76/667

UNIVERSITE
DE NANCY 1

UER
SCIENCES MATHEMATIQUES

Sc. N. 76/66 A

L'AIDE A LA CONCEPTION DANS LE PROJET REMORA

UNIVERSITE NANCY

THESE

BIBLIOTHEQUE SCIENCES NANCY 1
D 095 182324 6

pour l'obtention
du doctorat de spécialité
mathématiques appliquées

soutenue le 29/9/1976 par
odile THIERY

Jury :

M. GRIFFITS	Président
G. BENCI	Examinateur
JC. DERNIAME	Examinateur
O. FOUCAUT	Examinateur
C. ROLLAND	Examinateur

UNIVERSITE
DE NANCY 1

UER
SCIENCES MATHEMATIQUES

L'AIDE A LA CONCEPTION DANS LE PROJET REMORA

THESE

pour l'obtention
du doctorat de spécialité
mathématiques appliquées

soutenue le 29/9/1976 par
odile THIERY

Jury :

M. GRIFFITS	Président
G. BENCI	Examinateur
JC. DERNIAME	Examinateur
O. FOUCAUT	Examinateur
C. ROLLAND	Examinateur

Je remercie vivement Monsieur le Professeur M.GRIFFITHS pour l'honneur qu'il me fait de présider le jury.

Que Monsieur J.C.DERNIAME trouve ici l'expression de ma gratitude pour la formation en Informatique que je lui dois et sa participation au jury.

Je remercie Monsieur G.BENCI pour l'intéret qu'il a témoigné à ce travail et l'honneur qu'il me fait en participant au jury.

Madame ROLLAND, Directrice de l'UER de Mathématiques de NANCY II, est à l'origine du projet REMORA dans lequel s'insère ce travail.
Je tiens à lui exprimer ma profonde reconnaissance pour avoir su, par ses conseils et encouragements, me faire progresser dans l'élaboration de cette thèse et, de façon générale, dans ma conception de la recherche en Informatique.

J'adresse mes remerciements à Madame O.FOUCAUT pour l'aide qu'elle m'a prodiguée et pour l'honneur qu'elle m'a fait d'être membre du jury.

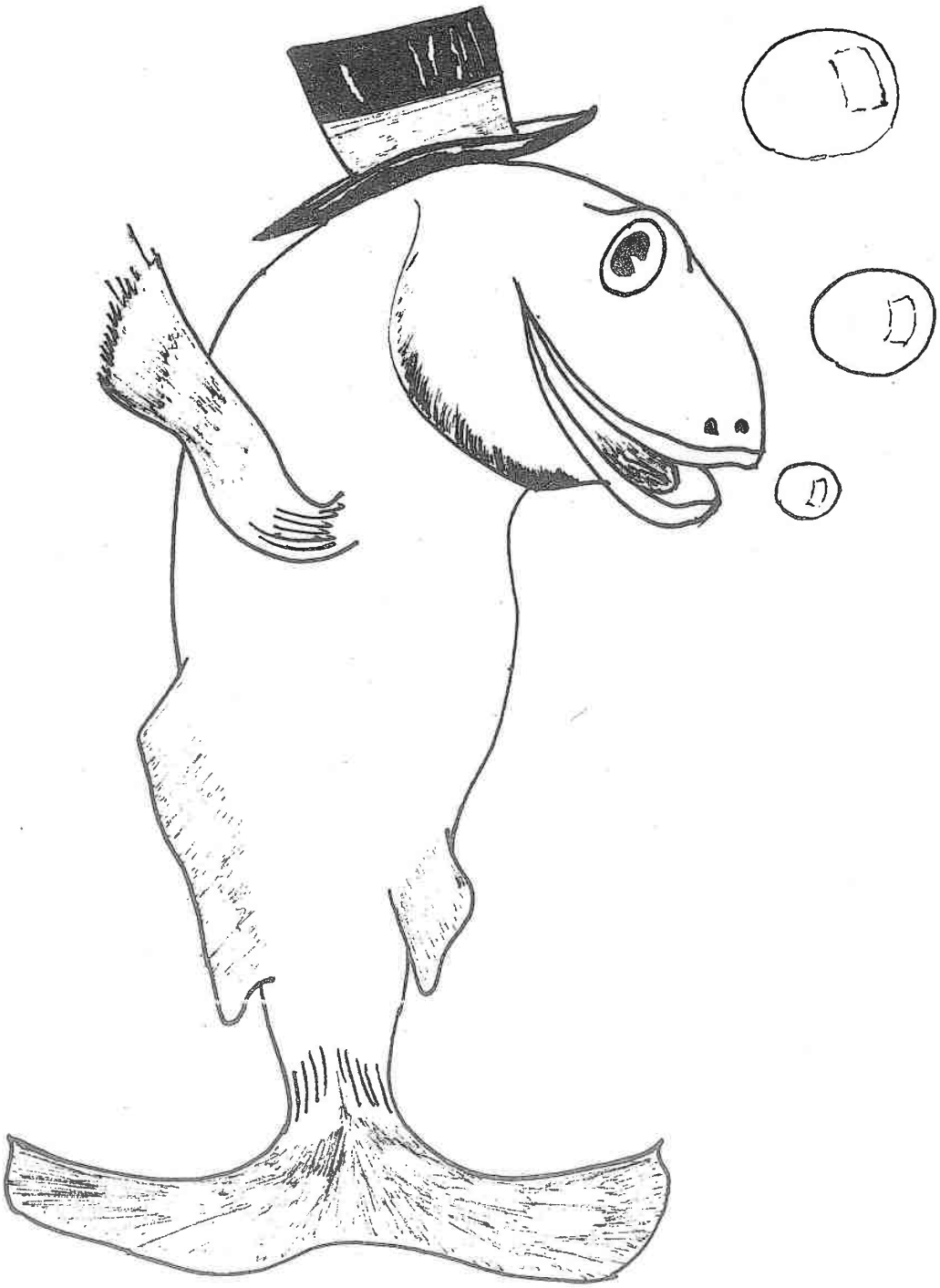
Je tiens à remercier particulièrement Mesdames C.ROLLAND et O.FOUCAUT pour les qualités de coeur et la grande disponibilité dont elles ont su faire preuve.

J'ai beaucoup apprécié le climat régnant au sein du groupe REMORA.
Que chacun soit remercié pour l'amitié et l'entr'aide que l'on y trouve.

Je remercie Messieurs J.P.FINANCE et J.THIERY pour l'aide qu'ils m'ont apportée dans la formalisation mathématique des algorithmes.

Je tiens à remercier Mesdames M.QUINTA et N.GAUTIER , Monsieur J.THIERY sans qui la réalisation matérielle de cette thèse n'aurait pas été possible.

L'AIDE A LA CONCEPTION DANS REMORA



SOMMAIRE

Pages

CHAPITRE O - INTRODUCTION

I Origine du projet I

2 Objectif poursuivi dans le projet II

3 Caractéristique des éléments du système IV

3-I La méthode IV

3-2 Les modèles V

3-3 Le langage VII

3-4 Les outils VIII

4 Etat d'avancement des travaux et développement à court terme XI

5 Notre contribution au projet REMORA XIII

6 Conclusion XIV

CHAPITRE I - LE SCHEMA CONCEPTUEL

I-0 Introduction I

I-I Objectifs et qualités du schéma conceptuel 4

I-2 L'approche du monde réel au sens de REMORA 5

I-2-I Les éléments constitutifs du monde réel à représenter dans le SI 5

I-2-2 Définition des éléments constitutifs du monde réel 7

I-3 Choix d'une structure conceptuelle : représentation du monde réel 9

I-4 Désignation des objets II

I-5 Etude détaillée des objets du schéma conceptuel I3

I-5-I Les objets et leurs caractéristiques I3

a-Définition

b-La classe de vocabulaire

c-Les caractéristiques

I-5-I-I L'application I3

a-Définition

b-La classe de vocabulaire

c-Les caractéristiques

I-5-I-2 La procédure I5

a-Définition

b-La classe de vocabulaire

c-Les caractéristiques

Pages

I-5-I-3 Le module 16

a-Définition

b-La base de vocabulaire

c-Les caractéristiques

I-5-I-4 Le mot 17

a-Définition

b-La base de vocabulaire

c-Les caractéristiques

I-5-I-5 Le lot 20

a-Définition

b-La base de vocabulaire

c-Les caractéristiques

I-5-I-6 L'entité 22

a-Définition

b-La classe de vocabulaire

c-Les caractéristiques

I-5-2 Les relations représentées dans le schéma conceptuel 22

I-5-2-I Relations entre applications et procédures 23

I-5-2-2 Relations entre lots et procédures 24

I-5-2-3 Relations entre modules et procédures 24

I-5-2-4 Relations entre lots et mots 25

I-5-2-5 Relations entre modules et mots 26

I-5-2-6 Relations entre mots et procédures 27

I-5-2-7 Liens mots-mots 27

I-5-2-8 Liens module-module 28

I-5-2-9 Relations entité-entité 30

I-5-2-IO Liens mots-entité 30

CHAPITRE II - LA GESTION DU SCHEMA CONCEPTUEL DANS REMORA

II-0 Introduction I

II-I Reconnaissance des objets et des relations 2

II-I-I Correspondance entre types d'objet monde réel-schéma conceptuel 2

II-I-2 Détermination des éléments 2

II-I-3 Reconnaissance des objets et des caractéristiques 4

II-I-3-I Les algorithmes de dépouillement 4

II-I-3-2 Les algorithmes de calcul 5

II-I-3-2-I Le genre d'un mot 5

II-I-3-2-2 Le type d'un module 5

Pages

A- Les étapes du déroulement de l'algorithme 6

B- L'algorithme 6

C- Exemple 6

II-I-3-2-3 La nature d'un mot 8

A- Les étapes du déroulement 8

de l'algorithme 8

B- Déroulement de l'algorithme I0

B-I Caractéristiques I0

B-2 Description modulaire de l'algorithme I0

C- Exemples I5

II-I-4 Les relations I9

II-I-4-I Nature des relations détectées I9

II-I-4-2 L'algorithme 20

A- Déscription modulaire 20

B-Exemples 22

II-2 Choix de représentation de la structure conceptuelle 25

CHAPITRE III - L'AIDE A LA CONCEPTION DU SYSTEME D'INFORMATION

III-0 Introduction 1

III-I L'aide à la conception 2

III-I-I Typologie des méthodes de conception 2

III-I-I-I Qui aider?

III-I-I-2 Comment aider?

III-I-2 Les caractéristiques du procédé de conception retenu et les erreurs qu'il engendre 5

III-I-3 Les fonctions d'aide à la conception 7

III-2 Analyse détaillée des fonctions 9

III-2-I La fonction controle 9

III-2-I-I Définition 9

III-2-I-2 Les controles et les moyens envisageables pour l'assurer I0

a-Les controles sémantiques globaux de l'énoncé

b-Les controles sémantiques de fidèlité

c-Les controles sémantiques sur les éléments

d-Les controles syntaxiques

Pages

III-2-2 La fonction intégration 14

III-2-2-I Définition 14

III-2-2-2 Les controles et les moyens envisageables pour l'assurer 16

a-Les controles sémantiques sur les éléments

aI-La détection des synonymies implicites

a2-La détection des polysèmes

b-Les controles globaux sur la cohérence du nouveau SC

c-Les controles de fidélité

d-Les controles de complétude

III-2-3 La fonction documentation 23

III-2-3-I Définition 23

III-2-3-2 Les moyens de mise en oeuvre envisageables 23

a-Quel type d'informations doit-on mettre dans le stock documentaire?

b-Comment gérer le stock documentaire?

c-Quels moyens peut-on fournir pour l'interrogation du stock documentaire?

CHAPITRE IV - LA PHASE DE CONCEPTION DANS REMORA

IV-0 Introduction I

IV-I L'ensemble des fonctions assurées par un système de conception tel que REMORA 2

IV-2 Analyse des outils de la phase de conception de REMORA 4

IV-2-I Le générateur 6

A- Fonction : transformation des définitions

B- Algorithme

IV-2-2 Le simulateur d'édition 4

A- Fonction : aide à la conception

B- Algorithme

IV-2-3 Le documenteur 4

A- Fonction : aide à la conception

B- Algorithme

Pages

IV-2-4 L'éditeur 7

A- Fonction : aide à la conception

B- algorithme

IV-2-5 Les opérateurs de classification 7

A- Fonction : construction BO

B- Algorithme

IV-2-6 Le pilote 7

A- Fonction : pilotage

B- Algorithme

IV-2-7 Le gestionnaire de base 8

A- Fonction : construction BO

B- Algorithme

IV-2-8 Les analyseurs 8

A- Fonctions : { - reconnaissance des objets de la BO
- controle
- integration }

AI-Reconnaissance des objets de la BO

A2-Controle

A3-Integration

B- Algorithmes

BI-Reconnaissance des objets de la BO

B2-Controle

B2I-Controle syntaxique

B22-Controle sémantique

B3-Integration

B3I-Le module

B32-Le mot

IV-3 Mise en oeuvre des outils dans le déroulement de la phase de conception I7

IV-3-0 Introduction I7

IV-3-I Survol de la mise en oeuvre I8

IV-3-2 Présentation de l'exemple 20

IV-3-3 Déroulement des taches d'aide à la conception sur l'exemple 22

IV-3-3-I Première étape "Connaissance du contexte" 22

IV-3-3-2 Deuxième étape "Définition du problème" 28

IV-3-3-3 Troisième étape "Prise en charge des descripteurs de traitement par le système" 30

Pages

A-Les controles syntaxiques

B-Analyse des objets nouveaux

IV-3-3-4 Quatrième étape

"Détection et correction des incohérences" 36

IV-3-3-5 Cinquième étape

"Visualisation du nouvel état conceptuel" 40

CHAPITRE V - CONCLUSION

CHAPITRE 0

INTRODUCTION

LE PROJET REMORA - PRESENTATION GENERALE

1 - ORIGINE DU PROJET

Les difficultés rencontrées dans l'application des méthodes traditionnelles pour la conception et la réalisation d'automatisations concrètes issues du domaine de la gestion des organisations nous ont incité à chercher une méthodologie d'approche et d'appréhension des problèmes qui redonne à la conception sa primauté et à partir de laquelle puisse être assurée la construction du système d'informations.

Dans ce domaine qu'est l'organisation, certains pensent que la finalité du système d'informations est d'être vu comme un modèle aussi fidèle que possible de l'organisation, qui soit capable de fournir à tout instant et dans les délais compatibles avec l'exigence des gestionnaires, toutes les informations dont ils ont besoin (20,25,37,48...)

D'autres considèrent que la finalité du système d'informations est de satisfaire les besoins définis dans l'organisation en termes de traitements, pour la production de messages résultats. (38,31,14,39,41,53...)

Ces deux finalités ne nous semblent pas aussi opposées que beaucoup le déclarent. En effet, si l'on cherche à satisfaire le premier objectif on remplit les conditions de réalisation du second, le système d'informations contenant, par définition, toutes les informations nécessaires à la satisfaction des besoins des utilisateurs. De même si on se fixe le second, on construit progressivement un modèle de l'organisation qui est enrichi chaque fois qu'un nouveau besoin est exprimé, ce qui permettra d'atteindre à long terme le premier objectif.

Cependant, ces deux façons d'aborder les problèmes ont conduit à des méthodes différentes :

. dans le premier cas "l'approche par les données" consiste à aborder la conception du système d'informations par la conception des bases de

données (3,37,21,17)

. Dans le second, "l'approche par les traitements" consiste à prendre comme point de départ les besoins des utilisateurs et à construire un système qui permette la satisfaction de ces besoins.

Les méthodes du 2ème type (27,41 ,47,56,58,43,57,66) n'ont pas su réaliser l'intégration progressive des nouveaux besoins dans le contexte de ceux qui avaient déjà été traités, conduisant à des systèmes rigides n'évoluant que par adjonction.

Les approches du premier type n'ont pas, à l'heure actuelle, proposé de méthode de conception de la base de données satisfaisante.

Notre opinion est que dans le domaine des organisations auquel nous nous intéressons et en particulier dans le sous-ensemble des procédures administratives, les besoins sont clairement définissables.

Notre approche est donc une approche par les besoins mais qui, prenant simultanément en compte l'analyse de l'aspect données et de l'aspect traitements, assure la construction progressive et intégrée du système d'informations.

2 - OBJECTIF POURSUIVI DANS LE PROJET REMORA

En d'autres termes, notre objectif est de définir un système de conception et de construction d'un système d'informations qui, à partir d'une définition externe des problèmes de gestion permette l'élaboration du système d'informations organisé en base de données et base de programmes.

La définition externe d'un problème est une définition des besoins du gestionnaire.

Celle du modèle de conception revient, à notre sens, à la définition d'un quadruplet (modèles, méthode, langage, outils) en s'appuyant sur l'hypothèse suivante :

"la base de données et la banque de programmes sont conçues et construites progressivement et par intégration à partir de la définition externe des besoins exprimés par le gestionnaire".

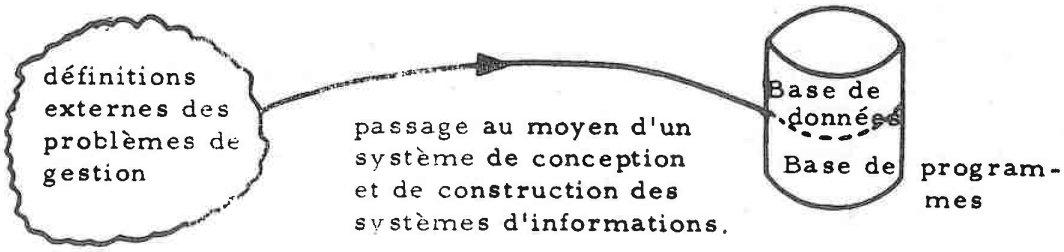


Figure 1

Chacun des éléments de ce système joue un rôle spécifique :

la méthode définit les étapes à franchir pour assurer le passage des définitions externes des problèmes de gestion au système d'informations automatisé, et donne pour chacune d'elles, un ordre logique d'étude des objets à y décrire et des choix à y effectuer.

Les modèles fournissent, pour chaque niveau de définition du système d'informations associé à chacune des étapes de la méthode de conception les concepts permettant de le décrire.

Le langage offre les moyens de l'expression des définitions des concepts.

Les outils visent à assister le concepteur en contrôlant la cohérence de ses définitions et en le remplaçant chaque fois que cela est possible.

Ces éléments sont cependant très dépendants les uns des autres ; en effet, les modèles sont associés aux étapes de la méthode, le langage utilise leurs concepts et les outils ont des fonctions liées à la fois à la méthode, aux modèles et au langage ; on retrouve leurs équivalents dans tout système de conception assisté par ordinateur (53,66)

3 - CARACTERISTIQUES DES ELEMENTS DU SYSTEME

3.1 La méthode

Nous proposons d'effectuer le passage de la définition externe d'un problème de gestion à sa définition programmée en trois étapes : phase de conception, phase de structuration, phase d'implémentation et composition.

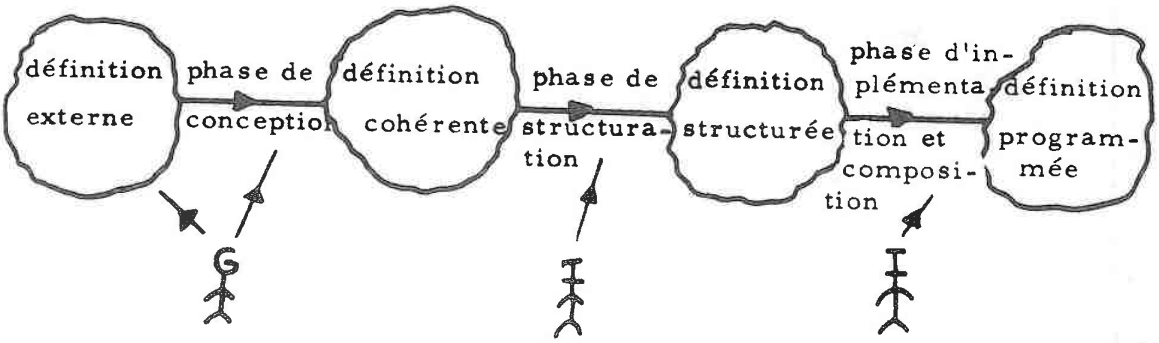


Figure 2 : Phases de la conception

- La phase de conception est celle ou l'utilisateur définit son problème. Il souhaite pouvoir le faire par un énoncé c'est-à-dire en exprimant ce qu'il veut voir réalisé indépendamment de la façon dont ce sera effectivement résolu. Dans notre méthode d'approche un problème étant défini par ses résultats nous proposons que cet énoncé soit fait par les définitions de ces résultats. L'objectif de la phase de conception est alors de parvenir à une définition complète et cohèrente du problème à partir de ses définitions. Le modèle conceptuel d'expression des besoins (59,60) est un moyen d'atteindre ce but.

- La phase de structuration est caractérisée par la prise en compte de paramètres concernant les choix des accès, qui seront utilisés dans le système d'informations pour atteindre les résultats définis dans la phase de conception. Il s'agit essentiellement du choix d'accès à une information par

"donnée" ou par "programme". C'est d'après ces choix que peuvent alors être structurés les données et les traitements ce qui conduit à la définition structurée du problème.

- La phase d'implémentation et composition correspond à la prise en compte des choix d'implémentation physique des structures de données et des structures de traitement définis au niveau précédent. C'est le niveau où le système d'informations est organisé en base de données et base de programmes.

- Ces trois phases sont dans une certaine mesure indépendantes les unes des autres : la modification d'un paramètre d'un niveau ne remet pas en cause les définitions effectuées au niveau précédent.

La méthode autorise le passage d'un niveau à un autre en déroulant des algorithmes qui s'appuient sur les correspondances entre modèles et en tenant compte des paramètres de choix fournis par les interlocuteurs humains.

3.2 Les modèles

Chaque phase de la méthode conduit à un niveau intermédiaire de définition du système d'informations. Pour chacun de ces niveaux nous proposons un modèle c'est-à-dire un ensemble de concepts qui lui sont spécifiques.

Dans le champ d'une application de gestion un problème nous semble défini dans le monde réel par

un événement

et des besoins

Les besoins sont décrits par des

messages

qui sont l'expression des

propriétés faisant l'objet de définitions

des

objets manipulés

et de leurs associations.

A ce modèle externe nous associons un schéma conceptuel qui définit les objets permettant de donner une description du système d'informations à l'issue de la phase de conception. La figure n° 3 met en évidence les objets de ce schéma et la correspondance qui existe avec ceux du monde réel.

événement ⟶ procédure
message ⟶ lot
définitions ⟶ modules
propriétés ⟶ mots
objets ⟶ entités
associations ⟶ relations

Figure 3

On trouve dans la présente thèse une analyse détaillée de ce schéma qui nous semble suffisamment général pour pouvoir être utilisé en dehors du contexte de Remora dans toute méthode de conception intégrant les aspects "données" et "traitements".

A la phase de structuration correspond un autre modèle : le schéma interne. Les concepts associés à ce schéma représentent des éléments du système d'informations automatisé : les unités, les algorithmes programmés et les structures. Le passage du schéma conceptuel au schéma interne est rendu possible par l'existence de règles de correspondance entre les objets des deux schémas ou par l'application de règles de regroupement définissant un objet du schéma interne comme un agrègat d'objets du schéma conceptuel, ainsi que le montre la figure 4.

	Schéma interne		Schéma conceptuel
Un	algorithme programmé	correspond à une	procédure
Une	unité	est un agrègat de	modules
Une	structure	correspond à un ou est un agrègat de	lot mots, entités, relations

Figure 4

3.3 Le langage

- Le langage du système de conception doit permettre dans la phase de conception l'expression des énoncés des problèmes et celles des paramètres dans les autres phases. L'expression des paramètres peut se faire de façon classique et ne mérite pas de définition de langage particulier.

En revanche, nous avons recherché ce que pourrait être le langage de la phase de conception. Dans d'autres systèmes ou méthode de conception (4,53,66) nous rencontrons des langages qui se composent d'un ensemble de règles de description associées à chaque objet du modèle retenu. Un langage de ce type ne nous apparaît pas adapté à une conception progressive à partir de l'expression des besoins des utilisateurs. C'est pourquoi nous proposons dans notre langage de conception de décrire et de définir les objets du problème réel à l'aide des concepts du schéma conceptuel.

Ainsi, le langage permet la description des messages sur des descripteurs de documents et l'expression de la définition des résultats apparaissant sur les messages sur des descripteurs de traitement. Ces définitions permettent d'introduire tous les objets du schéma-conceptuel mais non toutes leurs caractéristiques. Certaines seront déduites par les outils du système, d'autres concernant la dynamique des données seront fournies par l'utilisateur sur un descripteur de mot.

La partie du langage concernant la description des messages et la définition des résultats est décrite dans (36, 59, 60, 54) celle concernant la dynamique des données est en cours d'étude.

Les caractéristiques du langage sont en fait très liées à celle de la méthode d'analyse du modèle conceptuel d'expression des besoins.(59, 60, 54).

C'est une méthode d'analyse par étapes qui consiste à construire pour le résultat à définir un arbre de décomposition en intermédiaires.

définis à leur tour ; il lui correspond un langage déclaratif ou les définitions apparaissent dans l'ordre des résultats de l'arbre qui n'est pas l'ordre du calcul correspondant.

Cette analyse étant modulaire l'expression des définitions dans le langage est elle-même modulaire.

De plus, l'analyse s'appuie sur une typologie des résultats les classant en résultats inconditionnels, résultats conditionnels, et liste de résultats ; à laquelle est associée une typologie équivalente des définitions explicites dans le langage.

C'est enfin, un langage généré par les outils dans un langage de programmation dont le choix est fait par l'informaticien à la phase d'implémentation.

3.3 Les outils

Nous nous sommes posé la question de savoir si "l'on peut ramener la construction d'un système d'informations à sa seule définition à un niveau conceptuel, un automate assurant ensuite toutes les tâches nécessaires à la réalisation effective ?"

Contrairement aux objectifs fixés dans le projet Isdos (49,66) émettant la possibilité d'effectuer les meilleurs choix possibles et de manière automatique à chaque niveau, nous avons préféré choisir comme le rapporte WATERS (70) ou (18,28,53) une méthode de conception assistée par ordinateur.

* Le pilotage

Cette assistance est en fait un pilotage du concepteur réalisé par un outil privilègié le pilote qui joue un rôle de coordonnateur entre les différents éléments du système de conception ; il s'appuie sur la méthode et les modèles dont il a la connaissance et sur un ensemble d'outils regroupés dans un automate. Lors de l'entrée dans le système de toute définition

ou de tout paramètre il lance les outils chargés de les analyser ou de les stocker, et communique en retour des anomalies ou les conclusions déterminées par les outils. Il produit également toute la documentation sur le système dans l'état de construction où il se trouve.

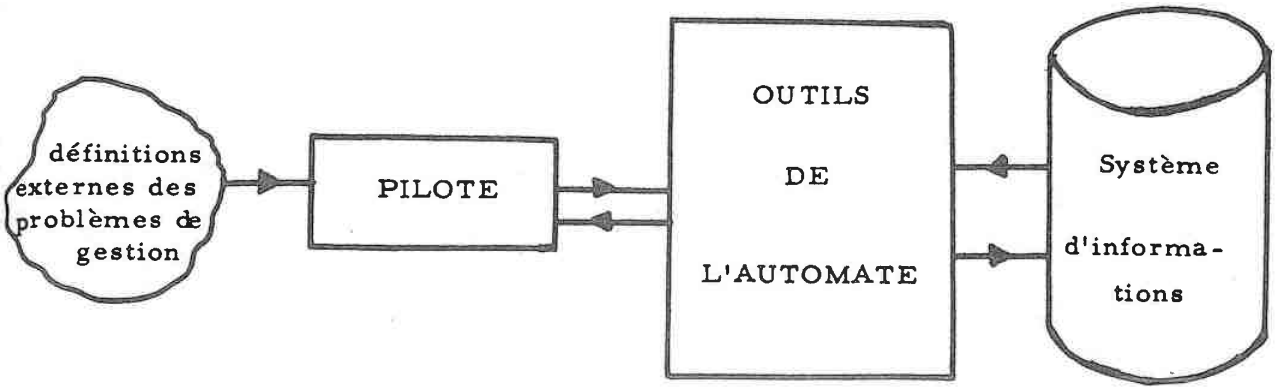


Figure 5 : Le pilotage

* Les fonctions des outils de l'automate

Suivant le niveau auquel ils interviennent les outils assurent des fonctions différentes :

- les outils d'aide à la conception apportent dans la première phase la connaissance du contexte dont il a besoin pour poursuivre sa définition du problème et simulent pour lui son exécution à l'instant ou il le décrit. Ainsi, par exemple, l'analyseur vérifie la cohérence du vocabulaire de l'application, le simulateur d'état imprimé permet par la production de maquettes une mise au point aisée des documents qui seront produits par le système d'informations.

- Les outils d'aide au choix et à la structuration en réalisant des classifications des objets définis dans la phase de conception guident le concepteur dans la détermination des choix concernant les accès et la structuration des informations et des traitements.

Ainsi les opérateurs de classification développent des raisonnements déductifs qui évaluent certaines propriétés des mots représentant les données de l'application, et en proposent une classification. L'architecte déduit des paramètres d'accès proposés par le concepteur une structuration des données et des traitements qui lui est soumise pour approbation ou modification.

- Les outils d'aide technique exécutent de manière automatique des tâches technologiques incombant normalement aux analystes ou aux programmeurs.

Ainsi, le générateur réordonne les définitions désordonnées d'un problème et les traduit en programmes exécutables.

- Les opérateurs de datamatique permettent l'évaluation automatique des formats de toutes les informations ayant servi d'intermédiaires dans la définition d'un problème et de tous ses résultats.

* Implémentation des schémas conceptuel et interne

Le fonctionnement de tous ces outils nécessite la connaissance du système d'informations au niveau de conception ou de construction où ils interviennent. C'est pourquoi le système comporte deux images du système d'informations s'appuyant sur les schémas conceptuel et interne. Elles sont organisées en base de données et sont appelées base des objets élémentaires et base des objets agrégés. La base des objets élémentaires contient les occurences, les caractéristiques et les relations de tous les objets utilisés pour décrire le problème dans la phase de conception. Ces objets sont ceux du schéma conceptuel donnés ci-dessus.

La base des objets agrégés contient suivant le même principe, les objets décrivant le système d'informations à l'issue de la phase de structuration.

- La base des objets élémentaires est construite progressivement par l'outil gestionnaire à partir des résultats communiqués par l'analyseur

qui dépouille les descripteurs, supports du langage de conception.

- La base des objets agrégés est construite par l'architecte et par les opérateurs de classification à partir de la base des objets élémentaires et à partir des paramètres de la phase de structuration.

La figure numéro 5 montre l'intervention des outils et des bases dans la construction du système d'informations.

4 - ETAT D'AVANCEMENT DES TRAVAUX ET DEVELOPPEMENT A COURT TERME

Les caractéristiques générales que nous venons de présenter pour chaque composant du système (modèles, méthode, langage, outils) laissent entrevoir l'ampleur du projet Remora.

L'importance des études à mener pour parvenir à une définition complète du système est telle que nous avons été conduits à choisir des domaines à approfondir.

Nous nous sommes fixés comme objectif à court terme (2 ans) de définir la partie du système correspondant à la phase de conception, en d'autres termes de proposer les concepts associés à ce niveau dans une conception intégratrice des aspects "données" et "traitements", la méthode pour les appréhender, le langage pour les décrire et les outils nécessaires à une réalisation cohérente et facile.

Le sous-ensemble déjà défini de ce modèle et que nous nommons modèle conceptuel d'expression des besoins à permis de proposer un schéma conceptuel décrit dans cette thèse ; une méthode d'approche des problèmes dans la thèse de W. PEREA (54) ; un langage explicité dans la thèse de D. LAMY (36) et des outils d'aide à la conception détaillés dans celle-ci.

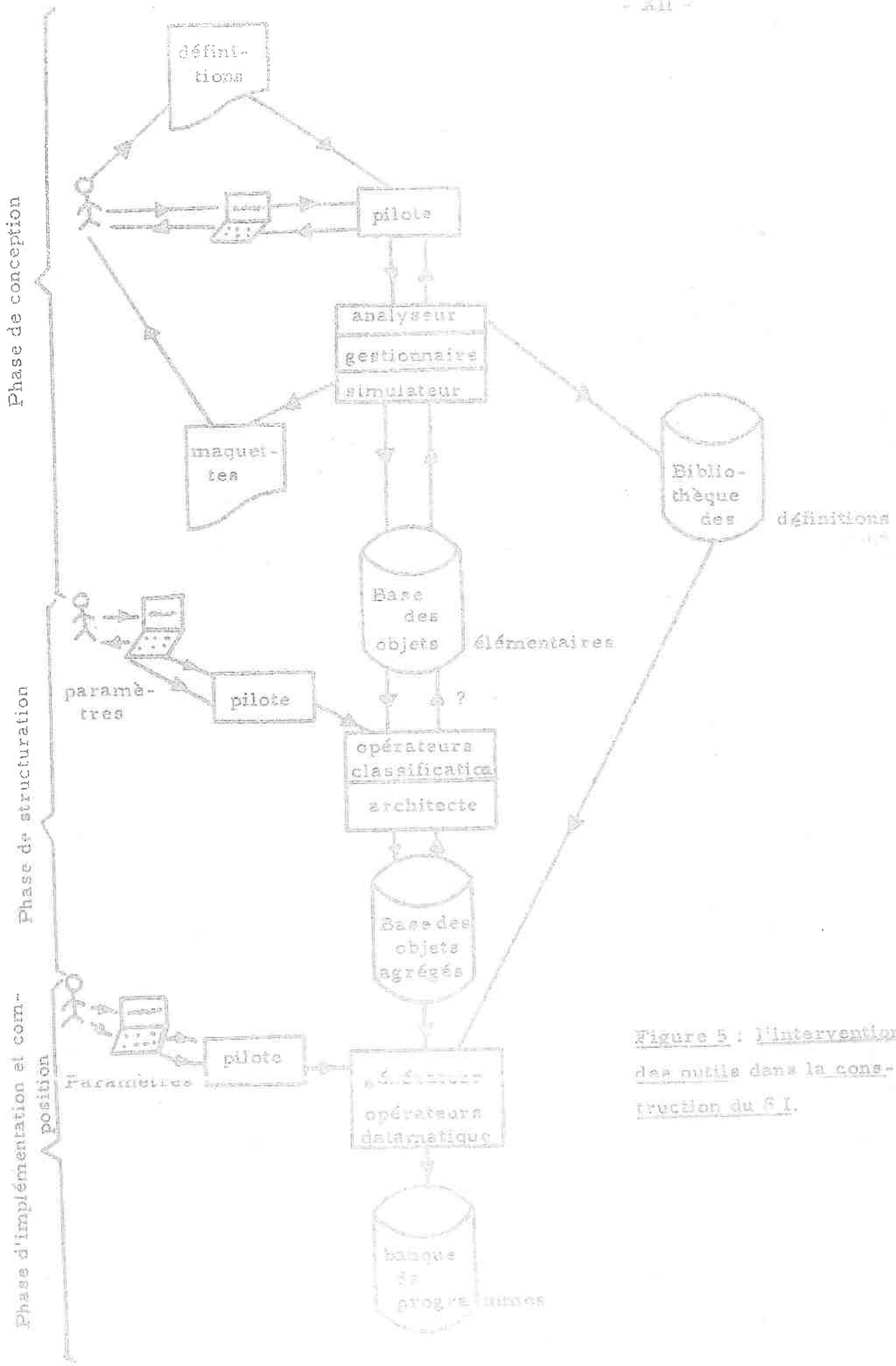


Figure 5 : L'intervention des outils dans la construction du S I.

