MILLIERS_DE_TONNES COMMENT PRODUCTION D HYDROCARBURES;
11 ENDANAL

```

Les entrées du module qui ne nécessitent pas l'intervention de l'utilisateur

Pour analyser une requête d'interrogation, pour éditer le contenu de la variable interrogée, le module d'interrogation de la Base des Opérandes d'un modèle doit pouvoir :

- vérifier l'existence de la variable,
- localiser ses réalisations,
- ...

Pour cela, il est nécessaire qu'il dispose du Dictionnaire du Modèle ainsi que de la Base des Opérandes du Modèle.

### II.3.10.2. Les informations délivrées par le module

Toutes les informations délivrées par le module d'Interrogation de la Base des Opérandes d'un modèle sont visualisées. Elles sont de trois

ordres : des éléments statistiques, la trace de la Description des requêtes d'interrogation et enfin les résultats de la simulation.

. Les éléments statistiques

Ils concernent l'analyse et l'exécution des requêtes d'interrogation. Actuellement, ils sont plutôt destinés aux informaticiens puisqu'on y trouve :

- la taille de la zone dynamique disponible avant l'analyse des requêtes d'interrogation,
- la taille de la zone dynamique occupée par le Dictionnaire du Modèle,
- la taille de la zone dynamique disponible pour la mise en mémoire des variables à éditer,
- le taux de remplissage des tableaux utilisés par l'éditeur :

Actions d'édition

Lignes à éditer

...

. La trace de la Description des requêtes d'interrogation

Il s'agit de la trace de l'analyse des requêtes d'interrogation. Cette trace comporte, le cas échéant, les codes des messages d'erreurs qui ont pu être constatées. Un message indique par ailleurs le nombre de ces erreurs.

. Les résultats de la simulation

Pour chaque variable dont l'utilisateur souhaite analyser le contenu, le module d'Interrogation de la Base des Opérandes d'un modèle, édite un tableau dans lequel figurent :

- l'identificateur de la variable
- l'unité dans laquelle sont exprimées ses réalisations (à condition que l'utilisateur l'ait précisé dans la requête correspondante)
- les commentaires associés (même remarque que pour l'unité)
- la matrice correspondant aux réalisations de l'opérande.

Exemple :

QPETCONS(T)	UNITE: MILLIERS_DE_TONNES
CONSOMMATION INTERIEURE DE PETROLE	
T= 1980 :	8120.5
T= 1981 :	8770.2
T= 1982 :	9471.8
T= 1983 :	10230.0
T= 1984 :	11048.0

QUANTGNL(T)	UNITE: MILLIERS_DE_TONNES
EXPORTATION DE GNL	
T= 1980 :	-55432.0
T= 1981 :	-26415.0
T= 1982 :	10892.0
T= 1983 :	57505.0
T= 1984 :	114450.0

QUANT(TYPHC,T)	UNITE: MILLIERS_DE_TONNES						
	TYPHC = 1	TYPHC = 2	TYPHC = 3	TYPHC = 4	TYPHC = 5	TYPHC = 6	TYPHC = 7
T=1980	8120.5	5427.0	4466.9	48411.0	3235.0	N V	N V
T=1981	8770.2	5861.2	4824.2	47762.0	3235.0	N V	N V
T=1982	9471.8	6330.1	5210.2	47060.0	3235.0	N V	N V
T=1983	10230.0	6836.5	5627.0	46302.0	3235.0	N V	N V
T=1984	11048.0	7383.4	6077.1	45484.0	3235.0	N V	N V

### II.3.10.3. Remarques et restrictions à propos des variables interrogées

Il est possible que certaines variables du modèle, que l'utilisateur considère comme des variables résultats, soient inexploitable du fait de leur trop grande taille (matrices creuses ...). Dans ce cas, il est possible de définir des agrégations, en introduisant de nouvelles relations (dites d'agrégation des résultats), dans le modèle, au même titre que n'importe quelle autre relation.

Il faut alors rajouter le texte de ces relations à la Description des relations du modèle en ne négligeant pas leurs impacts sur la Description des Objets du modèle, puis réexécuter l'intégralité des étapes du logiciel S.A.M.S.S.

### CHAPITRE III : RÉALISATION PROGRAMMÉE

III.1. TECHNIQUES UTILISEES

III.2. REGLES DE CONSTITUTION D'UN MODULE S.A.M.S.S.

III.3. UTILISATION DE LA "CLASS" DANS S.A.M.S.S.

III.4. ENVIRONNEMENT S.A.M.S.S.

III.5. LA PHASE DE DEFINITION COHERENTE D'UN MODELE .

III.6. LA PHASE D'EXECUTION D'UN MODELE

III.7. CARACTERISTIQUES DES FICHIERS GERES PAR S.A.M.S.S.

III.8 DESCRIPTION ALGORITHMIQUE : L'exemple du module GENGRA

III.9. TAILLE DES MODULES DE S.A.M.S.S.



Le logiciel SAMSS a été découpé en 9 modules (dont un est généré automatiquement à partir des informations produites par ses prédécesseurs), programmés en PASCAL, l'ensemble comptant plus de 15.000 lignes d'instructions (hormis le module de Résolution Numérique, dont la taille est trop fortement liée au modèle. A titre d'exemple, pour le modèle MEDEE3, ce module comporte environ 12.000 lignes de PASCAL).

Pour gérer un tel ensemble, et développer de façon cohérente, de nouvelles versions (prototype → version 1 → version actuelle → version incluant les modifications ponctuelles à venir...) une méthodologie était indispensable.

### III.1. TECHNIQUES UTILISEES

Les techniques que nous avons choisies, relèvent pour certaines d'entre elles de la problématique du "génie logiciel".

- . Le langage PASCAL permet la déclaration de types. Grâce à cette notion, nous avons défini les différents objets manipulés par le logiciel, et nous avons assuré l'homogénéité des traitements sur les objets.
- . Par la programmation structurée, nous avons découpé le système de façon très fonctionnelle. De cette façon, les mêmes fonctions, les mêmes primitives peuvent être réutilisées dans des modules différents (notion de "commun").
- . La notion de "class" du langage PASCAL nous a permis de stocker de façon unique les descriptions de l'application. Celles-ci sont codées dans un langage intermédiaire qui sera exploité pour générer le module de Résolution Numérique d'un modèle. Ainsi les modules de définition du modèle (Création du Dictionnaire, Macro-génération des Relations, Génération du Graphe d'Evaluation des Relations) vont entre autre produire un langage intermédiaire, composé d'éléments alloués (Record) dans une zone d'allocation dynamique (récupérée dans l'espace laissé libre par le chargement d'un

module dans sa région d'exécution). L'ensemble des Records alloués sont structurés fonctionnellement sous la forme d'arborescences, dont le parcours s'effectue à l'aide de "pointeur" sur la "class". La "class" est échangée entre les modules, via un fichier séquentiel, enrichie progressivement par de nouvelles informations dans les différentes étapes de la définition du modèle. Elle est chargée en fonction des besoins, lors des étapes d'exécution.

### III.2. REGLES DE CONSTITUTION D'UN MODULE SAMSS

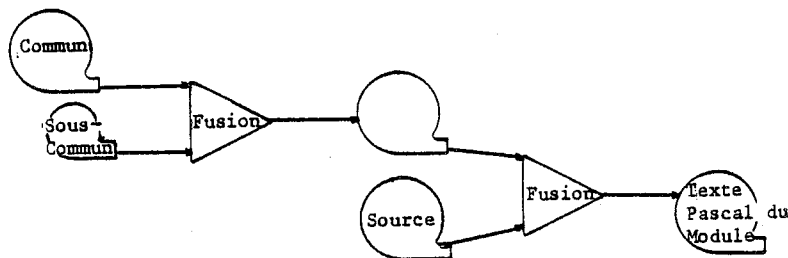
Sans entrer dans les détails de points sur la constitution des modules du logiciel SAMSS, nous rappelons que chaque module SAMSS est obtenu par la "fusion" de deux, voire trois textes sources PASCAL :

- . un "commun" : texte source PASCAL, commun à tous les modules du logiciel SAMSS, et qui comporte :
  - les constantes du logiciel,
  - la déclaration des éléments de la "class", leur analyse et leur gestion (sauvegarde, récupération, édition),
  - l'analyseur lexical du logiciel,
  - les procédures de gestion des erreurs,
  - les procédures de gestion des entrées/sorties sur les fichiers standards,
  - les début et fin d'exécution, communs à tous les modules,
  - les procédures d'analyse et de recherche d'un identificateur d'objet du modèle.
  
- . un "sous-commun" : éventuellement vide, c'est un texte source PASCAL commun à un sous-ensemble du logiciel SAMSS. On y trouvera par exemple la déclaration, l'analyse et la gestion des éléments du Graphe d'Evaluation des Relations, toutes choses utiles pour les modules de Génération de ce Graphe, de Contrôle de Cohérence, ou de

Génération du module exécutable correspondant au modèle, mais inutile par ailleurs.

- un "source" : texte source PASCAL, spécifique du module et de sa fonction dans le logiciel.

On trouvera ci-après le schéma de constitution d'un module SAMSS.



### III.2.1. Tableau de constitution des modules SAMSS

Nom du Module	Texte du Commun	Texte du Sous-Commun	Texte Source Constitutif
CREDIC	COMMUN	-	CREDIC
FMATBD	COMMUN	-	FMATBD
MACROG	COMMUN	-	MACROG
GENGRA	COMMUN	ANGR	GENGRA
CONTRO	COMMUN	ANGR	CONTRO
REORDO	COMMUN	ANGR	REORDO
INITBD	COMMUN	-	INITBD
RESNUM	COMMUN	SOFT	Texte Pascal correspondant au modèle généré par REORDO
INTBOP	COMMUN	-	INTBOP

#### Remarques :

- Pour homogénéiser le mode de constitution d'un module SAMSS, nous avons introduit un "sous-commun" vide lorsque le module n'exploitait pas de sous-commun spécifique.
- Le module RESNUM est en fait, produit automatiquement par le module REORDO.

### III.3. L'UTILISATION DE LA "CLASS" DANS SAMSS

On trouvera dans le PASCAL USER MANUAL AND REPORT [52] tout ce que recouvre le concept de "Class" en PASCAL. Notons cependant que :

- La "Class" définit un ensemble d'éléments, de Record, de même type dont le nombre peut varier pendant l'exécution du programme ;
- Chaque déclaration d'une variable appartenant à la "class" introduit un nombre ("set") variable d'éléments de ce type. Initialement, le "set" est vide ;
- Chaque activation de la procédure NEW (avec le bon paramètre : type de l'élément), génère (ou alloue) un nouvel élément dans la "class" et affecte la valeur par laquelle la nouvelle variable pourra être accédée. Cette valeur est appelée "Pointeur", et elle peut être assignée à des variables de type "pointeur" ;
- Chaque variable "pointeur" est liée par sa déclaration, à une "class" et à cause de ce lien, elle ne doit pointer que sur des éléments de cette "class". Il existe une valeur "NIL" qui ne pointe sur aucun élément quelque soit sa "class" ;
- L'utilisation de la structure de "class" autorise l'élaboration d'une information correspondant à n'importe quel type de graphe, les "pointeurs" représentant alors les liens et les variables composantes représentant alors les noeuds.

#### III.3.1. Les techniques de gestion de la "class"

Pour allouer un nouvel élément dans la "class", on utilisera l'instruction PASCAL : NEW. Pour libérer des éléments alloués (récupération de zones de travail,...) à partir d'un pointeur passé en paramètre, on utilisera l'instruction PASCAL : RESET.

Pour pallier l'inexistence de tableaux à bornes variables, nous aurons recours à une extension CRISS du langage PASCAL standard : NEW étendu. On trouvera ses spécifications dans le Manuel "Complément au PASCAL USER MANUAL" [ 52 ] .

### III.3.2. Liaison PASCAL-Système d'Exploitation

La gestion de la "class", par un programme PASCAL, implique la connaissance à tout moment de l'état de l'espace de travail dans lequel il s'exécute. pour cela, la présence des trois variables prédéclarées : \$DEBZONDY, \$FINZONDY et \$ACTZONDY est salutaire puisqu'elles permettent de connaître l'état de la zone dynamique.

\$DEBZONDY : Valeur entière de l'adresse (absolue) du début de la Zone Dynamique

\$FINZONDY : Valeur entière de l'adresse (absolue) de la fin de la Zone Dynamique

\$ACTZONDY : Valeur entière de l'adresse (absolue) du dernier Record allouée en Zone Dynamique

### III.3.3. Implémentation d'un Module en mémoire

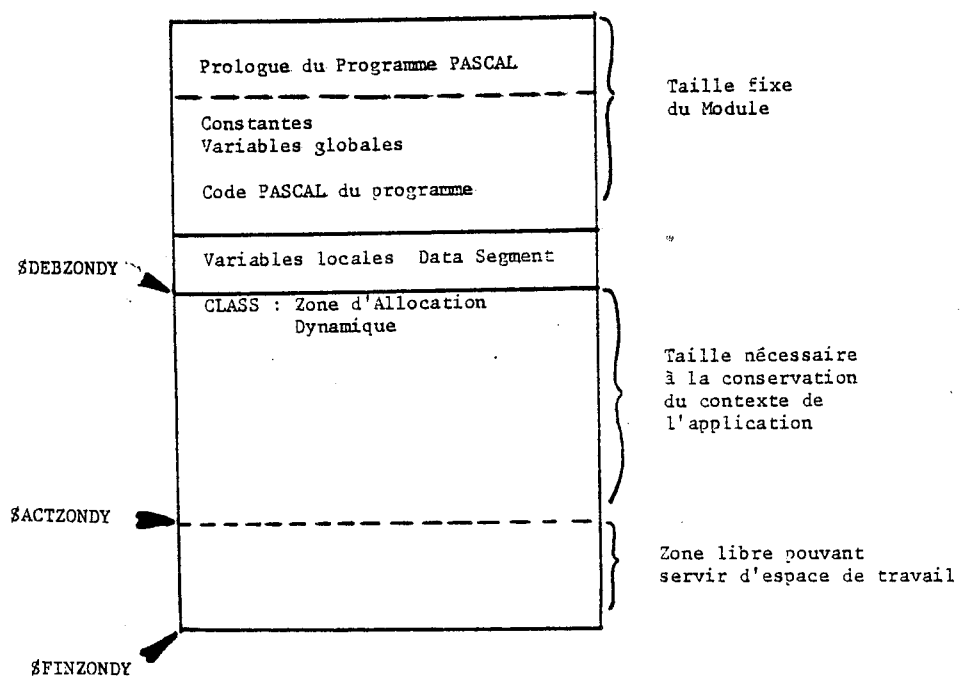


Schéma de la géographie d'un programme dans sa région d'exécution

### III.4. ENVIRONNEMENT SAMSS.

Le logiciel SAMSS évolue dans un environnement qui dépend de :

- la disposition d'un compilateur PASCAL qui comporte un certain nombre d'extensions par rapport au langage standard, en particulier dans ses liaisons avec le système d'exploitation. Parmi ces extensions on citera plus particulièrement :
  - les instructions MVC, CLC, NEW étendu, RESET, REWRITE,...
  - les variables prédéclarées \$COND, \$PARM,...
  - la taille des identificateurs doit être de 18 caractères minimum,
- l'"Operating System" de l'ordinateur hôte, puisque SAMSS, par l'intermédiaire de PASCAL, communique avec le système d'exploitation pour gérer la place mémoire, récupérer le contexte de mise en oeuvre, etc.
- l'utilisation : batch essentiellement,
  - cf. la partie Mise en Oeuvre du Manuel Didactique, de Référence et de Mise en Oeuvre du logiciel SAMSS. [ ]
- la disponibilité de fichiers séquentiels de type "move-mode" et "locate-mode" ainsi que de fichiers "accès-direct",
- la possibilité de récupérer la zone d'Allocation Dynamique,
- l'existence de fichiers imprimante de type PL/1 ou Fortran, avec concaténation possible.



Chacune des fonctions de SAMSS fait l'objet d'un module. On ne s'étonnera donc pas de retrouver ici les noms de module pour désigner une fonction du logiciel.

### III.5.1. Credic : Création du Dictionnaire du Modèle

Entrées : INPUT = fichier qui contient la Description en langage SAMSS (voir en Annexe sa syntaxe) des composants caractéristiques du Modèle.

Sorties : . SAVECLAS = fichier contenant le Dictionnaire du Modèle (Description codée des objets qui permettent de caractériser le phénomène modélisé).

. FIMPR = liste de contrôle des inputs utilisateur.

Fonctions : Analyse la Description des Objets du Modèle et génère la structure Dictionnaire correspondante. Cette structure sauvegardée dans le fichier SAVECLAS, est une arborescence à 3 branches : indices, opérandes, combinaisons d'indices.

Cette arborescence contient deux types d'informations :

- une description de la structure logique des objets manipulés (libellé, type, nature, plage de variation...)
- une description de la représentation physique de ces objets (position dans la Base des Opérandes, taille de l'objet; présence en mémoire centrale,...).

### III.5.2. Fmatbd : Formatage de la Base des Opérandes du Modèle

Entrées : SAVECLAS = fichier contenant le Dictionnaire du Modèle

Sorties : BASOPMOD = fichier qui contiendra la représentation physique des objets de type opérande du Modèle.

SAVECLAS = fichier Dictionnaire du Modèle complété par le lien "Opérande → adresse dans Basopmod".

FIMPR = statistiques d'exécution de la fonction.



Fonctions : Réserve et formate l'espace mémoire annexe destiné à recevoir les réalisations des opérandes du modèle.

Produit les liens entre cet espace annexe et le Dictionnaire du Modèle.

### III.5.3. Macrog : Macrogénération des Relations du Modèle

Entrées : . INPUT = fichier qui contient la Description en langage L.P.P. (voir sa syntaxe en Annexe) des Relations qui caractérisent le phénomène modélisé.

. SAVECLAS = fichier qui contient le Dictionnaire du Modèle.

Sorties : . FINGR = fichier qui contient les entrées nécessaires à la fonction de génération du Graphe d'Evaluation des Relations du Modèle.

. FPIV = fichier qui contient le texte PASCAL correspondant à chacune des Relations du Modèle (pavés-relation).

. SAVEREL2 = descripteur du fichier FPIV.

. FACCES = fichier qui contient les liens "relations → opérandes y intervenant", utile dans la phase d'exécution.

. FIMPR = liste de contrôle des inputs utilisateur, et statistiques d'exécution de la fonction.

Fonctions : . Analyse la Description des Relations du Modèle.

. Contrôle l'existence des opérandes et des indices utilisés.

. Transforme la formulation L.P.P. des Relations du Modèle pour obtenir un ensemble de "pavés-relations" (texte PASCAL dont la syntaxe n'est pas vérifiée ici) dont on conserve les caractéristiques d'implémentation (descripteur du fichier FPIV).

. Génère les liens "relations-opérandes y intervenant", et les entrées de la fonction suivante.

### III.5.4. Gengra : Génération du Graphe d'Evaluation des Relations du Modèle

Entrées : INPUT = fichier FINGR (produit par la fonction Macrog) qui contient le texte des relations réduites à leurs opérandes.  
SAVECLAS = fichier contenant le Dictionnaire du Modèle.

Sorties : . FGRAPHE = fichier qui contient le Graphe d'Evaluation des Relations du Modèle.

. FIMPR = liste de contrôle des inputs générés par la fonction Macrog, et statistiques d'exécution de la fonction.

Fonctions : Analyse le texte des relations réduites à leurs opérandes.

