



**HAL**  
open science

# Conception et réalisation du système de gestion de séries chronologiques du logiciel MODULECO

Belkacem Sefsaf

► **To cite this version:**

Belkacem Sefsaf. Conception et réalisation du système de gestion de séries chronologiques du logiciel MODULECO. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1981. Français. NNT: . tel-00295259

**HAL Id: tel-00295259**

**<https://theses.hal.science/tel-00295259>**

Submitted on 11 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.

# THESE

*présentée à*

**l'Institut National Polytechnique de Grenoble**

*pour obtenir le grade de*

**DOCTEUR DE 3<sup>ème</sup> CYCLE**

*par*

**Belkacem SEFSAF**



**CONCEPTION ET REALISATION  
DU SYSTEME DE GESTION DE SERIES CHRONOLOGIQUES  
DU LOGICIEL *MODULECO*.**



soutenue le 19 novembre 1981 devant la Commission d'Examen :

Monsieur G. VEILLON : Président

Messieurs M. ADIBA  
G. BEAUME  
C. DELOBEL  
B. OUDET } Examineurs



# INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Année universitaire 1979-1980

Président : M. Philippe TRAYNARD

Vice-Présidents : M. Georges LESPINARD

M. René PAUTHENET

## PROFESSEURS DES UNIVERSITES

MM.	ANCEAU François	Informatique fondamentale et appliquée
	BENOIT Jean	Radioélectricité
	BESSON Jean	Chimie Minérale
	BLIMAN Samuel	Electronique
	BLOCH Daniel	Physique du Solide - Cristallographie
	BOIS Philippe	Mécanique
	BONNETAIN Lucien	Génie Chimique
	BONNIER Etienne	Métallurgie
	BOUVARD Maurice	Génie Mécanique
	BRISSONNEAU Pierre	Physique des Matériaux
	BUYLE-BODIN Maurice	Electronique
	CHARTIER Germain	Electronique
	CHERADAME Hervé	Chimie Physique Macromoléculaires
Mme	CHERUY Arlette	Automatique
MM.	CHIAVERINA Jean	Biologie, Biochimie, Agronomie
	COHEN Joseph	Electronique
	COUMES André	Electronique
	DURAND Francis	Métallurgie
	DURAND Jean-Louis	Physique Nucléaire et Corpusculaire
	FELICI Noël	Electrotechnique
	FOULARD Claude	Automatique
	GUYOT Pierre	Métallurgie Physique
	IVANES Marcel	Electrotechnique
	JOUBERT Jean-Claude	Physique du Solide - Cristallographie
	LACOUME Jean-Louis	Géographie - Traitement du Signal
	LANCIA Roland	Electronique - Automatique
	LESIEUR Marcel	Mécanique
	LESPINARD Georges	Mécanique
	LONGQUEUE Jean-Pierre	Physique Nucléaire Corpusculaire
	MOREAU René	Mécanique
	MORET Roger	Physique Nucléaire Corpusculaire
	PARIAUD Jean-Charles	Chimie - Physique
	PAUTHENET René	Physique du Solide - Cristallographie
	PERRET René	Automatique

.../...

MM.	PERRET Robert	Electrotechnique
	PIAU Jean-Michel	Mécanique
	PIERRARD Jean-Marie	Mécanique
	POLOUJADOFF Michel	Electrotechnique
	POUPOT Christian	Electronique - Automatique
	RAMEAU Jean-Jacques	Chimie
	ROBERT André	Chimie Appliquée et des matériaux
	ROBERT François	Analyse numérique
	SABONNADIÈRE Jean-Claude	Electrotechnique
Mme	SAUCIER Gabrielle	Informatique fondamentale et appliquée
M.	SOHM Jean-Claude	Chimie - Physique
Mme	SCHLENKER Claire	Physique du Solide - Cristallographie
MM.	TRAYNARD Philippe	Chimie - Physique
	VEILLON Gérard	Informatique fondamentale et appliquée
	ZADWORNY François	Electronique

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

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

#### Personnalités habilitées à diriger des travaux de recherche (décision du Conseil Scientifique)

##### E.N.S.E.E.G.

MM.	ALLIBERT Michel
	BERNARD Claude
	CAILLET Marcel
Mme	CHATILLON Catherine
MM.	COULON Michel
	HAMMOU Abdelkader
	JOUD Jean-Charles
	RAVAINE Denis
	SAINFORT

C.E.N.G.

MM. SARRAZIN Pierre  
SOUQUET Jean-Louis  
TOUZAIN Philippe  
URBAIN Georges

Laboratoire des Ultra-Réfractaires ODEILLO

**E.N.S.M.E.E.**

MM. BISCONDI Michel  
BOOS Jean-Yves  
GUILHOT Bernard  
KOBILANSKI André  
LALAUZE René  
LANCELOT Francis  
LE COZE Jean  
LESBATS Pierre  
SOUSTELLE Michel  
THEVENOT François  
THOMAS Gérard  
TRAN MINH Canh  
DRIVER Julian  
RIEU Jean

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

MM. BOREL Joseph  
CHEHIKIAN Alain  
VIKTOROVITCH Pierre

**E.N.S.I.E.G.**

MM. BORNARD Guy  
DESCHIZEAUX Pierre  
GLANGEAUD François  
JAUSSAUD Pierre  
Mme JOURDAIN Geneviève  
MM. LEJEUNE Gérard  
PERARD Jacques

**E.N.S.H.G.**

M. DELHAYE Jean-Marc

**E.N.S.I.M.A.G.**

MM. COURTIN Jacques  
LATOMBE Jean-Claude  
LUCAS Michel  
VERDILLON André



Je tiens à remercier,

Monsieur G. VEILLON, Professeur à l'Institut National Polytechnique de Grenoble, Directeur de l'ENSIMAG, qui a bien voulu me faire l'honneur de présider le jury de cette thèse.

Monsieur C. DELOBEL, Professeur à l'Université Scientifique et Médicale de Grenoble, qui a accepté d'être membre du jury.

Monsieur M. ADIBA, Professeur à l'Université Scientifique et Médicale de Grenoble, pour l'intérêt qu'il a porté à mon travail et pour sa participation au jury.

Monsieur G. BEAUME, Ingénieur en chef à Eca Automation, qui a bien voulu faire partie du jury.

Monsieur B. OUDET, Professeur à l'Université Scientifique et Médicale de Grenoble, qui ne m'a pas ménagé ses encouragements et qui m'a conseillé tout au long de mon travail.

Je remercie également tous mes camarades de l'équipe MODULECO pour leur fructueuse collaboration.

Elisabeth DUBOIS a dactylographié ce manuscrit avec patience et compétence, je la remercie vivement.

Enfin, je remercie tout le personnel du service de reproduction qui a assuré la réalisation matérielle de cette thèse.





# PLAN

Page

## INTRODUCTION

### CHAPITRE 1

1. LA GESTION DES DONNEES ECONOMIQUES	1.1
1.1. Les données économiques	1.2
1.1.1. Les modèles économétriques	1.2
1.1.2. Les séries chronologiques	1.3
1.1.3. Interaction "modèle économétrique - séries"	1.4
1.2. Les utilisateurs	1.5
1.3. Les traitements	1.6
1.3.1. Gestion	1.6
1.3.2. Manipulations	1.6
1.3.2.1. Manipulations algébriques	1.7
1.3.2.2. Manipulations ensemblistes	1.8
1.3.2.3. Autres manipulations	1.9
1.3.3. Interface "modèles économétriques - séries"	1.10
1.4. Problèmes posés et esquisse de solutions	1.10
1.4.1. Gestion du temps	1.10
1.4.2. Gestion de données à plusieurs dimensions	1.12
1.4.3. Gestion base centrale - bases personnelles	1.13
1.4.4. Interface modèles économétriques - séries	1.14
1.4.5. Facilité d'utilisation par l'économiste	1.14

### CHAPITRE 2

2. LES LACUNES DES SGBD GENERAUX POUR LA GESTION DES SERIES CHRONOLOGIQUES	2.1
2.1. Rappel sur les principaux modèles de données	2.1
2.1.1. Le modèle hiérarchique	2.2
2.1.2. Le modèle réseau	2.3
2.1.3. Le modèle relationnel	2.4
2.2. Etude de quelques SGBD existants	2.5
2.2.1. Un SGBD de type hiérarchique : MIISFIIT	2.5
2.2.2. Un SGBD de type réseau : SOCRATE	2.7
2.2.3. Quelques prototypes relationnels	2.10
2.3. Insuffisances par rapport aux problèmes posés	2.11

**CHAPITRE 3**

3. LES LOGICIELS SPECIFIQUES ET LEURS INSUFFISANCES	3.1
3.1. Types de logiciels	3.1
3.2. Principaux aspects	3.2
3.3. Etude des trois principaux logiciels	3.3
3.3.1. Le logiciel TROLL	3.4
3.3.2. Le logiciel CRONOS	3.5
3.3.3. Le logiciel ARGOS-C	3.7
3.4. Insuffisances par rapport au problème posé	3.9
3.5. Pour la construction d'un logiciel spécifique : MODULECO	3.10

**CHAPITRE 4**

4. STRUCTURE GENERALE DE MODULECO	4.1
4.1. Rappel des objectifs	4.1
4.2. Les différentes composantes du logiciel MODULECO	4.2
4.3. Interface avec le sous-système de gestion de séries chronologiques	4.5

**CHAPITRE 5**

5. LE SOUS-SYSTEME DE GESTION DE SERIES CHRONOLOGIQUES	5.1
5.1. Les solutions apportées aux problèmes de gestion de séries chronologiques	5.1
5.1.1. Gestion simple de données à plusieurs dimensions	5.1
5.1.2. Gestion élaborée de la dimension temps	5.3
5.1.3. Gestion base centrale - segments	5.4
5.1.4. Interface "sous-système de gestion de séries - sous-système de modélisation"	5.6
5.1.5. Facilité d'utilisation	5.7
5.2. Présentation générale du sous-système	5.7
5.2.1. Organisation des données	5.7
5.2.2. Protection des données	5.12
5.2.3. Cohérence des données	5.13
5.2.4. Conflits d'accès	5.13
5.2.5. Traitement sur les données	5.14
5.2.5.1. Gestion	5.14
5.2.5.2. Manipulations	5.15
CONCLUSION	5.17

## **ANNEXES**

1. PRINCIPAUX PROBLEMES D'IMPLEMENTATION - CHOIX
2. PRESENTATION DES COMMANDES

## **BIBLIOGRAPHIE**



## **INTRODUCTION**



En économétrie et notamment en modélisation, le recours à l'outil informatique est devenu indispensable.

Les modèles macroéconomiques atteignant des tailles de plus en plus grandes, leur construction et leur utilisation nécessitent des calculs et des manipulations de données que seul l'ordinateur peut effectuer.

Alors que de nombreux logiciels ont été développés pour traiter tel ou tel aspect de la modélisation, la définition d'un logiciel complet n'a pas reçu jusqu'à présent toute l'attention qu'elle requiert.

Dans le cadre du développement d'un logiciel de modélisation macroéconomique, nous étudions plus particulièrement l'aspect manipulation des séries chronologiques (suites d'observations dans le temps).

Les problèmes que soulève cette fonction sont liés à la nature des séries chronologiques traitées, au mode de travail particulier de certaines catégories d'utilisateurs, aux interactions entre données économiques et à l'orientation spécifique du logiciel pour les économistes.

La nature des séries traitées, séries chronologiques à plusieurs dimensions, conduit à gérer la dimension temps d'une part, et les autres dimensions d'autre part.

La gestion du temps doit permettre de caractériser une série chronologique pour éviter la multiplication de noms; on distinguera, notamment les séries ayant le même nom mais des périodicités distinctes : la périodicité d'une série est l'intervalle de temps séparant deux observations successives. De plus, l'aspect dynamique du critère temps (le nombre d'observations d'une série augmente parallèlement à l'évolution de la dimension temps) introduit une contrainte supplémentaire. Enfin, il convient de ne pas restreindre les possibilités de définition de domaines temporels : il faut prendre en compte aussi bien des séries périodiques (l'intervalle de temps entre chaque observation est toujours le même) que des séries apériodiques; en outre, les séries peuvent être de périodicité classique (annuelle, semestrielle, trimestrielle, etc.) ou quelconque.



La tendance de plus en plus confirmée de la modélisation macroéconomique à traiter des modèles économétriques mondiaux, multirégionaux, etc., conduit à prendre en compte des séries à plusieurs dimensions. La gestion de tableaux multidimensionnels est un problème connu [L.H. Quam, 1980]. Il est particularisé et accentué par le fait que les tableaux sont chronologiques et que les éléments de dimensions ne sont pas seulement des entiers mais aussi des libellés. De plus, les observations de série sont des éléments de tableaux pouvant représenter trois états distincts : l'état "aberrant" (la valeur n'a pas de sens), l'état "manquant" (la valeur n'est pas disponible) ou enfin, la mesure du fait économique représenté par la série pour la date correspondante.

Pour préserver la généralité économique du logiciel, le mode de travail propre à certaines catégories d'économistes implique un compromis entre les besoins de la modélisation macroéconomique et les besoins de la gestion d'une base centrale de séries statistiques. Les modélisateurs sont des utilisateurs économistes particuliers. Ils ne peuvent se contenter d'utiliser directement les séries de la base centrale. Alors que celle-ci doit être mise à jour dès qu'une nouvelle information est disponible, la modélisation macroéconomique nécessite un ensemble de données qui soit "figé" pendant un certain temps. De même, il peut être nécessaire de faire des manipulations qui ne sont pas possibles dans une base publique de données. Ainsi, des modifications ponctuelles permettent de vérifier l'impact d'une variation d'une ou plusieurs variables économiques sur d'autres variables économiques. Elles ne doivent pas être reportées sur les séries de référence du site. De même en est-il des nombreuses manipulations nécessaires à la préparation des séries chronologiques pour l'alimentation d'un modèle macroéconomique.

Les principales données économiques manipulées dans un logiciel de modélisation macroéconomique sont les séries chronologiques et les modèles économétriques. Les interactions entre ces deux types de données consistent à alimenter le modèle macroéconomique en séries chronologiques d'une part,

et à récupérer les résultats produits par le modèle d'autre part. La contrainte essentielle imposée par ces interactions est l'efficacité en temps d'accès notamment lors de l'alimentation du modèle pendant la phase de simulation.

L'orientation spécifique du logiciel pour les économistes amène à définir un langage qui tienne compte de leur vocabulaire. De plus, les manipulations complexes qui sont effectuées nécessitent une aide importante du système. Ainsi, l'entrée des valeurs d'une série chronologique à plusieurs dimensions est une opération complexe si le système ne guide pas l'utilisateur.

Nous montrons que globalement ces problèmes sont non ou mal pris en charge par les logiciels de gestion de bases de données existants.

La solution que nous envisageons s'articule autour de deux axes distincts :

- la gestion d'une base centrale de séries chronologiques appelée à contenir toutes les séries de référence du site,
- la gestion d'un ensemble de segments utilisateurs permettant à tout utilisateur de se créer son ou ses ensembles de données sur lesquels il effectue les manipulations qu'il désire.

Les séries chronologiques à plusieurs dimensions sont traitées à l'aide de calendriers et de nomenclatures.

Les calendriers permettent à l'utilisateur de définir assez simplement des domaines temporels quelconques. Les nomenclatures sont un moyen d'indiquer des séries à plusieurs dimensions.

Les interactions "modèles économétriques - séries chronologiques" ont reçu une attention particulière puisque 'un interface spécifique a été développé.

Enfin, l'économiste retrouve son vocabulaire dans le langage de mise en oeuvre et bénéficie de nombreuses aides, notamment pour la saisie des données.

Pour présenter la solution retenue, nous avons adopté le plan suivant :

. Le chapitre 1 présente "l'univers" de l'économiste à travers les données manipulées, les classes d'utilisateurs mises en cause et les traitements effectués. Les principaux problèmes soulevés sont étudiés en fin de chapitre.

. Après un rappel sur les principaux modèles de données et l'étude de quelques SGBD classiques, un tableau récapitulatif des lacunes des SGBD généraux par rapport aux problèmes posés est dressé dans le chapitre 2.

. Dans le chapitre 3 sont étudiés quelques logiciels spécifiques, développés pour résoudre des problèmes identiques à ceux qui nous sont posés. Un tableau des insuffisances de ces logiciels est présenté en fin de chapitre.

. Un rappel des principaux objectifs du logiciel MODULECO, cadre de notre étude, une présentation sommaire des principales composantes de ce logiciel et une étude des interactions entre le sous-système de gestion de séries et les autres composantes de MODULECO sont faits dans le chapitre 4.

. Nous présentons le sous-système de gestion de séries dans le chapitre 5. Nous détaillons les solutions apportées aux problèmes posés en fin du chapitre 1, puis nous exposons quelques détails d'implémentation et les commandes de mise en oeuvre du sous-système.

. Enfin, la conclusion situe les limites du travail effectué et présente des extensions possibles.

. En annexes sont étudiés quelques problèmes d'implémentation et certains choix qui ont été faits. En outre, les commandes de mise en oeuvre du logiciel sont présentées de manière détaillée.

## CHAPITRE I



## 1. LA GESTION DES DONNEES ECONOMIQUES

De manière générale, la manipulation des données économiques ne comprend pas seulement la gestion et l'accès à la base de données économiques. Elle recouvre également le maniement de toutes les informations nécessaires au modèle économétrique, à son utilisation ou à son analyse.

Sous le titre de manipulation de données, on entend donc la définition de tous les types de données intervenant dans la mise en oeuvre d'un modèle économétrique, leurs interconnexions et leur mode d'accès ainsi que toutes les opérations impliquées lors de cette mise en oeuvre.

Pour concevoir et réaliser le logiciel de gestion de données, nous nous inspirerons des recommandations du rapport ANSI/SPARC [ANSI/SPARC, 1975] qui distingue trois niveaux de description (figure 1.1) :

- étude de l'univers de l'économiste
- définition du schéma conceptuel et des schémas externes
- définition du schéma interne

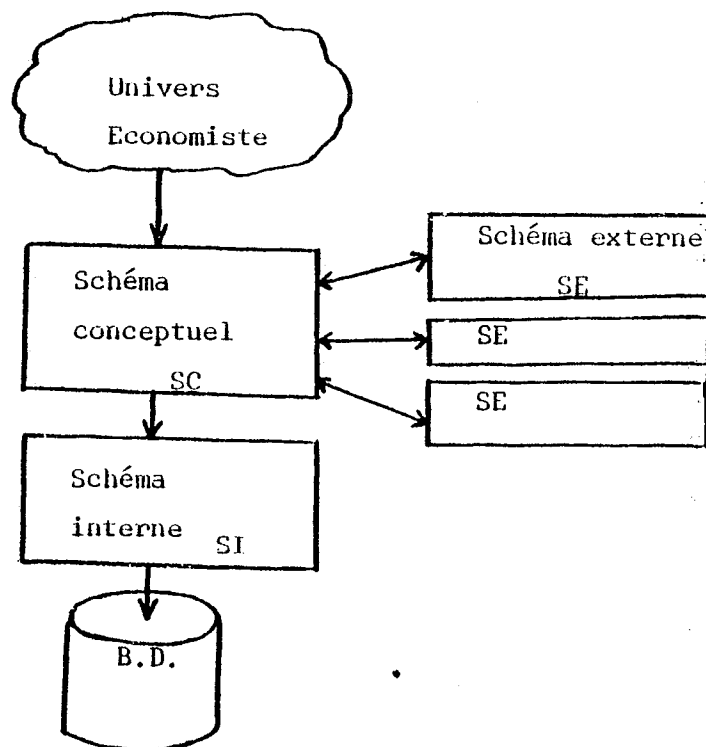


Figure 1.1. : les trois niveaux de description d'une base de données

La première étape se fait en étroite collaboration avec les utilisateurs économistes [Rapport final, 1978] et permet de recenser les données, les classes d'utilisateurs et les traitements. La deuxième étape consiste à modéliser l'univers décrit à l'aide d'un modèle de données connu (hiérarchique, réseau, relationnel) ou voisin d'un modèle de données connu, on obtient ainsi le schéma conceptuel qui s'exprime en terme des langages offerts. Le schéma conceptuel permet de construire des schémas externes correspondant à des classes d'utilisateurs. La troisième étape consiste à définir le schéma interne par le choix des organisations physiques des données et des chemins d'accès physiques.

L'objet de ce chapitre est de décrire la première étape. La deuxième étape sera décrite dans le chapitre 5. Enfin, quelques détails de la troisième étape seront abordés dans le chapitre 5.

### 1.1. Les données économiques

En modélisation macroéconomique on distingue essentiellement deux grands groupes de données : les modèles économétriques et les séries chronologiques.

#### 1.1.1. Les modèles économétriques

Un modèle macroéconomique est défini comme un ensemble d'équations aux différences dont la solution est supposée décrire le comportement d'une économie nationale [D. Gabay et Al, 1978].

Les équations expriment une (ou plusieurs) variables(s) en fonctions d'autres variables et des coefficients. Les variables sont des séries chronologiques. Les coefficients sont le plus souvent fixes : ils ne varient pas dans le temps; leurs valeurs sont calculées par des techniques économétriques. L'application de ces techniques justifie le nom de modèle économétrique.

Formellement, un modèle économétrique s'écrit [P. Malgrange, 1979]

$$\Phi(Y_t, Y_{t-1} \dots Y_{t-p}, X_t, X_{t-1} \dots X_{t-q}, \alpha, e)$$

où  $Y$  est le vecteur des variables endogènes : une variable endogène est une variable calculée par le modèle

- X est le vecteur des variables exogènes : une variable exogène est une variable dont les valeurs alimentent le modèle.
- $\alpha$  est le vecteur des coefficients
- e est le vecteur des variables d'écart : ce sont des séries chronologiques qui permettent selon l'expression de l'économiste de "caler un modèle".

L'exemple suivant illustre par une équation les données présentées :

$$X_{1t} = 2 * X_{3t-1} + C * Y_{1t} + E$$

$X_{1t}$  est une variable endogène ou variable calculée

$X_{3t-1}$ ,  $Y_{1t}$  sont des variables exogènes

C est un coefficient

E est une variable d'écart

Les problèmes informatiques posés par la manipulation de modèles économétriques résident dans :

- la mise à la disposition des économistes d'un langage facilitant l'écriture des équations du modèle
- la fourniture d'un outil d'édition qui permette la mise au point du modèle
- la mise en oeuvre du modèle qui nécessite son alimentation en coefficients et en séries chronologiques.

### 1.1.2. Les séries chronologiques

Une série chronologique est une suite de valeurs dans le temps.

Une série simple est une série chronologique dont la seule dimension est le temps. La figure 1.2 ci-dessous représente la Production Intérieure Brute ou PIB sur l'intervalle  $t_1 \dots t_n$ .

temps	$t_1$	$t_2$	$t_3$	$t_4$	---	$t_n$
PIB	100.5	110.6	140.3	140.8		160.7

Figure 1.2 : une série simple



Outre le temps, une série chronologique peut avoir une ou plusieurs autres dimensions. L'étude d'une économie régionale, multirégionale ou mondiale fait apparaître ce besoin [P. Malgrange, 1979]. Pour un modèle mondial, la série chronologique PIB représente la Production Intérieure Brute pour l'ensemble des pays étudiés par le modèle (cf. figure 1.3) Il faut retrouver dans l'hyper-tableau de valeurs qui représente la PIB, les valeurs de la PIB de chaque pays.

	temps	$t_1$	$t_2$	$t_n$
dimension PAYS				
France		100.1	100.5	110.3
Italie		100.6	110.4	110.0
Espagne		110.4	105.6	94.6

Figure 1.3 : Un exemple de série chronologique à une dimension\*

(\*) : le temps étant considéré comme une dimension implicite, seules les dimensions explicites sont mentionnées.

### 1.1.3. Interactions "modèles économétriques - séries chronologiques"

La figure 1.4. illustre les interactions entre les séries chronologiques et les modèles économétriques.

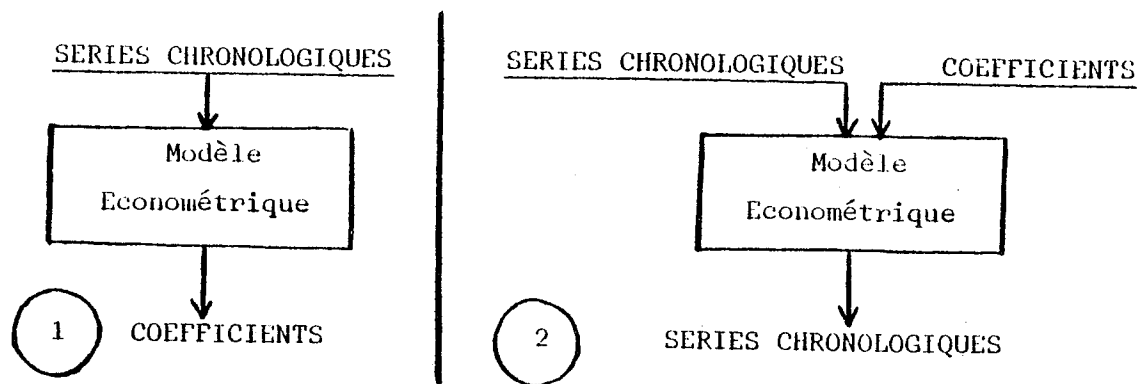


Figure 1.4 : Interactions "modèles économétriques-séries chronologiques"

Ces interactions consistent à :

- 1 - alimenter le modèle économétrique en séries chronologiques, celui-ci produit alors des coefficients
- 2 - alimenter le modèle économétrique en coefficients et en séries chronologiques : il produit alors des séries chronologiques.

## 1.2. Les utilisateurs

On distingue généralement quatre classes d'utilisateurs dans les SGBD classiques [M. Adiba et Al, 1976], [M. Adiba et Al, 1978] : l'administrateur de la base, l'administrateur d'application, le programmeur d'application, l'utilisateur.

En économie ces quatre classes d'utilisateurs se retrouvent avec des rôles différents.

Nous nous intéressons plus particulièrement à mettre en évidence le rôle de l'administrateur de la base et à souligner les grandes catégories d'usagers que l'on retrouve dans la classe utilisateur.

L'administrateur de la base de données économiques est plus communément appelé le gestionnaire de la base de données économiques [B. Pincemaille, 1979]. Son rôle est d'assurer les mises à jour régulières des séries chronologiques dites de "référence". Ces séries de référence, accédées par tous les utilisateurs du site, sont en nombre pratiquement fixe (peu de nouvelles séries sont créées). Le gestionnaire de la base de données économiques doit veiller au maintien de la cohérence de ces séries et à leur sécurité.

Dans la classe utilisateur, on distingue essentiellement deux catégories d'usagers dont les intérêts et les choix techniques ne sont pas compatibles :

- l'utilisateur consultant de base de données : qu'il soit utilisateur courant ou occasionnel, s'intéresse essentiellement à des manipulations simples sur les séries de référence
- le modélisateur : construit et manipule des modèles macroéconomiques. Il construit des sous-ensembles des séries de référence sur lesquels il désire travailler indépendamment de l'évolution des séries de référence.

### 1.3. Les traitements

A travers l'étude des traitements, il s'agit de souligner les particularités de certaines fonctions dans le domaine économique.

#### 1.3.1. Gestion

Ce sont les trois fonctions usuelles qu'assume tout SGBD : création, destruction, mise à jour. A travers ces fonctions doivent être assurées la cohérence, la confidentialité et la sécurité des données.

Si la création et la destruction de séries chronologiques sont des opérations classiques, la mise à jour est beaucoup plus caractéristique. L'exemple suivant met en évidence le problème posé :

$X = Y + Z$  où X, Y et Z sont des séries chronologiques, X étant obtenue à partir de Y et de Z, on dit que X est une série dérivée. Pour le maintien de la cohérence, la mise à jour de Y et/ou de Z doit entraîner la mise à jour de X .

#### 1.3.2. Manipulations

Trois types de manipulations sont effectuées sur les séries chronologiques.

### 1.3.2.1. Les manipulations algébriques

Elles permettent d'effectuer des opérations algébriques (addition, soustraction, multiplication, division) sur les séries chronologiques, d'appliquer certaines fonctions standards sur ces mêmes séries chronologiques. Ces fonctions sont, soit des fonctions classiques (SIN, COS, LOG, ...), soit des fonctions à caractère purement économique [Système CRONOS, 1976D], [Système CRONOS, 1976E].

Remarques : - Les opérations algébriques (+, -, \*, /) sont effectuées sur chaque terme des séries opérandes.

- L'addition d'une scalaire à une série se traduit par l'addition de ce scalaire à chacun des termes de la série. La remarque vaut pour les autres opérations (-, \*, /).

