

50376
1986
35

50376
1986
35

№ d'ordre : 1316

THÈSE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

pour l'obtention du titre de

DOCTEUR DE 3^{ème} CYCLE

Spécialité : AUTOMATIQUE

par

DUPONT DANIEL



OBSERVATION ET AIDE A LA DECISION :

Application à la conception d'un observateur
numérique couplé à un système expert

Soutenue le 30 Janvier 1986 devant la Commission d'Examen

Mrs P. VIDAL
C. VASSEUR
J.M. TOULOTTE
C. CHOPIN
M. COUVREUR

Président
Rapporteur
Examineur
Examineur
Examineur

Ce travail est en partie financé par le Conseil
Régional du Nord/Pas de Calais et par l'I.N.S.E.R.M.
(CRL n° 805014).

AVANT - PROPOS

Le travail exposé dans ce mémoire est le résultat d'une collaboration suivie entre le Centre d'automatique de l'Université des Sciences et Techniques de Lille et le Centre de Réanimation Respiratoire de l'Hôpital Calmette de Lille.

Avant de présenter cette étude, je tiens à exprimer ma profonde gratitude à Monsieur le Professeur Pierre VIDAL, Directeur du Centre d'Automatique de Lille, qui m'a accueilli dans son laboratoire.

Monsieur le Professeur Christian VASSEUR a été responsable scientifique et rapporteur de cette thèse. Qu'il veuille bien trouver ici l'expression de ma reconnaissance.

Je remercie Monsieur Jean Marc TOULOTTE, Professeur à l'Université des Sciences et Techniques de Lille, de son aide et des conseils qu'il m'a apportés et pour sa présence parmi les membres du jury.

Que Monsieur le Professeur C. CHOPIN, responsable du Service de Réanimation à l'Hôpital B de Lille, accepte mes sincères remerciements pour l'accueil reçu au sein de son service. Je lui suis reconnaissant d'avoir accepté de juger ce travail dont il est l'initiateur.

Michel COUVREUR, Maître-Assistant à l'Institut Universitaire de Technologie de Béthune, a su trouver la patience pour m'encadrer durant cette étude. Que ce travail, qui est aussi le sien, soit l'expression de ma reconnaissance et de ma sincère Amitié pour l'attention, les conseils et l'aide constante qu'il m'a apportés.

Mes remerciements s'adressent enfin aux personnels et collègues chercheurs qui m'ont accueilli, notamment à Monsieur CEURSTEMONT pour son aide technique, Messieurs VEINNANT et HOUZE pour la réalisation pratique de ce mémoire, Brigitte, Bernard, Didier, François, Patrick, Philippe et, enfin Fabrice pour avoir su créer une ambiance agréable.

PLAN GENERAL

CHAPITRE I :

OBSERVATION DES PROCESSUS ET
AUTOMATISATION DE LA PRISE DE DECISION

I COMMUNICATION ENTRE OBSERVATEURS ET PROCESSUS

I.1 SYSTEME D'AIDE A LA COMMUNICATION

I.2 OBSERVATION EN PERCEPTION

I.3 OBSERVATION EN CONCEPTUALISATION (COMPREHENSION)

II AIDE A LA DECISION

II.1 OBSERVATION ET PRISE DE DECISION

II.2 MECANISME D'AIDE A LA DECISION

III CONCLUSION

PREMIERE PARTIE :
CONCEPTION DES SYSTEMES D'OBSERVATION

CHAPITRE II :
OUTILS DE CONCEPTION DES SYSTEMES D'OBSERVATION

I CONCEPTION DES SYSTEMES D'OBSERVATION

II NECESSITE D'UNE METHODOLOGIE DE CONCEPTION

III ETAT DE L'ART OU SYNTHESSES DES METHODES EXISTANTES

III.1 CONCEPTION LOGICIELLE

III.2 CONCEPTION MATERIELLE

III.3 METHODES GLOBALES

IV CONCLUSION

CHAPITRE III :
CONCEPTION D'UN SYSTEME DE SURVEILLANCE
INDIVIDUELLE DE MALADES EN REANIMATION
APPLICATION DE LA METHODE DEVELOPPEE PAR CALVEZ

I BUT DU PROJET

II PREMIERE ETAPE : DEFINITION DES SPECIFICATIONS

II.1 MODELE DE CALVEZ

II.2 APPLICATION AU PROJET

III DEUXIEME ETAPE : LA STRUCTURE FONCTIONNELLE

III.1 MODELE DE CALVEZ

III.2 APPLICATION AU PROJET

IV TROISIEME ETAPE : LA DESCRIPTION OPERATOIRE

IV.1 MODELE DE CALVEZ

IV.2 APPLICATION AU PROJET

V QUATRIEME ETAPE : LA STRUCTURE EXECUTIVE

V.1 MODELE DE CALVEZ

V.2 APPLICATION AU PROJET

VI DERNIERE ETAPE : LA REALISATION

VI.1 MODELE DE CALVEZ

VI.2 APPLICATION AU PROJET

VII CONCLUSION

DEUXIEME PARTIE :

PRISE DE DECISION

CHAPITRE IV :

LES SYSTEMES EXPERTS

I INTRODUCTION

II ARCHITECTURE D'UN SYSTEME EXPERT

III CONNAISSANCE - FAITS - MOTEUR D'INFERENCE

III.1 DICHOTOMIE CONNAISSANCE-EXPLOITATION

III.2 LA CONNAISSANCE

III.3 LES FAITS

III.4 MOTEUR D'INFERENCE

IV TYPE D'APPLICATION DES SYSTEMES EXPERTS

IV.1 CLASSIFICATION - DIAGNOSTIC

IV.2 SURVEILLANCE - CONTROLE

IV.3 RECHERCHE HEURISTIQUE

IV.4 C.F.A.O.

IV.5 E.A.O.

IV.6 SYSTEME D'AIDE A LA CONCEPTION DE SYSTEME EXPERT

V CONCLUSION

CHAPITRE V :
SYSTEME EXPERT EN LIGNE
CONCEPTION D'UNE STATION DE SURVEILLANCE
ET DE PRISE DE DECISION EN TEMPS REEL

I POSITION DU PROBLEME

- I.1 UTILISATION D'UNE TECHNOLOGIE PLUS RAPIDE
- I.2 STRUCTURATION DE LA BASE DE CONNAISSANCE
- I.3 AMELIORATION DES ALGORITHMES DE RESOLUTION
- I.4 INTRODUCTION DE PARALLELISME

II ETUDE DE L'EFFICACITE D'UN SYSTEME EXPERT

- II.1 PRESENTATION DE LA STRUCTURE INTERNE DES CONNAISSANCES
- II.2 TEMPS DE REPONSE PAR RAPPORT A L'ORDRE DES REGLES
- II.3 TEMPS DE REPONSE EN FONCTION DU NOMBRE DE REGLES

III AMELIORATIONS APORTEES AU MOTEUR D'INFERENCE

- III.1 AMELIORATION DU CHAINAGE AVANT
- III.2 AMELIORATION DU CHAINAGE ARRIERE
- III.3 AMELIORATION DUE A L'UTILISATION D'UNE TECHNOLOGIE PLUS RAPIDE
- III.4 CONCLUSION

IV PERSPECTIVES

IV.1 UTILISATION D'UNE MEILLEURE STRUCTURE INTERNE

IV.2 INTRODUCTION DE PARALLELISMES

IV.3 NOTION DE TEMPS

V CONCLUSION : SYSTEMES EXPERTS EN LIGNE

INTRODUCTION GENERALE

Le schéma général d'un système de contrôle/commande s'organise autour de trois pôles majeurs que sont :

- > l'observation du système à commander
- > l'élaboration d'une commande, compte tenu des observations et des grandeurs de consigne
- > l'application des commandes sur le système.

L'approche classique qui consiste à modéliser le système à commander est, pour certaines classes de problèmes, peu ou pas réalisable, dans la mesure où la connaissance de ces systèmes est empirique et de la compétence des experts du domaine, ou met en évidence une quantité de paramètres telle que l'élaboration d'un modèle est quasi irréalisable.

L'apport des développements en matière de systèmes experts permet de palier la difficulté de mettre au point un modèle formel, dans la mesure où un système expert constitue un dispositif qui intègre la connaissance du système observé sous une forme déclarative, liant les effets caractérisant le comportement du système, aux causes qui conduisent à ce comportement.

Le travail que nous présentons dans ce mémoire a pour objectif de décrire le principe d'une chaîne d'observation et de prise de décision dans un contexte temps réel, ainsi que les méthodes liées au développement d'un tel appareil.

Dans cet optique le mémoire est scindé en deux parties.

La première partie liée au système d'observation, correspond aux chapitres I, II et III dans lesquels sont respectivement abordés :

- > La notion d'observation et les concepts mis en oeuvre dans la suite du développement.
- > Les méthodes de conception existantes, tant logicielle que matérielle, de systèmes de contrôle/commande numériques, ainsi que les critères de choix d'une méthode particulière.
- > L'illustration des phases de conception d'un observateur numérique en utilisant comme support un système d'acquisition et d'analyse de signaux électrophysiologiques issus de malades soumis à une surveillance intensive et permanente.

La deuxième partie est quant à elle liée au dispositif de prise de décision correspondant aux deux derniers chapitres.

Le chapitre IV aborde l'organisation générale d'un système expert, ainsi que les critères qui permettent de l'insérer dans une chaîne automatique d'observation et de prise de décision.

Le chapitre V a, quant à lui, pour objectif de décrire un système conçu autour d'un moteur d'inférence d'ordre zéro et de donner les performances envisageables dans le cadre où ce système expert est couplé au générateur de faits constitué par l'observateur précédemment décrit.

PREMIERE PARTIE

CONCEPTION DES SYSTEMES D'OBSERVATION

CHAPITRE I

OBSERVATION DES PROCESSUS ET
AUTOMATISATION DE LA PRISE DE DECISION

CHAPITRE I :
OBSERVATION DES PROCESSUS ET
AUTOMATISATION DE LA PRISE DE DECISION

<u>I</u>	<u>COMMUNICATION ENTRE OBSERVATEUR ET PRISE DE DECISION.....</u>	p.	1
<u>I.1</u>	<u>SYSTEME D'AIDE A LA COMMUNICATION.....</u>	p.	1
<u>I.1.1</u>	<u>INTRODUCTION.....</u>	p.	1
<u>I.1.2</u>	<u>L'OBSERVATION.....</u>	p.	2
	<u>* La perception humaine.....</u>	p.	2
	<u>* Le percept.....</u>	p.	3
	<u>* Le concept.....</u>	p.	3
	<u>* Observation des processus physiques.....</u>	p.	4
<u>I.2</u>	<u>OBSERVATION EN PERCEPTION.....</u>	p.	5
<u>I.3</u>	<u>OBSERVATION EN CONCEPTUALISATION.....</u>	p.	6
<u>II</u>	<u>SYSTEME D'AIDE A LA DECISION.....</u>	p.	8
<u>II.1</u>	<u>OBSERVATION ET PRISE DE DECISION.....</u>	p.	8
<u>II.2</u>	<u>MECANISME D'AIDE A LA DECISION.....</u>	p.	9
<u>II.2.1</u>	<u>ROLE DE L'EXPERT, DU COGNITICIEN.....</u>	p.	9
<u>II.2.2</u>	<u>PHASE D'INTEGRATION.....</u>	p.	10
<u>II.2.3</u>	<u>PHASE D'EXPLOITATION.....</u>	p.	12
<u>III</u>	<u>CONCLUSION.....</u>	p.	13

I COMMUNICATION ENTRE OBSERVATEUR ET PROCESSUS

I.1 SYSTEME D'AIDE A LA COMMUNICATION

I.1.1 INTRODUCTION

Le fonctionnement d'un processus se traduit par un ensemble de manifestations (génération de signaux sonores, visuels, électriques...) représentant autant de messages qui permettent à un élément extérieur d'entrer en communication avec ce processus.

Par cette communication, l'élément extérieur ou observateur cherche à approfondir sa connaissance et sa compréhension du processus.

Cette communication couvre alors deux finalités essentielles :

- * L'observation du processus dans le but d'approfondir la connaissance du processus.

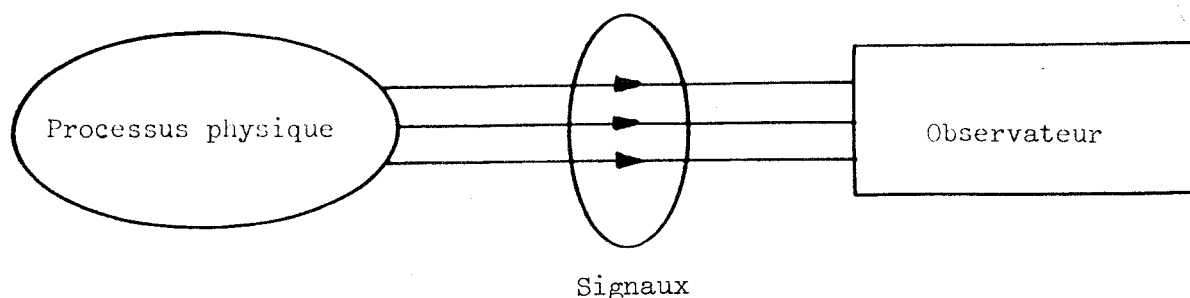


Figure I.1

Ce principe est retrouvé dans l'observation du ciel par l'astronome.

En effet celui ci observe uniquement dans le but de mieux comprendre et connaître les différentes constellations.

- * L'observation du processus dans le but de décider d'une action à entreprendre.

Le processus fournit des informations susceptibles d'influencer l'observateur dans le choix de sa commande en fonction de différents critères de commande.

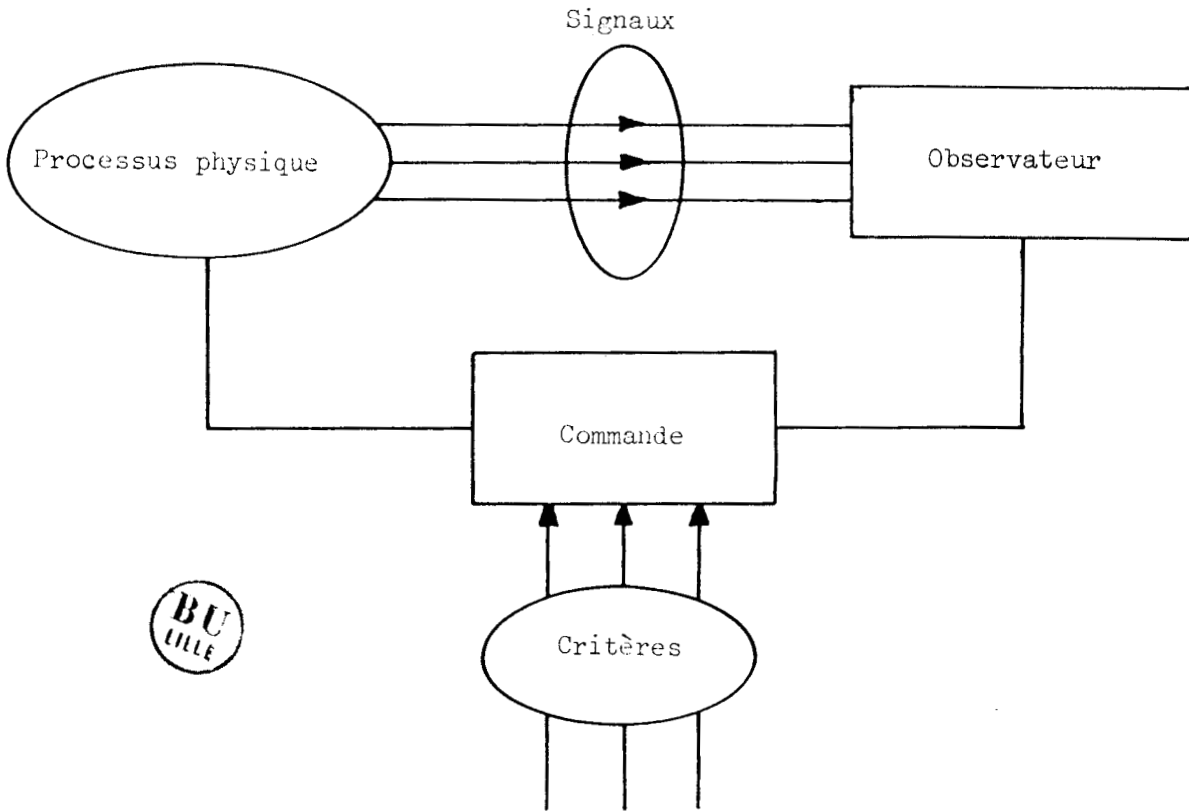


Figure I.2

C'est le cas des systèmes de surveillance ou de contrôle des processus avec notion de commande sur celui ci : ceci implique une "boucle de retour" sur le processus.

Dans le cadre de cette étude, nous nous sommes intéressés uniquement à la communication du processus vers l'observateur.

Cette communication unidirectionnelle constitue l'observation.