Gènère la structure Graphe d'Evaluation des Relations. Cette structure arborescente contient deux types d'informations :

- une description du phénomène modélisé vu sous l'angle de ses opérandes, appelée aussi matrice d'incidence du problème par rapport à ses opérandes
- une description du phénomène modélisé vu sous l'angle de ses relations, appelée aussi, matrice d'incidence du problème par rapport à ses relations.

Evalue ce Graphe, c'est-à-dire qu'en utilisant les principes suivants :

- toute relation de calcul peut être évaluée si chacun des éléments de sa partie droite est connu (soit Donnée, soit évalué au préalable),
- toute relation de contrôle peut être évaluée si tous ses éléments sont connus.

On affectera à chaque relation un "niveau" auquel elle peut être évaluée.

### III.5.5. Contro : Contrôle de la Cohérence "Description-Utilisation des Opérandes" du Modèle.

Entrées : . SAVECLAS = fichier contenant le Dictionnaire du Modèle.  
 . FGRAPHE = fichier contenant le Graphe d'Evaluation des Relations du Modèle.

Sorties : . FIMPR = liste des incohérences probables entre la Description et l'Utilisation des Opérandes du Modèle.

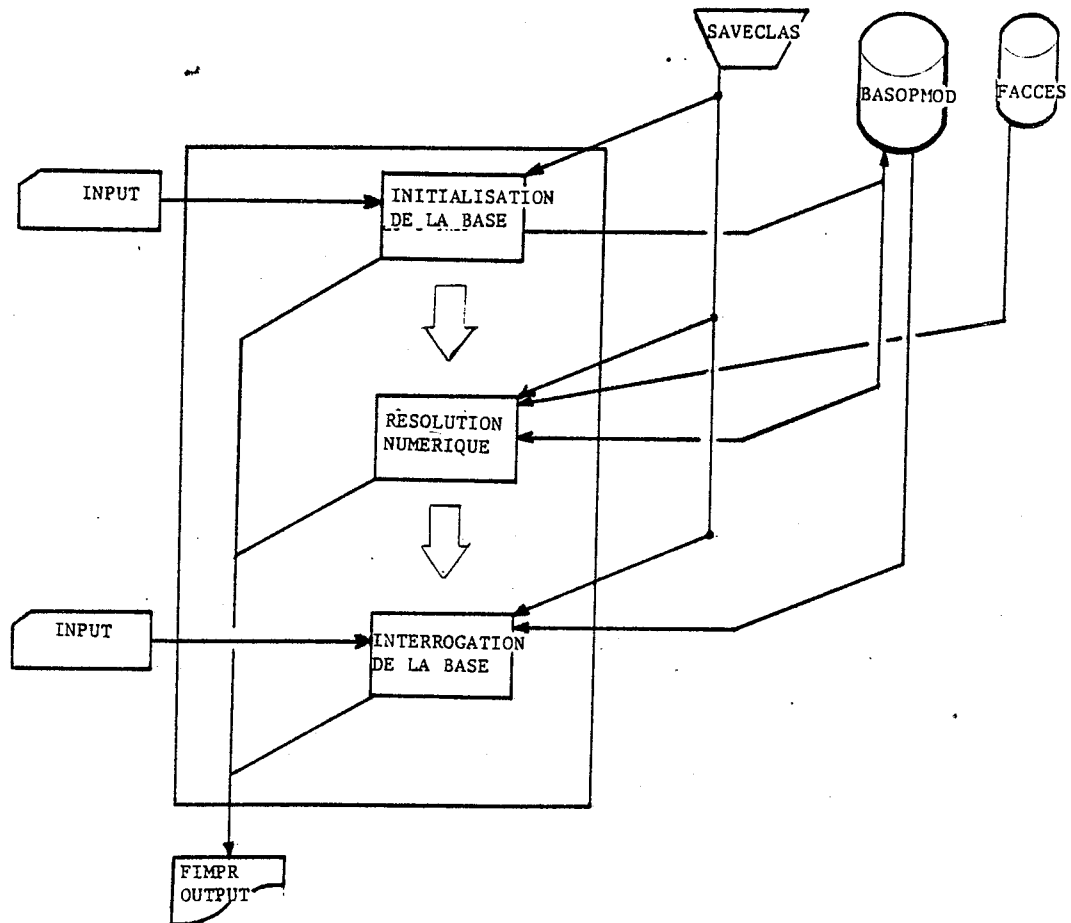
Fonctions : . Compare les informations fournies par l'utilisateur quant à la nature de ses opérandes et celles calculées par la fonction "Gengra".  
 . Emet un message d'erreur (plus ou moins grave) lorsque l'équivalence stricte n'est pas vérifiée, en précisant les numéros des relations qui utilisent l'opérande incriminé.

### III.5.6. Reordo : Génération du module exécutable correspondant au Modèle.

Entrées : . SAVECLAS = fichier qui contient le Dictionnaire du Modèle.  
 . FGRAPHE = fichier qui contient le Graphe d'Evaluation des Relations du Modèle.  
 . FPIV = fichier qui contient le texte PASCAL, non réordonné, de chacune des relations du modèle.  
 . SAVEREL2 = fichier qui contient la description du fichier PFIV.

Sorties : . Fonction RESNUM = load module correspondant au modèle de l'utilisateur.  
 . FIMPR = statistiques d'exécution de la fonction.  
 . TEMPLIST = liste de contrôle de la compilateur du programme PASCAL généré automatiquement et correspondant à la Résolution Numérique du Modèle.

- Fonctions : . Génère les déclarations (au sens PASCAL du terme) des Constantes, des Types et des Variables du Modèle, à partir du Dictionnaire du Modèle.
- . En respectant leur ordre d'évaluation (contenu dans la matrice d'incidence du problème par rapport à ses relations), réorganisation des "pavés-relations".
  - . Compilation du texte PASCAL issu de la fusion des textes ainsi produits, des procédures de gestion de la mémoire : SOFT, et du "commun".

III.6. LA PHASE D'EXECUTION D'UN MODELE

### III.6.1. Initbd : Initialisation de la base des Opérandes du Modèle.

Entrées : . INPUT = fichier qui contient la Description en langage SAMSS (voir en Annexe sa syntaxe) des Initialisations des Variables du Modèle.

. SAVECLAS = fichier qui contient le Dictionnaire du Modèle.

Sorties : . BASOPMOD = Base des Opérandes du Modèle, initialisée.

. FIMPR = Liste de contrôle des inputs de l'utilisateur.  
Liste des variables dont le Modèle attend encore l'initialisation.

Statistiques d'exécution de la fonction d'initialisation.

Fonctions : . Analyse la Description des Initialisations des variables du Modèle, en vérifiant que les réalisations d'un Opérande défini comme un entier (réel) sont bien des entiers (réels) que l'opérande est bien défini comme devant être initialisé, et que la série chronologique fournie pour une variable est complète.

. Mise à jour de la Base des Opérandes à partir des résultats de l'analyse et après avoir initialiser cette Base à  $-\infty$ .

. Vérifie que tout Opérande défini comme devant être initialisé, l'a effectivement été.

### III.6.2. Resnum : Résolution Numérique du Modèle.

Entrées : . SAVECLAS = fichier qui contient le Dictionnaire du Modèle.

. BASOPMOD = fichier où sont stockées physiquement les réalisations des variables initialisées ou non du modèle.

. FACCES = fichier de liaison entre une relation et ses opérandes.

Sorties : . BASOPMOD = Base des Opérandes du Modèle, mise à jour au cours de la Résolution Numérique.

- . FIMPR = fichier trace de la Résolution Numérique. On y trouve les "drapeaux" (messages d'erreur émis lorsque une relation de contrôle du Modèle n'est pas satisfaite).

Fonctions . Exécute l'ensemble des calculs et des contrôles définis par l'utilisateur dans son texte de Description des Relations du Modèle, et ce pour toute date de simulation appartenant à l'intervalle (Date de Début de Simulation, Date de fin de simulation).

- . Opère les mises à jour correspondantes de la Base des Opérandes du Modèle.

### III.6.3. Intbop : Interrogation de la Base des Opérandes du Modèle.

Entrées : . INPUT = fichier qui contient la Description en langage SAMSS (voir en Annexe sa syntaxe) des Requêtes d'Interrogation de la Base des Opérandes du Modèle.

- . SAVECLAS = fichier qui contient le Dictionnaire du Modèle.
- . BASOPMOD = fichier où sont stockées, physiquement les réalisations des Variables du Modèle.

Sorties : . FIMPR = liste de contrôle des Inputs de l'utilisateur.  
Statistiques d'exécution de la fonction d'édition.

- . OUTPUT = fichier de type imprimante où sont éditées, sous forme de tableaux, les réalisations des opérandes que l'utilisateur souhaite analyser.

Fonctions : . Analyse la Description des Requêtes d'Interrogation de la Base des Opérandes du Modèle

- . Localise ces Opérandes dans la Base des Opérandes.
- . Edite, en respectant les caractéristiques exprimées par l'utilisateur, les réalisations de ces opérandes.

### III.7. CARACTERISTIQUES DES FICHIERS GERES PAR SAMSS

Pour se gérer, le logiciel SAMSS fait appel à treize fichiers différents, que l'on peut regrouper en trois grandes classes :

- . Ceux qui caractérisent l'application automatisée
  - INPUT : Description de l'application
  - SAVECLAS = Dictionnaire de l'application
  - BASOPMOD : Base des Opérandes de l'application
  - FACCES : Liaisons Relations-Opérandes.
  
- . Ceux qui permettent de générer le programme Pascal de Résolution Numérique de l'application.
  - FPIV : fichier-pivot des relations
  - SAVEREL2 : descripteur du fichier-pivot
  - FGRAPHE : graphe d'évaluation des relations
  - FSREL1 : descripteur des niveaux d'évaluation du graphe
  - FPIVREOR : "source" Pascal du module de Résolution Numérique.
  
- . Ceux qui permettent de suivre la définition et la Résolution Numérique de l'application.
  - FIMPR : listes de contrôle des entrées d'un module et statistiques d'exécution
  - OUTPUT : Sorties des tableaux demandées, ou trace de la structure "Dictionnaire"
  - OUTPUTGR : trace de la structure "Graphe"
  - FTRACE : suivi d'exécution des modules.



III.7.1. INPUT : . Textes descriptifs d'une application SAMSS = fichiers résultant de la mise à jour des textes de références de l'application (Description des Objets du Modèle, Description des Relations du Modèle, Description des Initialisations des Variables du Modèle, Description des Requêtes d'Interrogation de la Base des Opérandes du Modèle) à l'aide des fichiers d'"Update" du scénario envisagé.

Format : image de carte - 80 colonnes - Array of char

Accès : Séquentiel move mode

Ouverture : Lecture

Procédure y accédant en lecture : REPLY (lecture d'un enregistrement)

Déclaration : dans le "Commun" - INPUT : FILE OF ENRSOURCE ;

Type : ENRSOURCE = ARRAY (1..LGENRSOURCE) OF CHAR ;

Constante : LGENRSOURCE = 80 ;

Variables liées : CHCNT : caractère courant de l'enregistrement  
 CNEXT : caractère suivant le caractère courant.

Remarque : Ces fichiers ne sont pas conservés.

III.7.2. SAVECLASS : Fichiers d'échange de la structure "Dictionnaire du Modèle"

Format : enregistrement de taille fixe = 88 caractères - Array of char

Accès : Séquentiel - locate mode

Ouverture : Lecture pour chargement dans la zone d'application dynamique

Ecriture pour sauvegarde de la structure

Procédures y accédant : en lecture ANALYSECL (adresse début, type de record, RECUPCLASS)

en écriture ANALYSECL (adresse début, type de record, SAUVECLASS)

Déclaration : Dans le "Commun" - SAVECLAS : TSAVECLAS ;

Type : TSAVECLAS = FILE LOCATE OF ENRCLASS ;  
ENRCLASS = ARRAY (1..LGSCCLASS) OF CHAR ;

Constante : LGSCCLASS = 88 ;

Variables liées : PTRDIC

Remarque : Ce fichier sera conservé.

III.7.3. BASOPMOD : Fichier BASE des OPérandes du MODèle = espace mémoire structuré en "segments" et "pages" où sont stockées les réalisations des opérandes du Modèle.

Format : enregistrement de taille fixe correspondant à la taille d'une "page". Array of char

Accès : direct, via le numéro de page, sauf en FMATBD sur IBM où il est séquentiel "locate putmod"

Ouverture : Lecture/Ecriture

Procédures y accédant : en lecture CHARGER (Numop)  
 en écriture PFORMATBASE  
 LIBERERMEM

Déclaration : dans les sources des modules qui l'utilise :

FMATBD  
 INITBD  
 RESNUM  
 INTBOP

BASOPMOD : TBASEOPMOD ; (\* uniquement dans le module  
 FMATBD sous IBM\*)

BASOPMOD : FILE DIRECT OF PAGE : (\*dans tous les autres  
 cas\*)

Type : TBASEOPMOD = FILE LOCATE PUTMOD OF PAGE ; en FMATBD  
 sur IBM

PAGE = ARRAY (0..TAILLEPAGE) OF CHAR ;

Constante : TAILLEPAGE = 7049 ; sur IBM 360/67 avec disque 2314  
 = 13030 sur IBM 370/168 avec disque  
 3330  
 = 4092 sur ICL 2980 sous VME/B

Variables liées : NBPAGSEGP nombre de pages du segment qui contient les variables indépendantes de la date de simulation.

NBPAGSEGT nombre de pages d'un segment qui contient les réalisations de variables pour une date de simulation donnée.

Remarque : Ce fichier est conservé.

III.7.4. FACCES : Fichier de liaison entre une relation et ses opérands, il permet, au moment de la Résolution Numérique du Modèle, de charger les réalisations des opérands utilisés par une relation avant l'évaluation de celle-ci.

Format : enregistrement de taille fixe - Array of char

Accès : direct via le numéro de la relation correspondante

Ouverture : Lecture/Ecriture

Procédure y accédant : en lecture ACCES (type de relation, numéro de relation) dans le module RESNUM  
en écriture GACCES dans le module MACROG

Déclaration : dans les sources des modules qui l'utilise :  
en MACROG : FACCES : FILE DIRECT (NBMAXREL) OF TACCES ;  
en RESNUM : FACCES : FILE DIRECT OF TCHAINAC ;

Type : TACCES = ARRAY (1 .. LGACCES) OF CHAR ;  
TCHAINAC = ARRAY (1 .. TAILLECHAC) OF CHAR ;

Constantes : LGACCES = 270 ;  
TAILLECHAC = 270 ;                   = NBMAXOPR \* LGLIBOP  
NBMAXREL = 1000 ;                   = nombre maximum de relations  
  d'un modèle

Variables liées : NUMREL                   = numéro d'ordre de la relation  
  dans la description fournie par  
  l'utilisateur

Remarque : Ce fichier est conservé.

III.7.5. FPIV : "Fichier-PIVot" du Modèle = texte Pascal généré à partir de la Description des Relations, mais pas encore réordonné.

Format : enregistrement de taille fixe correspondant à la taille d'une "page" Array of char

Accès : direct via un numéro de page

Ouverture : Lecture/Ecriture

Procédures y accédant : en lecture TRANSFERT dans le module REORDO  
en écriture MCVFPIV dans le module MACROG

Déclaration : dans les sources des modules qui l'utilise  
FPIV : FILE DIRECT OF PAGEFPIV ;

Type : PAGEFPIV = ARRAY (1 .. TPAGFPIV) OF CHAR ;

Constantes : TPAGFPIV = 7049 ; sur IBM 360/67 avec unités de  
disque 2314  
TPAGFPIV = 13030 ; sur IBM 370/168 avec unités de  
disque 3330  
TPAGFPIV = 4092 ; sur ICL 2980 sous VME/B

Remarque : Dans la mesure où les modifications ponctuelles sur le texte des Relations ne sont pas encore incorporées au logiciel, ce fichier intermédiaire peut ne pas être conservé au delà de la génération du module exécutable correspondant au modèle.

III.7.6. SAVEREL2 : Descripteur du "fichier-pivot" d'un modèle. Généré en même temps que le "fichier-pivot", il indique pour chaque relation, la taille de texte Pascal macrogénéré et son implémentation dans FPIV. Il permettra d'extraire les "pavés-relations" pour les réordonner et produire ainsi le texte source Pascal compilable et exécutable correspondant au modèle.

Format : enregistrement de taille fixe

Accès : Séquentiel locate mode

Ouverture : en écriture dans le module MACROG  
en lecture dans le module REORDO

Déclaration : dans les sources des modules qui l'exploite  
SAVEREL2 : FILE LOCATE OF TYPREL2 ;

Type : TYPREL2 = RECORD  
ADRPR : TYPADRPR ;  
TAILLEPR : INTEGER ;  
END ;

TYPADRPR = RECORD  
NOPAG : SHRTINT ;  
DEPPAG : INTEGER ;  
END ;

Constante : NBMAXREL = 1000 ;

Variables liées : REL2REOR tableau descripteur de l'implémentation  
des NBMAXREL "pavés-relations"

Remarque : *idem que pour FPIV.*

III.7.7. FGRAPHE : Fichier d'échange de la structure "Graphe d'Evaluation des Relations d'un Modèle"

Format : enregistrement de taille fixe = 80 caractères - Array of char

Accès : Séquentiel locate mode

Ouverture : en lecture pour chargement dans la zone d'allocation dynamique  
en écriture pour sauvegarde de la structure

Procédure y accédant : en lecture ANALGR (adresse de début, type de record, RECUPGR)  
en écriture ANALGR (adresse de début, type de record, SAVEGR)

Déclaration : dans le sous-commun "Analyse du Graphe" : ANGR  
FGRAPHE : TFGRAPHE ;

Type : TFGRAPHE = FILE LOCATE OF ENRSAVEGR ;  
ENFSAVEGR = ARRAY (1 .. LGSGR) OF CHAR ;

Constantes : LGSGR = 80 ;

Variables liées : PTGRAF

Remarque : Cette structure n'a d'intérêt à être conservée que tant que le load module de Résolution Numérique du Modèle n'est pas disponible. On peut donc la détruire après cette opération.

III.7.8. FSREL1 : Il permet de mémoriser la liste des relations évaluables pour chacun des niveaux du Graphe.

Format : enregistrement de taille fixe = 80 caractères - Array of char

Accès : Séquentiel locate mode

Ouverture : Lecture pour chargement en zone d'allocation dynamique  
Ecriture pour sauvegarde de la structure

Procédures y accédant : en lecture ANALREL1 (adresse de début, type de record, RECUPREL1)  
en écriture ANALREL1 (adresse de début, type de record, SAVEREL1)

Déclaration : dans le source du module qui l'utilise : REORDO  
SAVEREL1 : TFSREL1 ;

Type : TFSREL1 = FILE LOCATE OF ENRSAVEREL1 ;  
ENRSAVEREL1 = ARRAY (1 .. LGSREL1) OF CHAR ;

Constante : LGSREL1 = 80 ;

Variables liées : PTREL1

Remarque : *Aucun échange de cette structure n'étant nécessaire dans l'état actuel de développement du logiciel, ce fichier n'est pas utilisé.*



III.7.9. FPIVREOR : Source Pascal du Module de Résolution Numérique d'un Modèle = Fichier-Pivot réordonné qui fusionne avec le sous-commun "SOFT" et le "COMMUN" constituera l'input du compilateur Pascal afin de produire le module exécutable du Modèle.

Format : enregistrement de taille fixe - image de carte -  
Array of char

Accès : Séquentiel move mode

Ouverture : Ecriture

Procédures y accédant : PBUFOUT pour la génération des déclarations  
GENMOD  
GENPROC (NIVPROC, SNIV, type de record)  
TRANSFERT (TAILLETR)  
MAIN

Déclaration : dans le module qui l'utilise : REORDO  
FPIVREOR : TFPIVREOR ;

Type : TFPIVREOR = FILE OF ENRPIVREOR ;  
ENRPIVREOR = ARRAY (1 .. LGENRPREOR) OF CHAR ;

Constante : LGENRPREOR = 80 ;

Variables liées : ENRPROC, ENRBEGIN, ENREND, ENRAPPEL,  
ENRPROC MOD ...

Remarque : Ce fichier peut être détruit une fois que la génération du load module de Résolution Numérique a pu se faire.

III.7.10. FIMPR : Liste de contrôle des inputs de l'utilisateur

- . erreurs
- . statistiques d'exécutions des modules

Format : fichier de type listing : DESCR132 : 132 colonnes

1ère colonne : caractère de contrôle pour

l'imprimante : ¶ = saut 1 ligne

1 = saut page