Les problèmes particuliers que soulèvent les opérations algébriques sont les suivants :

- Les séries chronologiques intervenant dans un calcul ne sont pas forcément définies sur les mêmes domaines temporels. Chaque opération algébrique est alors précédée d'un alignement chronologique (cf. 1.3.2.2) opération qui consiste à déterminer le domaine temporel commun.
- Certaines valeurs de séries ne sont pas connues : ces valeurs n'ont pu être obtenues : on dit alors qu'elles sont manquantes, d'autres ne pourront jamais être mesurées : elles sont dites indéterminées ou aberrantes. Si une série a "des trous" (certaines de ses valeurs sont manquantes ou aberrantes), il convient d'adopter une politique lorsqu'elle intervient dans un calcul. L'émission d'un message d'erreur sans effectuer l'opération est une des politiques appliquées, l'autre solution couramment adoptée consiste à considérer les valeurs résultats correspondant à des valeurs opérandes manquantes ou aberrantes, respectivement comme manquantes ou aberrantes.
- L'utilisateur dispose, la plupart des cas, de moyens simples lui permettant de coder valeur manquante ou valeur aberrante. La transcription automatique de ces codes doit être effectué lors de tout calcul sur les séries.

Exemple\*

BALANCE COMMERCIALE = EXPORTATIONS - IMPORTATIONS

t \ SERIE	1975	1976	1977	1978	1979
EXPORTATIONS	50 826	54 530	62 063	75 340	94 782
IMPORTATIONS	49 382	59 279	64 817	73 799	96 100
BALANCE COMMERCIALE	1 444	- 4 749	-2 754	1 540	-1 318

\*Exemple extrait de FRANCE (mai 1980) OCDE, Etudes économiques.

### 1.3.2.2. Manipulations ensemblistes

De nombreuses opérations sous-jacentes à des traitements sur les séries chronologiques se formalisent à l'aide des opérateurs de l'algèbre relationnel (Projection, Jointure, Fusion, ...) [M. Adiba et Al, 1976], [J.Y. Caleca, 1978]

Ainsi, la fusion, la différence ensembliste de séries chronologiques respectent les mêmes contraintes que celles énoncées dans [M. Adiba et Al, 1976].

De même la sélection de séries chronologiques consiste à obtenir les valeurs d'une série chronologique sur les valeurs fixées de son domaine de définition. Elle se traduit par des opérations de projection sur les constituants du domaine de définition de la série. Cette manipulation s'exprime, soit directement dans une opération de calcul :  $PIB_{FRANCE}$   $PIB_{FRANCE}_t = PIB_{FRANCE,t}$ , soit est sous-jacente dans les opérations d'édition : IMPRIMER PNB (si PNB est une série à plusieurs dimensions, il faut imprimer chaque série simple de la série PNB).

Plus caractéristique des séries chronologiques est l'alignement chronologique de séries. Il consiste en plusieurs opérations :

- L'alignement de la périodicité (cf. 1.4.1.) ramène des séries chronologiques de périodicités différentes à une même périodicité de référence. Cette opération s'accompagne, en général, d'opérations algébriques (§ 1.3.2.1) ou d'application de fonctions standards (§ 1.3.2.1).

- . L'alignement des intervalles de temps détermine un intervalle de temps commun à un ensemble de séries.
- . La superposition de deux séries chronologiques consiste soit à les juxtaposer s'il n'y a pas de recouvrement chronologique, soit, s'il y a recouvrement, à conserver les parties distinctes et à choisir pour la partie chronologique commune, une des deux séries.

### 1.3.2.3. Autre manipulations

Dans [D. Fouquet, 1979], D. Fouquet dresse une liste des manipulations les plus courantes faites par les économistes. Certaines des ces manipulations ont déjà été présentées. Parmi celles qui n'ont pas été présentées, on relève de nombreuses opérations globales et des opérations d'éditations.

Les manipulations les plus caractéristiques sont : la fusion, la comparaison, la copie de bases de données, l'édition de bases de données ou de données individuelles. Enfin, des fonctions d'interface sont fortement souhaitées [B. Pincemaille, 1979B] pour permettre aux utilisateurs l'échange de données entre divers logiciels (cf. figure 1.5).

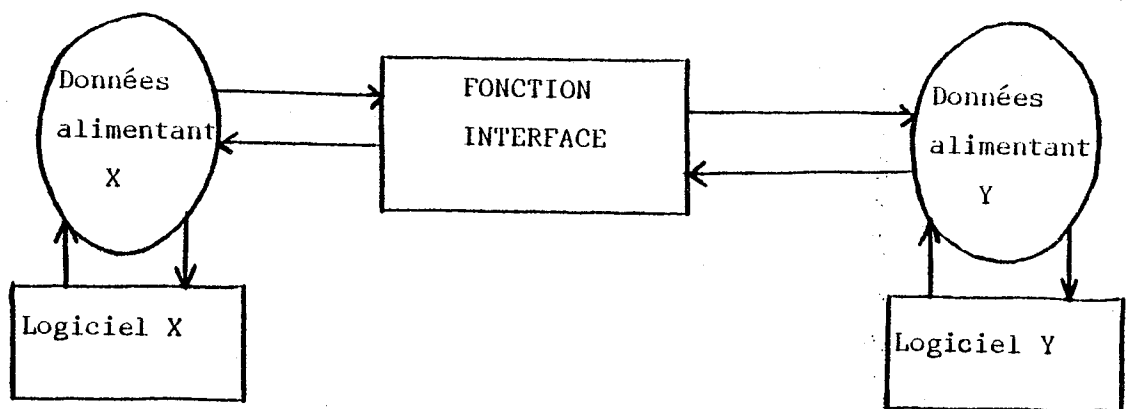


Figure 1.5 : travail sur des logiciels distincts

### 1.3.3. Interface "modèles économétriques - séries"

Les interactions "modèles-séries" (cf. 1.1.2) se traduisent par la nécessité de deux interfaces :

- un interface permettant d'alimenter le modèle économétrique en séries chronologiques;
- un interface permettant de conserver les séries chronologiques calculées par le modèle.

## 1.4. Problèmes posés et esquisse de solution

La gestion des séries chronologiques à plusieurs dimensions implique la gestion du critère temps et la gestion des autres dimensions de la série. Le mode de travail particulier des modélisateurs nécessite le développement de deux pôles de gestion distincts. L'axe privilégié "modèles-séries" impose le développement d'un interface spécifique. Enfin, les utilisateurs économistes souhaitent un logiciel facile à utiliser.

### 1.4.1. Gestion du temps

Selon les ouvrages, l'aspect temporel joue des rôles différents : dans [J.R. Abrial, 1974], le temps matérialisé par une date caractérise la donnée à laquelle il est attaché (une personne n'a qu'une date de naissance), il joue un rôle d'identifiant. Dans [N.G. Toan, 1977], l'aspect temporel est évoqué comme décrivant une chronologie.

L'aspect temporel qui nous préoccupe joue à la fois un rôle d'identifiant (la série chronologique  $(N,T)$  et la série chronologique  $(N,T')$  sont deux séries distinctes, et un rôle chronologique : il définit un domaine de validité dynamique pour une série (une série est définie sur  $t_1..t_p$  puis sur  $t_1..t_{p+1}, \dots$ ).

De par ces aspects particuliers, le temps est considéré comme une dimension privilégiée.

On conviendra de le représenter de la manière suivante :

- un triplet (p,dd,df) où :
  - . p est la périodicité de la série : une série est annuelle si elle a une valeur tous les ans, elle est trimestrielle si elle est définie tous les trimestres, etc.
  - . dd est la date correspondant à la première observation de la série.
  - . df est la date correspondant à la dernière observation de la série.
- Une liste de dates  $dt_1, \dots, dt_n$ .

Nous désignons par calendrier indifféremment le triplet (p,dd,df) ou la liste de dates  $dt_1, \dots, dt_n$ . Le domaine temporel d'une série est parfaitement défini par la donnée du calendrier que l'on associe à la série.

La volonté de ne pas restreindre la définition de la dimension temps revient à offrir la possibilité de définir des calendriers quelconques.

De manière à simplifier au maximum la gestion du temps, on distinguera deux types de calendriers : les calendriers classiques et les calendriers définis.

Un calendrier classique est défini à l'aide des périodicités classiques: annuelle, quadrimestrielle, trimestrielle, bimensuelle, hebdomadaire, quotidienne.

Connaissant une date de référence  $d_1$  ( $dd \leq d_1 \leq df$ ) et la périodicité p, on peut déduire la date suivante  $d_2$  à l'aide d'un algorithme interne.

L'utilisateur est entièrement déchargé de la gestion de ces calendriers, seule la donnée du triplet (p,dd,df) est nécessaire.

Les calendriers définis permettent à l'utilisateur de définir des domaines temporels qui ne peuvent être représentés à l'aide des calendriers classiques (par exemple, des années budgétaires de 15 mois), la définition de ces calendriers est à la charge de l'utilisateur, le logiciel doit ensuite gérer ces calendriers.



### 1.4.2. Gestion de données à plusieurs dimensions

Le temps étant une dimension particulière (cf. 1.4.1), on notera une série chronologique à  $n$  dimensions :

$$(N,T) (D_1,D_2,\dots,D_n)$$

où  $N$  est le nom de la série

$T$  est le domaine temporel de la série

$D_1,D_2,\dots,D_n$  sont les domaines où sont prises les valeurs des dimensions de chacune des séries simples de la série à  $n$  dimensions.

La gestion des tableaux à plusieurs dimensions n'est pas un problème nouveau [L.H. Quam, 1980]. Toutefois, les tableaux que nous gérons sont des tableaux chronologiques et les dimensions ne sont pas uniquement des entiers : PIB(FRANCE).

La gestion des domaines  $D_1,\dots,D_n$  dimensionnant une série pose trois problèmes essentiels :

- Premièrement, extraire une série simple d'une série à plusieurs dimensions à partir de son identification explicite :

Exemple : L'utilisateur doit retrouver les valeurs de la série simple PIB(FRANCE) appartenant à la série à une dimension PIB, PIB étant dimensionnée par PAYS = (FRANCE,ITALIE,ESPAGNE).

De manière générale, la série simple  $(N,T) (d_1,\dots,d_n)$  doit être extraite de la série à  $n$  dimensions  $(N,T) (D_1,\dots,D_n)$

où  $d_1$  est un élément de la dimension  $D_1$

⋮

$d_n$  est un élément de la dimension  $D_n$ .

- Deuxièmement, régénérer à tout moment l'identification explicite de chacune des séries simples d'une série à plusieurs dimensions.
- Enfin, des deux nécessités précédentes il découle qu'il faut conserver et retrouver les éléments de dimensions d'une série.

### 1.4.3. Gestion base centrale - bases personnelles

Le modélisateur est un utilisateur bien particulier de bases de données puisqu'il veut se constituer à partir des données de référence des ensembles de données propres sur lesquels il désire :

- travailler sans que les mises à jour éventuelles survenant sur les données de référence soient répercutées sur ses propres données;
- effectuer des traitements qu'il ne peut faire sur les données de références;
- alimenter en séries chronologiques des modèles macroéconomiques et récupérer les résultats le plus efficacement possible;
- travailler sur des corpus de données raisonnables en dimension [Instruments de stratégies économiques, 1977] pour des raisons d'efficacité.

Il convient lors de la définition globale de privilégier l'aspect modélisation sans interdire l'aspect gestion et utilisation d'une grosse base de séries chronologiques [D. Fouquet, 1979].

Une solution est le développement de deux pôles de gestion distincts : la gestion d'une grosse base de séries statistiques appelée à contenir toutes les séries de référence, cette base serait orientée vers l'aspect documentaire; la gestion de bases personnelles qui incluerait des traitements beaucoup plus sophistiqués reflétant les besoins des modélisateurs.

La gestion de bases personnelles dont il est question ici présente des analogies avec les "vues instantanées" que préconisent Adiba et Lindsay dans [M. Adiba et Al, 1980] :

les données personnelles peuvent être le résultat de requêtes sur la base centrale, elles peuvent être "rafraîchies" régulièrement par les données de référence  
mais aussi des différences :

l'utilisateur peut faire de nombreux traitements sur ses données propres, y compris des mises à jour indépendamment des données de référence.

Les données ne sont pas forcément des sous-ensembles des séries de référence mais des données que s'est constitué l'utilisateur.

#### 1.4.4. Interface "modèle économétrique - séries"

Privilégier le modélisateur, c'est aussi privilégier l'axe "modèle-séries". Pour ce faire, l'interface spécifique permettant d'alimenter un modèle économétrique et de récupérer les résultats de ce modèle doit respecter au moins deux conditions :

- une contrainte d'efficacité : pendant la mise en oeuvre du modèle économétrique, son alimentation en séries chronologiques doit être la plus rapide possible. Cette contrainte justifie aussi le développement de deux pôles de gestion distincts [B. Sefsaf et Al, 1979];
- une contrainte de cohérence : la description des variables dans le langage de description de modèles doit être la même que celle du langage de gestion de séries.

#### 1.4.5. Facilité d'utilisation pour l'économiste

Pour offrir aux économistes un logiciel simple à mettre en oeuvre, il faut prendre en compte les aspects suivants :

- . La structure du langage, quelle que soit la tâche à mettre en oeuvre, doit rester homogène. La prise en compte de la spécificité de l'économiste est nécessaire. Ainsi, celui-ci doit retrouver à travers le langage son vocabulaire familier. La mise en oeuvre de chaque fonction doit être facilitée : un minimum de paramètres doivent être fournis lors d'une requête, la plupart pouvant être initialisés pour toute une session. Il ne faut pas oublier l'aspect documentaire très important lors de l'apprentissage : l'utilisateur doit être guidé au moment de la soumission d'une requête.
- . La facilité d'entrée des données : l'entrée des valeurs de séries est une opération qui peut être très longue si le logiciel ne guide pas l'utilisateur, notamment pour une série chronologique à plusieurs dimensions. En outre, des facilités doivent être offertes pour la saisie de plusieurs valeurs identiques, pour coder valeur manquante et valeur aberrante (cf. 1.3.2.1).
- . Complète intégration du logiciel : l'utilisateur doit pouvoir passer d'une tâche à une autre le plus facilement possible.

## CHAPITRE II



## 2. LES LACUNES DES SGBD GENERAUX POUR LA GESTION DES SERIES CHRONOLOGIQUES

Dans le chapitre précédent, nous avons présenté l'univers de l'économiste. Pour modéliser cet univers, nous avons le choix entre deux solutions :

- adopter un modèle de données général, soit parmi les modèles classiques (hiérarchique, réseau, relationnel) [S.A. Borkin, 1980], [W.W. Chu, 1979], soit parmi des modèles individuels [H. Tardieu et Al, 1975], [V. Chamlam, 1979] et décrire la solution suivant ce modèle;
- s'inspirer de logiciels conçus spécifiquement pour répondre à cet univers et en déduire une solution qui prenne en compte les problèmes posés.

Après un bref rappel des caractéristiques des trois modèles les plus répandus, nous justifions notre choix de la seconde solution en soulignant à travers l'étude de MIISFIIT [CETE, 1978], SOCRATE [SOCRATE] et de quelques prototypes relationnels, les lacunes des SGBD généraux conçus selon des approches classiques, pour gérer des séries chronologiques.

Un tableau récapitulatif situe les capacités de ces logiciels par rapport aux problèmes posés à la fin du chapitre 1.

### 2.1. Rappel sur les principaux modèles de données

Le processus de modélisation permet de visualiser la réalité de manière indépendante de tout traitement informatique [M. Adiba, 1978]. Divers modèles de données ont été proposés et les trois modèles les plus connus sont décrits dans de nombreux ouvrages [C.J. Date, 1977], [M. Adiba et Al, 1978], [S.A. Borkin, 1980], [W.W. Chu, 1979].

Nous rappelons ci-après les traits principaux de ces trois modèles.

### 2.1.1. Le modèle hiérarchique

Une base de données de type hiérarchique peut être représentée par une structure d'arbre. La figure 2.1 tirée de [M. Adiba et Al, 1978] illustre ce type de structure :

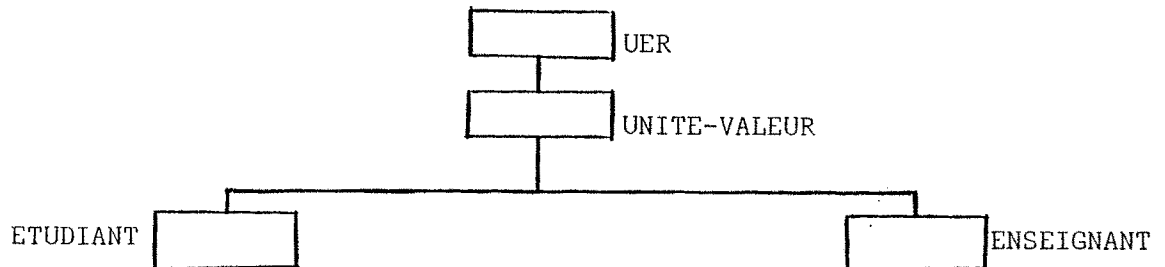


Figure 2.1 : Une structure hiérarchique

Les informations sont découpées en occurrences de segments, chacune représentant une relation de type 1:n, ceci est illustré par la figure 2.2. ci-dessous :

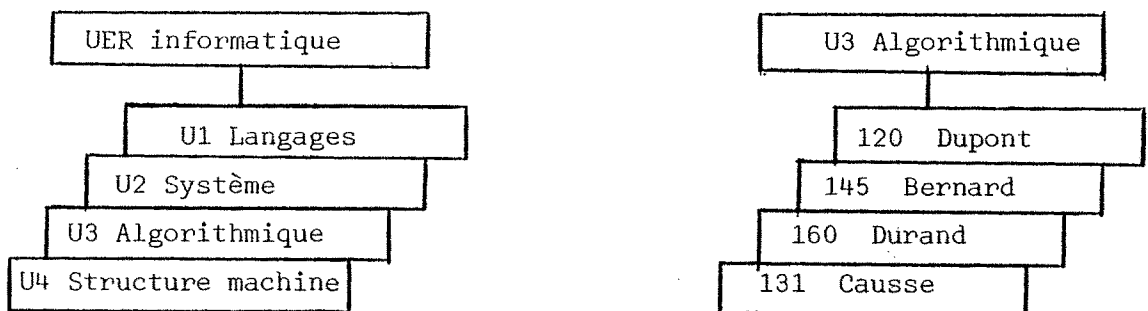


Figure 2.2 : Relations hiérarchiques de type 1:n

Parmi les SGBD qui adoptent cette approche, on peut citer IMS mais aussi MIISFIIT [CETE, 1978] que nous étudierons un peu plus loin. L'approche hiérarchique est une représentation simple de l'information. Toutefois, elle entraîne une dépendance vis à vis des structures physiques et elle modélise mal le monde réel, ce qui conduit à une duplication de données.

### 2.1.2. Le modèle réseau

Cette approche résoud en partie le problème de la duplication de données introduit par le modèle précédent. A l'origine de ce modèle se trouvent les travaux du groupe CODASYL [C. Dufour, 1980], [E.K. Clemons, 1979]. IDS a été le premier SGBD conçu selon cette approche. Enfin, SOCRATE [SOCRATE], [J.R. Abrial et Al, 1970] qui est un SGBD réseau reposant sur une approche différente de celle adoptée par CODASYL, a joué un rôle dans le développement des modèles réseaux. Il fait l'objet d'une étude plus détaillée en 2.2.2.

L'approche réseau propose une structure de données chaînées. La base est alors vue comme un ensemble de noeuds associés par des liens. Le lien matérialise l'association entre enregistrements.

A priori, il ne peut y avoir de restriction sur le type de l'association (1:1, 1:N, N:M), toutefois, la plupart des SGBD commercialisés en imposent.

La figure 2.3 ci-dessous donne un exemple de structure en réseau.

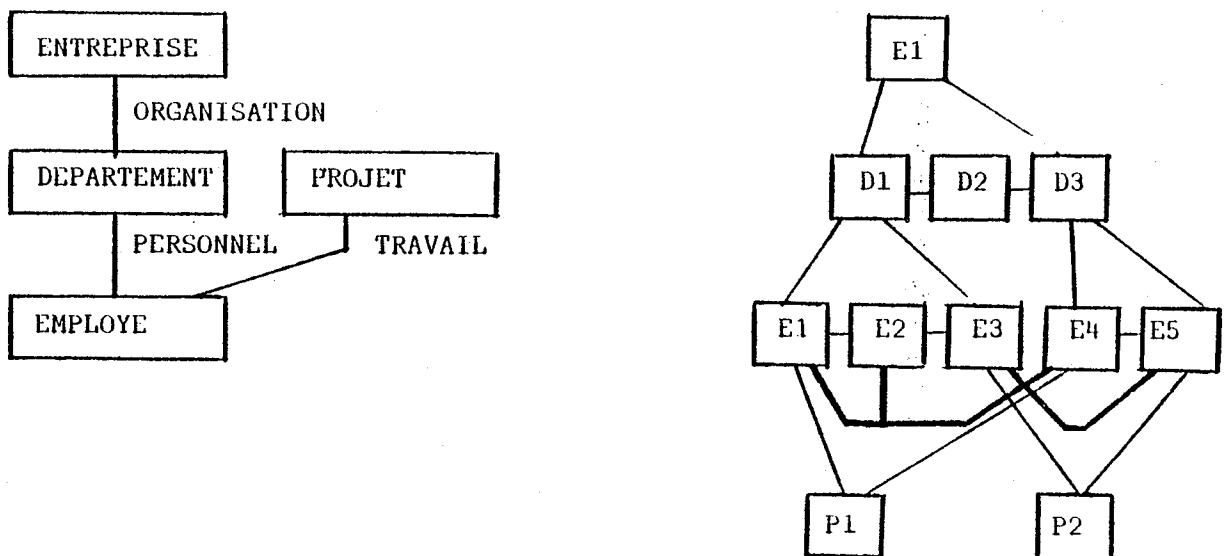


Figure 2.3 : Un exemple de structure en réseau



### 2.1.3. Le modèle relationnel

Les modèles précédents présentent le défaut de figer le chemin d'accès à l'information. Le modèle relationnel, introduit par CODD [E.F. Codd, 1970] apporte une solution à ce problème.

Il a par ailleurs les objectifs suivants :

- un meilleur découplage entre la structure logique et la structure physique des données
- une plus grande indépendance entre les données et leur utilisation
- permettre à chaque base d'évoluer au fur et à mesure des besoins des utilisateurs [M. Adiba, 1978].

Il repose sur la théorie des relations et constitue un axe important de la recherche dans les SGBD. Une relation est notée  $R(X_1:D_1, X_2:D_2, \dots, X_n:D_n)$  ou encore  $R(X_1, X_2, \dots, X_n)$  ou plus simplement  $R(X)$ . C'est un ensemble de vecteurs de  $n$  composantes appartenant au produit cartésien  $D_1 \times D_2 \times \dots \times D_n$  où chaque  $D_i$  désigne le domaine du constituant  $X_i$  de la relation. Les  $D_i$  ne sont pas forcément distincts.

Une relation est représentée par un tableau à  $n$  colonnes ( $n$  est le degré de la relation). Chaque ligne est une réalisation de la relation (cf. figure 2.4).

PERSONNE (N°SS, NOM, PRENOM, PROFESSION, DATENAIS)

PERSONNE

N° SS	NOM	PRENOM	PROFESSION	DATENAIS
1420442388320	DUPONT	JACQUES	FACTEUR	01/04/42
2500838450610	DURAND	FRANCOISE	STANDARDISTE	09/08/50
1370143720011	BASILE	JEAN YVES	MENUISIER	31/01/37
1530775505001	BAPTISTE	JEAN MARIE	PEINTRE	12/07/53
2271013608904	AMEDEE	MICHELINE	INSTITUTRICE	11/10/27
1480969927805	ROBY	JEAN	COIFFEUR	21/09/48

Figure 2.4 : Un exemple de relation

Dans le processus de modélisation relationnel, la description de l'univers en terme de relations est suivi d'opérations de normalisation destinées à supprimer des informations redondantes, à obtenir une indépendance vis à vis des modifications des relations et de rendre les relations neutres vis à vis des requêtes [E.F. Codd, 1971], [E.F. Codd, 1971b].

Associés au modèle relationnel, de nombreux langages de manipulations ont été proposés [M. Adiba et Al, 1976], [C.J. Date, 1977].

Les principales caractéristiques de ces langages sont :

- ils se veulent le moins procédural possible
- certaines requêtes nécessitent l'utilisation d'un langage hôte
- ils n'expriment aucune contrainte d'implémentation physique
- ils sont conçus pour des classes d'utilisateurs quelconques.

## 2.2. Etude de quelques SGBD existants

Peut-on se satisfaire de logiciels généraux existants pour gérer des séries chronologiques ?

L'étude sommaire de quelques SGBD généraux situe les limites de ces logiciels par rapport aux problèmes posés.

### 2.2.1. Un SGBD de type hiérarchique : MIISFIIT

MIISFIIT est un système de gestion de bases de données développé par le C.E.T.E. d'Aix-en-Provence. Il gère une base de données de type hiérarchique.

Pour en décrire les principales caractéristiques, nous nous appuyerons sur les présentations qui en ont été faites dans [Journées banque de données, 1971], [CETE, 1978], [CETE, 1978B].

L'expression d'une base de données MIISFIIT se fait à l'aide de trois objets : les caractéristiques, les individus et les relations entre les individus.

Pour décrire et manipuler des individus, MIISFIIT offre les deux langages classiques de tout SGBD :

- le LDD ou Langage de Définition de Données
- le LMD ou Langage de Manipulation de Données.

De plus, MIISFIIT fournit à l'utilisateur un moyen de "dialoguer" avec sa base grâce au langage conversationnel. Le LDD permet de créer autant de bases de données physiques (BDP) que l'on veut (commande CREER), de créer des individus dans les BDP (commande IND) et d'y décrire leurs caractéristiques (commande CAR ou ACC si la caractéristique est une caractéristique d'accès).

Si MIISFIIT permet de résoudre le problème de gestion de bases personnelles puisqu'il offre la possibilité de créer autant de BDP que nécessaire l'exemple ci-dessous montre toutefois ses importantes faiblesses pour la gestion du critère temps et surtout pour la gestion de données à plusieurs dimensions.

Exemple

CREER BDSERIE MAX(1000)	Création d'une base de données physique BDSERIE de 1000 enregistrements
IND SERIE	Description de l'individu SERIE
CAR NOMSER CARACTERE (16)	Description des caractéristiques simples de la série
CAR TYPESER MOT	
CAR PERIODICITE CARACTERE (16)	
CAR DOCUMENTATION CARACTERE (80)	Gestion du temps MIISFIIT
IND VALEUR (SERIE)	
CAR VALSER MOT	individu VALEUR qui se décompose en couples VALSER-DATVAL
IND DATVAL (VALEUR)	
CAR JOUR MOT	Individu DATMISAJOUR rattaché à l'individu SERIE. Il est composé des caractéristiques JOUR, MOIS et ANNEE.
CAR MOIS MOT	
CAR ANNEE MOT	
IND DATEMISAJOUR (SERIE)	
CAR JOUR MOT	
CAR MOIS MOT	
CAR ANNEE MOT	

A l'exemple précédent, on s'aperçoit que la gestion du temps permise par MIISFIIT est rudimentaire, elle est prise en charge par l'utilisateur qui doit définir ses couples dates-valeurs. De plus, la description de séries à plusieurs dimensions est impossible à l'aide du LDD tel qu'il est défini dans [CETE, 1978]. En outre, le LMD MIISFIIT [CETE, 1978], orienté essentiellement vers la gestion et la consultation est très éloigné des besoins exprimés en 1.3 : manipulations algébriques et manipulations ensemblistes notamment.

### 2.2.2. Un SGBD de type réseau : SOCRATE

Développé à l'IMAG puis commercialisé par la Société ECA-AUTOMATION, SOCRATE est un SGBD réseau par référence.

Pour présenter SOCRATE, nous nous inspirerons des présentations qui en ont été faites dans [SOCRATE], [J.R. Abrial et Al, 1970], [G. Vigliano, 1970], [J.R. Abrial et Al, 1972].

SOCRATE permet le traitement des structures hiérarchiques et des structures en réseau. Il manipule des entités.

Une entité est un bloc pouvant avoir plusieurs réalisations. Un bloc regroupe des données élémentaires et/ou d'autres blocs.

Une entité, un bloc ou une donnée élémentaire est défini par sa ou ses caractéristiques.

La structure des objets SOCRATE est décrite à l'aide du langage de définition de structure.

Celui-ci permet de décrire :

- la structure (ou schéma) d'une base de données
- les sous-ensembles ou sous-structures vus par les différents utilisateurs (sous-schémas)
- des structures de données classiques : structure ponctuelle, structure hiérarchique linéaire, structure hiérarchique arborescente, structure en réseau.