Cependant, toutes les composantes de ce modèle se rapportent à l'aspect statique de la conception du SI ; nous le complétons donc actuellement en définissant de manière symétrique les caractéristiques sémantiques et dynamiques des éléments du schéma conceptuel, les paramètres qui régissent l'évolution du SI, en proposant la forme de leur expression et en définissant enfin la méthode permettant d'introduire ces nouvelles définitions et les outils pour le faire de manière cohérente.

5 - NOTRE CONTRIBUTION AU PROJET REMORA

Dans la présentation générale du projet Remora qui vient d'être faite, on souligne que le premier choix méthodologique est celui de la conception et construction d'un système d'informations en trois phases : de conception, de structuration, d'implémentation et composition.

Notre travail s'inscrit dans la phase de conception. Il vise à montrer l'intérêt d'utiliser, dans le cadre d'une méthode de conception intégratrice des données et des traitements, la notion de schéma conceptuel et tout particulièrement pour résoudre les problèmes d'aide à la conception qui se posent à ce niveau.

Plus précisément, notre travail comporte un aspect théorique développé dans les trois premiers chapitres et un aspect plus pratique de réalisation présenté au chapitre 4.

Le chapitre 1 propose une modélisation du monde réel et sa représentation dans le schéma conceptuel. On y trouve la définition des concepts retenus : en particulier la description des composants-types du schéma conceptuel, de leurs caractéristiques et de leurs relations.

Le chapitre 2 est consacré à l'étude de l'implémentation du schéma conceptuel. Nous y abordons les problèmes de la reconnaissance des types d'éléments et la valorisation de leurs caractéristiques et nous proposons les algorithmes correspondants.

Dans l'hypothèse où nous nous plaçons d'une phase conceptuelle aboutissant à la définition d'un schéma conceptuel il était naturel que nous nous intéressions à la manière d'aider le concepteur à produire ce résultat.

L'objet du chapitre 3 est de mettre en évidence toutes les fonctions d'aide à la conception qu'un système de conception doit assurer et nous proposons les types de moyens envisageables pour les satisfaire.

Le chapitre 4 est consacré à la présentation de la phase de conception dans Remora et en particulier, d'une part, au développement des algorithmes choisis pour les outils d'aide à la conception ; d'autre part, à l'étude sur un exemple, de l'utilisation de ces outils par le concepteur.

6 - CONCLUSION

Notre espoir dans ce travail est double.

Il est d'abord, de définir les concepts minimaux dans une méthode de conception qui se situe entre l'analyse classique par les traitements et l'approche base de données.

Il est ensuite, de valider les idées avancées par l'expérimentation sur des cas réels. C'est un objectif que nous poursuivons parallèlement en réalisant un prototype du système que nous utilisons expérimentalement à la mise en oeuvre de cas concrets et, avec profit dans l'enseignement.

CHAPITRE 1

LE SCHEMA CONCEPTUEL

CHAPITRE I - LE SCHEMA CONCEPTUEL

I - 0 - INTRODUCTION

A l'origine quand il s'agissait de concevoir un système d'informations (nous noterons en abrégé SI), chaque groupe de réalisateurs définissait sa propre démarche. Peu à peu, on a vu apparaître des méthodes d'analyse souvent réduites à des guides documentés de la conduite d'une automatisation de gestion. Ces méthodes se situent toutes dans l'approche "traitements" telle que nous l'avons suggéré dans l'introduction.

Certes elles ont permis une progression vers la rationalisation parce qu'elles ont mis en évidence certaines difficultés, suggéré une typologie des traitements (saisies, contrôle...) et proposé leur standardisation, défini, enfin, certains outils dont l'utilité n'est plus à démontrer tels que ceux qui assurent une documentation automatique de l'analyse au cours de la conduite d'un projet ou ceux qui prennent en charge l'automatisation de fonctions technologiques telles que la cinématique des fichiers.

Cependant, il nous semble que ces méthodes n'ont pas posé le problème de la conception du SI dans toute son étendue. En particulier, il nous apparaît que ces méthodes se cantonnent dans la résolution des problèmes du niveau de l'implémentation physique d'un système d'information en mettant trop l'accent sur la définition et la manipulation d'objets strictement informatiques (fichiers, données, programmes). Il ne faut pourtant pas oublier que, même au niveau de l'implémentation physique, il s'agit de résoudre des problèmes de représentation de phénomènes qui se produisent dans l'univers de l'organisation et dont il est nécessaire de rendre compte dans le SI. En centrant la réflexion sur les aspects techniques liés aux contraintes et aux particularités de la technologie on néglige les aspects de représentation qui devraient pour l'essentiel orienter la solution technique définitive. On risque de ce fait de construire un SI qui ne permette pas de savoir ce qui se passe réellement dans l'univers de l'organisation.

S'il est vrai que de toute façon la solution à laquelle il faut aboutir sera exprimée dans les termes et en fonction des moyens techniques retenus, la difficulté de la constuire d'emblée, la nécessité apparaissant ultérieurement d'en changer certains éléments conduisent à définir une solution qui, tout en exprimant l'ensemble du problème de représentation en ignore les aspects techniques. Une expression logique de la solution retenue pour le système d'information nous apparaît une étape indispensable du processus de conception et de réalisation.

Dans cette optique, l'approche "Bases de données" soulignée dans l'introduction en opposition à m'approche précédente, nous semble avoir apporté une pierre à l'édifice de cette expression logique et va même plus loin en proposant de construire une expression ou image conceptuelle des phénomènes de l'organisation qui seront traduits par des données.

Dans le but de différencier les problèmes techniques des problèmes conceptuels et de permettre au concepteur-réalisateur de mieux aborder la construction du SI, des équipes aussi diverses que le Club Banque de Données [14,15] en France ou le DBGT de Codasyl [16] et Guide/Share 29 aux Etats-Unis se sont attachées à définir les différents niveaux de conception d'une base de données.

Alors a été introduit le concept de schéma conceptuel, ainsi défini dans le rapport intermédiaire du groupe de travail de l'ANSI-SPARC [3] en Février 1975 : "le schéma conceptuel" représente la vue qu'à l'entreprise de la structure qu'elle tente de modéliser dans la Base de données".

L'approche conceptuelle est actuellement développée sous diverses formes dont on a une vision globale dans l'ensemble des publications du congrés international de Freudenstadt [32], par des chercheurs qui ont essentiellement une optique Base de données.

Nous devons regretter qu'ils expurgent totalement la notion de traitement pour ne raisonner que sur les données.

Nous avons souvent constaté que les problèmes de gestion sont plus naturellement perçus par les gestionnaires en termes de traitements qu'en termes exclusifs de données.

Conservant une approche orientée "problèmes" et cherchant à définir une méthode de conception de SI intégratrice des aspects Données et Traitements, il nous a paru fondamental dans ce cadre de faire un effort de conceptualisation du même type que celui que nous avons cité sur l'approche Base de Données.

Nous nous proposons donc dans ce chapitre de définir les concepts attachés au niveau conceptuel de l'élaboration d'un SI.

A notre connaissance, et cependant avec des hypothèses différentes sur la démarche, les propositions de BENCI, BOGAERT, BODART, CABANES dans [4] sont les seules à se situer dans cet esprit.

De manière similaire nous serons amenés à définir un modèle du monde réel puis les objets-types du schéma conceptuel et leur correspondance avec ceux du monde réel ainsi que les caractéristiques types qui leur sont attachées.

Au préalable, nous présenterons les objectifs et les qualités que doit avoir un schéma conceptuel.

I - 1 - OBJECTIFS ET QUALITES DU SCHEMA CONCEPTUEL

Au terme de l'analyse, le schéma conceptuel doit représenter formellement la partie des phénomènes de l'organisation que l'on a décidé de prendre en compte dans le SI.

Cette expression, qui est une traduction du monde réel en termes de types doit exclure toute considération d'ordre technique correspondant à des préoccupations d'implémentation. De cette façon sera assurée l'indépendance des solutions d'implémentations rendant possibles divers choix d'implémentations pour un même schéma. En particulier, si pour des raisons de performances ou des nécessités technologiques on est conduit à modifier l'implémentation, il suffira de dérouler le processus de construction du SI à partir du résultat de l'étape conceptuelle.

Si le schéma conceptuel doit être considéré comme le point de départ de l'implémentation, il est surtout l'aboutissement de l'analyse. Il doit donc être l'expression fidèle, complète, cohérente, extensible et simple de phénomènes pris en compte par l'organisation.

Fidèle, cela signifie traduction sans biais et sans déformation du réel organisationnel.

Complète, de telle façon que chaque utilisateur retrouve dans la représentation le point de vue particulier qui l'intéresse.

Cohérente, de telle sorte que la sémantique de la représentation ne comporte ni contradiction, ni ambiguité.

Extensible, pour que la prise en compte de nouveaux phénomènes n'oblige pas systématiquement à remettre en cause les représentations déjà implémentées.

Simple, en recourant à un minimum de concepts pour traduire toute la sémantique (l'objectif de non redondance sémantique est un des moyens de la simplicité).

I - 2 - L'APPROCHE DU MONDE REEL AU SENS DE REMORA

Sans doute est-il vain comme le remarquent B.B.B.C. dans [4] de vouloir proposer un modèle unique du monde réel. Chaque individu de l'organisation, en fonction des préoccupations qu'il a, le perçoit à sa manière. Aussi, le concepteur réalisateur du SI devra-t-il s'efforcer de prendre en compte les différents points de vue de telle sorte que chaque utilisateur de SI retrouve les aspects du réel qui l'intéressent. Parmi les travaux sur la modélisation du monde réel, nous pouvons citer ceux de MAC GEE [40] et de TAYLOR [65] Plus récemment ceux d'ABRIAL [1] , BODART [7] ,du Club Banques de Données de l'IRIA [14] et diverses communications présentées lors de la conférence de FREUDENSTADT sur la modélisation dans les SGBD [32] . Certains de ces travaux sont exclusivement orientés structuration des données [1] alors que d'autres prennent en compte de manière partielle les traitements [7] , [14] .

Pour notre part, c'est moins en termes de modèle que nous approchons le monde réel, qu'en termes d'inventaire des phénomènes qu'il conviendra de prendre en compte dans la représentation.

Notre effort porte sur la définition de concepts et d'un langage permettant de représenter sous forme de structures de données et de traitements de ces structures les phénomènes qui constituent le réel organisationnel.

Au préalable nous estimons indispensable de définir les catagories de phénomènes qu'il faut prendre en compte dans le SI.

I - 2-1 - Les éléments constitutifs du monde réel à représenter dans le SI

Nous faisons l'hypothèse que le monde réel est constitué d'occurrences d'objets (ex : la collection des employés de l'entreprise, les produits en stocks à la date du 30 Septembre 1976, les clients de l'entreprise inclus dans la liste des personnes avec lesquelles elle est en relation,...)

A un instant donné, le monde réel nous est accessible par la description de son état. Nous appellerons situation, la description, définie spatialement et temporellement, des objets qui constituent le réel. Une situation exige que l'on décrive les objets en eux-mêmes, à travers les propriétés qu'ils ont, mais aussi les uns par rapport aux autres, à travers des associations d'objets dans lesquelles chacun d'eux joue un rôle spécifique. Une même occurrence d'objet peut appartenir à plusieurs associations. (ex : une livraison associe 15 unités, du produit AX842, au client Jacques ROBEL, à un lieu de livraison 15 avenue Antoine - NANCY -).

La représentation des objets et des associations entre objets permet de rendre compte dans une perspective statique de ce qui est à un instant donné dans le réel organisationnel. Une telle représentation ignore les transformations qui se produisent dans le réel et qu'il faut nécessairement prendre en compte.

Dans une vision dynamique on doit aussi représenter les transformations subies par les objets ou leurs associations. Ces transformations traduisent autant la nature des objets que les propriétés explicites qu'on peut leur attribuer.

Un évènement est un fait qui provoque la modification de l'état d'un ou plusieurs objets. Ce fait se situe dans le temps et se caractérise par une collection de paramètres. Ces transformations sont engendrées par des évènements qui sont :

. soit déclenchés par des agents extérieurs à l'organisation et qui sont en relation avec elle (commande d'un client, demande d'une entreprise d'être dans la liste des fournisseurs pour une gamme de matières, demande d'une personne pour être dans la liste des ambauchés éventuels,...).

. soit déclenchés par l'organisation. Dans ce cas, il peut s'agir d'évènements institutionnels, intervenant à des fréquences établies (facturation en début de chaque semaine des envois faits aux clients la semaine précédente, exécution tous les 20 de chaque mois de la paye,...). Il peut aussi s'agir d'évènements intervenant à des moments irréguliers et provoqués par les agents de l'organisation à la suite de décisions (lancement d'un lot de fabrication, embauche du personnel, modification de la grille des salaires, modification de la valeur du point de base pour le calcul des rémunérations, affectation du personnel à un projet,...).

Un évènement qui entraîne la transformation de l'état d'un ou plusieurs objets peut engendrer une séquence plus ou moins complexe et immédiate d'autres transformations qui lui sont logiquement rattachées (ex : dans le cas de la paye, le fait que l'on soit le 20 du mois va engendrer plusieurs transformations qui vont permettre de calculer le salaire mensuel net de chaque membre du personnel et aussi le cumul de SS, le cumul de base d'imposition,... et d'autres cumul pouvant servir à calculer des primes trimestrielles ou autres...).

Il est évident que notre système de représentation devra permettre de rendre compte de l'interdépendance logique et temporelle des transformations liées à un évènement, de même qu'il devra permettre d'exprimer qu'un même objet peut subir des transformations différentes, soit en fonction de la nature des évènements qui le concernent, soit en fonction de leur séquence.

Traduire les transformations subies par les objets ou les associations d'objets c'est définir la plupart des états possibles que peut prendre le réel organisationnel en fonction d'un état initial donné et d'un ensemble d'évènements probables. Traduire ces transformations c'est aussi décrire les caractéristiques dynamiques des objets , les caractéristiques statiques des objets ou de leurs associations, que nous conviendrons d'appeler des propriétés. Les états possibles et les transformations sont très nombreux et il n'est pas possible de s'intéresser à tous. Parmi tous les états possibles, ceux qui correspondent aux besoins d'information des utilisateurs du SI méritent d'être définis de façon précise et explicite. Ils le seront sous la forme de messages indiquant la nature et le contenu des échanges entre le SI et les demandeurs d'informations. De la même manière, seront décrits les échanges entre le perçu organisationnel et le SI pour que se fasse l'ajustement permanent entre le réel et sa représentation et pour que l'on détermine selon quelle procédure, avec quel rythme et avec quel décalage les phénomènes survenus dans le réel sont repris dans le SI et les caractéristiques dynamiques seront décrites par des définitions.

Nous récapitulons l'ensemble des concepts qui nous permettent d'évoquer les catégories de phénomènes qui devront figurer dans le SI : objet, association d'objets, propriété, définition, évènement message.

I - 2 - 2 - Définition des éléments constitutifs du monde réel

Objet : C'est un constituant de l'organisation concret ou abstrait qu'un individu ou un groupe conçoit et perçoit comme ayant une existence extrinsèque et dont la connaissance présente pour lui un intérêt qui justifie que le système d'information en donne une représentation.

Exemple : - un client
- un produit.

Association d'objet : C'est une collection de deux ou plusieurs objets qui permet de représenter des situations complexes dans laquelle chacun d'eux joue un rôle particulier qui doit être spécifié.

exemple : "propriété" est une association entre "personne" et "véhicule".

propriété : C'est une qualité que l'on reconnaît aux deux classes d'éléments antérieurement définis.

Exemple : - pour un objet :
le n° matricule est une propriété de l'employé
- pour une association :
le n° d'atelier de travail est propriété de l'association existant entre atelier et employé.

définition : C'est l'expression des transformations relatives soit aux objets soit aux associations.

Exemple : - expression du salaire net d'un ouvrier.

évènement : C'est un fait réel qui provoque les transformations des objets ou des associations.

Exemple : - la fin du mois.

message : C'est la description des échanges entre le SI et l'environnement qui l'alimente ou qui l'utilise.

Exemple : - le bulletin de salaire
- un libellé sur une console.

L'approche d'un problème de gestion telle que nous l'avons définie dans ses grandes lignes dans l'introduction et qui est développée dans Perea [54] nous a conduit à n'appliquer la méthode de conception qu'à un sous-ensemble des phénomènes de l'organisation considéré comme un tout, isolé et indépendant du reste. On peut trouver dans Perea [54] une méthode d'approche modulaire de l'organisation qui induit son découpage en sous-systèmes.

Nous en donnons la définition suivante :

sous-système : un domaine de l'organisation qui concerne une même catégorie d'objets (produits - clients - fournisseurs - employés) ou une même catégorie d'évènements relatifs à une préoccupation de l'organisation (embaucher - acheter - vendre - produire - concevoir).

I - 3 - CHOIX D'UNE STRUCTURE CONCEPTUELLE

A l'approche du monde réel que nous venons de définir, nous faisons correspondre un modèle de représentation qui appartient à la classe des modèles conceptuels.

Nous avons fait le choix d'établir une correspondance biunivoque entre les éléments constitutifs du monde réel et ceux du schéma conceptuel qui le représentera.

Cette correspondance est schématisée par le tableau suivant (⟷ signifie "correspond").

Monde réel		Schéma conceptuel
évènement	⟷	procédure
message	⟷	lot
définition	⟷	module
propriété	⟷	mot
association	⟷	relation
objets	⟷	entité
sous-système	⟷	application

On doit remarquer que dans le schéma conceptuel d'un sous-système la représentation est faite en terme de types.

Le concept de type dans le schéma conceptuel est une notion agrégée proche, pour nous, de ce qui est explicité en [4 p. 388] et en [46 p. 463 à 465].

Le type procédure :

Une procédure est la traduction en termes du modèle conceptuel de l'évènement et des transformations qu'il engendre.

Exemple :

. La procédure "paie" appartient au type-procédure de l'application "gestion du personnel".

Le type-module :

Un module est l'expression d'une définition en termes du modèle conceptuel.

Exemple :

. Le module "calcul du salaire net" appartient au type-module de la procédure "paie" de l'application "gestion du personnel".

Le type-mot :

Un mot est la représentation d'une propriété en termes du modèle conceptuel.

Exemple :

. Le mot "SALNET" appartient au type-mot du module "calcul du salaire net", de la procédure "paie", de l'application "gestion du personnel" et représente le salaire d'un employé.

Le type-lot :

Un lot est la représentation d'un message en termes du modèle conceptuel.

Exemple :

. Le lot "bulletin de salaire" appartient au type-lot de la procédure "paie", de l'application "gestion du personnel".

Le type-entité :

Une entité est la représentation d'un objet en termes du modèle conceptuel.

Exemple :

. L'entité "AGENT" appartient au type-entité de l'application "gestion du personnel".

Le type-relation :

Une relation est la représentation d'une association d'éléments en termes du modèle conceptuel.

Exemple de relation entre mot et module :

SALNET est un mot qui appartient au module "calcul du salaire net".

Le type-application :

Une application est la représentation d'un sous-système en termes du modèle conceptuel.

Exemple :

Le type-application du SI est composé de la "gestion du personnel" et de "la gestion des stocks".

I - 4 - DESIGNATION DES OBJETS

Dans l'univers de la représentation on ne travaille que sur des désignations qui sont les substituts des objets du monde réel.

Le choix des désignations doit se faire de manière à pouvoir distinguer dans le SI les objets qui sont distincts dans le monde réel et qui appartiennent à la même classe.

Puisque dans le modèle conceptuel on ne travaille qu'en termes de types le choix des désignations ne concerne que la description des types.

L'ensemble des désignations constitue ce que nous appelons le vocabulaire de l'application.

La création et la gestion de ce vocabulaire dans tout système, qu'il soit manuel ou automatisé, est compliquée par la multiplicité des interlocuteurs ayant accès aux désignations ; ceux qui les créent n'étant pas obligatoirement les seuls à les utiliser.

Ainsi par exemple l'objet "quantité en stock" nommée "QTE" par le gestionnaire de stock est aussi utilisé par le statisticien qui désire étudier les mouvements d'entrées-sorties de stock des divers produits.

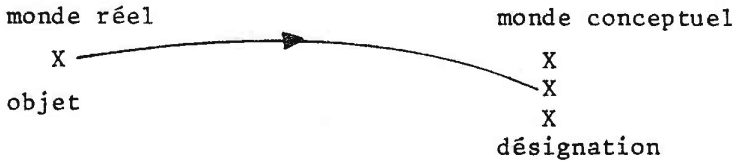
Si les deux interlocuteurs utilisent des noms différents pour désigner le même objet le vocabulaire comportera des synonymes. En revanche deux utilisateurs qui s'ignorent peuvent très bien désigner par QTE l'un la "quantité en stock" et l'autre "la quantité commandée par un client". Dans ce cas le vocabulaire comportera des polysèmes.

Plus généralement les problèmes de représentation en termes du modèle conceptuel sont des problèmes de désignation des objets du monde réel par des objets du monde conceptuel.

Logiquement, pour respecter la bijection que nous avons introduite par hypothèse entre les "éléments constitutifs" du monde réel et ceux de l'univers de sa représentation, à tout objet du premier domaine doit correspondre un objet du second.

Inversement à tout objet du schéma conceptuel doit correspondre un objet et un seul du monde réel

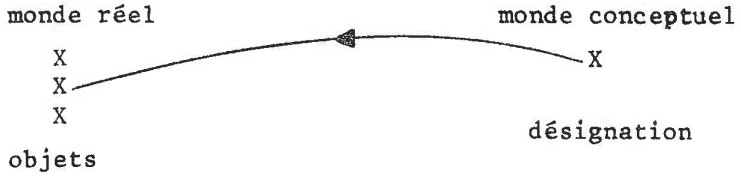
Si pour des raisons du type de celles que nous avons citées précédemment (multicplicité des interlocuteurs, erreurs et confusions de leur part.. plusieurs désignations dans le monde conceptuel correspondent à un même objet du monde réel, ces désignations sont synonymes :



Dans la mesure où la synonymie est explicite c'est à dire lorsque l'on sait que X, Y, Z désignent indifféremment le même objet du monde réel elle est acceptable puisque contrôlée.

En revanche les cas de synonymie implicite provoqueront des difficultés liées à la redondance sémantique qu'ils introduisent.

De la même manière, si les circonstances ont conduit les concepteurs à la situation où une même désignation dans le schéma conceptuel correspond en fait à plusieurs objets distincts du monde réel, ils ont créé sans s'en rendre compte des polysèmes :



On conçoit alors toutes les ambiguités et les incohérences qui peuvent naître d'une telle situation. La polysémie n'est jamais acceptable.

Partant des problèmes de désignation liés à ceux de la représentation que nous venons d'évoquer ici, nous montrerons au chapitre 3 quelles solutions leur apporter ; et notamment, si l'on veut que le système de représentation soit réellement utilisé, comment il convient de lui conserver une certaine souplesse, tout en éliminant les inconvénients liés à la synonymie et la polysémie, en recourant à des outils qui permettent de gérer les synonymies explicites, et de détecter les synonymes implicites et les polysèmes.

Nous y préciserons, en particulier, les choix effectués dans Remora face à ces problèmes et expliciterons au chapitre 4 les algorithmes de gestion et de détection associés aux outils.

I - 5 - ETUDE DETAILLEE DES OBJETS DU SCHEMA CONCEPTUEL*

Nous ferons cette étude en 2 étapes : une description des objets et de leurs caractéristiques suivie de l'étude de leurs relations.

I - 5-1 - Les objets et leurs caractéristiques :

Nous avons adopté un même plan de présentation pour chacun des objets précisant en trois points :

a) La définition de l'objet

En complément aux définitions du paragraphe I-3, nous présenterons les définitions communes à l'équipe Remora et ferons des rapprochements avec les termes employés par d'autres systèmes.

b) La classe du vocabulaire à laquelle il se rattache

Nous préciserons pour chacune des classes d'objets la portée des désignations.

c) Les caractéristiques

Les caractéristiques des objets du schéma conceptuel sont de deux types :

- Les caractéristiques liées à la reconnaissance de l'objet au moyen de désignations dans la représentation conceptuelle.

Nous les appellerons caractéristiques descriptives.

- Les caractéristiques liées à l'usage que l'on va faire des objets qui correspondent :

. soit à un usage que l'on peut qualifier de statique car il ne concerne que le rôle d'un objet vis à vis des autres dans le schéma conceptuel à un instant donné (nous les nommerons caractéristiques d'usage).

. soit à un usage dynamique car il se rapporte à l'aspect transformationnel du système (nous les nommerons caractéristiques dynamiques).

I - 5-1-1 - L'application

a - Définition

Parmi les définitions qui se rapprochent de la notre, nous pouvons citer celle de : - MALLET |41|

* mot objet (non associé à "monde réel") sera employé dans le sens usuel d' "élément".

"Famille de procédures appartenant à un même domaine d'activité de l'entreprise, traitant d'informations concernant une même population, déclenchées par des évènements de même nature et nécessaire pour l'obtention des mêmes sorties".

- BODART [7 p. 15 et 8 p. 79]

. "Décrit l'enchaînement des phases relatives à un flux d'informations, flux homogène et permanent dans le temps".

. "Correspond à un ensemble d'activités homogènes exercées dans le cadre d'un sous-système".

- S.C.A.P.F.A.C.E. [39]

. "Ensemble des chaînes utilisant une donnée d'entrée pour un même utilisateur".

- Club Banque de Données [14 p. 11]

. Suite ordonnée de plusieurs fonctions logiques pour générer une ou plusieurs structures logiques de sortie associées à une même fonctions de l'entreprise".

Exemples d'application :

. gestion du personnel
. gestion commerciale.

Nous nous attachons à l'application qui est l'ensemble des procédures automatisables, notion proche de celle qui définit PECCOUD dans [53] .

"Application perçue par les futurs utilisateurs comme un ensemble de commandes, opérations automatiques de traitement de l'information en entrée ou en sortie du service informatique considéré comme une boîte noire".

Le découpage de l'organisation en applications est supposé réalisé, suivant une méthode quelconque. PEREA en propose une dans [54] à partir d'une structuration de l'organisation en systèmes et sous-systèmes.

b - Classe de vocabulaire associée au type-application

Le vocabulaire géré par le système Remora est défini pour une application ;la classe ne comporte donc qu'une désignation, celle de l'application elle-même.

c - Les caractéristiques d'un objet du type-application

L'application n'a que des caractéristiques descriptives :

. une désignation unique qui permet de la nommer

. une définition en clair ou titre explicitant le symbole qui lui est attribué.

. un résumé précisant les finalités de chaque occurence de type-procédure la composant.

I - 5-1-2 - La procédure :

a - Définition d'un objet de type-procédure

Notre vue de la procédure se rapproche de celle de MALLET [44] ; de la "phase" de BODART [7 p. 13 et 8 p. 81] , de la "fonction logique" du Club Banque de Données [14 p. 11] , de la "procédure fonctionnelle" que définit le groupe II d'INFORSID [33 p. 4] .

Pour nous une procédure correspond à une activité de gestion comme la paie du personnel ou la gestion d'un stock de marchandises qui est déclenchée par un évènement tel que la fin du mois ou la rupture de stock.

La qualité du découpage en procédure est d'être orthogonal : cela signifie qu'une procédure ne peut être décomposée en une suite de procédures se transmettant entre elles des lots logiques d'informations regroupés en messages.

b - La classe de vocabulaire associée au type-procédure.

Elle est constituée de l'ensemble des désignations de procédure, chaque occurrence de type-procédure possédant un nom et un seul.

Le vocabulaire étant défini pour une application la portée de la designation d'un objet de type-procédure est l'application.

c - Les caractéristiques d'un objet de type-procédure

La procédure est munie de caractéristiques :

α - descriptives qui sont :

- une désignation de procédure vérifiant la règle d'unicité et servant de repère de chaque occurence de type-procédure.

- une définition en clair qui, en quelques mots, précise le sens de la procédure mnémoniquement synthétisé dans la désignation.

- un résumé explicitant les finalités de la procédure.

β - dynamiques :

Les caractéristiques dynamiques de la procédure sont en cours de

spécification. On trouvera dans la thèse de CHESSERON [12] leur inventaire complet. Nous pouvons dès lors citer par exemple :

- l'expression de l'évènement qui déclenche la procédure (exprimé temporellement).

Exemples :

. L'arrivée d'une commande entraîne le lancement de la procédure "fabrication".

. la fin du mois ou la rupture de stock provoque le lancement de la procédure "gestion du stock".

. chaque jour est lancée la procédure "facturation".

I - 5-1-3 - Le module

a - Définition d'un objet de type-module

Nous pouvons rapprocher le module de Remora du "processus logique" du Club Banque de Données [14 p. 11] du "processus" ou de "l'opération élémentaire" de BODART [7 p. 12] et [8 p. 82] , de la "procédure élémentaire" de MARTZLOFF [42] et du "module" de CIVA [22] .

Pour nous un module de traitement est un sous-ensemble cohérent de la procédure ; il représente la définition des résultats de la procédure répondant aux besoins exprimés suivant la méthode de découpage modulaire exposée par PEREA dans 54 .

Exemples de modules :

. calcul du salaire brut

. calcul des cotisations ouvrières.

b - La classe de vocabulaire associée au type-module

Elle est composée de l'ensemble des désignations de module ; chaque objet de type-module possédant un nom qui ne sera pas forcément unique puisqu'une désignation de module peut avoir des synonymes.

Cette exigence est conditionnée par notre souhait d'aboutir à l'élaboration d'une véritable banque de programmes qui puisse être partagée entre différents utilisateurs qui ne sont pas exclusivement ceux qui l'ont construite.

Un utilisateur peut donc utiliser le corps d'un module défini par lui-même ou par tout autre en lui donnant un nouveau nom, synonyme du premier.

Le vocabulaire étant associé à une application, la portée d'une désignation de module est celle pour laquelle il est défini.

c - Les caractéristiques d'un objet de type-module

Le module a des caractéristiques :

. descriptives, soit :

- une désignation qui repère un module et un seul (cependant plusieurs identificateurs pouvant être associés au même module).

- une définition en clair ou titre.
- un résumé des actions effectuées par le module.

. dynamiques qui sont :
- les paramètres du module (s'il existent dans le cas particulier de modules-sous-programmes).

I - 5-1-4 - Le mot :

a - Définition d'un objet de type-mot :

Notre notion du mot se rapproche de celle du Club Banque de Données [14 p. 6] :

"Le mot est la plus petite unité d'informations ayant une signification vis à vis de l'organisation".

Le mot de Remora correspond à la "donnée élémentaire" de BODART [8 p. 4] ; à la "rubrique" de SCAPFACE [39] , au "mot" de Minos [43] ; il est proche de la rubrique de FLORY et KOULOUMDJIAN [26 p. 4] (bien qu'elle soit, dans ce cas, définie par son champ d'application),

Exemples de mot :

. nom
. adresse
. salaire brut.

b - La classe de vocabulaire associée au type-mot

Elle est composée de l'ensemble des désignations des mots d'une application. Un mot peut recevoir un ou plusieurs noms mais une désignation ne peut correspondre qu'à un seul mot.

En effet, comme nous l'avons suggéré au paragraphe I - 4, nous acceptons que le concepteur manipule, pour la commodité de son analyse,des synonymes gérés par le système en collaboration avec lui. (cf. chapitre III).

La portée des désignations de mot est donc toute l'application.

Une exception est cependant faite pour les résultats intermédiaires : éléments introduits dans le raisonnement d'analyse [54] , traduits par les générateurs en mémoires de travail, il est naturel que nous acceptions qu'une même désignation repère plusieurs d'entre eux.

Exemple :

TRAV est mémoire de travail sémantiquement différente dans les modules SALNET et SALBRUT.

Afin de lever toute ambiguité et de ne pas, en particulier, détecter de polysème à la rencontre d'une telle situation, nous avons réduit la portée des désignations des résultats intermédiaires au domaine le plus petit soit "le module".

c - Les caractéristiques d'un objet de type-mot :

Un mot possède des caractéristiques :

- descriptives qui sont :
  - une désignation qui n'obéit pas forcément à la règle d'unicité.
  - une définition en clair qui exprime le rôle du mot et sa signification de façon plus explicite que l'identificateur.
- d'usage :

Ces caractéristiques donnent d'une part le genre du mot, d'autre part sa nature vis à vis de l'application.

α - le genre du mot :

Le langage Remora autorise les tableaux et par conséquent les indices. En fait, les définitions conceptuelles excluent,par nature, tout raisonnement sur les structures de données déjà composées. Nous autorisons cependant le regroupement structuré qu'est le tableau, parce que nous considérons que leur usage est devenu naturel.

Certains mots pourront ainsi avoir, soit le genre "tableau", soit le genre "indice".

β - La nature du mot vis à vis de l'application ou nature-a :

C'est la propriété du mot spécifiant son rôle vis à vis de l'application.

Elle peut-être :

Donnée :

C'est la nature d'un mot utilisé dans l'application et dont la valeur fixée à l'extérieur de l'application n'y est ni modifiée, ni évaluée.

Exemples :

- nom d'un client
- plafond S.S.

Résultat :

C'est la nature d'un mot dont le contenu est évalué dans l'application et qui appartient à au moins un lot.

Exemples :

. salaire net et montant des cotisations S.S. évalués dans le module CALSAL et édités sur le document BULLETIN dans l'application GESTION DU PERSONNEL.

Variable :

C'est la nature d'un mot dont la valeur est réévaluée dans l'application à partir de la valeur précédente.

Exemples :

. la quantité en stock est réévaluée après l'exécution de la procédure FACTURATION.

. l'avoir d'un client est réévalué après l'exécution de la procédure GESTION DES COMMANDES.

Résultat intermédiaire :

C'est la nature d'un mot dont le contenu est évalué temporairement dans l'application et qui n'appartient à aucun lot.

Exemples :

. un indice de tableau

. un intermédiaire de calcul.

Le classement précédent n'est définitif que si la définition de l'application est stable. Or elle ne l'est pas immédiatement puisque le système admet que l'on décrive les procédures les unes après les autres sans ordre précis. Le classement est donc élaboré au fur et à mesure des définitions successivement entrées.

C'est ainsi que nous serons amenés à parler du mode d'utilisation d'un mot dans un module (caractéristique du lien mot-module) |cf. I-5-2-5| ou dans un lot (caractéristique du lien mot-lot) |cf. I-5-2-4| , puis de sa nature -p vis à vis d'une procédure |cf.I-5-2-6| (caractéristique du lien mot-procédure), pour nous permettre de définir enfin sa nature -a vis à vis de l'application.

En fait il s'agit toujours de la nature du mot vis à vis du contexte où on l'observe ; celui-ci pouvant se définir comme le champ d'utilisation d'un mot : s'il est employé dans un seul module alors son contexte général est ce module, s'il appartient à deux modules d'une même procédure, son contexte est cette procédure, à deux modules de deux procédures différentes, c'est l'application en entier.

Nous verrons par la suite à quel point l'information sur la nature des mots est utile au système. Elle nous permettra en particulier de vérifier que toute entrée est cohérente avec le vocabulaire déjà présent et d'intégrer progressivement toute nouvelle définition dans l'ensemble de celles qui ont déjà été fournies. En outre la classification des mots ainsi que d'autres paramètres tels le rattachement des mots aux entités permettront, à la phase de structuration, de déterminer les fichiers en entrée de l'application, ceux en sortie et ceux devant être mis à jour.

I - 5-1-5 - Le lot :

a - Définition d'un objet de type-lot :

Un lot est la représentation d'un message et repère l'ensemble des informations apparaissant en sortie de l'application sur un même document.

Le terme de "lot" apparaît dans d'autres travaux en général avec une définition plus restrictive et souvent affecté d'un contenu sémantique que nous ne lui accordons pas. C'est le cas du "lot" du Club Banque de Données [14 p. 7] , du "lot" de Minos [43] et de la "rubrique composée" de FLORY et KOULOUMDJIAN [26 p. 6] .