Accès : Séquentiel move mode

Ouverture : Ecriture

Procédures y accédant en écriture : ECRCARTE

ERREUR

PENTETESTEP

RECUPPARM

RECUPSTAT

Déclaration : COMMON FIMPR : file of enrimp ;

Type : ENRIMP : ARRAY (1 .. LGIMP) OF CHAR ;

Constante : LGIMP = 132 ;

Variables liées : LIGLUES : n° ENREG SOURCE DEJA LU

LIGNER : n° ENREG SOURCE DEJA LISTE

Remarque : *uncatalg*

III.7.11. FTRACE : Fichier d'aide au debugging  
 = trace des appels de procédures ...  
 + trace lourde d'exécution des modules sans édition  
 du dictionnaire et du graphe

Format : fichier de type listing : DESCR132 : 132 colonnes  
 1ère colonne : caractère de contrôle pour l'imprimante  
 ¶ = saut 1 ligne  
 l = saut page

Accès : Séquentiel move mode

Ouverture : Ecriture

Procédure y accédant en écriture : toutes les procédures du  
 logiciel SAMSS par write

Déclaration : COMMUN FTRACE : file of enrtrace ;

Type : ENRTRACE = ARRAY (1 .. LGOUT) of char ;

Constante : lgout : 132 ;

Variables liées : tcsor : booléan ;

Remarque : *uncatalg*

III.7.12. OUTPUT : Fichier de sortie des résultats de la Résolution Numérique du Modèle d'édition du Dictionnaire sous forme codée.

Format : fichier de type listing : DESCR 132 - 132 colonnes  
 1ère colonne : caractère de contrôle pour l'imprimante :  
 b = saut 1 ligne  
 l = saut page

Accès : Séquentiel move mode

Ouverture : Ecriture

Procédures y accédant en écriture : PENTETESTEP par write  
 RECUPARM par write  
 RECUPSTAT par write  
 DISPLAY par write

Déclaration : COMMUN OUTPUT : file of entrace ;

Type : ENTRACE : ARRAY (1 .. LGOUT) of char ;

Constante : LGOUT = 132 ;

Variables liées :

Remarque : *uncatalg*

### III.8. DESCRIPTION ALGORITHMIQUE : l'exemple du module GENGRA

Nous avons profité au maximum des qualités du langage PASCAL et de la décomposition organique du logiciel SAMSS. C'est ainsi que la documentation de base figure dans les sources des programmes. Nous y voyons plusieurs avantages :

- . Le langage PASCAL est structuré et "parlant" (cf. les notions de type et de procédure). Il est donc inutile de réécrire ce que le langage peut exprimer.
- . La documentation évolue de fait avec la vie du programme, ce qui constitue une excellente protection contre l'obsolescence pour peu que les mises à jour respectent les règles de documentation.
- . Le travail de documentation est indépendant du site sur lequel le logiciel est installé car la documentation se fait uniquement par insertion de commentaires.

L'utilisation d'un métalangage pour décrire organiquement les modules du logiciel SAMSS, vise à faciliter la compréhension et la lisibilité des algorithmes, en les libérant des détails. Nous avons repris, pour ce métalangage, les conventions utilisées dans le langage PASCAL, ce qui facilite la correspondance entre le dossier technique et les sources des programmes.

	Description algorithmique	Variables Principales affectées	Appel des procédures correspondant
<u>MAIN</u>	<p>début</p> <p>Édition de l'"entête" des fichiers standards</p> <p>Récupération des paramètres d'exécution</p> <p>Récupération de la structure "Dictionnaire"</p> <p>Pour toute relation faire</p> <p>début</p> <p>Analyse du texte de la relation réduite à ses seuls opérandes</p> <p>Évaluation du vecteur de connaissance de la relation</p> <p>Insertion de la relation dans la structure "Graphe" d'évaluation des relations</p> <p>fin</p> <p>Évaluation du "Graphe"</p> <p>Sauvegarde de la structure "Graphe"</p> <p>fin</p>	<p>Zone d'allocation dynamique</p> <p>FGRAPHE</p>	<p>PENTETESTEP</p> <p>RECUPPARM</p> <p>ANALYSECL (PTDEBCLASS, CLASSE RECUPCLASS)</p> <p>ANALREL</p> <p>EVALYCONREL</p> <p>INTRELGR</p> <p>EVALGRAPHE</p> <p>ANALGR (DEBGRAPHE, GRAPH, SAVE)</p>
<u>ANALREL</u>	<p>Analyse le texte de la relation réduite à ses seuls opérandes</p> <p>début</p> <p>Analyse et Mémoire le type (Calcul/Contrôle) de la relation</p> <p>Analyse et Mémoire la nature (Statique/Dynamique) de la relation</p> <p>Analyse et Mémoire la liste des opérandes utilisés par la relation</p> <p>fin</p>	<p>RELINPGR.TYPRELIN</p> <p>RELINPGR.NATRELIN</p> <p>RELINPGR.LSTOPIN</p>	<p>ANALTYPREL</p> <p>ANALNATREL</p> <p>ANALLSTOP</p>
<u>EVALYCONREL</u>	<p>Évaluation du Vecteur de connaissance de la relation</p> <p>début</p> <p>Initialisation des compteurs des différents types d'opérandes susceptibles d'être rencontrés</p> <p>Pour tout opérande de la relation courante faire</p> <p>début</p> <p>Recherche de l'opérande dans le "graphe"</p> <p>Incrémenter le compteur correspondant au type de l'opérande (DONNEE, VARIABLE, INCONNUE)</p> <p>fin</p> <p>fin</p>	<p>NBDONREL, NBINCREL, NBVARREL</p> <p>NBDONREL, NBINCREL, NBVARREL</p>	<p>RECHDANSGR (NAMEIN, ADRVARGR, ADRAPPEND)</p>
<u>INTRELGR</u>	<p>Introduction de la Relation courante dans le graphe d'Évaluation des Relations du Modèle</p> <p>début</p> <p>Vérifier la "calculabilité" d'une relation</p> <p>Ajouter le descripteur de la relation dans le graphe d'évaluation des Relations du Modèle</p> <p>Ajouter les opérandes de la relation non déjà connus, dans le Graphe d'évaluation des Relations du Modèle</p> <p>Mises à jour du Graphe, induites par l'adjonction de la relation courante :</p> <ul style="list-style-type: none"> <li>. Création du lien Relation courante-opérandes utilisé par celle-ci.</li> <li>. Adjonction de la Relation courante dans les différentes listes de relations qui utilisent un des opérandes intervenant dans la Relation courante</li> <li>. Si Relation Courante = Relation de calcul alors</li> </ul> <p>début</p> <p>Mise à jour du descripteur du premier opérande de la Relation Courante</p> <p>Évaluation de l'ensemble des relations qui doivent être évaluées avant la Relation Courante pour que celle-ci soit calculable</p> <p>Mise à jour de la liste des relations qui utilisent l'opérande calculé par la Relation Courante (à évaluer après la Relation Courante)</p> <p>fin</p> <p>Si non Évaluation de l'ensemble des relations qui doivent être calculées avant la Relation Courante</p> <p>fin</p>	<p>FGRAPHE</p>	<p>VERIFCALCREL</p> <p>INSERDESCRC</p> <p>INSERVARINC</p> <p>INSLSTOPRC</p> <p>INSADRR</p> <p>MAJDESCROP1</p> <p>GENLSTRELAV</p> <p>MAJLSTRELAP</p> <p>GENLSTRELAV</p>

	Description algorithmique	Variables Principales affectées	Appel des procédures correspondant.
EVALGRAPHE	<p>Recherche du niveau d'évaluation de chacune des relations introduites dans le Graphe</p> <pre> <u>début</u>   <u>Pour</u> tout niveau d'évaluation <u>faire</u>     <u>début</u>       <u>Pour</u> toute relation dont le niveau d'évaluation n'est pas encore connu <u>faire</u>         <u>début</u>           <u>Si</u> on connaît le niveau d'évaluation de toutes les relations qui doivent être évaluées avant la Relation courante <u>alors</u>             <u>début</u>               Niveau d'évaluation de la Relation courante = Niveau courant               Mise à jour de la liste des relations à évaluer après la relation courante             <u>fin</u>           <u>fin</u>         <u>fin</u>       <u>fin</u>     <u>fin</u>   <u>fin</u> </pre>	FGRAPHE	<p>DETERMINNIV(NIVCOURANT)</p> <p>MAJNIVRELAR(NIVCOURANT)</p>

III.9. LA TAILLE DES MODULES DU LOGICIEL : exprimée en K octets

Nom du Modulé	Taille du Module Exécutable	TLDS Data Segment Length (entête)	TLFB Buffer System Length	TOTAL
CREDIC	60	4	16	80
FMATBD	45	4	16	65
MACROG	120	4	30	154
GENGRA	75	4	30	109
CONTRO	65	4	16	85
REORDO	110	4	30	144
INITBD	65	4	30	95
RESNUM	60	4	16	80
INTBOP	60	4	16	80

\* Pour RESNUM, il faut rajouter la taille des "pavés-modèles" qui est fonction du nombre des relations et de leur complexité.

Les tailles que nous indiquons dans la colonne "total" sont les tailles minimales requises pour charger les modules correspondants. Pour les faire exécuter, il faut prévoir en plus :

- . pour tous les modules, l'espace de travail nécessaire à l'élaboration ou au chargement du *Dictionnaire du Modèle* ;
- . spécialement pour MACROG, l'espace de travail nécessaire à la macro-génération de la relation la plus longue ;
- . spécialement pour CONTRO, GENGRA et REORDO, l'espace de travail nécessaire à l'élaboration ou au chargement du *Graphe d'Evaluation des Relations du modèle* ;



- . spécialement pour INITBD et INTBOP, l'espace de travail nécessaire pour assurer la présence en mémoire centrale de l'opérande du modèle dont la taille est la plus grande ;
  
- . spécialement pour RESNUM, l'espace de travail nécessaire pour assurer la présence simultanée en mémoire centrale de tous les opérandes d'une relation, quelle qu'elle soit ;
  
- . spécialement pour REORDO et MACROG, l'espace de travail nécessaire au chargement du descripteur du "*fichier-pivot*".

## CHAPITRE IV : UN EXEMPLE D'UTILISATION

IV.1. PRESENTATION GENERALE DU MODELE

IV.2. CONSTRUCTION DU MODELE

IV.3. INTERPRETATION DES RESULTATS - INTERETS DE LA SIMULATION

IV.4. MISE EN OEUVRE TECHNIQUE DU LOGICIEL

#### IV.1. PRÉSENTATION GÉNÉRALE DU MODÈLE

C'est à travers un exemple simple qu'il semble le plus parlant de mettre en évidence les qualités et les défauts d'un logiciel, ce qu'il peut apporter à l'utilisateur (sur le plan de l'analyse), et ses possibilités de mise en oeuvre. L'exemple qui va servir de base dans ce chapitre, a été développé à l'occasion des premières "Journées d'Etudes du CRISS" (25 et 26 mai 1982), consacrées à la confrontation d'expériences, tant "utilisateurs" que "concepteurs", d'outils informatiques d'aide à la modélisation en Sciences Sociales. Evidemment, on pourra nous reprocher d'avoir opté pour un cas d'école, mais celui-ci n'est pas pour autant exempt de tout intérêt. Il présente en effet l'avantage d'être assimilable, du point de vue conceptuel, par des "non-économistes".

Le problème que nous cherchions à résoudre peut se définir comme suit : partant de l'équilibre "Ressources-Emplois" de l'année 1979, quelles quantités de Gaz Naturel Liquéfié (G.N.L.), l'Algérie doit-elle exporter chaque année au cours du plan 1980-1984 pour assurer le maintien de cet équilibre, tout en réalisant les objectifs planifiés d'investissement et de consommation ?

Le modèle de base, capable de fournir une réponse à cette question, sous diverses hypothèses de valorisation du G.N.L. sur le marché mondial, est l'équation d'équilibre entre les ressources et les emplois de la comptabilité nationale algérienne :

$$P.I.B.(T) + IMPORTATIONS(T) - EXPORTATIONS(T) = CONSOMMATION(T) + INVESTISSEMENTS(T)$$

Une désagrégation sommaire de cette équation, d'une part selon les diverses branches de production, d'autre part en distinguant les quantités en termes physiques (tonnes, m<sup>3</sup>, ... etc) et les prix unitaires, a permis de définir les composants caractéristiques du problème. L'évolution de chacun de ces composants a été décrite soit comme des données exogènes (cas des investissements planifiés), soit comme un résultat, calculé par des relations prenant en compte les mouvements distincts (en termes physiques) des prix exprimés en monnaie constante et des taux d'inflation relatifs aux diverses branches d'activité.

## IV.2. CONSTRUCTION DU MODÈLE

### IV.2.1. LE PROBLEME VU PAR L'ECONOMISTE

Il s'agit de mettre en évidence le G.N.L. à l'intérieur de l'équation de base. Après avoir compulsé les documents en notre possession [42], [43] et [44], nous avons pu caractériser la P.I.B. annuelle par ses 11 branches de production (l'Agriculture, les Hydrocarbures, les Mines et Carrières, l'Energie et l'Eau, les Industries Manufacturières, le BTP, les Transports, les Commerces, les Autres Services, les Droits de Douane et les Services de l'Etat). De même, la désagrégation par branches de production des IMPORTATIONS et EXPORTATIONS annuelles a permis d'en faire apparaître 9 (les Aliments, les Hydrocarbures, les Matières Premières, les Semi-Ouvrées, les Equipements Agricoles, les Equipements Industriels, les Biens de Consommation, les Services Non Financiers et les Ajustements). Cette première décomposition nous permet donc d'isoler le secteur auquel nous nous intéressons tout particulièrement : les Hydrocarbures.

Si l'Algérie importe des hydrocarbures (en quantités minimes, il est vrai), ce n'est certainement pas du G.N.L. Il est donc inutile (pour notre exemple) de faire porter notre effort de désagrégation sur les hydrocarbures importés annuellement. Par contre, les hydrocarbures qui entrent en jeu dans le calcul de la P.I.B. annuelle peuvent être scindés en 7 catégories (Pétrole Brut destiné à la consommation intérieure, Produits Raffinés destinés à la consommation intérieure, Gaz destiné à la consommation intérieure, Pétrole Brut destiné à l'exportation, Produits Raffinés destinés à l'exportation, Gaz destiné à l'exportation, et G.N.L. destiné à l'exportation), et ceux qui permettent de connaître le montant des exportations annuelles peuvent être répartis en 3 grandes classes (Pétrole Brut, Produits Raffinés, G.N.L.).

Pour faire apparaître le G.N.L. exporté dans l'équation de base, il suffira donc d'écrire que :

$$\text{EXPORTATIONS}(T) = \sum_{\substack{\text{branche de} \\ \text{production}}} \text{EXPORTATIONS (Branche de production, T)}$$

ou encore

$$\begin{aligned} \text{EXPORTATIONS}(T) = & \sum_{\substack{\text{branche de} \\ \text{production} \neq \text{hydrocarbures}}} \text{EXPORTATIONS (Branche de production, T)} \\ & + \sum_{\substack{\text{nature des} \\ \text{HC} \neq \text{GNL}}} \text{EXPORT d'HC (nature des HC, T)} \\ & + \text{EXPORTATIONS (GNL, T)} \end{aligned}$$

et que :

$$\text{PIB}(T) = \sum_{\text{secteur PIB}} \text{PIB (secteur PIB, T)}$$

ou encore

$$\begin{aligned} \text{PIB}(T) = & \sum_{\substack{\text{secteur PIB} \\ \neq \text{hydrocarbures}}} \text{PIB (secteur PIB, T)} \\ & + \sum_{\substack{\text{nature des HC} \\ \neq \text{GNL exporté}}} \text{PIB des HC (nature des HC, T)} \\ & + \text{PIB (GNL exporté, T)} \end{aligned}$$

Nous supposons connus (évalués par des moyens extérieurs au modèle), les taux de croissance annuelle de la PIB, des IMPORTATIONS, des EXPORTATIONS et de la CONSOMMATION par branche d'activité. De même une projection sur 4 ans de la tendance actuelle des taux d'inflation concernant les 3 derniers éléments (projection faite dans le cadre du plan quadriennal) nous permet de connaître les taux d'inflation annuelle utiles. Si de plus, on fournit d'une part, les données 79 (exprimées en millions de DA, en quantités = prix unitaire) de chacun des composants désagrégés de l'équation de base, et d'autre part les moyens de connaître leur évolution (équations), nous sommes en mesure de répondre à la question initiale.

#### IV.2.2. SA TRADUCTION EN LANGAGE S.A.M.S.S.

Passer de la définition théorique d'un modèle à sa formulation S.A.M.S.S. n'est qu'une question de patience et de rigueur. Pour ce faire, l'utilisateur dispose du "Manuel de Référence du logiciel SAMSS" [7].

L'automatisation du modèle via SAMSS peut se décomposer en deux grandes phases :

- . Description du Modèle (vérification de la syntaxe et de la cohérence des deux premiers textes utilisateurs et codification de ceux-ci pour traitement par l'ordinateur),
- . Résolution Numérique et Interrogation des Résultats (analyse des données fournies par l'utilisateur, exécution de la simulation, édition des résultats demandés),

pour lesquelles l'utilisateur doit fournir 4 descriptions :

- . Description des composants caractéristiques du phénomène étudié.
- . Description des liens entre ces composants.
- . Description des initialisations de certains de ces composants.
- . Description des requêtes d'interrogation de la Base des Opérandes du Modèle.

IV.2.2.1. Constitution des textes utilisateursDescription des composants caractéristiques du phénomène