I.1.2 L'OBSERVATION

* La perception humaine

Face à une situation, l'homme dispose de ses différents sens qui lui fournissent des informations et le renseignent sur son environnement.

De fait, il adapte son comportement face à ces informations qu'il doit reconnaître, classer et analyser

"La perception peut donc être définie comme l'ensemble des mécanismes qui agissent sur l'information sensorielle et qui en réalisent l'interprétation, la classification et l'organisation." /SIRI-84/

L'interprétation de l'information sensorielle (ou image réelle de l'environnement) fait donc intervenir des notions d'abstraction, basées sur la mémorisation des situations ou images précédentes.

C'est pourquoi deux objets mentaux ont été définis dans la perception humaine :

- Le percept
- le concept.

** Le percept*

"Le percept primaire représente en quelque sorte une image, une copie presque fidèle de la réalité, son contenu sensoriel est riche". /CHAN-83/

Or face aux différentes informations accessibles, le cerveau opère un tri en ne retenant que les données présentant un intérêt :

La perception apparaît dès le début comme un processus sélectif /SIRI-84/.

Cette sélection de l'information, au regard d'une situation donnée, est conditionnée par le vécu de l'individu, qui représente, en quelque sorte, son expérience, son apprentissage :

De l'image d'un chien, un individu ne retiendra que les crocs si son vécu est entaché d'expériences malheureuses avec les chiens.

** Le concept*

A partir du percept l'individu fait une opération d'abstraction :

L'image du chien déclenche une réaction de peur chez l'individu précédent.

Il a ainsi associé dans son passé :

Chien (image réelle) = Peur (image abstraite)

et la simple vue d'un chien, même si celui ci ne présente aucune réaction d'agressivité, lui "commandera" un sentiment de crainte.

Du percept au concept, il existe une continuité :
"Le percept est, comme l'image, un objet de mémoire, mais ne possède qu'une faible composante sensorielle, voire pas du tout..." /CHAN-83/.

Le résultat final de la perception, "considéré comme une interprétation de la sensation", est lié au phénomène de mémorisation sélective ou préhension élémentaire. Il est bien connu, en psychologie, que l'on ne perçoit bien que l'image affective, celle qui est chargée d'une signification personnelle.

C'est ainsi que l'accumulation des expériences passées peut modifier la perception elle-même, privilégier certains détails, en effacer d'autres de façon absolue et imprimer sa marque sur le comportement de l'individu". /MORI-74/

* Observation d'un processus physique

Il existe certaines analogies entre la perception humaine et l'observation d'un processus.

En effet un processus émet des informations qui caractérisent son état.

Face à ces informations, l'observateur (homme ou machine) sélectionne, par le biais des capteurs utilisés, les informations présentant un intérêt.

Par exemple, lors de l'analyse d'un électrocardiogramme (ECG), on ne s'intéresse suivant le contexte (réanimation, contrôle de routine...) qu'à certaines dérivations bien définies.

Ensuite de l'information issue du capteur, on ne retient que certaines caractéristiques ou composantes (par exemple sélection d'un certain spectre fréquentiel par la définition de la bande passante d'un filtre).

Cette sélection de l'information utile réalisée, l'observateur interprète ces données en fonction d'une expérience, d'une compétence qui lui a été fournie et/ou qu'il a mémorisée (cette phase d'apprentissage représente pour un observateur humain sa formation d'étudiant et professionnelle tandis que pour un mécanisme elle représente le cycle de résolution que le concepteur lui a intégré).

Ainsi la définition de la perception donnée dans /SIRI-84/ peut être reprise pour l'observation :

L'observation est l'ensemble des mécanismes qui agissent sur l'information sensorielle (issue des capteurs) et qui en réalisent l'interprétation, la classification et l'organisation.

De même que deux notions avaient été distinguées dans la perception humaine :

- Percept (composantes sensorielles de l'information)
- Concept (abstraction de l'information)

VASSEUR /VASS-82/ décompose l'observation d'un processus en deux phases distinctes :

- La perception à laquelle est liée la notion de capteurs (similaire à un organe des sens, et donc possédant une composante sensorielle)
- La compréhension à laquelle sont liées les notions d'interprétation et de signification des objets perçus (abstraction de l'information).

Par analogie avec la perception humaine nous parlerons d'observation en conceptualisation plutôt que d'observation en compréhension.

I.2 OBSERVATION EN PERCEPTION

L'observation en perception a deux finalités :

- > sélectionner l'information utile par le choix des capteurs utilisés, et ne retenir que les composantes des signaux nécessaires à leur interprétation.
- > adapter l'information au mécanisme qui opère l'observation en conceptualisation.
En effet si cette étape est réalisée par un outil informatique et que le capteur délivre un signal analogique il est nécessaire d'utiliser un convertisseur analogique numérique, transformant l'information dans une forme perceptible par cet outil.

A ce niveau de l'observation il n'y a pas eu d'interprétation du signal, seul le contenu "sensoriel" de l'information a été affecté.

Néanmoins les dispositifs qui opèrent cette perception implique plus ou moins une perte de l'information qu'on cherche à minimiser.

La conversion analogique numérique en est certainement le meilleur exemple :

La fréquence d'échantillonnage est choisie de telle manière que le signal ne soit dégradé que dans la limite où il reste représentatif du processus étudié.

Ainsi l'observation en perception représente une interface entre le processus et l'observation en conceptualisation.

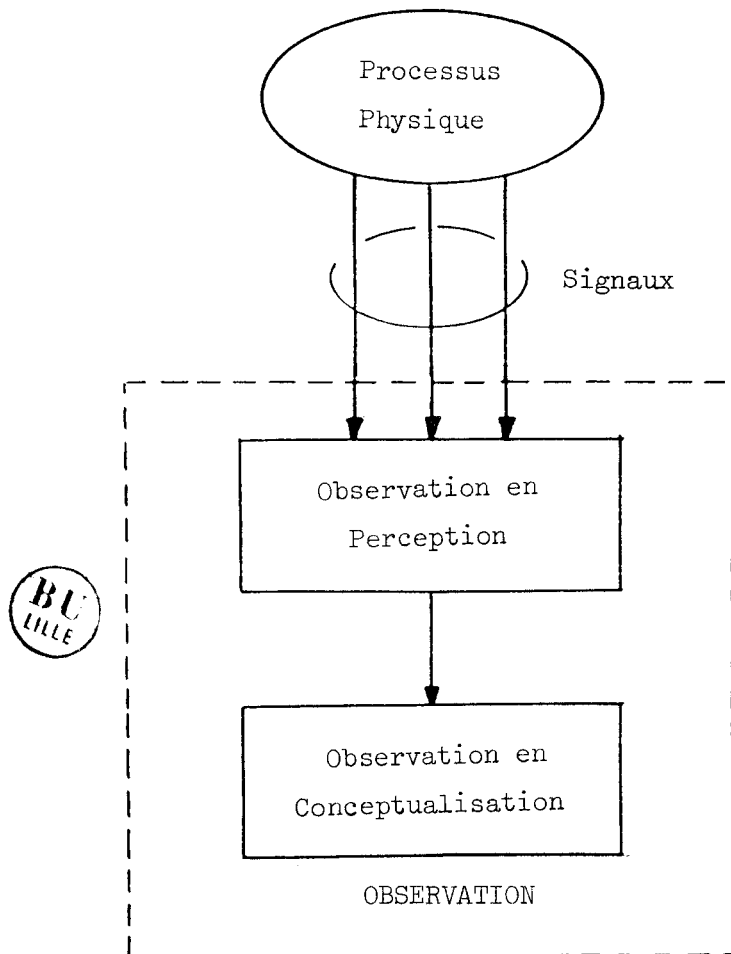


Figure I.3

I.3 OBSERVATION EN CONCEPTUALISATION

Le but de cette étape est d'extraire les informations utiles à la compréhension du processus.

Ces informations doivent permettre d'accéder à une connaissance non seulement quantitative du processus par l'extraction de données numériques (fréquence, tension...) mais aussi à une connaissance qualitative permettant de caractériser l'état du processus observé.

Par exemple si on s'intéresse à l'analyse de l'ECG, la détermination de la fréquence cardiaque permet d'accéder à une information qualitative sur l'état du patient à savoir si il est :

- soit en bradycardie
- soit dans un état normal
- soit en tachycardie.

Cette phase d'observation permet donc d'extraire des informations de plus en plus élaborées.

De même qu'à partir des percepts, le cerveau effectue une suite de généralisation qui aboutit à des notions de plus en plus abstraites /SIRI-84/, le couplage perception/conceptualisation décrit une structure récursive dans la mesure où un niveau de conceptualisation peut lui même devenir un niveau de perception et ainsi de suite .../VASS-82/.

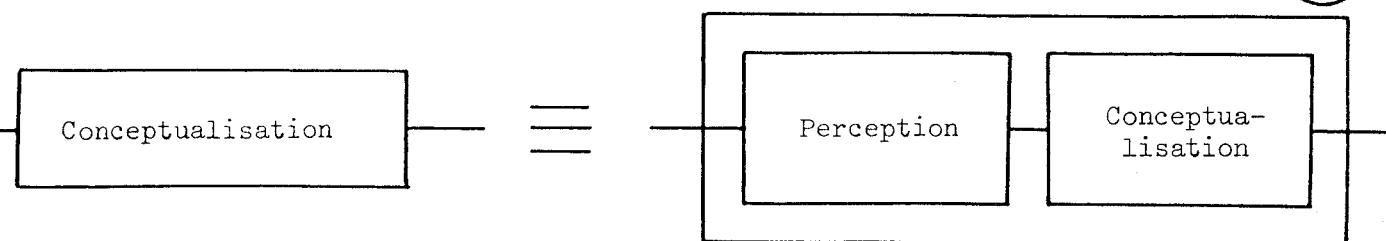


Figure I.4

On retrouve donc la continuité entre la perception et la conceptualisation existant dans la perception humaine.

Cependant plus on accède à des niveaux élevés d'abstraction de l'information plus cette information perd de son contenu sensoriel.

Si on reprend l'exemple de la détermination du rythme cardiaque nous pouvons dire que la valeur numérique de la fréquence cardiaque possède encore une composante sensorielle au sens où l'on peut reconstruire le signal de

l'ECG (même infidèlement) en fonction de modèle de référence, par contre lorsqu'on dit qu'un patient est en bradycardie on n'a même plus accès à la composante temporelle du signal, on sait seulement que la fréquence cardiaque se situe dans une certaine fourchette.

Cependant ces informations très élaborées permettent d'appréhender le processus très rapidement et d'en connaître les caractéristiques essentielles.

Dans ce sens, la conception d'un système d'observation ajoute à la notion de perception, les notions d'aide à la communication et à la compréhension du processus et constitue donc un outil d'aide à la réflexion et à la décision.

II SYSTEME D'AIDE A LA DECISION

II.1 OBSERVATION ET PRISE DE DECISION

L'observation tel qu'elle a été définie précédemment permet l'appréciation du comportement du système en fournissant des informations plus ou moins élaborées suivant le degré d'observation retenu.

Ces informations doivent aider un observateur dans son raisonnement afin de le guider vers la solution recherchée qui peut être l'émission d'un diagnostic ou la prise d'une décision (commande ou action sur le processus).

Ce diagnostic a pour but de caractériser la normalité du système observé, c'est à dire de décider si le système est dans un état normal ou pas.

S'il est dans un état normal il n'y a pas d'action à opérer, par contre il y a lieu d'agir sur le système afin de le ramener dans un état normal si une anomalie est détectée.

Pour exemple, considérons un système de régulation en position.

La position courante du système est comparée à la consigne et si il y a concordance il n'y a pas lieu d'agir.

Par contre si la position diffère de la consigne, on agit sur le système au moyen de régulateur (PI, PID...).

Dans cet exemple, le caractère normal du système correspond à une erreur nulle.

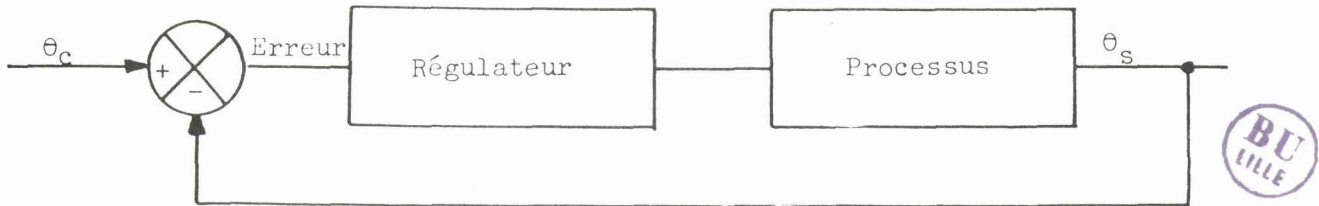


Figure I.5

Dans ce système l'observation, la prise de décision, la commande sont faites automatiquement sans intervention humaine.

Ainsi la décision consiste, face aux informations disponibles, à émettre un avis sur le comportement normal ou non du système observé.

La notion de normalité correspond à des critères définis par des personnes habituées à traiter les informations issues de la surveillance de ces processus. Ceci implique donc une compétence ou une expertise.

Dans ce sens la conception d'un système d'observation dit "intelligent" doit fournir un niveau d'abstraction des informations plus élevé que le percept et le concept afin de fournir une proposition de décision.

Ce niveau d'abstraction supplémentaire, représentant une étape plus ambitieuse que les précédentes, doit intégrer les résultats de l'observation en conceptualisation et doit élaborer à partir de ces informations un raisonnement aussi proche que possible du raisonnement humain de manière à proposer un diagnostic.

II.2 MECANISME D'AIDE A LA DECISION

II.2.1 ROLE DE L'EXPERT, DU COGNITICIEN

Lorsque l'on désire construire un système décisionnel on doit lui "fournir" une masse importante de connaissances représentant l'expertise du domaine étudié.

L'homme chargé d'intégrer cette connaissance dans le système, appelé "Cogniticien" ou "Ingénieur cognitif", ne possède généralement pas cette expertise du domaine et donc, pour mener à bien ce projet, il doit faire appel à un homme ayant cette compétence : le nom d'"Expert" désigne cet homme de l'art.

L'expert a pour rôle de transmettre au cogniticien, non pas l'ensemble de ces connaissances mais le schéma des relations qui permettent à partir des faits initiaux d'établir des conclusions sur la base de son raisonnement.

Ainsi l'utilisateur futur, qui n'est pas forcément un expert, trouvera dans le système une connaissance réaliste du processus.

Aussi, il est évident qu'il n'y aura d'aboutissement au projet que si un dialogue s'instaure entre l'expert et le cogniticien.

Ce dialogue nécessaire avant toute expérience permet d'établir une maquette du processus de raisonnement :

De l'examen de cette maquette le cogniticien aura comme tâches :

- > L'étude de faisabilité
- > La réalisation du prototype de raisonnement
- > Le contrôle de la validation du prototype par l'expert.

Donc si on veut simuler un raisonnement proche de celui de l'expert, il faut connaître comment il procède et rassembler non seulement toutes les connaissances qu'il utilise mais aussi les différentes stratégies de résolution et leurs conditions d'application.

En effet l'expert utilise en plus d'une connaissance formelle, une connaissance de nature heuristique, traduisant ses intuitions et son expérience.

Le cogniticien doit assurer l'intégration dans le système décisionnel de ces différentes connaissances.

II.2.2 PHASE D'INTEGRATION

Cette phase d'intégration conduit l'expert à exprimer et à formaliser son savoir sous forme de règles de décision dont l'ensemble constitue, ce qu'on appelle, une base de connaissance du système décisionnel.

La définition de cette base de connaissance apparaît dès le début de la conception comme une étape fondamentale.