- Trois types de relations permettent de définir des structures réseau :
- le type référence : analogue au type pointeur, il permet d'atteindre directement une réalisation d'entité quelconque  
employé REFERENCE UNE voiture DE UNE entreprise
  - le type anneau : associé à une caractéristique de type référence, il permet de décrire une filiation.  
employé REFERENCE UNE voiture  
voiture ANNEAU
  - le type inverse : un inverse permet de définir un sous-ensemble de réalisations d'une entité répondant à certains critères.  
élève INVERSE TOUT enfant

Les structures réseau de SOCRATE n'apportent guère d'améliorations par rapport à MIISFIIT. L'exemple suivant montre que SOCRATE tout comme MIISFIIT ne permet de décrire que des séries simples (aucune dimension hormis le temps) au prix d'une gestion rudimentaire du critère temps.

Exemple

ENTITE 200 SERIE	. Déclaration d'une entité de
DEBUT	200 blocs SERIE
DATMISAJOUR	. Caractéristique NOMSER de 16 c
DEBUT	caractères
JOUR DE 1 A 31	. bloc DATMISAJOUR ayant 3 caractères :
MOIS DE 1 A 12	JOUR entier entre 1 et 31
ANNEE DE 1900 A 2000	MOIS entier entre 1 et 12
FIN	ANNEE entier entre 1900 et 2000
TYPESER (PRIMAIRE,DERIVEE)	. Caractéristique TYPESER prenant ses valeurs dans la liste de valeurs PRIMAIRE,DERIVEE
PERIODICITE MOT 16	. Caractéristique PERIODICITE de 16 caractères

```

ENTITE 100 VALEURS
  DEBUT
    DATVAL
      DEBUT
        JOUR DE 1 A 31
        MOIS DE 1 A 12
        ANNEE DE 1800 A 2000
      FIN
    VALSER DE -9.E15 A 9.E15
  FIN
DOCUMENTATION TEXTE 4
FIN

```

- . entité de 100 blocs VALEURS
- . bloc DATVAL ayant 3 caractéristiques (JOUR, MOIS, ANNEE)
- . caractéristique VALSER prenant des valeurs réelles entre -9.E15 et 9.E15
- . caractéristique DOCUMENTATION de 4 lignes maximum

Si SOCRATE présente de sérieuses lacunes pour la gestion du temps et de séries multidimensionnelles, il faut noter que la gestion de bases personnelles, tout comme dans MIISFIIT, est possible.

Le langage de requête permet de manipuler les structures SOCRATE. Il se subdivise en un langage de citation et un langage de traitement.

- . Le langage de citation permet de désigner les objets de la base et leurs propriétés, des variables de travail, des objets appartenant à des structures de données classiques.

Nomser DE TOUTE série AYANT périodicité = 'ANNUELLE'

- . Le langage de traitement contient des éléments d'itération (POUR...FIN) des instructions conditionnelles (SI...ALORS...FIN, SI...ALORS...SINON...FIN), des opérateurs de condition (<, ≤, =, ≥, >, =), des opérateurs logiques (ET,OU), des opérateurs algébriques (+, -, \*, /) et des commandes élémentaires permettant la génération (commande G), la suppression (commande S et SE), la mise à jour (commande M) et l'interrogation (commande I).

En outre, SOCRATE offre divers services tels que un éditeur de texte, un macrogénérateur, de nombreux utilitaires permettant entre autres, de faire des manipulations globales sur la base, des mécanismes sûrs permettant d'assurer la confidentialité, l'intégrité et la sécurité de la base.

Si le langage de description de structure est complètement inadapté pour la description de séries chronologiques à plusieurs dimensions, le langage de manipulation SOCRATE est très puissant et permet d'effectuer la plupart des traitements présentés en 1.3.

### 2.2.3. Quelques prototypes relationnels

Bien que l'expérience de l'utilisation de logiciels relationnels soit encore peu étendue, il semble intéressant d'étudier brièvement quelques prototypes pour en tirer les avantages et les inconvénients par rapport à l'application qui nous préoccupe.

Pour faire cette étude, nous nous appuyerons surtout sur [M. Adiba et Al, 1976].

Le développement des premiers modèles relationnels remonte à 1967, mais ce ne sont pas des SGBD relationnels, tout au plus permettent-ils de valider des mécanismes de structuration en mémoire. Il faut attendre 1970 [E.F. Codd, 1970] pour relancer la recherche sur les modèles relationnels. On assiste alors à une véritable éclosion de systèmes relationnels qui se distinguent soit par le modèle dont ils s'inspirent [n-aire, binaire], soit par la nature du langage. On trouve dans [M. Adiba et Al, 1976] un panorama des principaux systèmes développés.

Les principales caractéristiques de ces systèmes sont les suivantes :

- Ils offrent une architecture à plusieurs niveaux permettant d'assurer l'indépendance des données [N. Spyrtatos et Al, 1977].

Ainsi, SYSTEM-R [M.M. Astrahan et Al, 1976], [M.W. Blasgen et Al, 1979] a une architecture à deux niveaux :

- . le RSS (Relational Storage System) : système de stockage relationnel qui gère les périphériques, l'allocation mémoire, les mécanismes de recouvrement, les accès concurrents, les index de relations et les pointeurs entre relations. Le RSS est accessible via un interface nommé RSI pour Relational Storage Interface.
- . le RDS (Relational Data System) : système de données relationnelles permettant la description et la manipulation de relations. Le RDI (Relational Data Interface) est l'interface externe pouvant être appelé directement au travers d'un langage hôte ou par le langage offert par SYSTEM R : SEQUEL.

- Ils offrent des langages relationnels de haut niveau utilisables directement ou par l'intermédiaire d'un langage hôte. Ces langages regroupent les traditionnels LDD et LMD des SGBD classiques et permettent de créer, supprimer, modifier des relations, insérer et supprimer des n-uplets, exprimer des contraintes d'intégrité.
- Ils contiennent des mécanismes très étudiés de contrôle des accès concurrents, de protection des informations [F.M. Bancilhon et Al, 1977]
- Ils offrent la possibilité de définir des vues de la base de données.

Par rapport à notre application, les principales faiblesses des systèmes relationnels sont les suivantes :

- . perte d'efficacité qui résulte de la volonté des prototypes relationnels d'assurer l'indépendance des données
- . pas de possibilité de gestion sophistiquée de l'aspect temporel des données chronologiques
- . langages trop généraux pour satisfaire des utilisateurs spécifiques qui désirent, à travers le langage, manipuler des objets connus
- . langages inadaptés aux traitements des données économiques
- . gestion de bases personnelles sous la forme de "vues instantanées" [M. Adiba et Al, 1980] qui ne correspond pas au besoin exprimé en 1.4.3.

### 2.3. Insuffisances par rapport au problème posé

Le tableau ci-après récapitule les lacunes des SGBD généraux en resituant chacun d'entre eux par rapport aux cinq problèmes posés : gestion du temps, gestion de séries à plusieurs dimensions, gestion de bases personnelles, interface modèles économétriques - séries, facilité d'utilisation pour l'économiste.



Logiciel Problèmes posés	MIISFIIT	SOCRATE	Prototypes Relationnels
Gestion du temps	Pas de gestion du critère temps. L'utilisateur associe à chaque valeur la date correspondante	Pas de gestion du critère temps L'utilisateur associe à chaque valeur la date correspondante	Pas de gestion du critère temps. Si on veut le définir dans une relation, c'est un constituant comme un autre dont il faut donner toutes les valeurs
Gestion de séries chronologiques à plusieurs dimensions	impossible	impossible	possible
Gestion de bases personnelles	possible	possible	possible mais inadaptée au problème posé
Interface modèles économétriques-séries	impossible	impossible	impossible
Facilité d'utilisation pour l'économiste	Langages simples mais inadaptés aux besoins de l'économiste	Langage de manipulation puissant, simple à utiliser mais il manque de certaines facilités que souhaite l'économiste	Langages généraux inadaptés aux besoins des économistes

### CHAPITRE III



### 3. LES LOGICIELS SPECIFIQUES ET LEURS INSUFFISANCES

Nous avons vu dans le chapitre précédent que les logiciels généraux de bases de données ne répondent pas aux problèmes posés : gestion du temps, gestion de séries à plusieurs dimensions, interface modèle-séries. L'étude doit être orientée vers des logiciels spécifiques qui ont été développés pour résoudre des problèmes semblables aux nôtres. Ces logiciels sont nombreux, notamment aux Etats-Unis, et vont du logiciel de gestion de séries au logiciel complet intégrant les séries et les modèles économétriques.

Après avoir rappelé les principaux types de logiciels rencontrés, nous en présentons les aspects essentiels. Nous étudions plus en détail trois logiciels qui sont les plus caractéristiques du problème posé puis nous en soulignons les insuffisances. Enfin, nous présentons les principaux apports du logiciel MODULECO, cadre de notre étude, par rapport aux logiciels spécifiques existants.

#### 3.1. Types de logiciels

Les logiciels étudiés se répartissent dans les classes suivantes :

- Gestion de grosses bases de données statistiques (plusieurs milliers de séries) : le logiciel CRONOS [Système CRONOS, 1976], [Système CRONOS, 1976B] qui fait l'objet d'une étude plus détaillée, est le plus caractéristique de cette classe, mais on y range aussi le logiciel SAD [Journées banques de données, 1971] qui a été développé par la délégation à l'aménagement du territoire et à l'action régionale.

- Traitement de fichiers d'enquêtes : à partir de données statistiques brutes, le logiciel permet de générer des tableaux statistiques de formats divers. Parmi les logiciels de ce type, on retiendra le logiciel OSIRIS [Système OSIRIS, 1978].

- Traitement de tableaux statistiques : ils permettent d'effectuer des traitements élaborés sur des tableaux statistiques : ARGOS-C [ARGOS-C, 1977C], que nous présenterons plus loin, BIBLOS [BIBLOS, 1976] et EXPRESS [EXPRESS] sont des logiciels de ce type.

- Logiciels complets de modélisation économétrique : ils offrent tous les services nécessaires à la modélisation et incluent un module de gestion de séries chronologiques. Le logiciel le plus typique de cette classe est le système TROLL [Troll, 1974] que nous verrons plus en détail en 3.3.1. A mentionner aussi IAS [K. Plasser, 1978] qui offre les services de ce type de logiciels.

- Gestion de fichiers de séries chronologiques : les logiciels de ce type sont nombreux. Ils répondent à un besoin ponctuel et gèrent des petits fichiers de séries chronologiques (quelques centaines de séries) qui alimentent de petits modèles économétriques. L'utilisation de ces logiciels décroît du fait de la taille croissante des modèles macroéconomiques. BANK [M. Norman, 1977], DATABANK [M.C. McCracken et Al, 1973], HASH [T. Harrison et Al, 1977] et APACHE [Groupe de travail Apache-Xing, 1978] sont des logiciels de cette classe.

- Exploitation de banques de données financières : le système SCARABEE [Journées banques de données, 1971] développé par le Centre d'Enseignement Supérieur des Affaires (CESA) est un logiciel de ce type.

### 3.2. Principaux aspects

Tous ces logiciels économétriques gèrent des séries chronologiques ou des tableaux statistiques (un tableau statistique est un ensemble de valeurs où le critère temps n'intervient pas forcément). La plupart des logiciels offrent un langage orienté vers l'utilisateur avec parfois des facilités intéressantes, entre autres, l'entrée en conversationnel d'une série avec des messages très explicites du système et le codage simple des valeurs manquantes et/ou aberrantes. Les possibilités du langage varient en fonction du logiciel mais elles incluent au minimum la gestion (création, modification, destruction) d'un ensemble de séries. La gestion du temps est plus ou moins sophistiquée selon les logiciels mais elle ne dépasse pas la gestion des périodicités classiques. La périodicité la plus fréquemment gérée est la

périodicité annuelle. Seuls les logiciels complets de modélisation économétrique tel TROLL, par exemple [Troll, 1974]. incluent un interface modèle économétrique - séries. Tous les autres logiciels obligent à des duplications de données importantes pour utiliser des logiciels de traitement de modèles économétriques.

La majorité des logiciels gèrent des séries chronologiques simples. Seuls ARGOS-C [ARGOS-C, 1977C] et EXPRESS [Express] permettent de gérer des séries chronologiques à plusieurs dimensions. En outre, CRONOS [Système CRONOS, 1976B], par un artifice assez complexe (codes numériques associés à la série) gère deux dimensions qui correspondent obligatoirement à deux nomenclatures internes.

La gestion d'ensembles de données personnelles est assurée par deux types de logiciels : les logiciels de gestion de fichiers de séries chronologiques et les logiciels de modélisation économétrique. Toutefois, ces systèmes ne peuvent gérer en parallèle une grosse base centrale de séries chronologique.

### 3.3. Etude des trois principaux logiciels

L'étude détaillée de tous les logiciels recensés au paragraphe 3.1 serait trop longue, aussi nous avons choisi de présenter trois de ces logiciels qui nous semblent les plus typiques du problème que nous avons à résoudre : le logiciel TROLL, le logiciel CRONOS et le logiciel ARGOS-C. L'étude du logiciel TROLL nous permettra de situer les limites d'un logiciel qui englobe pourtant la plupart des aspects dégagés dans le chapitre 1. Le logiciel CRONOS nous permettra de souligner les problèmes de gestion d'une grosse base de données statistiques, enfin, l'étude du logiciel ARGOS-C sera essentiellement axé sur ses possibilités de gestion de données multidimensionnelles.

### 3.3.1. Le logiciel TROLL

Quelques études portant sur les logiciels économétriques disponibles à l'heure actuelle placent le logiciel TROLL comme l'un des plus complets. C'est le cas de [H. Eisenpress, 1975] qui sous le titre "An ideal Econometric Computer Program" (un logiciel économétrique idéal) recense TROLL comme étant un logiciel "en vue". De la même manière, l'étude effectuée par Ruderman dans [G. Ruderman, 1978] sur les logiciels macroéconomiques disponibles aux Etats-Unis montre que TROLL est un logiciel puissant.

Ce logiciel, développé d'abord au MIT (Massachusetts Institute Technology) de 1966 à 1972, puis ensuite par le NBER (National Bureau of Economic Research) inclut des programmes sophistiqués d'économétrie [Système TROLL, 1978]. TROLL fournit notamment des commandes d'estimation, des commandes de simulation, des commandes d'édition de modèles et des commandes d'édition de données chronologiques [TROLL, 1974]. Comme caractéristiques générales, notons que TROLL gère un système de noms hiérarchisés incluant notamment le nom de l'utilisateur et le type du fichier.

Exemple : DUPONT.DATA.PIB

la variable PIB du fichier de type DATA de l'utilisateur  
DUPONT

A priori, il n'y a pas de limite quant au nombre de niveau de hiérarchie. TROLL gère 11 types de fichiers prédéfinis.

La fonction "base de données" de TROLL regroupe toutes les commandes pouvant être faites sur des fichiers de type DATA.

Le système TROLL gère le temps de manière sophistiquée : l'utilisateur n'indique que la périodicité et la date de début des données. La date de fin est déterminée automatiquement à la fin de la saisie des valeurs, de même que les dates correspondant à chaque valeur lors d'une entrée de série en conversationnel ou lors d'une impression de série. L'utilisateur peut stocker des séries de trois périodicités différentes : annuelle, trimestrielle, mensuelle.

La gestion des séries chronologiques à plusieurs dimensions n'est pas possible en TROLL. Celui-ci gère des matrices si l'on assimile la périodicité à un nombre de colonnes. Une matrice  $m,n$  est traitée comme un vecteur de périodicité  $n$  et de  $m*n$  éléments. Hormis cette possibilité, TROLL gère exclusivement des séries chronologiques simples.

La hiérarchie des noms permet à TROLL de gérer des ensembles de données personnels. Aucune limite n'étant imposée au nombre de séries que peut contenir un fichier, la gestion parallèle d'une grosse base centrale de séries chronologiques peut être faite à l'aide des outils fournis par TROLL.

L'interface modèles-séries est réalisé dans TROLL. La complète intégration de ce dernier facilite cet interface. TROLL est un logiciel interactif. Son excessive verbosité lui est parfois reprochée [Instruments de stratégies économiques, 1977]. Toutefois, elle permet, alliée au format libre des commandes, au système de documentation et d'erreur, de faciliter l'apprentissage du système [Instruments de stratégies économiques, 1977]. Le langage témoigne de la facilité d'utilisation de TROLL pour l'économiste : il utilise le vocabulaire de ce dernier pour désigner des paramètres ou des noms de commandes. Le système guide l'utilisateur lors de l'entrée des valeurs de séries. Enfin, à tout moment, l'utilisateur peut demander des renseignements en tapant le symbole réservé ? et éventuellement ?? si les renseignements fournis ne lui suffisent pas.

### 3.3.2. Le logiciel CRONOS

Contrairement au logiciel TROLL qui gère un ensemble de fichiers utilisateurs, le logiciel CRONOS, utilisé par l'Office Statistique des Communautés Européennes (OSCE), gère une grosse base de données statistiques. Celle-ci contient des séries chronologiques réparties en domaines (exemple : statistiques conjoncturelles, comptabilité nationale, énergie, etc.). A chaque domaine est associé un SEF (Sous-Ensemble de Fichier) pouvant contenir 150 000 séries mensuelles ou 300 000 séries annuelles. Toute série



est identifiée par un numéro à 10 chiffres. CRONOS distingue deux types de séries : les séries primaires directement fournies par l'utilisateur et les séries dérivées issues de résultats de calcul. CRONOS assure la mise à jour automatique des séries dérivées chaque fois que les séries dont elles dérivent sont mises à jour.

CRONOS assure une gestion du temps aussi évoluée que TROLL. Le nombre des périodicités définies est même plus important puisque CRONOS gère quatre périodicités : annuelle, semestrielle, trimestrielle et mensuelle.

La gestion des séries à plusieurs dimensions est complexe : d'une part, CRONOS possède deux nomenclatures internes : PAYS et UNITES. Pour toute série de la base, l'utilisateur doit indiquer le code pays et le code unité, permettant au système de lui associer des libellés pays et unité à l'édition. D'autre part, il est possible de découper l'identifiant de la série en sous-codes analytiques, sorte de nomenclatures numériques qui facilitent le classement.

Exemple : 08.5.420.2.06 (les caractères . sont utilisés pour faciliter la lecture)

Si ce numéro est relatif au commerce extérieur, on a le découpage suivant :

08	sous-code	pays	déclarant
5	"	"	classe de produits
420	"	"	pays partenaire
2	"	"	sens du flux (importations ou exportations)
06	"	"	unité de mesure

Bien que moins verbeux que TROLL, CRONOS est très facilement utilisable par l'économiste. Ce dernier a à sa disposition plusieurs langages adaptés à ses besoins : langage de gestion, langage de consultation [Système CRONOS, 1976B], langage de calcul [Système CRONOS, 1976C] auquel est associée une bibliothèque très riche de fonctions de calcul [Système CRONOS, 1976D], [Système CRONOS, 1976E].

CRONOS ne permet pas la gestion de bases personnelles. En outre, ce n'est pas un logiciel de modélisation économétrique, l'interface modèles-séries est inexistant.

### 3.3.3. Le logiciel ARGOS-C

Pour décrire les caractéristiques principales du logiciel ARGOS-C, nous nous appuierons sur les brochures [ARGOS-C, 1977C], [ARGOS-C, 1977B], [ARGOS-C, 1977].

ARGOS-C est un logiciel développé par l'INSEE (Institut National de la Statistique et des Etudes Economiques). Il est destiné au traitement des données agrégées telles que tableaux statistiques ou séries chronologiques.

L'entité manipulée par ARGOS est le Tableau caractérisé par son nom et un ou plusieurs critères repérant les données dans le tableau. Chaque valeur du critère est appelée modalité du critère, c'est une suite de caractères numériques ou alphabétiques.

Exemple : Tableau 554B321

Critère DATE modalités	7201	7202	7203	...	7512	7601	7602
	203	203.5	202		321	334	345

Les modalités vont de 7201 à 7602 et peuvent correspondre à des dates mensuelles.

Si un tableau a plusieurs critères, toute valeur contenue dans le tableau est repérée par un ensemble de modalités comprenant une modalité de chaque critère, de façon identique aux coordonnées d'un point dans le plan ou dans l'espace. Cet ensemble de modalités repère une case du tableau. Un tableau ARGOS-C peut contenir plusieurs données par case, on a alors une subdivision en sous-cases. Les sous-cases peuvent être nommées ou non; si elles ne le sont pas, on peut les adresser par leur numéro d'ordre : SCS(n) = sous-case n°. La figure 3.1 tirée de [ARGOS-C, 1977C] donne un exemple d'un tableau à 2 critères et 2 sous-cases.

Figure 3.1 : Tableau CONSTRUC

Critères PAYS modalités

		ALL	ESP	FRA	GBR	JPN	SUE	
Critère	1965	NML 0.82	0.56	0.44	1.39	3.30	0.77	
		1.02 NMC	0.29	0.48	1.07	5.36	1.17	
ANNEE	modalités :	1974	2.05	2.30	1.71	1.93	11.16	2.25
			2.11	1.43	1.34	1.26	17.58	2.21

La présentation sommaire ci-dessus permet de faire les constatations suivantes :

- ARGOS-C gère les données multidimensionnelles à l'aide de critères et de modalités. De plus, ARGOS-C gère des listes d'intitulés communes à tous les tableaux d'un fichier. Elles permettent d'associer à chaque modalité, un intitulé en clair qui rend l'édition plus explicite.

Exemple

CODE	INTITULE
ALL	ALLEMAGNE FEDERALE
ESP	ESPAGNE
FRA	FRANCE
GBR	GRANDE BRETAGNE
JPN	JAPON
SUE	SUEDE

- La gestion du temps par ARGOS-C est rudimentaire. Le temps est une dimension comme les autres, géré à l'aide de critères et de modalités. L'utilisateur doit fournir la liste des dates du domaine temporel d'une série.

Le langage de gestion et d'interrogation d'ARGOS-C est riche : il ne comprend pas moins de 38 commandes. Bien qu'orienté vers l'économiste (il gère des données spécifiques), il est difficile à utiliser.

ARGOS-C permet la gestion d'ensembles de données personnels puisque les tableaux sont gérés dans des fichiers qui peuvent être personnalisés.

Enfin, de même que CRONOS, ARGOS-C n'est pas un logiciel de modélisation économétrique. De ce fait, l'interface modèles-séries est inexistant.

### 3.4. Insuffisances par rapport au problème posé

Pour mettre en évidence les insuffisances des trois logiciels spécifiques pour la gestion des séries chronologiques, nous dressons un tableau récapitulatif qui situe les trois logiciels par rapport aux cinq problèmes soulevés en 1.4.

Logiciels Problèmes	TROLL	CRONOS	ARGOS-C
Gestion du temps	La gestion du temps par TROLL est sophistiquée. Néanmoins, le nombre de périodicités gérées est insuffisant.	Tout comme TROLL, la gestion du temps est satisfaisante. Mais la remarque faite pour TROLL reste valable pour CRONOS : le temps n'est pas géré de manière générale, le nombre de périodicités gérées est insuffisant	La gestion du temps est rudimentaire. L'utilisateur dispose de moyens pour décrire des dimensions. A charge pour lui de donner une interprétation temporelle à l'une des dimensions
Gestion de données à plusieurs dimensions	Seule la gestion des matrices est permise moyennant la suppression de la dimension temps	Gestion de deux dimensions à l'aide de nomenclatures internes. Possibilité de découpage de l'identifiant de la série pour faciliter le classement	Possibilité de gestion de données à plusieurs dimensions. Les outils sont toutefois complexes à mettre en oeuvre
Gestion base centrale bases personnelles	Bien que n'offrant guère de manipulations globales, cette double gestion est possible en TROLL	CRONOS ne gère qu'une base centrale séries chronologiques. Pas de possibilité de gestion de bases personnelles	L'organisation proposée par ARGOS-C permet cette double gestion
Interface modèles-séries	oui	non	non
Facilité d'utilisation pour l'économiste	TROLL est, parmi les trois logiciels, le plus simple à utiliser. Il offre le plus de facilités pour l'économiste	Bien que n'offrant pas de facilités interactives particulières, CRONOS offre plusieurs langages très adaptés aux besoins de l'économiste	ARGOS-C offre un langage riche mais difficile à mettre en oeuvre

### 3.5. Pour la construction d'un logiciel spécifique : MODULECO

Outre leurs insuffisances par rapport à la gestion de séries chronologiques, les logiciels de modélisation économétrique existants ne permettent pas, ou très difficilement, de traiter de gros modèles, ce qui est pourtant la tendance actuelle de la modélisation macroéconomique. De plus, tous ces logiciels, à l'exception du système XING [Groupe de travail Apache-Xing, 1978] dont l'usage est réservé aux petits modèles, ont été construits à l'étranger. Ces systèmes sont loués à leur constructeur avec impossibilité voire interdiction de les modifier. De plus, aucun système existant n'a été conçu pour fonctionner sur du matériel informatique français.

De manière générale, il n'existe pas actuellement sur le marché international de système répondant aux cinq critères essentiels ci-dessous [P. Népomiatshy et Al, 1972]:

- transportabilité d'un ordinateur à un autre
- possibilité d'adaptation aux besoins spécifiques de l'utilisateur
- facilité d'utilisation pour un non informaticien
- structure souple du système permettant le remplacement aisé d'un algorithme ou module le composant par un autre plus récent, afin de pouvoir suivre le progrès scientifique
- intégration qui permet le passage aisé d'une tâche à l'autre

Le logiciel MODULECO (MODULEs pour ECONomistes) [F. Rechenmann et Al, 1978], [A. Tarabout et Al, 1980], [P. Népomiatshy et Al, 1978] dans lequel se situe notre étude, répond à ces cinq critères. Nous en rappelons les principaux aspects dans le chapitre suivant.

## CHAPITRE IV



## 4. STRUCTURE GENERALE DE MODULECO

Le projet MODULECO [B. Oudet et Al, 1981], [Rapport final, 1978] rassemble des compétences pluridisciplinaires : économistes, mathématiciens, informaticiens. Financé par des organismes tels que l'INSEE, la Direction de la Prévision et l'INRIA, il doit aboutir à la réalisation d'un logiciel de modélisation macroéconomique.

Après un rappel des objectifs de ce logiciel, nous en présentons brièvement les différentes composantes. Enfin, les interactions essentielles entre le sous-système de gestion de séries et les autres composantes de MODULECO sont exposées.

### 4.1. Rappel des objectifs

Les objectifs principaux qui ont guidé la conception du logiciel MODULECO sont la portabilité, l'extensibilité et l'utilisation simple pour l'économiste. La portabilité de MODULECO est assurée par sa division en une partie commune portable et une partie propre à chaque type d'ordinateur hôte. Cette dernière partie, réduite au minimum, permettra de réaliser des installations à un coût faible. Par ailleurs, MODULECO offre à l'utilisateur la possibilité d'ajouter ses propres programmes à la bibliothèque, lui permettant d'adapter le système à ses besoins spécifiques, bien que la bibliothèque de la version définitive soit constituée par un jeu important d'algorithmes.

Enfin, de nombreuses facilités sont offertes à l'utilisateur pour lui permettre de se familiariser rapidement avec le logiciel. Par exemple, grâce à la documentation interactive, l'utilisateur débutant peut utiliser le système après avoir lu seulement quelques pages d'introduction.



## 4.2. Les différentes composantes du logiciel MODULECO

Le système MODULECO comprend les utilitaires système ou interface système hôte, le sous-système de gestion des objets de base, le sous-système d'entrées-sorties, le gestionnaire de commandes, le langage MODULECO et les environnements d'utilisation.

Les utilitaires systèmes ou interface système hôte [R. Gardien et Al, 1980], [R. Gardien et Al, 1981] constituent la partie non portable du système. Les utilitaires systèmes pour l'implémentation sur HB68 ont été réalisés par Noëlle Vitry (C.I.C.G.).

Le sous-système de gestion des objets de base [A. Tarabout, 1979], [A. Tarabout, 1980], [P. Lespinasse, 1981] prend en charge les objets de MODULECO : les annuaires, les segments et les liens. Les annuaires répertorient les objets de MODULECO. Les segments sont des parties de fichier formant un tout pour l'utilisateur : par exemple tous les segments contenant ses séries. Enfin, les liens permettent à l'utilisateur d'accéder en consultation seulement, aux segments et annuaires d'autres utilisateurs. Alain Tarabout (INRIA) puis Pascale Hamoniaux (Club MODULECO) se sont occupés de cette partie de MODULECO.

Le sous-système d'entrées-sorties [R. Gardien et Al, 1980B], [R. Gardien 1981], réalisé par Robert Gardien (Cap Sogeti Logiciel), permet d'offrir aux autres fonctions de MODULECO, plusieurs organisations de données et divers modes d'accès. Il pallie les insuffisances du langage de programmation choisi, dans ce domaine.