```

1  INSERER
2  INDICE
3  TYPHC DE 1 A 7;      (* TYPES D'HYDROCARBURES *)
4                      (* 1 PETROLE BRUT DESTINE A LA CONSOMMATION INTERNE*)
5                      (* 2 PRODUITS RAFFINES " " " " " " *)
6                      (* 3 GAZ " " " " " " *)
7                      (* 4 PETROLE BRUT DESTINE A L'EXPORTATION *)
8                      (* 5 PRODUITS RAFFINES " " " " " *)
9                      (* 6 GAZ " " " " " *)
10                     (* 7 G.N.L. " " " " *)
11  ELPID DE 1 A 11;    (* ELEMENTS DE LA P.I.B: *)
12                     (* 1 AGRICULTURE *)
13                     (* 2 HYDROCARBURES *)
14                     (* 3 MINES ET CARRIERES *)
15                     (* 4 ENERGIE ET EAU *)
16                     (* 5 INDUSTRIES MANUFACTURIERES *)
17                     (* 6 B.T.F. *)
18                     (* 7 TRANSPORTS *)
19                     (* 8 COMMERCES *)
20                     (* 9 AUTRES SERVICES *)
21                     (* 10 DROITS DE DOUANES *)
22                     (* 11 SERVICES DE L'ETAT *)
23  IE DE 1 A 9;        (* CLASSIFICATION DES IMPORT-EXPORT: *)
24                     (* 1 ALIMENTS *)
25                     (* 2 HYDROCARBURES *)
26                     (* 3 MATIERES PREMIERES *)
27                     (* 4 SEMI-OUVREES *)
28                     (* 5 EQUIPEMENT AGRICOLE *)
29                     (* 6 EQUIPEMENT INDUSTRIEL *)
30                     (* 7 BIENS DE CONSOMMATION *)
31                     (* 8 SERVICES N.F. *)
32                     (* 9 AJUSTEMENT *)
33  NATHC DE 1 A 3;     (* NATURE DE HC EXPORTES: *)
34                     (* 1 PETROLE BRUT *)
35                     (* 2 PRODUITS RAFFINES *)
36                     (* 3 G.N.L. *)
37  TALIM DE 1 A 8;     (* TYPES D'ALIMENTS IMPORTES: *)
38                     (* 1 LAIT *)
39                     (* 2 BLE *)
40                     (* 3 AUTRES CEREALES *)
41                     (* 4 PRODUITS CEREALIERES *)
42                     (* 5 SUCRE *)
43                     (* 6 GRAINES OLEAGINEUSES *)
44                     (* 7 CORPS GRAS *)
45                     (* 8 VIANDES ET DIVERS *)
46  T 1979, 1980, 1981, 1982, 1983, 1984; (* DATES DE SIMULATION *)
47
48
49  OPERANDE
50  PRIX(TYPHC,T) REEL DONIN; (* PRIX DES HC PAR CLASSE D'HC A L'INSTANT T *)
51  TCPHC(TYPHC) REEL DONSC; (* TAUX DE CROISSANCE DES PRIX DES HC *)
52  PIBEXCEPTHC(T) REEL; (* PIB NON COMPRIS CELLE ISSUE DES HC *)
53  PIB(ELPID,T) REEL DONIN; (* PIB PAR ELEMENT DE LA PIB A L'INSTANT T *)
54  TCPID(ELPID) REEL DONEX; (* TAUX DE CROISSANCE DE LA PIB *)
55  PIBHCNLOPP(T) REEL; (* PIB ISSUE DES HC NON COMPRIS LE GNL *)
56  TICHC(TYPHC,T) REEL; (* PIB ISSUE DES HC PAR TYPE D'HC *)
57  PRIXPETINT(T) REEL; (* PRIX DU PETROLE BRUT SUR LE MARCHÉ INT. *)
58  COEFTRANSF REEL DONSC; (* COEF. PASSAGE DES PRIX EXT. AUX PRIX INT. *)
59  PRIXPRINT REEL DONIN; (* PRIX DES PRODUITS RAFFINES SUR LE MARCHÉ IN-
60  TERIEUR *)
61  QUANT(TYPHC,T) REEL DONIN; (* QUANTITES D'HC PAR CLASSE D'HC *)
62  TCQUANTHC(TYPHC) REEL DONSC; (* TAUX DE CROISSANCE DES QUANTITES D'HC *)
63  QPETPROD(T) REEL DONEX; (* QUANTITE DE PETROLE PRODUIT A T *)
64  EXPORTCOFF(T) REEL; (* COUT DES EXPORTS HORS HC *)
65  EXPORT(IE,T) REEL DONIN; (* COUT DES EXPORTS PAR CLASSE D'EXPORT *)
66  TCEXP(IE) REEL DONEX; (* TAUX DE CROISSANCE DES EXPORTS PAR CLASSE D'
67  EXPORTS *)
68  TIEXP(IE) REEL DONEX; (* TAUX D'INFLATION DES EXPORTS PAR CLASSE D'
69  EXPORTS *)
70  EXPHCNLOPP(T) REEL; (* COUT DES EXPORTS D'HC HORS GNL *)
71  EXPORHC(NATHC,T) REEL; (* COUT DES EXPORTS D'HC PAR CLASSE D'HC *)
72  IMPORT(T) REEL; (* COUT DES IMPORTS A L'INSTANT T *)
73  IMPORT(IE,T) REEL DONIN; (* COUT DES IMPORTS PAR CLASSE D'IMPORT *)
74  TCIMP(IE) REEL DONEX; (* TAUX DE CROISSANCE DES IMPORTS PAR CLASSE D'
75  IMPORTS *)
76  TIIMP(IE) REEL DONEX; (* TAUX D'INFLATION DES IMPORTS PAR CLASSE D'
77  IMPORTS *)
78  INFALIM(TALIM,T) REEL DONIN; (* COUT DES IMPORTATIONS ALIMENTAIRES PAR CLAS-
79  SE D'ALIMENTS *)
80  CONSON(T) REEL; (* COUT DE LA CONSOMMATION A L'INSTANT T *)
81  CONSPUBL(T) REEL DONIN; (* COUT DE LA CONSOMMATION PUBLIQUE *)
82  CONSPRIV(T) REEL DONIN; (* COUT DE LA CONSOMMATION PRIVEE *)
83  TCSPUBL REEL DONEX; (* TAUX DE CROISSANCE DE LA CONS. PUBLIQUE *)
84  TCSPRIV REEL DONEX; (* TAUX DE CROISSANCE DE LA CONS. PRIVEE *)
85  TICSPUBL REEL DONEX; (* TAUX D'INFLATION DE LA CONS. PUBLIQUE *)
86  TICSPRIV REEL DONEX; (* TAUX D'INFLATION DE LA CONS. PRIVEE *)
87  INVEST(T) REEL DONSC; (* COUT DES INVESTISSEMENTS A L'INSTANT T *)
88  QUANTGNL(T) REEL; (* QUANTITE DE GNL EXPORTEE A L'INSTANT T *)
89  QUANTPETCONS(T) REEL; (* QUANTITE DE PETROLE BRUTE CONSOMMEE SUR LE
90  MARCHÉ INTERIEUR *)
91  QGAZEXP(T) REEL; (* QUANTITE GAZ POUR GNL EXPORTEE *)
92  QGAZPROD(T) REEL; (* QUANTITE DE GAZ A PRODUIRE *)
93  RESCAP(T) REEL; (* TAUX DE COUVERTURE DES IMPORTATIONS *)
94  PIB(T) REEL; (* PRODUCTION INTERIEURE BRUTE *)
95  EXPORT(T) REEL; (* EXPORTATIONS *)
96  FINHAJ

```

Description des relations entre composants

```

1
2  (*@CALCUL DES PRIX DES HYDROCARBURES A L'INSTANT T *)
3  FOR TYPHC := MINTYPHC TO MAXTYPHC DO
4    PRIX(TYPHC,T) := PRIX(TYPHC,T-1) * (1 + TCFHC(TYPHC));
5
6  (*@CALCUL DE LA PIB HORS HYDROCARBURES A L'INSTANT T *)
7  $VARINTR := 0;
8  FOR ELPFB := MINELPFB TO MAXELPFB DO
9    IF ELPFB <> 2 THEN $VARINTR := $VARINTR + PIB(ELPFB,T);
10  PIBEXCEPTHC(T) := $VARINTR;
11
12  (*@CALCUL DE LA PIB PAR TYPE D'ELEMENT DE LA PIB *)
13  FOR ELPFB := MINELPFB TO MAXELPFB DO
14    IF ELPFB <> 2 THEN PIB(ELPFB,T) := PIB(ELPFB,T-1) * (1+TCPB(ELPFB));
15
16  (*@CALCUL DE LA PIB ISSUE DES HC HORS GNL *)
17  $VARINTR := 0;
18  FOR TYPHC := MINTYPHC TO MAXTYPHC -1 DO
19    $VARINTR := $VARINTR + PIBHC(TYPHC,T);
20  PIBHCGNLOFF(T) := $VARINTR;
21
22  (*@CALCUL DU PRIX DU PETROLE BRUT SUR LE MARCHÉ INTERIEUR *)
23  TYPHC := 4;
24  PRIXPETINT(T) := COEFTRANSF * PRIX(TYPHC,T);
25
26  (*@CALCUL DU PRIX DES PRODUITS RAFFINES SUR LE MARCHÉ INTERIEUR *)
27  TYPHC := 5;
28  PRIXPRINT(T) := (COEFTRANSF * PRIX(TYPHC,T)) - (1.275 * PRIXPETINT(T));
29
30  (*@CALCUL DE LA PIB ISSUE DES HC PAR TYPE D'HYDROCARBURES *)
31  FOR TYPHC := MINTYPHC TO MAXTYPHC -1 DO
32    CASE TYPHC:INTEGER OF
33      (* VA PETROLE BRUT *)
34      1,4: PIBHC(TYPHC,T) := QUANT(TYPHC,T) * PRIXPETINT(T);
35      (* VA PRODUITS RAFFINES *)
36      2,5: PIBHC(TYPHC,T) := QUANT(TYPHC,T) * PRIXPRINT(T);
37      (* VA GAZ DESTINE A LA CONSOMMATION INTERNE *)
38      3: PIBHC(TYPHC,T) := QUANT(TYPHC,T) * PRIX(TYPHC,T);
39      (* VA GAZ EXPORTÉ SOUS FORME DE GAZ *)
40      6: PIBHC(TYPHC,T) := 0;
41    END;
42
43  (*@CALCUL DES QUANTITES D'HYDROCARBURES PAR TYPE D'HYDROCARBURES *)
44  FOR TYPHC := MINTYPHC TO MAXTYPHC - 1 DO
45    IF TYPHC <> 4 THEN QUANT(TYPHC,T) := QUANT(TYPHC,T-1) * (1 + TCQUANTHC(TYPHC))
46    ELSE QUANT(TYPHC,T) := QPETPROD(T) - QUANTPETCONS(T);
47
48  (*@CALCUL DE LA QUANTITE DE PETROLE CONSOMME SUR LE MARCHÉ INTERIEUR *)
49  TYPHC := 1;
50  QUANTPETCONS(T) := QUANT(TYPHC,T-1) * (1 + TCQUANTHC(TYPHC));
51
52  (*@CALCUL DU COUT DES EXPORTATIONS EXCEPTÉES CELLES D'HYDROCARBURES *)
53  $VARINTR := 0;
54  FOR IE := MINIE TO MAXIE DO
55    IF IE <> 2 THEN $VARINTR := $VARINTR + EXPORT(IE,T);
56  EXPORTHCOFF(T) := $VARINTR;
57
58  (*@CALCUL DU COUT DES EXPORTATIONS PAR CLASSE D'EXPORTATION *)
59  FOR IE := MINIE TO MAXIE DO
60    IF IE <> 2 THEN EXPORT(IE,T) := EXPORT(IE,T-1) * (1+TCEXP(IE)+TIEXP(IE));
61
62  (*@CALCUL DU COUT DES EXPORTATIONS D'HYDROCARBURES HORS GNL *)
63  $VARINTR := 0;
64  FOR NATHC := MINNATHC TO MAXNATHC - 1 DO
65    $VARINTR := $VARINTR + EXPORTHC(NATHC,T);
66  EXPHCGNLOFF(T) := $VARINTR;
67
68  (*@CALCUL DU COUT DES EXPORTATIONS D'HC PAR NATURE D'HYDROCARBURES *)
69  FOR NATHC := MINNATHC TO MAXNATHC - 1 DO
70    BEGIN
71      TYPHC := NATHC + 3;
72      EXPORTHC(NATHC,T) := QUANT(TYPHC,T) * PRIX(TYPHC,T);
73    END;
74
75  (*@CALCUL DU COUT DES IMPORTATIONS *)
76  $VARINTR := 0;
77  FOR IE := MINIE TO MAXIE DO
78    $VARINTR := $VARINTR + IMPORT(IE,T);
79  IMPORT(T) := $VARINTR;
80
81  (*@CALCUL DU COUT DES IMPORTATIONS PAR TYPE D'IMPORTATION *)
82  FOR IE := MINIE TO MAXIE DO
83    IF IE <> 1 THEN IMPORT(IE,T) := IMPORT(IE,T-1) * (1+TCIMP(IE)+TIIMP(IE))
84    ELSE BEGIN
85      $VARINTR := 0;
86      FOR TALIM := MINALIM TO MAXALIM DO
87        $VARINTR := $VARINTR + IMPALIM(TALIM,T);
88      IMPORT(IE,T) := $VARINTR;
89    END;
90
91  (*@CALCUL DU COUT DES IMPORTATIONS ALIMENTAIRES PAR TYPE D'ALIMENT *)
92  IF := 1;
93  FOR TALIM := MINTALIM TO MAXTALIM DO
94    IMPALIM(TALIM,T) := IMPALIM(TALIM,T-1) * (1+TCIMP(IE)+TIIMP(IE));
95
96  (*@CALCUL DU COUT DE LA CONSOMMATION *)
97  CONSONH(T) := CONSPUBL(T) + CONSPRIV(T);

```



```

98
99 (*@CALCUL DU COUT DE LA CONSOMMATION PUBLIQUE *)
100 CONSPUBL(T) := CONSPUBL(T-1) * (1+TCCPUBL+TICPUBL);
101
102 (*@CALCUL DU COUT DE LA CONSOMMATION PRIVEE *)
103 CONSPRIV(T) := CONSPRIV(T-1) * (1+TCCPRIV+TICPRIV);
104
105 (*@CALCUL DE LA QUANTITE DE GNL A EXPORTER POUR EQUILIBRER LA COMPTA *)
106 TYPHC := 7;
107 $VARINTR1 := PRIX(TYPHC,T);
108 TYPHC := 6;
109 $VARINTR2 := PRIX(TYPHC,T);
110 QUANTGNL(T) := ((PIBEXCEPTHC(T) + PIBHCGNLOFF(T)) +
111                IMPORT(T)
112                (EXPORTHCOFF(T) + EXPORTHCGNLOFF(T)) -
113                CONSUM(T)
114                INVEST(T)
115                ) / ($VARINTR1 - (0.816 * $VARINTR1) + (1.816 * $VARINTR2));
116
117 (*@CALCUL DE LA QUANTITE DE GAZ EXPORTEE SOUS FORME GNL *)
118 QGAZEXP(T) := QUANTGNL(T) * 1.816;
119
120 (*@CALCUL DU TAUX DE COUVERTURE DES IMPORTATIONS *)
121 TYPHC := 7;
122 $VARINTR := EXPORTHCOFF(T) + EXPHCGNLOFF(T) + (QUANTGNL(T) * PRIX(TYPHC,T));
123 RESGAF(T) := ($VARINTR - IMPORT(T)) / IMPORT(T);
124
125 (*@CALCUL DE LA QUANTITE DE GAZ A PRODUIRE *)
126 TYPHC := 3;
127 QGAZPROD(T) := QGAZEXP(T) + QUANT(TYPHC,T);
128
129 (*@CALCUL DE LA PIB(T) *)
130 TYPHC := 7;
131 $VARINTR1 := 0.816 * PRIX(TYPHC,T);
132 TYPHC := 6;
133 $VARINTR := 1.816 * QUANTGNL(T) * PRIX(TYPHC,T);
134 PIB(T) := PIBEXCEPTHC(T) + PIBHCGNLOFF(T) + $VARINTR1 - $VARINTR2;
135
136 (*@CALCUL EXPORT(T) *)
137 TYPHC := 7;
138 EXPORT(T) := EXPORTHCOFF(T) + EXPHCGNLOFF(T) + (QUANTGNL(T) * PRIX(TYPHC,T));
139
140 (*@CONTROLE PIB(T) - EXPORT(T) + IMPORT(T) = CONSUM(T) + INVEST(T) *)
141 $VARINTR := PIB(T) - EXPORT(T) + IMPORT(T) - CONSUM(T) - INVEST(T);
142 CONTROL($VARINTR,EG,1,CONTINUE,2001);
143 (*@FIN*)

```

### Description des initialisations de certains composants

```

1
2 (*--COMPTADZ--DOHIN--*)
3 PRIX(TYPHC,T) (*MILLIERS DA/T*)
4 POUR 1979:
5
6 0.65,
7 0.629,
8 0.899,
9 0.027,
10 0.2424;
11
12 PIB(ELPIB,T) (*MILLIONS DA*)
13 POUR 1979:
14 7840,
15 33700,
16 410,
17 1540,
18 14280,
19 16220,
20 5265,
21 15650,
22 7325,
23 5880,
24 12870;
25
26 EXPORT(IE,T) (*MILLIONS DA*)
27 POUR 1979:
28 431,
29 33255,
30 261,
31 188,
32 1,
33 4,
34 4,
35 2517,
36 1339;
37
38 IMPORT(IE,T) (*MILLIONS DA*)
39 POUR 1979:
40 5081,
41 621,
42 1687,
43 7689,
44 222,
45 11777,
46 4358,
47 11975,
48 2910;
49
50 IMPALIM(TALIM,T) (*MILLIONS DA*)
51 POUR 1979:
52 336,
53 944,
54 458,
55 686,
56 423,
57 197,
58 530,
59 1507;

```

57	CONSPUBL(T)	(*MILLIONS DA*)
58	POUR 1979:	17470;
59		
60	CONSPRIV(T)	(*MILLIONS DA*)
61	POUR 1979:	55310;
62		
63	QUANTHC(TYPHC,T)	(*MILLIERS DE TONNES*)
64	POUR 1979:	7519,
65		5025,
66		4136,
67		49013,
68		3235,
69		*
70		9117;
71		
72		(*--COMPTADZ--DONSC ET DONEX--*)
73	TCPHC(TYPHC):	0,
74		0,
75		0,
76		0.15,
77		0.15,
78		0,
79		0.15; (*TAUX DE CROISSANCE PRIX GNL EXPORT*)
80		
81	TCPIB(ELPIB):	0.137,
82		*
83		0.179,
84		0.137,
85		0.192,
86		0.166,
87		0.233,
88		0.233,
89		0.233,
90		0.256,
91		0.268;
92		
93	COEFTRANSF:	0.816; (*PASSAGE PRIX EXTERNES AUX PRIX INTERNES*)
94		
95	TCQUANTHC(TYPHC):	0.08,
96		0.08,
97		0.08,
98		*
99		0,
100		0,
101		*
102		
103		
104	QPETPROD(T)	(*MILLIERS DE TONNES*)
105		
106	POUR 1979:	56532
107	POUR 1980:	56532
108	POUR 1981:	56532
109	POUR 1982:	56532
110	POUR 1983:	56532
111	POUR 1984:	56532;
112		
113	TCEXP(IE):	-0.099,
114		*
115		0.134,
116		-0.028,
117		-0.5,
118		-0.5,
119		-0.5,
120		0.036,
121		-0.5;
122		
123	TIEXP(IE):	0.12,
124		*
125		0,
126		0.103,
127		-0.5,
128		-0.5,
129		-0.5,
130		0.103,
131		-0.5;
132		
133	TCIMP(IE):	0.10,
134		0,
135		0.041,
136		0.042,
137		0.043,
138		0.044,
139		0.095,
140		0.03,
141		-0.5;
142		
143	TIIMP(IE):	0.07,
144		0.152,
145		0.152,
146		0.151,
147		0.15,
148		0.149,
149		0.15,
150		0.10,
151		-0.5;
152		
153	TCCPUBL:	0.118;
154		
155	TICPUBL:	0.106;
156		
157	TCCPRIV:	0.095;
158		
159	TICPRIV:	0.103;
160		
161	INVEST(T)	(*MILLIONS DE DA*)
162	POUR 1979:	50700
163	POUR 1980:	69534
164	POUR 1981:	80103
165	POUR 1982:	92278
166	POUR 1983:	106304
167	POUR 1984:	122463;
168	ENDINIT	

. Description des requêtes d'interrogation de la Base des Opérandes