On peut schématiser cette étape par le synoptique suivant (figure I.6) :

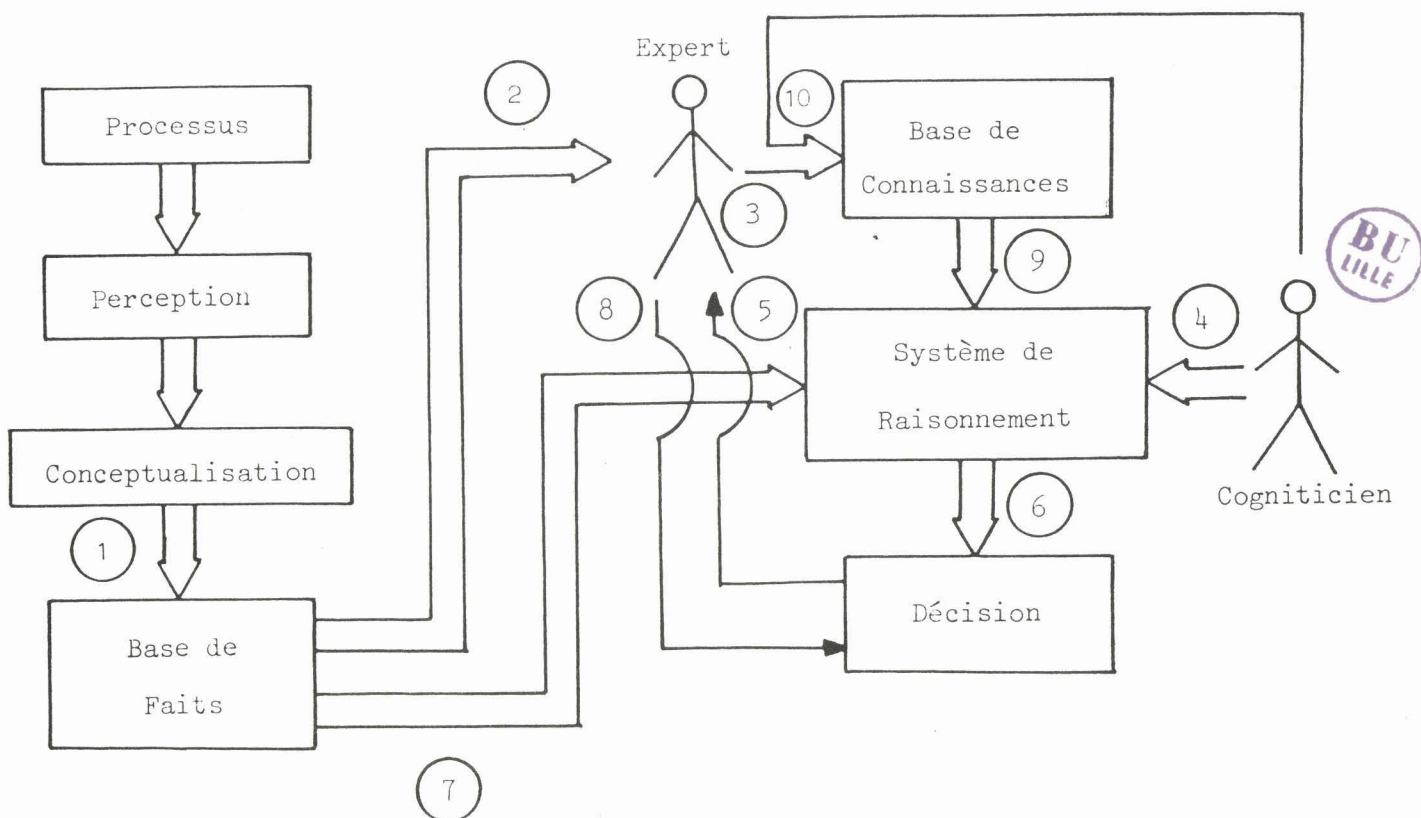


Figure I.6

- (1) L'observation en conceptualisation est considérée comme un générateur de faits et alimente ce qu'on peut nommer la base de faits (faits = paramètres, caractéristiques, tendances du processus...)
- (2) L'expert appréhende les faits, et à partir de sa connaissance et de son expérience émet une décision ou un diagnostic (8)
- (3) Son rôle est alors d'analyser sa démarche de raisonnement afin de pouvoir la formaliser et permettre son introduction dans la base de connaissance. Durant cette étape il est aidé par le cogniticien (10).
- (4) Le cogniticien est chargé de construire un module capable d'appréhender la base de faits (7) et à partir de la base de connaissance (9) d'émettre une proposition de décision (6). •

(5) L'expert compare alors la décision émise par le système à la sienne et si les deux coïncident on peut affirmer que le système décisionnel répond bien à ce qu'on attend de lui dans le cas envisagé.

Sinon deux cas sont envisageables :

--> soit l'expert a mal formalisé sa connaissance.

--> soit le cogniticien a conçu un module n'utilisant pas de manière satisfaisante la base de faits et la base de connaissance.

Lorsque les décisions du système et de l'expert correspondent dans tous les cas de figures traités, le système décisionnel peut passer à la phase d'exploitation.

II.2.3 PHASE D'EXPLOITATION

On peut alors schématiser son utilisation par le synoptique suivant (figure I.7) :

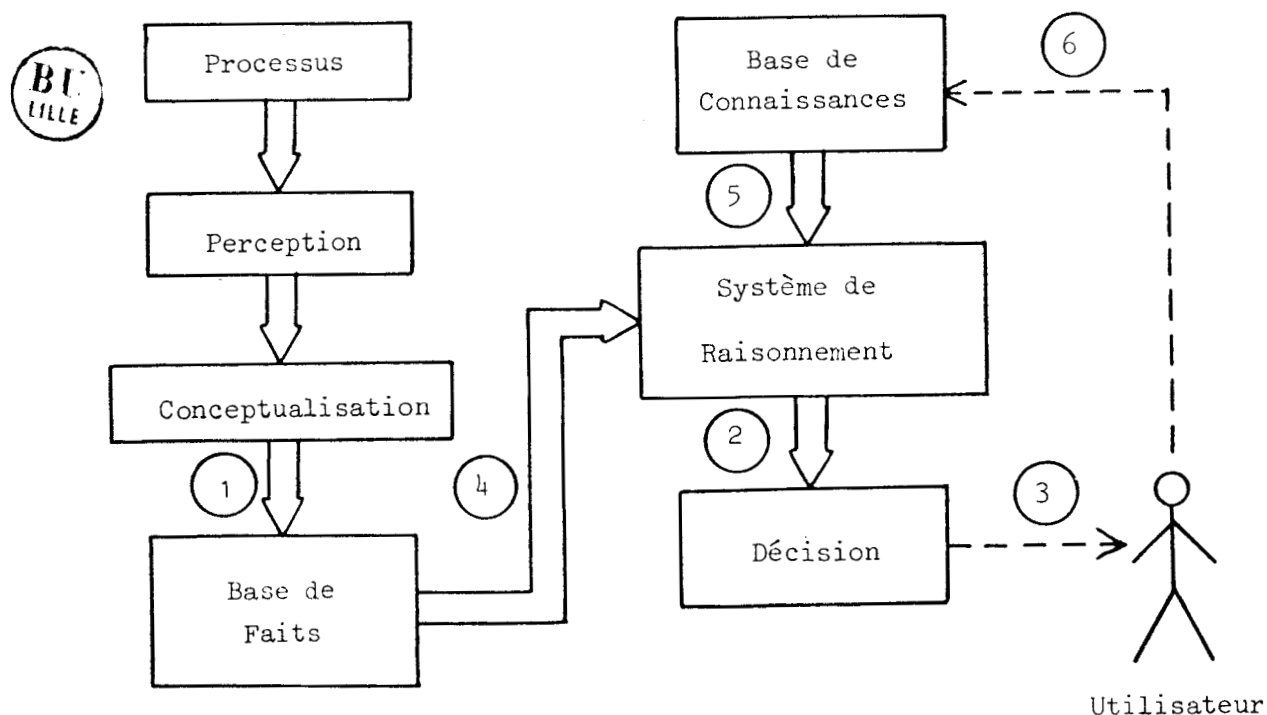


Figure I.7

- (1) Comme précédemment la base de faits est alimentée par les résultats de l'observation en conceptualisation.
- (2) Le système de raisonnement élabore une décision à partir des faits (4) et de la connaissance fournie par l'expert (5)
- (3) Ainsi l'utilisateur dispose d'une proposition de diagnostic et s'il le désire, il peut remettre à jour la base de connaissance.
Cette facilité permet à l'expert de modifier la base de connaissance en rectifiant une connaissance qu'il juge inappropriée.
De plus nous avons vu que la base de connaissance devait renfermer la meilleure connaissance, on doit donc permettre à l'utilisateur de modifier cette base de connaissance afin qu'il puisse ajouter les progrès faits dans la connaissance du domaine.

Cette capacité offerte à l'utilisateur de pouvoir modifier la base de connaissance représente une grande qualité d'un système décisionnel :

Elle autorise son évolution et apporte une flexibilité au système.

III CONCLUSION

Le premier chapitre de ce mémoire a permis de montrer que l'observation d'un processus pouvait se concevoir à deux niveaux de complexité différents:

- > L'observation en conceptualisation
Cette étape réalise l'interprétation de l'observation.
- > L'observation dite "intelligente"
A partir de l'interprétation des informations issues de l'observation en conceptualisation, le système propose une décision en fonction de la compétence qu'il a du domaine et qu'un expert a fourni.

Dans la suite de ce travail nous développons deux parties :

- > Une première partie traite de la conception des systèmes d'observation.
Cette partie est décomposée en deux chapitres.

- (1) Un premier chapitre expose les outils de conception des systèmes d'observations et la nécessité de suivre une méthodologie de conception.
- (2) Le deuxième chapitre applique la méthode développée par CALVEZ, /CALV-82/ et /CALV-83/, à la conception d'un système de surveillance de malades en réanimation.

Le système tel qu'il est présenté constitue un exemple de système d'observation en conceptualisation, interprétant les signaux issus de l'observation de différentes fonctions physiologiques traitées en réanimation :

--> La fonction cardiovasculaire.

Par l'analyse du signal de l'électrocardiogramme (ECG) et de la tension artérielle (TA) ou de la pression artérielle pulmonaire (PAP).

--> La fonction respiratoire.

Par la mesure des échanges gazeux et de la mécanique ventilatoire (signaux issus d'un respirateur : CO₂, Pression, Volume).

Cette unité de surveillance sera complétée par un système décisionnel afin de constituer un observateur intelligent au sens défini précédemment.

--> C'est pourquoi la deuxième partie traite de la prise de décision.

Un premier chapitre présente et définit ce qu'est un système expert. Le deuxième chapitre traite de l'étude de l'efficacité d'un moteur d'inférence et des améliorations qu'il est possible de lui amener afin de diminuer les temps de réponse et qu'il puisse prendre en compte l'aspect temporel et évolutif de l'information. Ceci de manière à rendre possible le couplage de ce système expert à un dispositif de surveillance de processus considéré comme un générateur de faits alimentant en ligne la base de faits de ce système expert. /COUV-85/

CHAPITRE II

OUTILS DE CONCEPTION DES SYSTEMES D'OBSERVATION

CHAPITRE II :

OUTILS DE CONCEPTION DES SYSTEMES D'OBSERVATION

<u>I</u>	<u>CONCEPTION D'UN SYSTEME D'OBSERVATION.....</u>	p.	1
<u>II</u>	<u>NECESSITE D'UNE METHODOLOGIE DE CONCEPTION.....</u>	p.	4
<u>III</u>	<u>ETAT DE L'ART OU SYNTHESE DES METHODES EXISTANTES.....</u>	p.	7
<u>III.1</u>	<u>CONCEPTION LOGICIELLE.....</u>	p.	7
<u>III.1.1</u>	<u>CONCEPTION.....</u>	p.	7
<u>III.1.2</u>	<u>DEVELOPPEMENT.....</u>	p.	12
<u>III.2</u>	<u>CONCEPTION MATERIELLE.....</u>	p.	12
	1- <i>Structure circulaire.....</i>	p.	13
	2- <i>Structure avec un bus commun.....</i>	p.	14
	3- <i>Structure centralisée.....</i>	p.	14
	4- <i>Structure pleinement connectée.....</i>	p.	15
<u>III.3</u>	<u>METHODES GLOBALES.....</u>	p.	16
<u>IV</u>	<u>CONCLUSION.....</u>	p.	18

I CONCEPTION D'UN SYSTEME D'OBSERVATION

La conception d'un système d'observation ou d'une centrale de surveillance conduit à constater que :

1- L'observation d'un processus se fait généralement par l'intermédiaire de plusieurs signaux, caractéristiques du processus, qui évoluent de manière simultanée.

Cette simultanéité conduit le système d'observation à intégrer la notion de tâches parallèles capables de traiter l'ensemble de ces signaux.

2- La prise de décision, qui suit la phase d'observation, implique des contraintes de temps qui dépendent de la dynamique du processus observé.

En effet, il faut s'assurer que l'occurrence d'un événement est détectée et induit la prise de décision correspondante dans la limite de temps où le traitement, associé à cette décision, agit efficacement sur le processus.

Pour cela le système d'observation doit analyser l'ensemble des signaux et proposer une décision en temps réel, et donc réagir en un minimum de temps.

Ces deux notions majeures que sont :

- le parallélisme
- l'analyse en temps réel

se rencontrent également dans la conception des systèmes de contrôle/commande complexes.

Les systèmes d'observation et de prise de décision sont proches de ces systèmes de contrôle/commande.

En effet, dans la chaîne contrôle/commande d'un processus (Figure II.1), l'observation détermine l'état dans lequel se trouve le processus; il s'ensuit l'élaboration d'une action qui amène le processus dans un état souhaité par l'opérateur (consigne).

Le choix de l'action, dans ce cas, correspond à la prise de décision.

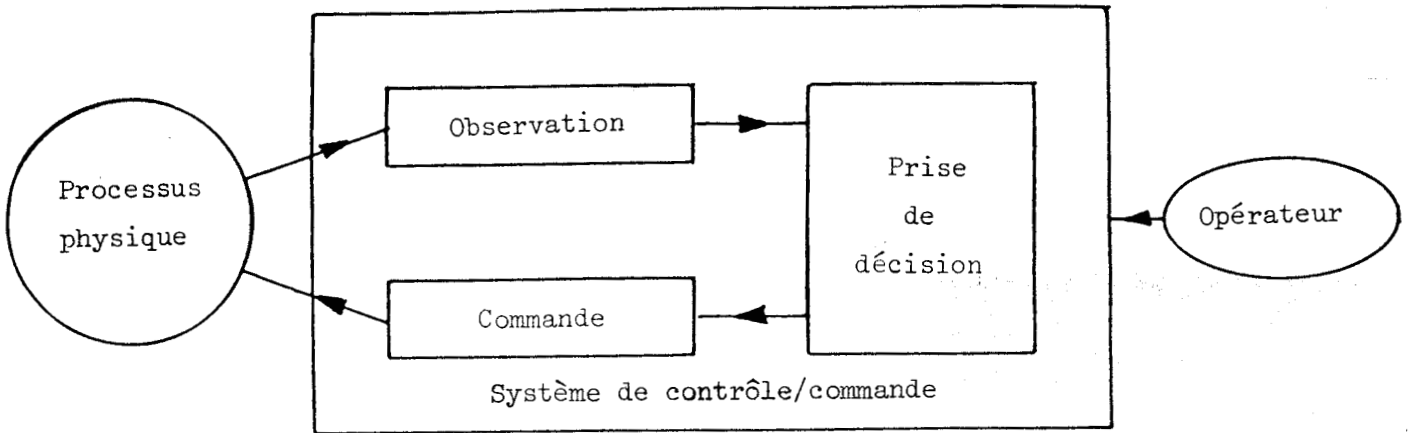


Figure II.1

L'évolution de l'état du processus, face à une commande, va induire, par le biais de l'observation de cet état, une modification de la commande.

L'observation joue le rôle de "feed back" ou boucle de retour des processus de régulation ou d'asservissement, bien connu des automaticiens (critère d'observabilité).

Dans le cas d'un observateur "intelligent", on ne s'intéresse qu'à l'observation et à la prise de décision. (Figure II.2)

Cet outil est vu comme une aide à la décision nécessitant le contrôle et l'expertise humaine.

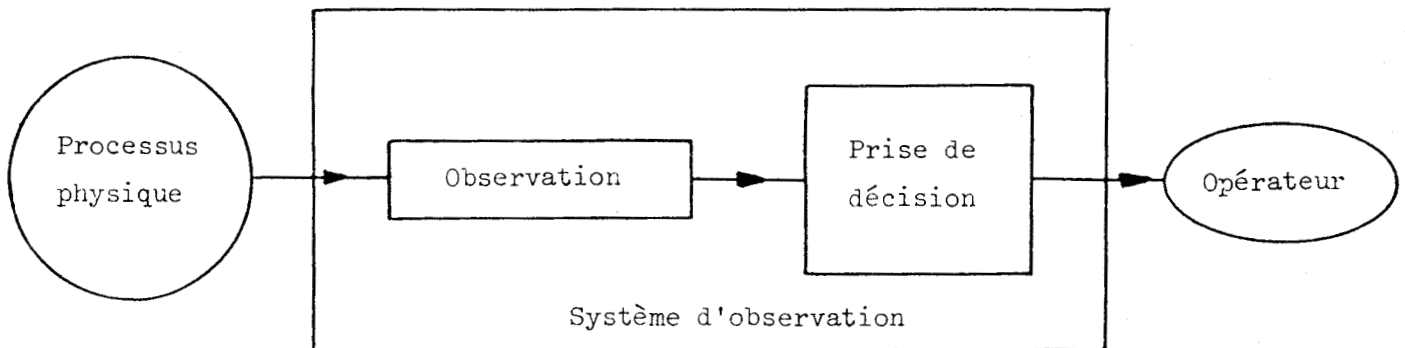


Figure II.2

Considérons le P.C.C. (Poste de Contrôle et de Commande) d'une centrale nucléaire, par exemple.

Le P.C.C. a été conçu pour résoudre le problème de communication entre les hommes et le réacteur /PIGN-81/.

Le pilotage du réacteur est automatisé et les opérateurs n'ont qu'une fonction de surveillance.

Cependant, en cas d'anomalies ou de perturbations du réacteur, l'opérateur est averti par la génération d'une alarme et doit donc "agir" en conséquence : le réacteur communique une information à l'opérateur et attend une réponse en retour.