Nous sommes en fait beaucoup plus proches du "type d'imprimé" de SCAPFACE [39] qui est défini comme "un ensemble structuré de rubriques organisées à l'intérieur d'un cadre".

Exemple de lot :

- le lot "bulletin de salaire" correspond au document qui ,pour chaque ouvrier ,édite les informations de la paie, dans le cadre de l'application GESTION DU PERSONNEL.

b - La classe de vocabulaire associée au type-lot

Elle est composée de l'ensemble des désignations des lots d'une application ; chaque objet de type-lot possédant un nom qui n'est pas forcément unique puisqu'un lot peut avoir des synonymes.

Le vocabulaire étant associé à une application, la portée d'une désignation de lot est l'application en entier.

c - Les caractéristiques d'un objet de type-lot :

Le lot à des caractéristiques descriptives et dynamiques.

Les caractéristiques descriptives sont :

- une désignation qui permet de nommer un objet de type-lot et un seul (mais un lot peut recevoir plusieurs noms).
- une définition en clair qui spécifie le type de document sorti.
- les groupes qui composent le lot (et leurs caractéristiques).
- les zones qui composent les lignes d'un groupe (et leurs caractéristiques).

Les caractéristiques dynamiques sont :

- l'ordre de présentation

Exemples :

. les "bulletins de salaire" doivent être édités par ordre des matricules croissants.

. les "bordereaux" montants des salaires par catégorie" doivent être édités par ordre des services.

- la date dans la période d'exécution de la procédure.

Exemples :

. Les bordereaux récapitulatifs des cotisations S.S. doivent être édités tous les 27 de chaque mois.

I - 5-1-6 - L'entité :

a - Définition d'un objet de type-entité :

Définie comme l'image d'une classe d'objets ayant les mêmes propriétés, l'entité de Remora se rapproche de celle de SCAPFACE dans [39] , de BODART dans [8] , du groupe de l'IRIA (Moulin - Randon - Savoysky - Spaccapietra - Tardieu) dans [46] , du rapport Ansi-Sparc [3] , du groupe (Bodart - Benci - Bogaert - Cabannes) dans [4 p. 394] .

Exemples d'entité :

. CLIENT
. PRODUIT
. AGENT.

b - La classe de vocabulaire associée au type-entité :

Elle est composée de l'ensemble des désignations d'entités définies pour une "application".

Une entité peut recevoir un nom et un seul. La portée d'une désignation d'entité est donc "l'application".

c - Caractéristiques de l'objet-entité :

Dans l'état actuel des travaux du groupe nous n'affectons à l'entité que des caractéristiques descriptives :

- une désignation qui obéit à la règle d'unicité.
- une définition en clair qui la définit plus explicitement.

Cependant nous pensons qu'elle possède également des caractéristiques dynamiques, telle la durée de vie, qui seront précisées dans les travaux de CHESSERON [12] .

I - 5-2 - Les relations représentées dans le schéma conceptuel :

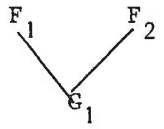
Il existe entre les objets du schéma conceptuel des relations qui sont la traduction des associations du monde réel. Certains types d'objet forment par ailleurs des ensembles sur lesquels peuvent être définis des liens structuralistes.

Pour schématiser ces liens nous utiliserons un symbolisme proche de celui que proposent FLORY et KOULOUMDJIAN dans [26 "relations entre groupes" p. 14] , se référant eux-mêmes aux travaux de TAYLOR [65] à ce sujet.

Considérant deux ensembles F et G où F est composé, des éléments $\{F_i\}$ et G est composé des éléments $\{G_i\}$, nous avons défini deux types de relations non orientées entre F et G :

Les relations (n,1)

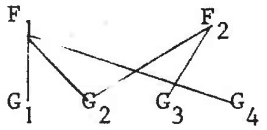
A un élément F_1 de l'ensemble F correspond un élément G_1 de G unique et plusieurs élément de F peuvent correspondre à G_1.

Exemple : 

ou

Les relations (n,n)

A un élément F_1 correspond plusieurs éléments de G et vice versa.

Exemple : 

Remarque :

La plupart des relations que nous allons définir étant de nature (n,n) nous ne donnerons la description complète que pour la première, en nous contentant d'expliciter les ensembles F et G et de donner le schéma correspondant dans les autres cas.

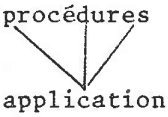
I - 5-2-1 - Les relations entre applications et procédures

Elles sont du type (n,1).

Si F est l'ensemble des procédures et G l'ensemble des applications, alors à une procédure correspond une application et une seule, mais plusieurs procédures peuvent être reliées à la même application.

Soit :

F procédures — plusieurs procédures peuvent correspondre a une application



G application — [représente le lien hiérarchique inférieur application → procédures]

F procédure a une procédure ne correspond qu'une application

G application [représente le lien hiérarchique supérieur procédure → application]

I - 5-2-2 - Les relations entre les lots et les procédures

Elles sont du type (n,n).

F étant l'ensemble des procédures,

G celui des lots,

A une procédure correspond un ou plusieurs lots d'informations.

F procédure

G lots [lien hiérarchique inférieur procédure-lots]

et plusieurs procédures peuvent être reliées au même lot.

F procédures

G lot [lien hiérarchique supérieur lot-procédures] .

I - 5-2-3 - Les relations entre les modules et les procédures :

Elles sont du type (n,n).

F étant l'ensemble des procédures,

G celui des modules.

A une procédure correspond un ou plusieurs modules

F procédure

G modules [lien hierarchique inférieur procédure-modules]

et plusieurs procédures peuvent être reliées au même module.

F procédures

G module [lien hierarchique supérieur module-procédure]

Remarquons qu'à cette relation est associée une caractéristique que nous appelons "type du module".

Nous définissons deux types : "maître" et "esclave". Un module maître

(ou directeur) est un module qui n'est jamais appelé par un autre. Un module esclave est appelé au moins une fois, mais peut appeler d'autres modules esclaves (ces notions sont explicitées, dans le cadre du langage de conception, par LAMY dans [36]).

Un module pouvant avoir des types différents dans différentes procédures, il est important de garder ces informations sur les relations module-procédure.

Nous dirons, comme pour la nature du mot, que le type d'un module est défini par rapport à un contexte.

Ce sera le plus souvent la procédure dans laquelle il est décrit, cependant, s'il appartient à diverses procédures, son contexte de définition sera l'application.

I - 5-2-4 - Les relations entre les lots et les mots

Elles sont du type (n,n).

F étant l'ensemble des lots,

G celui des mots,

A un lot correspond un ou plusieurs mots.

F lot

[lien hiérarchique inférieur lot-mot]

G mots

et plusieurs lots peuvent être reliés au même mot.

F lots

[lien hiérarchique supérieur mot -lots] .

G mot

Comme dans le cas précédent, on peut définir le "mode d'utilisation" d'un "mot dans un lot" qui spécifie le rôle du mot vis à vis du lot.

Un mot peut avoir :

- un mode "résultat intermédiaire", s'il est résultat d'une sommation locale au lot et obtenu à partir d'autres mots du lot.
- un mode "source" spécifiant que la valeur imprimée provient d'un mot de l'application.

Le rôle du mot est propre à chaque couple (mot, lot), il est donc logiquement caractéristique de la relation entre les mots et les lots et peut être différent d'un lot à un autre.

Rappelons cf. I - 5-1-4 que le mode d'utilisation d'un mot dans un

lot sera un des éléments d'évaluation de sa nature (selon un procédé développé au chapitre II).

I - 5-2-5 - Les relations entre les modules et les mots :

Elles sont du type (n,n).
F étant l'ensemble des modules,
G celui des mots.
A un module correspond un ou plusieurs mots

F module
G mots

[lien hierarchique inférieur module-mots]

à plusieurs modules peuvent être reliés au même mot.

F modules
G mot

[lien hiérarchique supérieur mot-modules] .

Tout comme le couple (mot,lot), le couple (mot, module) a une propriété : " le mode d'utilisation d'un mot dans un module", caractéristique du lien mot-module.

Elle est sémantiquement voisine de la précédente, le mode d'utilisation d'un mot dans un module exprimant le rôle du mot dans ce module.

Nous avons défini quatre modes d'utilisation :

. le mode "résultat intermédiaire" correspond à un mot dont le contenu est évalué dans le module.

. le mode "donnée déclarée" correspond à un mot déclaré comme faisant partie d'une transaction d'entrée.

. le mode "variable déclarée" correspond à un mot déclaré comme devant être mis à jour.

. le mode "non significatif" correspond à un mot qui n'a pas pu être classé ; comme nous le verrons au chapitre II les modes sont déduits ou évalués par un algorithme qui n'est pas totalement déterministe.

Comme le mode d'utilisation d'un mot dans un lot, cette caractéristique entrera dans l'évaluation de sa nature.

I - 5-2-6 - Les relations entre les mots et les procédures

Elles sont du type (n,n).

F étant l'ensemble des procédures,

G celui des mots.

A une procédure correspond un ou plusieurs mots.

F procédure

[lien hiérarchique inférieur procédure-mots]

G mots

et plusieurs procédure peuvent être reliées au même mot.

F procédures

[lien hierarchique supérieur mot-procédures]

G mot

Le couple (mot, procédure) est affecté d'une propriété : "la nature du mot vis à vis de la procédure" ou nature -p ; caractéristique du lien mot-procédure, elle spécifie le rôle du mot dans la procédure et peut être différente d'une procédure à une autre.

La définition en extension de la nature -p est :

- "donnée" : nature d'un mot dont le contenu est utilisé dans la procédure sans y être ni modifié, ni évalué.
- "résultat" : nature d'un mot dont le contenu est évalué dans la procédure et appartenant à au moins un lot.
- "variable" : nature d'un mot dont le contenu est mis à jour dans la procédure.
- "résultat intermédiaire" : nature d'un mot dont le contenu est évalué dans la procédure mais qui n'appartient à aucun lot.

Comme nous l'avons dit en I - 5-1-4 la nature-p, évaluée à partir des différents modes d'utilisation d'un mot, permet de déterminer la nature-a (vis à vis de l'application) du mot (cf. chapitre II).

I - 5-2-7 - Les liens mot-mot ou liaisons d'influence

Ce sont des liens particuliers, similaires aux relations du "type calcul" de FLORY et KOULOUMDJIAN dans [26 p. 7] , qui expriment le rôle des

composants d'un calcul. Créés par le système selon un processus explicité par PORTMANN dans [55] , ils permettent la gestion automatique des tailles des résultats intermédiaires.

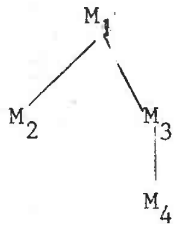
I - 5-2-8 - Les liens module - module

Comme le montre PEREA dans [54] , l'approche d'analyse proposée dans Remora est une approche modulaire. La modularité transparaît aussi dans le langage de conception [36] , par le fait qu'un module décrit puisse en appeler d'autres ou être lui-même appelé par d'autres. (cf. la notion de "type" d'un module explicité en I - 5-2-3). Il existe donc un lien structurel entre les modules. La connaissance des liens entre les modules d'une procédure peut être formalisée par une relation binaire et son graphe associé :

Définissons la relation R_p par : $M\ R_p\ M'$ si et seulement si M est "maître" de M' ou encore M "appelle" M',avec M et M' appartenant à la même procédure p.

Exemple :

Si nous avons l'arbre d'appel



M_1, M_2, M_3, M_4 étant quatre modules appartenant à la même procédure PROC1

nous dirons que :

- M_1 "appelle" M_2 et M_3, M_3 "appelle" M_4.

c'est à dire $M_1\ R_{proc1}\ M_2$, $M_1\ R\ proc_1\ M_3$

et $M_3\ R_{proc1}\ M_4$.

De même définissons la relation réciproque R_p^{-1} soit "est esclave de" ou encore "est appelé par".

- Sur l'<u>exemple</u> précédent nous pouvons dire que :

M_2, M_3 sont "appelés" par M_1, M_4 est "appelé" par M_3 c'est à dire :

$M_2 \; R^{-1}_{proc1} \; M_1$, $M_3 \; R^{-1}_{proc1} \; M_1$

et $M_4 \; R^{-1}_{proc1} \; M_3$

Introduisons la fermeture transitive R^*_p de la relation R_p pour la procédure p :

$M \; R_p \; M'$ si et seulement si $\exists n \in \mathbb{N}^*$ t.q. $M \; R^n_p \; M'$

en définissant la relation R^n_p $(n \in \mathbb{N}^*)$ par :

$M \; R^n_p \; M'$ si et seulement si $\exists M_1 \; ..... \; M_n - 1$ vérifiant :

$$M \; R_p \; M_1 \; , \; ... \; M_k \; R_p \; M_{k+1} \; ..., \; M_{n-1} \; R_p \; M'$$

$$k \in (1,..,n-2)$$

ou encore M' est "en aval" de M.

Dans l'exemple nous pouvons dire que

$M_1 \; R^*_{proc1} \; M_2$, $M_1 \; R^*_{proc1} \; M_3$, $M_3 \; R_{proc1} \; M_4$ (avec n = 1)

et $\left.\begin{array}{l} M_1 \; R_{proc1} \; M_3 \\ M_3 \; R_{proc1} \; M_4 \end{array}\right\} \Longrightarrow \underset{(n=2)}{M_1 \; R^2_{proc1} \; M_4}$ donc $M_1 \; R^*_{proc1} \; M_4$

La relation R^*_p a pour graphe l'ensemble des couples (M, M') ou M' est un module dépendant de la connaissance de M. Il y a un graphe par procédure :

<u>Exemple</u> : ici le graphe de R^*_{proc1} est :

$\{(M_1, M_2) \; , \; (M_1, M_3) \; , \; (M_1, M_4) \; , \; (M_3, M_4)\}$

I - 5-2-9 - Les relations entité-entité :

Les relations entre entités représentent les associations entre les objets du monde réel.

Exemples :

. Il existe une liaison entre AGENT, ATELIER et SERVICE qui exprime qu'un agent appartient à un atelier dans un service.

. Il existe un lien entre COMMANDE, PRODUIT et CLIENT qui exprime qu'un client a fait une commande d'un produit.

Les relations que nous percevons, sont celles que les travaux sur les bases de données prennent en compte.

Nous pouvons citer en particulier [53] (p. 2. 36 à 2.39), [3] , [4] , [1] , [65] , [46] , [26] , [16] .

Dans Remora, certaines de ces relations devront être décrites par l'utilisateur en même temps que les autres descriptions sous une forme encore non pensée, que KRIEGUER [35] étudie. (Nous pensons à ce propos à une expression relationnelle). D'autres seront déduites par le système à partir de l'analyse des descripteurs de traitement.

I - 5-2-10 - Les liens mot-entité

Elles sont du type (n,1).

Si F est l'ensemble des mots et G l'ensemble des entités, alors à un mot correspond une entité et une seule mais plusieurs mots peuvent être reliés à la même entité, soit schématiquement :

F entité — à 1 entité peuvent correspondre n mots

G mots — Exemple : "matricule", "nom" sont rattachés à l'entité AGENT.

F entité — A un mot ne correspond qu'une entité et une seule.

G Mot — Exemple : "nom" ici ne peut qualifier que l'entité AGENT, le nom du CLIENT doit être repéré autrement.

Ces relations, comme celles qui existent entre entités, ont été mises en évidence par les travaux sur les Bases de Données.

Dans le cadre de ses recherches KRIEGUER [35] sera chargé de préciser la façon d'exprimer ces liens dans le langage de Remora.

CHAPITRE 2

LA GESTION DU SCHEMA CONCEPTUEL DANS REMORA

CHAPITRE II - LA GESTION DU SCHEMA CONCEPTUEL DANS REMORA

II - 0 - INTRODUCTION

Dans une méthode d'approche "Base de données", la description du schéma conceptuel au moyen d'un langage de définition est une image de la structure conceptuelle de la base qui en constitue une représentation à ce niveau et sert de référence aux utilisateurs ainsi qu'aux outils du SGBD.

De la même manière la description du schéma conceptuel,proposé au chapitre 1 dans le cadre de notre propre approche,va constituer une représentation de tout le système d'informations au niveau conceptuel.

Cette représentation sera utilisée par le système Remora lui-même et les différents interlocuteurs du niveau conceptuel et servira de référence au système pour l'étape de structuration.

Mais la façon d'obtenir cette description est différente, puisqu'au lieu de fournir un langage de définition des objets de la structure, nous proposons [54] une méthode d'analyse des problèmes qui s'applique directement aux objets du monde réel.

C'est donc le système qui, à partir des définitions fournies par le concepteur,va d'une part établir les correspondances entre les deux modèles présentés au chapitre précédent pour reconnaître les types-objets de la structure à partir des types-objets du monde réel, d'autre part reconnaître et évaluer les objets, leurs caractéristiques et leurs liens pour construire le schéma conceptuel.

L'objet de ce chapitre est d'abord de mettre en évidence ces mécanismes de déduction et d'évaluation en présentant les algorithmes mis en oeuvre , ensuite, de donner les idées générales de l'implémentation du schéma conceptuel sous forme de la Base des Objets élémentaires que nous avons introduite dans la présentation générale du projet en tête de ce travail.

II 1 - RECONNAISSANCE DES OBJETS ET DES RELATIONS

II - 1-1 - Correspondance entre types d'objets monde réel - schéma conceptuel

Le concepteur est amené par le processus méthodologique qu'on lui propose [54] à décrire les objets du monde réel dans le langage de conception dont les supports sont les descripteurs.

Ainsi c'est sur le descripteur de traitement qu'il exprime la définition des objets de leurs propriétés et leur association.

C'est sur le descripteur de document qu'il décrit les messages et sur le descripteur de procédure qu'il définit l'évènement en déclenchant son exécution.

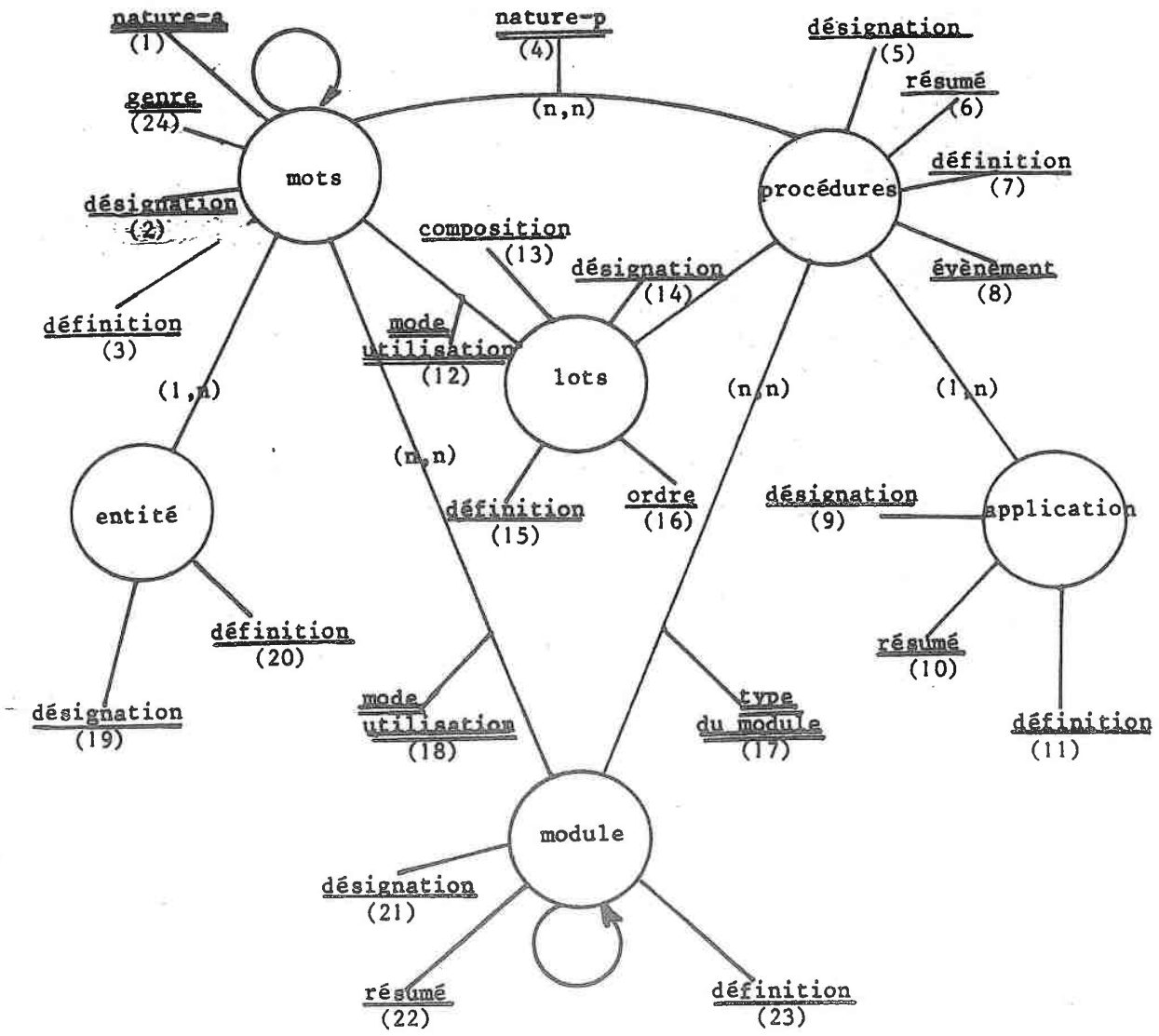
C'est donc par la reconnaissance des descripteurs que le système va établir les correspondances des types-d'objets : le descripteur de traitement assurant celles qui existent entre définition et module, objet et entité, propriété et mot, association et relation ; le descripteur de document assurant la correspondance entre message et lot ; enfin le descripteur de procédure celle qui existe entre évènement et procédure.

II - 1-2 - Détermination des éléments

L'étude du schéma conceptuel au chapitre I a mis en évidence ses composants et leurs propriétés schématisés sur la figure 1.

C'est l'analyse des descripteurs qui mettra en évidence les objets, leurs liens et valorisera leurs caractéristiques.

Pour la clarté de la présentation nous montrerons d'abord comment sont reconnues les occurences des différents types-d'objets et évaluées leurs caractéristiques puis comment sont déterminées les relations. ces actions se déroulant en fait simultanément.



L'arc qui relie les composants du schéma conceptuel exprime l'existence des relations d'appartenance entre les composants

- figure 1 - Schéma Conceptuel de Remora -

Remarque :

Pour simplifier le texte nous confondrons occurrence d'un type d'objet et objet c'est à dire, par exemple, nous parlerons du module M1 à la place de l'occurrence du type-modules l'identificateur M1.

II - 1-3 - Reconnaissance des objets et des caractéristiques

II - 1-3-1 - Les algorithmes de dépouillement

Ils correspondent à la valorisation des objets et de leurs caractéristiques par déduction directe à partir du dépouillement des descripteurs. (soulignées d'un simple trait figure 1).

Leur reconnaissance est triée au choix de la forme de l'expression du langage où les mots-clés sont positionnels : c'est à dire associés à une position typographique sur les descripteurs.

Ainsi à partir du descripteur de traitement, on valorise :

identificateur : [1] titre : [2] procédure : [3] application : [4]						
désignation en clair	condition	occurence	calcul	donnée	ident.	appel
[5]	[6]	[7]	[8]	[9]	[10]	[11]

- figure 2 - Descripteur de traitement -

. "la désignation" de module (caractéristique 21 figure 1), de procédure (5 figure 1), d'application (9 figure 1) (colonnes [1] , [3] , [4] figure 2).

. le "titre" du module (23 figure 1) (colonne [2] figure 2).

. "la désignation" d'un mot (2 figure 1) (colonnes [6] , [7] , [8] , [9] , [10] figure 2).

. "la désignation" d'un indicatif (qui repère une entité) (19 figure 1) (colonne [7] figure 2).

. la "définition en clair" d'un mot (3 figure 1) (colonne [5] figure 2).

. "la désignation" d'un module appelé (21 figure 1) (colonne [11] figure 2).

Les autres descripteurs sont construits suivant les mêmes principes ; un travail de dépouillement analogue nous permettra de déterminer les caractéristiques :

- sémantiques (9, 10, 11 figure 1) de l'application sur le descripteur de l'application.
- sémantiques et dynamiques (5, 6, 7, 8 figure 1) de la procédure sur le descripteur de procédure.
- sémantiques du mot et de l'entité (2, 3, 19, 20 figure 1) sur le descripteur de mot.
- sémantiques et dynamiques (13, 14, 15, 16 figure 1) sur le descripteur de document.

II - 1-3-2 - Les algorithmes de calcul :

Ils permettent, à partir des informations successives fournies par le concepteur, de valoriser progressivement les caractéristiques du type "variable calculée" : ce sont le genre et la nature d'un mot, le type d'un module. (soulignés de deux traits figure 1).

II - 1-3-2-1 - Le genre d'un mot : (caractéristique 24 figure 1).

La recherche du genre d'un mot (introduit au chapitre I) résulte d'une analyse syntaxique du langage par l'algorithme :

<<Tout mot trouvé entre deux parenthèses ouvrante et fermante ou une virgule et une parenthèse fermante ou deux virgules non associées à des opérateurs logiques ou arithmétiques est du genre "indice" (codé "I") ; le mot précédant directement la parenthèse ouvrante est du genre "tableau" (codé "T")>>.

II - 1-3-2-2 - Le type d'un module (caractéristique 17 figure 1).

A - Les étapes du déroulement de l'algorithme :

Elles sont deux et correspondent aux deux contextes d'observation d'un module :

- Tout module, au moment où il est reconnu (entrée d'un descripteur de traitement), est affecté d'un type valable dans le contexte local du module en cours de définition.

- Tout module reconnu peut, dans le contexte de la procédure où il vient d'apparaître, être affecté d'un nouveau type, compte-tenu de ces multiples apparitions dans cette procédure.

B - L'algorithme :

1 - Détermination du type dans le contexte local du module :

<<Tout module trouvé dans la zone repérée par le mot-clé "Identificateur" est maître (codé "MA"), tout module cité dans la zone de mot-clé "Appel" est esclave (codé "AP")>>.

2 - Détermination du type dans le contexte de la procédure :

<<Tout module qui n'est jamais appelé est "maître", tout module appelé au moins une fois est "esclave">>.

C - Exemple :

Supposons qu'un concepteur, après avoir défini le module INIT schématisé ci-dessous, demande sa prise en charge au système Remora.

identificateur : INIT titre : procédure : PAIE application :						
désignation en clair	condition	occurence	calcul	données	ident.	appel
						CALBI COTISO

Les 3 modules INIT, CALBI, COTISO seront alors immédiatement affectés des types "MA", "AP", "AP" (contexte module INIT) et des mêmes types pour le contexte procédure PAIE.

L'introduction de la description de CALBI qui appelle le module BI-TIT

identificateur : CALBI		procédure : PAIE				
désignation en clair	condition	occurence	calcul	données	ident.	appel
						BI-TIT

a deux effets :

. L'affectation locale du type "MA" à CALBI, du type "AP" à BI-TIT dans le contexte du module CALBI.

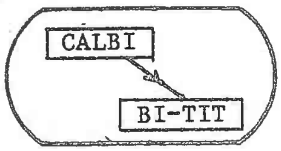
. La comparaison des différents types éventuellement rencontrés dans des contextes modules différents pour CALBI et BI-TIT :

- CALBI déjà rencontré avec le type "AP" va conserver ce type pour le contexte de la procédure PAIE.

Profitons de cet exemple pour montrer, même schématiquement, comment le système grâce à la notion de contexte peut accepter un ordre quelconque de l'entrée des définitions et en effectuer l'intégration progressive.

Supposons que le concepteur décide de décrire d'abord le module CALBI puis seulement ensuite INIT.

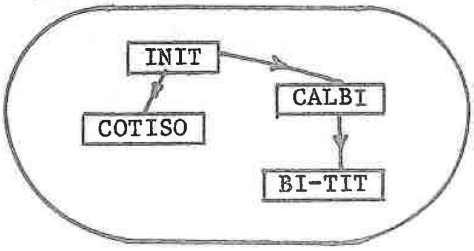
CALBI étant le seul module décrit



PAIE

est classé "MA"
et BI-TIT classé "AP"
dans les 2 contextes module et procédure.

L'entrée de INIT correspondant pour la procédure PAIE en graphe d'appel



PAIE

conduit a classer INIT en "MA" ;

CALBI et COTISO en "AP" dans le contexte module INIT, puis prenant en compte le type de CALBI (déjà rencontré "MA") a réévaluer son type vis à vis de la procédure PAIE en "AP".

II - 1-3-2-3 - La nature d'un mot :

A - Les étapes de déroulement de l'algorithme :

La nature d'un mot, nous l'avons dit en I - 5-1-4, est une variable dont la valeur évolue au fur et à mesure de l'avancement de la définition conceptuelle de l'application.

Elle est d'autre part définie pour chacun des trois niveaux de l'approche de conception modulaire :

. application
. procédure
. module

soit pour les trois contextes d'observation que nous appellerons

. contexte local module
. contexte procédure
. contexte application.

Ces trois expressions de la nature (nous dirons mode d'utilisation dans le premier cas) sont évaluées à chaque fois qu'un mot est cité sur un descripteur pris en charge par le système.

Si le mot n'est utilisé que dans un seul module alors il y a identité entre les trois nature ; s'il est utilisé dans plusieurs modules (ou lots) appartenant à une seule procédure alors nature-p ≡ nature-a ; enfin s'il appartient à plusieurs modules de plusieurs procédures alors les trois natures pourront être différentes.

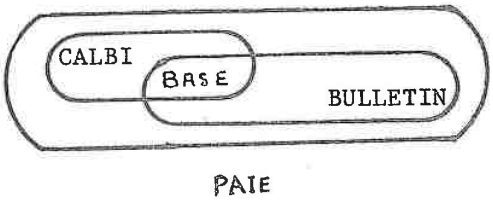
Exemple :

. BI reçoit le mode d'utilisation "résultat intermédiaire" dans CALBI de la procédure PAIE.

. BASE reçoit le mode "Donnée" dans CALBI, "source" dans le lot BULLETIN de la procédure PAIE et "donnée" dans le module COTI de la procédure RECAP-COTI ;

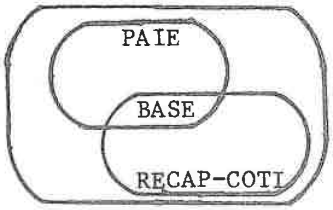
→ BI, utilisé dans le seul module CALBI, a un mode d'utilisation ("donnée") identique à sa nature-p et identique à sa nature a.

→ BASE en revanche, utilisé dans le module CALBI et dans le lot BULLETIN a un mode d'utilisation dans chaque module. Il appartient à 2 modules de la même procédure PAIE, ce que nous schématisons par :



pour laquelle on peut lui définir une nature-p. Il appartient à la procédure RECAP-COTI pour laquelle on lui définit une autre nature-p.

Il est donc à la fois dans la procédure PAIE et dans la procédure RECAP-COTI (il appartient alors à l'intersection de PAIE et RECAP-COTI) soit :



Application: GESTION DU PERSONNEL

et possède une nature-a spécifique à l'application GESTION DU PERSONNEL évaluée à partir des diverses nature-p.

La connaissance de ces trois natures sera précieuse au système pour qu'il effectue des contrôles (nous le verrons au chapitre 3) et pour qu'il définisse la dynamique de l'application exprimée par l'enchaînement des procédures les unes vis à vis des autres, à partir des lots logiques d'informations qu'elles se transmettent.

Remarquons aussi que l'évaluation de nature-a n'est définitive que du seul point de vue du système et à condition que l'application soit stable. En réalité il existe des relations entre applications qui nécessitent le même type de raisonnement sur la nature mais que, dans notre cas, nous laissons à la charge de l'Administrateur du SI.

B - Déroulement de l'algorithme :

B-1 - Caractéristiques :

Cet algorithme n'est pas déterministe dans la mesure où il opère lui-même des choix.

Ainsi, dans le contexte local du module nous avons introduit un mode indéterminé " " , correspondant au cas où le système n'a pas assez d'éléments pour classer le mot.

- dans le contexte local du lot nous avons introduit un mode semi-déterminé ("S") correspondant au cas où le système ne peut choisir entre "Donnée" et "Résultat".

- dans le contexte de la procédure si un mot n'a reçu aucun mode d'utilisation significatif alors il lui est affectée une nature-p = "Donnée",

- dans le contexte de l'application, la nature-a est déterminée à partir d'une hiérarchie (empiriquement définie) des diverses nature-p possibles.

B-2 - Description modulaire de l'algorithme :

Nous avons choisi d'une part une description modulaire et d'autre part, lorsque différents cas se présentent, une description à l'aide de prédicats de la forme :

$$
\begin{array}{lcl}
\cdot\ C_1 & \Longrightarrow & I_1 \\
\cdot\ C_2 & \Longrightarrow & I_2 \\
\vdots & & \\
\cdot\ C_{n-1} & \Longrightarrow & I_{n-1} \\
\cdot\ C & \Longrightarrow & I_n
\end{array}
$$

où $C_1, C_2 \ldots C_{n-1}$, conditions s'excluant par définitions, sont évaluées collatérallement et où C représente $\overline{C_1} \wedge \overline{C_2} \wedge \ldots \wedge \overline{C_{n-1}}$.

Le Ij correspondant au Cj vrai est exécuté.

Niveau 1

1 - Détermination du mode d'utilisation d'un mot dans le contexte du module. (caractéristique 18 figure 1).

2 - Détermination du mode d'utilisation d'un mot dans le contexte du lot. (12 figure 1).

3 - Détermination de la nature-p d'un mot (4 figure 1).

4 - Détermination de la nature-a d'un mot (1 figure 1).

Niveau 2

1 - mode utilisation/module *:

m représentant un mot, MU son mot d'utilisation : pour chaque m d'un descripteur de traitement entrant :

- *m* $\in$ *à la zone de mot-clé IDENT* $\Rightarrow$ [$\emptyset \| m \Longrightarrow MU(m) = V$ *ou* $\neg(\emptyset \| m) \Longrightarrow MU(m) = RI$]
- *m* $\in$ *à la zone de mot-clé DONNE* $\Longrightarrow MU(m) = D$
- *m est indice de travail* $\Longrightarrow MU(m) = RI$
- $C \Longrightarrow MU(m) = "\ \ "$

Niveau 2

2 - MU/lot :

pour chaque m d'un descripteur de document entrant :

- *m* $\in$ *à la zone de mot-clé SOURCE* $\Longrightarrow MU(m) = S$
- *m* $\in$ *à la zone de mot-clé NOM* $\Longrightarrow MU(m) = RI$

* Le signe || est la représentation habituelle de la concaténation.

Niveau 2

3 - Nature-p :

pour chaque m d'une procédure

3 - 1 Détermination du mode principal (MUp).
3 - 2 Détermination de la nature-p (Nat-p).