```
1 PIB(T) FORMAT 2 UNITE MILLIONS-DINARS COMMENT PRODUCTION INTERIEURE BRUTE;  
2 EXPORT(T) UNITE MILLIONS_DINARS;  
3 IMPORT(T) UNITE MILLIONS_DINARS;  
4 CONSON(T) UNITE MILLIONS_DINARS;  
5 INVEST(T) UNITE MILLIONS_DINARS;  
6 RESCAP(T) COMMENT POURCENTAGE DE COUVERTURE DES IMPORTATIONS;  
7 QGAZPROD(T) UNITE MILLIONS_M3 COMMENT PRODUCTION DE GAZ;  
8 QPETCONS(T) UNITE MILLIERS_DE_TONNES COMMENT CONSOMMATION INTERIEURE DE PETROLE;  
9 QUANTGNL(T) UNITE MILLIERS_DE_TONNES COMMENT EXPORTATION DE GNL;  
10 QUANT(TYPHC,T) UNITE MILLIERS_DE_TONNES COMMENT PRODUCTION D'HYDROCARBURES;  
11 ENDANAL
```

#### IV.2.2.2. Automatisation proprement dite

Cette première étape franchie, il faut mettre en oeuvre le logiciel S.A.M.S.S. Nous verrons plus loin l'ensemble des possibilités techniques de mise en oeuvre, mais avant nous allons suivre pas à pas l'automatisation de notre exemple.

1. On demande l'exécution complète, depuis la création du dictionnaire jusqu'à l'édition des résultats. La présence d'une unité syntaxique (commentaire en l'occurrence) de plus de 256 caractères entraîne l'apparition de plus de 50 erreurs dans le module de création du dictionnaire d'où l'arrêt de l'automatisation, le dictionnaire n'est pas créé.
2. Après correction (par update) de la description des indices et variables du modèle, on relance l'exécution. Celle-ci sera de nouveau arrêtée après l'étape d'analyse de cette description, sur constat de la présence d'une variable dont le nom générique comporte plus de 11 caractères (max autorisé). Le dictionnaire du modèle n'est toujours pas créé.
3. Après une nouvelle correction, l'exécution demandée (à partir de l'étape 1) permet la création du dictionnaire, la réservation et le formatage de la base des opérandes associée à ce dictionnaire, et entame l'analyse du texte des relations. Celle-ci révèle la présence de 5 erreurs toutes dues à l'orthographe incorrecte de variables ou d'indices.
4. On corrige (par update) la description des relations et on relance l'exécution, mais uniquement à partir de l'analyse de cette description. En effet le dictionnaire et la base des opérandes restent inchangés. La syntaxe de la description des relations étant correcte, l'automatisation se poursuit par la génération du graphe d'ordonnement des relations puis par les contrôles de cohérences, ceux-ci faisant apparaître 9 incohérences probables.

FICHIER FIMPR

```

*****
** VERSION SAMSS = VEO2          STEP = CREDIC          COMPILER LE 81.280          EXECUTE LE 82.280 A 18 HEURE(S)
**
** MODELE COMPTADZ ALGERIE
**
** PARAMETRES UTILISES : CREDICO NOEDITCL NOTRACE
**
** ZONE DYNAMIQUE DISPONIBLE AVANT RECUPERATION DU DICO : 520200
*****

```

```

1 INSERER
2 INDICE
3 TYPHC DE 1 A 7; (* TYPES D'HYDROCARBURES *)
4 (* 1 PETROLE BRUT DESTINE A LA CONSOMMATION INTERNE*)
5 (* 2 PRODUITS RAFFINES " " " " " *)
6 (* 3 GAZ " " " " " *)
7 (* 4 PETROLE BRUT DESTINE A L'EXPORTATION *)
8 (* 5 PRODUITS RAFFINES " " " " *)
9 (* 6 GAZ " " " " *)
10 (* 7 G.N.L. " " " " *)
11 ELPID DE 1 A 11; (* ELEMENTS DE LA P.I.B: *)
12 (* 1 AGRICULTURE *)
13 (* 2 HYDROCARBURES *)
14 (* 3 MINES ET CARRIERES *)
15 (* 4 ENERGIE ET EAU *)
16 (* 5 INDUSTRIES MANUFACTURIERES *)
17 (* 6 B.T.P. *)
18 (* 7 TRANSPORTS *)
19 (* 8 COMMERCES *)
20 (* 9 AUTRES SERVICES *)
21 (* 10 DROITS DE DOUANES *)
22 (* 11 SERVICES DE L'ETAT *)
23 IE DE 1 A 9; (* CLASSIFICATION DES IMPORT-EXPORT: *)
24 (* 1 ALIMENTS *)
25 (* 2 HYDROCARBURES *)
26 (* 3 MATIERES PREMIERES *)
27 (* 4 SEMI-OUVREES *)
28 (* 5 EQUIPEMENT AGRICOLE *)
29 (* 6 EQUIPEMENT INDUSTRIEL *)
30 (* 7 BIENS DE CONSOMMATION *)
31 (* 8 SERVICES N.P. *)
32 (* 9 AJUSTEMENT *)
33 NATHC DE 1 A 3; (* NATURE DE HC EXPORTES: *)
34 (* 1 PETROLE BRUT *)
35 (* 2 PRODUITS RAFFINES *)
36 (* 3 G.N.L. *)
37 TALIM DE 1 A 8; (* TYPES D'ALIMENTS IMPORTES: *)
38 (* 1 LAIT *)
39 (* 2 BLE *)
40 (* 3 AUTRES CEREALES *)
41 (* 4 PRODUITS CEREALIERES *)
42 (* 5 SUCRE *)
43 (* 6 GRAINES OLEAGINEUSES *)
44 (* 7 CORPS GRAS *)
45 (* 8 VIANDES ET DIVERS *)
46 T 1979, 1980, 1981, 1982, 1983, 1984; (* DATES DE SIMULATION *)
47
48
49 OPERANDE
50 PRIX(TYPHC,T) REEL DONIN; (* PRIX DES HC PAR CLASSE D'HC A L'INSTANT T *)
51 TYPHC(TYPHC) REEL DONSC; (* TAUX DE CROISSANCE DES PRIX DES HC *)
52 PEXCEPTHC(T) REEL; (* PIB NON COMPRIS CELLE ISSUE DES HC *)
53 PIB(ELPID,T) REEL DONIN; (* PIB PAR ELEMENT DE LA PIB A L'INSTANT T *)
54 TCPID(ELPID) REEL DONEX; (* TAUX DE CROISSANCE DE LA PIB *)
55 PIBCGNLOFF(T) REEL; (* PIB ISSUE DES HC NON COMPRIS LE GNL *)
56 PIBHC(TYPHC,T) REEL; (* PIB ISSUE DES HC PAR TYPE D'HC *)
57 PRIXPETINT(T) REEL; (* PRIX DU PETROLE BRUT SUR LE MARCHE INT. *)
58 COEFTRANSF REEL DONSC; (* COEF. PASSAGE DES PRIX EXT. AUX PRIX INT. *)
59 PRIXPRINT(T) REEL DONIN; (* PRIX DES PRODUITS RAFFINES SUR LE MARCHE
60
61 INTERIEUR *)
62 QUANT(TYPHC,T) REEL DONIN; (* QUANTITES D'HC PAR CLASSE D'HC *)
63 TCQUANTHC(TYPHC) REEL DONSC; (* TAUX DE CROISSANCE DES QUANTITES D'HC *)
64 QPETPROD(T) REEL DONEX; (* QUANTITE DE PETROLE PRODUIT A T *)
65 EXPORTHCOFF(T) REEL; (* COUT DES EXPORTS HORS HC *)
66 EXPORT(IE,T) REEL DONIN; (* COUT DES EXPORTS PAR CLASSE D'EXPORT *)
67 TCEXP(IE) REEL DONEX; (* TAUX DE CROISSANCE DES EXPORTS PAR CLASSE D'
68 EXPORTS *)
69 TIEXP(IE) REEL DONEX; (* TAUX D'INFLATION DES EXPORTS PAR CLASSE D'
70 EXPORTS *)
71 EXPHCNLOFF(T) REEL; (* COUT DES EXPORTS D'HC HORS GNL *)
72 EXPORTHC(NATHC,T) REEL; (* COUT DES EXPORTS D'HC PAR CLASSE D'HC *)
73 IMPORT(T) REEL; (* COUT DES IMPORTS A L'INSTANT T *)
74 IMPORT(IE,T) REEL DONIN; (* COUT DES IMPORTS PAR CLASSE D'IMPORT *)
75 TCIMP(IE) REEL DONEX; (* TAUX DE CROISSANCE DES IMPORTS PAR CLASSE D'
76 IMPORTS *)
77 TIIMP(IE) REEL DONEX; (* TAUX D'INFLATION DES IMPORTS PAR CLASSE D'
78 IMPORTS *)
79 IMPALIM(TALIM,T) REEL DONIN; (* COUT DES IMPORTATIONS ALIMENTAIRES PAR CLAS
80 SE D'ALIMENTS *)
81 CONSON(T) REEL; (* COUT DE LA CONSOMMATION A L'INSTANT T *)
82 CONSPUBL(T) REEL DONIN; (* COUT DE LA CONSOMMATION PUBLIQUE *)
83 CONSPRIV(T) REEL DONIN; (* COUT DE LA CONSOMMATION PRIVEE *)
84 TCCPUBL REEL DONEX; (* TAUX DE CROISSANCE DE LA CONS. PUBLIQUE *)
85 TCCPRIV REEL DONEX; (* TAUX DE CROISSANCE DE LA CONS. PRIVEE *)
86 TICPUBL REEL DONEX; (* TAUX D'INFLATION DE LA CONS. PUBLIQUE *)
87 TICPRIV REEL DONEX; (* TAUX D'INFLATION DE LA CONS. PRIVEE *)
88 INVEST(T) REEL DONSC; (* COUT DES INVESTISSEMENTS A L'INSTANT T *)
89 QUANTGNL(T) REEL; (* QUANTITE DE GNL EXPORTEE A L'INSTANT T *)
90 QPETCONS(T) REEL; (* QUANTITE DE PETROLE BRUTE CONSOMMEE SUR LE
91 MARCHE INTERIEUR *)
92 QGAZEXP(T) REEL; (* QUANTITE DE GAZ POUR GNL EXPORTEE *)
93 QGAZPROD(T) REEL; (* QUANTITE DE GAZ A PRODUIRE *)
94 RESGAP(T) REEL; (* TAUX DE COUVERTURE DES IMPORTATIONS *)
95 PIB(T) REEL; (* PRODUCTION INTERIEURE BRUTE *)
96 EXPORT(T) REEL; (* EXPORTATIONS *)

```

```

STATISTIQUES DE MISE A JOUR DU DICTIONNAIRE
NR INDICES INSERES : 6 NB INDICES SUPPRIMES : 0 NB INDICES MODIFIES : 0 NB INDICES CONNUS : 6
NB OPERANDES INSERES : 39 NB OPERANDES SUPPRIMES : 0 NB OPERANDES MODIFIES : 0 NB OPERANDES CONNUS : 39
NB COMBINAISONS RECONNUES : 10
TAILLE DU DICTIONNAIRE DU MODELE : 42520CTETS
LE PROGRAMME DE MISE A JOUR DU DICO DU MODELE NE COMPORTE PAS D'ERREUR

```

FICHIER FIMPR

```

*****
** VERSION SAMSS = VFD2          STEP = FMATBD          COMPILER LE 81.280          EXECUTE LE 82.280 A 18 HEURE(S) **
** COMPTADZ ALGERIE **
** PARAMETRES UTILISES NOTRACE          NOEDITCL          NOCREDICO **
**
** TAILLE DE LA ZONE DYNAMIQUE DISPONIBLE =          491528
*****
** ZONE DYNAMIQUE OCCUPEE PAR LE DICTIONNAIRE =          4252
** ZONE DYNAMIQUE DISPONIBLE APRES CHARGEMENT DU DICTIONNAIRE =          487276
** NRVALT =          6          NBPAGSECT =          1          NBPAGSEGO =          1
*****
LE PROGRAMME DE FORMATTAGE DE LA BASE DES OPERANDES NE COMPORTE PAS D'ERREUR

```

FICHIER FIMPR

```

*****
** VERSION SAMSS = VFD2          STEP = MACROG          COMPILER LE 81.331          EXECUTE LE 82.280 A 18 HEURE(S) **
**
** NODELE COMPTADZ ALGERIE **
**
** PARAMETRES UTILISES : FIRSTMACROG NOEDITCL NOTRACE NOCREDICO NOTRLEG **
**
** ZONE DYNAMIQUE DISPONIBLE AVANT RECUPERATION DU DICO : 436224
*****
1
2 (*@CALCUL DES PRIX DES HYDROCARBURES A L'INSTANT T *)
3 FOR TYPHC := MINTYPHC TO MAXTYPHC DO
4   PRIX(TYPHC,T) := PRIX(TYPHC,T-1) * (1 + TCFHC(TYPHC));
5
6 (*@CALCUL DE LA PIB HORS HYDROCARBURES A L'INSTANT T *)
7 $VARINTR := 0;
8 FOR ELPB := MINELPB TO MAXELPB DO
9   IF ELPB <> 2 THEN $VARINTR := $VARINTR + PIB(ELPB,T);
10 PIBEXCEPTC(T) := $VARINTR;
11
12 (*@CALCUL DE LA PIB PAR TYPE D'ELEMENT DE LA PIB *)
13 FOR ELPB := MINELPB TO MAXELPB DO
14   IF ELPB <> 2 THEN PIB(ELPB,T) := PIB(ELPB,T-1) * (1+TCPB(ELPB));
15
16 (*@CALCUL DE LA PIB ISSUE DES HC HORS GNL *)
17 $VARINTR := 0;
18 FOR TYPHC := MINTYPHC TO MAXTYPHC - 1 DO
19   $VARINTR := $VARINTR + PIBHC(TYPHC,T);
20 PIBHCNLOFF(T) := $VARINTR;
21
22 (*@CALCUL DU PRIX DU PETROLE BRUT SUR LE MARCHÉ INTERIEUR *)
23 TYPHC := 4;
24 PRIXPETINT(T) := COEFTRANSF * PRIX(TYPHC,T);
25
26 (*@CALCUL DU PRIX DES PRODUITS RAFFINES SUR LE MARCHÉ INTERIEUR *)
27 TYPHC := 5;
28 PRIXPRINT(T) := (COEFTRANSF * PRIX(TYPHC,T)) - (1.275 * PRIXPETINT(T));
29
30 (*@CALCUL DE LA PIB ISSUE DES HC PAR TYPE D'HYDROCARBURES *)
31 FOR TYPHC := MINTYPHC TO MAXTYPHC - 1 DO
32   CASE TYPHC OF
33     (* VA PETROLE BRUT *)
34     1,4: PIBHC(TYPHC,T) := QUANT(TYPHC,T) * PRIXPETINT(T);
35     (* VA PRODUITS RAFFINES *)
36     2,5: PIBHC(TYPHC,T) := QUANT(TYPHC,T) * PRIXPRINT(T);
37     (* VA GAZ DESTINE A LA CONSOMMATION INTERNE *)
38     3: PIBHC(TYPHC,T) := QUANT(TYPHC,T) * PRIX(TYPHC,T);
39     (* VA GAZ EXPORTÉ SOUS FORME DE GAZ *)
40     6: PIBHC(TYPHC,T) := 0;
41   END;
42
43 (*@CALCUL DES QUANTITES D'HYDROCARBURES PAR TYPE D'HYDROCARBURES *)
44 FOR TYPHC := MINTYPHC TO MAXTYPHC - 1 DO
45   IF TYPHC <> 4 THEN QUANT(TYPHC,T) := QUANT(TYPHC,T-1) * (1 + TCQUANTHC(TYPHC))
46   ELSE QUANT(TYPHC,T) := QPETPROD(T) - QPETCONS(T);
47
48 (*@CALCUL DE LA QUANTITE DE PETROLE CONSOMME SUR LE MARCHÉ INTERIEUR *)
49 TYPHC := 1;
50 QPETCONS(T) := QUANT(TYPHC,T-1) * (1 + TCQUANTHC(TYPHC));
51
52 (*@CALCUL DU COUT DES EXPORTATIONS EXCEPTÉES CELLES D'HYDROCARBURES *)
53 $VARINTR := 0;
54 FOR IE := MINIE TO MAXIE DO
55   IF IE <> 2 THEN $VARINTR := $VARINTR + EXPORT(IE,T);
56 EXPORHC(T) := $VARINTR;
57
58 (*@CALCUL DU COUT DES EXPORTATIONS PAR CLASSE D'EXPORTATION *)
59 FOR IE := MINIE TO MAXIE DO
60   IF IE <> 2 THEN EXPORT(IE,T) := EXPORT(IE,T-1) * (1+TCEXP(IE)+TIEXP(IE));
61

```

```

62 (*@CALCUL DU COUT DES EXPORTATIONS D'HYDROCARBURES HORS GNL *) 12
63 $VARINTR := 0;
64 FOR NATHC := MINNATHC TO MAXNATHC - 1 DO
65   $VARINTR := $VARINTR + EXPORTHC(NATHC,T);
66 EXPHCGNLOFF(T) := $VARINTR;
67

68 (*@CALCUL DU COUT DES EXPORTATIONS D'HC PAR NATURE D'HYDROCARBURES *) 13
69 FOR NATHC := MINNATHC TO MAXNATHC - 1 DO
70 BEGIN
71   TYPHC := NATHC + 3;
72   EXPORTHC(NATHC,T) := QUANT(TYPHC,T) * PRIX(TYPHC,T);
73 END;
74

75 (*@CALCUL DU COUT DES IMPORTATIONS *) 14
76 $VARINTR := 0;
77 FOR IE := MINIE TO MAXIE DO
78   $VARINTR := $VARINTR + IMPORT(IE,T);
79 IMPORT(T) := $VARINTR;
80