PIGNON décrit ce principe en écrivant dans /PIGN-81/ :

" La machine parle à l'opérateur, elle communique par l'intermédiaire des petits voyants de plastique blanc, vert et rouge (alarmes). La machine réclame quelque chose. L'opérateur lui donne ce qu'elle veut ... Le voyant s'éteint, la machine est satisfaite, l'opérateur aussi."

" Continuellement, la machine réagit à toutes les actions entreprises par les conducteurs en imprimant la liste de tous ses changements."

Ainsi l'opérateur sert de lien ou boucle la chaîne système d'observation-commande en amenant son savoir et en contrôlant l'état du processus afin de déterminer l'action à entreprendre (Figure II.3).

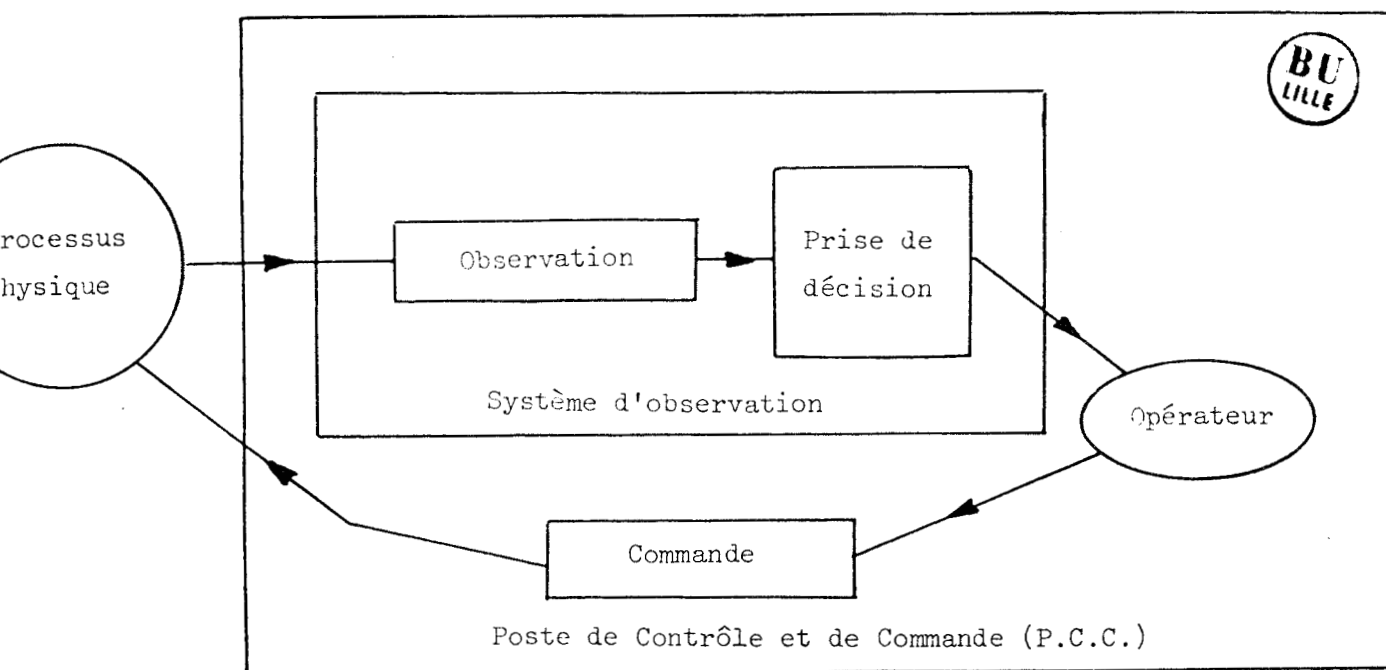


Figure II.3

La différence entre les systèmes de contrôle/commande et les systèmes d'observation "intelligents" réside dans le fait que, pour la deuxième catégorie, l'action est suggérée par le système mais l'opérateur a le libre arbitre de l'appliquer ou d'en appliquer une autre : ce sont des aides à la décision.

Le concepteur, face aux cahiers des charges de plus en plus lourds, aux fonctions complexes à réaliser, à l'intégration des systèmes informatisés, est amené à faire des choix aussi bien matériels que logiciels lors de la conception de ces systèmes d'observation intelligents.

Or dans le cadre des systèmes de contrôle/commande, des méthodes de conception ont été développées pouvant s'appliquer aux systèmes d'observation vu leurs différences minimales avec les systèmes de contrôle/commande.

C'est pourquoi dans la suite de ce travail, après avoir présenté quelques raisons globales sur la nécessité d'utiliser une méthodologie de conception, nous présentons quelques méthodes.

Il n'est pas dans notre intention d'en dresser une liste exhaustive mais plus d'essayer d'en cerner le principe ou le dénominateur commun.

II NECESSITE D'UNE METHODOLOGIE DE CONCEPTION

Sous le vocable *Informatique Industrielle*, les automaticiens se sont inspirés des outils utilisés par les informaticiens pour mettre en oeuvre avec efficacité le Logiciel des systèmes. De même, ils se sont intéressés aux techniques de la microélectronique afin de sélectionner la solution Matérielle, support d'exécution du Logiciel, répondant au mieux aux problèmes posés./CALV-83/

La conception de systèmes de contrôle/commande met en évidence les étapes successives suivantes à prendre en compte :

- 1 - Quelle description choisir ?
- 2 - Comment choisir et définir l'architecture fonctionnelle la mieux adaptée :
 - * Au niveau logiciel :
 - Comment décomposer la complexité du problème en sous-parties ?
 - Comment définir les interactions entre ces sous-parties ?

* Au niveau matériel :

- Quelle solution choisir ?
 - . Mono ou multi processeurs
 - . Centralisée ou répartie
 - . Structure étoile, ligne, circulaire...

3 - Quels sont les critères d'efficacité d'une solution ?

Le travail en équipe, souvent rendu nécessaire par la complexité du projet, nécessite de répartir et de décrire les tâches afin d'aboutir à un système répondant à la demande lors de l'intégration des sous-parties traitées par les différentes personnes.

Si pour des systèmes simples on peut se passer de méthodes de conception et aboutir à une solution plus ou moins satisfaisante, il n'en est pas de même pour les systèmes complexes où leur utilisation devient impérative face à l'ensemble des problèmes rencontrés.

Il est donc nécessaire de développer ou d'utiliser des outils permettant de diminuer les coûts de conception et d'arriver à des solutions fidèles aux besoins du demandeur.

De plus, si on s'intéresse au temps consacré aux différentes étapes de la réalisation d'un système, on aboutit au partage décrit Figure II.4. /ZELK-78/

Développement :

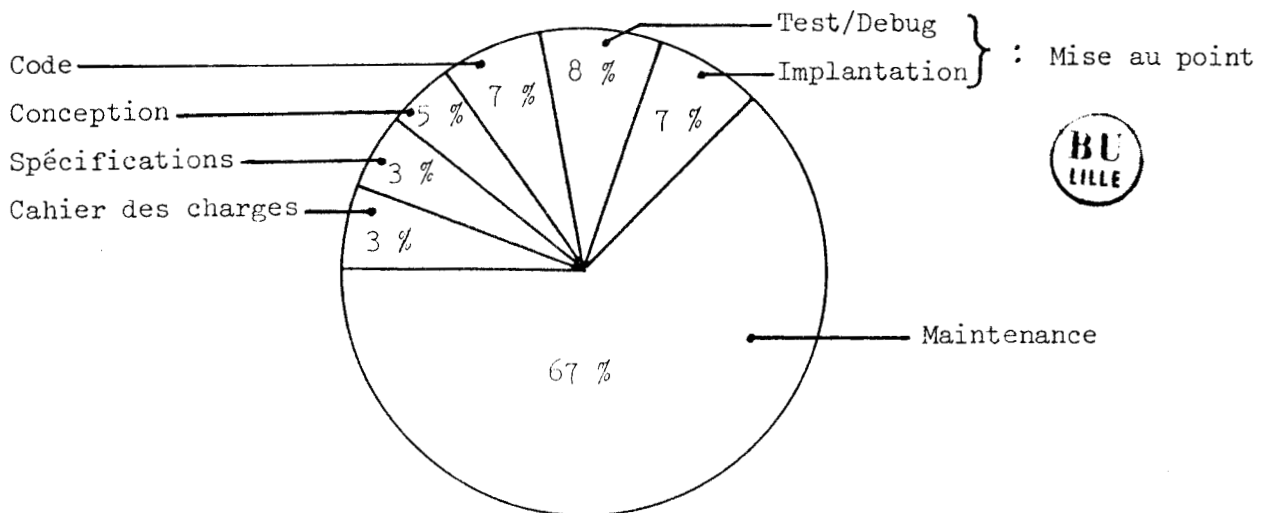


Figure II.4

C'est à dire que seulement 33 % du temps est consacré au développement du système et 67 % à la maintenance.

Et sur les 33 % du développement environ la moitié (15%) est attribuée à la mise au point (test, debug, implantation).

En définitive 18 % du temps est consacré :

- à la définition du cahier des charges
- à la définition des spécifications
- à la conception proprement dite
- et à l'écriture du logiciel.

Ce qui a permis à MYERS de dire /MYER-78/ :

"Nous essayons de résoudre un problème en expédiant la conception; si bien que la fin du projet sert à rechercher les erreurs faites parce que la conception a été expédiée..."

De même la politique, trop souvent adoptée, de dire "l'important c'est que ça marche" implique des systèmes peu flexibles et d'une maintenance peu aisée (67 % du coût...).

MEILLIR PAGE-JONES /MEIL-80/ exprime différentes raisons sur le fait que la maintenance est si coûteuse en décrivant les différentes étapes d'une modification à opérer sur un système.

Il faut :

- 1 - Comprendre comment le système fonctionne (ou pourquoi il ne fonctionne pas).
- 2 - Savoir où implanter la modification demandée.
- 3 - Déterminer son impact sur le reste du système.
- 4 - Concevoir la modification.
- 5 - Tester le système.
- 6 - Rédiger un document de suivi.

On conçoit aisément la difficulté rencontrée, si la conception a été "expédiée" et/ou si on ne dispose pas de documents décrivant le système existant.

En résumé, une méthode de conception doit permettre à moindre coût :

- de définir l'architecture fonctionnelle du système la mieux adaptée.

- de décrire et de valider les algorithmes à y implanter.
- d'aboutir à un système offrant la qualité de maintenabilité.
- d'envisager la possibilité de modifications, donc de prévoir l'évolution du système.

III ETAT DE L'ART OU SYNTHESE DES METHODES EXISTANTES

Il existe peu de procédures intégrant l'ensemble du processus de développement complet d'un projet en partant du cahier des charges pour aboutir à la réalisation logicielle et matérielle.

Cependant la réalisation d'un projet passe par différentes étapes (Figure II.5) de développement auxquelles peuvent être associées divers outils et/ou méthodes d'analyse.

III.1 CONCEPTION LOGICIELLE

La conception logicielle passe par différentes étapes, qui sont :

- L'analyse, la décomposition fonctionnelle.
- La définition des spécifications.
- L'écriture du code.
- La mise au point : test et intégration.
- La maintenance.

Bien que certains projets /DOD -82/, les Ateliers de Génie Logiciel (A.G.L.) /ADI -84/, et certaines méthodes /MEIL-80/ ont pour objectif de répondre à ces différentes étapes, on distingue en général deux catégories d'outils/méthodes liés : /KRAK-82/

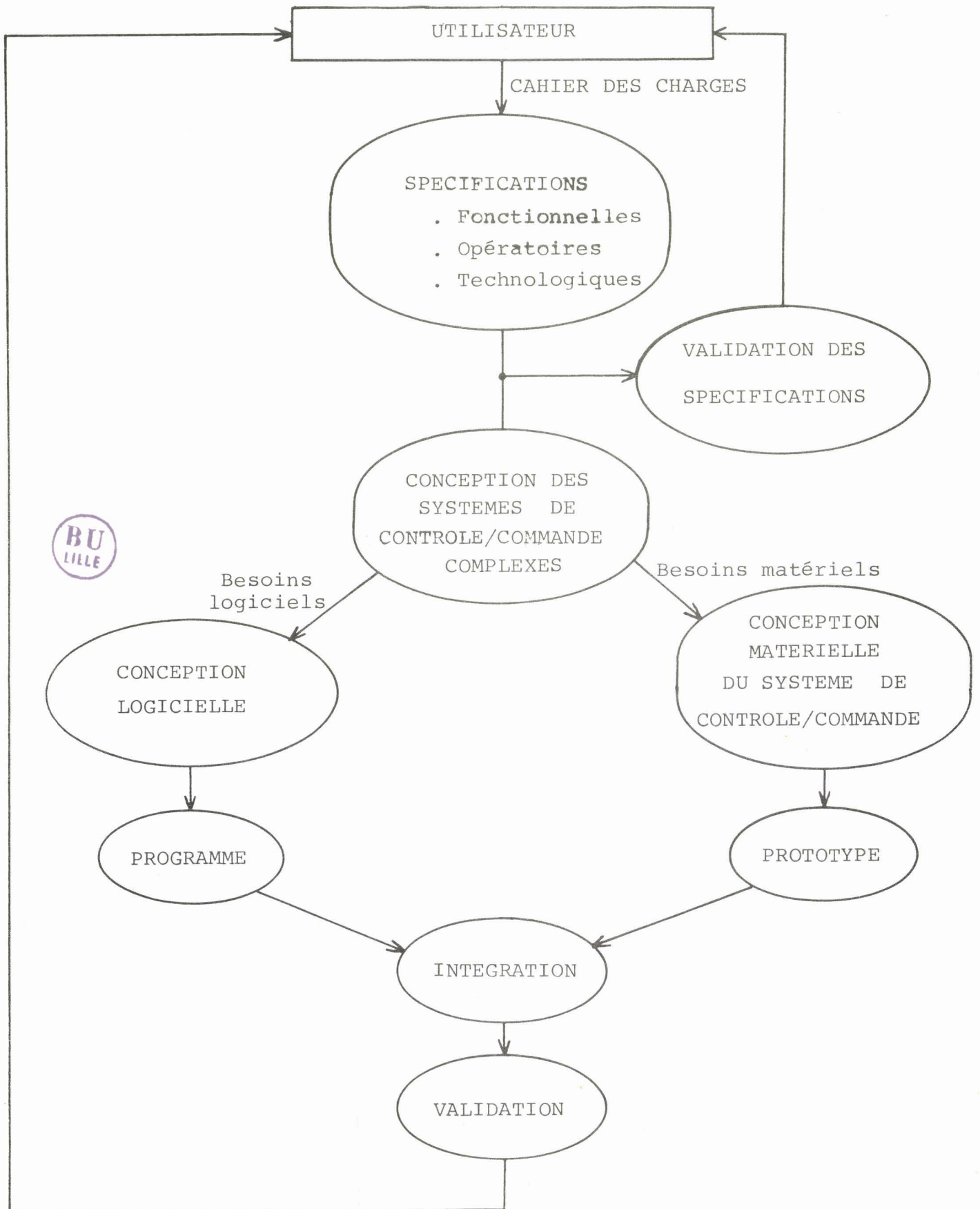
- Soit à la conception
- Soit au développement.

III.1.1 CONCEPTION

Lorsque le concepteur aborde la phase de conception il dispose de divers moyens :

- > Soit l'utilisation d'outils formels (tels les graphes fonctionnels : GRAFCET, Réseau de Pétri) ou d'outils semi-formels (tels les ordinogrammes).
- > Soit l'application de méthodes généralement semi-formelles (tels le GEMMA ou SADT).

Figure II.5



BU
LILLE

Les outils ne peuvent être considérés comme des méthodes au sens où ils servent à exprimer la solution, mais n'aident pas à sa détermination.

En effet les graphes fonctionnels sont des modes de représentation du cahier des charges, basés sur la notion d'étapes et de transitions /BRAR-80/, faciles à employer et fournissant une description suivant le déroulement naturel du processus et indépendant de la technologie employée.

C'est une des raisons pour laquelle la commission du groupe Systèmes Logiques de l'AFCEC a proposé le GRAFCET en tant qu'outil de description du cahier des charges d'un automatisme logique /GRAF-77/.

Le fonctionnement de l'automatisme peut être représenté par un ensemble :

- d'étapes auxquelles sont associées des actions
- de transitions auxquelles sont associées des réceptivités (condition logique autorisant le franchissement des transitions).
Un ensemble de règles d'évolution fixe les conditions de franchissement.
- de liaisons orientées reliant les étapes aux transitions et les transitions aux étapes.

De même que le GRAFCET, les Réseaux de Pétri permettent de simuler des systèmes à actions discrètes, ces actions étant séquentielles ou parallèles.

Les graphes fonctionnels sont suffisamment formels et bien connus du point de vue théorique pour permettre une analyse assez fine de la conception et servir à définir le comportement fonctionnel du processus.

Seulement ce sont des outils et non des méthodes. C'est pourquoi ils doivent être associés à des méthodes visant à décomposer le comportement du processus en tâches reliées entre elles et communiquant.