Le gestionnaire de commandes [P. Lespinasse et Al, 1981], pris en charge par Pascale Hamoniaux et Robert Gardien, gère la bibliothèque des commandes de MODULECO. Il offre des outils permettant la création, la mise à jour et la destruction de commandes. De plus, il met à la disposition des utilisateurs un éditeur leur permettant d'écrire ou de corriger un programme en Pascal, Fortran ou en langage MODULECO avant de les cataloguer. Enfin, quel que soit l'environnement d'utilisation, les appels de commande vers cet environnement sont analysés par le gestionnaire de commandes.

Le langage MODULECO [F. Rechenmann et Al, 1980], [R. Gardien et Al, 1981] est un langage algorithmique qui permet de programmer de nouvelles commandes en combinant des commandes existantes. Il est réalisé par Robert Gardien et Pascale Hamoniaux. Bruno Oudet (USMG) développe un interface dans ce langage pour aider les utilisateurs débutants. En outre, il se charge de la rédaction des manuels utilisateurs. Les environnements d'utilisation sont au nombre de cinq. On distingue le sous-système de gestion de séries que nous présentons dans le chapitre 5, l'environnement écriture de modèles économétriques, le sous-système d'estimation, le sous-système de simulation et le sous-système d'optimisation.

Pour entrer le texte des équations d'un modèle, l'économiste dispose d'un langage particulier : le Langage Externe d'écriture des Equations ou LEEE [P. Népomiatshy, 1979], [P. Népomiatshy et Al, 1981]. Ce langage, aussi proche que possible de celui des économistes, facilite la construction d'un modèle et son traitement par le système (systématisation des notations). Il permet de décrire des modèles complexes sans pénaliser la description de modèles simples (seul un sous-ensemble réduit du langage est utilisé). Il traite les variables multidimensionnelles à l'aide de la déclaration d'ensembles :

Exemple : ENSEMBLE PAYS = (FRANCE, ITALIE, USA)  
EXO PNB(PAYS)

Par la suite, on peut utiliser PNB(ITALIE) dans le modèle.

Ce langage permet de décrire des manipulations complexes sur les séries chronologiques et constitue un outil de calcul puissant. L'analyse, la traduction et l'interprétation de ce langage sont réalisés par François Rechenmann (INRIA) et Danielle Ettinger (Club MODULECO).

Pour mettre au point un modèle écrit en LEEE, l'utilisateur dispose d'un éditeur de texte MODULECO qu'il peut appeler à l'aide de la commande EDITMOD [F. Rechenmann, 1980C].

Le sous-système d'estimation [B. Pincemaille, 1979C], [F. Rechenmann, 1979B], [J.P. Grandjean et Al, 1980], réalisé par Frank Sylvain (INRIA), sera muni d'un ensemble d'algorithmes classiques. L'estimation se fait à partir du texte en LEEE d'une équation ou d'une liste d'équations à l'aide de la commande ESTIM. L'opération consiste à obtenir les coefficients du modèle économétrique à partir des séries chronologiques alimentant celui-ci.

Le sous-système de simulation [F. Rechenmann, 1979], [F. Rechenmann, 1980B], [H. Jayet, 1980B], particulièrement étudié dans MODULECO, met en oeuvre, outre les modèles économétriques, tous les objets qui leur sont rattachés. La manipulation de ces objets est grandement facilitée, aussi bien pour des simulations simples que pour des simulations très complexes [B. Pincemaille, 1980],[H. Jayet, 1980]. La simulation permet d'obtenir, par résolution des équations du modèle, les valeurs des variables endogènes pour des valeurs données des variables d'entrée (séries chronologiques, coefficients, etc.) [B. Oudet et Al, 1981]. Enfin, il faut noter qu'une simulation étant effectuée période par période, le stockage des séries chronologiques, tel qu'il est fait par le sous-système de gestion de séries, devient inefficace notamment dans le cas de gros modèles et/ou si l'on effectue plusieurs simulations successives. Les algorithmes travaillent alors sur des segments transposés dans lesquels le stockage est fait période par période, chaque période regroupant les observations de toutes les séries. La transposition est automatique par l'appel de la commande SIMULER ou commandée par TRANSPOSER.

Le sous-système de simulation est pris en charge par François Rechenmann et Danielle Ettinger, de plus, Dominique Granier (Club MODULECO) réalise les commandes permettant d'imprimer les résultats d'une simulation.

Le sous-système d'optimisation [P. Népomiashtchy, 1981] permet d'inclure à MODULECO des programmes d'optimisation capables de traiter des modèles de grande taille. Les commandes de ce sous-système sont définies et réalisées par Pierre Népomiashtchy (INRIA).

### 4.3. Interface avec le sous-système de gestion de séries chronologiques

A l'aide du schéma ci-dessous, nous exposons les interfaces entre le sous-système de gestion de séries et les autres composantes de MODULECO.

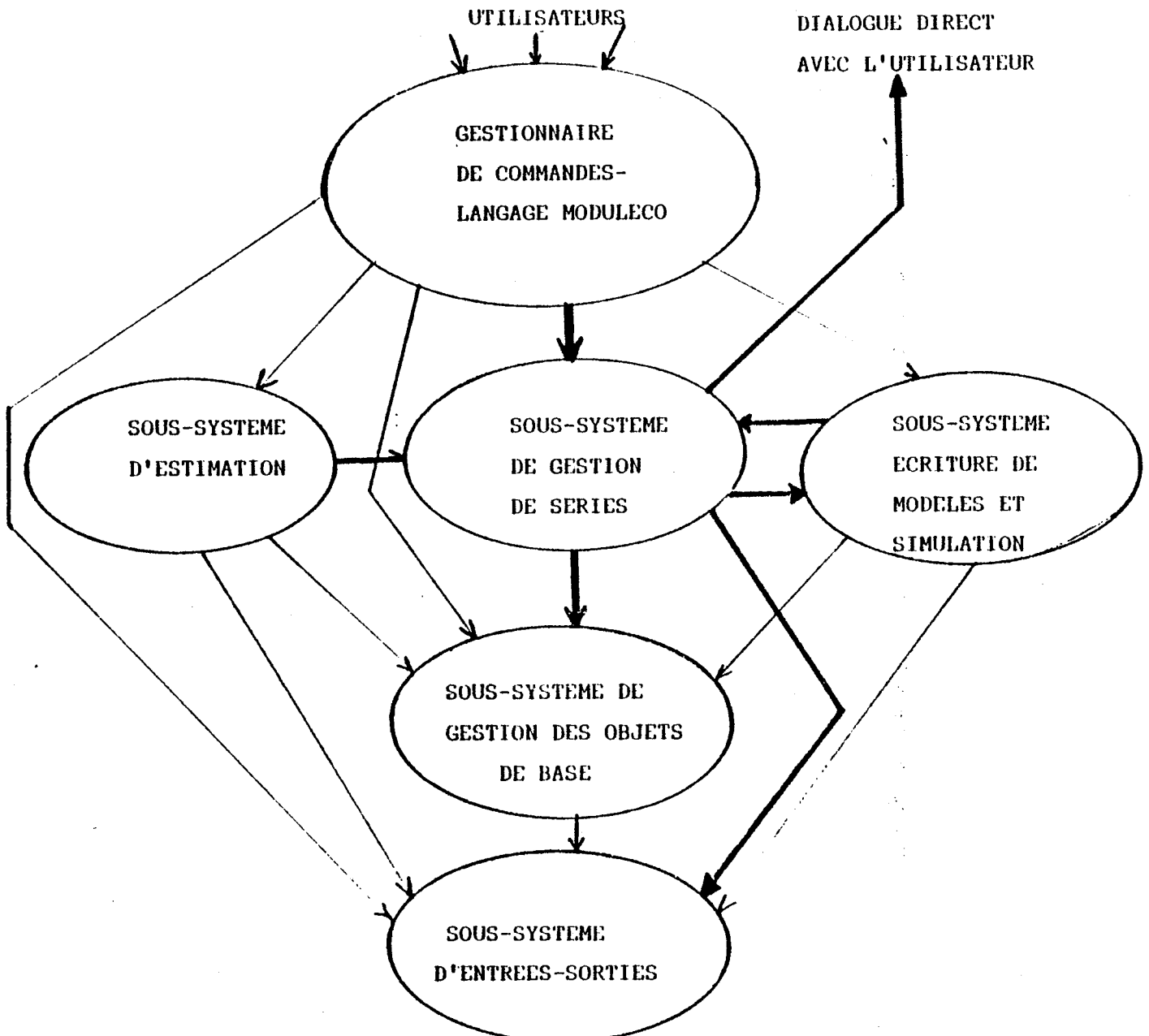


Figure 4.1 : Interface "sous-système de gestion de séries - autres fonctions de MODULECO"

Le gestionnaire de commandes analyse les commandes du sous-système de gestion de séries puis appelle ce dernier en lui transmettant un message contenant tous les paramètres analysés. Celui-ci effectue la tâche demandée, initialise éventuellement un dialogue avec l'utilisateur notamment lors de la saisie des valeurs de série puis appelle le sous-système de gestion des objets de base et le sous-système d'entrées-sorties pour extraire ou stocker les objets référencés par l'utilisateur.

Le sous-système de gestion de séries met à la disposition du sous-système d'estimation des fonctions lui permettant d'alimenter un modèle économétrique en séries chronologiques. Ces mêmes fonctions sont utilisées par le sous-système d'écriture de modèles et de simulation pour alimenter un modèle en simulation. En outre, des outils permettent à ce sous-système de stocker les résultats d'un modèle économétrique.

Enfin, le sous-système de gestion de séries utilise le LEE pour décrire des manipulations ponctuelles sur les séries chronologiques.

## CHAPITRE V



## 5. LE SOUS-SYSTEME DE GESTION DE SERIES CHRONOLOGIQUES

Pour présenter le sous-système de gestion de séries chronologiques, nous exposons d'abord les solutions apportées aux problèmes soulevés en 1.4, ensuite nous détaillons la structure du sous-système à travers les choix d'organisation des données et les traitements que peut effectuer l'utilisateur sur ces données.

### 5.1. Les solutions apportées aux problèmes de gestion des séries chronologiques

Le logiciel que nous proposons [B. Sefsaf, 1980] pour résoudre les problèmes de gestion de séries chronologiques est un logiciel spécifique. Il privilégie un type de donnée particulier en offrant des mécanismes permettant de gérer au mieux les données chronologiques à plusieurs dimensions. Il favorise une classe d'utilisateurs bien précise par un formalisme qui s'inspire du langage économique. Cette démarche a facilité la phase de conception qui s'est faite en étroite collaboration avec les futurs utilisateurs du logiciel. Enfin, le sous-système de gestion de séries est fortement intégré au logiciel MODULECO (cf. 4.3). Il manipule des séries chronologiques. Celles-ci sont caractérisées par un nom qui permet de les citer et un domaine de définition qui permet de repérer les données à l'intérieur de la série.

Le domaine de définition se décompose en un domaine temporel et un domaine constitué par les autres dimensions d'une série.

#### 5.1.1. Gestion simple de données à plusieurs dimensions

Pour gérer des données à plusieurs dimension, l'utilisateur dispose de deux moyens. Il gère ses données comme des tableaux classiques à la composante temporelle près : il lui suffit alors d'indiquer à la création de ses données, la taille maximum de chaque dimension.

Exemple : SERIE IMPORT DIMENSION = 10,10

La série IMPORT est une série à deux dimensions composée de  
100 séries simples : IMPORT(1,1) à IMPORT(10,10)



L'utilisateur gère ses données à plusieurs dimensions à l'aide de nomenclatures.

Une nomenclature désigne, dans la terminologie économique, un ensemble de rubriques mutuellement exclusives permettant de classer des objets d'un genre déterminé [Ph. Richard et Al, 1979].

Les nomenclatures jouent un rôle important dans la manipulation de données économiques au point qu'elles ont donné lieu au développement de systèmes de gestion de nomenclatures [SABINE].

Le rôle assigné aux nomenclatures dans le sous-système de gestion est plus restreint. Elles désignent des tables de libellés mutuellement exclusifs permettant de gérer de manière claire les séries à plusieurs dimensions. La figure 5.1 résume l'utilisation actuelle des nomenclatures dans le logiciel.

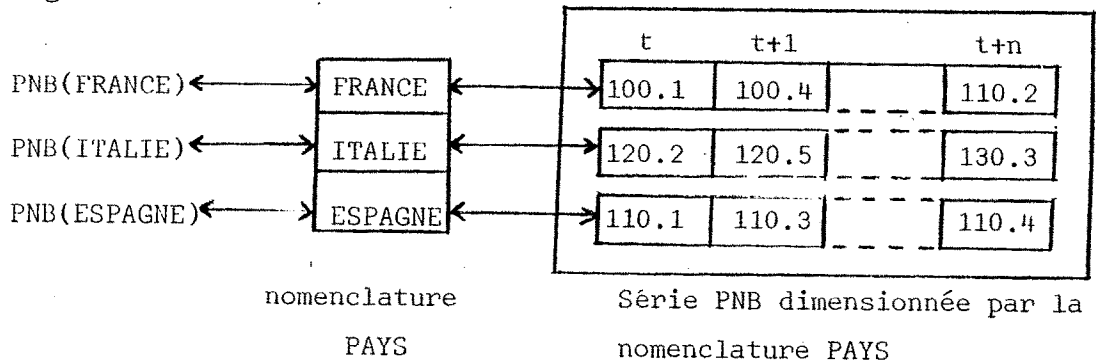


Figure 5.1 : Rôle d'une nomenclature dans le logiciel

Le fléchage à double sens indique, d'une part qu'à partir d'un identifiant de série (PNB(FRANCE) par exemple), le logiciel restitue ou repère les valeurs de la sous-série correspondante; d'autre part, lors de l'entrée des valeurs le logiciel guide l'utilisateur en lui restituant les identifiants successifs des séries pour lesquelles il veut donner les valeurs.

L'utilisateur peut créer et détruire des nomenclatures. Il définit les associations séries-nomenclatures dans les commandes de création de séries : une série est dimensionnée par plusieurs nomenclatures (SERIE IMPORT DIMENSION = PAYS, PRODUIT), plusieurs séries peuvent être dimensionnées par la même nomenclature (SERIE IMPORT DIMENSION = PAYS, PRODUIT, SERIE EXPORT DIMENSION = PAYS, PRODUIT).

### 5.1.2. Gestion élaborée de la dimension temps

Pour offrir un compromis satisfaisant entre la simplicité de gestion et la possibilité de définir des domaines temporels quelconques, le logiciel de gestion de séries fournit à l'utilisateur deux types de calendriers. Les calendriers classiques ne nécessitent de l'utilisateur que la donnée du triplet  $(p, dd, df)$  et sont gérés par un algorithme interne. Dans la version actuelle, cinq calendriers classiques sont mis à la disposition de l'utilisateur et correspondent aux périodicités classiques : annuelle, trimestrielle, mensuelle, hebdomadaire et journalière. L'utilisateur les désigne à l'aide de cinq noms réservés : ANNUEL, TRIMESTRIEL, MENSUEL, HEBDOMADAIRE, QUOTIDIEN.

Les calendriers quelconques sont créés par l'utilisateur. Au lieu de lui imposer la donnée de toutes les dates  $d_1 \dots d_n$  composant le calendrier, le logiciel lui permet de le définir comme un produit cartésien d'ensembles.

Soit  $E_1, E_2, \dots, E_n$  n ensembles

avec  $E_1 = (a_1, a_2, \dots, a_m)$

$E_2 = (b_1, b_2, \dots, b_m)$

$\vdots$

$E_n = (z_1, z_2, \dots, z_m)$

la périodicité  $p$  de ces calendriers est déterminée par l'ordre de croisement des ensembles  $E_1, \dots, E_n$ . Si  $p = E_1/E_2/\dots/E_n$

la date début correspondant à  $p$  est  $dd = a_1/b_1/\dots/z_1$

la date de fin est  $df = a_m/b_m/\dots/z_m$

$date_i = a_i/b_i/\dots/z_i$

Chaque ensemble constituant le calendrier peut être une liste d'entiers, une liste de libellés ou un intervalle.

Nous donnons ci-dessous un exemple de calendrier quelconque (fig. 5.2).

JOUR = 1 A 30  
 MOIS = JANVIER, FEVRIER, MARS, AVRIL, MAI, JUIN  
 ANNEE = 1971, 1973, 1974, 1976, 1977  
 CALENDRIER CALDEF PERIODE = JOUR/MOIS/ANNEE

Le calendrier CALDEF composé des ensembles ANNEE, MOIS et JOUR a la périodicité que définit le produit cartésien des ensembles JOUR, MOIS et ANNEE dans cet ordre.

Il représente une plage de dates allant de 1/JANVIER/1970 au 30/JUIN/1977.

Figure 5.2. : Un exemple de calendrier défini

Une fois le calendrier créé ou implicitement si le calendrier est classique, le logiciel en assure la gestion. Il repère ou restitue une observation à partir de l'identifiant de la série et de la date de l'observation, il guide éventuellement l'utilisateur lors de la saisie des valeurs en lui restituant les dates successives pour lesquelles il attend des valeurs.

Les associations séries-calendriers sont fournies par l'utilisateur à la création des séries : une série n'a qu'un calendrier (SERIE PNB CALENDRIER CALDEF) plusieurs séries peuvent avoir le même calendrier (SERIE PNB CALENDRIER CALDEF, SERIE PIB CALENDRIER CALDEF).

### 5.1.3. Gestion base centrale - segment

La solution que nous proposons permet à chaque site de disposer d'une base centrale appelée à contenir toutes les séries de référence du site et offre aux modélisateurs la possibilité de se créer autant de segments qu'ils le désirent. Nous résumons cette organisation à l'aide du schéma ci-après.

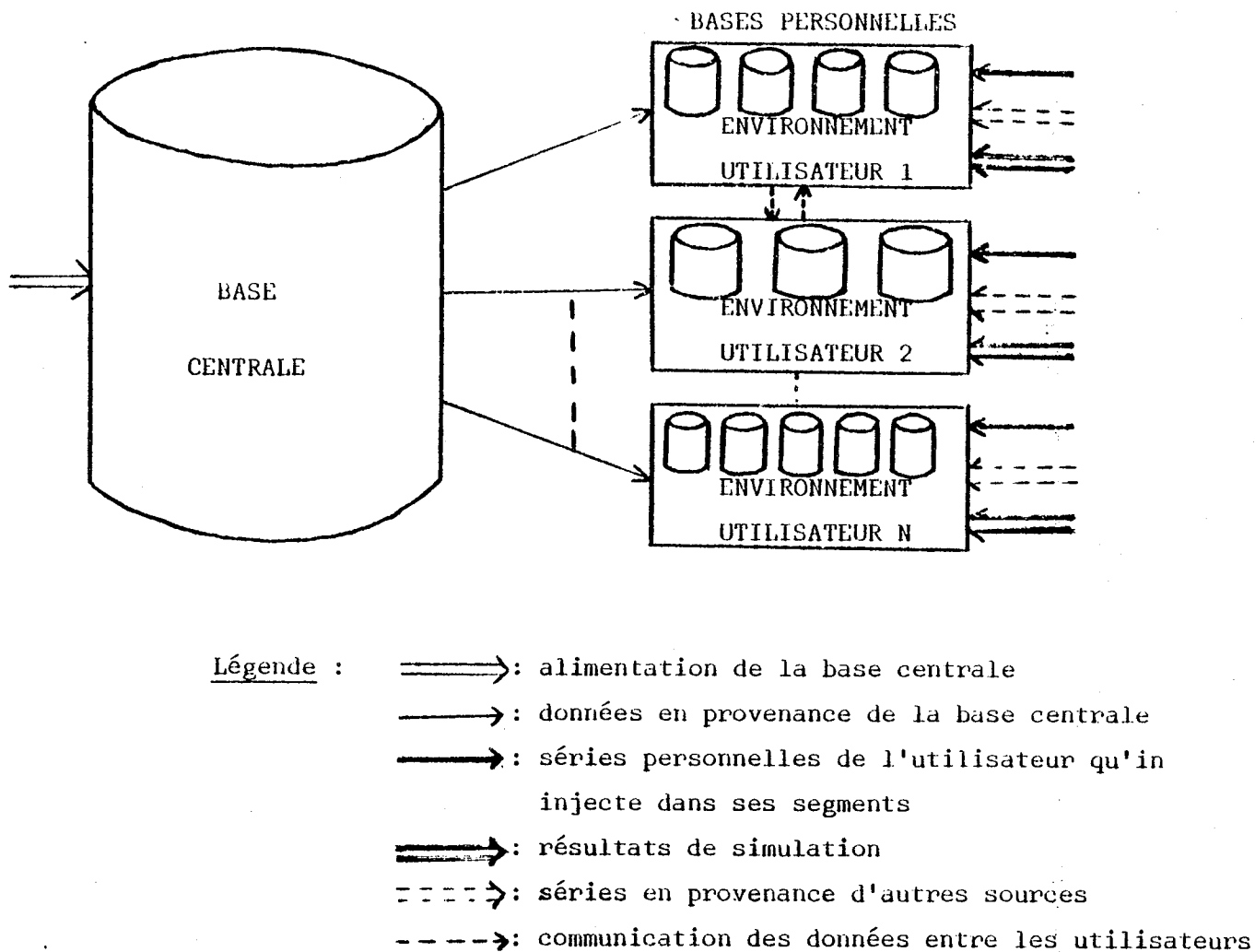


Figure 5.3 : Architecture générale de la base de données

La base centrale gérée par le gestionnaire de base de données économiques dispose de commandes à cet effet [B. Sefsaf, 1980C] et est accessible en consultation seulement aux utilisateurs du site. Une protection par mot de passe permet de filtrer les accès en consultation à certaines séries ou à certains groupes de séries. Le nombre de séries que peut contenir la base centrale est quelconque. Le nombre d'observations d'une série n'est pas limité.

Les segments de séries permettent à l'utilisateur de se constituer des ensembles de données personnels en nombre quelconque. Chaque segment contient des séries de même calendrier. La mise à jour des séries de référence de la base centrale n'influe pas sur les mêmes données présentes dans les segments. Réciproquement, les mises à jour que l'utilisateur peut être amené à faire sur une série d'un segment ne sont pas répercutées sur la même série de la base centrale.

Pour gérer et manipuler les séries des segments, l'utilisateur dispose d'un langage de traitement beaucoup plus riche que celui de la base centrale.

Bien que l'organisation globale et les traitements aient été spécifiés [B.Sefsaf, 1979], [B. Sefsaf, 1980B], [B. Sefsaf, 1980C], il convient de noter que dans la version actuelle du logiciel, seule la gestion des segments est opérationnelle. Aussi, au lieu d'alourdir la présentation en détaillant l'organisation de la base centrale et des segments, nous insisterons plus particulièrement sur ce qui a été réalisé.

#### 5.1.4. Interface "sous-système de gestion de séries - sous-système de modélisation"

Nous désignons par sous-système de modélisation, le sous-système d'estimation et le sous-système de simulation.

Le sous-système de gestion de séries offre à ces deux sous-systèmes des fonctions internes leur permettant d'extraire les séries des segments pour alimenter les modèles économétriques et de manipuler des dates.

De plus, le sous-système de simulation dispose d'autres fonctions internes pour stocker dans les segments les séries chronologiques résultats.

Enfin, au lieu de développer un outil de calcul spécifique pour effectuer les manipulations algébriques décrites en 1.3.2.1, le sous-système de gestion de séries inclut un interface permettant d'utiliser la puissance de description de calcul du LEEE pour décrire des manipulations ponctuelles sur les séries des segments. L'utilisateur utilise ces possibilités à l'aide de la commande CALCULER (cf. 5.2.5.2).

### 5.1.5. Facilité d'utilisation

Outre les facilités offertes globalement par MODULECO [B. Oudet et Al, 1981] (documentation interactive, commandes à syntaxe unique, nombreuses valeurs par défaut pour les paramètres), le sous-système de gestion de séries guide l'utilisateur pour l'entrée des valeurs de séries, des libellés de nomenclatures. Il admet des codes en clair pour valeur manquante (lettre M ou mot clé MANQ) et pour valeur aberrante (lettre A ou mot clé ABER).

L'utilisateur peut donner x valeurs identiques à l'aide de la structure répétitive /X/valeurs. La commande DEFBASE (cf. 5.2.5.1) permet à l'utilisateur de donner une valeur à de nombreux paramètres, lui évitant lors d'une session de redonner ces paramètres dans chaque commande.

Enfin, l'utilisateur programmant en langage MODULECO peut manipuler facilement tous les objets du sous-système de gestion de séries à l'aide de fonctions MODULECO [F. Rechenmann, 1980] , [B. Sefsaf, 1980C].

## 5.2. Présentation générale du sous-système

Cette présentation du logiciel met en évidence l'organisation des données qui a été retenue ainsi que les solutions qui ont été adoptées pour résoudre les problèmes classiques des SGBD (protection, cohérence, conflits d'accès). En outre, les commandes de manipulation réalisées sont présentées brièvement.

### 5.2.1. Organisation des données

Les segments sont les supports d'information offerts par le sous-système de gestion des objets de base [P. Lespinasse, 1981] pour stocker toutes les données de MODULECO. Un segment, décrit par son descripteur, repère une partie de fichier. Il se subdivise en membres (7 au maximum) qui peuvent contenir des informations de longueur et d'organisation différente [A. Tarabout, 1980], [P. Lespinasse, 1981].

La base centrale et les segments de séries utilisateurs sont réalisés à partir des segments de base. Un segment documentaire global géré par le sous-système de gestion des objets de base répertorie toute la documentation associée aux données [P. Lespinasse, 1981]. La figure 5.4 ci-dessous résume l'organisation globale de la base centrale.

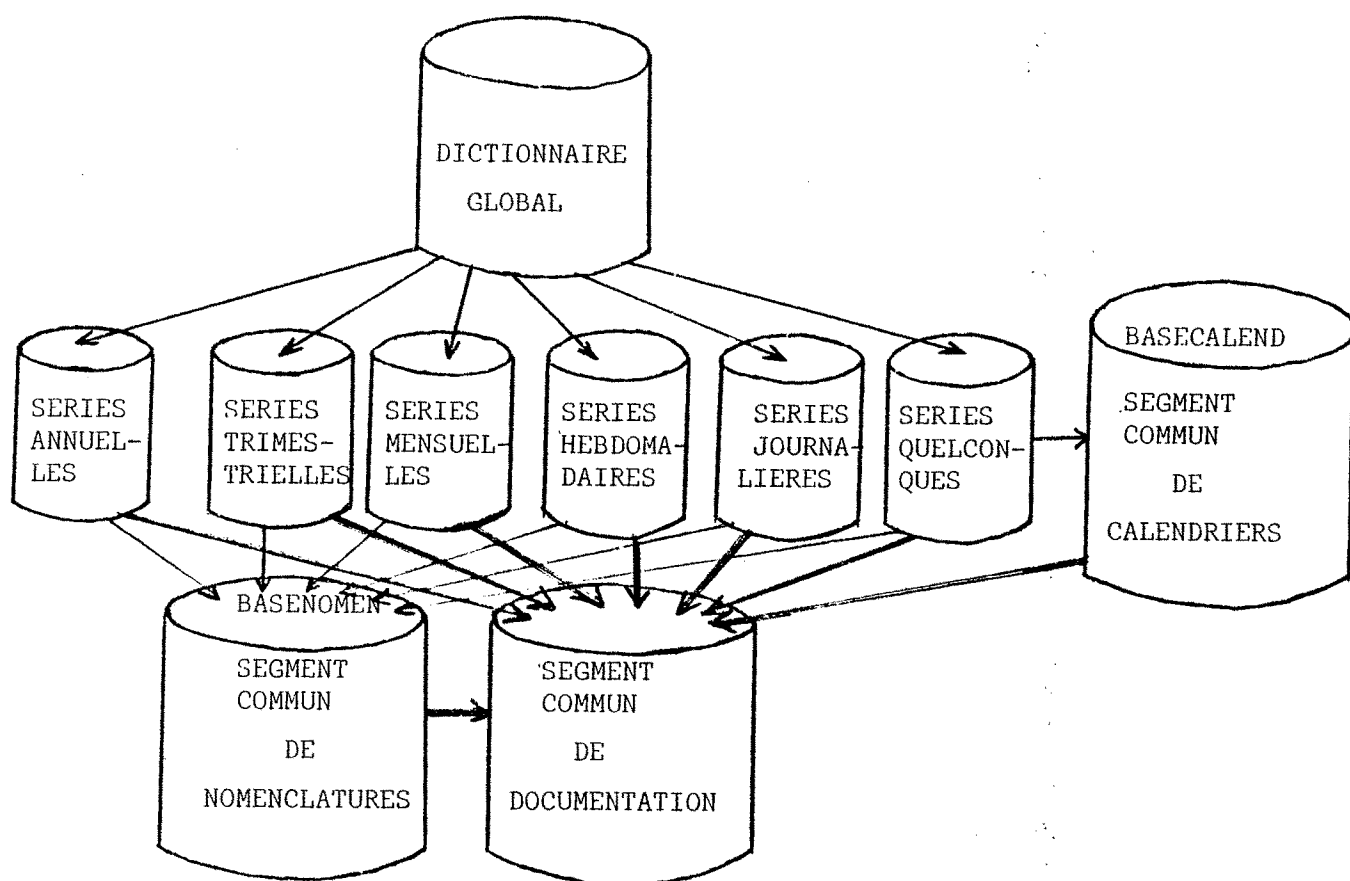


Figure 5.4 : Organisation générale de la base centrale

En plus du segment documentaire, 9 segments initiaux permettent de stocker les données de la base centrale.