Niveau 2

4 - Nature-a :

Soient Nat-a la nature de m vis à vis de l'application,
$\mathcal{N}_p(m)$ *l'ensemble de ses natures-p.*
pour chaque m d'une application

- $V \in \mathcal{N}_p(m) \Longrightarrow Nat\text{-}a(m) = V$
- $R \in \mathcal{N}_p(m) \Longrightarrow Nat\text{-}a(m) = R$
- $RI \in \mathcal{N}_p(m) \Longrightarrow Nat\text{-}a(m) = RI$
- $C \Longrightarrow Nat\text{-}a(m) = D$

Niveau 3

3 - 1 Détermination du mode principal :

Soit MU_p *le mode d'utilisation principal d'un mot* m, $\mathcal{M}(m)$ *l'ensemble de ses modes d'utilisation :*

- $V \in \mathcal{M}(m) \Longrightarrow MU_p(m) = V$
- $D \in \mathcal{M}(m) \Longrightarrow MU_p(m) = D$
- $RI \in \mathcal{M}(m) \Longrightarrow$
  - $(S \notin \mathcal{M}(m) \Longrightarrow MU_p(m) = RI)$
  - *ou*
  - $(S \in \mathcal{M}(m) \Longrightarrow MU_p(m) = RI \wedge S$
- $C \Longrightarrow$
  - $(S \notin \mathcal{M}(m) \Longrightarrow MU_p(m) = \text{" "})$
  - *ou*
  - $(S \in \mathcal{M}(m) \Longrightarrow MU_p(m) = S)$

Niveau 3

3 - 2 Détermination de nature-p :

Soit Nat-p la nature de m vis à vis de la procédure :

- $MU_p(m) = V \Longrightarrow Nat\text{-}p(m) = V$
- $MU_p(m) = RI \Longrightarrow Nat\text{-}p(m) = RI$
- $MU_p(m) = RI \wedge S \Longrightarrow Nat\text{-}p(m) = R$
- $MU_p(m) = (D \text{ ou } S \text{ ou } \text{" "}) \Longrightarrow Nat\text{-}p(m) = D$

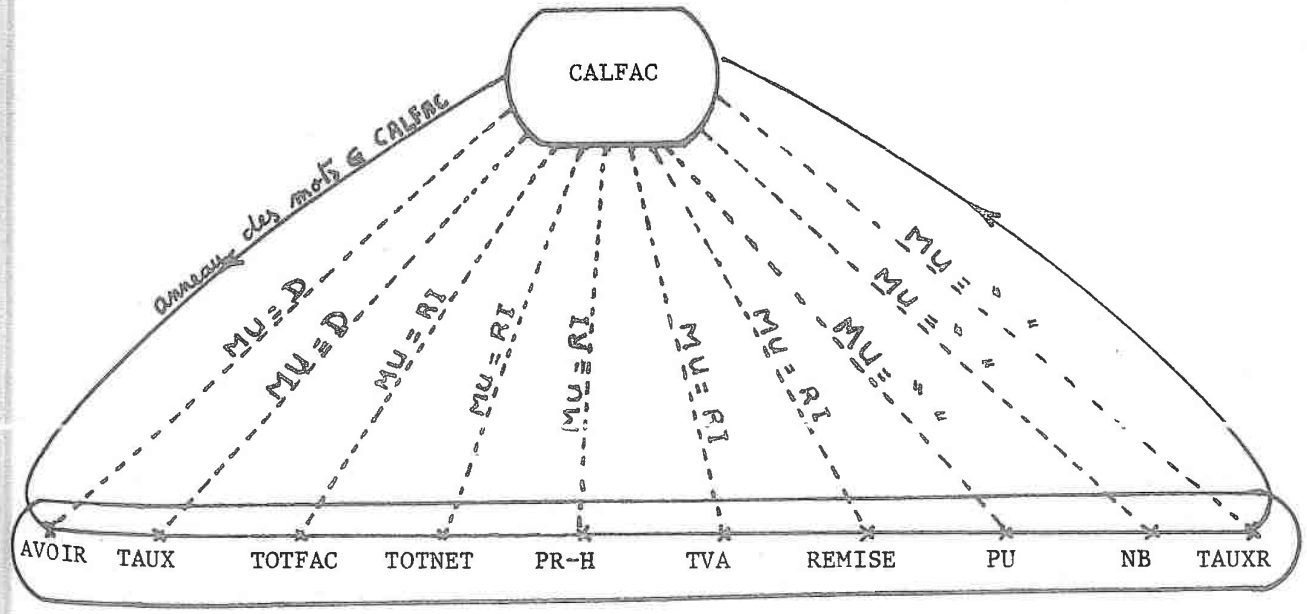
C - Exemple

1 - Evaluation des modes d'utilisation dans le contexte local d'un module

Le passage du module 1 de l'algorithme sur le descripteur :

identificateur : CALFAC			procédure : FACT				
	désignation en clair	*condition*	occurence	*calcul*	*donnée*	ident	appel
01	FACTURE PAR CLIENT		CLIENT			TOTFAC	
02	TOTAL PAR PRODUIT		PRODUIT			+TOTNET	
03	PRIX NET/PRODUIT			*PU*NB*		+PR-H	
03	TVA/PRODUIT			*TAUX*PR-H*		+TVA	
02	REMISE/CLIENT	*TOTNET*>500		*TAUXR*TOTNET*		-REMISE	
02	AVOIR/CLIENT				*AVOIR*	-AVOIR	
02	TAUX TVA				*TAUX*		

permet d'évaluer les modes et de les stocker dans la base des objets élémentaires comme caractéristique du lien $mot_i \longrightarrow module_j$, sous la forme :



-figure 3-
Résultats de l'algorithme de calcul sur un descripteur de traitement.

AVOIR
TAUX } sont classés "Donnée" puisque déclarés comme tel

TOTFAC, TOTNET
PH-R, TVA, REMISE } sont "résultat Intermédiaire" puisque résultats d'un calcul

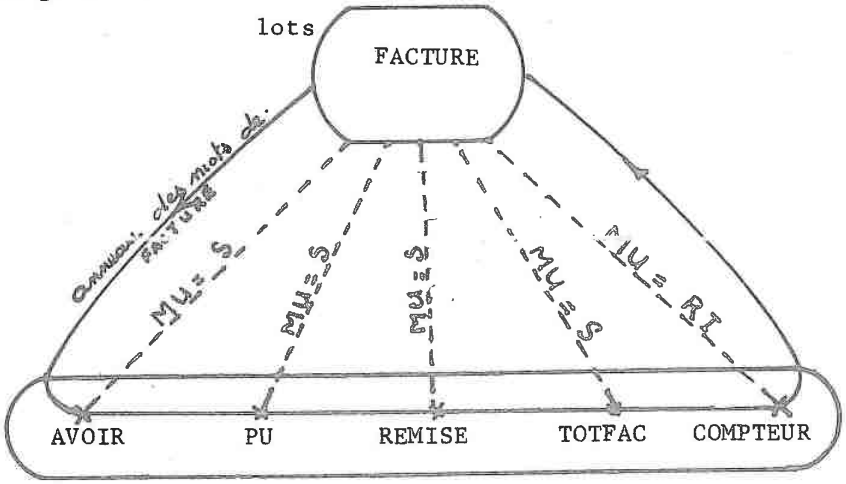
PU, NB
TAUXR } sont " " non significatif.

2 - Evaluation des modes d'utilisation des mots d'un lot

L'exécution du module 2 de l'algorithme à partir de l'analyse du bordereau numéro trois du descripteur de document décrivant le message FACTURE de la procédure FACT :

NOM	IMAGE	SUM SOURCE	IDENT
	Z(4).ZZ	SOURCE	AVOIR
	Z(4).ZZ	SOURCE	PU
	Z(4).ZZ	SOURCE	REMISE
	Z(4).ZZ	SOURCE	TOTFAC
COMPTEUR	Z(6).ZZ	SUM	TOTFAC

évalue les modes et les range dans la base des objets élémentaires comme la caractéristique du lien $mot_i \longrightarrow lot_j$ sous la forme représentée sur la figure 4 :



-figure 4- Résultats de l'algorithme de calcul exécuté sur un descripteur de document.

AVOIR, PU
REMISE, TOTFAC } sont "Source"

COMPTEUR est "résultat intermédiaire" puisque trouvé en NOM (résultat de la sommation sur TOTFAC).

3 - Evaluation des nature-p :

L'anneau des mots de CALFAC comprend les mots :

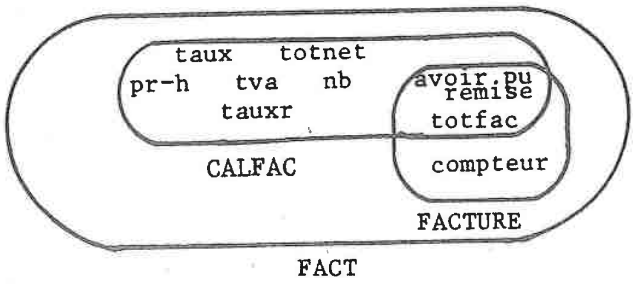
{avoir, taux, totfac, totnet, pr-h, tva, remise, pu, nb, tauxr}

Celui de FACTURE :

{avoir, pu, remise, totfac, compteur}

Leur intersection ou CALFAC $\cap$ FACTURE définit la liste

{avoir, pu, remise, totfac} est schématisée par :



Les mots de cette liste sont ceux pour lesquels la nature-p ne sera pas identique au mode d'utilisation ; le module 3 de l'algorithme va la déterminer :

a - en évaluant leur mode principal à partir de leurs modes d'utilisation, soit :

. AVOIR a pour modes $\{ D_{CALFAC} , S_{FACTURE} \}$

$\Longrightarrow$ mode principal : D

. PU a pour modes $\{ " \ "_{CALFAC} , S_{FACTURE} \}$

$\Longrightarrow$ mode principal : S

. REMISE a pour modes $\{ RI_{CALFAC} , S_{FACTURE} \}$

⟹ mode principal : RI

. TOTFAC a pour modes $\left\{ RI_{CALFAC} , S_{FACTURE} \right\}$

⟹ mode principal : RI

b - en évaluant la nature-p de chacun des mots de FACT d'après leur mode-principal, ce qui sera archivé sous forme d'une mise dans la BO des caractéristiques des liens $mot_i \longrightarrow procédure_j$, suivant le schéma de la figure 5 :

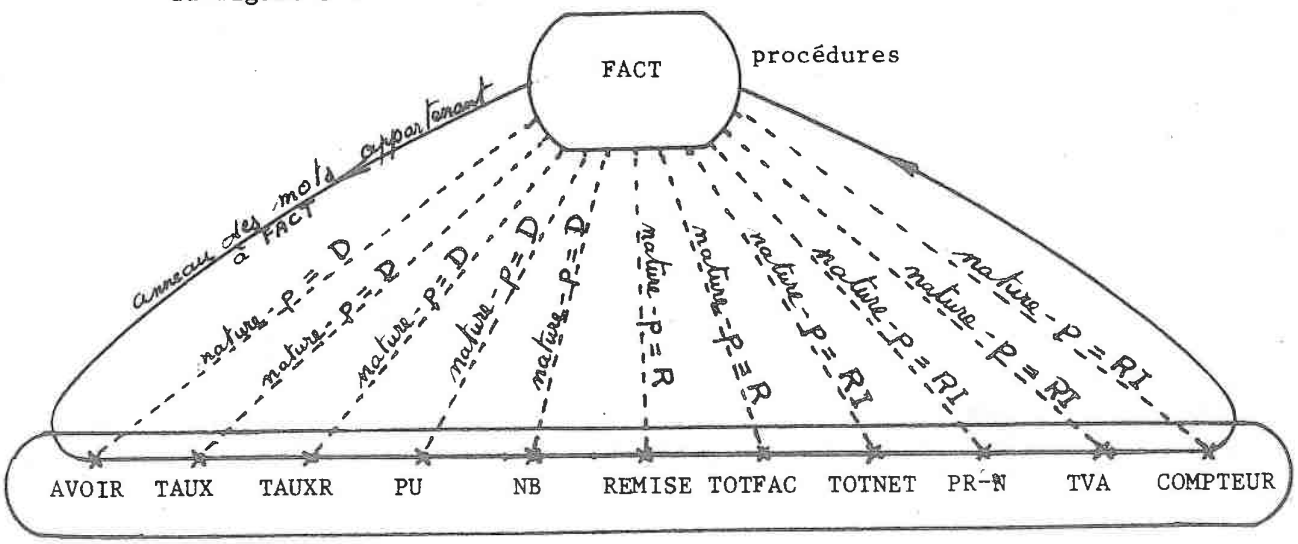


figure 5 : résultats de l'algorithme de calcul de nature-p

AVOIR, TAUX	sont classés "Donnée"
TAUX, PU NB	données définies par choix du système
REMISE TOTFAC	sont "résultats"
TOTNET PR-H TVA COMPTEUR	sont "résultats intermédiaires".

4 - Evaluation des natures-a :

Supposons, afin de compléter l'exemple, que PU dont la nature-p = "donnée", soit en fait un résultat édité transmis d'une procédure PRODUIT vers la procédure FACT; PU a donc une nature-p "résultat" dans PRODUIT.

PU, seul mot de l'intersection des procédures PRODUIT et FACT dans le contexte de l'application COMMERCE, soit PRODUIT $\cap$ FACT = {PU} , a donc une nature-a qui peut être différente de sa nature-p et qui, en tout état de cause, va être définie par le module 4.

PU est classé "R" et la mise à jour dans la BO des caractéristiques nature-a de chaque mot_i de GESTION-COMMERCIALE est schématisée en 6 :

Les 11 réalisations de l'entité mot de l'application COMMERCE sont :

n° occurence	caractéristique identificateur	caractéristique nature-a
1	AVOIR	D
2	TAUX	D
3	TAUXR	D
4	PU	R
5	NB	D
6	REMISE	R
7	TOTFAC	R
8	TOTNET	RI
9	PR-H	RI
10	TVA	RI
11	COMPTEUR	RI

figure 6 - Résultats de l'algorithme de détermination de nature-a

II - 1-4 - Les relations :

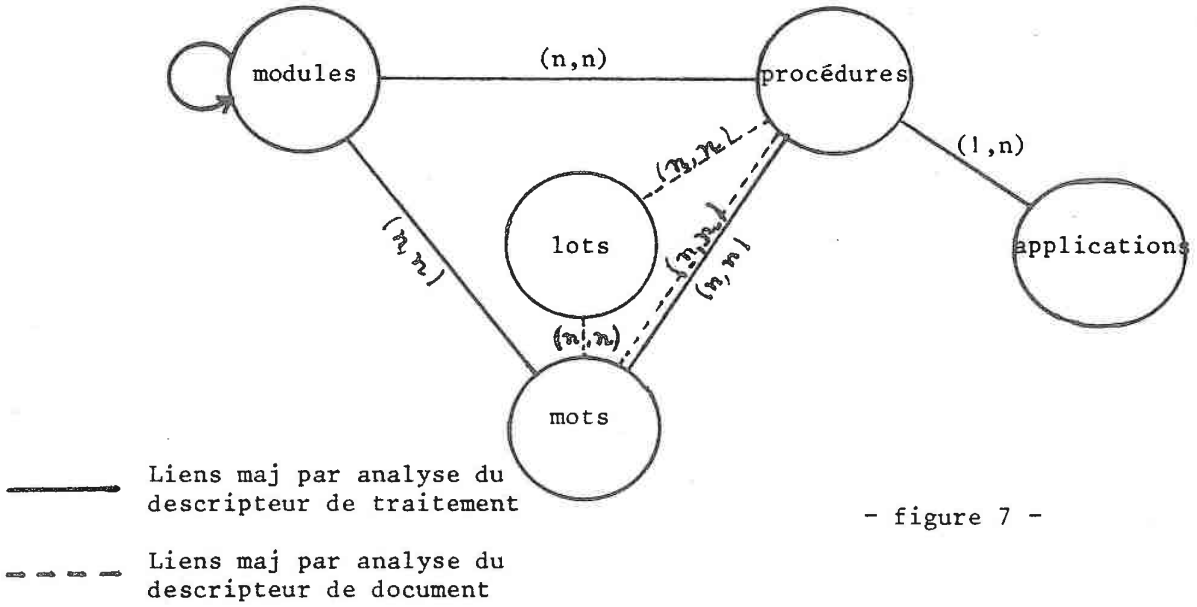
II - 1-4-1 - Nature des relations détectées :

Les relations à reconnaître et valoriser sont toutes exprimées sur les descripteurs de traitement, de mot ou de document.

Nous limiterons notre présentation à celle des processus de dépouillement

des descripteurs de traitement et de document puisque le descripteur de mot n'est pas parfaitement défini 35 .

Ces processus auront pour résultat la mise en place (ou mise à jour) des liens suivants :



- figure 7 -

II - 1-4-2 - L'algorithme

A - Description modulaire

Niveau 1
1 - Détermination des liens sur le descripteur de traitement.
2 - Détermination des liens sur le descripteur de document.

Niveau 2

1 - Descripteur de traitement :

1 - 1 Les composants des liens module ⟷ procédure sont donnés dans les zones de mots-clés "IDENTIFICATEUR" et "APPEL" pour le module, "PROCEDURE" pour la procédure.

1 - 2 Les composants des liens procédure ⟷ application sont donnés par les zones de mots-clés "PROCEDURE" pour la procédure et "APPLICATION" pour l'application.

1 - 3 Les composants des liens mot ⟷ module
et mot ⟷ procédure
sont donnés par les zones de mots-clés "CONDITION", "OCCURENCE", "CALCUL", "DONNEE", "IDENT" pour le mot, "IDENTIFICATEUR" pour le module et "PROCEDURE" pour la procédure.

1 - 4 Le lien module-module sera valorisé grâce aux zones de mot-clé "IDENTIFICATEUR" pour le module appelant, "APPEL" pour le module appelé.

Niveau 2

2 - Descripteur de document :

1 - 1 Les composants des liens lot ⟷ procédure sont donnés par le bordereau numéro 1.

1 - 2 Les composants des liens mot ⟷ lot
mot ⟷ procédure
sont donnés par le bordereau numéro 1 pour le lot et la procédure, par les zones de mots-clés "NOM" et "IDENT" pour le mot.

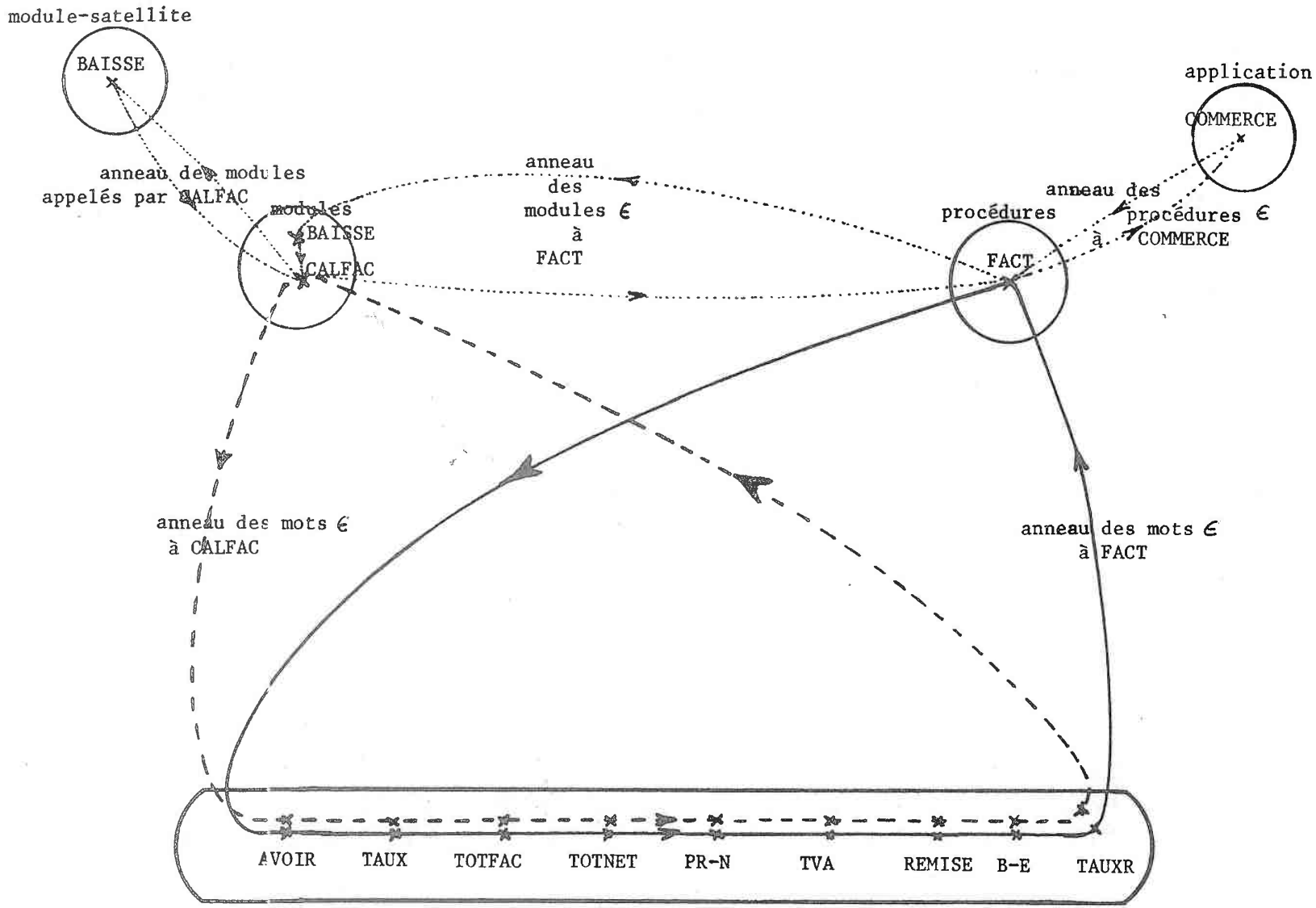
B - Exemple :

1 - Evaluation des liens à partir d'un descripteur de traitement

Le passage du module 1 de l'algorithme sur le descripteur : [cf. exemple de module II - 2-1-2-3] .

Identificateur : CALFAC	PROCEDURE : FACT		*Application : COMMERCE*				
	Désignation en clair	CONDITION	OCCURENCE	CALCUL	DONNEE	IDENT	*APPEL*
01	FACTURE		CLIENT			TOTFAC	
02	TOTAL		PRODUIT			+TOTNET	
03	PRIX NET			PU*NB		+PR-H	
03	TVA			TAUX*PR-H		+TVA	
02	REMISE	TOTNET>500		TAUXR*TOTNET		-REMISE	
02	AVOIR				AVOIR	-AVOIR	
02	TAUX TVA				TAUX		
02	BAISSE EXCEPT					BAE	*BAISSE*

entraîne la mise à jour des liens "composants" dans la BO, figurée en 8, ainsi que de tous les liens inverses (de "composés"). (Pour plus de clarté seuls les liens directs sont représentés).



-figure 8-

Résultats de l'algorithme de dépouillement des relations
sur un descripteur de traitement (liens directs)

2 - Evaluation des liens à partir d'un descripteur de document

Reprenons l'exemple de lot du paragraphe II - 2-1-2-3.

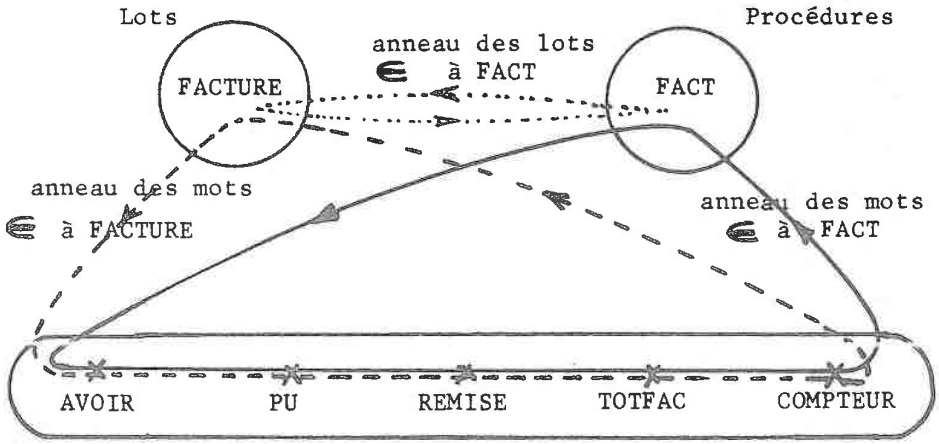
L'exécution du deuxième module de l'algorithme à partir de l'analyse

. du bordereau 1 permet la reconnaissance du nom du lot "FACTURE" et de celui de la procédure FACT à laquelle il est rattaché.

. du bordereau 3 :

NOM	IMAGE	SUM OU SOURCE	IDENT
	Z(4).ZZ	SOURCE	AVOIR
	Z(4).ZZ	SOURCE	PU
	Z(4).ZZ	SOURCE	REMISE
	Z(4).ZZ	SOURCE	TOTFAC
COMPTEUR	Z(6).ZZ	SUM	TOTFAC

entraîne la mise à jour des liens dans la BO, figurée en 9 (ainsi que de tous les liens inverses non figurés).



- figure 9 - Résultats de l'algorithme de dépouillement sur un descripteur de documents (liens directs).

II - 2 - CHOIX DE REPRESENTATION DE LA STRUCTURE CONCEPTUELLE

Comme nous l'avons rappelé au début de ce chapitre, nous avons choisi d'implémenter le schéma conceptuel en base de données, dite Base des Objets Elémentaires.

Compte tenu de nos objectifs opérationnels nous n'avions pas d'autre choix possible , pour gérer cette base, que d'utiliser le SGBD (Système de gestion de base de données) SOCRATE disponible à Nancy sur IRIS 80.

SOCRATE [2] , [9] , [63] , [64] est un SGBD disposant d'un langage de requêtes permettant l'accès à la base en conversationnel ou en batch par interface (cobol en particulier). Ces possibilités sont toutes deux indispensables au déroulement du système tel qu'il est actuellement prévu.

La notion d'entité SOCRATE s'adapte bien à celle d'objet introduit dans le schéma conceptuel, de sorte que la BO comporte six entités :

- l'application
- la procédure
- le module
- le lot
- le mot
- l'entité (au sens donné du terme)

munies des caractéristiques fonctionnelles, dynamiques, sémantiques que nous avons spécifiées au chapitre I et dont nous venons de voir les principes d'évaluation.

Exemple :

L'entité "application" a pour caractéristiques :

- un identificateur
- un résumé
- une désignation en clair

Les relations à traduire sont du type "réseau ", c'est à dire des "anneaux" dans les deux sens.

Exemple :

Un mot peut être rattaché à n modules et inversement un module manipule n mots.

En SOCRATE deux solutions sont possibles pour représenter les liens entre entités :

- soit par imbrication des entités,
- soit par la combinaison anneau-référence (anneau du père vers les fils puis les frères, référence des fils vers le père).

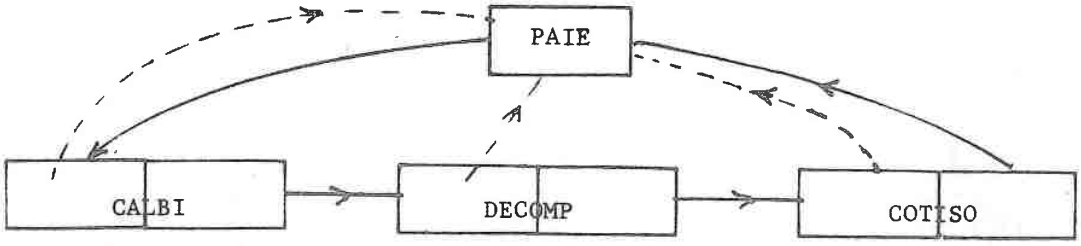
La première solution comporte deux mouvements majeurs :

- une suppression d'un objet de niveau i entraîne la suppression de tous les objets (de niveau > i) qui lui sont inclus.
- on ne peut accéder à un objet de niveau i qu'après avoir accédé à tous les objets de niveau < i, qui nous ont déterminés à choisir la deuxième solution.

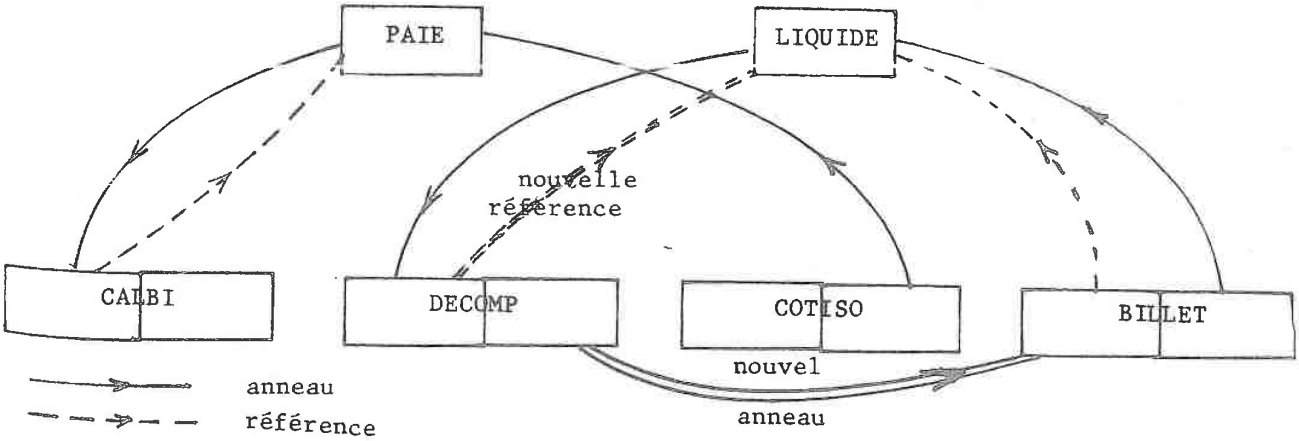
Cependant la seule représentation "anneau/référence" des entités reliées ne permet pas de résoudre entièrement le problème des réseaux car en SOCRATE à un anneau correspond un seul pointeur référence vers le père, un seul vers le frère.

<u>Exemple</u> :

La procédure paie est composée des trois modules CALBI, COTISO et DECOMP.



Si l'on veut représenter le fait que la procédure LIQUIDE est composée de DECOMP et de BILLET :



On constate que les pointeurs de DECOMP sont mis à jour et que nous perdons l'information précédemment établie.

Aussi avons nous introduit des objets fictifs correspondant aux occurrences des relations entre les objets liés. Ces objets fictifs contiennent "deux références" permettant de repérer les deux objets (pères) en relation, les objets liés contenant, eux, des "anneaux".

Cette solution, comme nous l'avons déjà souligné, assure l'indépendance des entités représentées, elle comporte de plus l'avantage de nous permettre la représentation des propriétés des relations.

Exemple :

A la relation mot-module est associée la propriété "mode d'utilisation" d'un mot dans un module qui va devenir une caractéristique au sens "SOCRATE" de l'entité "virtuelle" mot-module.

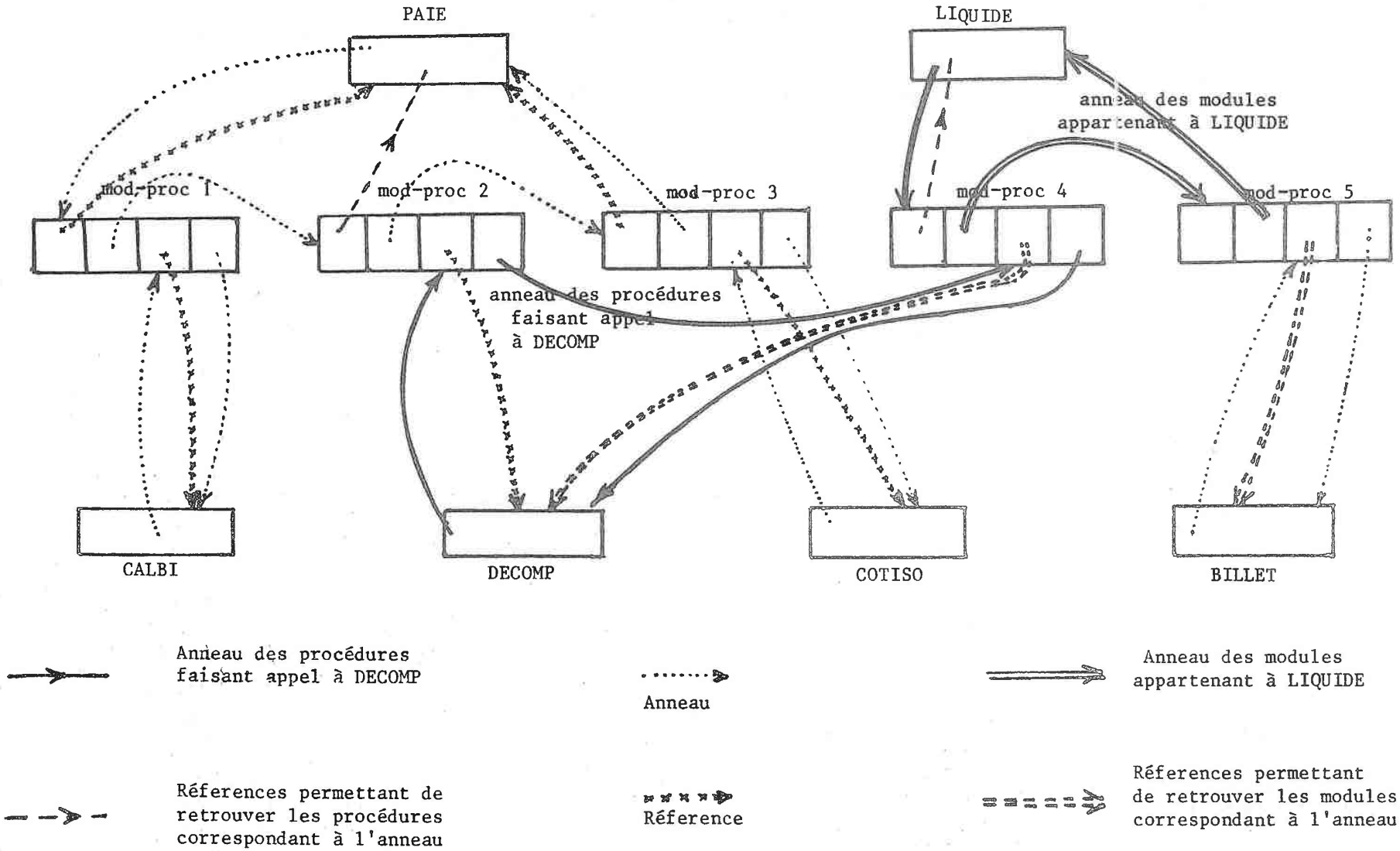
Nous avons ainsi introduit des "objets fictifs" pour tous les liens (n,n) et donc des entités virtuelles dans la structure SOCRATE.

Exemple :

Module-proc représente la liaison entre les modules et les procédures.

La figure 10 montre, sur l'exemple précédent, que l'objet fictif module-proc permet de représenter l'ensemble des relations entre les modules et les procédures.

-figure 10- Représentation des réseaux à l'aide d'objets fictifs.



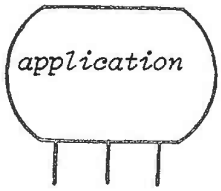
En suivant ce principe pour tous les réseaux à représenter nous aboutissons à la structure de la BO donnée par la figure 11 et à sa description SOCRATE de la figure 12.

On peut facilement établir entre ces deux figures les correspondances mettant en évidence :

. les entités de la base (entourées d'un trait plein figure 11) munies de leurs caractéristiques et leurs déclarations repérées par leur noms (soulignées figure 12).

Exemple :

figure 11 entité



correspond à

figure 12 ENTITE (10) APPLICATION

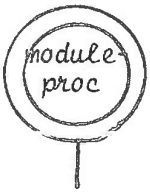
```
DEBUT
 NOM1
  :
FIN
```

. les entités virtuelles de la base (entourées d'un double trait figure 11) munies de leur caractéristique et leurs déclarations repérées par leur nom (soulignées d'un double trait figure 12).

Exemple :

figure 11

entité virtuelle



correspond à

figure 12

ENTITE (1000) MODULE-PROC

```
DEBUT
 TYPROC
  :
FIN
```

. Les anneaux représentés par —>— figure 11 dont les déclarations se trouvent dans les entités (figure 12).