Ainsi le GEMMA /GEMM/ (Guide d'Etudes des Modes de Marches et Arrêts), méthode semi-formelle, constitue une liste des modes de fonctionnement permettant d'adjoindre au GRAFCET de fonctionnement normal, des éléments nécessaires à la gestion des modes de marches.

Un guide graphique facilite la recherche des spécifications en le représentant sous la forme d'un automate et en le structurant en trois zones :

- les états de fonctionnement
- les états d'arrêts de la partie opérative
- les états de défaillance de la partie opératoire.

La démarche proposée comporte deux temps :

- 1 - Le recensement des différents modes envisagés, et la mise en évidence des enchainements qui les relient.
- 2 - La transformation en un GRAFCET d'ordre supérieur ou l'intégration au GRAFCET de marche normale.

D'autre part, certaines méthodes visent à décomposer le comportement fonctionnel des systèmes et font intervenir les notions de tâches, d'événements ou de messages.

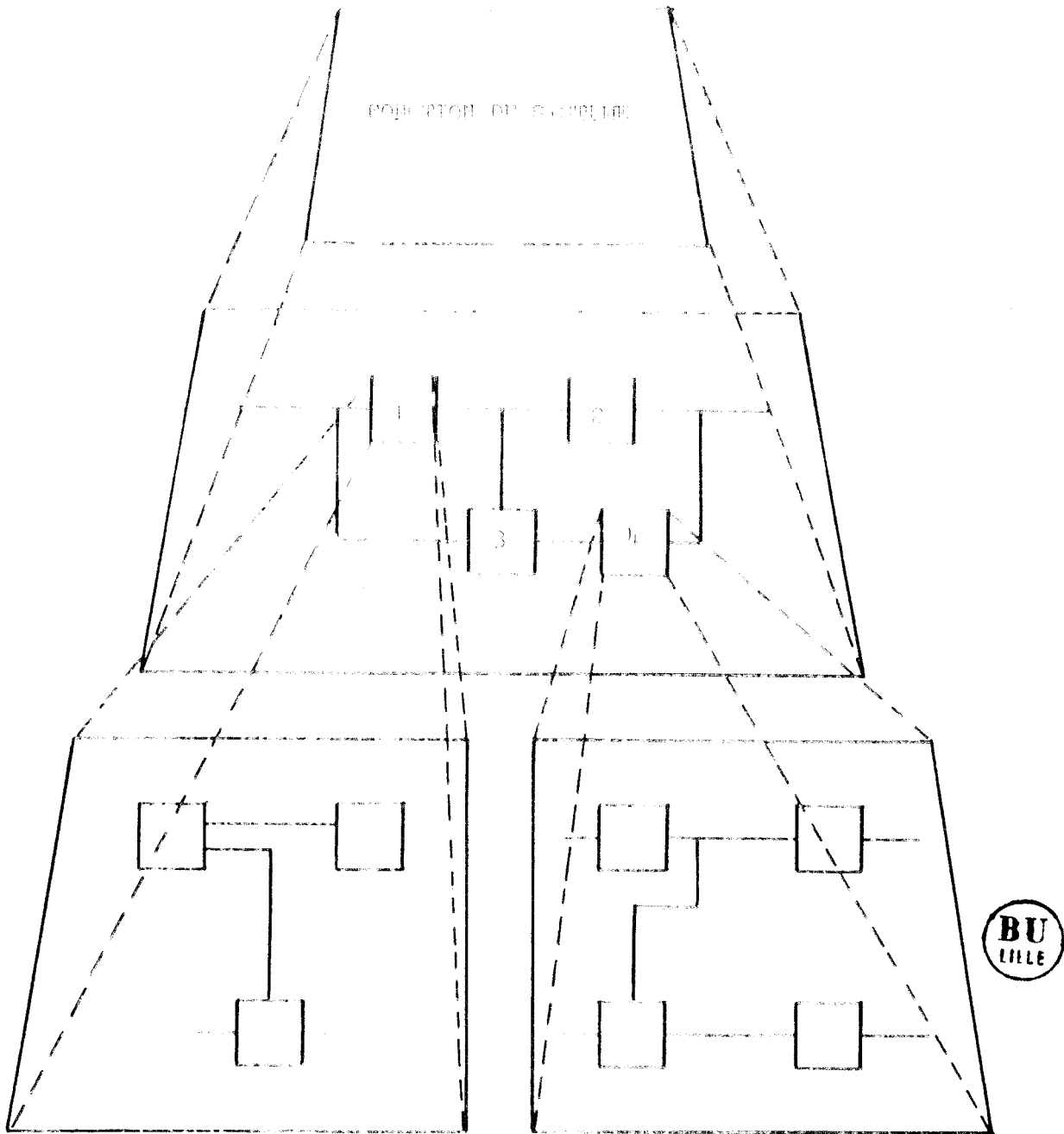
Elles sont en général basées sur :

- le concept de niveau
- la décomposition descendante (Top Down)
- la décomposition modulaire
- l'analyse structurée /YOUR-78/, /DEMA-78/.

SADT /CONN-80/ (Structured Analysis and Design Technique) est composé de modèles semi-formels pour réaliser l'analyse et l'étude des systèmes. C'est une méthode basée sur la décomposition en niveaux successifs d'un problème, fournissant une structure hiérarchique, structurée et modulaire.

Le principe de base de SADT est la décomposition descendante des activités sous forme de boîtes noires.

Son modèle est constitué par un ensemble organisé de diagrammes. Au niveau le plus général, un seul diagramme résume l'ensemble du sujet abordé et les diagrammes des niveaux inférieurs apportent un nombre limité de détails sur un sujet bien déterminé. Chacun des diagrammes s'imbriquent très exactement dans le modèle entier, précisant ainsi les relations entre ses composants et le reste du système (figure II.6) /LISS-83/.



Structure hiérarchique
des diagrammes SADT

Figure II.6

SADT permet de mettre en évidence la dualité entre les décompositions en activités et en données (*actigrammes*, *datagrammes*), ce qui permet de donner une représentation fidèle de l'aspect fonctionnel.

SADT n'est pas utilisé pour la conception détaillée dans la mesure où les constructions de base de la programmation structurée (*séquence*, *sélection*, *itération*) en ont volontairement été omises; c'est l'aboutissement d'études dirigées vers les premières étapes du cycle de vie du logiciel. /LISS-83/

Les méthodes classées en tant que "analyse modulaire et structurée" permettent de concevoir des programmes plus lisibles mais bien souvent limité aux descriptions séquentielles.

C'est pourquoi des langages de description (tel le langage ADA) et des représentations graphiques ont été utilisés permettant l'expression du comportement de la solution sous la forme d'activités parallèles.

III.1.2 DEVELOPPEMENT

Si les méthodes d' "Analyse modulaire et structurée" facilitent la conception de programme, elles offrent également des possibilités pour la mise au point.

En effet le programme conçu est clair et lisible et donc facilite la détection des anomalies. De plus la maintenance est rendue plus aisée par la modularité des systèmes : lors d'une modification, tout le système n'est plus à corriger et on peut connaître l'impact ou les "effets de bord" de cette modification.

Les documents élaborés lors de la conception du système peuvent donc servir d'outil de travail pour la mise au point et pour la maintenance.

D'où l'intérêt de s'interroger sur les qualités d'une bonne conception et l'élaboration d'un document qui reflète cette conception...

III.2 CONCEPTION MATERIELLE

Peu de méthodes sont proposées pour répondre au choix d'une architecture matérielle et bien souvent cette recherche reste intuitive et du domaine de l'expérience du concepteur.

Certaines publications présentent différentes structures (série, parallèles, étoiles...) avec leurs critiques tel /FATM-83/ ou l'intérêt des structures multiprocesseurs et des structures réparties /VERJ-83/; mais le plus souvent la conception des structures matérielles sont abordées dans des méthodes de conception globale : c'est à dire partant du cahier des charges et aboutissant à la réalisation finale du logiciel, du matériel, et de l'intégration de l'un sur l'autre.

La conception d'une structure multiprocesseurs se rapporte essentiellement aux méthodes d'échanges d'informations qui ont pour fonction la mise en oeuvre :

- des communications interprocesseur
- de l'architecture des interconnexions.

Les communications interprocesseur se font par l'intermédiaire de transferts de données, réalisés :

- soit via une mémoire commune (structure centralisée)
- soit via un bus (structure répartie).

Dans le cas d'une mémoire commune, les éléments ne sont pas directement reliés entre eux et n'ont donc pas d'accès direct, par contre une structure de bus crée une voie de communication entre les éléments.

Dans les architectures d'interconnexions, quatre structures de bases sont distinguées :

- architecture circulaire
- architecture avec bus commun
- architecture étoile
- architecture pleinement connectée.

Les autres structures pouvant être considérées comme des combinaisons ou des variantes de ces quatre structures de base.

I) Structure circulaire

Les données passent par chaque processeur en suivant une voie circulaire. (Figure II.7)

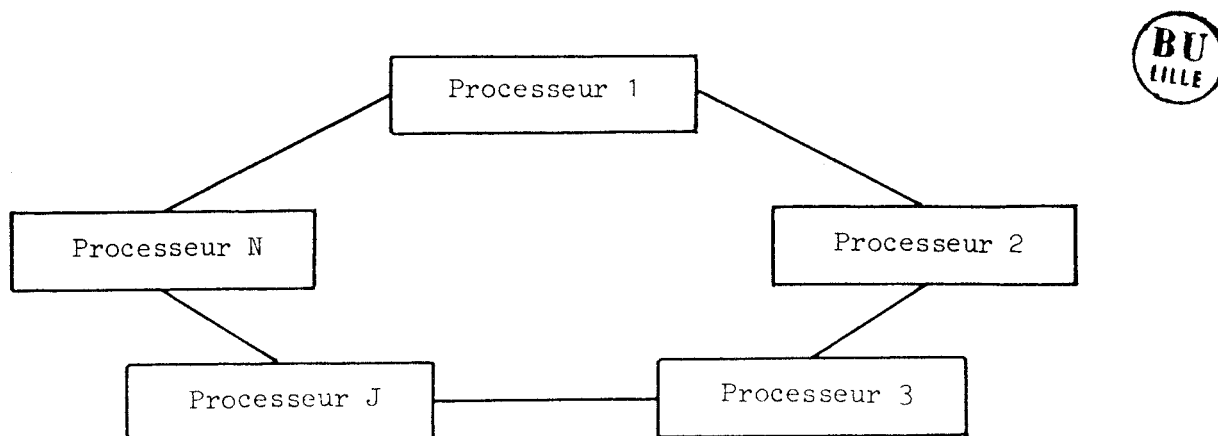


Figure II.7

Cette structure est bien adaptée pour assurer une transmission complète.

Lorsqu'un processeur envoie un message, il parcourt toute la chaîne et la vérification d'une transmission correcte est effectuée en comparant le message envoyée avec celui reçu.

Par contre cette structure est peu flexible : la chaîne doit être coupée lorsqu'on désire rajoutée un élément, et si un processeur tombe en panne, ou si une liaison entre deux processeurs est coupée, l'échange d'informations ne peut plus circuler et toute la structure en pâtit.

II) Structure avec un bus commun

Dans cette structure, les données sont transmises par l'intermédiaire d'un bus commun. (Figure II.8)

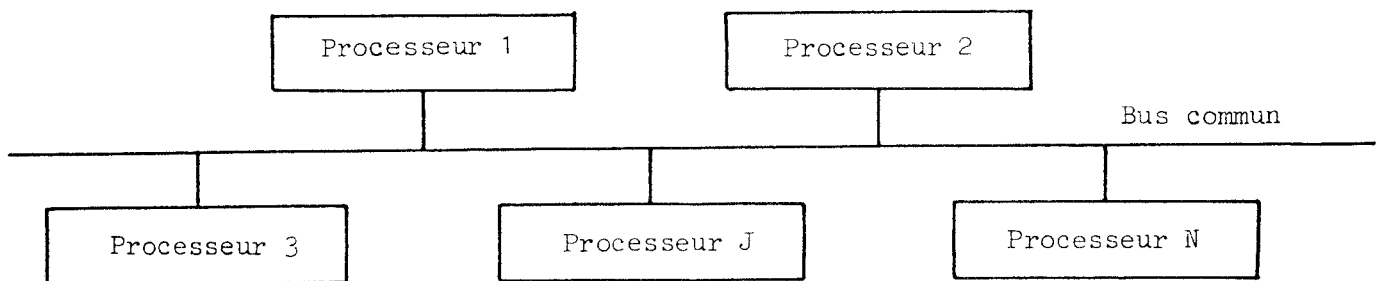


Figure II.8

Cette structure possède l'avantage de faciliter la connexion ou la déconnexion d'un élément sans interagir sur l'ensemble du système et sans devoir couper le bus.

De plus un processeur en panne ne pénalise pas tout le système.

L'inconvénient de cette structure est qu'une panne sur le bus bloque évidemment tout le système.

Comme pour la structure circulaire, le contrôle est décentralisé.

III) Structure centralisée

Cette structure est caractérisée par un contrôleur central relié à chaque processeur. (Figure II.9)

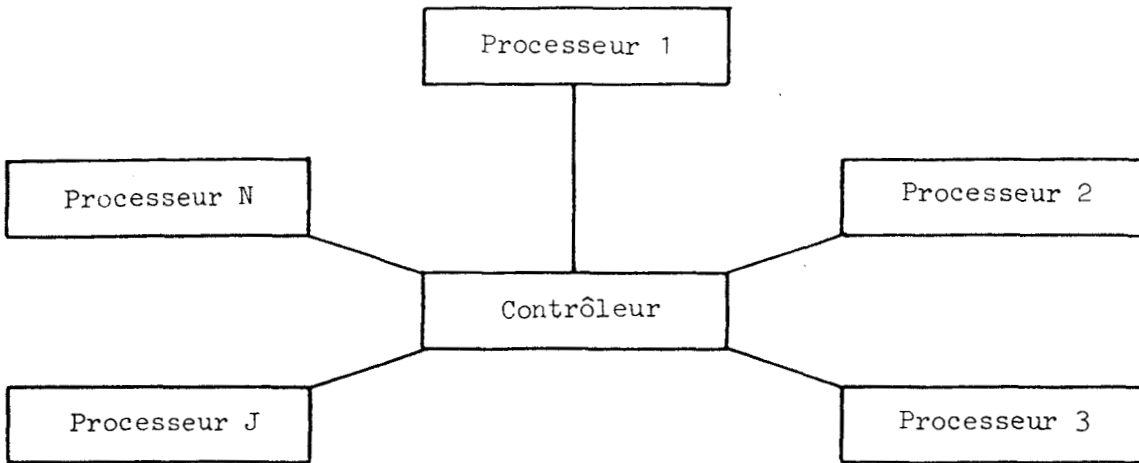


Figure II.9

Comme chaque processeur est relié par une branche séparée, le contrôleur n'est pas affecté si un processeur, ou si une liaison tombe en panne.

De même un élément peut être déconnecté ou connecté sans affecter tout le système.

L'inconvénient de cette structure réside dans la centralisation du contrôleur : si il a une défaillance tout le système est bloqué.

IV) Structure pleinement connectée



Dans cette structure tous les processeurs sont reliés entre eux (Figure II.10).

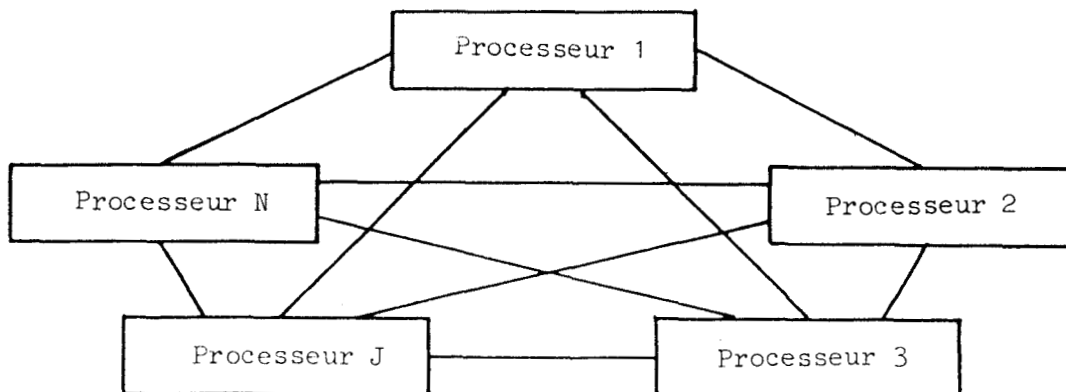


Figure II.10

Dans le cas d'une défaillance d'une liaison entre deux processeurs, un troisième processeur peut servir d'intermédiaire pour rétablir la communication.

De même, la défaillance d'un processeur n'affecte pas les autres communications et donc ne pénalise pas le système global.

Ces différents avantages se soldent malgré tout par un inconvénient de taille :

- le nombre important de liaisons.

Pour un système à N processeur il faut $\frac{N(N-1)}{2}$ liaisons.