81 (*@CALCUL DU COUT DES IMPORTATIONS PAR TYFF D'IMPORTATION *) 15
82 FOR IE := MINIE TO MAXIE DO
83   IF IE <> 1 THEN IMPORT(IE,T) := IMPORT(IE,T-1) * (1+TCIMP(IE)+TIIMP(IE));
84 ELSE BEGIN
85   $VARINTR := 0;
86   FOR TALIM := MINTALIM TO MAXTALIM DO
87     $VARINTR := $VARINTR + IMPALIM(TALIM,T);
88   IMPORT(IE,T) := $VARINTR;
89 END;
90

91 (*@CALCUL DU COUT DES IMPORTATIONS ALIMENTAIRES PAR TYPE D'ALIMENT *) 16
92 IE := 1;
93 FOR TALIM := MINTALIM TO MAXTALIM DO
94   IMPALIM(TALIM,T) := IMPALIM(TALIM,T-1) * (1+TCIMP(IE)+TIIMP(IE));
95

96 (*@CALCUL DU COUT DE LA CONSOMMATION *) 17
97 CONSON(T) := CONSPUBL(T) + CONSPRIV(T);
98

99 (*@CALCUL DU COUT DE LA CONSOMMATION PUBLIQUE *) 18
100 CONSPUBL(T) := CONSPUBL(T-1) * (1+TCCPUBL+TICPUBL);
101

102 (*@CALCUL DU COUT DE LA CONSOMMATION PRIVEE *) 19
103 CONSPRIV(T) := CONSPRIV(T-1) * (1+TCCPRIV+TICPRIV);
104

105 (*@CALCUL DE LA QUANTITE DE GNL A EXPORTER POUR EQUILIBRER LA COMPTA *) 20
106 TYPHC := 7;
107 $VARINTR1 := PRIX(TYPHC,T);
108 TYPHC := 6;
109 $VARINTR2 := PRIX(TYPHC,T);
110 QUANTGNL(T) := ((PIBEXCEPTHC(T) + PIBHCGNLOFF(T)) +
111                IMPORT(T) -
112                (EXPORTHC(T) + EXPHCGNLOFF(T)) -
113                CONSON(T) -
114                INVEST(T)
115                ) / ($VARINTR1 - (0.816 * $VARINTR1) + (1.816 * $VARINTR2));
116

117 (*@CALCUL DE LA QUANTITE DE GAZ EXPORTEE SOUS FORME GNL *) 21
118 QGAZEXP(T) := QUANTGNL(T) * 1.816;
119

120 (*@CALCUL DU TAUX DE COUVERTURE DES IMPORTATIONS *) 22
121 RESCAP(T) := (EXPORT(T) - IMPORT(T)) / IMPORT(T);
122

123 (*@CALCUL DE LA QUANTITE DE GAZ A PRODUIRE *) 23
124 TYPHC := 3;
125 QGAZPROD(T) := QGAZEXP(T) + QUANT(TYPHC,T);
126

127 (*@CALCUL DE LA PIB(T) *) 24
128 TYPHC := 7;
129 $VARINTR1 := 0.816 * PRIX(TYPHC,T) * QUANTGNL(T);
130 TYPHC := 6;
131 $VARINTR := 1.816 * QUANTGNL(T) * PRIX(TYPHC,T);
132 PIB(T) := PIBEXCEPTHC(T) + PIBHCGNLOFF(T) + $VARINTR1 - $VARINTR;
133

134 (*@CALCUL EXPORT(T) *) 25
135 TYPHC := 7;
136 EXPORT(T) := EXPORTHC(T) + EXPHCGNLOFF(T) + (QUANTGNL(T) * PRIX(TYPHC,T));
137

138 (*@CONTROLE PIB(T) - EXPORT(T) + IMPORT(T) = CONSON(T) + INVEST(T) *) 26
139 $VARINTR := PIB(T) - EXPORT(T) + IMPORT(T) - CONSON(T) - INVEST(T);
140 CONTROL($VARINTR,EG,1,CONTINUE,2001);

```

```

*****
STATISTIQUES DE MACROGENERATION DES RELATIONS DU MODELE:
NB RELATIONS MACROGENEREES PAR CETTE EXEC : 26 NB TOTAL DE RELATIONS DU MODELE : 27
NB ENREGISTREMENTS EN ENTREE DE MACG : 141 NB ENREGISTREMENTS PRODUITS PAR MACG : 265
NB ENREGISTREMENTS PRODUITS PAR MACG POUR LE MODELE : 265
NB VARIABLES INTERNEDIAIRES INTRODUITES DANS DICO : 3
*****
LE PROGRAMME D'ENTREE DES RELATIONS NE COMPORTE PAS D'ERREUR

```

FICHIER FIMPR

```

*****
** VERSION SAMSS = VEO2          STEP = GENGRA          COMPILER LE 81.280          EXECUTE LE 82.280 A 18 HEURE(S) **
**
** MODELE COMPTADZ ALGERIE **
**
** PARAMETRES UTILISES : FIRSTGGR NOEDITCL NOTRACE NOCREDICO NOEDITGR **
**
** ZONE DYNAMIQUE DISPONIBLE AVANT RECUPERATION DU DICO : 485376 **
*****
1  CC ST      PRIX_1TC      :PRIX_1TP      :TCPHC_2      ;      1
2  CC ST      PIBEXCEPTHC_3TC :PIB_4TC      ;      2
3  CC ST      PIB_4TC      :PIB_4TP      :TCPBIB_5      ;      3
4  CC ST      PIBHCGNLOFF_3TC :PIBHC_1TC    ;      4
5  CC ST      PRIXPETINT_3TC  :COEFTRANSF_6 :PRIX_1TC     ;      5
6  CC ST      PRIXPRINT_3TC   :COEFTRANSF_6 :PRIX_1TC     ;      6
7  CC ST      PRIXPETINT_3TC  ;              ;              ;      7
8  CC ST      PIBHC_1TC      :QUANT_1TC    :PRIXPETINT_3TC ;      7
9  CC ST      PRIXPRINT_3TC  :PRIX_1TC    ;              ;      8
10 CC ST      QUANT_1TC      :QUANT_1TP    :TCQUANTHC_2  ;      8
11 CC ST      QPETPROD_3TC   :QPETCONS_3TC ;              ;      9
12 CC ST      QPETCONS_3TC   :QUANT_1TP    :TCQUANTHC_2  ;      9
13 CC ST      EXPORHCOFF_3TC  :EXPORT_7TC   ;              ;      10
14 CC ST      EXPORT_7TC     :EXPORT_7TP   :TCEXP_8      ;      10
15 CC ST      TIEXP_8        ;              ;              ;      11
16 CC ST      EXPHCGNLOFF_3TC :EXPORHC_9TC  ;              ;      12
17 CC ST      EXPORHC_9TC    :QUANT_1TC    :PRIX_1TC     ;      13
18 CC ST      IMPORT_3TC     :IMPORT_7TC   ;              ;      14
19 CC ST      IMPORT_7TC     :IMPORT_7TP   :TCIMP_8      ;      14
20 CC ST      TIIMP_8        :IMPALIM_10TC ;              ;      15
21 CC ST      IMPALIM_10TC   :IMPALIM_10TP :TCIMP_8      ;      15
22 CC ST      TIIMP_8        ;              ;              ;      16
23 CC ST      CONSOH_3TC     :CONSPUBL_3TC :CONSPRIV_3TC ;      17
24 CC ST      CONSPUBL_3TC   :CONSPUBL_3TP :TCCPUBL_6    ;      17
25 CC ST      TICPUBL_6      ;              ;              ;      18
26 CC ST      CONSPRIV_3TC   :CONSPRIV_3TP :TCCPRIV_6    ;      18
27 CC ST      TICPRIV_6      ;              ;              ;      19
28 CC ST      QUANTGNL_3TC   :PRIX_1TC     :PIBEXCEPTHC_3TC ;      20
29 CC ST      PIBHCGNLOFF_3TC :IMPORT_3TC   :EXPORHCOFF_3TC ;      20
30 CC ST      EXPHCGNLOFF_3TC :CONSOH_3TC   :INVEST_3TC    ;      21
31 CC ST      QGAZEXP_3TC    :QUANTGNL_3TC ;              ;      21
32 CC ST      RESGAP_3TC     :EXPORT_3TC   :IMPORT_3TC    ;      22
33 CC ST      QGAZPROD_3TC   :QGAZEXP_3TC  :QUANT_1TC     ;      22
34 CC ST      PIB_3TC        :PRIX_1TC     :QUANTGNL_3TC  ;      23
35 CC ST      PIBEXCEPTHC_3TC :PIBHCGNLOFF_3TC ;              ;      24
36 CC ST      EXPORT_3TC     :EXPORHCOFF_3TC :EXPHCGNLOFF_3TC ;      24
37 CC ST      QUANTGNL_3TC   :PRIX_1TC     ;              ;      25
38 CC ST      PIB_3TC        :EXPORT_3TC   :IMPORT_3TC    ;      25
39 CC ST      CONSOH_3TC     :INVEST_3TC   ;              ;      26
FIN NORMALE D'EVALUATION DU GRAPHE. NIVMAX= 8

```

```

*****
STATISTIQUES DE GENERATION DU GRAPHE D'EVALUATION DES RELATIONS DU MODELE
*****
LE PROGRAMME DE GENERATION DU GRAPHE D'EVALUATION NE COMPORTE PAS D'ERREUR

```

FICHIER FIMPR

```

*****
** VERSION SAMSS = VEO2          STEP = CONTR0          COMPILER LE 81.280          EXECUTE LE 82.280 A 18 HEURE(S) **
**
** COMPTADZ ALGERIE **
**
** PARAMETRES UTILISES NOTRACE NOTRACL NOEDITCL NOEDITGR NOCREDICO NOGENGR **
**
** TAILLE DE LA ZONE DYNAMIQUE DISPONIBLE = 516104 **
*****

```

\*\*\* CONTROLE DE COHERENCE ENTRE LE DICTIONNAIRE DES OPERANDES ET LA MATRICE D'INCIDENCE

```

... WARNING      621  AMBIGUITE POUR L'OPERANDE  PRIX_1TC
UTILISE DANS LES RELATIONS NO :      25 , 24 , 20 , 13 , 7 , 6 , 5 , 1 ;

... WARNING      621  AMBIGUITE POUR L'OPERANDE  PIB_4TC
UTILISE DANS LES RELATIONS NO :      3 , 2 ;

... WARNING      614  REDONDANCE D'INFORMATION POUR L'OPERANDE  PRIXPRINT_3TC
UTILISE DANS LES RELATIONS NO :      7 , 6 ;

... WARNING      621  AMBIGUITE POUR L'OPERANDE  QUANT_1TC
UTILISE DANS LES RELATIONS NO :      23 , 13 , 8 , 7 ;

... WARNING      621  AMBIGUITE POUR L'OPERANDE  EXPORT_7TC
UTILISE DANS LES RELATIONS NO :      11 , 10 ;

... WARNING      621  AMBIGUITE POUR L'OPERANDE  IMPORT_7TC
UTILISE DANS LES RELATIONS NO :      15 , 14 ;

... WARNING      621  AMBIGUITE POUR L'OPERANDE  IMPALIM_10TC
UTILISE DANS LES RELATIONS NO :      16 , 15 ;

... WARNING      621  AMBIGUITE POUR L'OPERANDE  CONSPUBL_3TC
UTILISE DANS LES RELATIONS NO :      18 , 17 ;

... WARNING      621  AMBIGUITE POUR L'OPERANDE  CONSPRIV_3TC
UTILISE DANS LES RELATIONS NO :      19 , 17 ;

```

\*\* DES INCOHERENCES POUVANT ETRE GRAVES ONT ETE DETECTEES; LE DICTIONNAIRE DES OPERANDES DOIT ETRE VALIDE AVANT DE CONTINUER

```

*****
** ZONE DYNAMIQUE OCCUPEE PAR LE DICTIONNAIRE = 4324 **
** ZONE DYNAMIQUE OCCUPEE PAR LE GRAPHE = 5584 **
** ZONE DYNAMIQUE DISPONIBLE APRES CHARCMENT DES CLASSES 506196 **
*****
LE PROGRAMME CONTROLE DE COHERENCE DES OPERANDES COMPORTE 9 ERREUR(S)

```



5. Après avoir vérifié qu'il ne s'agissait que d'incohérences probables mais ne pouvant pas déboucher sur des erreurs au moment de la Résolution Numérique on demande une nouvelle exécution à partir de la génération du module exécutable correspondant au modèle, qui révélera l'erreur de syntaxe.
  
6. Pour corriger cette erreur, nous devons remonter au niveau de la description des relations, modifier ce texte (toujours par update) et relancer l'analyse à partir de ce nouveau texte. Celui-ci sera accepté, on génèrera alors le graphe d'ordonnancement correspondant, on contrôlera la cohérence entre la description et l'utilisation des variables.
  
7. Après avoir à nouveau vérifié que les incohérences probables détectées ne sont pas fatales, on redemande l'automatisation à partir de la génération du module exécutable correspondant au modèle. La syntaxe étant cette fois-ci correcte, le module exécutable est généré, et l'analyse de la description des initialisations est activée, faisant apparaître 5 erreurs (variable inexistante dans dico, séparateur ', ' attendu, série chronologique incomplète).
  
8. La description des initialisations ayant été modifiée (via update) pour en tenir compte, on demande l'automatisation à partir de l'initialisation. Cette fois-ci tout est OK, la résolution numérique peut s'effectuer et les variables dont nous demandons l'état après la Résolution Numérique seront éditées. Il reste alors à analyser ces résultats et à faire vivre le modèle à partir de cet embryon.

FICHER FIMPR

```

*****
**   VERSION SAHSS = VE02           STEP = REORDO           COMPILE LE 82.146           EXECUTE LE 82.281 A 10 HEURE(S)
**
**   MODELE COMPTADZ ALGERIE
**
**   PARAMETRES UTILISES : NOTRADECL   NOTRAREORDO   NOTRLDECL   NOTRLREORDO   NOEDITCL   NOTRACE   NOCREDICO   NOEDIGR
**   NB "PAVES-RELATIONS" PAR "PAVE-MODELE" :           19
**
**   ZONE DYNAMIQUE DISPONIBLE AVANT RECUPERATION DU DICO : 440320
*****

```

```

*****
STATISTIQUES DE REORDONNEMENT DES RELATIONS MODELE
*****
LE PROGRAMME DE GENERATION DU MODELE           NE COMPORTE PAS D'ERREUR

```

FICHER FIMPR

```

*****
**   VERSION SAHSS = VE02           STEP = INITBD           COMPILE LE 81.280           EXECUTE LE 82.281 A 10 HEURE(S)
**
**   MODELE COMPTADZ ALGERIE
**
**   PARAMETRES UTILISES : FIRSTINIT   NOEDITCL   NOTRACE   NOCREDICO
**
**   ZONE DYNAMIQUE DISPONIBLE AVANT RECUPERATION DU DICO : 485376
*****

```

```

1
2   (*--COMPTADZ--DONIN--*)
3   PRIX(TYPHC,T)           (*MILLIERS DA/T*)
4   POUR 1979:              *,
5                           *,
6                           0.65,
7                           0.629,
8                           0.899,
9                           0.027,
10                          0.2424;
11
12  PIB(ELPIB,T)            (*MILLIONS DA*)
13  POUR 1979:              7840,
14                          33700,
15                          410,
16                          1540,
17                          14280,
18                          16220,
19                          5265,
20                          15650,
21                          7325,
22                          5880,
23                          12870;
24
25  EXPORT(IE,T)           (*MILLIONS DA*)
26  POUR 1979:              431,
27                          33255,
28                          261,
29                          188,
30                          1,
31                          4,
32                          4,
33                          2517,
34                          1339;
35
36  IMPORT(IE,T)           (*MILLIONS DA*)
37  POUR 1979:              5081,
38                          621,
39                          1687,
40                          7689,
41                          222,
42                          11777,
43                          4358,
44                          11975,
45                          2910;
46
47  IMPALIM(TALIM,T)       (*MILLIONS DA*)
48  POUR 1979:              336,
49                          944,
50                          458,
51                          686,
52                          423,
53                          197,
54                          530,
55                          1507;
56
57  CONSPUBL(T)            (*MILLIONS DA*)
58  POUR 1979:              17470;
59
60  CONSPRIV(T)           (*MILLIONS DA*)
61  POUR 1979:              55310;
62
63  QUANT(TYPHC,T)         (*MILLIERS DE TONNES*)
64  POUR 1979:              7519,
65                          5025,
66                          4136,
67                          49013,
68                          3235,
69                          *,
70                          9117;
71
72  (*--COMPTADZ--DONSC ET DONEX--*)
73  TCPHC(TYPHC):          0,
74                          0,
75                          0,
76                          0.15,
77                          0.15,
78                          0,
79                          0.15; (*TAUX DE CROISSANCE PRIX GHL EXPORT*)
80
81  TCPIB(ELPIB):          0.137,
82                          *,
83                          0.179,
84                          0.137,
85                          0.192,

```

```

86          0.188,
87          0.233,
88          0.233,
89          0.233,
90          0.256,
91          0.268;
92
93 COEFTRANSF:      0.816; (*PASSAGE PRIX EXTERNES AUX PRIX INTERNES*)
94
95 TCQUANTHC(TYPHC):  0.08,
96                  0.08,
97                  0.08,
98                  *,
99                  0,
100                 0,
101                 *;
102
103
104 QPETPROD(T) (*MILLIERS DE TONNES*)
105
106   POUR 1979:      56532
107   POUR 1980:      56532
108   POUR 1981:      56532
109   POUR 1982:      56532
110   POUR 1983:      56532
111   POUR 1984:      56532;
112
113 TCEXP(IE):       -0.099,
114                 *,
115                 0.134,
116                 -0.028,
117                 -0.5,
118                 -0.5,
119                 -0.5,
120                 0.036,
121                 -0.5;
122
123 TIEXP(IE):       0.12,
124                 *,
125                 0,
126                 0.103,
127                 -0.5,
128                 -0.5,
129                 -0.5,
130                 0.103,
131                 -0.5;
132
133 TCIMP(IE):       0.10,
134                 0,
135                 0.041,
136                 0.042,
137                 0.043,
138                 0.044,
139                 0.095,
140                 0.03,
141                 -0.5;
142
143 TIIMP(IE):       0.07,
144                 0.152,
145                 0.152,
146                 0.151,
147                 0.15,
148                 0.149,
149                 0.15,
150                 0.10,
151                 -0.5;
152
153 TCCPUBL:         0.118;
154
155 TICPUBL:         0.106;
156
157 TCCPRIV:         0.095;
158
159 TICPRIV:         0.103;
160
161 INVEST(T) (*MILLIONS DE DA*)
162   POUR 1979:      50700
163   POUR 1980:      69534
164   POUR 1981:      80103
165   POUR 1982:      92278
166   POUR 1983:      106304
167   POUR 1984:      122463;
168 PRIXPRINT(T)
169   POUR 1979: *;
170 ENDINIT

```

\*\*\*\*\* LISTE DES OPERANDES DONT ON ATTEND ENCORE L'INITIALISATION \*\*\*\*\*