Un segment dictionnaire répertorie et permet de retrouver toutes les séries de la base centrale. De même que les nomenclatures, tous les calendriers quelconques sont stockés dans un même segment. Enfin, les séries à calendriers classiques sont stockées dans des segments distincts selon leur

périodicité et toutes les séries à calendriers quelconques sont réunies dans un même segment.

L'organisation des segments de séries utilisateurs s'inspire globalement de la même organisation que la base centrale. Un dictionnaire constitué par des annuaires que gère le sous-système des objets de base recense tous les segments appartenant à un même utilisateur. Les calendriers quelconques et les nomenclatures sont stockés dans les segments SEGCALEND et SEGNOMEN. Seule différence avec la base centrale, un segment utilisateur contient des séries de même calendrier. La figure 5.5 résume cette organisation globale des segments utilisateurs.

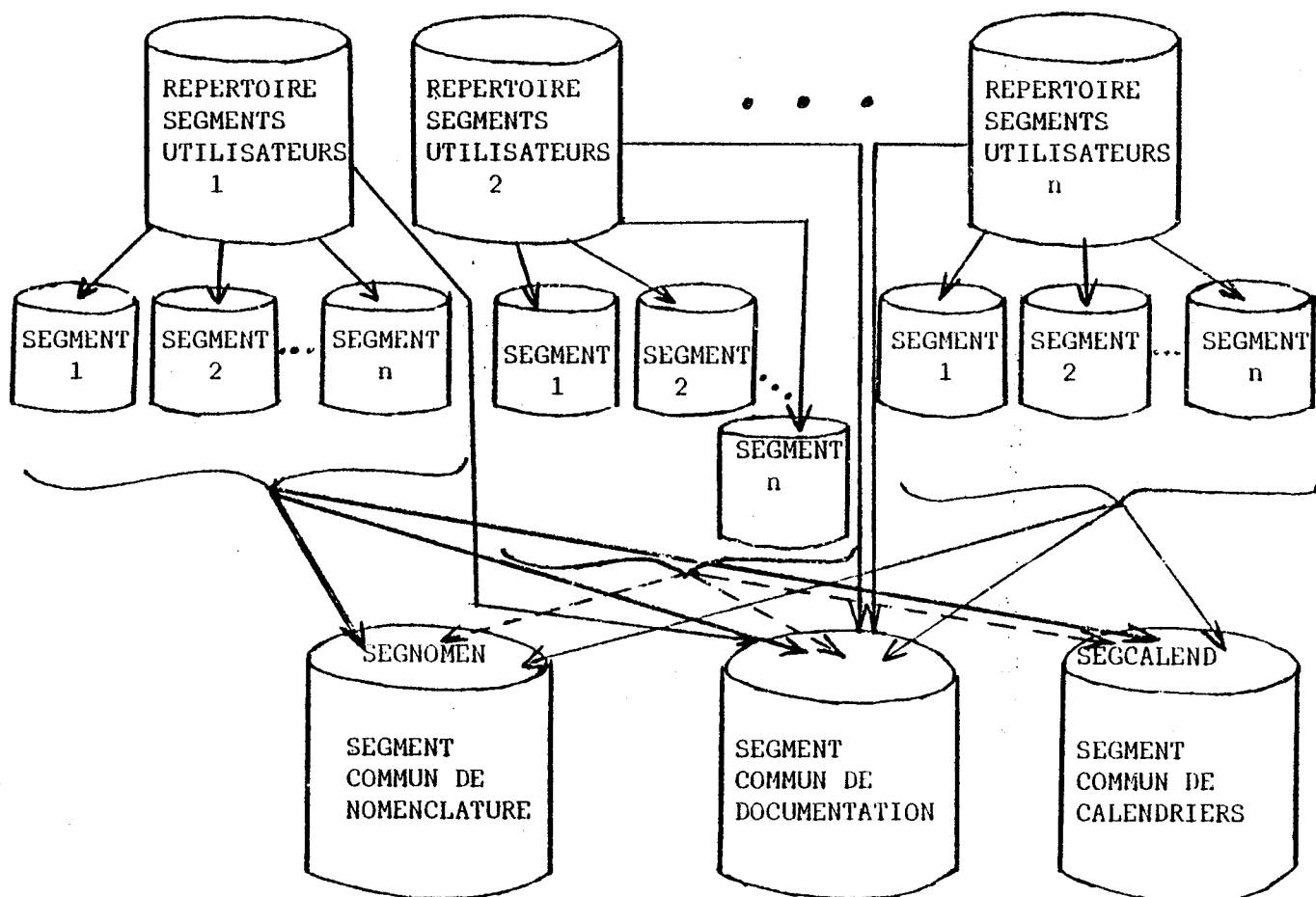


Figure 5.5 - Organisation générale des segments utilisateurs



Le descripteur d'un segment contient, outre des informations de gestion utilisées par le sous-système de gestion des objets de base [A. Tarabout, 1980], les renseignements nécessaires à la gestion du calendrier associé aux séries du segment.

Chaque segment de série utilisateur se subdivise en trois membres distincts :

- un membre catalogue contient les informations descriptives de chaque série;
- un membre valeur contient les valeurs de chaque série;
- enfin, le troisième membre contient tous les noms des nomenclatures utilisées dans le segment.

Le segment de nomenclatures SEGNOMEN est constitué de deux membres :

- un membre catalogue contient tous les descripteurs de nomenclatures,
- un membre valeur contient les libellés de la nomenclature.

Enfin, le segment SEGCALEND a deux membres :

- un membre catalogue répertorie tous les descripteurs de calendriers quelconques,
- un membre valeur contient la description des ensembles de chaque calendrier.

Quel que soit l'objet (série, calendrier, nomenclature), son descripteur contient :

- . le nom de l'objet
- . un numéro d'enregistrement permettant de retrouver la documentation
- . des informations permettant de retrouver les valeurs dans le membre valeur
- . la taille de l'objet (nombre de séries simples si c'est un descripteur de séries, nombre de libellés si c'est une nomenclature, nombre d'ensembles si c'est un calendrier).

Le reste des informations est différent selon l'objet.

Si c'est une série, le descripteur contient :

- . la date de dernière mise à jour
- . les informations permettant de gérer les dimensions de la série
- . le type de la série : la série est primaire si elle est issue directement de sources statistiques, elle est dérivée si elle provient de résultats de calcul
- . un code de validité permettant d'avertir les utilisateurs de mises à jour importantes ayant eu lieu sur la série.

Si c'est une nomenclature ou un calendrier, un compteur de suppression rangé dans le descripteur permet de contrôler la validité de l'opération de destruction de l'un de ces objets.

### 5.2.2. Protection des données

La protection des données est un problème complexe et la désignation d'un système de protection parfait s'avère impossible [F.M. Bancilhon et Al, 1977]. Pour présenter les solutions adoptées dans la version actuelle du logiciel, nous répondrons aux trois questions que posent Bancilhon et Spyrtos dans [F.M. Bancilhon et Al, 1977] :

Contre qui protéger l'information ?

Quelle information protéger ?

Comment protéger cette information ?

Pour l'ensemble de MODULECO, les données vont être protégées contre la malveillance ou les erreurs de l'utilisateur et contre les problèmes systèmes. L'unité d'information protégée est le segment. Le système de protection est mis en oeuvre par l'interface système hôte [R. Gardien et Al, 1981].

Pour protéger les segments contre la malveillance ou les erreurs de l'utilisateur, MODULECO interdit à celui-ci d'écrire sur un segment qui ne lui appartient pas. Tous les segments d'un utilisateur donné sont inaccessibles en écriture à tous les autres utilisateurs. En outre, l'utilisateur peut restreindre les droits en consultation sur ces segments en associant à chacun d'eux une liste des personnes autorisées à consulter ce segment. Tout utilisateur ne figurant pas sur la liste ne peut consulter le segment auquel se rapporte cette liste.

La protection contre les problèmes système permet de manière générale d'assurer une "reprise à chaud" qui consiste à restaurer immédiatement après toute défaillance du système un état cohérent des données, et une "reprise à froid" qui consiste à effectuer des sauvegardes régulières et à reprendre, après une panne prolongée du système, une version cohérente des données sauvegardées antérieurement.

Si aucun mécanisme interne à MODULECO n'a été prévu pour assurer les reprises à froid (la récupération se fera à partir des sauvegardes qu'effectue le système hôte), la reprise à chaud est assurée par la gestion de segments "temporaires" [N. Vitry, 1981] : à chaque ouverture d'un segment en écriture, celui-ci est recopié dans un segment "temporaire" sur lequel seront faites toutes les écritures ultérieures. En fin de commande, il y a recopie dans le segment permanent. Si un problème système intervient en cours de commande, l'utilisateur est averti et retrouve un état cohérent de ses segments : l'état précédant l'appel de la commande.

### 5.2.3. Cohérence des données

Outre la cohérence assurée implicitement par le système de protection adoptée : l'interdiction d'écriture sur des données n'appartenant pas à l'utilisateur évite des mises à jour intempestives, le sous-système de gestion de séries assure la cohérence des données de diverses manières. La destruction d'un calendrier ou d'une nomenclature est interdite tant que des séries sont encore définies suivant ces objets. Par ailleurs, aucune mise à jour n'est autorisée sur les calendriers et les nomenclatures. Enfin, la mise à jour automatique des séries dérivées qui consiste, lors de toute mise à jour d'une série primaire à mettre à jour toutes les séries dérivées qui en dépendent, doit être implémentée très prochainement et permettra de renforcer cette cohérence.

### 5.2.4. Conflits d'accès

La résolution des conflits d'accès est assurée, pour l'ensemble de MODULECO, par l'interface système hôte [R. Gardien et Al, 1981].

Les conflits sont réglés de manière globale au niveau des segments. L'opération d'écriture est "bloquante" : toute écriture sur un segment bloque automatiquement toutes les opérations qui suivent sur ce segment (que ce soit des lectures ou des écritures).

L'opération de lecture est "sélective" : les opérations de lecture qui suivent sont admises, les opérations d'écriture sont mises en file d'attente.

### 5.2.5. Traitements sur les données

Nous ne présenterons dans les traitements ni les fonctions MODULECO que nous décrivons en annexe, ni les fonctions internes. Seules les commandes disponibles pour l'utilisateur seront étudiées. En outre, nous n'alourdirons pas la présentation en présentant à la fois les commandes de la base centrale et les commandes des segments. Seules ces dernières seront décrites.

#### 5.2.5.1. Gestion

Pour initialiser un contexte de travail, l'utilisateur dispose de la commande DEFBASE qui lui permet de fixer une valeur à de nombreux paramètres de travail (identificateur de segment, date de début, date de fin, ...). Toute nouvelle commande DEFBASE en cours de session écrase les anciennes valeurs des paramètres par celles de la commande. Tout paramètre non cité dans la commande conserve son ancienne valeur. Enfin, la commande DEFBASE permet à l'utilisateur d'imprimer son contexte de travail à tout moment.

Avant de créer des séries dans un segment, l'utilisateur initialise le segment à l'aide de la commande INITSEG en lui passant le nom du segment et les informations relatives au calendrier (triplet (p,dd,df) si c'est un calendrier classique, nom du calendrier si c'est un calendrier quelconque). Si le calendrier est quelconque, il le crée à l'aide de trois commandes CRCAL, CRENT et CRLIB.

CRCAL permet de créer le descripteur de calendrier : l'utilisateur fournit le nom du calendrier et le nom des ensembles du calendrier.

CRENT permet d'entrer la composition de tous les ensembles d'entiers du calendrier, enfin CRLIB permet à l'utilisateur de donner la composition des ensembles de libellés du calendrier.

Les nomenclatures sont créées à l'aide de la commande CRNOM qui admet le nom, la dimension et la description de la nomenclature. Si la description de la nomenclature n'est pas entrée entièrement, l'utilisateur reçoit le message :

LIBELLE DE LA NOMENCLATURE nom NUMERO xxx :

?

Il peut alors donner les libellés de sa nomenclature un par un.

Pour créer des séries, l'utilisateur dispose de la commande CRSERIE à laquelle il transmet le nom de la série, sa dimension, les nomenclatures la dimensionnant ou les bornes maximum des dimensions ainsi que le nom du segment où créer la série. La saisie des valeurs peut se faire de trois manières : toutes les valeurs sont données dans la commande, aucune valeur n'est donnée dans la commande et l'utilisateur les donne en différé à l'aide d'une autre commande, l'entrée des valeurs se fait en conversationnel et le système dialogue avec l'utilisateur pour les valeurs restant à donner.

La destruction des objets gérés par le sous-système de gestion de séries se fait à l'aide de la commande DETRUIRE. L'utilisateur peut détruire des segments, des séries, des sous-ensembles de séries, des nomenclatures et des calendriers.

Dans le sous-système de gestion de séries, seules les séries chronologiques peuvent être mises à jour. L'utilisateur ne dispose pas de commandes de mise à jour de nomenclatures ou de calendriers.

Les mises à jour de séries peuvent se faire à l'aide de deux commandes. La commande MAJSER permet de modifier l'ensemble des informations d'une série (changement du nom, de la documentation, des valeurs), la modification des valeurs (mise à jour, invalidation) peut porter sur toute la série ou un sous-ensemble.

La commande MAJVAL permet de modifier les valeurs d'un ensemble de séries, voire de toutes les séries du segment.

#### 5.2.5.2. Manipulations

L'utilisation de la puissance de description du LEEE pour effectuer des manipulations algébriques ponctuelles sur les séries chronologiques permet à l'utilisateur de faire de nombreuses opérations [B. Sefsaf, 1979] :

- addition, soustraction, produit, division, grâce aux opérateurs classiques +, -, \*, /
- élévation à la puissance grâce à l'opérateur \*\*
- ordre SOMME : est utilisé pour toute sommation sur un indice ou sur plusieurs. Permet en particulier d'obtenir une série simple à partir d'une série à plusieurs dimensions

- opérateurs de sous-tableaux : permet d'obtenir des sous-ensembles de séries
- constructions de fonctions personnelles
- évocation de fonctions standard, prédéfinies, externes ou construites.

Pour effectuer l'une quelconque ou plusieurs de ces opérations, l'utilisateur appelle la commande CALCULER à laquelle il transmet principalement les noms des segments contenant les séries origines, l'intervalle de dates sur lequel effectuer le calcul et le texte en LEEE de l'expression de calcul. Les séries résultats sont imprimés et/ou sauvegardés dans un segment résultat.

L'échange d'informations entre utilisateurs peut se faire à l'aide de deux commandes : COPSERIE et COPSEG.

COPSERIE permet à l'utilisateur de copier une ou plusieurs séries d'un segment dans un autre. Le segment d'origine peut lui appartenir ou appartenir à un autre utilisateur qui l'autorise en consultation.

La commande COPSEG permet de copier un ou plusieurs segments dans un segment résultat. Les segments d'origines sont conservés. Si X séries ont le même nom dans la liste des segments origines, il est convenu que la série qui se retrouvera dans le segment résultat sera celle rencontrée dans le dernier segment consulté (l'ordre de consultation est fixé par l'ordre des segments de la liste transmise dans la commande). Par ailleurs, l'utilisateur peut indiquer à l'appel de COPSEG, une liste de séries qu'il ne veut pas retrouver dans le segment destinataire.

Pour imprimer les objets du sous-système de gestion de séries, l'utilisateur dispose de plusieurs commandes.

La commande IMPNOMEN lui permet d'imprimer ses nomenclatures, IMPCALEND lui imprime ses calendriers. Enfin, à l'aide de la commande IMPSERIE, il peut imprimer un sous-ensemble d'une série, une ou plusieurs séries et avec la commande IMPSEG, il peut demander l'impression de toutes les séries d'un ou plusieurs segments. IMPSERIE et IMPSEG offrent à l'utilisateur la possibilité de sortir, soit la partie documentaire des séries, soit la partie valeurs, soit l'ensemble. De plus, l'utilisateur peut indiquer, à l'appel d'IMPSEG une liste de séries qu'il ne veut pas imprimer.

## CONCLUSION

L'objet de cette étude était de présenter un logiciel permettant de gérer les données temporelles. Nous avons particulièrement insisté sur la spécificité de ces données qui nous a amené, après une étude approfondie des logiciels existants, à développer un logiciel spécifique. L'originalité de notre travail réside dans la synthèse qui a été faite des possibilités des divers logiciels spécifiques étudiés et des extensions qui y ont été apportées. Ces extensions permettent à notre logiciel de résoudre de manière satisfaisante les problèmes soulevés dans MODULECO mais aussi de dépasser son cadre de conception initial et d'offrir une solution à des problèmes similaires qui se posent dans d'autres domaines et notamment dans les sciences sociales (démographie, électologie, histoire, ...)

Au-delà des résultats présentés, il faut bien convenir que du travail reste à faire pour rendre le logiciel entièrement opérationnel. Nous avons touché du doigt quelques problèmes mais la liste n'est pas exhaustive. A terme, notre souhait est d'aboutir à une version complète cohérente du système (la version actuelle ne comprend que la gestion des segments). Par la suite, les extensions que nous aimerions voir apporter à notre travail se situent dans plusieurs directions. Parmi celles qui nous semblent les plus importantes, on peut citer notamment :

- le développement de commandes graphiques permettant de visualiser des séries chronologiques
- le développement d'un système documentaire associé à la base centrale statistique
- le développement d'un système perfectionné de gestion des nomenclatures.





## ANNEXES

## ANNEXE I - PLAN

	Page
1. PRINCIPAUX PROBLEMES D'IMPLEMENTATION - CHOIX	A1.1
1.1. Choix du langage	A1.1
1.1.1. Impératifs	A1.1
1.1.2. Les possibilités offertes	A1.2
1.1.3. La solution retenue	A1.3
1.1.3.1. Avantages	A1.3
1.1.3.2. Inconvénients	A1.3
1.1.4. Incidences sur le logiciel : un module d'entrées- sorties	A1.4
1.2. Choix d'une méthode d'accès	A1.5
1.2.1. Méthodes proposées	A1.5
1.2.2. Solution retenue	A1.7

## 1. PRINCIPAUX PROBLEMES D'IMPLEMENTATION-CHOIX

Une étape importante du développement de MODULECO a été le choix d'un langage de programmation. Nous soulignons dans ce chapitre quelques incidences de ce choix sur le logiciel de gestion de séries. Par ailleurs, le choix d'une méthode d'accès fait partie des phases importantes de réalisation du sous-système de gestion de séries. Après un rappel des principales méthodes existantes, nous présentons celle retenue en soulignant l'étape de réalisation.

### 1.1. Choix du langage

Le développement du logiciel MODULECO ayant été décidé à la suite du rapport [Rapport final, 1978], il restait à faire le choix d'un langage de programmation pour réaliser ce logiciel.

#### 1.1.1. Les possibilités offertes

Elles sont assez faibles au départ : seul le langage CPL/1 sous-ensemble de PL/1 développé et maintenu par la société CAP/SOGETI semble jouir d'un préjugé favorable auprès des divers organismes intéressés par MODULECO de par le poids industriel de CAP/SOGETI. Toutefois, les insuffisances notables de CPL/1 amènent les personnes chargées de la réalisation de MODULECO à proposer un autre langage : PASCAL [P. Maurice, 1978B], [P. Maurice, 1978]. Dès lors que ce choix est posé, il faut faire une analyse sérieuse des impératifs de réalisation et des possibilités comparées des deux langages.

### 1.1.2. Impératifs

C'est dans le rapport intitulé "Pascal pour MODULECO" [C. Brunet et Al, 1979] qu'est dressé un tableau comparatif des possibilités des deux langages face à un certain nombre d'impératifs. Ce tableau, que nous reproduisons ci-dessous (fig. A1.1) conclut à la nette supériorité de Pascal par rapport à CPL/1.

Critères de comparaison	CPL/1	PASCAL
Portabilité	**	**
Compilation séparée	**	*
Lisibilité de programme	**	***
Double et simple précision		*
Arithmétique flottante		***
Interface autres langages (FORTRAN, ASSEMBLEUR)	**	**
E/S intégrées au langage		**
Efficacité code généré		***
Allocation dynamique		***
Structure de bloc		***
Récursivité		***
Tableaux à plusieurs dimensions		***
Fiabilité détection d'erreur à la compilation	*	**
Fiabilité exécution (dépassement des bornes)		**
Maintenance (poids de la société de maintenance)	***	**

Figure A1.1 : Tableau comparatif CPL/1 - PASCAL

Seuls deux critères sont à l'avantage de CPL/1 : la compilation séparée et la maintenance. Le second critère est à l'origine de nombreuses réticences quant au choix de Pascal mais une action "Pascal" à laquelle MODULECO participe activement désigne la société CAP/SOGETI pour développer et maintenir le compilateur Pascal. Pascal est adopté comme langage support du logiciel MODULECO.

### 1.1.3. La solution retenue

Le langage choisi est globalement le Pascal standard tel qu'il a été défini par Jensen et Wirth dans [K. Jensen et Al, 1980] auquel seront apportées quelques extensions nécessaires parmi lesquelles figurent la compilation séparée et les fichiers en accès direct.

#### 1.1.3.1. Avantages

Le nombre de publications sur le langage Pascal atteste de l'intérêt croissant porté à ce langage depuis quelques années.

Nous trouvons dans [R. Fortier, 1975], [G.M. Scheider, 1979], [B.W. Ravenel, 1979] de nombreuses justifications à cet intérêt parmi lesquelles :

- Pascal possède des structures de données très évoluées
- Pascal utilise très largement la notion symbolique (scalaires, ensembles), ce qui le rend relativement indépendant de la machine
- Pascal a une syntaxe très simple
- sa traduction est simple et par là-même efficace
- il est lisible et facilite ainsi la maintenance et le transfert des programmes
- il est relativement sûr car il permet à son traducteur de détecter un maximum d'erreurs avant toute exécution.

En résumé, on ne peut être plus concis que Robert Fortier dans [R. Fortier, 1975] qui clôt ainsi la liste des avantages de Pascal : Pascal est un langage assez complet sans être trop complexe.

Ce manque de complexité est à l'origine de certaines insuffisances du langage. A travers la liste des principales insuffisances qui sont reprochées à Pascal, nous soulignons ci-dessous celles qui nous préoccupent le plus dans le cadre de notre application.

#### 1.1.3.2. Inconvénients

Bien que non exhaustive, la liste des insuffisances de Pascal dressée dans [G.M. Scheider, 1979] donne une idée des faiblesses qui sont reprochées au langage. Parmi ces faiblesses, on relève notamment :

- l'absence de tableaux à bornes dynamiques : obligation est faite de figer à la déclaration les bornes des tableaux, c'est une des contraintes qui nous a beaucoup gêné dans notre application

- impossibilité d'initialiser des variables à la déclaration
- absence de l'opérateur d'exponentiation
- enfin, et surtout, faiblesse notable des procédures d'entrées-sorties :

Pascal écrit dans un fichier des enregistrements de structure homogène et de même longueur : on ne peut écrire dans un fichier des enregistrements de types différents et/ou de longueur distincte. De plus, l'occupation d'espace en accès direct Pascal n'est pas optimisée : Pascal écrit des enregistrements de longueur X dans des enregistrements physiques de longueur Y. Le blocage n'étant pas effectué, il y a un enregistrement logique par enregistrement physique. Si X est très petit devant Y, la place perdue est importante.

La faiblesse des procédures d'entrées-sorties ont entraîné le développement d'un module spécifique d'entrées-sorties [R. Gardien et Al, 1980B].

#### 1.1.4. Incidence sur le logiciel : un module d'entrées-sorties

Par rapport aux entrées-sorties de Pascal, le module d'entrées-sorties apporte les améliorations suivantes :

- . Il évite la prolifération de fichiers : tout l'espace d'un utilisateur est géré dans un seul fichier physique découpé en "segments".
- . Il permet d'écrire des enregistrements de types différents dans un segment : chaque segment est découpé en membres (7 au maximum), chaque membre peut contenir un type d'enregistrement différent.
- . Il permet de mélanger dans un membre des enregistrements de même longueur mais de structures totalement différentes.
- . Il offre un interface homogène même si les entrées-sorties Pascal varient en fonction du site d'implémentation : il est fort possible qu'un site ne bénéficie pas des entrées-sorties en accès direct. Le choix lui est alors laissé : soit il supporte la faiblesse Pascal car le module d'E/S simule l'accès direct à l'aide de l'accès séquentiel Pascal, soit il reprogramme la partie du module d'E/S qui effectue les entrées sorties dans un langage de base qui permet l'accès direct (assembleur par exemple).

- . Il permet d'écrire et de lire simplement des enregistrements de structure dynamique (leur longueur n'étant pas connue a priori, ils sont gérés dans plusieurs tableaux successifs chaînés) à l'aide des routines : ECRCH et LIRCH qui permettent respectivement d'écrire ou de lire un enregistrement par "fraction".
- . Enfin, il optimise l'occupation d'espace en accès direct en gérant un espace des clés logiques et un espace des clés physiques. Il effectue un mapping et permet de regrouper plusieurs clés logiques dans une même clé physique. Les buffers physiques écrits contiennent plusieurs enregistrements logiques.

## 1.2. Choix d'une méthode d'accès

Par le choix d'une méthode d'accès, nous entendons le choix d'une méthode de représentation logique qui, à chaque valeur prise par un index clé fasse correspondre l'adresse de l'enregistrement correspondant. La méthode ne doit privilégier aucun des index.

### 1.2.1. Rappel sur les diverses méthodes

Deux méthodes sont utilisées pour représenter un index :

- la méthode arborescente
- la méthode du hash code.

a) Le principe de la méthode arborescente, illustré par la figure A1.2 ci-après reprise dans [D.E. Knuth, 1973], consiste à cheminer dans un arbre pour retrouver ou insérer une clé. Aux feuilles de l'arbre sont stockés les numéros d'enregistrement des données correspondant à chacune des clés.



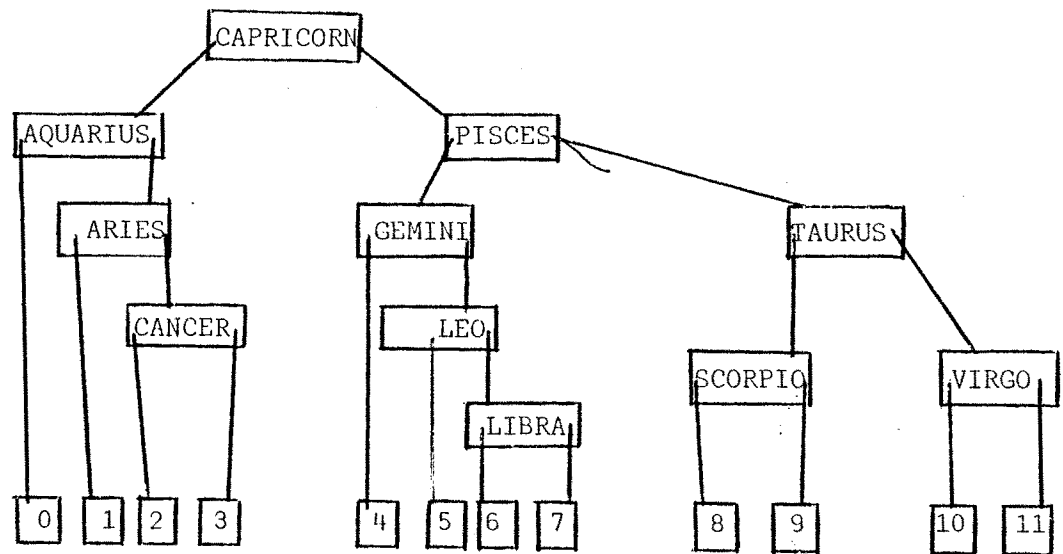


Figure A1.2 : Un exemple d'arbre binaire

Pour que les accès restent rapides, une arborescence doit toujours rester "bien balancée" (toutes les feuilles se trouvent au même niveau ou décalées d'au plus un niveau).

Une arborescence mal balancée n'est remise à l'état bien balancé qu'au prix d'algorithmes coûteux.

L'emploi des b-arbres proposés par R. Bayer et E. McCreight permet d'éviter le coût de ce rééquilibrage : un b-arbre est une arborescence bien balancée associée à des opérations qui préservent cet équilibre [D.E. Knuth, 1973], [R. Bayer et Al, 1977].