Les systèmes multitâches et/ou multiprocesseurs impliquent des problèmes de tâches concourantes ou parallèles, de liaisons entre tâches, ne rendant pas aisé une représentation rigoureuse.

C'est pourquoi des outils formels ont été développés dans le but d'en faciliter la description.

Ainsi les Réseaux de Pétri, indépendants de tout langage, permettent d'exprimer le comportement d'architecture de machines et de protocoles de communication /DIAZ-82/, /ROUC-83/.

III.3 METHODES GLOBALES

Si il existe des outils/méthodes répondant partiellement au problème de conception logicielle et matérielle, peu de méthodes ont été développées couvrant l'ensemble du cycle de la réalisation des systèmes de contrôle/commande complexes /LONC-83/.

CALVEZ /CALV-82/, /CALV-83/ a développé une méthodologie de conception pour les applications de commandes des processus complexes en temps réel.

Il propose une approche constructive en cinq niveaux :

- 1 - Niveau de la description externe :
correspondant aux spécifications de l'application.
- 2 - Niveau de la description fonctionnelle :
correspondant à la topologie de l'application
(constituants + interconnexions)
- 3 - Niveau de la description opératoire :
obtenue en ajoutant au niveau précédent la
description algorithmique de chaque fonction.

- 4 - Niveau de la description exécutive :
correspondant à la structure exécutive et à l'intégration de la description opératoire.
- 5 - Niveau de la réalisation :
c'est à dire la description complète de la réalisation (matériel + logiciel)

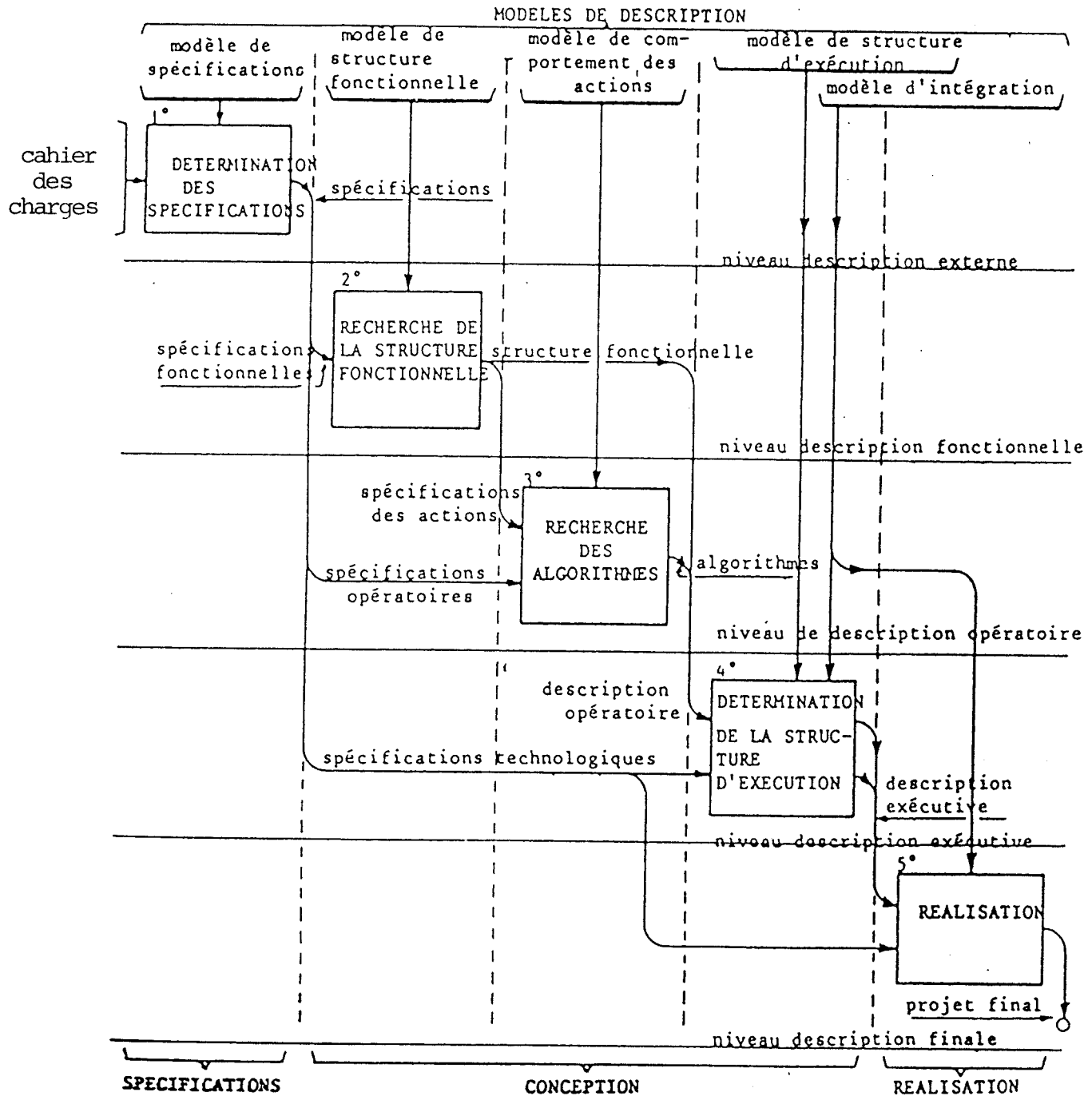


Figure II.11

Un tel modèle permet de définir une méthodologie en cinq étapes :

- * la définition des spécifications de l'application.
- * la définition de la structure fonctionnelle.
- * la définition de la structure opératoire.
- * la définition de la structure exécutive.
- * la définition de la réalisation.

La figure II.11 indique le fil conducteur de la méthode où chaque étape est représentée par un rectangle avec :

- à gauche : les spécifications à utiliser
- en haut : le modèle de description de l'étape à respecter.
- à droite : le résultat conforme au modèle et aux spécifications.

Ainsi la méthodologie consiste en une démarche par étape qui procède des spécifications vers la réalisation.

IV CONCLUSION

Le fait que la méthode développée par CALVEZ soit globale, dans le sens où elle tient compte de tous les problèmes rencontrés lors de la conception d'un système complexe en partant du cahier des charges vers la réalisation, nous a semblé intéressant.

C'est pourquoi nous avons décidé de l'utiliser pour le développement de notre application.

Le chapitre suivant décrit, étape par étape, la méthode suscitée, prenant pour support la conception d'un système d'observation et de prise de décision automatique, constitué par une station de surveillance individuelle de malade en réanimation.

CHAPITRE III

CONCEPTION D'UN SYSTEME DE SURVEILLANCE

INDIVIDUELLE DE MALADES EN REANIMATION

APPLICATION DE LA METHODE DEVELOPEE PAR CALVEZ

CHAPITRE III :
CONCEPTION D'UN SYSTEME DE SURVEILLANCE
INDIVIDUELLE DE MALADES EN REANIMATION
APPLICATION DE LA METHODE DEVELOPPEE PAR CALVEZ

<u>I BUT DU PROJET</u>	p.	1
<u>II PREMIERE ETAPE : DEFINITION DES SPECIFICATIONS</u>	p.	3
<u>II.1 MODELE DE CALVEZ</u>	p.	3
<u>II.2 APPLICATION AU PROJET</u>	p.	4
<u>II.2.1 LES SPECIFICATIONS FONCTIONNELLES</u>	p.	4
* <i>Traitement local</i>	p.	5
* <i>Traitement global</i>	p.	7
<u>II.2.2 LES SPECIFICATIONS OPERATOIRES</u>	p.	11
<u>II.2.3 LES SPECIFICATIONS TECHNOLOGIQUES</u>	p.	13
<u>III DEUXIEME ETAPE : LA STRUCTURE FONCTIONNELLE</u>	p.	14
<u>III.1 MODELE DE CALVEZ</u>	p.	14
<u>III.1.1 MODELE DE LA STRUCTURE FONCTIONNELLE</u>	p.	14
* <i>Les règles de description</i>	p.	14
* <i>Les règles de comportement</i>	p.	14
* <i>Règles pour les actions</i>	p.	14
* <i>Règles pour les variables partagées</i>	p.	16
* <i>Règles pour les ports</i>	p.	16
<u>III.1.2 DETERMINATION DE LA STRUCTURE</u> <u>FONCTIONNELLE</u>	p.	16
<u>III.1.3 CARACTERISTIQUES D'UNE STRUCTURE</u> <u>FONCTIONNELLE</u>	p.	17
<u>III.1.4 CONCLUSION</u>	p.	18
<u>III.2 APPLICATION AU PROJET</u>	p.	18

<u>IV TROISIEME ETAPE : LA DESCRIPTION OPERATOIRE</u>	p. 32
<u>IV.1 MODELE DE CALVEZ</u>	p. 32
<u>IV.1.1 MODELE DE COMPORTEMENT DES ACTIONS</u>	p. 32
* <i>Règles de description et de comportement</i>	p. 32
* <i>Règles d'échanges interaction</i>	p. 33
<u>IV.1.2 CARACTERISTIQUES DE LA DESCRIPTION</u> <u>OPERATOIRE</u>	p. 33
<u>IV.1.3 DETERMINATION DES ALGORITHMES</u>	p. 34
<u>IV.1.4 IMPORTANCE DU LANGAGE DE DESCRIPTION</u>	p. 34
<u>IV.2 APPLICATION AU PROJET</u>	p. 34
* <i>Exemples d'actions permanentes</i>	p. 35
* <i>Exemples d'actions temporaires</i>	p. 35
<u>V QUATRIEME ETAPE : LA STRUCTURE EXECUTIVE</u>	p. 37
<u>V.1 MODELE DE CALVEZ</u>	p. 37
<u>V.1.1 MODELE DE LA STRUCTURE EXECUTIVE</u>	p. 37
* <i>Règles pour les processeurs</i>	p. 39
* <i>Règles pour les mémoires</i>	p. 39
* <i>Règles pour les liaisons</i>	p. 39
<u>V.1.2 MODELE D'INTEGRATION</u>	p. 39
* <i>Correspondance entre les éléments</i>	p. 40
* <i>Correspondance entre structures</i>	p. 40
<u>V.1.3 CARACTERISTIQUES DE LA SOLUTION</u>	p. 40
<u>V.1.4 RECHERCHE DE LA STRUCTURE EXECUTIVE</u>	p. 41
* <i>Fréquence maximale d'activation</i>	p. 41
* <i>Temps d'exécution maximum</i>	p. 41
<u>V.1.5 POSSIBILITES DE CONSTRUCTION</u>	p. 42
* <i>Regroupement de structures prédéfinies</i>	p. 42
* <i>Construction par raffinement</i>	p. 42
* <i>Construction par abstraction</i>	p. 42
<u>V.1.6 CRITERE DE CONSTRUCTION</u>	p. 42
* <i>La modularité</i>	p. 42
* <i>La minimisation du coût</i>	p. 43
* <i>Sûreté de fonctionnement</i>	p. 43
<u>V.2 APPLICATION AU PROJET</u>	p. 43

VI DERNIERE ETAPE : LA REALISATION

VI.1 MODELE DE CALVEZ.....p. 47
VI.1.1 DEVELOPPEMENT PARALLELE DU MATERIEL
ET DU LOGICIEL.....p. 47
VI.1.2 REALISATION MATERIELLE.....p. 47
VI.1.3 REALISATION LOGICIELLE.....p. 47
VI.1.4 INTEGRATION.....p. 49

VI.2 APPLICATION AU PROJET.....p. 49

VII CONCLUSION.....p. 50

I BUT DU PROJET

L'objectif de ce travail est la réalisation d'un système informatisé individuel de surveillance et de gestion de données de malades en réanimation en vue d'une aide au diagnostic. /DOMI-84/, /COUV-85/

Actuellement, l'équipement d'une chambre de réanimation suppose l'acquisition d'une série de modules électroniques de surveillance à visée cardiovasculaire et respiratoire.

La redondance du matériel tant au niveau des fonctions qu'à celui des alarmes engendre des difficultés, non pas dans l'utilisation des appareils pris isolément, mais plutôt dans l'utilisation globale de l'ensemble des appareils.

On aboutit ainsi à des difficultés d'analyse par le personnel et à des prix de revient très élevés.

L'idée est de concevoir un module unique et individuel intégrant les principales fonctions de surveillance pour apporter une plus grande cohérence, donc une plus grande efficacité dans l'utilisation des appareils par le personnel soignant.

Plus précisément, il s'agit d'analyser en temps réel plusieurs fonctions physiologiques :

- * La fonction cardiovasculaire par l'analyse du signal de l'électrocardiogramme (ECG) et de la tension artérielle (TA) ou de la pression artérielle pulmonaire (PAP).
- * La fonction respiratoire par la mesure des échanges gazeux et de la mécanique ventilatoire (signaux issus d'un respirateur : CO₂, Pression, Volume).

Cette liste n'est pas exhaustive, car la structure doit offrir la possibilité d'extension à l'analyse d'autres signaux/fonctions.

Pour chaque signal, il est à prévoir les différentes étapes du traitement de signal, c'est à dire :

- * l'acquisition du signal
- * sa mise en forme
- * le traitement associé à chaque signal

Les résultats issus de chaque analyse de ces signaux serviront de données d'entrée à un traitement global fournissant une synthèse ou un bilan physiologique du patient.

Ce traitement permet la corrélation entre les signaux; ainsi la génération des alarmes peut se faire, non seulement sur la détection d'une anomalie sur un signal, mais aussi sur la présence d'anomalies au niveau d'une ou plusieurs fonctions.

Les résultats de ces traitements seront visualisés sur une console graphique couleur. Il est donc important d'utiliser pleinement les capacités offertes par le graphisme afin d'offrir à l'utilisateur une visualisation synthétique et agréable du bilan :

Une bonne représentation graphique permet au personnel soignant d'apprécier et d'intégrer plus rapidement les résultats qu'une suite impressionnante de chiffres.

De même la conduite de l'appareil, grâce à un clavier, ne doit pas donner l'impression à l'utilisateur, de faire de l'informatique.

Il faut donc prévoir une utilisation de l'appareil en mode "Presse-Bouton".

Cette station de surveillance sert de générateur de faits cliniques à un système d'aide au diagnostic qui lui sera connecté afin d'offrir au personnel soignant un outil efficace d'aide à l'examen clinique.

Ce couplage apporte une dimension importante au système de monitoring dans le sens où il permet une surveillance et un suivi permanent du malade en réanimation en proposant à l'utilisateur une évaluation de l'état du patient /COUV-85/.

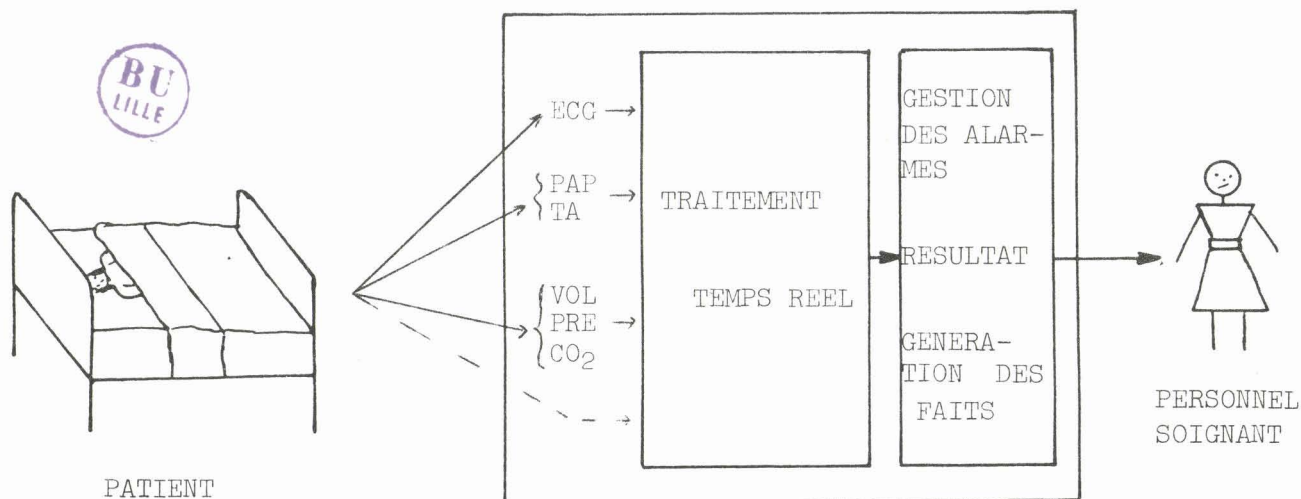


Figure III.1

Au regard des différents signaux à traiter en temps réel, des différentes fonctions à réaliser, ce projet s'intègre parfaitement dans la classe des systèmes d'observation complexes et nécessite ainsi l'utilisation d'une méthode de conception.

Les paragraphes suivants présentent les différentes étapes de conception de la méthode développée par CALVEZ, /CALV-82/, /CALV-83/ et l'application de chaque étape au système de surveillance présenté ci-dessus illustrera, en tant qu'exemple, cette méthode.

Chacun des paragraphes sera décomposé en deux parties :

- la description du modèle de l'étape de conception
- l'application au projet.