```

*****
STATISTIQUES D'INITIALISATION DE LA BASE DES OPERANDES DU MODELE
NB OPERANDES DU MODELE DECLARES A INITIALISER :      23
NB OPERANDES DEJA INITIALISES :                      23
NB OPERANDES INITIALISES AU COURS DE LA PRESENTE EXECUTION : 23
ESPACE MEMOIRE MIN POUR INIT BASE DES OPERANDES :    88 + TLDS +TLFB + TAILLE DU LOAD MODULE +TAILLE DICO
*****
LE PROGRAMME D'INITIALISATION DE LA BASE DU MODELE NE COMPORTE PAS D'ERREUR.

```

FICHIER FIMPR

```

*****
**      VERSION SAMSS = VEO2          STEP = RESNUM          COMPILER LE 82.281          EXECUTE LE 82.281 A 10 HEURE(S)
**      MODELE COMPTADZ ALGERIE
**      PARAMETRES UTILISES : NOTRACE   NOIMPOPEXEC   DEBSIM   PINSIH
**      ZONE DYNAMIQUE DISPONIBLE AVANT RECUPERATION DU DICO : 481280
*****

```

```

*****
STATISTIQUES D'EXECUTION DES RELATIONS DU MODELE :
*****
LE PROGRAMME DE RESOLUTION NUMERIQUE DU MODELE NE COMPORTE PAS D'ERREUR

```

FICHIER FIMPR

```

*****
**      VERSION SAMSS = VEO2          STEP = INTBOP          COMPILER LE 82.133          EXECUTE LE 82.146 A 03 HEURE(S)
**      COMPTADZ ALGERIE
**      PARAMETRES UTILISES NOTRACE   NOEDITCL          NOCREDICO SEQUENTIEL
**      TAILLE DE LA ZONE DYNAMIQUE DISPONIBLE = 438280
*****
1 PIB(T);
2 EXPORT(T) UNITE MILLIONS_DINARS;
3 IMPORT(T) UNITE MILLIONS_DINARS;
4 CONSONM(T) UNITE MILLIONS_DINARS;
5 INVEST(T) UNITE MILLIONS_DINARS;
6 RESCAP(T) COMMENT POURCENTAGE DE COUVERTURE DES IMPORTATIONS;
7 QGAZPROD(T) UNITE MILLIONS_M3 COMMENT PRODUCTION DE GAZ;
8 QPETCONS(T) UNITE MILLIERS_DE_TONNES COMMENT CONSOMMATION INTERIEURE DE PETROLE;
9 QUANTGML(T) UNITE MILLIERS_DE_TONNES COMMENT EXPORTATION DE GNL;
10 QUANT(TYPHC,T) UNITE MILLIERS_DE_TONNES COMMENT PRODUCTION D HYDROCARBURES;
11 ENDANAL
*****
**      ZONE DYNAMIQUE OCCUPEE PAR LE DICTIONNAIRE = 4320
**      ZONE DYNAMIQUE DISPONIBLE APRES CHARGEMENT DES CLASSES 433960
**      OCCUPATION DES TABLEAUX UTILISES PAR L'EDITEUR
**      TACTION (ACTIONS D'EDITION) 15 / 500
**      TLIGNE (LIGNES A EDITER) 7 / 200
**      TREMPLE (PARAMETRES DE REMPLISSAGE DES LIGNES DE RESULTATS) 2 / 100
**      TINDICE (PARAMETRES RELATIFS AUX INDICES) 2 / 50
*****
LE PROGRAMME D'ANALYSE DES RESULTATS DU MODELE NE COMPORTE PAS D'ERREUR

```

FICHIER OUTPUT

```

*****
**      VERSION SAMSS = VEO2          STEP = INTBOP          COMPILER LE 82.133          EXECUTE LE 82.146 A 03 HEURE(S)
**      COMPTADZ ALGERIE
**      PARAMETRES UTILISES NOTRACE   NOEDITCL          NOCREDICO SEQUENTIEL
**      TAILLE DE LA ZONE DYNAMIQUE DISPONIBLE = 438280
*****
\033c1\003

```

PIB(T)	UNITE:
T= 1980	:132780.0
T= 1981	:164790.0
T= 1982	:206590.0
T= 1983	:261330.0
T= 1984	:333050.0

EXPORT(T)	UNITE: MILLIONS_DINARS
T= 1980	: 26716.0
T= 1981	: 39377.0
T= 1982	: 58250.0
T= 1983	: 84707.0
T= 1984	:125260.0

IMPORT(T)	UNITE: MILLIONS_DINARS
T= 1980 :	51118.0
T= 1981 :	60247.0
T= 1982 :	71069.0
T= 1983 :	83908.0
T= 1984 :	99153.0

CONSONM(T)	UNITE: MILLIONS_DINARS
T= 1980 :	87645.0
T= 1981 :	105550.0
T= 1982 :	127130.0
T= 1983 :	153140.0
T= 1984 :	184480.0

INVEST(T)	UNITE: MILLIONS_DINARS
T= 1980 :	69534.0
T= 1981 :	80103.0
T= 1982 :	92278.0
T= 1983 :	106300.0
T= 1984 :	122460.0

RESGAP(T)	UNITE:
POURCENTAGE DE COUVERTURE DES IMPORTATIONS	
T= 1980 :	-0.4774
T= 1981 :	-0.3464
T= 1982 :	-0.1804
T= 1983 :	0.0224
T= 1984 :	0.2633

QGAZPROD(T)	UNITE: MILLIONS_M3
PRODUCTION DE GAZ	
T= 1980 :	-96198.0
T= 1981 :	-43145.0
T= 1982 :	24991.0
T= 1983 :	110060.0
T= 1984 :	213930.0

QPETCONS(T)	UNITE: MILLIERS_DE_TONNES
CONSUMMATION INTERIEURE DE PETROLE	
T= 1980 :	8120.5
T= 1981 :	8770.2
T= 1982 :	9471.8
T= 1983 :	10230.0
T= 1984 :	11048.0

QUANTGNL(T)	UNITE: MILLIERS_DE_TONNES
EXPORTATION DE GNL	
T= 1980	-55432.0
T= 1981	-26415.0
T= 1982	10892.0
T= 1983	57505.0
T= 1984	114450.0

QUANT(TYPHC,T)	UNITE: MILLIERS_DE_TONNES
PRODUCTION D'HYDROCARBURES	

	TYPHC = 1	TYPHC = 2	TYPHC = 3	TYPHC = 4	TYPHC = 5	TYPHC = 6	TYPHC = 7
T=1980	8120.5	5427.0	4466.9	48411.0	3235.0	N V	N V
T=1981	8770.2	5861.2	4824.2	47762.0	3235.0	N V	N V
T=1982	9471.8	6330.1	5210.2	47060.0	3235.0	N V	N V
T=1983	10230.0	6836.5	5627.0	46302.0	3235.0	N V	N V
T=1984	11048.0	7383.4	6077.1	45484.0	3235.0	N V	N V