Un des avantages de la méthode arborescente est de permettre le parcours des enregistrements selon l'ordre défini par chacun des index. Elle est souvent choisie pour cet avantage [J.M. Forrestier, 1979].

b) La méthode du hash code permet par une opération mathématique sur une valeur d'index (dite opération de hash code) d'obtenir l'adresse d'un enregistrement ou d'un emplacement contenant l'adresse de l'enregistrement. L'avantage principal de cette méthode est la rapidité. Les principaux points faibles de la méthode sont les suivants :

- Peu d'insertions peuvent détériorer les performances de façon notable.
- En pratique, le taux de remplissage est fréquemment faible.
- Enfin, jusqu'à tout récemment, le changement de la fonction de hachage pour des raisons de dégradations de performances, était une opération très complexe pouvant entraîner pendant quelque temps l'arrêt de l'exploitation normale du fichier [D.E. Knuth, 1973].

Il convient de noter toutefois que pour des fichiers statiques, les performances d'accès et d'occupation des mémoires adressées par hachage peuvent être parfaites et le hachage se révèle une technique meilleure que toutes les techniques concurrentes connues. Toutefois, les fichiers statiques deviennent de plus en plus rares et on en revient alors aux faiblesses du hachage classique citées précédemment. Tout récemment, de nouvelles techniques d'adressage basées sur le hachage, permettent de supprimer ces faiblesses [R. Fagin et Al, 1979], [W. Litwin, 1979], [P.A. Larson, 1978]. Les performances de ces méthodes sont proprement remarquables [M. Scholl, 1979] et supplantent de loin les techniques concurrentes. Le point commun de ces méthodes est de considérer que la fonction de hachage peut être modifiée dynamiquement.

### 1.2.2. Solution retenue

Notre choix dès le départ était guidé par des impératifs d'efficacité en temps d'accès (l'accès aux séries chronologiques doit être le plus rapide possible notamment en phase de simulation de modèles). Dès lors, le choix de la méthode du hachage était tout indiqué. L'apparition de nouvelles méthodes plus performantes [W. Litwin, 1979], [R. Fagin et Al, 1979] basées sur le hachage n'a fait que renforcer notre choix. Toutefois, en l'étape actuelle de développement du logiciel, nous nous sommes cantonnés à implémenter le hachage classique. La fonction de hash code choisie est la division, les collisions gérées par le module d'E/S [R. Gardien et Al, 1980B] sont gérées selon la méthode du chaînage de case. A terme, notre volonté est d'implémenter le hachage virtuel et notamment l'algorithme HVL décrit par Litwin dans [W. Litwin, 1979], [W. Litwin, 1978] au moins au niveau des segments où les insertions et les effacements sont fréquents.



## ANNEXE II - PLAN

	Page
Normes de présentation	A2.1
1. COMMANDES DE LA BASE CENTRALE	A2.2
1.1. Création de séries	A2.2
1.2. Création de nomenclatures	A2.5
1.3. Création de calendriers	A2.6
1.3.1. Création d'un descripteur de calendrier	A2.6
1.3.2. Création d'un ensemble d'entiers	A2.7
1.3.3. Création d'un ensemble de libellés	A2.8
1.4. Création d'entrées synonymes	A2.9
1.5. Mise à jour de séries	A2.10
1.5.1. Mise à jour d'une série	A2.10
1.5.2. Mise à jour des valeurs d'une liste de séries pour une même liste de dates	A2.13
1.6. Destruction	A2.15
1.7. Copie de séries	A2.16
1.8. Copie de segments dans la base centrale	A2.18
2. COMMANDES SUR LES SEGMENTS	A2.20
2.1. Création de segments	A2.20
2.2. Création de séries	A2.21
2.3. Création de nomenclatures	A2.23
2.4. Création de calendriers	A2.24
2.5. Mise à jour de séries	A2.26
2.5.1. Mise à jour d'une série	A2.26
2.5.2. Mise à jour des valeurs d'une liste de séries pour une même liste de dates	A2.28
2.6. Destruction	A2.30

	Page
2.7. Manipulation via le L3E	A2.31
2.8. Copie de segment dans un autre segment	A2.33
2.9. Impression de segments	A2.34
2.10. Copie de séries	A2.35
3. COMMANDES COMMUNES BASE CENTRALE - SEGMENTS	A2.38
3.1. Initialisation de variables globales	A2.38
3.2. Impression de séries	A2.40
3.3. Impression de nomenclatures	A2.42
3.4. Impression de calendriers	A2.43
4. FONCTIONS MODULECO SUR LA BASE DE DONNEES	A2.45
4.1. Extraction de tout ou partie des valeurs d'une série	A2.45
4.2. Extraction de nomenclatures	A2.46
4.3. Fourniture de la liste de nomenclatures d'une série	A2.47
4.4. Fonction renvoyant le nombre de dimensions d'une série	A2.48
4.5. Fonction renvoyant une liste d'identificateurs de calendriers pour une série donnée	A2.49

## NORMES DE PRESENTATION

Chaque commande est présentée selon le plan suivant :

- *ENTETE* : donne le nom de la commande et son nom abrégé
- *FORMAT* : donne la syntaxe de la commande
- *ARGUMENTS* : explicite chaque paramètre de la commande :
  - . son type
  - . la valeur prise par défaut si aucune valeur n'a été transmise à l'appel
  - . sa signification
- *EXEMPLES* : donne des exemples d'utilisation de la commande.

La syntaxe de commandes suit les normes suivantes :

- En majuscules : les noms de commandes et les mots clés introduisant les paramètres
- En minuscules : les paramètres à fournir
- Chaque mot clé annonçant un paramètre est suivi du signe =
- Les éléments d'une liste sont séparés par des ,
- Les paramètres entre < > sont facultatifs
- Les paramètres entre [.../...] sont exclusifs.

## 1. COMMANDES DE LA BASE CENTRALE

Tous les noms des commandes de la base centrale commenceront par un B, à l'exception de la commande COPBASE.

### 1.1. Création de séries

#### ENTETE

BCRSER (abv : BCS)

créé une série dans la base centrale

---

#### FORMAT

```
BCRSER SERIE=idser CALENDRIER=idcal
      <DATDEB=dd> <DATFIN=df> <DIMENSION=dimser> [<BORNES=listent>/
      <NOMENCLATURE=listnomen>] <MOTPASSE=motpasse>
      <DOC=doc> < VAL=listval ><OPTIONS=(<cv>,<typ>)>;
```

---

#### ARGUMENTS

- |       |   |
|-------|---|
| idser | <ul style="list-style-type: none"> <li>. identificateur de série</li> <li>. pas de valeur par défaut</li> <li>. nom de la série à créer dans la base centrale</li> </ul>  |
| idcal | <ul style="list-style-type: none"> <li>. identificateur de calendrier à choisir parmi ANNUEL, TRIMESTRIEL, MENSUEL, HEBDOMADAIRE, QUOTIDIEN ou identificateur quelconque</li> <li>. par défaut, valeur contenue dans la variable globale CALENDRIER ou sinon ANNUEL</li> <li>. périodicité de la série</li> </ul> |
| dd    | <ul style="list-style-type: none"> <li>. constante de type date</li> <li>. si la valeur n'a pas été transmise dans la commande, on vérifie que la variable globale DATEDEB contient la valeur cherchée</li> <li>. date de début des valeurs de la série</li> </ul>  |

- df . constante de type date  
. si la valeur n'a pas été transmise dans la commande,  
la valeur prise est :  
- soit la valeur contenue dans la variable globale  
DATEFIN,  
- soit la valeur calculée en fonction du nombre de  
valeurs transmises  
. date de fin des valeurs de la série
- dimser . constante de type entier  
. valeur par défaut : 0 (c'est une série simple)  
. dimension de la série
- listent . liste d'entiers  
. pas de valeur par défaut  
. liste des bornes maximum de chaque dimension de la  
série
- listnomen . liste d'identificateurs de nomenclature  
. pas de valeur par défaut, si le paramètre est absent  
cela signifie que c'est une série simple  
. identificateurs des nomenclatures dimensionnant la  
série si la série est à plusieurs dimensions
- motpasse . identificateur  
. pas de valeur par défaut, si le paramètre est absent,  
la série n'est pas protégée en consultation  
. identificateur du mot de passe protégeant la série en  
consultation
- doc . chaîne de caractères quelconques entre quotes  
. vide par défaut  
. documentation à associer à la série
- listval . liste de constantes réelles à choisir parmi A (valeur  
aberrante), M (valeur manquante) ou valeur quelconque.  
La répétition de i valeurs identiques se code :/i/valeur



- . par défaut, la série est créée avec des valeurs aberrantes sur tout son domaine de définition
  - . liste des valeurs constituant la série
- options
- . options permettant de positionner le code de validité et de déterminer le type de la série
  - . par défaut, pas d'argument OPTIONS équivaut à CODVAL=0 (code de validité de la série = 0) et TYPE=0 (la série est primaire)
  - . sens des options :
    - cv : le code de validité de la série CODVAL=1
    - typ : le type de la série TYPE=D (la série est dérivée)

EXEMPLES

- 1 : BCRSER SERIE=PNB DATEDEB= # 1960  
 DATEFIN =#1970 MOTPASSE=DUPONT  
 DOC='PRODUIT NATIONAL BRUT'  
 VAL = (100.1,/4/M,110.4,120.8,130.7,140.9,140.9,150.5);  
 Création dans la base centrale de la série simple PNB de  
 périodicité ANNUEL, définie de 1960 à 1970 protégée en consul-  
 tation par le mot de passe DUPONT
- 2 : BCRSER SERIE=PRODAGRI CALENDRIER=TRIMESTRIEL  
 DATEDEB= II/1966 DATEFIN= IV/1970  
 NOMENCLATURE=PRODUIT DIMENSION=1;  
 Création dans la base centrale de la série à une dimension  
 PRODAGRI de périodicité TRIMESTRIEL définie de II/1966 à  
 IV/1970, dimensionnée par la nomenclature PRODUIT, non protégée  
 en consultation. La série est préformatée avec des valeurs  
 aberrantes sur tout son domaine de définition

## 1.2. Création de nomenclatures

### ENTETE

BCRNOM (abv=BCN)

créée une nomenclature dans la base centrale

### FORMAT

BCRNOM NOMENCLATURE=idnomen

DESCRIPTION=listlib

<DOC=listdoc>;

### ARGUMENTS

- idnomen . identificateur de nomenclatures
  - . pas de valeur par défaut
  - . nom et taille de la nomenclature à créer
- listlib . liste d'identificateurs
  - . pas de valeur par défaut
  - . liste des libellés composant la nomenclature
- listdoc . liste de chaînes de caractères entre quotes
  - . liste vide par défaut
  - . chaînes documentaires à associer à la nomenclature et à chacun des libellés de la nomenclature

### EXEMPLES

1 : BCRNOM NOMENCLATURE=PAYS(4)

DESCRIPTION=(FRANCE,ITALIE,ESPAGNE,PORTUGAL);

Création de la nomenclature PAYS ayant 4 entrées composées des libellés FRANCE, ITALIE, ESPAGNE, PORTUGAL

2 : BCRNOM NOMENCLATURE=PRODUIT(5)

DESCRIPTION=(BLE,ORGE,SEIGLE,AVOINE,MAIS)

DOC='NOMENCLATURE PRODUIT RECENSANT 5 PRODUITS';

Création de la nomenclature PRODUIT ayant 5 entrées composées des libellés BLE, ORGE, SEIGLE, AVOINE, MAIS. On associe à la nomenclature une documentation générale.

### 1.3. Création de calendrier

#### 1.3.1. Création d'un descripteur de calendrier

##### ENTETE

BCRCAL (abv=BCC)

Crée un descripteur de calendrier défini dans la base centrale

---

##### FORMAT

```
BCRCAL <CALENDRIER=idcal>
      DESCRIPTION=listens
      <DOC=doc>;
```

---

##### ARGUMENTS

- |         |  |
|---------|--|
| idcal   | . identificateur   |
|         | . par défaut, valeur de la variable globale CALENDRIER                           |
|         | . identificateur du calendrier à créer   |
| listens | . liste d'identificateurs  |
|         | . pas de valeur par défaut   |
|         | . liste des identificateurs d'ensembles définissant la périodicité du calendrier |
| doc     | . chaîne de caractères quelconques entre quotes                                  |
|         | . chaîne vide par défaut   |
|         | . documentation à associer au calendrier   |

##### EXEMPLE

```
1 : BCRCAL CALENDRIER=CALDEF
      DESCRIPTION=(JOUR,MOIS,ANNEE);
```

Création d'un descripteur de calendrier pour le calendrier de nom CALDEF composé de trois ensembles JOUR, MOIS et ANNEE.

La périodicité du calendrier est JOUR/MOIS/ANNEE

## 1.3.2. Création d'un ensemble d'entiers

ENTETE

CRENT (abv=CRE)

Crée un ensemble d'entiers pour le calendrier défini auparavant  
par BCRCAL ou CRCAL (si segment)

FORMAT

CRENT <CALENDRIER=idcal>

ENSEMBLE=idens

[DESCRIPTION=listens/INTERVALLE=(bornemin,bornemax<,pas>)];

ARGUMENTS

- |          |   |
|----------|---|
| idcal    | . identificateur<br>. par défaut, calendrier courant<br>. identificateur de calendrier pour lequel est créé<br>l'ensemble d'entiers |
| idens    | . identificateur<br>. pas de valeur par défaut<br>. identificateur d'ensemble à créer   |
| listent  | . liste de constantes entières<br>. pas de valeur par défaut<br>. liste d'entiers composant l'ensemble à créer                      |
| bornemin | . constante entière<br>. pas de valeur par défaut<br>. borne minimum dans le cas où l'ensemble du calendrier<br>est un intervalle   |
| bornemax | . constante entière<br>. pas de valeur par défaut<br>. borne maximum dans le cas où l'ensemble du calendrier<br>est un intervalle   |
| pas      | . constante entière<br>. valeur par défaut = 1<br>. pas de l'intervalle décrit par bornemin, bornemax                               |

EXEMPLE

```

1 : BCRCAL CALENDRIER=CALDEF
      DESCRIPTION=(JOUR,MOIS,ANNEE);

CRENT CALENDRIER=CALDEF
      ENSEMBLE=JOUR
      INTERVALLE=(1,30);

CRENT ENSEMBLE=ANNEE
      DESCRIPTION=(1960,1962,1965,1966);

```

Création des ensembles JOUR et ANNEE du calendrier CALDEF dont le descripteur a été créé par BCRCAL

## 1.3.3. Création d'un ensemble de libellés

ENTETE

CRLIB (abv=CRL)

Crée un ensemble de libellés pour le calendrier défini auparavant par BCRCAL ou CRCAL

FORMAT

```

CRLIB <CALENDRIER=idcal>
      ENSEMBLE=idens
      DESCRIPTION=listlib;

```

ARGUMENTS

idcal	. identificateur
	. par défaut, calendrier courant
	. identificateur de calendrier pour lequel est créé l'ensemble de libellés
idens	. identificateur
	. pas de valeur par défaut
	. identificateur d'ensemble à créer

listlib . liste d'identificateurs  
 . pas de valeur par défaut  
 . liste des libellés composant l'ensemble

EXEMPLE

```
1 : BCRCAL CALENDRIER=CALDEF
      DESCRIPTION=(JOUR,MOIS,ANNEE);

CRLIB ENSEMBLE=MOIS
      DESCRIPTION=(JANVIER,FEVRIER,MARS);
```

Création de l'ensemble MOIS du calendrier CALDEF dont le descripteur a été créé par BCRCAL

1.4. Création d'entrées synonymesENTETE

```
BCRSYN (abv=BCRS)
définit un nom synonyme pour une série de la base centrale
```

FORMAT

```
BCRSYN SERIE=idser <CALENDRIER=idcal>
      SYNONYME=idsyn <MOTPASSE=motpasse>;
```

ARGUMENTS

idser . identificateur de série  
 . pas de valeur par défaut  
 . identificateur de série pour lequel on veut définir un synonyme

idcal . identificateur  
 . par défaut, la valeur contenue dans la variable globale CALENDRIER ou sinon ANNUEL  
 . identificateur de calendrier sous lequel est définie la série

idsyn . identificateur de série  
 . pas de valeur par défaut  
 . identificateur synonyme

motpasse . identificateur  
 . pas de valeur par défaut  
 . identificateur du mot de passe consultation de la  
 série

EXEMPLES

1 : BCRSYN SERIE=PRODUCTIONAGRICO  
 CALENDRIER=TRIMESTRIEL  
 SYNONYME=PRODAGR;

Définit l'entrée synonyme PRODAGR pour la série PRODUCTIONAGRICO de périodicité TRIMESTRIEL le paramètre MOTPASSE n'étant pas codé, cela signifie que la série n'est pas protégée en consultation

2 : BCRSYN SERIE=PRODNATBRUT  
 SYNONYME=PNB

Définit l'entrée synonyme PNB pour la série PRODNATBRUT de périodicité ANNUEL et non protégée en consultation

1.5. Mise à jour des séries1.5.1. Mise à jour d'une sérieENTETE

BMAJSER (abv=BMS)

Met à jour une série dans la base centrale

---

FORMAT

```

BMAJSER SERIE=idser <CALENDRIER=idcal>
[<DATMOD=listdat>/<INTERVMAJ=(datd,datf)>]<MOD=listval>
[<DATSUP=listdat>/<INTERVSUP=(datd,datf)>]
<MODIDENT=identser>
<MODMP=motpasse> <MODDOC=docmod>
<AJODOC=docsup> <OPTIONS=cv>;

```

---

ARGUMENTS

idser	. identificateur de série . pas de valeur par défaut . identificateur de la série à mettre à jour
idcal	. identificateur . par défaut, valeur contenue dans la variable globale CALENDRIER ou sinon ANNUEL . identificateur de calendrier sous lequel est définie la série
listdat	. liste de constantes dates . par défaut, date début de la série . liste de dates pour lesquelles on modifie, on ajoute ou on supprime des valeurs
datd	. constante date . par défaut date début de série . date à partir de laquelle des valeurs sont modifiées ou supprimées
datf	. constante date . par défaut date de fin de la série . date jusqu'à laquelle des valeurs sont modifiées ou supprimées
listval	. liste de constantes réelles . pas de valeur par défaut . liste des valeurs modifiant les valeurs ou rallongeant la série



- identser . identificateur de série  
 . pas de valeur par défaut  
 . identificateur de série remplaçant l'identificateur existant
- motpasse . identificateur  
 . pas de valeur par défaut  
 . identificateur de mot de passe remplaçant le mot de passe ou protégeant la série si celle-ci ne l'était pas
- docmod . chaîne de caractères quelconques entre quotes  
 . chaîne vide par défaut  
 . documentation modifiant la documentation existante
- docsup . chaîne de caractères quelconques entre quotes  
 . chaîne vide par défaut  
 . documentation rallongeant la documentation existante ou documentant la série si celle-ci ne l'était pas
- options . options\* permettant de positionner le code de validité  
 . par défaut, pas d'argument OPTIONS équivaut à CODVAL=0 (code de validité de la série=0)  
 . sens des options :  
     - cv : le code de validité de la série CODVAL=1

EXEMPLES

```
1 : BCRSER SERIE=PNB DATEDEB= #1960 DATEFIN= #1970
      VAL=(100.1,/4/M,110.4,120.8,130.7,140.9,150.5);
      BMAJSER SERIE=PNB DATEMOD=(#1961,#1962,#1963,#1964,#1971)
      MOD=(100.2,100.6,110.2,110.3,160.2);
```

- . Création dans la base centrale de la série simple PNB de périodicité ANNUEL, définie de 1960 à 1970
- . Mise à jour de la série simple PNB de périodicité ANNUEL consistant à modifier 4 valeurs de la série correspondant aux dates 1961, 1962, 1963, 1964 et à rallonger la série en créant sa valeur pour la date 1971

```

2 : BCRNOM NOMENCLATURE=PRODUIT(2)
      DESCRIPTION=(BLE,ORGE);

BCRSER SERIE=PRODAGRI CALENDRIER=TRIMESTRIEL
      DATEDEB= #II/1966 DATEFIN= #IV/1967
      NOMENCLATURE=PRODUIT DIMENSION=1;

BMAJSER SERIE=PRODAGRI CALENDRIER=TRIMESTRIEL
      MOD=(100.1,100.2,100.5,110.1,110.4,110.5,110.6,90.2,
           90.4,90.5,100.6,100.8,102.4,103.2)
      MODIDENT=PRODAG MODMP=DUPONT;

```

Création dans la base centrale de la série à une dimension PRODAGRI de périodicité TRIMESTRIEL définie de II/1966 à IV/1967, dimensionnée par la nomenclature PRODUIT créée auparavant par BCRNOM, non protégée en consultation. Les valeurs de la série sont fournies par la commande BMAJSER qui permet par ailleurs la modification du nom de la série qui s'appellera PRODAG et qui sera protégée en consultation par le mot de passe DUPONT. Implicitement la date début de la série est prise.

### 1.5.2. Mise à jour des valeurs d'une liste de séries pour une même liste de dates

#### ENTETE

BMAJVAL (abv=BMV)

Met à jour les valeurs d'une liste de séries de la base centrale pour un même intervalle de dates

---

#### FORMAT

```

BMAJVAL SERIE=listser
      <CALENDRIER=idcal> [ <DATMOD=lisdat><INTERVMAJ=(dat1,dat2)> ]
      <MOD=listval>
      [<DATSUP=listdat>/<INTERVSUP=(dat1,dat2)>];

```

---

ARGUMENTS

- listser . liste d'identificateurs de série  
 . pas de valeur par défaut  
 . liste des identificateurs de série à mettre à jour
- Si le premier identificateur de la liste = TOUTES, cela signifie que toutes les séries ayant la périodicité désignée par IDCAL sont à mettre à jour
- idcal . identificateur  
 . par défaut, valeur contenue dans la variable globale CALENDRIER ou sinon ANNUEL  
 . identificateur de calendrier sous lequel sont définies les séries
- listdat . liste de constantes dates  
 . pas de valeur par défaut  
 . liste de dates pour lesquelles on modifie, on ajoute ou on supprime des valeurs
- dat1,dat2 . constantes dates  
 . pas de valeur par défaut  
 . dates déterminant l'intervalle de modification ou de suppression
- listval . liste de constantes réelles  
 . pas de valeur par défaut  
 . liste des valeurs modifiant les valeurs ou ajoutées aux séries

EXEMPLES

```
1 : BCRSER SERIE=PNB DATEDEB= #1960 DATEFIN= #1970
    VAL=(100.1,/4/M,110.4;120.8.,130.7,140.9,150.5);
    BCRSER SERIE=PIB DATEDEB= #1959 DATEFIN= #1968
    VAL=(140.2,140.3,/3/M,150.2,150.3,150.8,150.9,170.2);
```

```
BMAJVAL SERIE=(PNB,PIB)
INTERVMOD=(#1961,#1963)
MOD=(100.2,100.6,110.2,140.5,140.6,150.0);
```

- . Création des séries PNB et PIB de périodicités ANNUEL définies respectivement de 1960 à 1970 et de 1959 à 1968
- . Mise à jour des valeurs de ces deux séries pour l'intervalle de dates 1961 à 1963

## 1.6. Destruction

### ENTETE

BDETRUIRE (abv=BDTR)

détruit un ou plusieurs objets de la base centrale

---

### FORMAT

```
BDETRUIRE <SERIE=listser <PERIODICITE=listidcal>>
          <CALENDRIER=listcal>
          <NOMENCLATURE=listnomen>
          <SYNONYME=listsyn>
```

---

### ARGUMENTS

- |           |   |
|-----------|---|
| listser   | . liste d'identificateurs de série  |
|           | . liste vide par défaut   |
|           | . liste des identificateurs des séries à détruire   |
| listidcal | . liste d'identificateurs   |
|           | . par défaut, si le paramètre SERIE est présent (sinon liste vide), valeur contenue dans la variable globale CALENDRIER ou sinon ANNUEL |
|           | . liste des identificateurs de calendrier auxquels sont rattachés les séries à détruire   |
| listcal   | . liste d'identificateurs   |
|           | . liste vide par défaut   |
|           | . liste des identificateurs des calendriers à détruire  |

- listnomen . liste d'identificateurs de nomenclatures  
 . liste vide par défaut  
 . liste des identificateurs des nomenclatures à détruire
- listsyn . liste d'identificateurs de série  
 . liste vide par défaut  
 . liste des identificateurs d'entrées synonymes à invalider

EXEMPLES

- 1 : BDETRUIRE SERIE=(PNB,PIB)  
 Destruction des séries PNB et PIB de périodicité ANNUEL
- 2 : BDETRUIRE CALENDRIER=CALDEF  
 NOMENCLATURE=(PRODUIT,PAYS);  
 Destruction du calendrier CALDEF et des nomenclatures PRODUIT et PAYS
- 3 : BDETRUIRE SERIE=(PRODAGRI,PRODINDUS)  
 PERIODICITE=(TRIMESTRIEL,MENSUEL);  
 Destruction de la série PRODAGRI de périodicité TRIMESTRIEL et PRODINDUS de périodicité MENSUEL

1.7. Copie de sériesENTETE

BCOPSER (abv=BCSR)

copie une ou plusieurs séries d'un segment donné dans la base centrale

FORMAT

```
BCOPSER <SEGMENT=idseg>
        <SERIE=listser>
        <MOTPASSE=motpasse>
        <DE=datel> <A=date2>
        <OPTIONS=(<nomen>,<cal>)>;
```

ARGUMENTS


---

idseg	. identificateur de segment . par défaut, valeur contenue dans la variable globale SEGMENT . identificateur du segment contenant les séries à recopier
listser	. liste d'identificateurs de série . pas de valeur par défaut . liste des identificateurs des séries à recopier
motpasse	. identificateur . par défaut, séries non protégées . mot de passe consultation des séries à recopier
datel,	. constantes date
date2	. par défaut, intervalle de définition des séries . intervalle de dates sur lequel recopier les séries
options	. options permettant d'indiquer si le calendrier et les nomenclatures doivent être recopiées dans la base centrale . par défaut, pas d'arguments OPTIONS équivaut à la non recopie du calendrier et des nomenclatures . sens des options : <ul style="list-style-type: none"> <li>- cal : le calendrier est recopié dans la base centrale</li> <li>- nomen : les nomenclatures sont recopiées dans la base centrale</li> </ul>

EXEMPLES

1 : BCOPSER SEGMENT=SEGSERIES

SERIE=(PNB,PIB)

DE= #1962 A= #1968;

Recopie dans la base centrale des séries PNB et PIB du segment  
SEGSERIES sur l'intervalle 1962 à 1968

2 : BCOPSER SEGMENT=SEGSERIES

SERIE=(ECHANGE,PRODAGRI)

OPTIONS=NOMEN;

Recopie dans la base centrale des séries ECHANGE et PRODAGRI  
du segment SEGSERIES, la recopie se fait sur tout l'intervalle  
de définition, les nomenclatures associées sont, elles aussi,  
recopiées

### 1.8. Copie de segments dans la base centrale

ENTETE

COPBASE (abv=CPB)

Copie un segment dans la base centrale

FORMAT

COPBASE <SEGMENT=idseg>

<SAUF=listser>

<OPTIONS=(<nomen>,<cal>)>;

ARGUMENTS

- idseg
- . identificateur de segment
  - . par défaut, valeur contenue dans la variable globale  
SEGMENT
  - . identificateur du segment à recopier dans la base centrale

- listser . liste d'identificateurs de série  
 . liste vide par défaut  
 . liste des séries à ne pas recopier
- options . options permettant d'indiquer si le calendrier et  
 les nomenclatures doivent être recopiés dans la base  
 centrale  
 . par défaut, pas d'argument OPTIONS équivaut à la non  
 recopie du calendrier et des nomenclatures  
 . sens des options  
 - cal : le calendrier est recopié dans la base  
 centrale  
 - nomen : les nomenclatures sont recopiées dans la  
 base centrale

EXEMPLES

1 : COPBASE SEGMENT=SEGSERIES  
 OPTIONS=(NOMEN);

Copie dans la base centrale du segment SEGSERIES, les nomencla-  
 tures sont aussi recopiées

2 : CPB SEGMENT=SEGSERIES  
 SAUF=(PNB) OPTIONS=(NOMEN);

Copie dans la base centrale du segment SEGSERIES à l'exception  
 de la série PNB

Les nomenclatures sont aussi recopiées.