II PREMIERE ETAPE : DEFINITION DES SPECIFICATIONS

II.1 MODELE DE CALVEZ

Cette première étape définit l'ensemble des spécifications, c'est à dire les données ou contraintes nécessaires et suffisantes pour la réalisation de l'application.

Elles correspondent à la description externe du problème et doivent décrire ce que fait un système (*QUOI*) et non comment il le fait (*COMMENT*).

On trouvera dans ces spécifications la description de toutes les contraintes devant intervenir sur le système et les réponses aux questions spécifiques que le concepteur rencontrera.

Elles doivent permettre de faciliter les étapes de conception et servent donc d'interface entre demandeur et concepteur.

C'est pourquoi CALVEZ les classe en trois catégories, chacune intervenant à une étape précise de la conception (figure II.13). :

- (1) Les spécifications fonctionnelles :
Elles expriment l'ensemble des fonctions que doit réaliser le système et le comportement du processus à chacune d'elle.

Elles sont utilisées pour l'étape de la description fonctionnelle.

- (2) Les spécifications opératoires :
- Elles concernent les caractéristiques des différentes fonctions. Ce sont des renseignements qui peuvent faciliter la réalisation des fonctions (méthode de calcul, algorithme...) ou qui expriment des conditions de fonctionnement (performances à respecter, sécurité...).

Elles servent de données pour la description opératoire.

- (3) Les spécifications technologiques :
- Elles expriment les contraintes imposées au système de commande face à son environnement.
Ce peut être :
- * des contraintes technologiques :
imposées par le processus (par exemple interface système - processus).
 - * des contraintes de fabrication :
choix des composants
 - * des contraintes liées à l'environnement :
(par exemple milieu hostile).

Elles sont utilisées durant les phases de détermination de la structure exécutive et de la réalisation.

II.2 APPLICATION AU PROJET

II.2.1 LES SPECIFICATIONS FONCTIONNELLES

Le système de surveillance doit permettre l'acquisition et l'analyse automatique en temps réel des fonctions physiologiques :

- Cardiovasculaire
Par l'intermédiaire de l'électrocardiogramme (ECG) et de la tension artérielle (TA) ou de la pression artérielle pulmonaire (PAP).
- Respiratoire
Par la mesure des échanges gazeux et de la mécanique ventilatoire (signaux : CO₂, Volume, Pression).

Le traitement de l'ensemble de ces fonctions comporte deux étapes essentielles :

--> un traitement propre à chaque signal que l'on nommera "traitement local".

--> un traitement, dit "global", analysant l'ensemble des résultats issus de ces traitements locaux, (cf. figure III.2).

* Traitement local

Le traitement local est composé de différentes étapes :

(1) L'acquisition du signal :
--> pour l'ECG, cette phase est réalisée par l'implantation d'électrodes de surfaces. Suivant leur position, elles définissent ce qu'on appelle des dérivations. Dans le cadre de ce projet, on s'intéresse à la dérivation standard D1.

--> dans le cas de l'analyse de la PAP un cathétérisme flotté fournit le signal et dans celui de la TA on utilise une sonde à ultra son.

--> l'analyse de la fonction respiratoire nécessite le traitement de trois signaux fournis par un ventilateur respiratoire :

* le volume de gaz expiré (VOL)

Ce signal est fourni par un débitmètre (pneumotachographe de FLEISH) placé en bout de tubulure respiratoire.

* la concentration instantanée en CO₂ du gaz expiré relevé à la bouche du patient.

Un analyseur de CO₂ à infrarouge délivre ce signal (capnigraphe de BECKMAN LB2)

* une mesure de pression (PRE)

Le capteur utilisé se trouve au niveau du respirateur.

(2) L'amplification

Les signaux physiologiques sont en général de faible valeur (par exemple pour l'ECG de quelque microvolts); il est donc nécessaire de les amplifier afin de les amener à une valeur sur laquelle le traitement de signal pourra être opéré.

(3) Le filtrage

Des perturbations externes peuvent affecter les signaux de diverses manières.

Celles ci peuvent amener les traitements vers une confusion possible entre ces perturbations aléatoires et une pathologie difficilement caractérisable.

Pour l'ECG, les perturbations à prendre en compte sont

* modulation de l'ECG par la respiration

* toussotements du malade

* défauts d'électrodes (débranchement de l'électrode, ou décollements passagers...)

* bruits... (50 Hz, parasites...)

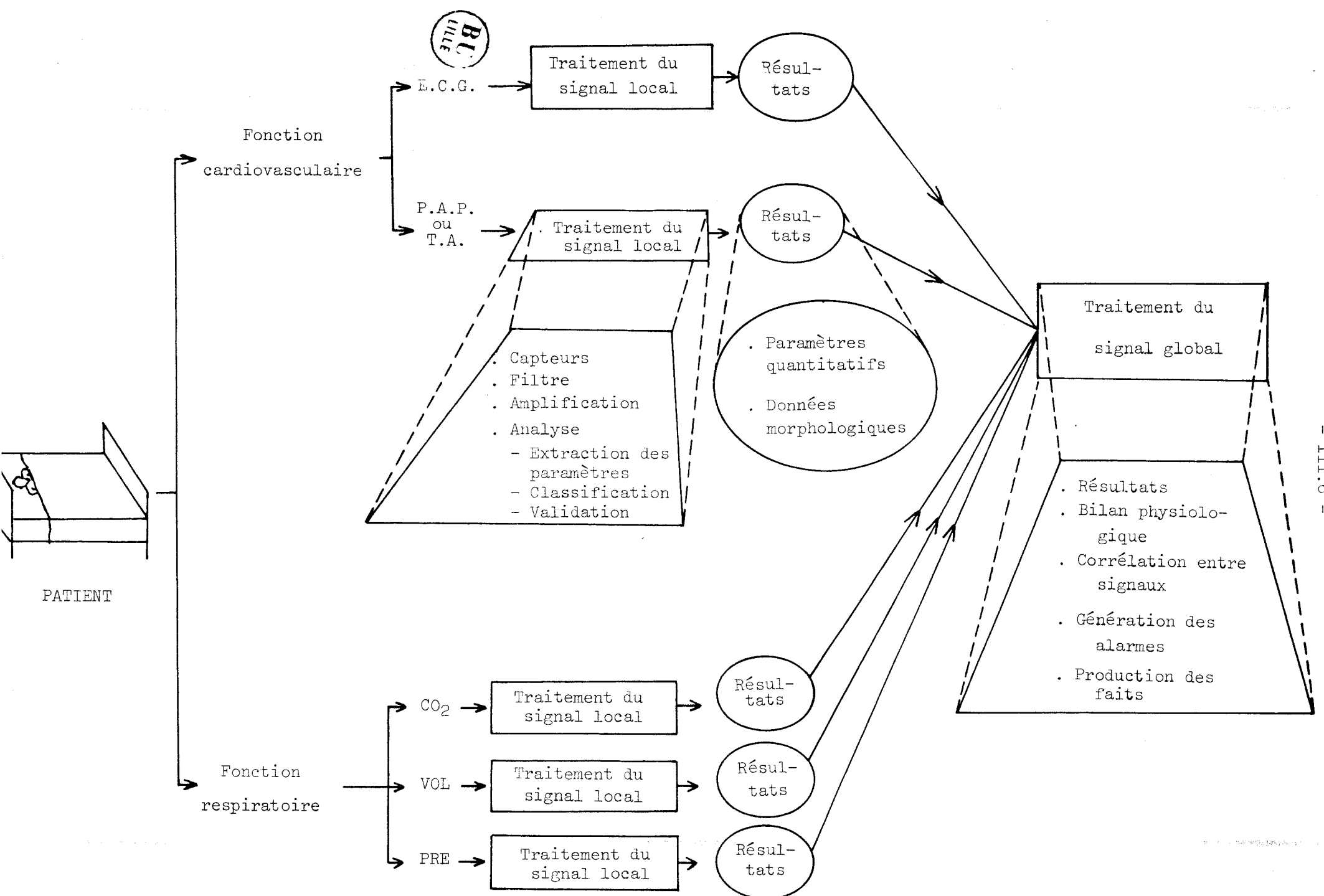


Figure III.2

- (4) La conversion analogique numérique (CAN)
Cette conversion est nécessaire afin que les signaux puissent être appréhendés par le système informatique. Un convertisseur analogique numérique est défini, en partie, par la fréquence d'échantillonnage :
- * pour l'ECG elle est de 100 Hz
 - * pour la PAP et les signaux respiratoires une fréquence de 50 Hz est suffisante.

- (5) Analyse du signal
Chaque traitement local est chargé de détecter les caractéristiques du signal qui lui est affecté. Il doit extraire des paramètres quantitatifs mais aussi des données relatives à la morphologie du signal.

Par exemple le traitement de l'ECG est chargé de détecter des faits relatifs à la fréquence cardiaque :

- * Rythme normal
- * Bradycardie
- * Tachycardie
- * Indéterminé (défauts d'électrode, parasites, ou autres ...)

et des faits relatifs à la morphologie du signal :

- * Rythme régulier sinusal
- * Arythmie à complexe fin
- * Rythme régulier à complexe large
- * Arythmie à complexe large unidirectionnel
- * Arythmie à complexe large polymorphe
- * Indéterminé (défauts d'électrode, parasites, ou autres ...).

En réanimation, ce sont essentiellement les anomalies du rythme qui intéressent les médecins réanimateurs pour l'ECG /LECE-83/.

Chaque analyse de signal peut considérer celui ci comme un générateur d'événements afin d'élaborer les deux informations suivantes :

- * Un vecteur de paramètres caractérisant l'événement
- * Le résultat d'une classification affectant l'événement à une situation.

* Traitement global/VASS-82/

Le traitement global est chargé de collecter et d'analyser les résultats issus des traitements propre à chaque signal afin de permettre :

- (1) de délivrer sur ordre de l'opérateur une image synthétique du bilan physiologique du patient.

Un tel bilan doit permettre de réaliser deux types d'observation :

- * une observation instantanée de l'état physiologique du patient
- * une observation de l'état physiologique antérieur.

On admet, en accord avec les médecins, que la largeur minimale de la fenêtre d'observation est de un quart d'heure.

Cette constatation conduit chaque traitement local à mémoriser par quart d'heure les résultats qu'il émet.

De plus, une mémorisation du passé physiologique, par tranche d'un quart d'heure, des différents traitements locaux, devient nécessaire pour établir de tels bilans. On limite à 24 heures cette mémorisation.

- (2) à l'opérateur d'établir d'éventuelles corrélations entre les signaux.

Pour cela il faut lui offrir la possibilité de visualiser simultanément les différentes situations traversées par chacun des signaux.

- (3) de générer des alarmes.

L'alarme dans ce cas n'est pas forcément conçue comme le résultat de l'examen d'un seul signal, mais aussi comme le résultat de la scrutation de l'ensemble des fonctions physiologiques (ou signaux).

Cette possibilité permet en outre de définir plusieurs types d'alarmes et même la programmation (par l'utilisateur) d'une gradation dans la gravité des alarmes.

- (4) de produire des faits

Ce système de surveillance doit être vu comme un générateur de faits alimentant un système d'aide à la décision. (cf. deuxième partie).

Ce système de surveillance ajoute donc à la notion de monitoring les notions de bilan physiologique et d'aide à l'examen clinique.

Ceci implique la nécessité de donner une grande importance à l'interactivité et à la visualisation des résultats au niveau de la console de pilotage.

En effet, dans un tel système et dans un tel contexte, il serait vain de vouloir dresser une liste impressionnante de

résultats sous forme de chiffres, il est préférable de concevoir des représentations synthétisant les résultats afin que le personnel soignant se rende compte facilement et rapidement de l'état du patient ou de son évolution par un regard sur des graphes (ou schémas) clairs et significatifs.

L'histogramme, qui est un moyen efficace et accepté par un grand nombre d'utilisateurs de présentations de résultats, offre une visualisation représentative du bilan physiologique.

Par exemple : histogrammes représentant le nombre de cycles à complexe normal et le nombre de cycles à complexe large, pour l'ECG, dans une tranche de temps donnée. (figure III.3)

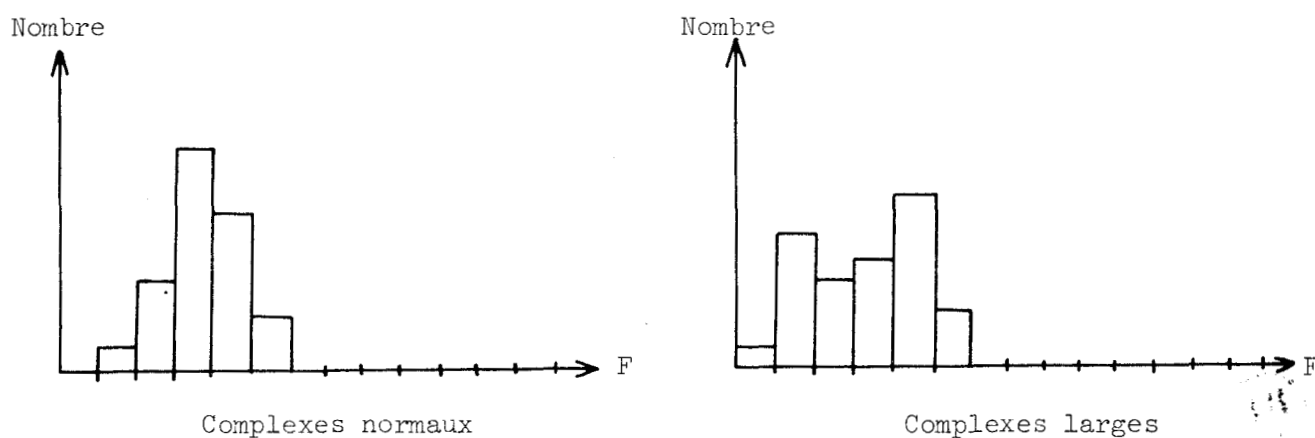


Figure III.3

Les corrélations entre signaux peuvent se faire au moyen de deux types de représentations dans une fourchette de temps désirée par l'utilisateur :

--> Une représentation temporelle : les chronogrammes
On visualise l'évolution dans le temps de plusieurs situations.

Par exemple pour l'ECG on peut visualiser les différents passages dans les trois situations :

- complexe normal : CN
- complexe fin : CF
- complexe large : CL

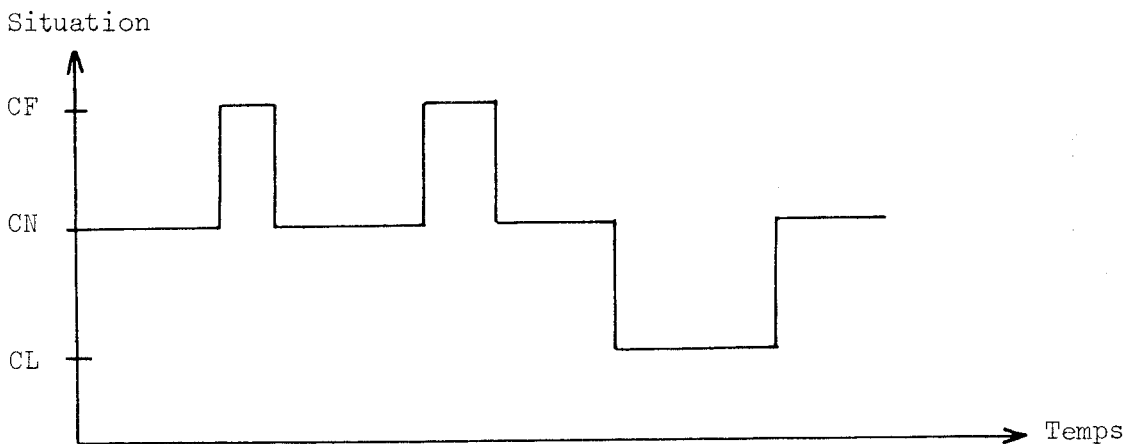
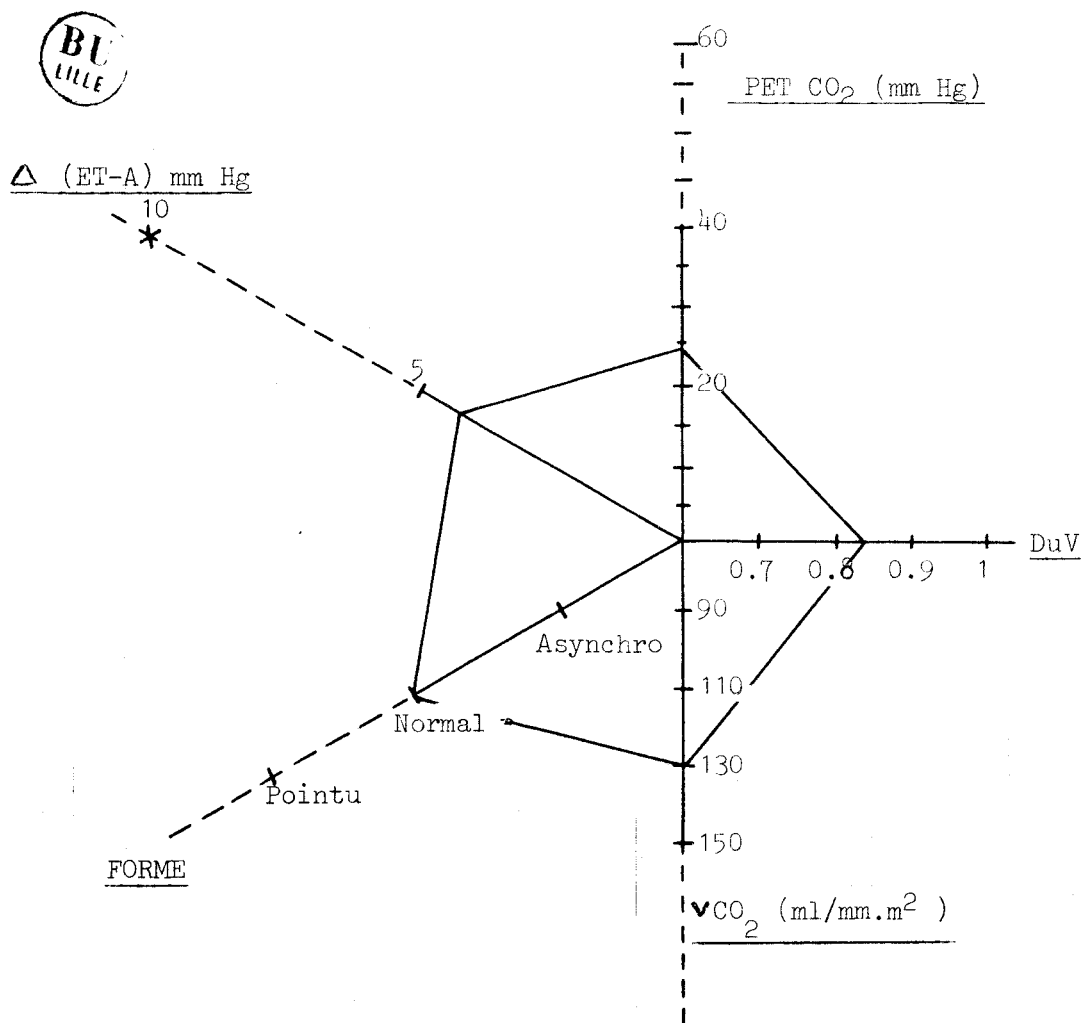