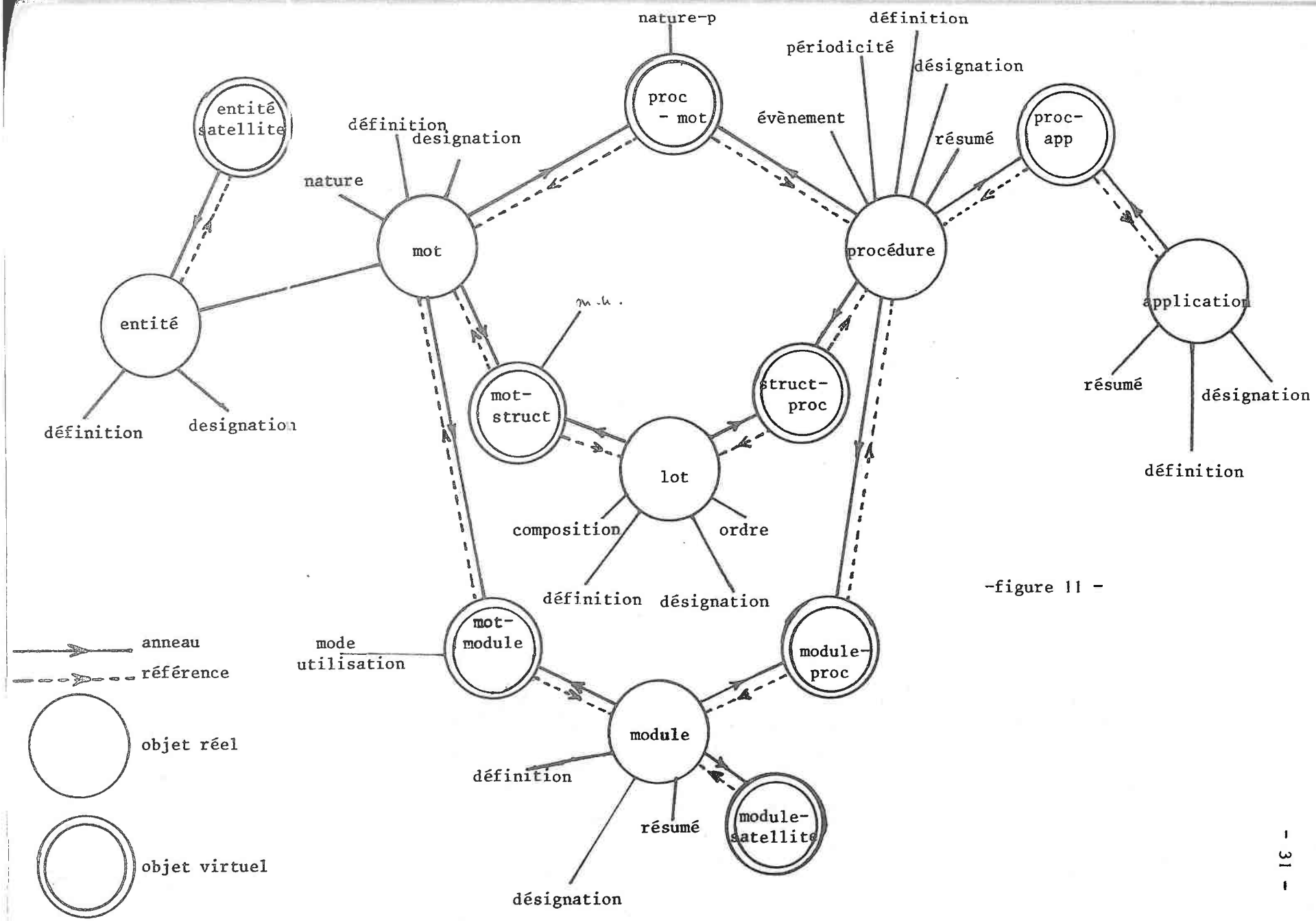
Exemple :

Les anneaux entre les procédures et les modules
sont représentés par —>— *NO-MODU1* ANNEAU
figure 6 —>— *NO-PROC3* ANNEAU

. Les références symbolisées par —•—>—• figure 11 dont les déclarations se trouvent dans les entités virtuelles (figure 12).

Exemple :

Les références entre les procédures et les modules
sont représentées par —•—>•— MODU1 REFERE NO-PROC3
—•—>•— PROC3 REFERE NO-MODU1


nature-p
définition
périodicité
désignation
évènement
résumé
proc
- mot
proc-
app
entité
satellite
définition
designation
nature
mot
procédure
entité
définition
designation
m.h.
mot-
struct
struct-
proc
lot
composition
ordre
définition
désignation
application
résumé
désignation
définition
mot-
module
mode
utilisation
module-
proc
module
définition
résumé
module-
satellite
désignation
anneau
référence
objet réel
objet virtuel

-figure 11 -

```
ENTITE(10) APPLICATION
  DEBUT
   NOM1 MOT(8)
   DESI11 MOT(30)
   DESI12 MOT(11)
   RES1 TEXTE (2 60)
   NO.PROC1 ANNEAU
  FIN
```

entité "application"

```
ENTITE(100) PROC.APP
  DEBUT
   PROC1 REFERE NO.APP DE UNE PROCEDURE
   APPLI REFERE NO.PROC1 DE UNE APPLICATION
  FIN
```

entité virtuelle "proc-app" faisant la liaison entre les procédures et l'application

```
ENTITE(100) PROCEDURE
  DEBUT
   NOM2 MOT(8)
   DESI21 MOT(30)
   DESI22 MOT(11)
   RES2 TEXTE (2 60)
   EVENEMENT MOT(10)
    PERIODICITE MOT(10)
   NO.APP ANNEAU
   NO.STRUCT1 ANNEAU
   NO.MODU1 ANNEAU
   NO.MOT3 ANNEAU
  FIN
```

entité "procédure"

nneau des modules à une procédure

référence à l'anneau des modules

```
ENTITE(1000) MODULE.PROC
  DEBUT
   TYPROC MOT(2)
   MODU1 REFERE NO.PROC3 DE UN MODULE
   PROC3 REFERE NO.MODU1 DE UNE PROCEDURE
  FIN
```

entité virtuelle "module-proc" faisant la liaison entre les procédures et les modules.

nneau des procédures faissant appel à 1 module

référence à l'anneau des procédures

```
ENTITE(1000) MODULE
  DEBUT
   NOM4 MOT(8) AVEC CLE (256 BLSD) FIN
   DESI41 MOT(30)
   DESI42 MOT(11)
   NO.PROC3 ANNEAU
   NO.MOT2 ANNEAU
   MOD.APPE ANNEAU
  FIN
```

entité "module"

```
ENTITE(100) ENTITEB
  DEBUT
   NOM6 MOT(8)
   DESI61 MOT(30)
   DESI62 MOT(11)
   NO.MOT4 ANNEAU
  FIN
```

entité "ENTITEB"

- figure 12 (suite) -

```
ENTITE(5000)  MOTB
  DEBUT
   NOM5 MOT(8) AVEC CLE (256 $LSD)   FIN
   DESI51 MOT(30)
   DESI52 MOT(11)
   PARTI MOT(1)
   NATURE MOT(2)
   IMAGE MOT(10)
   NO-STRUCT2 ANNEAU
   NO-MODU2 ANNEAU
   NO-PROC4 ANNEAU
   ENTIT1 REFERE NO-MOT4 DE UNE ENTITEB
  FIN
```

entité "motb"

```
ENTITE(5000) MOT-MODULE
  DEBUT
    NATMOD MOT(2)
   MOT2 REFERE NO-MODU2 DE UN MOTB
   MODU2 REFERE NO-MOT2 DE UN MODULE
  FIN
```

entité virtuelle "mot-mot" faisant la liaison entre les mots et les procédures.

```
ENTITE(1000) PROC-MO
  DEBUT
  NATPROC MOT(2)
   PROC4 REFERE NO-MOT3 DE UNE PROCEDURE
   MOT3 REFERE NO-PROC4 DE UN MOTB
  FIN
```

entité virtuelle "proc-mot" faisant la liaison entre les mots et les procédures

```
ENTITE(200) LOT
  DEBUT
   NOM3 MOT(8)
   DESI31 MOT(30)
   DESI32 MOT(11)
   TYP3 MOT(2)
   ORDRE MOT(10)
   NO-PROC2 ANNEAU
   NO-MOT1 ANNEAU
  FIN
```

entité "lot"

```
ENTITE(200) STRUCT-PROC
  DEBUT
   STRUCT1 REFERE NO-PROC2 DE UN LOT
   PROC2 REFERE NO-STRUCT1 DE UNE PROCEDURE
  FIN
```

entité virtuelle "struct-proc" faisant la liaison entre les lots et les procédures

```
ENTITE(5000) MOT-STRUCT
  DEBUT
   NAT-STRUCT MOT(2)
   MOT1 REFERE NO-STRUCT2 DE UN MOTB
   STRUCT2 REFERE NO-MOT1 DE UN LOT
  FIN
```

entité virtuelle "mot-struct" faisant la liaison entre le mot et les lots.

```
ENTITE(1000) MODULE-SATELLITE
  DEBUT
   NO-MODULE REFERE MOD-APPE DE UN MODULE
  FIN
```

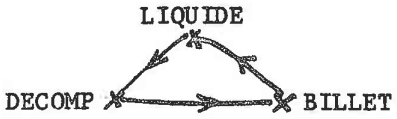
entité virtuelle "module satellite" représentant le lien module-module.

Les entités (réelles ou futures) assurent la possibilité de pénétration dans la base par n'importe quelle occurrence et de parcourir tous les anneaux sur lesquels elle se trouve.

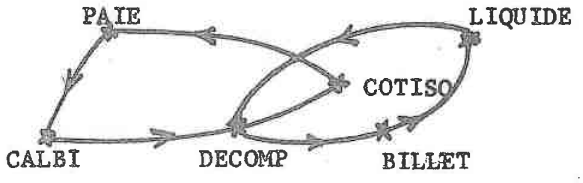
Il est par exemple de parcourir

- l'anneau des modules appartenant à la procédure PAIE

ou celui des modules appartenant à la procédure LIQUIDE



L'intersection des deux anneaux définissant le ou les modules appartenant à la fois aux procédures PAIE et LIQUIDE



De même on pourra parcourir tous les anneaux explicités dans la structure, que ce soient des anneaux de "composants" comme l'anneau des modules d'une procédure ou de "composés" comme l'anneau des procédures comprenant un module.

Il est aussi possible de parcourir des "combinaisons d'anneaux", assurant ainsi les accès nécessaires à une requête du type : Donner tous les mots qui sont rattachés à la procédure "PAIE", (parcours de l'anneau des mots pour chaque occurrence de l'anneau des modules).

Nous donnerons au chapitre 4 un exemple d'utilisation.

CHAPITRE 3

L'AIDE A LA CONCEPTION DU SYSTEME D'INFORMATION

CHAPITRE III - L'AIDE A LA CONCEPTION DU SYSTEME D'INFORMATIONS

III - 0 - INTRODUCTION :

L'objectif de ce chapitre est de définir l'aide à proposer au concepteur dans sa tâche de conception et de construction du système d'informations.

Nous avons émis l'idée, au début de ce travail, que la conception et la construction d'un SI se déroulent en trois étapes : conceptuelle, logique et physique.

Nous limitons notre analyse à la première, ce qui correspond dans Remora à la phase de conception , sachant que l'objectif de cette étape est de parvenir à une description conceptuelle de l'organisation ayant les qualités qui ont été enumérées au chapitre I (cohérence, complétude, extensibilité, non ambiguité, non redondance et fidélité).

On imagine, lorsque le domaine à représenter dans le SI est vaste, qu'il soit difficile à atteindre.

Aussi peut-on se demander s'il est possible de proposer une méthode d'aide au concepteur.

Tout comme [66] , [70] , [53] , [18] , [28] nous répondons par l'affirmative à cette question. Mais plusieurs solutions sont possibles et notre objectif est de définir celle qui serait préférable.

L'essentiel de ce chapitre est donc consacré à l'analyse des fonctions que doit, à notre avis, assurer une méthode d'aide à la conception et aux moyens (c'est à dire aux outils) que l'on peut imaginer pour les remplir.

Nous serons amenés à préciser le type de méthode de conception auquel nous nous intéressons et nous justifierons les fonctions d'aide à la description conceptuelle et des caractéristiques de la méthode de conception d'un "grand" SI.

III - 1 - L'aide à la conception

III - 1-1 - Typologie des méthodes de conception

Cette typologie résulte des réponses possibles aux deux questions :

III - 1-1-1 - Qui aider ?

Qui peut bénéficier des méthodes et outils d'aide à la conception que nous essayons de construire ?

L'orientation actuelle des travaux sur l'analyse de gestion met en exergue le désir de faire participer directement à sa conception du SI le gestionnaire, c'est à dire son utilisateur final.

C'était aussi, semble-t-il, la vocation initiale que TEICHROEW assignait à ISDOS.

Cette position implique le développement de langages sinon naturels au moins quasi naturels par lesquels le gestionnaire puisse "définir" les problèmes tels qu'il les ressent ; elle implique d'autre part un automatisme qui, à la limite, devrait être total.

Il nous semble difficile de soutenir cette position dans l'état actuel des propositions que peuvent faire les chercheurs en matière d'outils d'aide à la conception. Il est plus raisonnable de penser que la conception soit assurée par des concepteurs ayant la connaissance du modèle conceptuel et de sa correspondance avec celui du monde réel que l'on ne peut pas exiger de tous les gestionnaires.

Cette solution n'écarte pas l'intérêt de développer des langages de conception aussi proches que possible des langues naturelles, avec, en particulier, le souci d'obtenir que le gestionnaire puisse comprendre ce langage et apporter une contribution locale pour la définition du problème qu'il connaît bien.

Elle n'exclut pas non plus la recherche des outils qui permettront d'atteindre l'automatisme complet mais qui, dans une première étape, pourront de façon ponctuelle et peut être incomplète (par exemple parce qu'ils proposent plusieurs solutions) aider le concepteur à faire le meilleur choix.

III - 1-1-2 - Comment aider ?

On peut envisager tout un continuum de solutions qu'a déjà suggérées d'ailleurs le paragraphe précédent, allant d'une méthode entièrement manuelle qui offre simplement un canevas, à un automate qui, partant des définitions exprimées par les gestionnaires, construit intégralement le schéma conceptuel, en passant par des solutions intermédiaires de conception assistée.

Une méthode manuelle de conception propose, un canevas, un fil conducteur mettant en évidence les tâches à exécuter et peut-être les difficultés ou écueils à éviter. Elle se fonde aussi sur l'interprétation de l'homme et sa liberté de choix. C'est peut être un avantage permettant d'atteindre "l'optimum" parce que l'homme est capable de prendre en compte certains critères de choix qui peuvent être essentiels et qu'on ne sait pas encore numériser. Mais l'interprétation peut être erronée et les choix trop subjectifs et en tout état de cause cette souplesse est un danger pour la prise en compte des facteurs d'évolution et conduit à une mauvaise adaptabilité du SI.

Une méthode entièrement automatisée se propose, à partir des énoncés qui lui ont été fournis, d'aboutir, sans aide complémentaire des concepteurs, à la description conceptuelle.

C'est dans ce sens qu'interviennent les propositions d'ISDOS puisqu'à partir des énoncés rédigés dans le langage d'expression des problèmes (PSL), le système se charge d'effectuer les meilleurs choix lui permettant d'aboutir, dans ce cas, au-delà du stade conceptuel à la construction physique des programmes et à l'élaboration des fichiers.

Mais le PSL est-il véritablement un langage de niveau conceptuel ? Les véritables problèmes de conception n'ont-ils pas été résolus, au moins partiellement, à l'avance ?

L'automatisation totale comporte, à notre sens, trop de risques : elle implique une maîtrise des problèmes de la conception des SI dont on peut douter qu'elle soit déjà garantie.

La conception assistée est une solution intermédiaire possible entre les deux solutions extrêmes que nous venons d'énoncer. Elle associe l'automatisation apportée par des outils et la possibilité de certains choix laissée à l'homme dans les cas où la décision automatique est jugée impossible ou risquée.

Ce choix qui a été reconnu dans les méthodes d'analyse telles que [43] , [27] , [56] , [57] ... ou des projets de conception tels que [70] , [53] , [39] , [18] , [28] ... est aussi le nôtre dans REMORA.

III - 1-2 - Les caractéristiques du procédé de conception retenu et les erreurs qu'il engendre.

Dans l'optique où nous nous plaçons, il semble que la conception du SI ne puisse être réalisée que modulairement, par sédimentations successives et par plusieurs concepteurs ayant à traduire la multiplicité des points de vue de la masse des futurs utilisateurs du SI. Chacune de ces caractéristiques engendre son propre lot de potentialité d'erreurs et de non satisfaction de l'objectif assigné au schéma conceptuel.

La Modularité :

Compte tenu de l'ampleur des problèmes à résoudre, l'analyse de l'ensemble des phénomènes de l'organisation pris en compte ne peut se dérouler que par parties.

Mais est-on certain que le découpage soit parfait ? en d'autres termes, les sous-problèmes délimités sont-ils des modules ?

Dans la négative le partage sur lequel s'appuie l'analyse va engendrer des incohérences.

La sédimentation :

La conception du SI n'est pas réalisé en une seule fois mais par couches successives dans le temps.

Ainsi dans Remora la procédure "gestion du personnel" pourra être définie partiellement à un instant t_o(paie), complétée à un instant t_1 (gestion des travaux) et à un instant t_2 (gestion comptable).

Cette construction du SI par sédimentation peut entraîner des redites, des contradictions, des oublis.

La multiplicité des concepteurs :

Il n'est guère possible d'envisager qu'un même concepteur même "Administrateur de SI" puisse, seul, définir conceptuellement le SI. Dans la mesure ou plusieurs concepteurs interviennent, des redites, des ambiguités et des contradictions sont probables.

La Multiplicité des points de vue :

Les concepteurs doivent leur compte de la multiplicité des points de vue exprimés par les utilisateurs potentiels du SI.

Ils doivent en effet, traduire l'ensemble des visions du monde réel qu'ont les utilisateurs et qu'ils souhaitent retrouver dans le SI.

Cette hétérogenéité peut entraîner des redondances et des incohérences dans la description conceptuelle car le concepteur peut croire que deux visions exprimées sont distinctes ou au contraire identiques alors qu'elles ne le sont pas.

III - 1-3 - Les fonctions d'aide à la conception

Sachant que nous nous intéressons à une méthode de conception assistée limitée au seul niveau conceptuel, notre propos, dans ce paragraphe, est de mettre en évidence les fonctions que doit assurer le système d'aide à la conception du type retenu.

Ces fonctions ont pour but d'éléiminer les problèmes soulevés au paragraphe précédent et inhérents au type de conception auquel on s'attache, de manière à permettre l'obtention d'une description conceptuelle ayant toutes les caractéristiques que nous lui avons définies au chapitre I : cohérence - complétude - extensibilité - non ambiguité - non redondance - fidélité.

Elles apparaissent dans le tableau croisé de la figure 1 mettant en relation els objectifs à atteindre à l'étape conceptuelle et les caractéristiques de la conception engendrant les difficultés inventoriées au paragraphe précédent.

caractéristiques / qualités	conception modulaire	conception par sédimentation	multiplicité des concepteurs	héterogénéité des représentations
cohérente	documentation controle		documentation controle	documentation controle
complète	intégration controle	intégration	intégration	
extensible		intégration		
non ambigüe		documentation intégration	documentation intégration	documentation intégration
non redondante	documentation intégration controle	documentation intégration	documentation intégration	documentation intégration
fidèle	documentation controle	documentation intégration		

- figure 1-

Les deux fonctions nommées CONTROLE et INTEGRATION assurent des contrôles au sens littéral du terme mais à des niveaux différents.

La fonction CONTROLE a pour but lors de l'introduction d'un nouvel énoncé, d'une part, de vérifier qu'il est une représentation fidèle du phénomène pris en compte, d'autre part de vérifier sa cohérence interne.

La fonction INTEGRATION a pour objectif, sachant que la fonction contrôle a donné toutes les garanties sur la correction d'un énoncé, de vérifier la cohérence de son introduction dans l'ensemble des énoncés existants, la complétude de cet ensemble et la fidélité de la représentation de l'ensemble des phénomènes du monde réel étudiés.

La fonction DOCUMENTATION doit satisfaire le besoin qu'a le concepteur de connaître l'état de conception du SI à l'instant où il désire intervenir sur cet état.

Ces trois fonctions que l'on va analyser en détail au paragraphe suivant nous semblent recouvrir l'ensemble des besoins en matière d'aide à la conception au niveau où nous nous plaçons. La réflexion que nous proposons de limiter à ce niveau lui donne son originalité mais rend difficile les comparaisons avec les propositions d'autres chercheurs tel Waters (70, p. 17-18) qui ont plus volontiers raisonné sur les deux niveaux logiques et physiques de la construction d'un SI.

III - 2 - ANALYSE DETAILLEE DES FONCTIONS

III - 2-1 - La fonction contrôle

III - 2-1-1 - Définition :

Si nous appelons énoncé l'image descriptive dans le schéma conceptuel d'un phénomène du monde réel, la fonction contrôle recouvre l'ensemble des actions permettant de s'assurer de la correction de cet énoncé.

Un énoncé peut être déclaré correct s'il a certaines des qualités que l'on exige du schéma conceptuel c'est à dire s'il est cohérent, non ambigü et s'il est une image fidèle du phénomène décrit;ceci conduit à définir dans un premier temps deux types de contrôles sémantiques :

α : des contrôles globaux de l'énoncé.

β : des contrôles de fidélité.

Un énoncé est un tout décrit en une seule fois et sémantiquement indissociable (par exemple le module descriptif d'une définition du monde réel dans Remora) qui s'exprime à partir d'objets élémentaires de la structure conceptuelle dont il fournit, en fait, la définition (le module Remora exprime, par exemple, les propriétés des mots).

On peut imaginer, dans un deuxième temps, un troisième type de contrôle sémantique portant sur les corrections des éléments pris individuellement.

γ : les contrôles sur les éléments.

Enfin, l'énoncé est exprimé dans un langage (par exemple le langage de conception de Remora [36]) dont il faut vérifier la syntaxe, ce qui induit un dernier type de contrôle.

δ : les contrôles syntaxiques.

III - 2-1-2 - Les contrôles et les moyens envisageables pour les assurer

α - Les contrôles sémantiques globaux de l'énoncé

La question que l'on se pose ici est :

<<L'énoncé est-il cohérent, a-t-il un sens ?>>

Mais comment définir "le sens" d'un énoncé ?

Il semble que pour une structure conceptuelle donnée on ait défini une structuration de ses éléments dont les caractéristiques doivent être respectées dans tout énoncé. Par ce biais pourraît être opérée une certaine vérification de sa cohérence.

On peut aussi imaginer de transposer, au niveau conceptuel, les aides à la mise au point que l'on a vu apparaître dans les méthodes d'analyse [43] , [57] telles que la simulation. Ainsi, dans Remora, il existe un simulateur de maquette fonctionnant à partir du seul énoncé descriptif d'un document à éditer ; ce qui est un cas simple de définition de tels contrôles mais qu'on envisage d'étendre aux énoncés de problèmes dans les modules.

β - Les contrôles sémantiques de fidélité

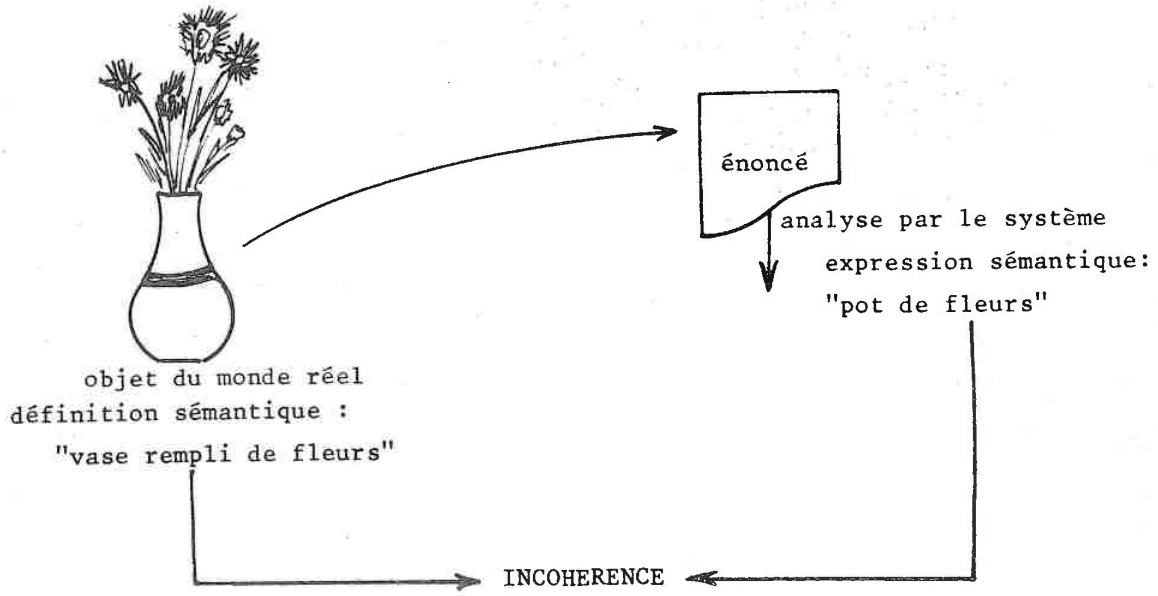
Par les contrôles on doit pouvoir répondre à la question.

<<L'image du phénomène réel perçue à travers l'énoncé est-elle représentation fidèle de la réalité>> ?

Les moyens dont on dispose actuellement pour assurer les contrôles de fidélité sont très limités : c'est souvent le concepteur qui vérifie en interprétant lui-même le résultat de sa description.

Est-il possible d'envisager une solution automatique ?

Une solution peut consister à mettre en parallèle, une définition sémantique du phénomène réel qu'aurait fournie le concepteur et l'expression sémantique que l'on peut déduire de l'énoncé.



L'explication sémantique pourrait être exprimée au moyen d'un langage naturel ou pseudo-naturel reconnaissable par un automate ; l'expression sémantique déduite de l'énoncé pourrait être composée par le système à partir des caractéristiques de l'image de l'objet représenté.

Exemple : On pourrait dans Remora vérifier la concordance entre le résumé du module et l'expression de son sens par le concepteur.

Cette solution s'apparente aux méthodes de preuves de programmes développées en programmation : les insertions sont une autre expression de l'algorithmique du programme qui doivent fournir des résultats identiques à ceux normalement obtenus par l'exécution du programme.

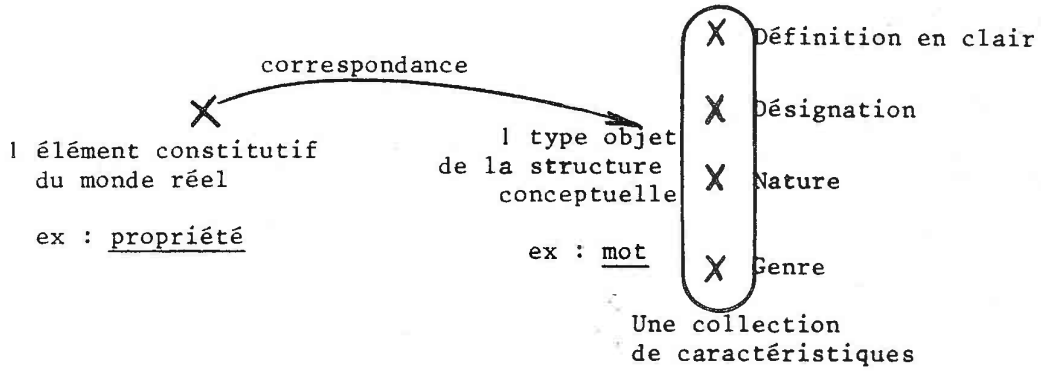
γ - Les contrôles sémantiques sur les éléments

De manière symétrique à la question que nous nous sommes posée en α, on se demande ici :

<<Chacun des éléments d'un énoncé a-t-il, pris individuellement, un sens ?>>

Et comme précédemment il faut se demander quelle peut être l'expression sémantique d'un élément reconnaissable par un automate ?

A chacun des types d'objet de la structure conceptuelle est associé un ensemble de caractéristiques telles que celles que nous avons définies dans REMORA au chapitre 1.



Chacune de ces caractéristiques prise individuellement ne peut pas totalement lever l'ambiguité sur le sens d'un élément, mais la collection de ses caractéristiques peut jouer ce rôle.

Par exemple, dans REMORA, la définition en clair, la nature d'un mot et son rattachement à une entité, en donnent un sens plus précis et plus sûr que chacun de ces caractéristiques puis isolèment.

On peut aussi définir des règles de compatibilité ou d'incompatibilité sur les champs de valeurs des caractéristiques.

Un moyen de contrôler la validité sémantique d'un élément peut être de vérifier que ces règles sont respectées.

Nous reconnaissons qu'il n'est pas aisé de trouver de telles règles, d'autant qu'il n'en exite pas obligatoirement entre toutes les caractéristiques. Mais nous développons ce moyen de contrôle dans Remora parce que nous sommes convaincus de son importance.

1 - Ainsi par exemple, dans le champ des valeurs de la caractéristique mode d'utilisation attachée à un mot, les deux valeurs "DONNEE" et "RESULTAT" s'excluent mutuellement.

Dans un même module, un mot reconnu à la fois en "Donnée" et en "Résultat" peut être déclaré anormalement défini.

2 - Nous pourrions poser comme hypothèse que si le mot clé "CUMUL" est détecté dans la définition en clair d'un mot, son mode d'utilisation ne être que résultat.

3 - De même si la définition en clair d'un mot contient le mot-clé "TABLEAU" son genre ne pourra être que "T".

δ - Les contrôles syntaxiques

Les moyens disponibles au niveau "programmation", proposés par la plupart des systèmes (avec en particulier les assembleurs et les compilateurs) sont translatables au niveau conceptuel, à condition que le langage de conception soit muni d'une syntaxe contrôlable.

C'est le cas en particulier du PSL d'ISDOS [66] et du langage de conception de REMORA [36] .

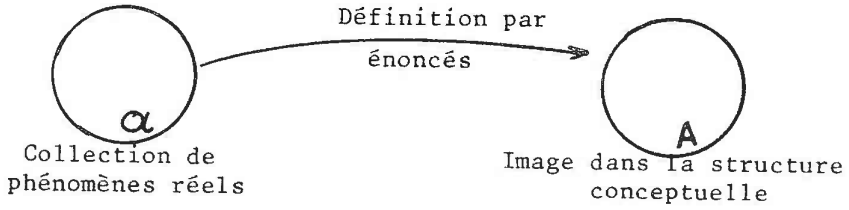
Le moyen envisageable est alors l'analyse de l'expression de la définition afin de vérifier si elle respecte la grammaire du langage. Nous ne nous étendrons pas sur ce domaine devenu classique.

III - 2-2 La fonction Intégration

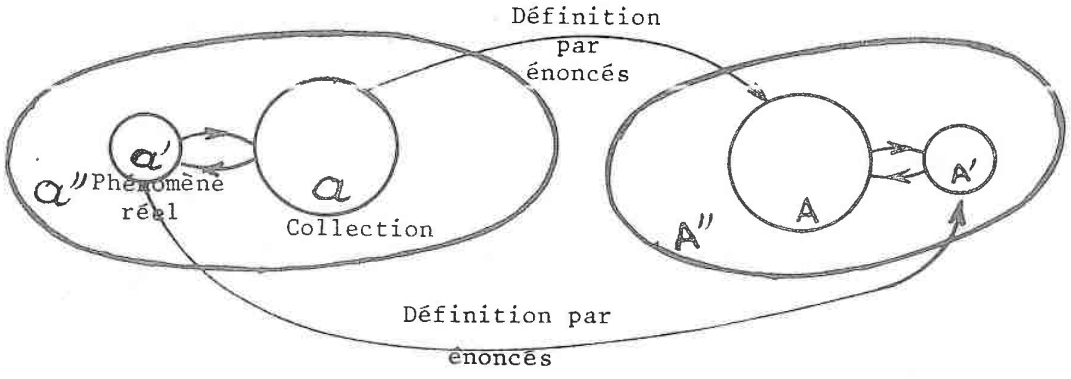
III - 2-2-1 - Définition

Elle recouvre l'ensemble des actions à entreprendre pour s'assurer de la cohérence de l'insertion d'un énoncé dans l'ensemble des énoncés déjà admis dans le schéma conceptuel.

Plus précisément, partant de l'hypothèse qu'une certaine collection $\mathcal{a}$ de phénomènes du monde réel à déjà été décrite par une collection $\mathcal{A}$ d'énoncés qui a toutes les propriétés requises : cette collection est cohérente, non ambigue et représente fidèlement $\mathcal{a}$.



Soit $\mathcal{a}$ un phénomène réel apparenté à la collection $\mathcal{a}$ et soit $\mathcal{A}'$ l'image dans le schéma conceptuel. La fonction CONTROLE a permis de s'assurer que $\mathcal{A}'$ est une image cohérente fidèle de $\mathcal{a}$.



La fonction INTEGRATION a pour objectif de vérifier que la nouvelle collection d'images $\mathcal{A}$" est d'une part cohérente et complète et que, d'autre part, elle est une image fidèle de la collection $\mathcal{a}$", en d'autres termes, que les relations de $\mathcal{a}'$ avec $\mathcal{a}$ ont été correctement traduites sous la forme des relations de $\mathcal{A}'$ avec $\mathcal{A}$.

Les fonctions controle et intégration interviennent à des niveaux différents. Les controles du premier niveau CONTROLE permettent de verifier la qualité d'un énoncé pris isolément,ceux du second niveau INTEGRATION permettent de verifier que l'insertion d'un nouvel énoncé ne détruit pas les qualités de l'ensemble déja décrit et définit un nouvel ensemble ayant ces mêmes qualités.

Les contrôles assurés par la fonction INTEGRATION peuvent être classés comme précédemment en :

α : Contrôles sémantiques sur les éléments

β : Contrôles globaux sur la cohérence du nouveau SC.

γ : Contrôles de fidélité.

auxquels s'ajoutent

δ : des contrôles de complétude.

III - 2-2-2 - Les contrôles et les moyens envisageables pour les assurer

α : Les contrôles sémantiques des éléments

Il s'agit ici de vérifier que chaque élément à un sens, non plus, comme pour la fonction CONTROLE par rapport à lui-même, mais par rapport au contexte le plus large de l'ensemble des éléments déjà acceptés dans le schéma conceptuel.

La question immédiate à se poser est la suivante :

<<S'agit-il d'un élément nouveau ou d'informations complémentaires sur un élément déjà décrit ?>>

Le concepteur peut avoir pensé définir un élément nouveau alors qu'il ne l'est pas. Le moyen de reconnaître un élément dans le SC est sa désignation. Un élément χ sera donc déclaré nouveau s'il n'existe pas d'objet dans le SC qui ait la même désignation et une définition équivalente.

Le contrôle à effectuer consiste à vérifier que le SC ne comprend aucun élément qui, bien que portant une désignation différente, ait les mêmes caractéristiques que χ .

Ce contrôle correspond donc à la détection des cas de synonymie implicite c'est à dire involontaire.

L'abandon de ce contrôle aurait pour conséquence des redondances sémantiques d'informations dans le SC qui pourraient conduire à des incohérences.

Inversement, le concepteur peut utiliser une désignation sans savoir qu'elle avait déjà été attribuée à un autre élément représentatif d'un objet du monde réel différent et créer ainsi un polysème.

Ce contrôle correspond à la détection des cas de polysémie.

Ces cas sont inacceptables car la correspondance entre les éléments constitutifs du monde réel et les objets-types du modèle conceptuel serait mise en défaut (chapitre 1) et conduirait à un SC ambigu.

α1 : La détection des synonymes implicites

Plusieurs solutions sont envisageables.

La première consiste à comparer systématiquement toutes les caractéristiques de l'élément "nouveau" à celles de tous les objets de même type appartenant au SC.