## 2. COMMANDES SUR LES SEGMENTS

### 2.1. Création de segments

#### ENTETE

INITSEG (abv=INSG)

Initialise un segment de série

---

#### FORMAT

INITSEG <SEGMENT=idseg> <CALENDRIER=idcal>  
 <DATEDEB=dd> <DATEFIN=df>  
 <DOC=doc> <LISTACCES=listutil>;

---

#### ARGUMENTS

- |       |  |
|-------|--|
| idseg | . identificateur de segment  |
|       | . par défaut, la valeur prise est celle contenue dans la variable globale SEGMENT                    |
|       | . identificateur du segment à initialiser  |
| idcal | . identificateur   |
|       | . par défaut, la valeur prise est celle contenue dans la variable globale CALENDRIER ou sinon ANNUEL |
|       | . identificateur du calendrier de référence des séries du segment                                    |
| dd    | . constante date   |
|       | . par défaut, la valeur prise est celle contenue dans la variable globale DATEDEB                    |
|       | . date de début des séries du segment  |
| df    | . constante date   |
|       | . par défaut, la valeur prise est celle contenue dans la variable globale DATEFIN                    |
|       | . date de fin des séries du segment  |

doc . chaîne de caractères quelconques entre quotes  
 . chaîne vide par défaut  
 . documentation générale à associer au segment

listutil . liste d'identificateurs  
 . liste vide par défaut  
 . liste des utilisateurs autorisés en consultation sur ce segment

EXEMPLES

```
1 : INITSEG SEGMENT=SEGSERIES
      DATEDEB= #1960 DATEFIN= #1970
      LISTACCES=(DUPONT,DURAND);
```

Initialisation du segment SEGSERIES qui contiendra des séries de périodicité ANNUEL définies de 1960 à 1970

Le segment est accessible en consultation aux utilisateurs DUPONT et DURANT.

2.2. Création de sériesENTETE

```
CRSER (abv=CS)
Crée une série dans un segment
```

FORMAT

```
CRSER <SEGMENT=idseg> SERIE=idser
      <DIMENSIONS=dimser>[<BORNEB=listent>/<NOMENCLATURE=listnomen>]
      <DOC=doc>
      <VAL=listval> <OPTIONS=(<cv>,<typ>);
```

ARGUMENTS

- idseg . identificateur de segment  
. par défaut, la valeur prise est celle contenue dans la variable globale SEGMENT  
. identificateur du segment où sont créées les séries
- idser . identificateur de série  
. pas de valeur par défaut  
. identificateur de la série à créer
- dimser . constante entière  
. par défaut, dimser=0 (c'est une série simple)  
. dimension de la série à créer
- listent . liste d'entiers  
. pas de valeur par défaut  
. liste des bornes maximum de chaque dimension de la série
- listnomen . liste d'identificateurs de nomenclatures  
. liste vide par défaut  
. liste des identificateurs de nomenclatures dimensionnant la série
- doc . chaîne de caractères quelconques entre quotes  
. chaîne vide par défaut  
. documentation à associer à la série
- listval . liste de constantes réelles à choisir parmi A (valeur aberrante), M (valeur manquante) ou valeur quelconque. La répétition de i valeurs identiques se code /i/valeur  
. par défaut, la série est créée avec des valeurs aberrantes sur tout son domaine de définition  
. liste des valeurs constituant la série

- options . options permettant de positionner le code de validité et de déterminer le type de la série
- . par défaut, pas d'argument OPTIONS équivaut à CODVAL=0 (code de validité de la série = 0) et TYPE=P (la série est primaire)
- . sens des options :
- cv : le code de validité de la série CODVAL=1
  - typ : le type de la série TYPE=D (la série est dérivée)

EXEMPLES

```
1 : INITSEG SEGMENT=SEGSERIES
      DATEDEB= #1960 DATEFIN= #1970
      LISTACCES=(DUPONT,DURAND)

CRSER SERIE=PNB
      VAL=(100.1,/4/M,110.4,120.8,130.7,140.9,140.9,150.5);
```

Initialisation du segment SEGSERIES qui contiendra des séries de périodicité ANNUEL définies de 1960 à 1970, création dans le segment SEGSERIES de la série PNB.

2.3. Création de nomenclaturesENTETE

CRNOM (abv=CN)

Crée une nomenclature dans un segment

FORMAT

```
CRNOM <SEGMENT=idseg>
      NOMENCLATURE=idnomen
      <DESCRIPTION=listlib>
      <DOC=doc>;
```

ARGUMENTS

- idseg . identificateur de segment  
 . par défaut, la valeur prise est celle contenue dans la variable globale SEGMENT  
 . identificateur du segment auquel sont rattachés les nomenclatures
- idnomen . identificateur de nomenclatures  
 . pas de valeur par défaut  
 . nom et taille de la nomenclature à créer
- listlib . liste d'identificateurs  
 . pas de valeur par défaut  
 . liste des libellés composant la nomenclature
- doc . Chaîne de caractères entre quotes  
 . chaîne vide par défaut  
 . documentation générale à associer à la nomenclature

EXEMPLES

1 : CRNOM SEGMENT=SEGSERIES

NOMENCLATURE=PAYS(4)

DESCRIPTION=(FRANCE,ITALIE,ESPAGNE,PORTUGAL);

Création dans le segment SEGSERIES de la nomenclature PAYS composée des libellés FRANCE, ITALIE, ESPAGNE, PORTUGAL.

## 2.4. Création de calendriers

ENTETE

CRCAL (abv=CC)

Crée un descripteur de calendrier défini dans un segment

---

FORMAT

```

CRCAL <SEGMENT=idseg> <CALENDRIER=idcal>
      DESCRIPTION=listens
      <DOC=doc>;

```

ARGUMENTS

- |         |  |
|---------|--|
| idseg   | <ul style="list-style-type: none"> <li>. identificateur de segment</li> <li>. par défaut, la valeur prise est celle contenue dans la variable globale SEGMENT</li> <li>. identificateur du segment pour lequel est créé le calendrier</li> </ul> |
| idcal   | <ul style="list-style-type: none"> <li>. identificateur</li> <li>. par défaut, la valeur prise est celle contenue dans la variable globale CALENDRIER</li> <li>. identificateur du calendrier à créer</li> </ul>                                 |
| listens | <ul style="list-style-type: none"> <li>. liste d'identificateurs</li> <li>. pas de valeur par défaut</li> <li>. liste des identificateurs d'ensembles constituant le calendrier</li> </ul>   |
| doc     | <ul style="list-style-type: none"> <li>. chaîne de caractères quelconques entre quotes</li> <li>. chaîne vide par défaut</li> <li>. documentation à associer au calendrier</li> </ul>  |
| note    | <ul style="list-style-type: none"> <li>. Pour créer un calendrier, il faut, comme sur la base centrale, utiliser aussi les commandes de création des ensembles du calendrier qui sont CRENT et CRLIB (cf. § 1.3.2 et 1.3.3)</li> </ul>           |

EXEMPLES

```

1 : CRCAL SEGMENT=SEGSERIES
      CALENDRIER=CALDEF
      DESCRIPTION=(JOUR,MOIS,ANNEE);

```

Création d'un descripteur de calendrier pour le calendrier de nom CALDEF composé des trois ensembles JOUR, MOIS, ANNEE dans le segment SEGSERIES.

## 2.5. Mise à jour de séries

### 2.5.1. Mise à jour d'une série

#### ENTETE

MAJSER (abv=MS)

Met à jour une série d'un segment

---

#### FORMAT

```
MAJSER <SEGMENT=idseg> SERIE=idser
      <[DATMOD=listdat/INTERVMOD=(dat1,dat2)]>
      <MOD=listval>
      <[DATSUP=listdat/INTERVSUP=(dat1,dat2)]>
      <MODIDENT=identser> <MODDOC=docmod>
      <AJOUDOC=docsup> <OPTIONS=cv>;
```

---

#### ARGUMENTS

- |         |  |
|---------|--|
| idseg   | . identificateur de segment  |
|         | . par défaut, la valeur prise est celle contenue dans la variable globale SEGMENT  |
|         | . identificateur du segment où est stockée la série à mettre à jour                |
| idser   | . identificateur de série  |
|         | . pas de valeur par défaut   |
|         | . identificateur de la série à mettre à jour                                       |
| listdat | . liste de constantes dates  |
|         | . par défaut, date début de la série   |
|         | . liste de dates pour lesquelles on modifie les valeurs ou on supprime des valeurs |

- dat1,dat2 . constante date  
 . par défaut, intervalle de définition de la série  
 . intervalle de mise à jour ou de suppression de valeurs
- listval . liste de constantes réelles  
 . pas de valeur par défaut  
 . liste des valeurs modifiant les valeurs dont les dates ont été indiquées dans listdat ou par dat1,dat2
- identser . identificateur de série  
 . pas de valeur par défaut  
 . identificateur de série remplaçant l'identificateur existant
- docmod . chaîne de caractères quelconques entre quotes  
 . chaîne vide par défaut  
 . documentation modifiant la documentation existante
- docsup . chaîne de caractères quelconques entre quotes  
 . chaîne vide par défaut  
 . documentation rallongeant la documentation existante ou documentant la série si celle-ci ne l'était pas
- options . option permettant de positionner le code de validité  
 . par défaut, pas d'argument OPTIONS équivaut à CODVAL=0 (code de validité de la série = 0)  
 . sens des options :  
 - cv : le code de validité de la série CODVAL=1

Exemples

```
1 : INITSEG SEGMENT=SEGSERIES
      DATEDEB= #1960 DATEFIN= #1970

      CRSER SERIE=PNB
      VAL=(100.1,/4/M,1104,120.8,130.7,140.9,140.9,150.5);
```



```

MAJSER SERIE=PNB
INTERVMOD=(#1961,#1964) MOD=(100.2,100.6,110.2,110.3)
MODIDENT=PRODNATBRUT AJODOC='CECI EST UNE SERIE MISE
A JOUR' OPTIONS=CV;

```

Création du segment SEGSERIES qui contiendra des séries de périodicité ANNUEL définies de 1960 à 1970, création dans le segment SEGSERIES de la série PNB; mise à jour de la série PNB définie dans le segment SEGSERIES : modification du nom de la série, elle s'appellera PRODNATBRUT, introduction d'une documentation, positionnement du code de validité et mise à jour de 4 valeurs à partir de #1961.

### 2.5.2. Mise à jour des valeurs d'une liste de séries pour une même liste de dates

#### ENTETE

MAJVAL (abv=MV)

Met à jour les valeurs d'une liste de séries d'un segment donné pour un même intervalle de dates

---

#### FORMAT

```

MAJVAL <SEGMENT=idseg> <SERIE=listser>
      <[DATMOD=(listdat)/INTERVMOD=(dat1,dat2)]> <MOD=listval>
      [<DATSUP=(listdat)>/<INTERVSUP=(dat1,dat2)>]

```

---

#### ARGUMENTS

- |         |   |
|---------|---|
| idseg   | <ul style="list-style-type: none"> <li>. identificateur de segment</li> <li>. par défaut, la valeur prise est celle de la variable globale SEGMENT</li> <li>. identificateur du segment contenant les séries à mettre à jour</li> </ul> |
| listser | <ul style="list-style-type: none"> <li>. liste d'identificateurs de série</li> <li>. par défaut toutes les séries du segment sont mises à jour</li> <li>. liste des identificateurs de série à mettre à jour</li> </ul>                 |

listdat . liste de constantes dates  
 . pas de valeur par défaut  
 . liste de dates pour lesquelles on modifie ou on supprime des valeurs

dat1,dat2 . constantesdate  
 . par défaut, intervalle de définition de la série  
 . dates déterminant l'intervalle de modification

listval . liste de constantes réelles  
 . pas de valeur par défaut  
 . liste des valeurs modifiant les valeurs

EXEMPLES

```
1 : INITSEG SEGMENT=SEGSERIES
      DATEDEB= #1959 DATEFIN= #1970;

      CRSER SERIE=PNB
      VAL=(A,100.1,141.0,M,110.4,120.8,130.7,140.9,140.9,150.5,/2/A);

      CRSER SERIE=PIB
      VAL=(140.2,140.3,131.0,M,150.2,150.3,150.8,150.9,170.2,
          121.0,A,A);

      MAJVAL SERIE=(PNB,PIB)
      INTERVMOD=(#1961,#1963)
      MOD=(100.2,100.6,110.2,140.5,140.6,150.0);
```

Initialisation du segment SEGSERIES destiné à contenir des séries de périodicité ANNUEL définies de 1959 à 1970.

Création des séries PNB et PIB.

Mise à jour des valeurs de ces deux séries pour l'intervalle de dates 1961 à 1963.

## 2.6. Destruction

### ENTETE

DTRSEG (abv=DSG)

Détruit des segments et des objets de segment

### FORMAT

```
DTRSEG <SEGMENT= listseg > <NOMSEG=idseg>
      <SERIES=listser>
      <CALENDRIER=idcal>
      <NOMENCLATURE=listnom>;
```

### ARGUMENTS

- |         |  |
|---------|--|
| listseg | . liste d'identificateurs de segment                           |
|         | . pas de valeur par défaut                                     |
|         | . liste des identificateurs de segment à détruire              |
| idseg   | . identificateur de segment                                    |
|         | . par défaut, valeur prise par la variable globale<br>SEGMENT  |
|         | . identificateur du segment contenant les objets à<br>détruire |
| listser | . liste d'identificateurs de série                             |
|         | . liste vide par défaut  |
|         | . liste des identificateurs des séries à détruire              |

- idcal . identificateur  
 . pas de valeur par défaut  
 . identificateur de calendrier à détruire
- listnom . liste d'identificateurs de nomenclature  
 . liste vide par défaut  
 . liste des identificateurs des nomenclatures à  
 détruire

EXEMPLES

```
1 : DTRSEG  SEGMENT=SEGSERIES
      SERIE=(PNB,PIB)
```

Destruction des séries PNB et PIB du segment SEGSERIES

2.7. Manipulation via le L3EENTETE

CALCULER (abv=CALC)

Exécute un texte en L3E qui décrit un ensemble de manipulations  
 sur des séries provenant de segments

FORMAT

```
CALCULER <SEGDR=listseg> <SEGDT=idseg> <SEGCoeff=idsgco>
          <SEGPARAM=listsgpar> <DE=dat> <A=dat>
          TEXTL3E=listinst
          <OPTIONS=(desc,sauve) >;
```

ARGUMENTS

- listseg . liste d'identificateurs de segment  
 . par défaut, valeur de la variable globale SEGMENT  
 . segments où sont prises les séries servant pour le  
 calcul

- idsgco . identificateur de segment . de coefficient
  - . pas de valeur par défaut
  - . segment où sont pris les coefficients
- listsgpar . liste d'identificateurs de segments de paramètres
  - . pas de valeur par défaut
  - . segments où sont pris les paramètres
- dat . constante date
  - . par défaut, intervalle défini par les variables globales DATEDEB, DATEFIN ou sinon intervalle commun de définition des séries
  - . intervalle de calcul
- listinst . liste de déclarations et d'instructions exprimée en L3E
  - . pas de valeur par défaut
  - . description de manipulations à effectuer sur des séries
- options . options permettant de déclencher l'impression des séries résultats et/ou la sauvegarde
  - . par défaut : impression et sauvegarde
  - . sens des options :
    - imp : les séries résultats sont imprimées
    - sauve : les séries résultats sont sauvegardées

EXEMPLES

```
1 : CALCULER SEGOR=SEGSERIES
      TEXTL3E=(X=Y*1.2;)
      OPTIONS=IMP;
```

Calcul de la série X obtenue à partir de la série Y prise dans SEGSERIES. La série X est imprimée.

## 2.8. Copie de segments dans un autre segment

### ENTETE

COPSEG (abv=CSG)

Copie un ou plusieurs segments de séries dans un autre segment

### FORMAT

```
COPSEG <SEGOR=listseg>
      SEGDT=idseg
      <SAUF=listser>;
```

### ARGUMENTS

listseg . liste d'identificateurs de segment  
 . par défaut, nom du segment contenu dans la variable globale SEGMENT  
 . identificateurs des segments à recopier

idseg . identificateur de segment  
 . pas de valeur par défaut  
 . identificateur du segment destinataire

listser . liste d'identificateurs de série  
 . liste vide par défaut  
 . liste d'identificateurs de série à ne pas recopier dans le segment destinataire

### EXEMPLES

```
1 : INITSEG SEGMENT=SEGMENTSER
      DATEDEB= # 1960 DATEFIN= # 1970
```

```
COPSEG SEGOR=SEGSERIES
      SEGDT=SEGMENTSER;
```

Copie du segment SEGSERIES dans le segment SEGMENTSER initialisé par la commande INITSEG

```
2 : COPSEG SEGOR=(SEGSER1, SEGSER2)
      SEGDT=SEGMENTSER SAUF=PNB;
```

Copie des segments SEGSER1 et SEGSER2 dans le segment SEGMENTSER  
 La série PNB n'est pas copiée

## 2.9. Impression de segments de séries

### ENTETE

IMPSEG (abv=ISG)

Imprime un ou plusieurs segments de série

---

### FORMAT

```
IMPSEG <SEGMENT=listseg>
      <DE=dat1> <A=dat2>
      <SAUF=listser> <SUPPORT=idsup>
      <OPTIONS=[doc/val ]>;
```

---

### ARGUMENTS

- |         |   |
|---------|---|
| listseg | <ul style="list-style-type: none"> <li>. liste d'identificateurs de segment</li> <li>. par défaut, valeur contenue dans la variable globale SEGMENT</li> <li>. liste des identificateurs des segments à éditer</li> </ul>       |
| dat1    | <ul style="list-style-type: none"> <li>. constante date</li> <li>. par défaut, valeur contenue dans DATEDEB ou sinon date début du segment</li> <li>. date à partir de laquelle doit commencer l'édition des valeurs</li> </ul> |
| dat2    | <ul style="list-style-type: none"> <li>. constante date</li> <li>. par défaut, valeur contenue dans DATEFIN ou sinon date fin du segment</li> <li>. date de fin d'édition des valeurs</li> </ul>                                |
| listser | <ul style="list-style-type: none"> <li>. liste d'identificateurs de série</li> <li>. liste vide par défaut</li> <li>. liste d'identificateurs de séries que l'on ne veut pas imprimer</li> </ul>                                |
| idsup   | <ul style="list-style-type: none"> <li>. identificateur</li> <li>. par défaut, impression sur terminal</li> <li>. identificateur du support d'impression</li> </ul>   |

- options . options permettant d'indiquer si l'on désire l'impression de la partie documentaire ou de la partie valeurs des séries d'un segment
- . par défaut, impression de la totalité des informations concernant chaque série
- . sens des options :
- doc : seule la partie documentaire des séries est éditée
  - val : seule la partie valeur des séries est éditée

EXEMPLES

1 : IMPSEG;

Impression de tout le segment courant sur tout l'intervalle de définition. Toutes les informations sont imprimées. Le support d'édition est le terminal.

2 : IMPSEG SEGMENT=SEGSERIES

DE= #1962 A= #1968

SAUF=PNB SUPPORT=IMP

OPTIONS=VAL;

Impression des valeurs des séries du segment SEGSERIES à l'exception de la série PNB, sur l'intervalle 1962 à 1968. L'impression se fait sur imprimante.

## 2.10. Copie de séries

ENTETE

COPSERIE (abv=CSR)

Copie une ou plusieurs séries d'un segment donné ou de la base centrale dans un autre segment

---



FORMAT

```

COPSERIE <SEGOR=idseg1>
          <SEGDT=idseg2>
          <SERIE=listser> <CALENDRIER=idcal>
          <DE=daté1><A=daté2>
          <MOTPASSE=motpasse>
          <OPTIONS=(<cal>,<nomen>)>;

```

ARGUMENTS

- 
- |         |  |
|---------|--|
| idseg1  | <ul style="list-style-type: none"> <li>. identificateur de segment</li> <li>. par défaut, copie de la base centrale</li> <li>. identificateur du segment contenant les séries à recopier</li> </ul>  |
| idseg2  | <ul style="list-style-type: none"> <li>. identificateur de segment</li> <li>. par défaut, valeur contenue dans la variable globale SEGMENT</li> <li>. identificateur du segment destiné à contenir les séries à recopier</li> </ul>  |
| listser | <ul style="list-style-type: none"> <li>. liste d'identificateurs de série</li> <li>. pas de valeur par défaut</li> <li>. liste des identificateurs des séries à recopier</li> </ul>  |
| idcal   | <ul style="list-style-type: none"> <li>. identificateur</li> <li>. par défaut, valeur contenue dans la variable globale CALENDRIER ou sinon ANNUEL</li> <li>. identificateur du calendrier des séries à recopier</li> </ul>  |
| date1,  | <ul style="list-style-type: none"> <li>. constantes dates</li> <li>. par défaut, intervalle défini par DATEDEB, DATEFIN ou sinon intervalle de définition contenu dans le descripteur du segment idseg2</li> <li>. intervalle de dates sur lequel on recopie les séries</li> </ul> |
| date2   |  |

- motpasse . identificateur  
 . pas de valeur par défaut  
 . mot de passe consultation des séries à recopier
- options . options permettant d'indiquer si le calendrier et les  
 nomenclatures doivent être recopiés dans le segment  
 destinataire  
 . par défaut, pas d'argument OPTIONS équivaut à la non  
 recopie du calendrier et des nomenclatures  
 . sens des options :  
 - cal : le calendrier est recopié dans le segment  
 destinataire  
 - nomen : les nomenclatures sont recopiées dans la  
 base centrale

EXEMPLES

1 : COPSERIE SEGOR=SEGSERIES  
 SEGDT=SEGDEST  
 SERIE=(PNB,PIB)  
 DE= #1962 A= #1968;

Recopie des séries PNB et PIB du segment SEGSERIE sur l'inter-  
 valle 1962 à 1968 dans le segment SEGDEST

2 : COPSERIE SEGDT=SEGSERIES  
 SERIE=(ECHANGE,PRODAGRI)  
 CALENDRIER=TRIMESTRIEL  
 DE= #II/1960 A= #IV/1970  
 MOTPASSE=DUPONT  
 OPTIONS=NOMEN;

Recopie des séries ECHANGE et PRODAGRI de périodicité trimes-  
 trielle, de la base centrale dans le segment SEGSERIES. La  
 recopie est faite sur l'intervalle II/1960 à IV/1970, les  
 séries sont protégées en consultation. Les nomenclatures  
 associées sont recopiées.

### 3. COMMANDES COMMUNES BASE CENTRALE-SEGMENT

#### 3.1. Initialisation de variables globales

##### ENTETE

DEFBASE (abv=DFB)

Définit les valeurs de certaines variables communes. Commande l'impression de ces valeurs.

---

##### FORMAT

```
DEFBASE <SEGMENT=idseg>
      <CALENDRIER=idcal>
      <MOTPASSE=motpasse>
      <DATEDEB=dd>
      <DATEFIN=df>
      <SUPPORT=idsup>
      <OPTIONS=imp>;
```

---

##### ARGUMENTS

idseg	. identificateur de segment
	. pas de valeur par défaut
	. identificateur de segment à affecter à la variable globale SEGMENT
idcal	. identificateur de segment
	. pas de valeur par défaut
	. identificateur de calendrier à affecter à la variable globale CALENDRIER
motpasse	. identificateur
	. pas de valeur par défaut
	. identificateur du mot de passe à affecter à la variable globale MOTPASSE
dd	. constante date
	. pas de valeur par défaut
	. date début à affecter à la variable globale DATEDEB

- df
  - . constante date
  - . pas de valeur par défaut
  - . identificateur de support à affecter à la variable globale DATEFIN
- idsup
  - . identificateur
  - . pas de valeur par défaut
  - . identificateur du support d'impression
- options
  - . options permettant l'impression des valeurs des variables communes courantes de la base de données
  - . par défaut, pas d'argument OPTIONS équivaut à : pas d'impression
  - . sens des options :
    - imp : provoque l'impression des valeurs des variables communes courantes

EXEMPLES

```
1 : DEFBASE CALENDRIER=TRIMESTRIEL
      MOTPASSE=DUPONT
      DATEDEB=#III/1960
      DATEFIN=#IV/1970
```

Affectation de valeurs aux variables globales CALENDRIER, MOTPASSE, DATEDEB et DATEFIN. L'utilisateur peut, par la suite, effectuer un ensemble de commandes travaillant sur la base centrale, sur des séries protégées par le mot de passe DUPONT, de périodicité TRIMESTRIEL et définies de III/1960 à IV/1970. Il peut à tout moment changer la valeur de n'importe quelle variable globale en réexécutant une nouvelle commande DEFBASE ou demander l'impression des valeurs de ces variables (cf. exemple 2).

```
2 : DEFBASE OPTIONS=IMP;
      Demande d'impression des valeurs des variables globales
```

### 3.2. Impressions de séries

#### ENTETE

IMPSER (abv=ISR)

Imprime une série, un sous-ensemble de série, un ensemble de séries d'un segment ou de la base centrale

---

#### FORMAT

```
IMPSER <SEGMENT=idseg>
      <SERIE=(listser)>
      <CALENDRIER=idcal>
      <DE=date1> <A=date2>
      <MOTPASSE=motpasse>
      <SUPPORT=idsup>
      <OPTIONS=[doc/val]>
```

---

#### ARGUMENTS

- |         |  |
|---------|--|
| idseg   | <ul style="list-style-type: none"> <li>. identificateur de segment</li> <li>. par défaut, valeur contenue dans la variable globale SEGMENT ou sinon base centrale</li> <li>. identificateur du segment où sont contenues les séries à imprimer</li> </ul>  |
| listser | <ul style="list-style-type: none"> <li>. liste d'identificateurs de série</li> <li>. par défaut, si base centrale, impression de toutes les séries ayant pour périodicité celle indiquée par l'argument CALENDRIER si celui-ci est présent ou sinon la valeur prise par défaut</li> <li>. liste des identificateurs des séries à imprimer</li> </ul> |
| idcal   | <ul style="list-style-type: none"> <li>. identificateur</li> <li>. par défaut, si base centrale, valeur contenue dans la variable globale CALENDRIER ou sinon ANNUEL. Si segment, valeur contenue dans descripteur de segment</li> <li>. identificateur du calendrier auquel sont rattachées les séries à imprimer</li> </ul>                        |

- date1 . constante date  
 . par défaut, valeur de DATEDEB ou sinon date début des séries  
 . date à partir de laquelle les valeurs sont imprimées
- date2 . constante date  
 . par défaut, valeur de DATEFIN ou sinon date fin des séries  
 . date limite d'impression des valeurs
- motpasse . identificateur  
 . par défaut, valeur de la variable globale MOTPASSE ou sinon pas de valeur  
 . mot de passe consultation des séries
- idsup . identificateur  
 . par défaut, valeur de la variable globale SUPPORT ou sinon terminal  
 . identificateur du support d'impression
- options . options permettant d'indiquer si l'on désire l'impression de la partie documentaire ou de la partie valeurs des séries  
 . par défaut, impression de la totalité des informations concernant chaque série  
 . sens des options :  
 - doc : seule la partie documentaire des séries est éditée  
 - val : seule la partie valeur des série est éditée

EXEMPLES

1 : IMPSER;

Imprime toutes les séries non protégées, de périodicité ANNUEL de la base centrale. L'impression se fait au terminal et toutes les informations concernant les séries sont imprimées. Les valeurs des séries sont imprimées sur tout l'intervalle de définition.

```

2 : DEFBASE SEGMENT=SEGSERIES
      SUPPORT=IMPRIMANTE;

      IMPSER SERIE=(PIB,PNB)
      DE= #1962 A= #1968
      OPTIONS=VAL;

```

Impression des valeurs des séries PIB et PNB du segment  
SEGSERIES, de périodicité ANNUEL sur l'intervalle 1962 à 1968

```

3 : IMPSER SERIE=(ECHANGE(*,*,BLE),PRODAGRI(FRANCE));

```

Impression au terminal des séries de la base centrale ECHANGE  
et PRODAGRI de périodicité ANNUEL. Toutes les informations  
concernant ces séries sont imprimées. Seuls des sous-ensembles  
de ces séries sont imprimés :

- ECHANGE étant défini avec les nomenclatures PAYS, PAYS,  
PRODUIT. Seuls les échanges de BLE pour tout pays sont  
imprimés
- PRODAGRI étant défini avec la nomenclature PAYS. Seule la  
production agricole de la FRANCE est imprimée. Les séries  
sont imprimées sur tout leur domaine de définition

### 3.3. Impression de nomenclatures

#### ENTETE

```
IMPNOLEN (abv=INM)
```

Imprime une ou plusieurs nomenclatures d'un segment de séries donné  
ou de la base centrale

---

#### FORMAT

```

IMPNOLEN <SEGMENT=idseg>
          <NOMENCLATURE=listnomen>
          <SUPPORT=idsup>;

```

---

ARGUMENTS

- idseg . identificateur de segment  
 . par défaut, valeur contenue dans la variable globale  
 SEGMENT ou sinon base centrale  
 . identificateur du segment contenant les nomenclatures  
 à imprimer
- listnomen . liste d'identificateurs de nomenclatures  
 . par défaut, toutes les nomenclatures sont imprimées  
 . liste des nomenclatures à imprimer
- idsup . identificateur  
 . par défaut, valeur contenue dans la variable globale  
 SUPPORT ou sinon terminal  
 . identificateur du support d'impression