```

*****
** ZONE DYNAMIQUE OCCUPEE PAR LE DICTIONNAIRE = 4320
** ZONE DYNAMIQUE DISPONIBLE APRES CHARGEMENT DES CLASSES 433960
** OCCUPATION DES TABLEAUX UTILISES PAR L'EDITEUR
** TACTION (ACTIONS D'EDITION) 15 / 500
** TLIGNE (LIGNES A EDITER) 7 / 200
** TREMPLE (PARAMETRES DE REMPLISSAGE DES LIGNES DE RESULTATS) 2 / 100
** TINDICE (PARAMETRES RELATIFS AUX INDICES) 2 / 50
*****

```

### IV.3. INTERPRÉTATION DES RÉSULTATS - INTÉRÊTS DE LA SIMULATION

Une première simulation de l'"Equilibre des comptes nationaux algériens" vient d'être tentée. Elle a permis de montrer clairement que le logiciel "répondait" au problème qui lui était posé.

Or la formulation du problème était volontairement limitée aux exportations de G.N.L., ce qui n'avait pour nous que valeur de test. Nous savons que la réalité des échanges algériens est infiniment plus complexe. La possibilité qu'offre le logiciel S.A.M.S.S., de prendre en compte cette complexité au travers de simulations multiples et variées permettra d'appréhender le système dans toute sa globalité et sa richesse.

En effet, la lecture des premiers résultats, même par un oeil non averti, permettait de formuler la conclusion partielle suivante : si le choix des hypothèses retenues est judicieux, alors l'Algérie n'avait pas besoin d'exporter de G.N.L. en 1979-80 et 81. On nous fera alors remarquer qu'il n'était peut-être pas nécessaire de recourir à de tels moyens pour s'en persuader, et que, plus généralement, la simulation d'un système pour lequel on a su construire un modèle n'apporte rien de neuf (sous-entendu : si on a pu construire le modèle, c'est que le phénomène est bien connu). Avec H.A. SIMON [45], nous nous inscrirons en faux contre cette affirmation. En effet, si la connaissance de chacun des liens existants entre les composants caractéristiques du phénomène étudié est nécessaire, elle n'implique pas pour autant la connaissance globale du phénomène, et seule une série de simulations permettra de tester les hypothèses faites.

Ainsi, dans l'exemple algérien choisi, plusieurs scénarios et sous-scénarios sont envisageables. Par exemple, sans remettre en cause la structure du modèle (scénario de référence), tout un corps de questions importantes nous viennent à l'esprit :

- l'évolution des prix du G.N.L. retenue pour le scénario de référence est-elle plausible ?

- les taux de croissance et d'inflation choisis sont-ils le reflet fidèle de la réalité du marché mondial des hydrocarbures ?

... etc

En jouant uniquement sur les hypothèses, il semble donc possible de contrôler la sensibilité du modèle. A partir du test 1, dit "scénario tendanciel de base" (on a essayé de prolonger les tendances passées et actuelles dans l'avenir), nous allons mettre en oeuvre divers scénarios dits d'encadrement, et c'est ce premier ensemble qui nous permettra d'isoler les "paramètres" du modèle.

Ces paramètres ou variables sensibles, une fois localisés, les économistes pourront abandonner la collecte d'informations non essentielles, pour reporter leurs efforts sur les problèmes structurels et élaborer une nouvelle série de scénarios. Se limiter à l'étude des exportations de G.N.L., pour autant que cela soit satisfaisant pour l'esprit, ne reflète pas la réalité des échanges de l'Algérie avec le monde. Il serait donc intéressant et nécessaire de prendre en compte la structure des exportations en général. Si le secteur des hydrocarbures (Pétrole Brut, Produits Raffinés, G.N.L.) y est prépondérant, les minerais (Phosphates, Fer ...) et certains produits finis industriels ou agricoles ne doivent pas pour autant être négligés. Le logiciel S.A.M.S.S., par ses facilités de mise en oeuvre, permet d'effectuer les transformations qu'imposent ce nouveau regard et si l'utilisation de l'outil informatique était contestable pour l'exemple de base, il en va tout autrement dès que le niveau de désagrégation du système étudié s'accroît. En effet, si l'esprit est capable d'interrelier une dizaine de variables (lorsque les interrelations sont simples), c'est quasiment impossible dès que le seuil de la cinquantaine est atteint.

La simulation apportera alors de la rigueur au discours, et permettra de mesurer les comportements dynamiques difficilement maîtrisables comme ceux induits par les "fonctions retards".



#### IV.4. MISE EN OEUVRE TECHNIQUE DU LOGICIEL S.A.M.S.S.

On ne s'attendra pas à trouver ici les détails de la mise en oeuvre du logiciel S.A.M.S.S. On pourra à ce sujet se reporter au manuel qui y est consacré [7].

##### IV.4.1. GENERALITES

L'exécution complète d'un travail soumis au logiciel S.A.M.S.S. comporte actuellement neuf étapes, regroupées en deux grandes phases :

- Description du modèle.
- Résolution Numérique du modèle.

L'intégration de ces neuf étapes, rendue possible grâce aux langages de commandes disponibles sur IBM ou ICL permet :

- d'exécuter l'ensemble des étapes en un seul passage (SAMSS=Package),
- de démarrer la session SAMSS à partir de l'une quelconque des étapes,
- d'arrêter son déroulement après exécution de l'une quelconque des étapes,
- d'arrêter son déroulement si une des étapes comporte des erreurs.

Chacune des étapes du "package" SAMSS correspond à l'exécution d'un module. Ces neuf modules sont les suivants :

CREDIC	création du Dictionnaire du Modèle
FMATED	formatage de la Base des Opérandes du Modèle
MACROG	macro-génération du texte Pascal des équations du Modèle
GENGRA	génération du graphe d'évaluation des équations du Modèle
CONTRC	contrôle de la cohérence définition/utilisation des variables du Modèle
REORDO	génération du module exécutable correspondant au Modèle
INITBD	initialisation de la Base des Opérandes du Modèle
RESNUM	exécution du Modèle
INTBOP	édition des résultats du Modèle

Les premiers résultats de l'utilisation du logiciel S.A.M.S.S. sont en général des diagnostics d'erreur (sous forme de messages numérotés) qui révèlent des problèmes de syntaxe (il manque un mot-clé, une caractéristique, ...). Ce type d'erreur est facile à corriger.

Parmi les erreurs qui peuvent apparaître par la suite, certaines sont imputables à l'utilisateur comme l'ensemble des incohérences probables détectées en rapprochant la définition et l'utilisation des variables du modèle.

D'autres ne sont pas à proprement parler des erreurs, même si elles provoquent l'abandon immédiat du travail. Ce sont plutôt des messages indiquant que certaines zones de travail sont saturées (zone dynamique disponible trop petite, nombre maximum de variables utilisées dans une relation supérieure à la limite ...), que l'espace physique réservé pour stocker un fichier est insuffisant (code condition), ...

Lors des premières exécutions complètes (résolution numérique du modèle), d'autres erreurs peuvent apparaître :

- l'absence de résolution et de résultats :

- . il subsiste des erreurs, détectées à la compilation des textes utilisateurs, des incohérences entre définition et utilisation des variables ... et la résolution numérique n'a pas eu lieu ;
- . les temps d'exécutions sont trop courts ;
- . une zone de travail est saturée ;
- . division par zéro dans l'une des relations du modèle ;
- . texte des requêtes d'interrogation vide ... etc.

- Les résultats obtenus ne correspondent pas à ceux attendus

En principe, le logiciel S.A.M.S.S. n'exécute que ce qu'on lui demande d'exécuter et si les résultats sont incorrects, dans la quasi-totalité des cas c'est que la programmation (textes sources) n'est pas la bonne. Avant d'accuser le logiciel, ou de faire appel à l'informaticien concerné, il convient donc de s'assurer que :

- . la résolution numérique s'est effectuée à partir de la "bonne" Base des Opérandes du Modèle ;
- . il ne subsiste pas de messages d'erreurs ;
- . la formulation des relations qui décrivent le comportement du phénomène modélisé correspond aux objectifs du concepteur.

IV.4.2. PRINCIPES DE FONCTIONNEMENT DE LA PROCEDURE S.A.M.S.S.

La procédure S.A.M.S.S. (écrite dans le langage de commandes de la machine-hôte du logiciel) contient l'ensemble des ordres nécessaires à l'exécution d'une session complète d'automatisation d'un modèle.

Le déroulement des étapes de cette procédure se fait de façon parfaitement linéaire et suivant l'ordre mentionné ci-dessus.

Pour qu'une étape puisse démarrer, l'exécution de l'étape précédente doit être satisfaisante. Ainsi, tant que le texte de Description des Objets du Modèle comporte des anomalies syntaxiques, le formatage de la Base des Opérandes du Modèle ne sera pas abordé. L'exécution du "package" sera abandonnée. Il en est de même pour chacune des neuf étapes.

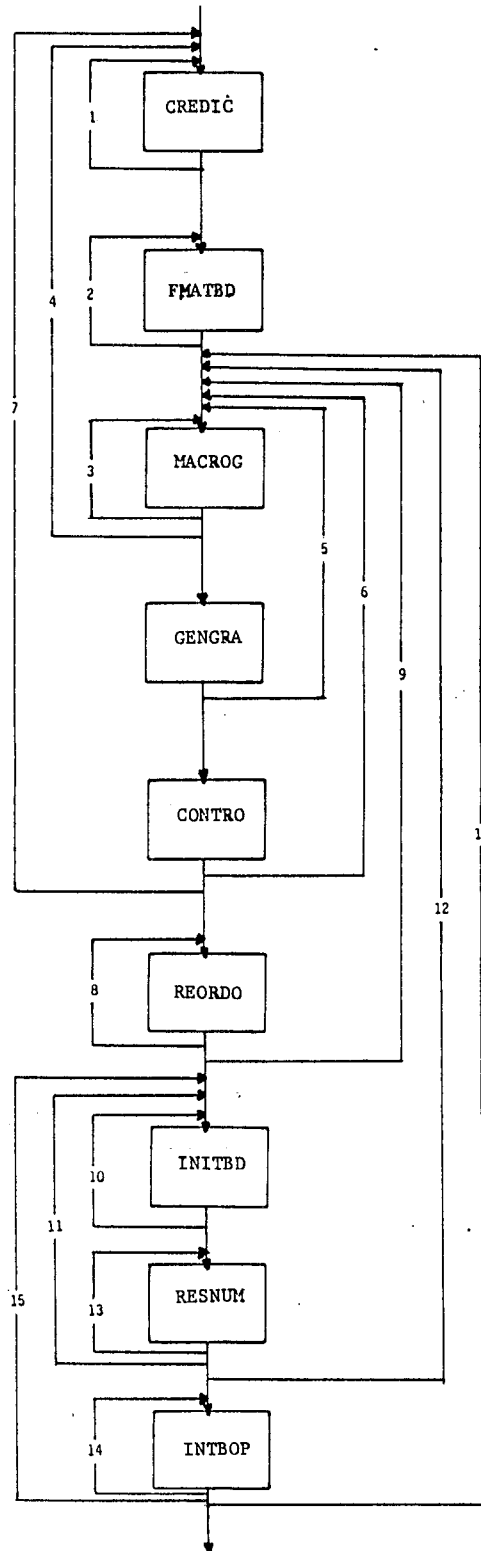
La procédure S.A.M.S.S. exige de l'utilisateur qu'il précise un certain nombre de paramètres (identification des fichiers qui contiennent les textes nécessaires à l'automatisation du Modèle, dates de début et fin de simulation ...) dont certains peuvent affecter le déroulement de la procédure. C'est par exemple le cas du paramètre qui permet de préciser l'étape de démarrage. En effet, par défaut, toute

session S.A.M.S.S. est réputée commencer par une Description du Modèle à automatiser. Or, la mise en oeuvre la plus fréquente de la procédure S.A.M.S.S. est celle qui, à partir d'un modèle stable et cohérent, vise à tester différents scénarios portant sur les données.

Chacun des éléments essentiels du Modèle (dans sa forme automatisée) est conservé. Le Dictionnaire du Modèle, le texte-Pivot des équations, la Base des Opérandes ... sont mémorisés, ce qui autorisera une mise en oeuvre simplifiée de la procédure S.A.M.S.S.

Il convient enfin de signaler ici, l'existence d'étapes secondaires (et après dans le cas particulier du module REOR) l'exécution des modules. Nous parlerons d'étapes secondaires parce qu'elles ne correspondent pas à des fonctions conçues dans le cadre du logiciel. Elles sont plutôt du ressort du langage de commandes (effacement de fichiers, compilations ...) ou d'utilitaires d'intérêt général (fusion, mise à jour des textes de référence de l'utilisateur). Elles sont cependant nécessaires pour protéger l'utilisateur contre ses fausses manipulations.

IV.4.3. SCHEMA DES POSSIBILITES DE REPRISES EN COURS D'AUTO-  
MATISATION D'UN MODELE VIA S.A.M.S.S.



CONCLUSION

La simulation des phénomènes socio-économiques est un mal rendu nécessaire par la complexité, l'ampleur et la nature des modèles élaborés. Elle favorise l'expérimentation et l'étude du comportement du modèle (représentation simplifiée du système étudié), autorisant ainsi la formulation des conclusions valables quant aux propriétés du système réel.

La simulation, qui a fait son apparition dans le domaine des Sciences Sociales avec la publication des travaux de FORESTER [2], a provoqué le développement de logiciels dits de "simulation" dont le but était de faciliter l'écriture et la résolution numérique de modèles. Ces logiciels se distribuent en trois grandes classes :

- logiciels de simulation discrète qui travaillent au niveau évènementiel,
- logiciels de simulation continue qui prennent en compte des équations différentielles ou des équations aux dérivées partielles (notion de flux),
- logiciels de simulation combinée qui associent les deux types précédents.

Le logiciel S.A.M.S.S. appartient à la première. Pour conclure, nous voudrions évoquer la place qu'il occupe, tant du point de vue du concepteur que de celui de l'utilisateur, avant d'envisager ses perspectives d'évolution.

Une étude critique et comparative de S.A.M.S.S. et des logiciels construits selon l'approche par événements comme SIMSCRIPT [46], GASP [47], SLAM [28], GPSS [48] ou SIMULA [49] d'une part, des logiciels dits "de simulation continue" comme CSMPTIII [50], ACSL [51], DYNAMO [52] ou MODULECO [40] mériterait certes quelque attention. Ce n'est pas le but poursuivi ici et nous nous contenterons de rappeler quelques uns des apports originaux de S.A.M.S.S. :

- capacités descriptives étendues,
- rapidité d'exécution,
- manipulation de variables représentées par des matrices à plusieurs dimensions,

- transportabilité,
- syntaxe aisément maîtrisable ...

Ces caractéristiques en font un outil qui suscite un intérêt certain chez les économistes modélisateurs puisque l'une des conclusions principales des journées d'études organisées par le CRISS les 25 et 26 mai 1982 autour du logiciel S.A.M.S.S. était d'en organiser la diffusion le plus rapidement possible (proposition émanant du directeur du Laboratoire d'Informatique en Sciences Humaines).

Mais laissons la parole aux utilisateurs de SAMSS/MEDEE3 : "L'objectif qui consistait à produire un logiciel utilisable par les économistes a été atteint, et les résultats de sa mise en oeuvre ont été très positifs pour l'exploitation de MEDEE3, tant au niveau européen qu'au niveau de chacun des membres de la Communauté. Cette mise en oeuvre prend du temps, mais certainement moins que pour d'autres logiciels analogues. Le logiciel signale toutes les erreurs, de façon très "pédagogique". Il exige - ce qui est normal - une préparation très minutieuse, pour ne pas perdre de temps en recherches d'incohérence et en corrections au stade de l'exécution. L'assistance d'un informaticien est nécessaire, mais seulement lorsque des difficultés importantes apparaissent lors de l'utilisation du logiciel, ce qui est rare".

Après presque 18 mois d'utilisation intensive du logiciel, un bilan objectif fait apparaître les quelques remarques suivantes :

Phase I : Formulation en "langage" SAMSS du modèle. Cette phase est caractérisée par les difficultés nées des différences d'approche entre économistes et informaticiens. La rigueur imposée par le passage en machine exige des informaticiens une formalisation de tous les aspects implicites du modèle, qu'il n'est pas facile de faire admettre aux économistes.

Exemple : Préviation des cas limites (divisions par zéro, utilisation dans des opérations d'une non valeur représentée par l'"INFINI" ...).

Analyse des conditions non satisfaites.

...



Phase II : Définition cohérente du modèle. Un bon point à l'actif du logiciel : la définition cohérente en "langage" SAMSS du modèle a été prise en charge par les économistes eux-mêmes, avec l'assistance des informaticiens, et non l'inverse.

Il subsiste cependant quelques problèmes :

- . Définition des équations du modèle : Si le langage d'expression des équations est assez rapidement maîtrisable, si le repérage d'erreurs telles que orthographe invalide des variables, boucles de longueur 1 ... s'effectue sans difficulté, il n'en est pas de même de la localisation des systèmes d'équations simultanées, "cauchemard de l'utilisateur". Celle-ci nécessite le plus souvent l'intervention d'un informaticien pour analyser le graphe d'ordonnancement produit.
  
- . Cohérence entre définition et utilisation des variables du modèle : présentées comme un outil d'aide à la modélisation, les sorties du module qui assure ces contrôles sont délicates à exploiter. En effet, d'une part le nombre d'incohérences détectées est important et d'autre part une proportion non négligeable de celles-ci ne sont que potentielles. Les vérifications longues et fastidieuses qu'elles supposent, lorsque le modèle est complexe, poussent l'utilisateur à les négliger, prenant le risque d'un "plantage" au cours de la Résolution Numérique.
  
- . Génération automatique du module exécutable correspondant au modèle : deux catégories de problèmes se posent à l'utilisateur à ce niveau. L'une relève de l'utilisateur lui-même puisqu'il s'agit des erreurs de formulation en "langage" SAMSS des équations du modèle. En effet ce n'est qu'au moment de la compilation du programme Pascal généré à partir du texte utilisateur que les erreurs de syntaxe sont détectées. Ces erreurs ne soulèvent aucune difficulté technique particulière, mais sont génératrices de pertes de temps non négligeables et donc entraînent des coûts importants. La seconde relève plutôt du domaine des infor-

maticiens : découpage des relations en procédures (problème lié aux contraintes d'utilisation du langage Pascal), interprétation des messages d'erreurs émis par le système d'exploitation hôte du logiciel, etc.

Phase III : Simulation du modèle. Les problèmes relatifs à la maîtrise et à la mise en oeuvre de cette phase sont assez simples.

• Au niveau des initialisations des variables du modèle :

Le respect de la syntaxe imposée par SAMSS ne pose pas de problème particulier. La présence de l'outil constitué par la liste des variables à initialiser et qui ne l'ont pas encore été est précieuse pour éviter des surprises désagréables au moment de la Résolution Numérique.

Une contrainte cependant qu'il serait agréable de lever : dans l'état actuel du logiciel, une série d'initialisations partielles aboutissant à l'initialisation complète n'est pas possible, il est nécessaire de faire le travail en une seule passe.

- Au niveau de la Résolution Numérique du modèle : C'est lors de cette étape que se révèlent les occurrences d'opérations inexécutables dont nous avons parlé plus haut (division par zéro, multiplication par  $\infty$ , ...). Si dans un premier temps, l'intervention d'un informaticien est indispensable (analyse du "dump" en hexadécimal pour repérer les relations concernées), l'utilisateur peut se familiariser avec ce travail et le faire lui-même. Cela représente cependant une source de pertes de temps importantes. La présence de l'outil constitué par la liste des "warning" (petits drapeaux distribués dans le modèle pour en contrôler la pertinence) est intéressante dans la mesure où elle permet de mieux "caler" le modèle dans les simulations à venir.

Dans un autre ordre d'idées, la confrontation d'expériences, les discussions concepteurs/utilisateurs du logiciel, font apparaître la nécessité d'en améliorer certains aspects, d'y introduire de nouvelles

fonctions, ce de manière à en rendre l'utilisation plus agréable d'une part, et d'en élargir le champ d'application à des domaines nouveaux relevant des Sciences Sociales, autres que ceux de la demande d'énergie à long terme d'autre part. Il y a là une "universélisation du logiciel" qui est non seulement tentatrice, mais possible et nécessaire.

Les plus caractéristiques pourraient être :

- Résolution des Systèmes d'Equations Simultanées : Pour les utilisateurs actuels de SAMSS, le problème n'est pas de disposer d'un outil informatique capable de résoudre ces systèmes, mais de pouvoir dans un premier temps identifier les équations dont le module de Génération du Graphe n'a pas pu définir le niveau d'évaluation, et dans un second temps, repérer les seules équations qui constituent ces systèmes, afin de "casser" ces "boucles".

Dans le domaine des Sciences Sociales, est-il pertinent en effet de représenter les "fedd-backs" sous forme de processus simultanés ? Il semble plus correct, le plus souvent, de jouer sur la variable temps en adoptant un pas de simulation mieux adapté au phénomène considéré, ou lorsque c'est possible, de recourir à des variables virtuelles reflétant une situation idéale.

Il serait peut être intéressant de pouvoir définir la liste des dates pour lesquelles on souhaite éditer les réalisations des variables, de façon déconnectée de la liste des dates de simulation.

En tout état de cause, si SAMSS était doté de la possibilité de résoudre les systèmes d'équations simultanées, cette possibilité devrait être optionnelle, car de nombreux utilisateurs continueront à rechercher ces systèmes, non pour les résoudre, mais pour les supprimer.

- Erreurs de Syntaxe dans la Formulation des Equations : Dans la version actuelle, les erreurs de syntaxe contenues dans la formulation (en langage SAMSS) des Equations du modèle ne sont découvertes qu'au moment de la génération du module correspondant au modèle. La solution à ce problème pourrait être fournie, dans le cadre des recherches effectuées

au CRISS sur les outils de "Génie Logiciel" liés au langage PASCAL. Il s'agirait d'ajouter un "syntax-checker" au stade de la macrogénération, ce qui permettrait de repérer et de corriger les erreurs de syntaxe à ce moment là.

Un tel outil, s'il est coûteux à l'utilisation (30 % du temps de compilation du programme PASCAL correspondant au modèle), permettrait cependant de reporter au niveau du texte utilisateur les erreurs qui lui sont imputables et éviterait une génération de graphe, un contrôle de cohérence, un réordonnement et une génération du module correspondant au modèle, à partir d'une description des relations non validée. En outre, sa mise en oeuvre pourrait être optionnelle :

- l'amélioration de la syntaxe d'initialisation de la Base des Opérandes (facteur répétitif, initialisation partielle de la Base, d'une variable ...etc),
- l'affinement des contrôles de cohérence entre définition et utilisation des variables du modèle,
- le "Stop and Go" : possibilité de prendre en compte des modifications de scénario de données selon les tranches de temps (distinction entre date de référence et date de simulation),
- le "glissement" du modèle par rapport au temps (transformer une date de simulation en date de référence), ...



BIBLIOGRAPHIE

- [1] VON BERTALANFFY L. - *Théorie générale des systèmes* - Paris - Dunod, 1973
- [2] FORRESTER J.W. - *Principles of Systems* - Cambridge - Wright-Allen Press, 1974
- [3] BAREL Y. - *Prospective et Analyse de Système* - Paris- La Documentation Française (coll. Travaux et Recherches de Prospective, n° 11), 1971
- [4] ROY B. - *Modélisation et Maîtrise des Systèmes* - Paris - Editions Hommes et Techniques (Actes du Congrès AFCET), 1977
- [5] RIGAL J.L. - *Scientisme, Analyse de Systèmes, Sciences et Métasciences* - Paris - Editions Hommes et Techniques (Actes du Congrès AFCET), 1977
- [6] BRUNET C.  
DUMARCHEZ P. - *SAMSS, Prototype* - Note de fin de contrat - Septembre 1978
- [7] BRUNET C.  
DUMARCHEZ P. - *Manuel Didactique et de Référence du Logiciel SAMSS* - 1981  
- *Manuel de Mise en Oeuvre du Logiciel SAMSS* - 1981  
- *Manuel d'Exploitation du logiciel SAMSS* - 1981
- [8] STRANCH P.G. - *Modeling in the Social Sciences : an approach to good theory and good policy* - (dans SIMULATION Vol. 26) - 1976
- [9] RECHENMAN F. - *Analyse et Modélisation descendante dans les systèmes socio-économiques* - (Thèse de Docteur-Ingénieur) - Grenoble - ENSIMAG, 1976
- [10] BAREL Y. - *L'idée de système dans les Sciences Sociales* - Paris - Compte-rendu journée AFCET, 1976
- [11] RECHENMAN F.  
RIVERA E.  
UVIETTA P. - *A modeling Language for the Analysis of Regional Systems* - (International Symposium "Simulation 75") - Zürich - Acta Press, 1975
- [12] GRANDJEAN E. - *Algorithmes de construction et de mise à jour d'un dictionnaire* - Grenoble - RR n° 28, 1976
- [13] LUCAS M.  
SCHOLL P.C.  
VOIRON J. - *Apprentissage et utilisation du traitement séquentiel pour la construction de programmes* - Grenoble - RR n° 74, 1977

- [ 14 ] WIRTH N. - *Algorithms + Data Structures = Programs* - Englewood Cliffs - Prentice-Hall - 1976
- [ 15 ] SCHOLL P.C. - *Introduction à la récursivité et aux arbres* - Grenoble - Support de cours I.P.G., 1977
- [ 16 ] MESAROVIC M.  
MACKO D.  
TAKAHARA Y. - *Theory of Multileveled Hierarchical Systems* - Academic Press, 1970
- [ 17 ] RIVERA PORTO F.E. - *Simulation et Analyse Interactive des Systèmes Dynamiques en Sciences Sociales* - Grenoble - Thèse 3ème Cycle, 1977
- [ 18 ] STENDEL J. - *Les systèmes informatiques de programmation économique* - Paris - Dunod, 1968
- [ 19 ] DIAS D'ALMEIDA F. - *Etude Numérique de la stabilité dynamique des modèles macro-économiques* - Grenoble - Thèse 3ème Cycle, 1980
- [ 20 ] - *Simulation of systems '79* - (Proceedings of the Congress) - Amsterdam - North-Holland Pub. company, 1980
- [ 21 ] ROSSIER E. - *Sur les fondements du calcul qualitatif en Sciences Economiques* -
- [ 22 ] DUFOUR J.  
OUDET B. - *Approche Informationnelle de l'analyse structurale des systèmes économiques* - Lyon - Colloque Structures Economiques et Econométrie, 1977
- [ 23 ] NEPOMIASTCHY P. - *Méthode de Pénalisation et Applications : Résolution et Optimisation de modèles macro-économiques* - Grenoble - Thèse de Doctorat d'Etat Es-Sciences, 1979
- [ 24 ] RACHDI M. - *Quelques algorithmes de résolution de modèles macro-économiques* - Paris - Thèse de 3ème Cycle, 1979
- [ 25 ] NEPOMIASTCHY P.  
RAVELLI A.  
RECHENMANN F. - *An automatic method to get an econometric model in quasi-triangular form* - Paris - RR n° 313 du Laboria, 1978
- [ 26 ] BRESSY Y.  
MERMET J.  
UVIETTA P. - *Langage de description de l'Organisation et de la Dynamique des Systèmes Socio-Economiques et Environnementaux* - Grenoble - RR n° 91



- [ 27] BRESSY Y.  
MERMET J.  
UVIETTA P. - *Mécanismes de Simulation Discrète et Continue liés aux modèles de Systèmes Dynamiques* - Grenoble - RR n° 206, 1980
- [ 28] PRITSKER A.A.  
PEDGEN C.D. - *Introduction to Simulation and SLAM* - West Lafayette - Systems Publishing Corporation, 1979
- [ 29] GABAY D.  
NEPOMIASTCHY P.  
RACHDI M.  
RAVELLI A. - *Etude, Résolution et Optimisation de Modèles Macroéconomiques* - Paris - RR n° 312 du Laboria, 1978
- [ 30] NATIONAL BUREAU OF  
ECONOMIC RESEARCH - *Troll : an introduction and demonstration*, 1972
- [ 31] MAAREK G. - *Modèles macroéconomiques et programmation linéaire* - Article paru dans la Revue Economique n° 5, 1980
- [ 32] PROPOI A.  
ZIMIN L. - *Dynamic Linear programming models of energy, resource, and economic-development systems* - Luxemburg (Autriche) - RR-81-14, 1981
- [ 33] HELGSAUN K. - *DISCO : a SIMULA-based language for continuous combined and discrete simulation* - Otlawa - Tuncer I. Oren, 1980
- [ 34] WIMMERT R.J. - *GAP : a GPSS/Fortran package* - University of South Florida - Annual Simulation Symposium
- [ 35] MATHEW R.J.  
KULIKOWSKI C.A.  
KAPLAN K.R. - *A multilevel representation for knowledge acquisition in medical consultation systems* - NIH - RR-643, 1977
- [ 36] SARGENT R.G. - *Validation of simulation models* - Syracuse, 1979
- [ 37] BABOCK P.D.  
STUTZMAN L.F.  
KROUP T.G. - *GEMS : a mathematical simulation language* - University of Connecticut, 1981
- [ 38] NURMI H. - *Reflection on the nature and use of systems models in the social sciences* - Turku (Finlande) - article paru dans la revue Modern Trends in Cybernetics and Systems, 1975
- [ 39] MALGRANGE P. - *Méthodes mathématiques de la modélisation macro-économique* - Paris - IRIA, 1979

- [ 40] OUDET B. - *Le projet MODULECO* - Paris - IRIA, 1980  
NEPOMIASTCHY P.
- [ 41] SEFSAF B. - *Conception et Réalisation du système de  
gestion de séries chronologiques du  
logiciel MODULECO* - Grenoble - Thèse de  
3ème Cycle, 1981
- [ 42] - *Analysis of Trade between the European  
Community and the arab league Countries* -  
Luxembourg - Eurostat, 1980
- [ 43] MINISTERE DE LA - *Annuaire Statistique de l'Algérie* -  
PLANIFICATION ET DE Alger - CNRES, 1980  
L'AMENAGEMENT DU  
TERRITOIRE
- [ 44] DIRECTION DES STATIS- - *L'Algérie en quelques chiffres* - Alger -  
TIQUES ET DE LA COMP- CNRES, 1980  
TABILITE NATIONALE
- [ 45] SIMON H.A. - *La Science des Systèmes* - Paris - L'EPI  
Editions, 1974
- [ 46] DELFOSSE C.M. - *Simulation continue et simulation dis-  
crète en SIMPSRIPT* - Arlington - CACI  
Inc., 1976
- [ 47] CELLIER F.E. - *Manuel des utilisateurs GASPV* - Institut  
d'Automatique et d'Electronique Indus-  
trielle - Suisse, 1978
- [ 48] BOBILIER P.A. - *Simulation avec GPSS et GPSSV* - Englewood  
KAKAN B.C. Cliffs - Prentice Hall, 1976  
PROBST A.R.
- [ 49] DAHL O.J. - *SIMULA 67 common base language* - Oslo -  
MYHRHAUG B. NCC, 1970  
NYGAARD K.
- [ 50] SPECKHART F.H. - *Un guide pour l'utilisation de CSMP* -  
GREEN (W.L.) Englewood Cliffs - Prentice Hall, 1976
- [ 51] PUGH A.L. - *DYNAMO II - User's manual* - M.I.T. Press,  
1970
- [ 52] WIRTH J. - *PASCAL USER MANUAL AND REPORT* - Springer  
Verlag, 1978



ANNEXES