Exemple dans Remora

Vérifier pour chaque mot du nouveau module introduit s'il existe un autre mot dans la description conceptuelle ayant la même définition en clair, le même genre et les mêmes modes d'utilisation.

Cette recherche est lourde et difficile à mettre en oeuvre si les caractéristiques ne sont pas toutes valorisées et facilement exploitables par un outil (par exemple si les désignations en clair ne sont pas structurées par un système du type mots-clés).

Une deuxième solution vise à restreindre la recherche :

. d'une part, en limitant le contexte de l'élément nouveau.

Exemple Remora :

Lors de la prise en compte d'un nouveau mot, on ne fera la recherche de synonyme que dans le contexte de la procédure en cours (et non pas de l'application).

. d'autre part en effectuant des recherches d'identité uniquement sur les caractéristiques qui sont déjà, sous une certaine forme, des expressions sémantiques.

Exemples dans Remora :

La définition en clair des objets - le résumé par mots-clés des modules.

Il est possible d'effectuer une recherche documentaire sur la définition en clair de l'élément nouveau et sur celle des objets de même type (à condition bien sûr de les avoir exprimées par un moyen tel que celui des mots-clés).

Peut être serait-il possible alors de concevoir une solution qui définit le nombre minimum de mots-clés identiques qu'il faudrait trouver pour conclure à la présence d'un synonyme ?

Cette solution, outre le fait qu'elle risque de laisser subsister des anomalies est encore lourde et désordonnée.

Les moyens actuellement disponibles ou imaginables sont loin d'être satisfaisants. On rencontre comme dans le paragraphe précédent la nécessité de définir des langages de spécifications permettant d'exprimer la sémantique d'un énoncé ou d'un objet élémentaire et qui soient utilisables par des outils.

Ce souci rejoint celui des chercheurs qui se préoccupent de bâtir au niveau de l'implémentation, des systèmes documentaires qui leur permettraient de résoudre les problèmes de la production automatique de programmes.

A ce propos on peut citer les travaux de DERNIAME à NANCY [23] et de MILLS [44] .

α2 : La détection des polysèmes

Plusieurs solutions sont ici aussi envisageables.

La première consiste, chaque fois que par l'introduction d'un énoncé le système prend en compte un élément, à chercher si un élément de même désignation existe déjà dans le SC et en conclure que, si oui, il s'agit d'une anomalie.

Au niveau des systèmes d'aide à l'analyse c'est ainsi que procèdent MACSI [53] et SCAPFACE [39] .

L'inconvénient de cette solution est évident puisque chaque fois que le concepteur cite un composant du SC le système le considère comme polysème et le rejette.

Cette solution, acceptable à la limite si toutes les définitions sont fournies simultanément, est inacceptable en revanche dans une approche modulaire acceptant la sédimentation.

Exemple :

Si l'on procédait ainsi dans REMORA le mot SALNET (salaire net d'un employé), apparaissant sur plusieurs documents (par exemple : le bulletin de salaire et le bordereau de règlement), serait toutes les fois, à partir de la seconde, rejeté par le système comme polysème.

La solution que nous adoptons consiste à utiliser (comme nous l'avons montré dans la fonction CONTROLE) l'expression de la sémantique d'un élément que l'on peut déduire des valeurs de ses caractéristiques.

Si parmi les caractéristiques il en est une qui soit déjà sémantique (expression par mots-clés du type de celle que nous avons introduite pour la synonymie) on peut envisager de limiter le contrôle à l'exploration de cette caractéristique.

Par exemple, si la désignation de mot SALNET est associée dans le SC et dans l'énoncé pris en compte, à une même définition en clair par mots-clés (SALAIRE DE EMPLOYE), on peut conclure qu'il s'agit de la représentation du même objet du monde réel (le salaire net de l'employé).

Cependant cette recherche est imprécise. Le langage des mots-clés, laisse au concepteur la liberté de définir à deux moment distincts un unique objet par des expressions non totalement identiques. Il sera difficile de reconnaître automatiquement que les deux expressions se rapportant en fait au même objet.

Il peut exister, parmi les caractéristiques d'un élément-type du SC, une caractéristique qui soit plus pertinente et que l'on choisisse pour effectuer la recherche.

Par exemple, dans REMORA, la caractéristique nature d'un mot est définie en extension par une liste de valeurs dont certaines s'excluent.

Ainsi si un mot déclaré de nature "DONNEE" dans un module, apparaît avec la même désignation dans un nouveau module muni de la nature "RI" on peut conclure que ces mots sont des polysèmes.

Mais cette investigation peut se révéler insuffisante :

Exemple :

Un mot désigné par X et défini dans 2 modules avec une nature "Résultat" sera considéré comme représentation d'un objet unique alors, qu'en fait, X peut vouloir désigner dans le 1° module le cumul des salaires nets et désigner dans le 2° module le cumul des cotisations.

On aboutit donc à la conclusion que la méthode d'investigation, pour être sûre, devra s'étendre à l'ensemble (ou à un sous-ensemble significatif) des caractéristiques définies pour un élément.

Ainsi, par exemple, dans REMORA, on pourrait utiliser à la fois la désignation en clair d'un mot et sa nature, le résumé d'un module et son type.

On trouvera au chapitre 4, les algorithmes développés à cette fin dans REMORA.

β : Les contrôles globaux sur la cohérence du nouveau SC

Il s'agit de tenter de répondre à la question :

<<Le nouvel état du SC est-il un état cohérent ?>>

La cohérence de la nouvelle collection d'énoncés qui correspond au nouvel état du SC n'apparaît qu'au travers de la cohérence de la structuration de l'ensemble.

C'est donc la validité de la structuration du SC qu les contrôles de ce type devront essayer de vérifier.

Compte tenu de la structuration existante sur les éléments types du modèle conceptuel, il faut vérifier d'une part, que les occurrences de ces éléments respectent les règles de structuration, d'autre part, que les valeurs des caractéristiques des éléments ne sont pas incompatibles avec ces règles.

En particulier, certaines caractéristiques temporelles doivent permettre de vérifier la cohérence dans le temps des différents éléments du SC. On pourra réaliser cela par une vérification de la continuité de la séquence temporelle des actions à entreprendre ou des états successifs que doit prendre la représentation de la caractéristique d'un objet.

Exemple :

Dans REMORA la structure en modules d'une procédure doit respecter les relations temporelles entre les modules qui s'expriment à travers les lots d'informations transitant d'un module à un autre.

Dans ISDOS on propose de représenter les structures par des graphes pour vérifier si tous les axes (ou flux de données) sont définis et si tous les process (ou module de calcul) s'enchaînent correctement par rapport aux données.

γ : Les contrôles de fidélité

Ils ont pour objet de s'assurer que le nouvel état du SC est une image fidèle de l'ensemble des phénomènes pris en compte. Soit si l'on se réfère au schéma introductif de la p.14 , que A'' est une image fidèle de a "

Ceci en fait n'est possible que si A " a un sens propre défini autrement qu'à partir du sens qui le composent.

Autrement dit, il nous semble que les contrôles de fidélité sont acceptables qu'aux états de la conception du SI qui correspondent à la valorisation complète d'un objet de niveau supérieur à celui de l'énoncé dans la structure conceptuelle.

Ces états sont la représentation d'une certaine classe de phénomènes définie dans la hiérarchie des phénomènes réels pris en compte.

Par exemple, dans Remora, on conçoit que les contrôles de fidélité soient possibles au niveau d'un énoncé (module). Ils sont alors pris en charge par la fonction CONTROLE puis au niveau d'une procédure, et ils sont alors pris en charge par la fonction INTEGRATION, et enfin au niveau de l'application.

Dans cette hypothèse les moyens à mettre en oeuvre sont du même type que ceux que nous avons proposés pour les contrôles de fidélité de la fonction CONTROLE.

δ : <u>Les contrôles de complétude</u>

Ils ont pour objectif de s'assurer que "l'état du SC est une représentation complète de l'ensemble des phénomènes pris en compte".

Nous faisons ici les mêmes restrictions que précédemment. Ces contrôles ne peuvent s'éxécuter que sur un état du SC qui soit la représentation d'une classe de phénomènes du monde réel que le concepteur a décrite intégralement.

Il semble qu'ils puissent être définis à partir de <u>règles de complétude</u> qui faut préciser dans chaque cas particulier de modèle conceptuel.

Par exemple, dans REMORA, au niveau d'une procédure :

. tous les lots et les modules doivent être décrits.

. tous les mots doivent avoir reçu leur caractéristique "définition en clair".

. tous les modules doivent avoir reçu la caractéristique "résumé".

. tous les mots qui n'ont pas reçu un mode d'utilisation autre que SOURCE (c'est à dire appartenant à un document de sortie) et qui ne sont pas des DONNEES de la procédure sont des résultats qui n'ont pas été définis.

On touve dans ISDOS des règles de complétude du même type ;par exemple : tout résultat figurant sur un document devant avoir été calculé, on vérifie qu'il existe éffectivement des traitements (ou process) à partir desquels ces résultats peuvent être déduits.

III - 2-3 - La Fonction Documentation

III - 2-3-1 - Définition

Elle recouvre l'ensemble des actions qui permettent de renseigner le concepteur à tout instant sur l'état du SI qu'il conçoit.

Remarquons cependant que si le concepteur a besoin du système documentaire, le système d'aide à la conception en a également besoin .

Il y a donc, en fait, deux types d'utilisateurs du système documentaire : le concepteur qui est l'intervenant extérieur, et le système qu'on peut qualifier d'interlocuteur interne.

Par exemple, dans REMORA, les analyseurs qui sont les outils associés à la fonction INTEGRATION précédemment décrite, exigent, pour leur fonctionnement, une connaissance de l'état du SI que leur donne le documenteur, outil chargé de la documentation.

Pour mettre en oeuvre la fonction documentation il faut résoudre trois problèmes :

1 - définir un stock documentaire pertinent.
2 - disposer de moyens permettant de créer et de mettre à jour ce stock
3 - disposer de moyens d'interrogation de ce stock.

III-2-3-2 -Mise en oeuvre :

α : Quel type d'informations doit-on mettre dans le stock documentaire ?

Dans l'hypothèse où nous nous sommes placés, la réponse à cette question est toute trouvée. Les types-objets que l'on a définis dans le modèle conceptuel, munis des caractéristiques et des relations choisies, sont, par définition, les éléments nécessaires à une "bonne" représentation conceptuelle du SI.

Le stock documentaire du niveau conceptuel est donc le schéma conceptuel.

Le stock documentaire, est pour cette raison tout à fait différent des dictionnaires de données (DD*) que proposent les méthodes d'analyse, bien que les objectifs d'utilisation soient de même nature.

Les DD sont constitués à partir des objets du modèle d'implémentation c'est à dire les fichiers - les programmes - les identificateurs de rubriques .

* Voir en fin de chapitre la bibliographie sur les DD.

β : Comment gérer le stock documentaire ?

Cette gestion dépend de deux facteurs :

- le choix d'implémentation du stock documentaire
- le choix d'expression des éléments du modèle conceptuel.

β1 - Pour implémenter le stock documentaire, qu'il soit du niveau conceptuel ou d'un autre niveau, on peut choisir ou la solution fichier ou la solution Base de données.

Il semble que, compte tenu de la multiplicité des objets à implémenter et des relations qui les lient, compte tenu également de la complexité de certains liens auxquels sont rattachées des caractéristiques, la deuxième solution soit plus adaptée.

C'est le choix que nous avons fait dans REMORA puisque la Base des Objets élémentaires qui correspond à l'implémentation du SC est structurée en base de données SOCRATE (sa description a été fournie au chapitre 2).

C'est aussi le choix de la méthode d'analyse MINOS, par vocation, cette méthode de type analytique centrée sur l'étude des informations, a développé très tôt les DD, et remplacé les anciens répertoires gérés dans des fichiers (mots - fichiers - modules programmés) par une base de données.

β2 : La gestion du stock documentaire recouvre les deux fonctions, de reconnaissance des types d'objets, de valorisation de leurs caractéristiques et de leurs relations, d'une part, de stockage de ses informations suivant le type d'implémentation retenu, d'autre part.

Nous ne nous étendrons pas sur le classique deuxième point mais davantage sur le premier.

La reconnaissance sera plus ou moins facilitée par la manière dont le concepteur fournit ses énoncés. Le travail de reconnaissance se limite à un dépouillement syntaxique du texte en langage de Définition des Données LDD, si, comme dans la pratique actuelle de la définition d'une base de données, le concepteur fournit par l'intermédiaire d'un LDD la description complète de chaque constituant et de ses relations avec les autres constituants.

Si, au contraire, comme c'est le cas dans REMORA, la méthode propose au concepteur de raisonner en termes d'éléments constitutifs du monde réelet de définir les problèmes à partir de cette réflexion, la reconnaissance consiste à repérer les objets de la structure conceptuelle à partir des éléments du monde réel correspondants et suppose donc un travail qui est, pour certains aspects, plus complexe que le dépouillement précédent.

Nous avons pour REMORA défini les algorithmes de reconnaissance pour la gestion de la BO au chapitre II.

γ : Quels moyens peut-on fournir pour l'interrogation du stock documentaire ?

La réponse à cette question est très fortement dépendante du type d'implémentation retenu. A une implémentation par fichiers, va correspondre un système d'interrogation plus lourd et surtout beaucoup plus procédural qu'à une implémentation base de données.

L'intérêt de l'implémentation en base de données apparaît plus manifeste ici, dans la mesure où, au niveau conceptuel, il est essentiel de pouvoir fournir au concepteur un langage d'interrogation non procédural qui lui permette par de simples citations d'accéder au stock documentaire, sans être obligé de connaître l'implantation dans les fichiers et de naviguer à travers leurs enregistrements.

Ainsi dans REMORA le concepteur peut, par l'intermédiaire du langage de requête SOCRATE, poser des questions telles que

quels sont les mots de la procédure PAIE ?
ou quels sont les modules où le mot SALNET est calculé ?
ou quels sont les résultats transmis du module CALBI au module CALCOS ?

..... etc .....

L'avantage pratique de ces questions apparaîtra mieux dans le procédé d'introduction d'un nouvel énoncé que nous développons au chapitre IV.

BIBLIOGRAPHIE

Articles ayant trait aux dictionnaires de données:

-DATA BASE JOURNAL

I- Data manager a free standing
- Data dictionary system vol 6 n°5

2- Lexicon an established Data
Dictionary system vol 6 n°7

3- The Data Dictionary
vol 6 n°2

-IBM System Journal

Data Dictionary/Directories by P.P.UHROWCZIK
vol I2 n°4

-L.MOISNI

Le dictionnaire des données:
un prealable indispensable
OI Informatique Octobre I974

-EDP Analyzer

The Data Dictionary/Directory Function
vol I2 n°II

-A.F.COLLARD

"A data dictionary directory"
Journal of Systems Management June I974
pp. 22-25. A discussion of the EAL System

Les dictionnaires d'information et les méthodes d'analyse:

-Méthode MINOS Sligos Systemes

Systéme BIBLOS (Répertoires sous forme de Base de données)

-CGI
"CORIG-PAC"
-Promo-Informatique
"ARMIN-PARM"
(Fichiers documentaires)

CHAPITRE 4

LA PHASE DE CONCEPTION DANS REMORA

CHAPITRE IV - LA PHASE DE CONCEPTION DANS REMORA

IV - 0 - INTRODUCTION

L'objectif de ce chapitre est de proposer, au travers du fonctionnement du système d'aide à la conception de Remora au niveau de la phase conceptuelle, les caractéristiques de tout système d'aide à la conception correspondant aux options énoncées au chapitre précédent.

Nous serons ainsi amenés à dire quelles fonctions, autres que l'aide à la conception telle que nous l'avons décrite au chapitre 3, doit assurer un tel système et à en donner une définition générale.

Nous citerons enfin les outils associés dans Remora à ces fonctions et décriront ceux que nous avons réalisés pour nous permettre de montrer, dans un dernier point, comment se déroule actuellement la phase de conception sur notre prototype expérimental.

IV - 1 - L'ENSEMBLE DES FONCTIONS ASSUREES PAR UN SYSTEME DE CONCEPTION TEL QUE REMORA

Si l'on a en mémoire l'objectif défini pour la phase de conception, soit parvenir à une description conceptuelle qui soit une image correcte de la réalité prise en compte, on songe immédiatement à deux types de fonctions que doit assumer un système d'aide à la conception :

A : Les aides à la conception que nous avons développées au chapitre précédent.

B : La fonction de construction du schéma conceptuel (soit la base des objets élémentaires dans REMORA) qui recouvre les deux tâches :

. de reconnaissance des composants du schéma conceptuel et de valorisation de leurs caractéristiques

. d'insertion au cours du temps de nouveaux éléments et de mise à jour des caractéristiques et des relations.

Ces deux tâches ont été décrites au chapitre II.

L'objectif de la phase de conception est aussi d'effectuer le passage de la définition externe d'un problème à sa définition conceptuelle (se reporter à l'introduction p. 3).

Cette transformation correspond, à notre sens, à une fonction du système d'aide à la conception que nous appelons fonction de transformation des définitions.

Elle correspond, dans Remora, à l'interprétation des descripteurs et leur traduction sous forme de modules et de lots stockés dans la bibliothèque des définitions.

Comme nous l'avons montré au chapitre précédent dans l'étude de la typologie des méthodes de conception, tout système d'aide à la conception doit assurer une fonction de guidage et de coordination que nous appelons pilotage.

Cette fonction est différente suivant la nature de la méthode d'aide à la conception envisagée :

. si la solution retenue est entièrement manuelle, il s'agit alors d'un guide manuel indiquant au concepteur les actions à entreprendre au cours de la phase de conception ;

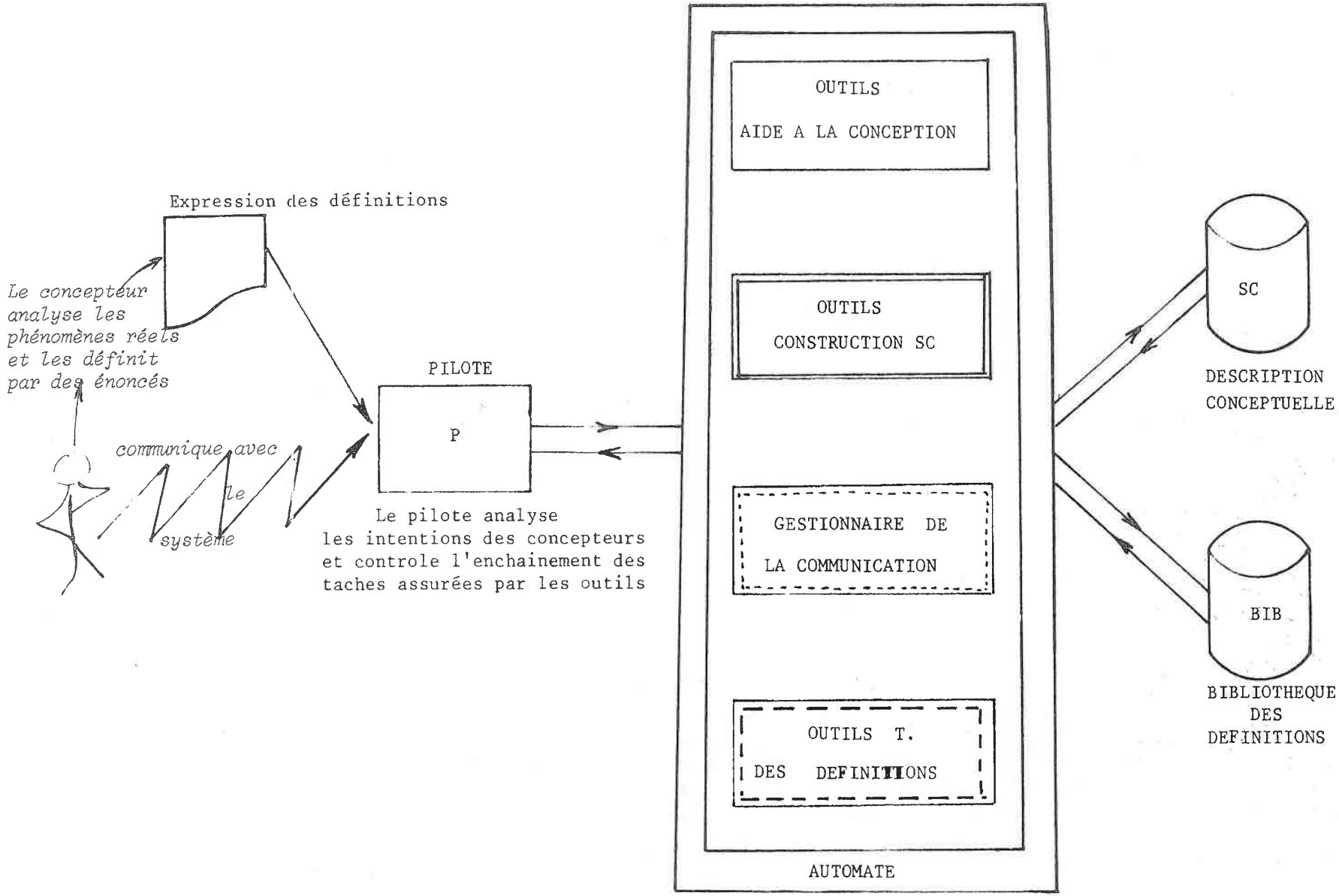
. si la solution est entièrement automatisée, le pilotage est totalement transparent à l'utilisateur, puisque tous les choix sont faits automatiquement.

. S'il s'agit d'une solution de conception assistée, semblable à celle que nous avons retenue dans Remora, une partie du pilotage sera automatisée et une autre par concertation avec le concepteur.

Dans la mesure ou deux interlocuteurs sont mis en présence l'un de l'autre, il est nécessaire de mettre en place un système de communication.

Les communications homme⟷système sont assurées par une fonction que nous appelons <u>communication</u>.

Si l'on fait l'hypothèse que toutes ces fonctions peuvent être assurées, partiellement ou intégralement par des outils (que nous chercherons à préciser au paragraphe suivant) on aboutit au schéma standard de fonctionnement d'un système d'aide à la conception commenté sur la figure 1.



- figure 1 - Outils d'un système de conception assistée

IV - 2 - ANALYSE DES OUTILS DE LA PHASE DE CONCEPTION DE REMORA

Le schéma représente le fonctionnement actuel du système d'aide à la conception de Remora (au cours de la phase de conception).

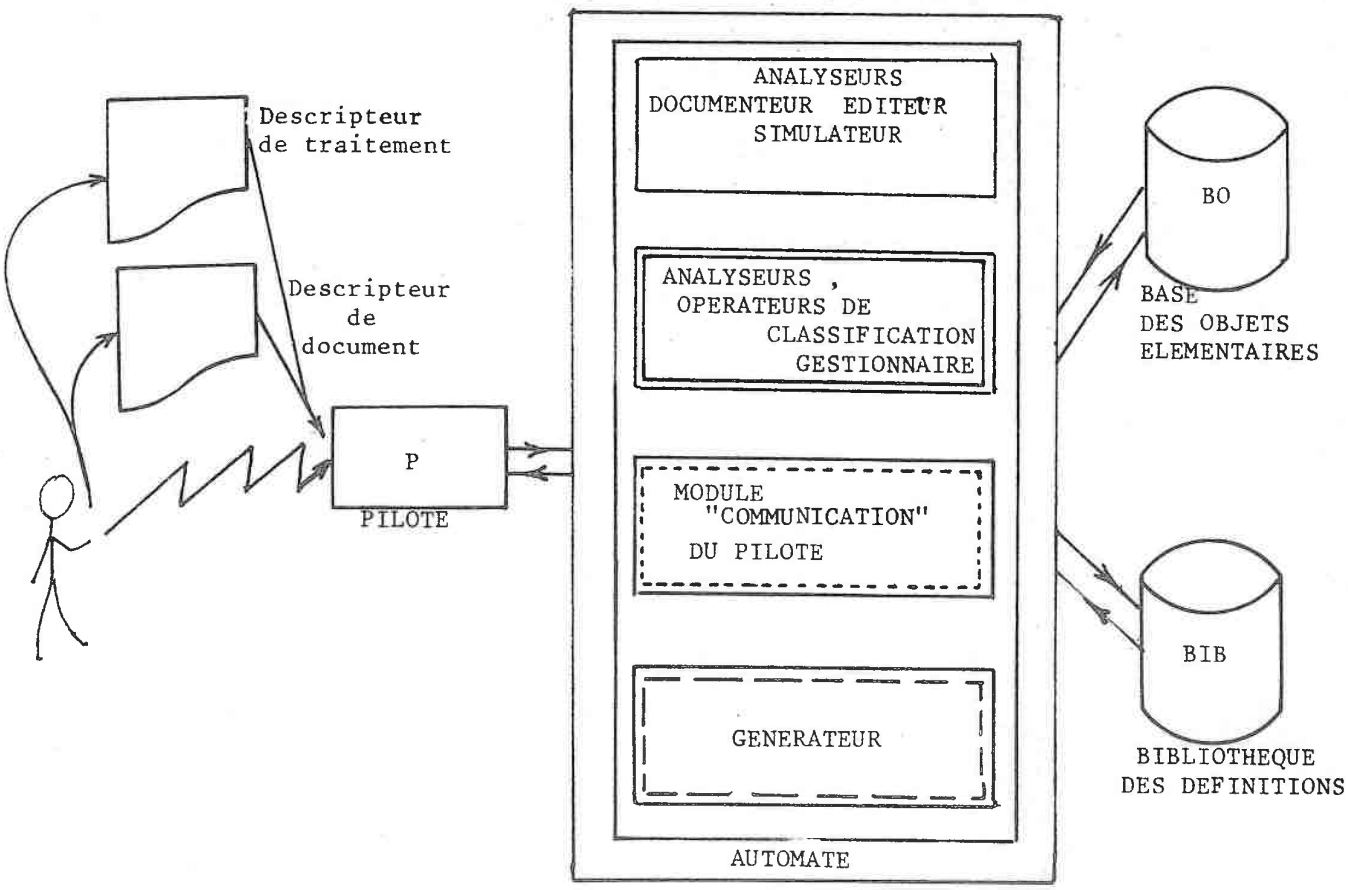


figure 2 - Schéma de fonctionnement du prototype Remora -

On remarque qu'il n'y a pas de correspondance biunivoque entre les fonctions précédemment définies et les outils choisis : un même outil peut assurer partiellement plusieurs fonctions et inversement.

Nous nous limiterons dans ce paragraphe à une analyse rapide des outils qui sortent du cadre de notre travail mais définirons en détail les algorithmes de ceux que nous avons réalisés : analyseurs - documenteur - gestionnaire.

IV - 2-1 - Le générateur :

A : Fonction : transformation des définitions

Le générateur est chargé d'une part de transformer les textes exprimés sur les descripteurs de traitements dans la syntaxe du langage de conception qui utilise le mode déclaratif traditionnel en textes intermédiaires où les déclarations ont été ordonnées.

Il est chargé, d'autre part, de transformer la définition des documents à éditer fournis par le concepteur sur les descripteurs de document sous forme d'une définition conceptuelle des lots.

B : Algorithme :

Le générateur s'appliquant au descripteur de traitement procède en plusieurs étapes explicitées par LAMY [36] ; le résultat de la première correspond à l'état conceptuel des définitions, stocké dans la Bibliothèque des définitions.

Le générateur s'appliquant aux descripteurs de documents a été défini par ORY [50] .

IV - 2-2 - Le simulateur d'édition :

A : Fonction : aide à la conception (contrôle)

Le simulateur aide le concepteur à vérifier que le document qu'il a défini est conforme à ce qu'il voulait obtenir.

B : Algorithme :

Il correspond à l'édition d'une maquette simulant le document de sortie décrit. (voir [50]).

IV - 2-3 - Le Documenteur :

A : Fonction : aide à la conception (documentation)

Le documenteur chargé de répondre aux interrogations du système et du concepteur sur le contenu de la BO.

B : Algorithme :

Compte tenu du choix que nous avons fait de gérer, sur le prototype, la base des objets élémentaires par le SGBD SOCRATE, le documenteur est un ensemble de macro-instructions (écrites dans le langage de requête SOCRATE) prises en charge par SOCRATE. Un exemple d'utilisation en est donné au paragraphe IV - 3.

IV - 2-4 - L'éditeur

A : Fonction : aide à la conception (documentation)

C'est l'outil chargé de toutes les éditions.

B : Algorithme :

Il est en partie celui que nous avons cité en IV - 2-1,B et est décrit dans la thèse de ORY [50] .

IV - 2-5 - Les opérateurs de classification

A : Fonction : construction BO (reconnaissance)

Ils sont chargés de la ventilation de l'ensemble des mots d'une application dans l'une des quatre classes : données - résultats - résultats intermédiaires - variables.

B : Algorithme :

A l'aide des algorithmes de calcul décrits au chapitre II, ils évaluent la nature des mots vis à vis de la procédure (ou nature-p), puis vis à vis de l'application (ou nature-a).

IV - 2-6 - Le pilote

A : Fonction : pilotage

Le pilote joue le rôle de chef d'orchestre ordonnant les tâches les unes par rapport aux autres au cours de la phase de conception.

Ces tâches sont soit statiques, soit dynamiques.

Nous désignons par tâches statiques celles dont l'enchaînement peut être prévu à l'avance ; par exemple à l'entrée d'une définition, il faut l'analyser, reconnaître les objets, faire des contrôles, corriger les erreurs, stocker les informations correctes ... etc ...

Les tâches dynamiques sont celles dont l'ordonnancement est imprévisible à l'avance et qu'il faut gérer en "temps réel" ; par exemple lorsque le concepteur demande des renseignements dans un ordre libre ou intervient pour imposer des corrections à un moment qu'il a choisi selon ses critères d'action propres.

B : Algorithme :

Il n'est pas à l'heure actuelle complètement défini et sera présenté dans la thèse de JANKOWSKI [34] .

On verra au paragraphe IV-3 comment est réalisé sur le prototype, l'enchaînement statique des tâches.

IV - 2-7 - Le gestionnaire de base :

A : Fonction : construction BO

Le gestionnaire de base est chargé des créations et mises à jour de la base des objets élémentaires.

B : Algorithme :

Comme nous l'avons rappelé précédemment, nous avons choisi de gérer la BO de Remora, par le SGBD de SOCRATE ([63] , [64]).

Aussi le gestionnaire comme le documenteur est réalisé pour un ensemble de macro-instructions (écrites en langage de requête SOCRATE) et interprétées par le SGBD SOCRATE.

Notre seul travail a donc consisté à construire une bibliothèque de macro-instructions auxquelles le gestionnaire de base, écrit en Cobol, fait appel : à chaque macro correspond dans le gestionnaire un module Cobol qui valorise les différents paramètres nécessaires à son exécution.

Cette façon de construire le gestionnaire à pour avantage de rendre facile l'adjonction de toute manipulation nouvelle de la base : elle sera traduite en effet par une macro-instruction et un appel supplémentaires.

IV - 2-8 - Les Analyseurs :

A : Fonction :
- reconnaissance des éléments de la BO
- contrôle
- intégration

A1 : Reconnaissance des éléments de la BO :

Ce sont les analyseurs qui, par analyse des textes des descripteurs vont reconnaître des types-objets et relations du schéma conceptuel et valoriser leurs caractéristiques, pour permettre au gestionnaire de créer la base des objets élémentaires.

A2 : Contrôle :

Les Analyseurs sont chargés de la fonction CONTROLE, telle qu'elle a été définie au chapitre III. Cependant, on pourra le constater par l'étude de l'algorithme , l'ensemble des contrôles prévus n'est pas encore totalement opérationnel.

A3 : Intégration :

Les analyseurs assurent avec certaines restrictions la fonction INTEGRATION définie au chapitre III.

B : Algorithme :

B1 : Reconnaissance des éléments de la BO :

Nous ne reviendrons pas sur les algorithmes de dépouillement décrits en II - 1-3-1, II - 1-4-2, de calculs décrits en II - 1-3-2-1, II - 1-3-2-2, II - 1-3-2-3 (uniquement contexte module et lot) qui permettent la reconnaissance des composants de la BO.

B2 : Contrôle :

B2.1 : Contrôles syntaxiques :

L'algorithme d'analyse syntaxique s'applique au contenu du descripteur de traitement dont nous rappelons le format sur la figure 3 :

Identificateur de descripteur : (1) Titre : (2) Procédure : (3) Application : (4)

				Désignation en clair	Condition		Repro		occurence ou indicatif	Calcul		Donnée	Identi.	Appel
(5)	(6)	(7)	(8)	(9)	(10)		(11)	(12)	(13)	(14)		(15)	(16)	(17)

- figure 3 - Descripteur de traitements

Prenons les conventions de notations suivantes :

. T1 est l'ensemble des formes correctes que peut prendre la zone occurrence.

Soient :
- (1) symbole
- (2) indice
- (3) indice : borne inférieure
- (4) indice : borne inférieure, borne supérieure
- (5) indice : borne inférieure, borne supérieure, pas

. T2 est l'ensemble des opérateurs que l'on peut trouver dans la zone calcul. Soient : {+, -, *, /, (,), **}.

. T3 est l'ensemble des opérateurs exprimant une condition.
Soient : { >, <, >=, <=, ≠, =; ET, OU, NON }.

. ER1 à ER10 sont des booléens initialisés à FAUX.

. SYMB une désignation d'élément quelconque.

. ØP un opérateur quelconque.

. U a la signification usuelle de "indéfini" ou "blanc".

L'analyse syntaxique du descripteur de traitement correspond à l'exécution séquentielle, zone par zone des actions suivantes :

ACTIONS	COMMENTAIRES
1 ((1) ou (2) ou (3) ou (4) = U) $\Longrightarrow$ ERR1 = VRAI	"Un élément de la première carte n'est pas déclaré".
2 (((5) ¬ = (1)) ou ((11) ¬ = (5) ‖ (6)) ou ((12) ¬ > (7))) $\Longrightarrow$ ERR2 = VRAI	"Mauvaise séquence de cartes"
3 ((8) ¬ > 00 ou ¬ < 100) $\Longrightarrow$ ERR3 = VRAI	"Nombre niveau non numérique"
4 ((17) ¬ = U et (14) ¬ = U) $\Longrightarrow$ ERR4 = VRAI	"Zone calcul non vide"
5 (SYMB $\in$ (16) et $\in$ (14)) $\Longrightarrow$ ERR5 = VRAI	"Variable non déclarée"
6 (ℓ(SYMB) > 8 ou (1° caractère (SYMB) > 0 et < 10)) $\Longrightarrow$ ERR6 = VRAI	"Identificateur incorrect"
7 ((OP $\in$ (14) et $\notin$ T2) ou (ØP $\in$ (10) et $\notin$ (T2 $\cup$ T3))) $\Longrightarrow$ ERR7 = VRAI	"Opérateur incorrect"
8 (((15) et (16)) ¬ = U et ((15) ¬ = (16))) $\Longrightarrow$ ERR8 = VRAI	"Mauvaise déclaration de données"
9 (((15) ou (16)) ¬ = U et (9) = U) $\Longrightarrow$ ERR9 = VRAI	"Définition en clair à blanc"
10 (((13) $\notin$ T1) ou ((13) $\in$ $T1_{2,3,4}$ et (10) = U)) $\Longrightarrow$ ERR10 = VRAI	"Zone occurrence incomplète et condition à blanc"

Remarque : Un exemple de contrôles syntaxiques est donné en IV-3. On peut trouver en [61] la liste complète des libellés d'erreur.

B2.2 : Contrôles sémantiques

Les Analyseurs effectuent un contrôle sur la correction du sens des mots en vérifiant (selon ce que nous avons défini au paragraphe III - 2-1-2 B) la compatibilité entre les différentes valeurs successives prises par le mode d'utilisation d'un même mot dans un même module (ou $\mathcal{M}_{mod}(m)$).