EXEMPLES

- 1 : IMPNOMEN;  
 Impression au terminal de toutes les nomenclatures de la base  
 centrale
- 2 : IMPNOMEN SEGMENT=SEGSERIES  
 NOMENCLATURES=(PAYS,PRODUIT)  
 SUPPORT=IMPRIMANTE;  
 Impression sur imprimante des nomenclatures PAYS et PRODUIT  
 du segment SEGSERIES

3.4. Impression de calendriersENTETE

IMPCAL (abv=ICL)

Imprime un calendrier défini d'un segment de séries donné ou un ou  
 plusieurs calendriers définis de la base centrale

---



FORMAT

```
IMPCAL <SEGMENT=idseg>
      <CALENDRIER=listcal>
      <SUPPORT=idsup>
```

ARGUMENTS

```
idseg      . identificateur de segment
           . par défaut, valeur contenue dans la variable globale
           . SEGMENT ou sinon base centrale
           . identificateur du segment contenant le calendrier à
           . éditer

listcal    . liste d'identificateurs
           . par défaut, si base centrale tous les calendriers
           . sont imprimés, sinon calendrier associé au segment
           . liste des identificateurs des calendriers à imprimer

idsup      . identificateur
           . par défaut, valeur contenue dans la variable globale
           . SUPPORT, sinon terminal
           . identificateur du support d'impression
```

EXEMPLES

```
1 : IMPCAL;
    Impression au terminal de tous les calendriers définis de la
    base centrale

2 : IMPCAL SEGMENT=SEGSERIES
    SUPPORT=IMPRIMANTE;
    Impression sur imprimante du calendrier défini du segment
    SEGSERIES
```

## 4. FONCTIONS MODULECO SUR LA BASE DE DONNEES

### 4.1. Extraction de tout ou partie des valeurs d'une série

#### ENTETE

VALSERIE (abv=VSR)

Fonction qui fournit en retour une, plusieurs ou toutes les valeurs d'une série d'un segment ou de la base centrale

---

#### FORMAT

VALSERIE (<segment>,série>,date1><,date2><,calendrier><,motpasse>)

---

#### ARGUMENTS

- |            |  |
|------------|--|
| segment    | . identificateur de segment  |
|            | . par défaut, valeur de la variable globale SEGMENT ou sinon base centrale                           |
|            | . identificateur du segment contenant la série à extraire  |
| série      | . identificateur de série  |
|            | . pas de valeur par défaut   |
|            | . identificateur de la série à extraire  |
| date1,     | . constantes dates   |
| date2      | . par défaut, intervalle défini par DATEDEB, DATEFIN ou sinon l'intervalle de définition de la série |
|            | . intervalle d'extraction de la série  |
| calendrier | . identificateur   |
|            | . par défaut, valeur contenue dans la variable globale CALENDRIER ou sinon ANNUEL                    |
|            | . identificateur du calendrier de la série à extraire  |
| motpasse   | . identificateur   |
|            | . pas de valeur par défaut   |
|            | . mot de passe consultation de la série à extraire   |

EXEMPLES

1 : SERIE=VALSERIE(SEGSERIES,PNB,#1960,#1960)

Affecte à la variable SERIE la valeur correspondant à 1960 de la série annuelle PNB du segment SEGSERIES

2 : ECHANGE=VALSERIE(,ECHANGE(FRANCE,ITALIE,BLE))

Affecte à la variable ECHANGE la série simple ECHANGE(FRANCE,ITALIE,BLE) de la série à trois dimensions ECHANGE de périodicité ANNUEL, définie dans la base centrale

4.2. Extraction de nomenclaturesENTETE

VALNOMEN (abv=VNM)

Fonction qui fournit les libellés d'une nomenclature d'un segment donné ou de la base centrale

FORMAT

VALNOMEN (nomenclature<,segment>)

ARGUMENTS

- |              |  |
|--------------|--|
| segment      | . identificateur de segment  |
|              | . par défaut, valeur de la variable globale SEGMENT ou sinon base centrale |
|              | . identificateur du segment contenant la nomenclature à extraire           |
| nomenclature | . identificateur de nomenclature   |
|              | . pas de valeur par défaut   |
|              | . identificateur de la nomenclature à extraire                             |

EXEMPLES

1 : PAYS=VALNOMEN(PAYS;SEGSERIES)

Affecte à la variable PAYS les libellés de la nomenclature  
PAYS du segment SEGSERIES

2 : NOMEN=VALNOMEN(PRODUIT)

Affecte à la variable NOMEN les libellés de la nomenclature  
PRODUIT de la base centrale

#### 4.3. Fourniture de la liste des identificateurs de nomenclature d'une série

ENTETE

NOMEN (abv=NM)

Fonction qui fournit la liste des identificateurs des nomenclatures  
d'une série d'un segment donné ou de la base centrale

FORMAT

NOMEN (série<,segment><,calendrier><,motpasse>)

ARGUMENTS

- |            |  |
|------------|--|
| série      | . identificateur de série  |
|            | . pas de valeur par défaut   |
|            | . identificateur de la série dont on veut connaître<br>les nomenclatures                               |
| segment    | . identificateur de segment  |
|            | . par défaut, valeur de la variable globale SEGMENT ou<br>sinon base centrale                          |
|            | . identificateur du segment où se trouve la série dont<br>on veut connaître les nomenclatures          |
| calendrier | . identificateur   |
|            | . par défaut, si base centrale, valeur contenue dans la<br>variable globale CALENDRIER ou sinon ANNUEL |
|            | . identificateur du calendrier auquel est rattachée la<br>série  |

EXEMPLES

1 : IDNOMEN=NOMEN(ECHANGE,SEGSERIES)

La variable IDNOMEN reçoit les identificateurs de nomenclatures PAYS, PAYS, PRODUIT de la série ECHANGE du segment SEGSERIES

2 : NOMENPROD=NOMEN(PRODAGRI)

Affecte à la variable NOMENPROD l'identificateur de nomenclature PRODUIT de la série PRODAGRI de la base centrale, de périodicité ANNUEL

#### 4.4. Fonction renvoyant le nombre de dimensions d'une série

ENTETE

DIMSER (abv=DSR)

Renvoie le nombre de dimensions d'une série d'un segment donné ou de la base centrale

FORMAT

DIMSER (série<,segment><,calendrier><,motpasse>)

ARGUMENTS

- |            |   |
|------------|---|
| série      | . identificateur de série   |
|            | . pas de valeur par défaut  |
|            | . identificateur de la série dont on veut connaître la dimension                                    |
| segment    | . identificateur de segment   |
|            | . par défaut, valeur contenue dans la variable globale SEGMENT ou sinon base centrale               |
|            | . identificateur du segment où se trouve la série dont on veut connaître la dimension               |
| calendrier | . identificateur  |
|            | . par défaut, si base centrale, valeur contenue dans la variable globale CALENDRIER ou sinon ANNUEL |
|            | . identificateur du calendrier auquel est rattachée la série dont on veut connaître la dimension    |

motpasse . identificateur  
 . pas de valeur par défaut  
 . mot de passe consultation de la série dont on veut  
 connaître la dimension

EXEMPLES

- 1 : ENT=DIMSER(ECHANGE,SEGSERIES)  
 ENT reçoit la valeur 3 qui est le nombre de dimensions de la  
 série ECHANGE
- 2 : SI DIMSER(PNB) # 0 ALORS  
 NOMEN(PNB)  
 PNB étant de dimension 0, il n'y a pas d'appel à la fonction  
 NOMEN

#### 4.5. Fonction renvoyant une liste d'identificateurs de calendriers pour une série donnée

ENTETE

BCALEND (abv=BCD)

Fonction renvoyant une liste d'identificateurs de calendriers pour  
 une série de la base centrale

FORMAT

BCALEND (série<,motpasse>)

ARGUMENTS

série . identificateur de série  
 . pas de valeur par défaut  
 . identificateur de la série dont on veut connaître les  
 identificateurs de calendriers

EXEMPLES

1 : CALENDRIER=BCALEND(PNB)

La série PNB étant de périodicité annuelle, la variable  
CALENDRIER reçoit l'identificateur ANNUEL

## BIBLIOGRAPHIE



Les références sont classées en quatre groupes :

- |   |      |
|---|------|
| 1. DOCUMENTS GENERAUX, OUVRAGES           | B.2  |
| 2. BIBLIOGRAPHIE SUR SYSTEMES SPECIFIQUES | B.9  |
| 3. PUBLICATIONS, OUVRAGES MODULECO        | B.13 |
| 4. NOTES INTERNES MODULECO                | B.15 |

1. DOCUMENTS GENERAUX, OUVRAGES

[J.R. Abrial et Al, 1970]

J.R. Abrial, J. Bas, G. Beaume, G. Henneron, R. Morin, G. Vigliano  
"Projet Socrate Spécifications Générales"  
Université de Grenoble, Août 1970

[J.R. Abrial et Al, 1972]

J.R. Abrial, J.P. Cahen, J.C. Favre, D. Portal, G. Mazare et  
R. Morin  
"Projet Socrate - Nouvelles Spécifications"  
(version 3)  
Université Scientifique et Médicale de Grenoble, Septembre 1972

[J.R. Abrial, 1974]

J.R. Abrial  
"Data Semantics"  
IFIP TC2 Working Conference, Cargese, Avril 1974

[M. Adiba et Al, 1976]

M. Adiba, C. Delobel  
"Les Modèles Relationnels de bases de données"  
IRIA - Sefi, Avril 1976

[M. Adiba, 1978]

M. Adiba  
"Un modèle relationnel et une architecture pour les systèmes de  
bases de données réparties -  
Application au projet Polyphème"  
Thèse de Doctorat d'Etat, Université de Grenoble, Septembre 1978

[M. Adiba et Al, 1978]

M. Adiba, J.Y. Caleca, C. Delobel, Ch. Euzet, D. Portal  
"Les systèmes de gestion de bases de données"  
I.F.C.I., Grenoble, Septembre 1978

[M. Adiba et Al, 1980]

M. Adiba, B.G. Lindsay

"Database snapshots"

IBM Research Report RJ 2772 3/7/1980

[ANSI/SPARC, 1975]

ANSI/X3/SPARC

"Study group on data base monoprocessor systems :  
interim report"

Bulletin of ACM-SIGMOD 7 n° 7, 1975

[M.M. Astrahan et Al, 1976]

M.M. ASTRAHAN et Al.

"System R : relationnal approach to data base management"

ACM - TODS Vol 1 n° 2 June 1976, pages 97-137

[F.M. Bancilhon et Al, 1977]

F.M. Bancilhon, N. Spyratos

"Protection of information in relationnal  
Data base managment systems"

Rapport de Recherche n° 243, INRIA, Août 1977

[R. Bayer et Al, 1977]

R. Bayer, K. Unterauer

"Prefix B. Trees"

ACM - TODS, Vol. 2 n° 1 March 1977, Pages 11-26

[M.W. Blasgen et Al, 1979]

M.W. Blasgen et Al.

"System R : an architectural update"

IBM Research Report RJ 2581 Juillet 1979

[S.A. Borkin, 1980]

Sheldon A. Borkin

"A Semantic Approach for Database Systems"

The MIT Press, Cambridge, Massachussets, and London, England, 1980

[J.Y. Caleca, 1978]

J.Y. Caleca

"Projet Polyphème : l'expression et la décomposition de transactions dans un système de bases de données réparties"

Thèse de Doctorat de 3ème cycle, Université de Grenoble, Septembre 1978

[CETE, 1978]

"Fonctionnalités du système MIISFIIT portable"

C.E.T.E Aix en Provence, Division Informatique et Gestion

Service Base de données, Septembre 1978

[CETE, 1978 B]

"Caractéristiques techniques du système MIISFIIT portable"

C.E.T.E. Aix en Provence, Division Informatique et Gestion

Service Base de données, Septembre 1978

[V. Chan Lam, 1979]

Vincent CHAN LAM

"Conception et réalisation d'une base de données ensembliste"

Thèse de Docteur Ingénieur, Université de Technologie de Compiègne,

Décembre 1979

[W.W. Chu et Al, 1979]

Westey W. Chu et Peter P. Chen

"Centralized and Distributed Data Base Systems"

First International Conference on Distributed Computing Systems,

Huntsville, Acabame, 1-4 Octobre 1979

[E.K. Clemons, 1979]

Erik k. Clemons

"Rational Data Base Standards : An examination of the 1978  
Codasyl DDLIC Report"

Information Systems Vol. 4, 1979, Pages 235-239

[E.F. Codd, 1970]

E.F. Codd

"A relational model of data for large shared data banks"

C.A.C.M., Volume 13 n° 6, June 1970, Pages 377-387

[E.F. Codd, 1971]

E.F. Codd

"Further Normalization of the Data Base Relational Model"

Courant Computer Symposia 6, New-York 1971

[E.F. Codd, 1971 B]

E.F. Codd

"Normalized Data Base Structure : A Brief Tutorial"

ACM - SIGFIDET Workshop - 1971

[C.J. Date, 1977]

C.J. Date

"An Introduction to Database systems"

(Second Edition), Addison Wesley Publishing Company, 1977

[C. Dufour, 1980]

Christian DUFOUR

"Codasyl : une norme souvent complexe"

01-Informatique n° 140, Mai 1980, Pages 51-59

[R. Fagin et Al, 1979]

R. Fagin, J. Nievergelt, N. Pippenger, H.R. Strong

"Extendible Hashing, a fast access method for dynamic files"

ACM - TODS Vol. 4 n° 3, September 1979, Pages 315-344

[J.M. Forrestier, 1979]

J.M. Forrestier

"Machine Relationnelle pour système segmenté"

Thèse de 3ème cycle, Université de Grenoble, Septembre 1978

[R. Fortier, 1975]

Robert Fortier

"Conception descendante de machine informatique"

Thèse de 3ème cycle, Université de Grenoble, Octobre 1975

[K. Jensen et Al, 1980]

K. Jensen, N. Wirth

"Pascal manuel de l'utilisateur"

Eyrolles, Paris 1980

[Journées Banques de données, 1971]

"Journées Banques de données"

AFCET, Aix en Provence, 16-18 Juin 1971

[D.E. Knuth, 1973]

D.E. Knuth

"The art of computer programming - sorting and searching"

Addison Wesley séries in computer science and information processing, 1973

[P.A. Larson, 1978]

P.A. Larson

"Dynamic Hashing"

BIT, Vol. 18, 1978, Pages 184-201

[W. Litwin, 1978]

W. Litwin

"Linear Virtual Hashing, a new tool for files and tables implementation"

IFIC TC-2 Working Conference on Data Base Architecture,

June 26-29, 1978, Venice, Italy

[W. Litwin, 1979]

W. Litwin

"Hachage virtuel : une nouvelle technique d'adressage de mémoires"

Thèse de Doctorat d'Etat, Université Pierre et Marie Curie

Paris 6, Mars 1979

[P. Maurice, 1978]

P. Maurice

"Pascal, un langage très évolué"

01-Informatique n° 125 Pages 59-63, Octobre 1978

[P. Maurice, 1978 B]

P. Maurice

"Pascal 80 Manuel d'utilisation"

INRIA, Décembre 1978

[L.H. Quam, 1980]

Lynn H. Quam

"A storage representation for efficient access to large multidimensional arrays"

SRI International, Technical note 220, April 1980

[B.W. Ravenel, 1979]

Bruce W. Ravenel

"Toward a Pascal Standard"

Language Resources, Inc, April 1979

[G.M. Scheider, 1979]

G. Michael Schneider

"Pascal : an overview"

University of Minnesota, April 1979

[M. Scholl, 1979]

M. School

"Performance analysis of new file organisations based on dynamic hash-coding"

R.R. n° 347, INRIA, Mars 1979

[SOCRATE]

"Socrate Manuel de présentation"

ERIA - ECA AUTOMATION

[N. Spyrtatos et Al, 1977]

N. Spyrtatos, F.M. Bancillon

"Name independance in relation\_ data base"

R.R. n° 241 INRIA, Août 1977

[H. Tardieu et Al, 1975]

H. Tardieu, H. Heckenroth, D. Nanci

"Etude d'une méthodologie d'analyse et de conception d'une  
base de données"

IAE Aix en Provence 1975

[N.G. Toan, 1977]

Nguyen Gia Toan

"URANUS une approche relationnelle à la coopération de bases de  
données"

Thèse de 3ème cycle, Université de Grenoble, Décembre 1977

[G. Vigliano, 1970]

G. Vigliano

"Projet Socrate Langages de Requetes"

Thèse de 3ème cycle, Université de Grenoble, Décembre 1970



2. BIBLIOGRAPHIE SUR SYSTEMES SPECIFIQUES

[Argos-C, 1977]

"Argos-C Manuel de présentation"

INSEE - Division Logiciel et Méthodes, Janvier 1977

[ARGOS-C, 1977 b]

"ARGOS-C Manuel de référence"

INSEE - Division Logiciel et Méthodes, Janvier 1977

[ARGOS-C, 1977 C]

"ARGOS-C Manuel du Statisticien"

INSEE - Division Logiciel et Méthodes, Janvier 1977

[BIBLOS, 1976]

"Biblos Manuel de l'utilisateur"

INSEE - Division Logiciel et Méthodes, Paris, Octobre 1976

[H. Eisenpress, 1975]

Harry Eisenpress

"An ideal econometric computer program"

IBM Corporate Headquarters, Armonk, N.Y., 1975

[Express]

"Express Reference Manual"

Management Decision System, Inc.

CEGOS - Tymshare, Saint-Cloud

[Groupe de travail Apache-Xing, 1978]

"Groupe de travail APACHE-XING" dans  
"Rapport final de la phase exploratoire du projet Moduleco-Annexes"  
Septembre 1978

[T. Harrison et Al, 1977]

Tim Harrison, Peter Smith  
"Hash, a program for econometric modelling"  
University of Southampton, May 1977

[Instruments de stratégies économiques, 1977]

"Les instruments de stratégie économiques et industrielles"  
Collection "études de politique industrielle" n° 16,  
ministère de l'industrie et de la recherche, documentation  
française, Paris 1977

[M.C. McCracken et Al, 1973]

M.C. McCracken, C.A. Sonnen  
"Databank Manual"  
Ottawa, Statistics Canada, January 1973

[M. Norman, 1977]

Morris Norman  
"Software package for economic modeling"  
RR-77-21, International Institute for Applied Systems Analysis,  
A-2361 Laxenburg, Austria, November 1977

[K. Plasser, 1978]

Klaus Plasser

"Brief Description of the IAS-System"

Symposium on Modelling and Simulation, Rehovot, Israel,

August 1978

[Ph. Richard et Al, 1979]

Ph. Richard, M. de Virville

"L'automatisation de la construction de tables de correspondance  
entre nomenclatures économiques"

R.A.I.R.O. Informatique/Computer Science

Vol 13 n° 4 1979 page 375-391

[SABINE]

"SABINE : implantation logique et physique"

CAP-France, Centre d'analyse et de programmation, Paris

[Système CRONOS, 1976]

"Système CRONOS pour la gestion des séries chronologiques

Manuel UO/3 Introduction"

Office statistique des communautés européennes, Bruxelles

Février 1976

[Système CRONOS, 1976 B]

"Système CRONOS - Manuel U/20/3 Consultation"

OSCE, Bruxelles, Février 1976

[Système CRONOS, 1976 C]

"Système CRONOS - Manuel U.1/0/3 langage de calcul"  
OSCE, Bruxelles, Février 1976

[Système CRONOS, 1976 D]

"Système CRONOS - Manuel U/11/03 Bibliothèque des fonctions  
de calcul"  
OSCE, Bruxelles, Février 1976

[Système CRONOS, 1976 E]

"Système CRONOS - Manuel U/12/2 Ajustements saisonniers"  
OSCE, Bruxelles, Février 1976

[Système OSIRIS, 1978]

"Système OSIRIS Manuel de l'utilisateur"  
Eurostat, division informatique, Bruxelles, Avril 1978

[Système TROLL, 1978]

"Le Système Troll" dans  
"Rapport final de la phase exploratoire du projet MODULECO-  
Annexes", Septembre 1978

[Troll, 1974]

"Troll Reference Manual"  
National Bureau of Economic Research, INC. Août 1974

### 3. PUBLICATIONS, OUVRAGES MODULECO

[C. Brunet et Al, 1979]

C. Brunet, V. Chari, B. Lepape, F. Rechenmann, A. Tarabout

"Pascal pour Moduleco ?"

Bulletin de liaison du sous-groupe Pascal, 5., AFCET, Paris, 1979

[D. Fouquet, 1979]

D. Fouquet

"Problèmes informatiques généraux et gestion de données" dans

"Méthodes mathématiques de la modélisation macroéconomique",

Les synthèses du SESORI, INRIA, Ed. P. MALGRANGE, Mai 1979

[D. Gabay et Al, 1978]

D. Gabay, P. Nepomiastchy, M. Rachdi, A. Ravelli

"Etude, Résolution et Optimisation de modèles macroéconomiques

Rapport de Recherche n° 312, INRIA, Juin 1978

[P. Malgrange, 1979]

"Méthodes mathématiques de la modélisation macroéconomique"

Les synthèses du SESORI, INRIA, Ed. P. Malgrange, Mai 1979

[P. Nepomiastchy et Al, 1978]

P. Nepomiastchy, B. Oudet, F. Rechenmann

"Moduleco, aide à la construction et à l'utilisation de modèles macroéconomiques"

Colloque International sur l'analyse et l'optimisation des systèmes, Versailles, 11-13 Décembre 1978

[P. Nepomiastchy et Al, 1981]

P. Nepomiastchy, F. Rechenmann

"The Equation Writing External Language of the Moduleco Software"

3<sup>rd</sup> Economics and Control Conference, Lyngby (Danemark),

June 22-24, 1981

[B. Oudet et Al, 1981]

B. Oudet, P. Nepomiastchy

"Le projet Moduleco"

Rapport de Recherche INRIA, Février 1981

[Rapport final, 1978]

"Rapport final de la phase exploratoire du projet Moduleco  
- Annexes", Septembre 1978

[F. Rechenmann et Al, 1978]

F. Rechenmann, A. Tarabout

"Présentation Générale du Système Moduleco" dans  
"La construction et l'utilisation des modèles macroéconomiques"  
Journées d'Etudes, Grenoble, 4-6 Octobre 1978

[G. Ruderman, 1978]

G. Ruderman

"Comparative study on systems currently available in the United  
States. for construction and use of macroeconomic models" dans  
"la construction et l'utilisation des modèles macroéconomiques",  
Journées d'Etudes Grenoble, 4-6 Octobre 1978

[B. Sefsaf, 1980]

B. Sefsaf

"Un logiciel de gestion de données chronologiques à plusieurs  
dimensions"

Deuxième conférence internationale sur "Bases de données dans les  
humanités et les sciences sociales"

Madrid, 16-19 Juin 1980

[A. Tarabout et Al, 1980]

A. Tarabout, F. Rechenmann, B. Sefsaf

"Moduleco : An integrated modular computer system for economists"  
3<sup>rd</sup> IFAC/IFORS Conference on Dynamic Modelling and Control of  
National Economies Warsaw/Poland, June 16-19 1980

4. NOTES INTERNES MODULECO

[R. Gardien, 1981]

R. Gardien

"Manuel de maintenance des entrées-sorties"

Moduleco - CSL, Réf. B13, Juin 1981

[R. Gardien et Al, 1981]

R. Gardien, F. Rechenmann

"Manuel de référence du langage Moduleco"

Moduleco-IMAG, réf. B11, Février 1981

[R. Gardien et Al, 1980]

R. Gardien, A. Tarabout, N. Vitry

"Spécifications fonctionnelles communes des noyaux"

Moduleco-IMAG, Réf. B01, Janvier 1980

[R. Gardien et Al, 1980 B]

R. Gardien, N. Vitry

"Organisation des entrées-sorties"

Moduleco-IMAG, Réf. G23, Mai 1980

[R. Gardien et Al, 1981]

R. Gardien, N. Vitry

"Spécifications Techniques des utilitaires systèmes"

Moduleco-IMAG, Réf. B14, Août 1981

[J.P. Grandjean et Al, 1980]

J.P. Grandjean, F. Sylvain

"Les algorithmes d'estimation dans la première version de Moduleco"  
Moduleco-INRIA-INSEE, Réf. G22, Février 1980

[H. Jayet, 1980]

H. Jayet

"Réalisation de variantes dans Moduleco"  
Moduleco-INSEE, Réf. G11, Janvier 1980

[H. Jayet, 1980 B]

H. Jayet

"Manuel de références simulation"  
Moduleco-INSEE, Réf. B05, Juin 1980

[P. Lespinasse, 1981]

P. Lespinasse

"Gestionnaire des objets de base"  
Moduleco-IMAG, Réf. B13, Juin 1981

[P. Lespinasse et Al, 1981]

P. Lespinasse, R. Gardien

"Edition, Création et Appel de Commande"  
Moduleco-IMAG, Réf. B12, Janvier 1981

[P. Nepomiastchy, 1979]

P. Nepomiastchy

"Langage Externe d'écriture des Equations"  
Moduleco-INRIA, Réf. C01, Juin 1979



[P. Nepomiastchy, 1981]

P. Nepomiastchy

"Manuel de références du système d'optimisation"

Moduleco-INRIA, Réf. B10, Février 1981

[B. Pincemaille, 1979]

B. Pincemaille

"Propositions pour la partie base de données de Moduleco"

Moduleco-DP, Réf. G02, Février 1979

[B. Pincemaille, 1979 B]

B. Pincemaille

"Interface Moduleco-ARGOS-C et Moduleco-APACHE"

Moduleco-DP, Réf. A06, Juin 1979

[B. Pincemaille, 1979 C]

B. Pincemaille

"L'estimation dans Moduleco"

Moduleco-DP, Réf. A07, Août 1979

[B. Pincemaille, 1980]

B. Pincemaille

"Exemples de lancement de simulation"

Moduleco-DP, Réf. G15, Février 1980

[F. Rechenmann, 1979]

F. Rechenmann

"Analyse, Résolution, Optimisation : Spécifications générales"

Moduleco - IMAG, Réf. A03, Mars 1979

[F. Rechenmann, 1979 B]

F. Rechenmann

"Le sous-système d'estimation"

Moduleco - IMAG, Réf. A11, Novembre 1979

[F. Rechenmann, 1980]

F. Rechenmann

"Liste des fonctions Moduleco"

Moduleco - IMAG, Réf. G28, Mars 1980

[F. Rechenmann, 1980 B]

F. Rechenmann

"Système de modélisation

1ère partie : organisation des objets rattachés aux modèles

2ème partie : structures de données"

Moduleco - IMAG, Réf. B04, Mai 1980

[F. Rechenmann, 1980 C]

F. Rechenmann

"Manuel de référence édition et manipulation de modèles"

Moduleco - IMAG, Réf. B07, Juillet 1980

[F. Rechenmann et Al, 1980]

F. Rechenmann, B. Sefsaf

"Spécifications complémentaires du langage Moduleco"

Moduleco - IMAG, Réf. G13, Janvier 1980

[B. Sefsaf, 1979]

B. Sefsaf

"Spécifications générales définitives de la base de données  
de Moduleco et du langage de manipulation associé"

Moduleco - IMAG, Réf. A09.1, Décembre 1979

[B. Sefsaf, 1980 B]

B. Sefsaf

"Spécifications techniques de la base de données de  
Moduleco et des commandes associées"

Moduleco - IMAG, Réf B03, Mai 1980

[B. Sefsaf, 1980 C]

B. Sefsaf

"Manuel de référence base de données"

Moduleco - IMAG, Réf. B06, Juillet 1980

[B. Sefsaf et Al, 1979]

B. Sefsaf, F. Rechenmann

"Evolution de la base de données et des segments -  
Les liaisons modèles-données"

Moduleco - IMAG, Réf. G10, Novembre 1979

[A. Tarabout, 1979]

A. Tarabout

"Structure générale du système Moduleco"

Moduleco - IMAG, Réf. A02, Octobre 1979

[A. Tarabout, 1980]

A. Tarabout

"Spécifications techniques de la machine de base"

Moduleco - IMAG, Réf. B02, Juin 1980

[N. Vitry, 1981]

N. Vitry

"Les segments temporaires"

Moduleco - IMAG, Réf. B15, à paraître

A U T O R I S A T I O N   D E   S O U T E N A N C E

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

VU le rapport de présentation de

Monsieur OUIDET, Professeur à l'USMG

Monsieur SEFSAF Belkacem

est autorisé à présenter une thèse en soutenance pour l'obtention du diplôme de  
DOCTORAT DE TROISIEME CYCLE, Spécialité "Génie Informatique".

Fait à Grenoble, le 26 octobre 1981

Le Président de l'I.N.P.-G.

**D. BLOCH**  
Président  
de l'Institut National Polytechnique  
de Grenoble