Cette vérification correspond au déroulement de l'algorithme :

$$(("D" \in \mathcal{M}_{mod}(m) \text{ et } "RI" \in \mathcal{M}_{mod}(m)) \text{ ou } ("D" \in \mathcal{M}_{mod}(m) \text{ et } "V" \in \mathcal{M}_{mod}(m))$$

$$\text{ou } ("V" \in \mathcal{M}_{mod}(m) \text{ et } "RI" \in \mathcal{M}_{mod}(m))) \Longrightarrow \text{COMPATIBLE} = \text{FAUX}$$

B3 : Intégration :

Parmi l'ensemble des tâches de contrôle définies au chapitre III dans le cadre de la fonction Intégration nous développons ici les tâches associées aux contrôles de polysème sur les mots et sur les modules.

B3.1 : Le module :

Hypothèse :

La détection des polysèmes dans la classe de vocabulaire des modules suit un algorithme reposant sur le type du module ; rappelons qu'il peut être maître ("MA") ou esclave ("AP").

Description de l'algorithme :

Soit TYPE-D le type qui a été assigné à l'objet "module" dans le contexte local d'un module.

Soit TYPE-B la caractéristique de l'objet de même désignation dans la BO.

L'algorithme de détection de polysèmes est : (les cas 1, 2, 3 s'excluant).

. Cas 1 :

TYPE-D = "AP" et TYPE-B = "AP" $\Longrightarrow$ COMPATIBLE = VRAI

. Cas 2 :

La paire (TYPE-D, TYPE-B) = ("MA", "AP") $\Longrightarrow$ COMPATIBLE = VRAI

. Cas 3 :

TYPE-D = "MA" et (TYPE-B = "MA" ou (TYPE-B = "AP" et

DEFINITION $\neg$ = U) $\Longrightarrow$ COMPTATIBLE = FAUX.
Base

B3.2 : Le mot :

Hypothèse :

La détection des polysèmes dans la classe de vocabulaire des mots repose sur trois critères :

- le genre du mot (cf. II - 1-3-2-1)
- le contexte du mot (cf. II - 1-3-2-2)
- les modes d'utilisation du mot, sa nature-p et sa nature-a (cf. II - 1-3-2-3).

La recherche de polysème s'effectue en deux étapes correspondant aux niveaux d'évaluation de la nature-p et de la nature-a d'un mot que nous avons introduits au chapitre 2.

Description de l'algorithme :

Comme nous l'avions fait dans ce chapitre nous présentons l'algorithme dans une description modulaire s'appuyant sur des expressions prédicatives.

Niveau 1

1 - Détermination des polysèmes dans le contexte de la procédure.
2 - Détermination des polysèmes dans le contexte de l'application.

Niveau 2

1 - Contexte de la procédure :

Soit $MU_p(m)$ le mode d'utilisation principal d'un mot (voir chapitre II) sélectionné parmi tous ses modes stockés dans la base, et MU(m) le mot d'utilisation du même mot dansun descripteur de traitement ou de document :

pour chaque mot m d'un descripteur entrant :

. la paire ($\{MU(m), MU_p(m)\} = \{RI, RI\}$) $\Longrightarrow$

CONTEXTE-TYPE (CONTYPE) et $\begin{cases} \text{CONTYPE} \Longrightarrow \text{COMPATIBLE} = \text{FAUX} \\ \text{CONTYPE} \Longrightarrow \text{COMPATIBLE} = \text{VRAI} \end{cases}$

. la paire ($\{MU(m), MU_p(m)\} = \{RI, D\}$ ou $\{RI, V\}$ ou $\{D, V\}$) COMPATIBLE = FAUX

. la paire ($\{MU(m), MU_p(m)\} = \{D, D\}$ ou $\{V, V\}$) $\Longrightarrow$ COMPATIBLE = VRAI

. C $\Longrightarrow$ COMPATIBLE = VRAI

Niveau 2

2 - Contexte de l'application :

Soit $NAT_p(m)$ la nature principale d'un mot vis à vis des procédures, sélectionnée parmi ses nature-p stockées dans la base (voir évaluation de nature-a du chapitre II) et NAT(m) la nature du même mot évaluée à partir de ses divers modes d'utilisation dans la procédure en cours :
pour chaque mot m de la procédure enc cours :

. la paire $(\{NAT(m), NAT_p(m)\} = \{RI, RI\}$ ou $\{D, D\}$ ou $\{V, V\})$

$\Longrightarrow$ COMPATIBLE = FAUX

. la paire $(\{NAT(m), NAT_p(m)\} = \{RI, V\}$ ou $\{R, V\})$

$\Longrightarrow$ COMPATIBLE = FAUX

. la paire $(\{NAT(m), NAT_p(m)\} = \{RI, R\}$ ou $\{R, R\})$

$\Longrightarrow$ COMPATIBLE = FAUX

. la paire $(\{NAT(m), NAT_p(m)\} = \{RI, D\}$ ou $\{R, D\})$

MOD-UTIL(MOD-UT) et $\begin{cases} \text{MOD-UT} \Longrightarrow \text{COMPATIBLE} = \text{FAUX} \\ \neg\text{MOD-UT} \Longrightarrow \text{COMPATIBLE} = \text{VRAI} \end{cases}$

. la paire $(\{NAT(m), NAT_p(m)\} = \{V, D\}$

MOD-UTIL (MOD-UT) et $\begin{cases} \text{MOD-UT} \Longrightarrow \text{COMPATIBLE} = \text{FAUX} \\ \neg\text{MOD-UT} \Longrightarrow \text{COMPATIBLE} = \text{VRAI} \end{cases}$

Niveau 3

Analyse de CONTEXTE-TYPE :

Soit m appartient au moins à deux modules (ou deux lots) soit M1 le premier, M2 le deuxième, R relation "appelle" définie au chapitre I.

. $\neg$ (M1 R M2 ou M2 R M1) $\Longrightarrow$ CONTYPE = FAUX

. M1 R M2 ou M2 R M1

$$\begin{cases} \text{. genre de m = I} \Longrightarrow \text{CONTYPE = VRAI} \\ \text{. genre de m} \neg \text{ = I} \Longrightarrow \text{CONTYPE = FAUX} \end{cases}$$

Niveau 3 :

Analyse de MOD-UTIL :

Soit $\mathcal{M}$(m) l'ensemble des modes d'utilisation de m :

. D $\in \mathcal{M}$(m) $\Longrightarrow$ MOD-UT = VRAI

. D $\in \mathcal{M}$(m) $\Longrightarrow$ MOD-UT = FAUX

IV - 3 - MISE EN OEUVRE DES OUTILS DANS LE DEROULEMENT DE LA PHASE DE CONCEPTION

IV -3 -0 - Introduction

Dans ce paragraphe nous montrerons sur un exemple comment la mise en oeuvre des outils orchestrés par le pilote permet la prise en charge au niveau conceptuel d'une nouvelle définition effectuée par un concepteur.

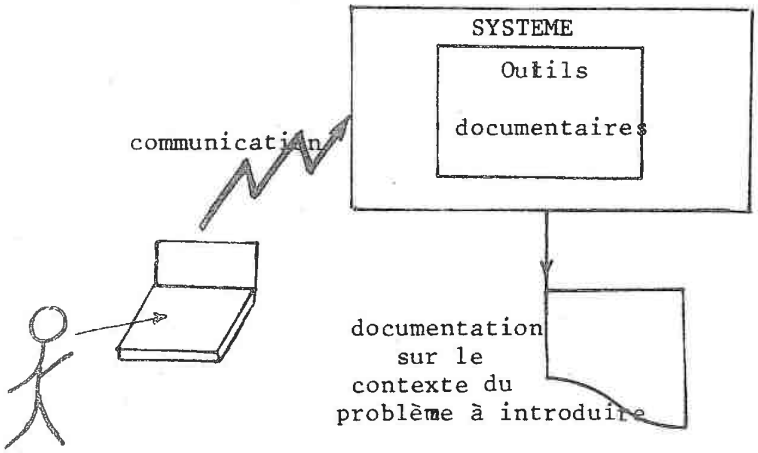
Nous qualifierons cette mise en oeuvre d'"externe" car elle correspond au point de vue du concepteur par opposition à la mise en oeuvre interne qui correspond au système.

Nous nous proposons d'étudier en priorité l'aspect "externe" en montrant l'enchaînement des étapes qui permettent, à partir d'une définition externe fournie par le concepteur, d'aboutir à une définition conceptuelle qui semble cohérente et qui soit correctement intégrée dans l'ensemble des définitions conceptuelles déjà acceptées et de nous limiter à fournir quelques aspects du déroulement interne associé ; pour un développement plus complet de cet aspect on peut se reporter à [60] (rapport explicitant en termes techniques le déroulement de la phase de conception), et à [61] (rapport montrant, dans le cadre d'un exemple complet, comment est réalisé l'aspect technique "interne" du pilotage).

IV -3 -1 - Survol de la mise en oeuvre :

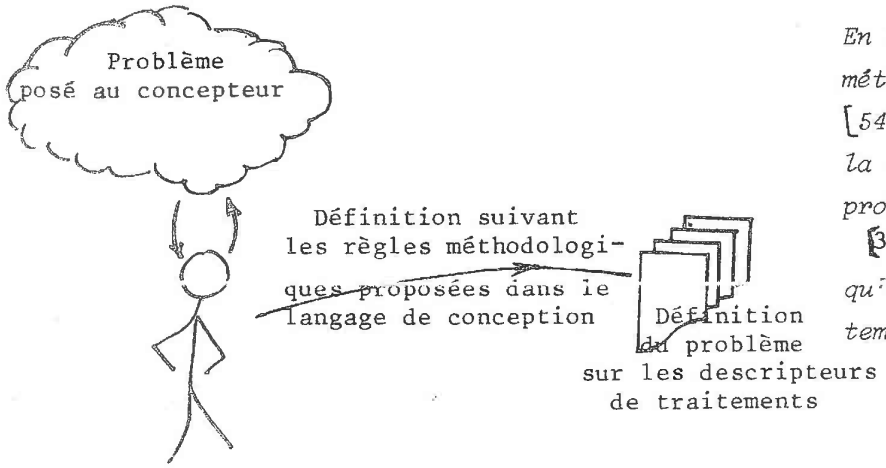
Le déroulement de la phase de conception dans "l'optique concepteur" (depuis la description d'une définition, jusqu'à l'insertion des nouveaux composants dans la BO) peut être schématisé en cinq étapes :

1ère étape : "connaissance du contexte"



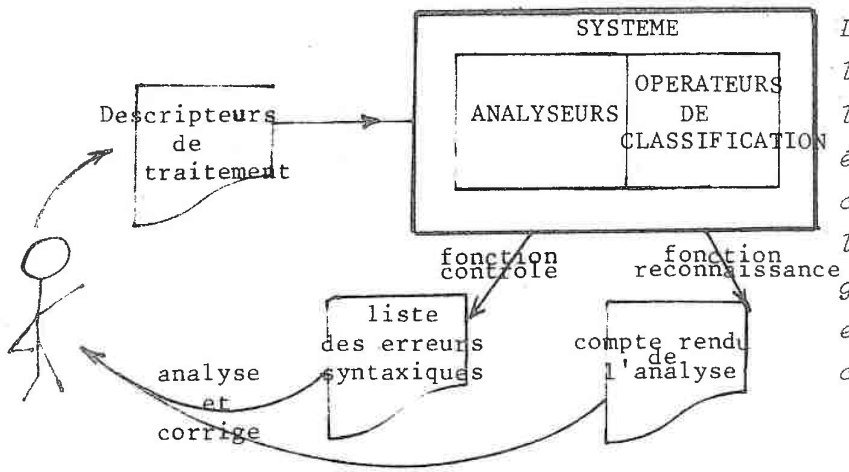
Pour bien intégrer sa définition dans le contexte de celles qui ont déjà été fournies, le concepteur peut acquérir une bonne connaissance de ce contexte en faisant appel aux outils documentaires que lui propose le système.

2ème étape : "Définition du problème"



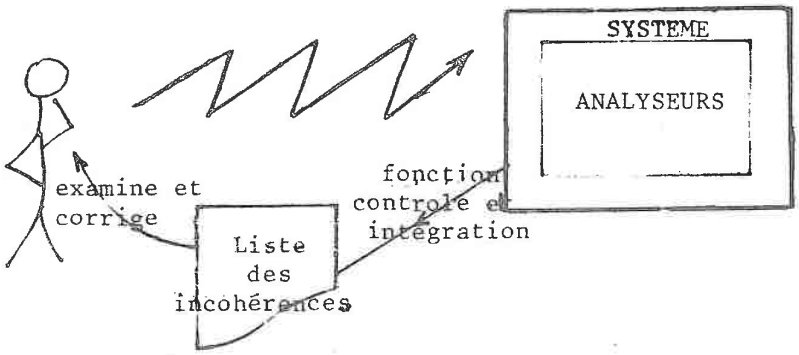
En s'appuyant sur les règles méthodologiques développées en [54], le concepteur fournit la description externe de son problème dans le langage Remora [36], sur le support associé qu'est le descripteur de traitement.

3ème étape : "Prise en charge des descripteurs par le système"



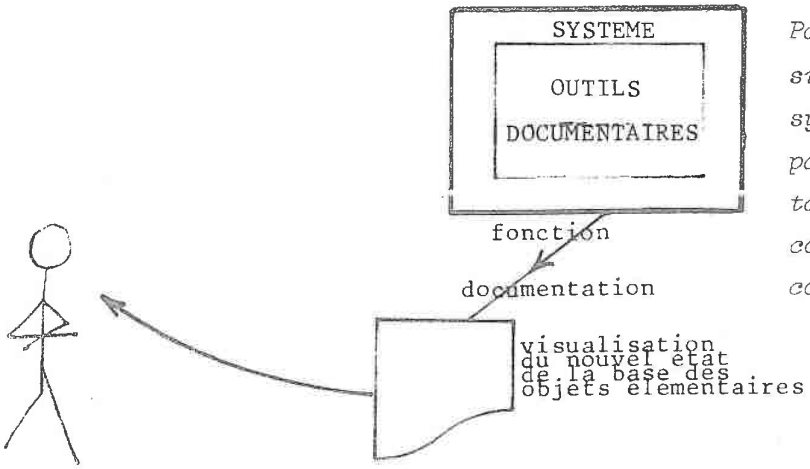
Les analyseurs vont vérifier la syntaxe du langage et informer le concepteur de ses érreurs éventuelles ; les opérateurs de classification s'assurent que tous les objets manipulés dans le langage ont bien les types requis et proposent au concepteur un compte rendu de leur analyse.

4ème étape : "Détection et correction des incohérences"



Le système effectue des contrôles de cohérence et tente l'intégration de la définition ; le concepteur utilise les renseignements fournis par le système pour apporter des corrections.

5ème étape : "Visualisation du nouvel état conceptuel"



Pour se donner une dernière possibilité de contrôle manuel le système propose au concepteur, par le biais des outils documentaires, la vision du nouvel état conceptuel en schématisant le contenu de la BO.

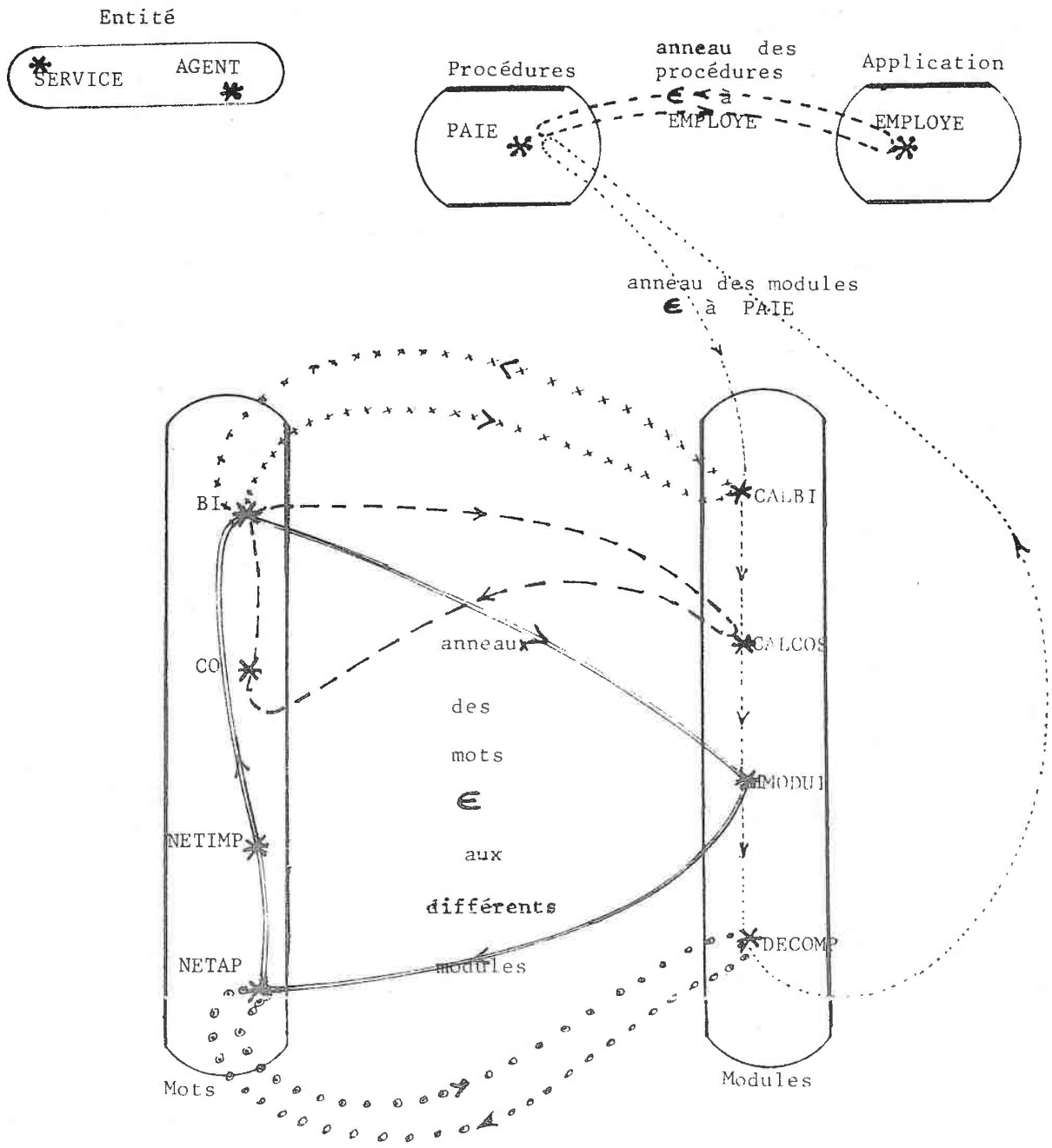
IV - 3 -2 - Présentation de l'exemple :

L'exemple que nous allons présenter est un exemple d'"école" destiné à simuler le déroulement réel de la phase de conception ; en particulier il a été conçu de façon à permettre des contrôles de cohérence significatifs.

Supposons que nous soyons dans la situation où un concepteur a déjà écrit une procédure PAIE dans le cadre d'une application EMPLOYE. Le contexte est synthétisé par l'état correspondant de la BO proposé sur la figure 5.

Supposons par ailleurs qu'un concepteur nouveau s'interesse à des statistiques sur les employés.

Il voudrait en particulier pouvoir faire la comparaison entre les cumuls des salaires bruts imposables et des salaires nets imposables versés par l'entreprise au cours de la dernière paie (ces statistiques ne l'interessant,pour le brut imposable,que pour les agents dont le matricule est superieur à 100);il désire connaitre d'autre part le cumul des cotisations ouvrières et patronales.



- Figure 5 - Etat initial de la base des objets élémentaires

IV - 3-3 - Déroulement des tâches d'aide à la conception sur l'exemple :

IV - 3-3-1 - Première étape : "Connaissance du contexte" :

Pour connaître l'environnement du problème qu'il a à décrire, il est logique de penser que le concepteur cherche à se documenter sur l'état conceptuel existant, par le biais de la connaissance de la base des objets élémentaires.

Utilisant l'approche modulaire de l'organisation qui lui a été suggérée il est naturel de penser que le concepteur va s'informer du général au particulier en suivant les différents niveaux hiérarchiques définis sur les composants du schéma conceptuel par une séquence de questions du type suivant :

- quels sont les composants en procédures de l'application en cours ? (contexte le plus grand dont l'abord est immédiat).
- quelle est la composition des procédures en modules ?
- quels sont les mots des différents modules ?
- quelles sont les entités définies pour l'application ?

Ces questions sont traitées par l'outil documenteur associé à un système de communication conversationnel par lequel il accepte les questions et fournit les réponses qui, dans le cas de notre exemple, sont proposées sur les listings d'ordinateur des figures 6 et 7.

- Figure 6 -

Question :

```
POUR TOUTE PROCEDURE X2
  I(0) NOM2 DE X2
  I(10) EST RATTACHEE A
  ECRIRE
  POUR TOUT NO-APP
    I(30) NOM1 DE APPLI
    ECRIRE
  FIN
FIN ?
```

Réponse :

```
PAIE        EST RATTACHEE A
                                EMPLOYE
```

Question :

```
POUR TOUTE PROCEDURE X1
  I(0) NOM2 DE X1
  I(10) 'CONTIENT LES MODULES'
  ECRIRE
  POUR TOUT NO-MODU1 DE X1
    I(30) NOM4 DE MODU1
    ECRIRE
  FIN
FIN
```

Réponse :

```
PAIE        CONTIENT LES MODULES
                                MODU1
                                CALB1
                                CALCBS
                                DECOMP
```

Question : I TOUT ENTITEB ?

Réponse :

```
ENTITEB 1
   NOM6:SERVICE
   DESI61:...
   DESI62:...
   NO-MOT4:...
ENTITEB 2
   NOM6:MAT
   DESI61:...
   DESI62:...
   NO-MOT4:...
```

- Figure 7 -

Question :

```
POUR TOUT MODULE X1
  I(0) NOM4 DE X1
  I(10) 'COMPREND LES MOTS'
  ECRIRE
  POUR TOUT NO-MOT2 DE X1
    I(30) NOM5 DE MOT2
    ECRIRE
  FIN
FIN
```

Réponse :

```
MOD1    COMPREND LES MOTS
                    TOCHSO
                    NETAP
                    NETIMP
                    BI
                    COTISO
                    MLT
                    ACOMP
                    SAISIE
                    PAIE
                    LIG
CALE1   COMPREND LES MOTS
                    BI
                    ACT
                    DIS
                    TB
                    INDI
                    TAUX
                    PROP    CALCOS    COMPREND LES MOTS
                    COMP
                    DEMI
                    INDE
                    ENF
                    ID
                    ZONE
                    CAT
                    TIT
                    ALX
                    SF
                    BF
                    BAF
                    EP
                    BAV
                    TAU
                    I
                    TABF
                    IP
                    INDP
                    HSUP    DECOMP    COMPREND LES MOTS
                    NAH
                    HSNO
                    HSNUI
                    HSDIM
                    NHS
                    IR
                    TAB
                    INDEM
```

```
*
NBIN
CODE
MIND
NBIND
GREVE
NOGREVE
NETTB
NETIR
NETSF
NETIP
NTB
NIR
NSF
NIP

COTISO
CO
CAT
TIT
BAS
TO
TB
ALX
BI
PLAFS
REAL
DIRA
SF
PLAFIR
TMUT
MLT
BLI
I
J

BASE
NETAP
U
NU
UN
V
```

Il est probable qu'à partir de cette connaissance générale de la procédure PAIE, le concepteur cherche à en approfondir certains aspects.

Par exemple on peut imaginer que dans la liste des mots (figure 7), ceux qui sont désignés par NETIMP, BI, CO lui suggèrent des sens qu'il souhaite voir confirmés ou infirmés par l'examen de leurs caractéristiques.

Le système d'information par question/réponse fonctionne pour cet exemple comme le montre la figure 8 :

- figure 8 -

Question : *I UN MOTB AYANT NOMS = 'NETIMP' ?*

Réponse :

```
MOTB 3
    NOM5:NETIMP
    DESI51: NET IMPOSABLE
    DESI52:...
    PARTI:...
    NATURE:...
    IMAGE:...
    NO-STRUCT2:...
    NO-MODU2: MOT-MODULE 3
    NO-PROC4: PROC-MOT 3
    ENTITI1:...
```

Question : *I UN MOTB AYANT NOM5 = 'BI' ?*

Réponse :

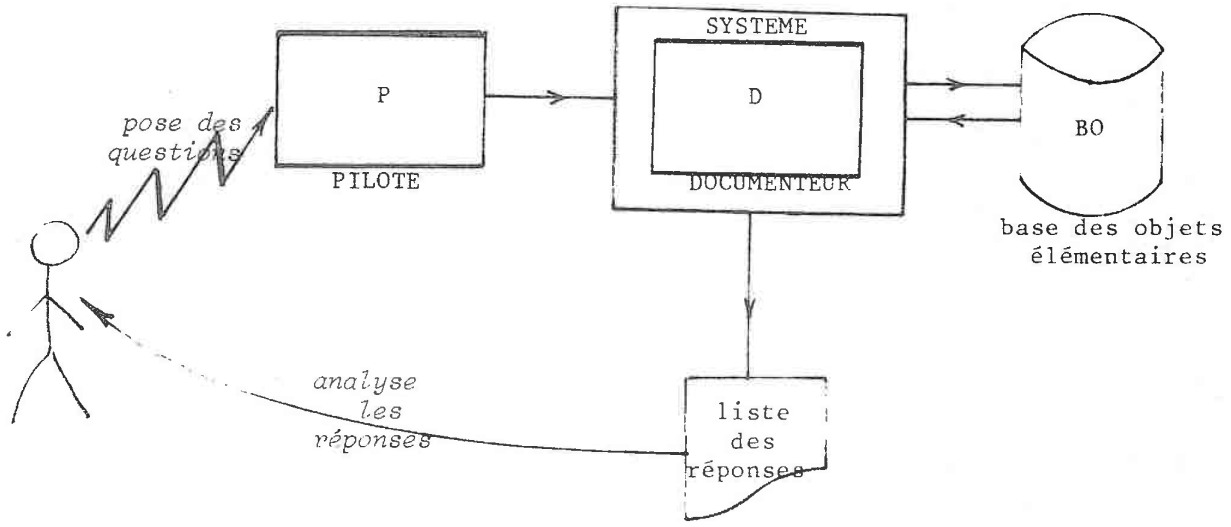
```
MOTB 4
    NOM5:BI
    DESI51: SALAIRE BRUT
    DESI52:...
    PARTI:...
    NATURE:...
    IMAGE:...
    NO-STRUCT2:...
    NO-MODU2: MOT-MODULE 4
              MOT-MODULE 11
              MOT-MODULE 65
              MOT-MODULE 87
    NO-PROC4: PROC-MOT 4
    ENTITI1:...
```

Question : *I UN MOTB AYANT NOM5 = 'CO'*

Réponse :

```
MOTB 60
    NOM5:CO
    DESI51: TABLEAU DES COTISATIONS
    DESI52:...
    PARTI:T
    NATURE:...
    IMAGE:...
    NO-STRUCT2:...
    NO-MODU2: MOT-MODULE 62
              MOT-MODULE 90
    NO-PROC4: PROC-MOT 60
    ENTITI1:...
```

QUELQUES ASPECTS DU DEROULEMENT INTERNE



Le Pilote gère les communications avec le concepteur et lance à partir de leur analyse l'outil de documentation qui intérroge la BO.

IV -3 -3-2 - Deuxième étape : définition du problème :

Conscient d'avoir acquis une connaissance suffisante du contexte du problème qu'il veut introduire, le concepteur peut alors procéder à sa définition.

Nous n'entrerons pas dans les détails de l'analyse à laquelle il procéde dont on trouvera en [54] l'aspect méthodologique largement développé.

Disons seulement, de manière schématique, que cette analyse lui permet de repérer trois résultats (au sens de "résultat final d'un calcul") ACOMP, TOT, COTISO qu'il définit par un raisonnement déductif décrit sur un descripteur de traitement : celui du module STATIS présenté sur la figure 9.

Identificateur de descripteur : STATIS 0 0 0 Titre : STATISTIQUES Procédure : PAIE Application : EMPLOYE

Repro 1-5	ligne 9-10	seq 11	12-13	Désignation en clair 14-54	Condition 55-79	80	Repro 1-10	11	occurence ou indicatif 12-31	Calcu 32-54	55	Donnée 56-63	Ident. 64-72	Appel 73-80
	01	1	01	CUMULS ENTREPRISE				2						
	05	1	02	CUMUL SALNET				2	MAT				ACOMP	
	10	1	03	SALAIRE NET				2		MAT > 100		NETIMP	+NETIMP	
	15	1	02	CUMUL SALBRUT				2	MAT				TOT	
	20	1	03	SALAIRE BRUT				2				BI	+BI	
	25	1	02	CUMUL COTISATIONS				2	I : 1,6				COTISC	
	30	1	03	TABLEAU DES COTISATIONS				2				***	CO(I)	

- figure 9 -

IV - 3 -3-3 - Troisième étape : Prise en charge des descripteurs de traitement par le système.

La prise en charge du descripteur par le système déclenche deux types d'actions fournissant en retour des informations au concepteur.

Ce sont :

1 - Le bilan de l'analyse syntaxique du document qu'il vient d'établir.

2 - La "photographie" des nouveaux objets introduits par son intervention et leurs caractéristiques.

A - Les contrôles syntaxiques :

Le bilan, dans le cas de l'exemple choisi, est présenté sur la figure 10. On y remarque que le principe d'édition choisi consiste à mettre en valeur (au moyen d'un soulignement par % % % %) les zones erronées, à l'endroit où elles figurent sur le descripteur et à fournir un message explicatif par type d'erreur.

L'analyse de ce bilan par le concepteur le conduit à ré-entrer un document corrigé (figure 11) qui subit le même traitement que le précédent.

Il y a donc itération de ce procédé jusqu'à épuisement des erreurs syntaxiques.

IDENTIFICATEUR : STATIS TITRE : STATISTIQUES APPLICATION : PAIE PROCEDURE : EMPLOYE

DESIGNATION EN CLAIR	CONDITION	OCCURENCE	CALCUL	DONNEE	IDENT	APPEL
01 CUMULS ENTREPRISE						
02 CUMUL SALNET		MAT				
03 SALAIRE NET			MAT > 100	NETIMP	ACOMP + NETIMP	
			%%%%			
02 CUMUL SALBRUT		MAT				
03 SALAIRE BRUT				BI	TOT + BI	
02 CUMUL COTISATIONS		I: 1,6			COTISO	
		%%%%				
03 TABLEAU DES COTISATIO				***	CO (I)	
				%%%%		

CAR INCORRECT EN COND OU CALCUL	STATIS 102	MAT > 100	NETIMP	+ NETIMP
ZONE COND A BLANC ET OCCUR INCOMPLETE	STATIS 252 I: 1,6			COTISO
CHIF. OU CAR. SPECIAL DS IDENT. PAS CONDITION OU CAL***				
MAUVAISE DECLARATION DE DONNEE.ZONE IGNOREE	STATIS 302		***	CO (I)

- figure 10 -

Identificateur de descripteur : STATIS 000 (1–8, 9–11) Titre : STATISTIQUES (12–54) Procédure : PAIE (55–62) Application : EMPLOYE (63–70)

Repro (1–8)	ligne (9–10)	seq (11)	(12–13)	Désignation en clair (14–54)	Condition (55–79)	(80)	Repro (1–10)	(11)	occurence ou indicatif (12–31)	Calcul (32–54)	(55)	Donnée (56–63)	Ident. (64–72)	Appel (73–80)
	01	1	01	CUMULS ENTREPRISE				2						
	05	1	02	CUMUL SALNET				2	MAT				ACOMP	
	10	1	03	SALAIRE NET	MAT > 100			2				NETIMP	+NETIMP	
	15	1	02	CUMUL SALBRUT				2	MAT				TOT	
	20	1	03	SALAIRE BRUT				2				BI	+BI	
	25	1	02	CUMUL COTISATIONS				2	I : 1, 6, 1				COTISO	
	30	1	03	TABLEAU DES COTISATIONS				2					CO(I)	

- figure 11 -

B - Analyse des objets nouveaux

Parallèlement le système communique au concepteur la "photographie" des nouveaux objets introduits par son intervention et leurs caractéristiques sous la forme d'un tableau du type de celui que nous présentons pour l'exemple choisi sur la figure 12 .

La réaction du concepteur devant ces informations peut être le refus et la correction d'une décision prise à tort par le système.

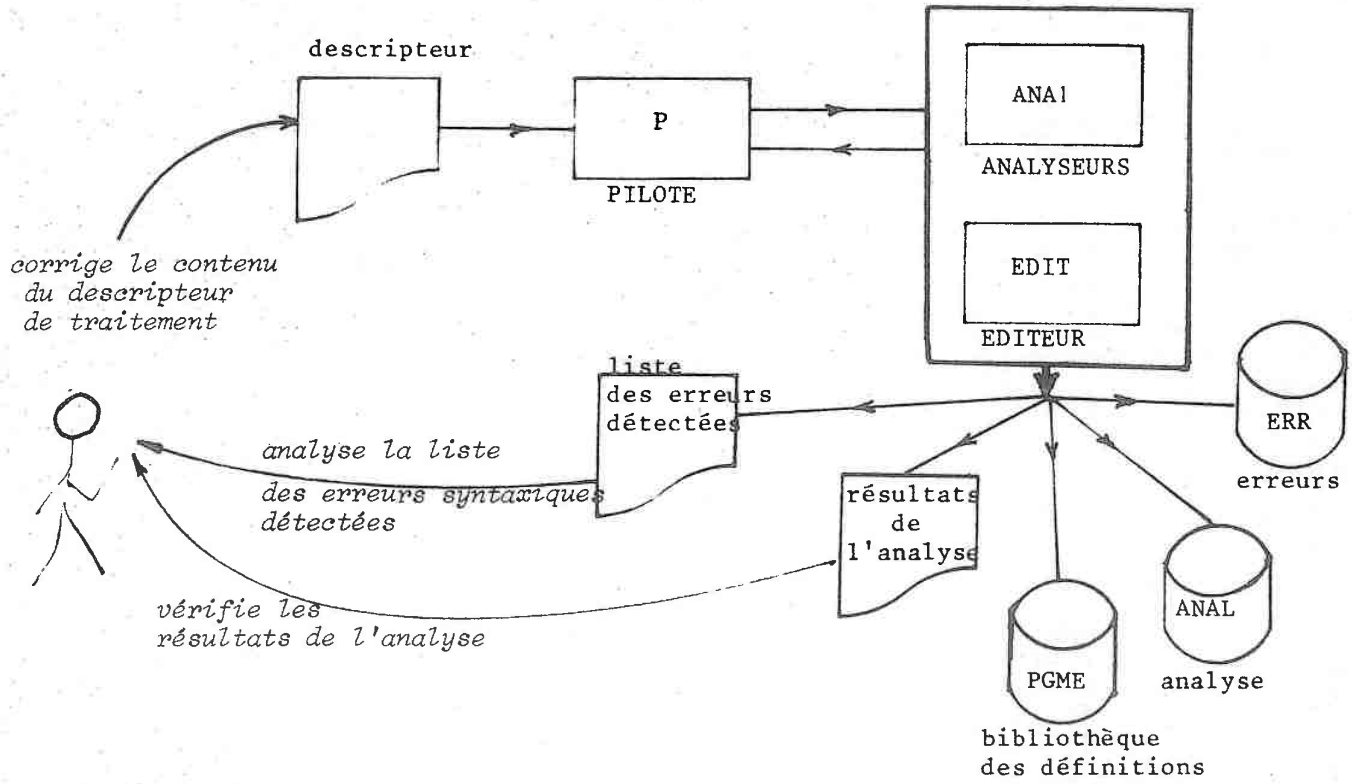
Résultat de l'analyse de STATIS

module STATIS décrit pour la procédure PAIE et l'application EMPLOYE

	APPLICATION EMPLOYE		PROCEDURE PAIE	MODULE STATIS
ont été détectés	IDENT	MODE UTILISATION	TYPE PARTICULIER	DESIGNATION EN CLAIR
1 mot	ACOMP	RI		CUMUL SALNET
1 entité	MAT		F *indicatif*	
	NETIMP	D		SALAIRE NET
	TOT	RI		CUMUL SALBRUT
6 mots	BI	D		SALAIRE BRUT
	COTISO	RI		CUMUL COTISATIONS
	I	RI	I *indice*	
	CO	RI	T *tableau*	TABLEAU DES COTISATIONS
	chaque mot	*a été affecté d'un mode d'utilisation*	*d'un type*	*et est associé à sa désignation en clair*

- figure 12 -

QUELQUES ASPECTS DU DEROULEMENT INTERNE



Le Pilote lance l'action des ANALYSEURS. Le premier module de cet outil :

- exécute les contrôles syntaxiques et stocke les informations sur les erreurs détectées dans un fichier temporaire (ERR).

- décide s'il faut ou non stocker la définition dans la bibliothèque (PGME).

- analyse le contenu du descripteur, reconnaît les objets, valorise leurs caractéristiques et les stocke dans un fichier intermédiaire (ANAL).

L'EDITEUR établit les documents (11 et 12) à partir des fichiers ERR et ANAL.

IV - 3 -3-4 - Quatrième étape : Détection et correction des incohérences

La recherche de la polysémie se traduit par la proposition au concepteur d'une liste d'objets dont l'analyse des caractéristiques a relevé des incohérences.

Ainsi la figure 13 correspondant à notre exemple montre que :

```
OACOMP   RID  CUMUL SALNET
                        ACOMPTE

STATIS

ONETIMP  D RI SALAIRE NET
                        NET IMPOS
ABLE
STATIS
```

figure 13 - Détection des incohérences -

- ACOMP dont l'usage a été défini D (module MODU1) pui RI (module STATIS),
- NETIMP dont l'usage a été défini RI (module MODU1) puis D(module STATIS),

ont une définition qui ne peut pas être déclarée cohérente.

Prenant connaissance de cette liste, le concepteur en conclut par exemple que :

- ACOMP est effectivement une désignation qu'il a utilisée à tort.
- L'usage de NETIMP est bien RI car c'est un résultat intermédiaire transmis de MODU1 vers STATIS.

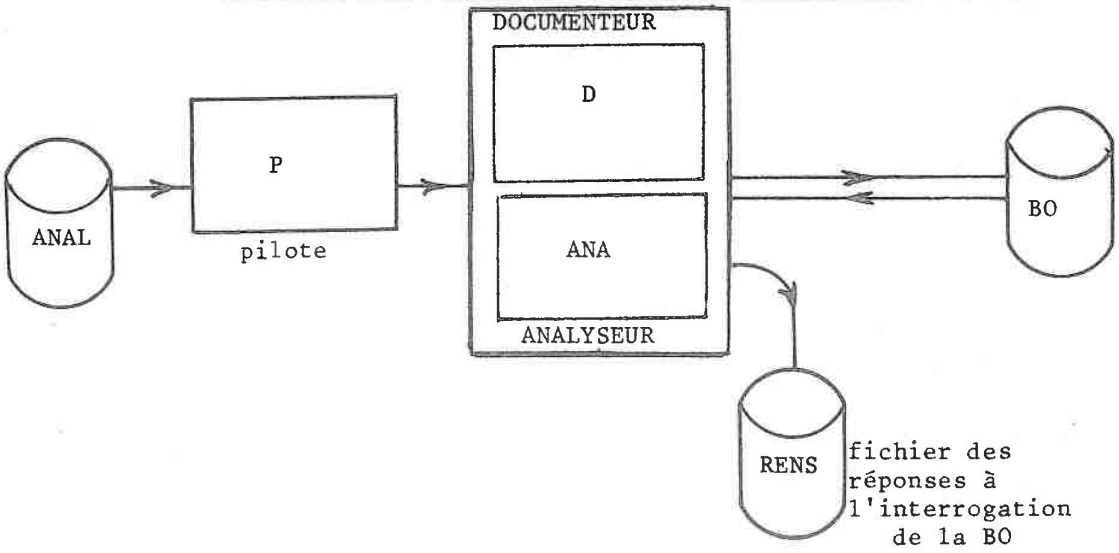
Remarquons dans l'état actuel de nos réalisations nous demandons au concepteur de corriger lui-même la liste des incohérences détectées :

- soit en changeant la désignation (c'est le cas ici de ACOMP qui sera remplacé par CUMSAL)
- soit en modifiant la caractéristique qui a été la cause de détection d'anomalie (dans cet exemple le mode d'utilisation de NETIMP sera "forcé" à RI).

QUELQUES ASPECTS DU DEROULEMENT INTERNE

La mise en oeuvre interne opérée par le système correspond à deux phases successives d'actions.

A - Phase : interrogation de la base des objets élémentaires :



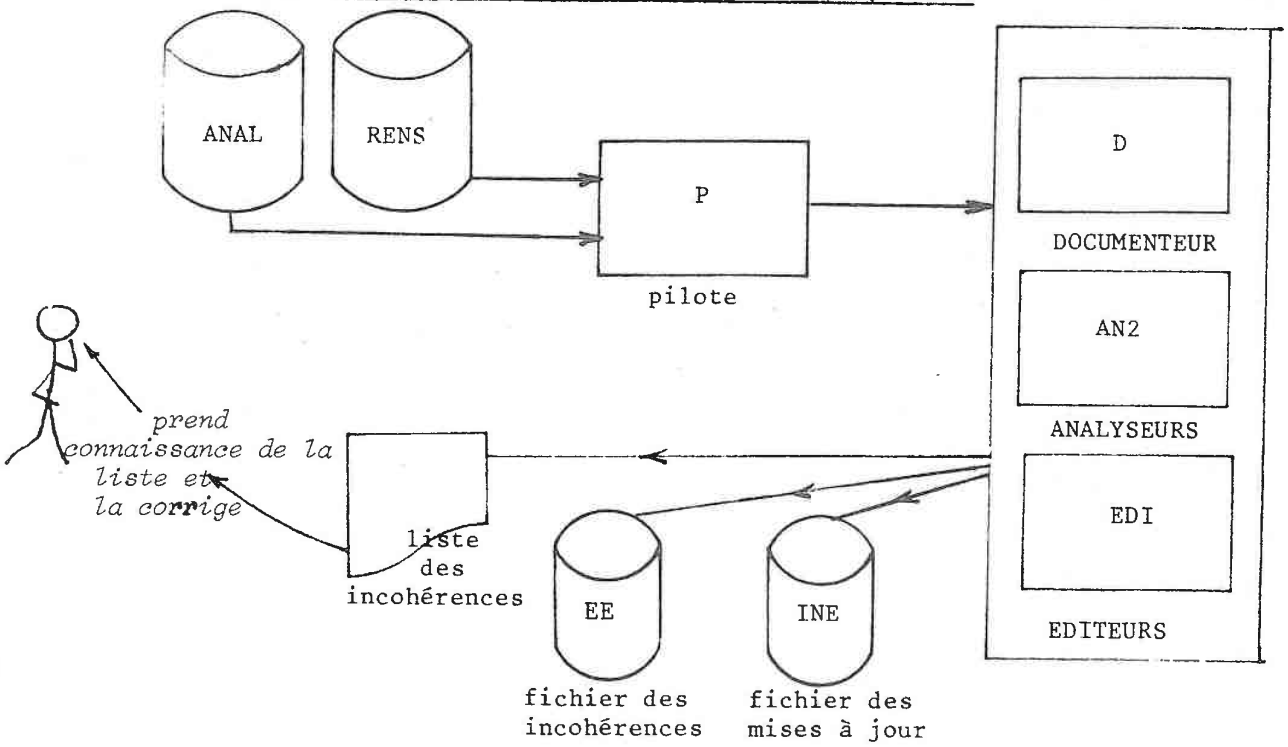
Le Pilote lance une action de documentation. Pour chaque élément du fichier ANAL, le DOCUMENTEUR vérifie s'il existe ou non dans la BO et stocke sa réponse dans un fichier temporaire RENS dont l'image, dans le cas de l'exemple traité est fournie par la figure 14.

	IDENT.	PRESENCE		IDENT.	PRESENCE		IDENT.	PRESENCE	DESIGNATION
APPLICATION:	EMPLOYE	OUI	PROCEDURE:	PAIE	OUI	MODULE:	STATIS	NON	

IDENT.	MODE DESC.	MODE BASE	DES. DESC.	DES. BASE	TYPE
ACOMP	RI	C	CUMUL SALNET	ACOMPTE	
MAT		F			F
NETIMP	D	RI	SALAIRE NET	NET IMPOSABLE	
TOT	RI		CUMUL SALBRUT		
BI	D		SALAIRE BRUT	BRUT IMPOSABLE	
COTISO	RI	RI	CUMUL COTISATIONS	COTISATIONS OUVRIER	
I	RI	RI		INDICE	I
CO	RI	RI	TABLEAU DES COTISATIONS	TABLEAU COTI	T

- figure 14 -

B - Phase : contrôles à partir de la polysèmie



A l'aide des fichiers ANAL et RENS, le deuxième module des Analyseurs effectue des contrôles de cohérence, stocke dans le fichier EE le résultat de son analyse et prépare, pour les éléments corrects, dans le fichier INE les informations nécessaires à leur prise en compte dans la BO. (figure 17).

IV -3 -3-5 - 5° étape : Visualisation du nouvel état conceptuel

La BO est l'image de la description du système d'informations à ce niveau de conception ; il semble naturel de demander au concepteur de projeter un dernier regard de contrôle sur image pour confirmer sa correction après l'insertion des dernières définitions.

Ainsi la figure 15 présente dans le cas de notre exemple le contenu de la BO, après l'insertion du module STATIS.

La réaction du concepteur pourra être ici comme à tout instant du déroulement de cette étape le refus de certains éléments et leur correction.

- figure 16 -

MODU1	EST RATTACHE A	PAIE	ET A POUR TYPE	MA
CALBI	EST RATTACHE A	PAIE	ET A POUR TYPE	AP
CALCOS	EST RATTACHE A	PAIE	ET A POUR TYPE	AP
DECOMP	EST RATTACHE A	PAIE	ET A POUR TYPE	AP
STATIS	EST RATTACHE A	PAIE	ET A POUR TYPE	MA
TOCHSO	EST RATTACHE A	MODU1	ET A POUR MODE	RI
NETAP	EST RATTACHE A	MODU1 DECOMP	ET A POUR MODE	RI
NETIMP	EST RATTACHE A	MODU1 STATIS	ET A POUR MODE	RI RI
BI	EST RATTACHE A	MODU1 CALBI CALCOS STATIS	ET A POUR MODE	D
COTISO	EST RATTACHE A	MODU1 CALCOS STATIS	ET A POUR MODE	RI RI RI
MLT	EST RATTACHE A	MODU1 CALCOS	ET A POUR MODE	RI RI
ACOMP	EST RATTACHE A	MODU1	ET A POUR MODE	D
SAISIE	EST RATTACHE A	MODU1	ET A POUR MODE	D
PAIE	EST RATTACHE A	MODU1	ET A POUR MODE	
LIQ	EST RATTACHE A	MODU1	ET A POUR MODE	
ACT	EST RATTACHE A	CALBI	ET A POUR MODE	
DIS	EST RATTACHE A	CALBI	ET A POUR MODE	
TB	EST RATTACHE A	CALBI CALCOS	ET A POUR MODE	RI
INDI	EST RATTACHE A	CALBI	ET A POUR MODE	
TAUX	EST RATTACHE A	CALBI	ET A POUR MODE	
PROP	EST RATTACHE A	CALBI	ET A POUR MODE	RI
COMP	EST RATTACHE A	CALBI	ET A POUR MODE	
DEMI	EST RATTACHE A	CALBI	ET A POUR MODE	
INDE	EST RATTACHE A	CALBI	ET A POUR MODE	
ENF	EST RATTACHE A	CALBI	ET A POUR MODE	
ID	EST RATTACHE A	CALBI	ET A POUR MODE	RI

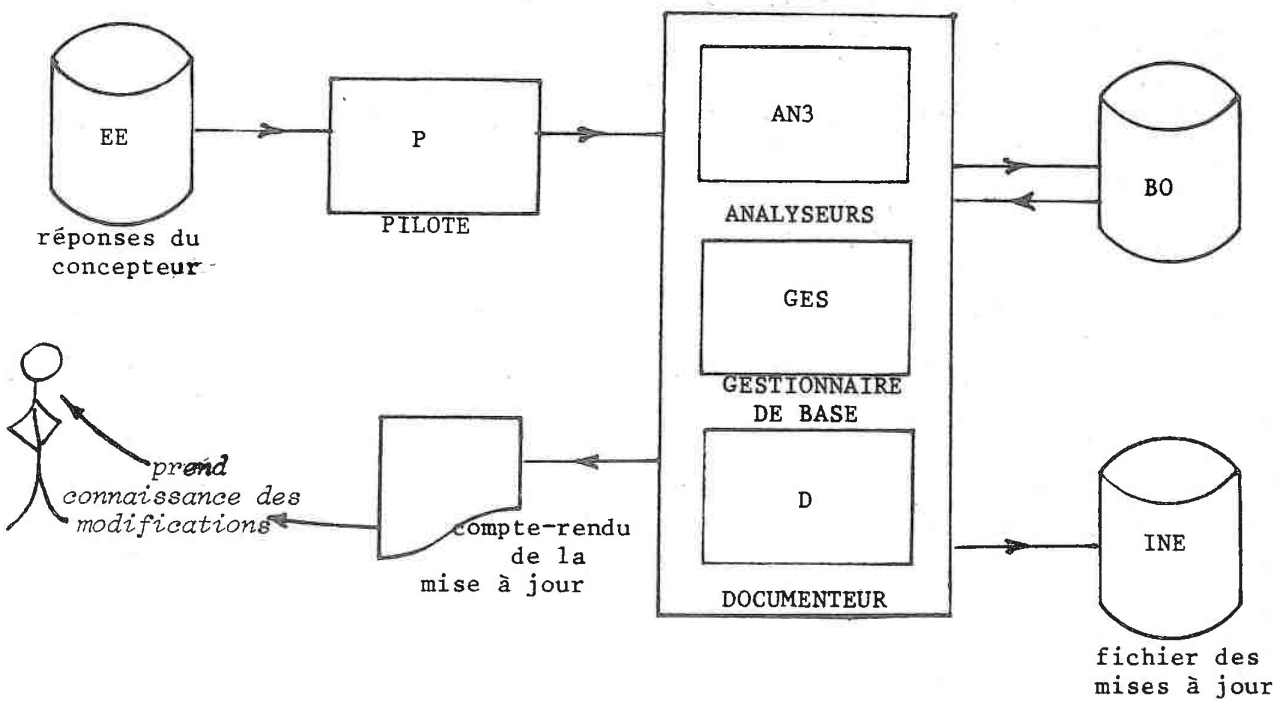
- figure 16 suite -

ZONE	EST RATTACHE A		ET A POUR MODE	
		CALBI		
CAT	EST RATTACHE A		ET A POUR MODE	
		CALBI		
		CALCOS		
TIT	EST RATTACHE A		ET A POUR MODE	
		CALBI		
		CALCOS		
ALX	EST RATTACHE A		ET A POUR MODE	
		CALBI		
		CALCOS		
SP	EST RATTACHE A		ET A POUR MODE	
		CALBI		RI
		CALCOS		
BF	EST RATTACHE A		ET A POUR MODE	
		CALBI		RI
BAF	EST RATTACHE A		ET A POUR MODE	
		CALBI		
EF	EST RATTACHE A		ET A POUR MODE	
		CALBI		RI
BAV	EST RATTACHE A		ET A POUR MODE	
		CALBI		RI
TAL	EST RATTACHE A		ET A POUR MODE	
		CALBI		RI
I	EST RATTACHE A		ET A POUR MODE	
		CALBI		RI
		CALCOS		RI
		STATIS		RI
TABF	EST RATTACHE A		ET A POUR MODE	
		CALBI		RI
IF	EST RATTACHE A		ET A POUR MODE	
		CALBI		RI
INDF	EST RATTACHE A		ET A POUR MODE	
		CALBI		
HSLF	EST RATTACHE A		ET A POUR MODE	
		CALBI		RI
NAH	EST RATTACHE A		ET A POUR MODE	
		CALBI		RI
HSNO	EST RATTACHE A		ET A POUR MODE	
		CALBI		
HSNLI	EST RATTACHE A		ET A POUR MODE	
		CALBI		
HSDIM	EST RATTACHE A		ET A POUR MODE	
		CALBI		
NHS	EST RATTACHE A		ET A POUR MODE	
		CALBI		RI
IR	EST RATTACHE A		ET A POUR MODE	
		CALBI		
TAB	EST RATTACHE A		ET A POUR MODE	
		CALBI		
INDEM	EST RATTACHE A		ET A POUR MODE	
		CALBI		RI
K	EST RATTACHE A		ET A POUR MODE	
		CALBI		RI
NBIN	EST RATTACHE A		ET A POUR MODE	
		CALBI		
CODE	EST RATTACHE A		ET A POUR MODE	
		CALBI		
MIND	EST RATTACHE A		ET A POUR MODE	
		CALBI		
NBIND	EST RATTACHE A		ET A POUR MODE	
		CALBI		

- figure 16 fin -

GREVE	EST RATTACHE A	CALBI	ET A POUR MODE	RI
NGREVE	EST RATTACHE A	CALBI	ET A POUR MODE	
RETTB	EST RATTACHE A	CALBI	ET A POUR MODE	RI
RETIR	EST RATTACHE A	CALBI	ET A POUR MODE	RI
RETSF	EST RATTACHE A	CALBI	ET A POUR MODE	RI
RETIP	EST RATTACHE A	CALBI	ET A POUR MODE	RI
NTB	EST RATTACHE A	CALBI	ET A POUR MODE	RI
NIR	EST RATTACHE A	CALBI	ET A POUR MODE	RI
NSF	EST RATTACHE A	CALBI	ET A POUR MODE	RI
NIP	EST RATTACHE A	CALBI	ET A POUR MODE	RI
CO	EST RATTACHE A	CALCOS	ET A POUR MODE	RI
		STATIS		RI
BAS	EST RATTACHE A	CALCOS	ET A POUR MODE	RI
TO	EST RATTACHE A	CALCOS	ET A POUR MODE	
PLAFS	EST RATTACHE A	CALCOS	ET A POUR MODE	
REAL	EST RATTACHE A	CALCOS	ET A POUR MODE	RI
BIRA	EST RATTACHE A	CALCOS	ET A POUR MODE	RI
PLAFIR	EST RATTACHE A	CALCOS	ET A POUR MODE	
TMUT	EST RATTACHE A	CALCOS	ET A POUR MODE	RI
OLI	EST RATTACHE A	CALCOS	ET A POUR MODE	
J	EST RATTACHE A	CALCOS	ET A POUR MODE	RI
BASE	EST RATTACHE A	DECOMP	ET A POUR MODE	V
L	EST RATTACHE A	DECOMP	ET A POUR MODE	RI
NL	EST RATTACHE A	DECOMP	ET A POUR MODE	RI
LN	EST RATTACHE A	DECOMP	ET A POUR MODE	RI
V	EST RATTACHE A	DECOMP	ET A POUR MODE	
TOT	EST RATTACHE A	STATIS	ET A POUR MODE	RI
CUMSAL	EST RATTACHE A	STATIS	ET A POUR MODE	RI

QUELQUES ASPECTS DU DEROULEMENT INTERNE



Compte tenu des réponses de l'utilisateur, le troisième module des ANALYSEURS peut compléter le fichier INE dont l'image complète, pour notre exemple est fournie par la figure 17.

A partir du fichier INE, le gestionnaire de base crée et met à jour la BO.

Les éditions systématiques sont produites par le DOCUMENTEUR qui édite les caractéristiques des occurrences des types-d'objet dont les réalisations ont été créées ou modifiées.

- figure 18 -

créations, mises à jour préparées par AN2

PAIE	EMPLOYE		
1STATIS	MA	STATISTIQUES	PAIE
2TOT	RI	CUMUL SALBRUT	STATIS
7BI	D	SALAIRE BRUT	STATIS
7COTISO	RI	CUMUL COTISATIONS	STATIS
7I	RII		STATIS
7CO IONS	RIT	TABLEAU DES COTISAT	STATIS

créations, mises à jour préparées par AN 3

2CUMSAL	RI	CUMUL SALNET	STATIS
7NETIMP	RI		STATIS

1 module désigné par STATIS et 2 mots (TOT et CUMSAL) doivent être créés

les liens mot-module munis de leurs caractéristiques "mode d'utilisation" entre

- *BI*
- *COTISO et STATIS*
- *I*
- *CO*
- *NETIMP*

doivent être mis à jour.

CONCLUSION

CHAPITRE V - CONCLUSION

Dans le cadre d'une méthode intégratrice des Données et des Traitements pour la conception des systèmes d' informations nous nous sommes interessés à l'aspect conceptuel en essayant de définir :

1- Quels types de phénomènes du monde réel on souhaite prendre en charge en informatique de gestion.

2- Quels concepts peuvent être proposés pour les représenter.

3- Comment un système d'aide à la conception peut conduire à un bon usage de ces concepts , de telle sorte que l'image conceptuelle obtenue soit une image fidèle , cohérente et complète de l'ensemble des phénomènes du monde réel pris en compte.

Ces propositions nous semblent ouvrir un axe de recherche qui est , apparemment , peu exploré pour l'instant.

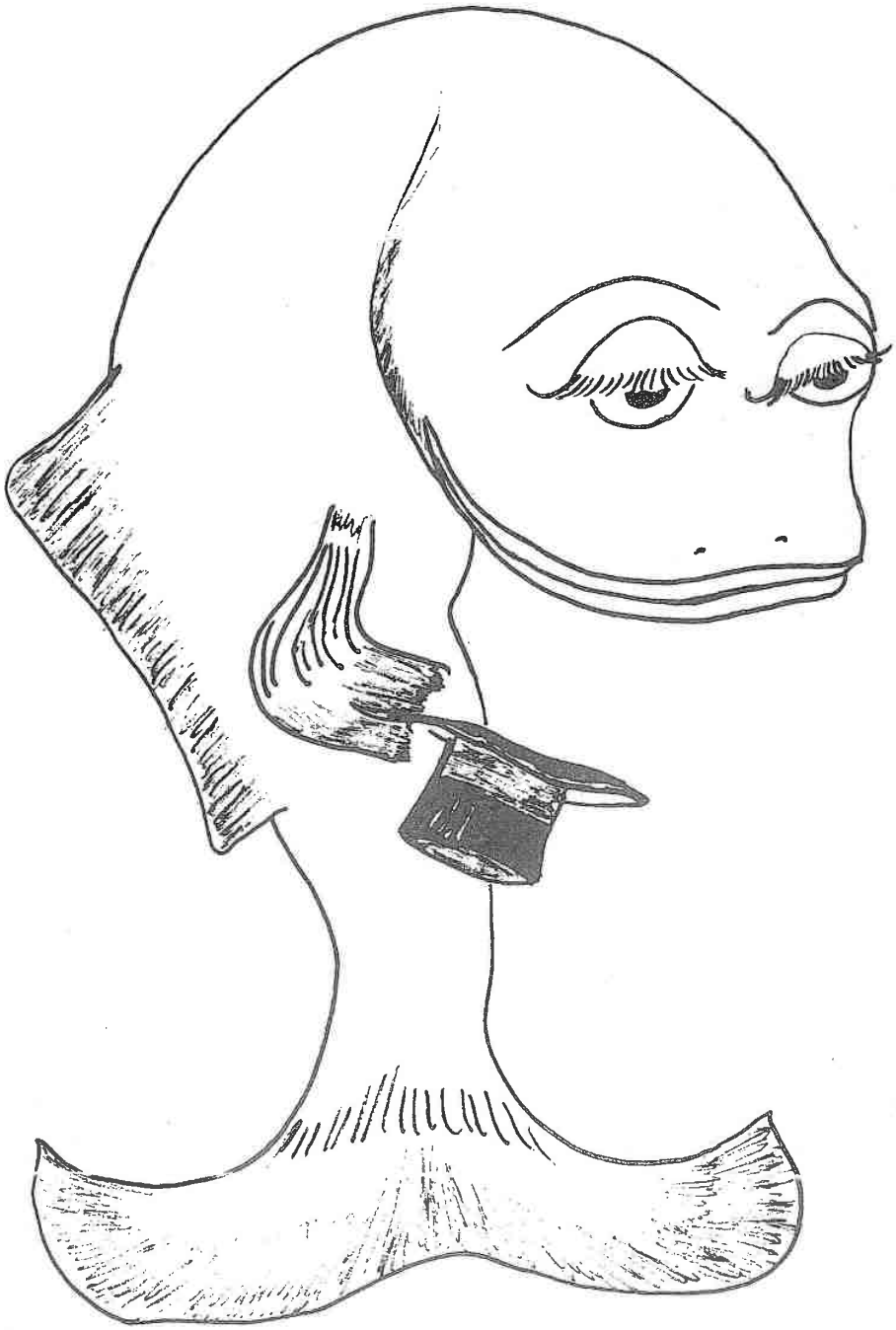
C'est la raison pour laquelle , si nous pensons avoir abouti à un certain nombre de résultats (dont nous ferons l'expérimentation) , nous sommes conscients des améliorations que l'on peut apporter aux solutions proposées et de l'ensemble des problèmes qui restent à résoudre.

Ainsi il est certain que le nombre des concepts sur lesquels repose cette première ébauche de SC , prenant compte à la fois de la définition conceptuelle des Données et de celle des Traitements , peut sans doute être réduit mais qu'en revanche d'autres caractéristiques , en particulier les caractéristiques temporelles , doivent être précisées.

Nous n'avons pas non plus été capables de répondre à la question : "Le schéma conceptuel obtenu par la prise en compte de plusieurs problèmes est-il la meilleure solution possible?"

De même nos propositions en matière de fonctions d'aide à la conception montrent bien , à notre avis , quels objectifs il faut poursuivre dans ce type d'actions mais mettent en évidence aussi toutes les réflexions sur les solutions de mise en oeuvre qui peuvent être développées.

C'est dans ce sens que nous poursuivrons ce travail.



BIBLIOGRAPHIE

BIBLIOGRAPHIE

I- J.R. ABRIAL

"Data Semantics"

IFIP working conference Cargese April 1974 pp. 1à59

2- J.R.ABRIAL

Projet SOCRATE

Cours avancé sur l'architecture des systèmes.Alpe d'Huez Dec. 72

3- ANSI/SPARC

"Report on Data Management Systems" 1975

4- G.BENCI,F.BODART,H.BOGAERT,A.CABANES

"Concept for the Design of a Conceptual Schema"

pp. 378 à 429 IFIP working conference FREUDENSTADT (All) 1976

5- M.T.BERTINI,Y.TALLINEAU

"Cobol structuré:Un modèle de programmation"

Edition d'informatique 1973

6- M.J.BLIN

"Structuration des dannées et des programmes,application à un problème de simulation" Thèse 3e Cycle Université PARIS VI 1975

7- F.BODART

"Structuration fonctionnelle du système informatique d'une organisation:propositions méthodologiques"

-Institut d'Informatique de Namur Avril 1975

-pp. 347 à 371 Séminaire Inforsid Aix en Provence 1975

8- F.BODART

"Cours d'analyse de système informatique de gestion"

Cours polycopié;Institut d'informatique Namur 1974

9- J.P.CAHEN,J.C.FAVRE,D.PORTAL,G.MAZARE,R.MOMIN

"Projet SOCRATE" .Nouvelles spécifications (version 3)

Université de Grenoble Septembre 1972

10- J.J.CHABRIER

"Acquisition et édition des fichiers,analyse des données dans le projet CIVA". Thèse de 3e Cycle Universite de Nancy 1973

II- A.DE CHELMINSKI

MACSI I

Thèse d'etat Université de Grenoble Juin I975

I2- M.CHESSERON

"Les élements conceptuels de la dynamique d'un SI"

Thèse de 3e Cycle à paraitre

I3- C.G.I.

"CORIG PAC" I966-I973

I4- Club Banque de Données

-"Rapport du groupe Analyse et Bases de Données" I975

-Bulletin de liaison du Club (volumes I0 à I5)

I5- Club Banque de Données

Colloque IRIA (journées d'Aix en Provence) 4 au 6 juin I973

I6- Codasyl

-"A survey of generalized Data Base Management Systems"

Systems Comitte 69

-"Data Base Group Report" ACM (New York I975)

I7- E.F.CODD

-"A relational Model of Data for large shared Data Banks"

Communication of ACM (vol.I3/number 6/June I970)

-"Normalized Data Base Structure:a brief tutorial"

Sigfider workshop II-I2 November I97I

-"Recent Investigations in relational Data Base System"

Information processing 74, North Holland I974

-"Further Normalization of the relational Data Base Model"

Courant Computer Sciences Symposium (New York I97I)

I8- D.COUGER

"Evolution of Business System Analysis Techniques computing surveys"

vol.5 num.3 I973

I9- CROCUS

"Principes de conception d'un système d'exploitation"

DUNOD I974

20- C.DATE

"An Introduction to Data Base Systems"

Addison Wesley I975

2I- C.DELOBEL

"A theory about Data in an Information System"

IBM Research Report R5.364 I972

22- J.C.DERNIAME

"CIVA:un système de programmation modulaire"

Thèse d'etat Université de Nancy I974

23- J.C.DERNIAME

" Un projet de fabrication et de documentation de programmes"

BIGRE n°I mars I976

24- DIJKSTRA

"Notes on structured programming in structured programming"

APIS studies in Data Processing academic Press I972

25- R.W.ENGLES

"A tutorial on Data Base Organisations"

IBM Technical Report TR 2004-I970

26- A.FLORY,J.KOULOUMDJIAN

"Sémantique et structure de données"

Rapport de recherche du laboratoire informatique Lyon I Déc. I973

27- GAMMA CONSEIL

"ARIANE" I970

28- J.S.GERO

"Ethics in Computer Aided Design:a polemic"

ACM SIGMA Newsletter vol.5 nr 4 I975

29- GUIDE/SHARE

"Data Base Management System Requirement"

SHARE Inc.(New York I970)

30- H.HABRIAS

"Méthode analytique,méthode syntétique en informatique d'organisation"

Informatique de gestion 1975

31- R.HURTEBISE

"Les systèmes d'informations aux fins de gestion"

Presses universitaires Quebec 1975

32- IFIP

Working Conference on modelling in Data Base Management

Freudenstadt Allemagne 1976

33- INFORSID Groupe II

"Méthodes et outils pour la conception et la réalisation des systèmes informatiques de gestion" Février 1976

34- D.JANKOWSKI

"Le pilotage de la phase de conception d'un SI dans REMORA"

Thèse de 3e Cycle Université de Nancy (à paraitre)

35- M.KRIEGER

"La description conceptuelle des données dans REMORA"

Thèse de 3e Cycle Université de Nancy (à paraitre)

36- D.LAMY

"Langage de conception et générateur associé dans REMORA"

Thèse de 3e Cycle Université de Nancy (à paraitre)

37- J.L.LEMOIGNE

"Les systèmes d'informations dans les organisations"

Presses universitaires 1974

38- P.LEVASSEUR,F.ALLIAUME,H.BRIAND,J.JAYEZ

"Conception et réalisation d'un logiciel conversationnel d'analyse fonctionnelle Rapport contrat CNRS-IRIA 1975

39- J.LUGUET

"SCAPFACE"

Thèse d'état Université de Toulouse 1975

40- MAC GEE

"File structures for generalized Data Management"

Proceedings IFIP's conference 1968 pp. 68 à 73

41- MALLET

"La méthode informatique"

Hermann 1971

42- MARTZLOFF

"Repenser l'Informatique"

Les éditions d'organisation 1974

43- Méthode MINOS

SLIGOS Systèmes 1970

44- H.D.MILLS

"Syntax-Directed Documentation for PL 360"

Communication of the ACM vol. 13/Nr. 4/April 1970

45- MISTRAL

Manuel d'utilisation et d'operation CII

46- P.MOULIN,J.RANDON,S.SAVOYSKI,S.SPACCAPIETRA,H.TARDIEU,M.TABOUL

"Conceptual model as a Data Base design tool"

IFIP working conference PP. 458 à 475 Freudenstadt (All) 1976

47- NCR

"ADS/PDS" 1967

48- NOLAN

"Managing the data ressource function"

49- J.C.NUMAMEKER

-"A methodology for the design and optimisation of Information processing systems" SJCC 1971

-"On the design and optimisation of Information Processing Systems" Ph.D Dissertation Case Western Reserve University 1969

50- D.ORY

"Software d'aide à l'analyse et à la programmation de gestion"

Thèse de 3e Cycle Université de Nancy Octobre 1972

51- C.PAIR

"Du problème à sa solution"

Journées IRIA Logique et Programmation 1975

52- C.PAIR,J.MAROLD

"Introduction à une méthode de programmation déductive"

Institut National Polytechnique de Nancy 1975

53- F.PECOUD

"MACSI-Méthode d'aide à la conception des systèmes d'informations"

Thèse d'état Université de Grenoble 1975

54- W.PEREA WILLACORTA

"Méthode d'analyse dans le projet REMORA"

These de 3e Cycle Université de Nancy Juin 1976

55- M.C.PORTMANN

"Utilisation des hypergraphes pour l'évaluation automatique des tailles des données" Thèse de 3e Cycle Nancy Septembre 1974

56- PROMO-INFORMATIQUE

"PARM-ARMIN" 1970

57- PROTEE

Société d' Informatique et de Systèmes 1970

58- REIX

"L'Informatique de gestion"

DUNOD 1970

59- C.ROLLAND (O.FOUCAUT,O.THIERY,D.LAMY,W.PEREA)

Rapport ATP-CNRS-IRIA 1 Document de travail Paques 1975

60- C.ROLLAND (O.FOUCAUT, O.THIERY,D.LAMY,W.PEREA)

Rapport ATP-CNRS-IRIA 2 Document de travail Octobre 1975

61- C.ROLLAND (O.FOUCAUT,O.THIERY,D.LAMY)

Rapport ATP-CNRS-IRIA 3 Document de travail Paques 1976

62- C.ROLLAND

"Analyse: du descripteur au superviseur"

Informatique et gestion n° 69 1975

63- SOCRATE

Manuel d'utilisation.Manuel d'opération

ERIA-ECA Automation Juillet 1974

64- SOCRATE

Manuel d'utilisation.Manuel d'opération

CII Juillet 1973

65- TAYLOR

"Generalized Data Base management system data structures and their mapping to physical storage" ISDOS working paper nr. 49 July 1971

66- D.TEICHROEW

"An introduction to computer based information processing system"

ISDOS working paper nr. 72 1973

67- A.THOMAS

"Techniques d'analyse en Informatique de gestion"

DUNOD 1970

68- WARNIER

"La transformation des programmes"

Ed. d'organisation 1973

69- WARNIER

"L'organisation des données d'un système"

Ed. d'organisation 1974

70- WATERS

"Méthodologie assistée par ordinateur dans la conception des systèmes informatiques" L'Informatique 1976

NOM DE L'ETUDIANT : MADAME LEGLISE EPOUSE THIERY ODILE

NATURE DE LA THESE : DOCTORAT DE SPECIALITE EN MATHEMATIQUES APPLIQUEES

VU, APPROUVE

& PERMIS D'IMPRIMER

NANCY, le 16/9/1976

LE PRESIDENT DE L'UNIVERSITE DE NANCY I

UNIVERSITE DE NANCY I
Le Président

M. BOULANGE