Figure III.4

--> Une représentation spatiale
 Un graphe est construit représentant à un instant donné les situations traversées par différents signaux.
 Ce graphe peut être rendu dynamique par la visualisation de son évolution dans le temps.

La figure III.5 montre différentes situations pour les échanges gazeux.



En outre il faut prévoir la visualisation simple des signaux tel un monitoring, mais à celle ci il est ajouté la possibilité de gel et de ZOOM de la trace.

L'accès aux différentes fonctions offertes à l'utilisateur, par l'intermédiaire d'un menu, se fera par une commande du type " Presse-Bouton " afin que les difficultés inhérentes au logiciel ne lui apparaissent pas.

Cet appareil doit être d'une grande souplesse opératoire et offrir un maximum de convivialité au risque de ne pas être utilisé dans toutes ses capacités.

II.2.2 LES SPECIFICATIONS OPERATOIRES

La diversité des formes appréhendées dans le contexte de la réanimation, ne permet pas de se référer à des standards habituels qui conditionnent des expériences plus favorables.

Dans ces conditions, il faut pallier ces inconvénients résultants de cette non standardisation par :

- * l'apprentissage des formes de référence pour chaque patient
- * la remise à jour permanente de ces formes de référence
- * l'introduction, au niveau du logiciel, de redondances permettant d'effectuer des recoupements et d'augmenter la fiabilité du système. (Utilisées surtout pour l'analyse de l'ECG) /LECE-83/

Dans le cas de la mécanique ventilatoire, les trois signaux étudiés ne sont pas synchronisés. Ceci est dû au fait que leur récupération s'effectue à divers niveaux de la tubulure du respirateur.

Or l'obtention du paramètre espace mort série (VDS) nécessite la construction de la courbe $CO_2 = f(\text{volume})$.

Ces deux signaux n'étant pas synchronisés, il y a donc lieu de reconstruire cette synchronisation.

Comme le début de cycle du signal CO_2 est difficilement caractérisable, le retard sera calculé à partir des valeurs maximales de fin de cycle (figure III.6).

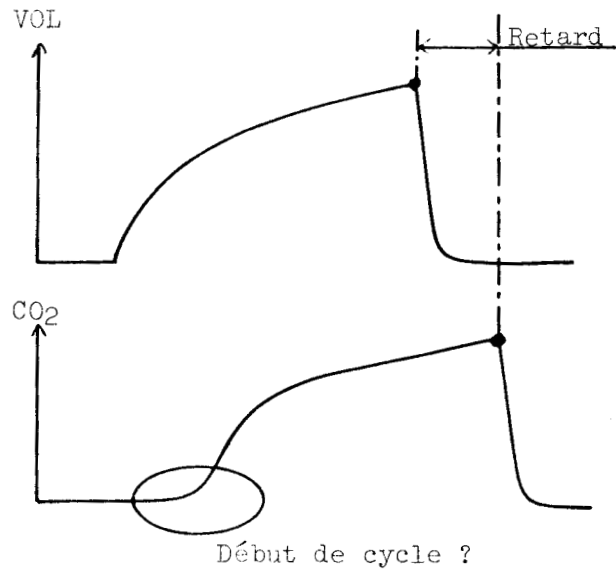


Figure III.6

Une fois cette étape de synchronisation réalisée, il convient de construire la courbe $CO_2 = f(\text{Volume})$.

VDS représente alors l'abscisse de la petite base d'un trapèze rectangle ayant la même aire que la surface expirée (figure III.7)

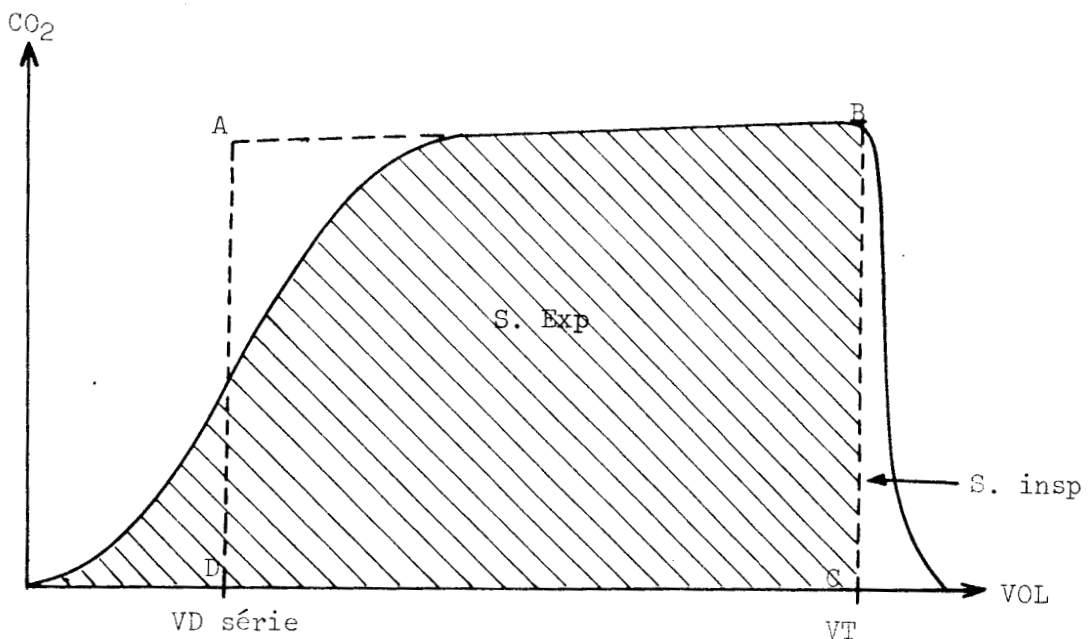


Figure III.7



Des études antérieures, /CHAM-79/ et /LECE-83/, ont permis de formuler des algorithmes de traitement, s'appliquant à ce projet, pour la fonction respiratoire et pour l'ECG.

Ces analyses sont à reprendre et à implanter sur le système de surveillance.

L'appareil doit présenter la qualité d'extensibilité, c'est à dire qu'il devra être possible d'adjoindre le traitement de nouveaux signaux sans pour cela remettre en cause toute la gestion de l'appareil.

Ceci implique que le système doit posséder une excellente modularité logicielle.

Cette caractéristique permettra, de plus, une modification aisée dans le cas où on cherche à apporter une amélioration aux traitements.

II.2.3 LES SPECIFICATIONS TECHNOLOGIQUES

Les cartes d'acquisition des signaux (amplification, filtrage, convertisseur analogique numérique) sont à réaliser.

Par contre les cartes processeurs sont imposées : on dispose de cartes INTEL au format Multibus SBC 80/05 (microprocesseur 8085, 512 octets de RAM, 4 ko de EPROM maximum) et de cartes SBC 80/24 (microprocesseur 8085, 4 ko de RAM, 32 ko de EPROM maximum).

La mémoire de masse est une mémoire à bulles (INTEL SBC 254 de 128 ko extensibles à 512 ko au format Multibus.

On dispose de plus d'une carte RAM INTEL SBC 032A (32 ko) au format Multibus.

Un rack INTEL 660 et les alimentations nécessaires recevront ces différentes cartes (8 au maximum).

La console de conduite est une Lyme 6000 avec un écran graphique et une liaison série RS232 (9600 Bauds)

Le désir de modularité se retrouve au niveau matériel : l'adjonction du traitement d'un nouveau signal doit se réaliser le plus simplement possible.

III DEUXIEME ETAPE : LA STRUCTURE FONCTIONNELLE

III.1 MODELE DE CALVEZ

Le but de cette étape est d'établir à partir des spécifications fonctionnelles une description de l'application sous la forme d'une structure fonctionnelle.

III.1.1 MODELE DE LA STRUCTURE FONCTIONNELLE

Le modèle fonctionnel utilisé pour cette étape décrit l'ensemble des règles devant être respectées par la structure fonctionnelle.

- Ces règles sont de deux types :
- > les règles de description d'une structure fonctionnelle
 - > les règles de comportement de chaque type de constituant.

* Les règles de description

La détermination d'une structure fonctionnelle aboutit à la définition de fonctions et de liens interfonctions.

Ces fonctions sont réalisées par des actions représentées par des rectangles et les trois types de liens sont réalisés (figure III.8) :

- > Soit par un événement pour la réalisation de synchronisation temporelle.
- > Soit par une variable partagée pour l'échange d'informations
- > Soit par un port jouant le rôle de dépôt de messages.





Lorsqu'une fonction n'assure d'échanges avec son environnement que par des messages, son évolution est asynchrone et elle sera qualifiée de module.

* Les règles de comportement

Ces règles fixent les spécifications du comportement externe de tous les constituants d'une structure fonctionnelle en faisant abstraction de leur réalisation. Le respect de ces règles entraînent une description utilisable comme spécifications pour les étapes futures.

* Règles pour les actions :

- > Une action dite élémentaire (c'est à dire non décomposable) se comportera comme une machine séquentielle cyclique.

Dénomination	Représentation	Signification
ACTION		FONCTION
EVENEMENT		SYNCHRONISATION
VARIABLE PARTAGEE		ECHANGE D'INFORMATIONS
PORT		DEPOT DE MESSAGES

L
I
E
N



Figure III.8

- > On distingue deux types d'actions :
 - les actions permanentes
Elles ne possèdent pas d'entrée d'événement ou de message.
 - les actions temporaires
Leur activité est sujette à la présence d'événements ou de messages.
- > Le temps d'exécution d'une action peut être considéré quelconque car il n'intervient pas à ce niveau de la description.
- > Une action durant son activité a accès à ses entrées et peut produire ses sorties.

** Règles pour les variables partagées*

- > Deux assignations simultanées donnent l'une ou l'autre des valeurs mais pas une combinaison des deux.
- > La consultation simultanément à une modification donne soit l'ancienne valeur, soit la nouvelle.

Ces règles expriment le respect des contraintes d'intégrité.

** Règles pour les ports*

- > Les ports servent de messages pour un transfert entre deux actions asynchrones (Producteur-Consommateur)
- > Le port est une file de mémorisation de messages qui n'ont pas été consommés par une action.

III.1.2 DETERMINATION DE LA STRUCTURE FONCTIONNELLE

Les règles et les conseils suivants facilitent la démarche pour arriver à une solution répondant au mieux aux spécifications.

- (1) L'application doit être considérée globalement, c'est à dire qu'il faut inclure dans la structure fonctionnelle le processus à commander, ceci afin de faciliter les tests et de décrire l'interface faisant partie de la solution à réaliser.
- (2) Afin que la structure fonctionnelle possède les propriétés du modèle fonctionnel il est indispensable que ce modèle soit respecté par celle-ci.
- (3) La sûreté de fonctionnement peut impliquer des redondances, mais la structure peut se déterminer, dans un premier temps, sans en tenir compte. Suivant les spécifications ces redondances pourront être introduites par la suite.
- (4) La structure fonctionnelle doit être construite en niveaux de description par des opérations de raffinement (construction descendante) ou d'abstraction (construction ascendante).
- (5) L'auteur décrit un processus de réflexion en trois phases pour chaque niveau :
 - * décomposition en niveaux :
 - travail d'analyse des spécifications
 - * construction de la structure :
 - à partir des actions et de leurs liens
 - travail de synthèse
 - * vérification de la structure

- (6) Le niveau de départ de la construction peut être quelconque étant donné les possibilités de construction ascendante ou descendante.
- (7) Le raffinement maximum de la décomposition doit aboutir à une description sous forme d'actions élémentaires (c'est à dire de fonctions ne faisant plus intervenir de parallélisme).
- (8) Un module doit englober toutes les actions à couplage serré (par exemple contraintes de temps).
- (9) Une action est définie comme étant une entité tel que les décisions nécessaires à sa réalisation se trouvent "cachées" pour l'extérieur (concept de *Boîte noire*).
- (10) La description peut être exprimée suivant une syntaxe précise permettant la vérification automatique.

III.1.3 CARACTERISTIQUES D'UNE STRUCTURE FONCTIONNELLE

Ces propriétés se déduisent du respect des règles du modèle et de l'approche considérée :

- (1) La décomposition de la structure fonctionnelle se poursuit tant qu'il existe des actions non élémentaires.
La solution finale exprimera donc le maximum de parallélisme pour la réalisation.
- (2) Le fait de considérer des temps d'exécution quelconque, et en particulier nuls, implique un synchronisme entre l'activité de toutes les actions liées à un même événement.
- (3) Toute structure fonctionnelle doit être bien définie, c'est à dire qu'elle est vivante et que chaque constituant est commandable et observable.
Une structure fonctionnelle doit donc inclure au moins une action permanente.
- (4) Le raffinement ou l'abstraction permettent une approche progressive de la solution et de décrire l'application en niveaux hiérarchisés de l'extérieur vers l'intérieur, facilitant la compréhension.
- (5) La modularité est une conséquence de cette structuration.
La séparation du problème se fait par regroupement d'actions à couplage serré afin d'aboutir à une description sous forme de modules et de communication de messages asynchrones.

III.1.4 CONCLUSION

La détermination de la structure fonctionnelle fait appel à la conception modulaire et structurée pour permettre de décomposer l'application en sous ensembles et d'aboutir à une structure modulaire, structurée et hiérarchique.

La hiérarchie peut se construire d'une manière descendante ou ascendante.

La démarche descendante construit progressivement une structure détaillée par raffinement successif jusqu'à l'obtention des actions élémentaires. Chaque étape de raffinement enrichissant la structure du module et de l'action.

La construction ascendante permet par le regroupement de modules de définir de nouveaux modules composites pour finalement aboutir à un seul module :

l'application dans son ensemble

Cette notion de module, où sont confinés tous les problèmes de contraintes de temps, facilitera l'étape de l'implantation matérielle.

III.2 APPLICATION AU PROJET

La figure III.9 présente, dans le formalisme de CALVEZ, le système de surveillance dans son contexte.

Le patient peut être perçu, à ce niveau, comme un générateur de signaux physiologiques que le système doit traiter, et le personnel soignant comme un observateur pouvant demander à cette station de surveillance la visualisation des résultats des traitements de signaux.

Au regard de cette observation, l'opérateur peut intervenir sur le malade par l'intermédiaire d'une thérapeutique appropriée et, grâce au système, en voir les conséquences.

De plus, ce système de surveillance alimente en ligne un système d'aide à la décision, qui propose au personnel soignant un diagnostic en fonction de l'état du patient ou de son évolution, afin de le guider dans sa démarche de raisonnement du choix de la thérapeutique.

Ce schéma (figure III.9) présente les liens entre le système de surveillance et son environnement représenté par les actions :

- > Patient (équivalent à un générateur de signaux physiologiques)
- > Système d'aide au diagnostic

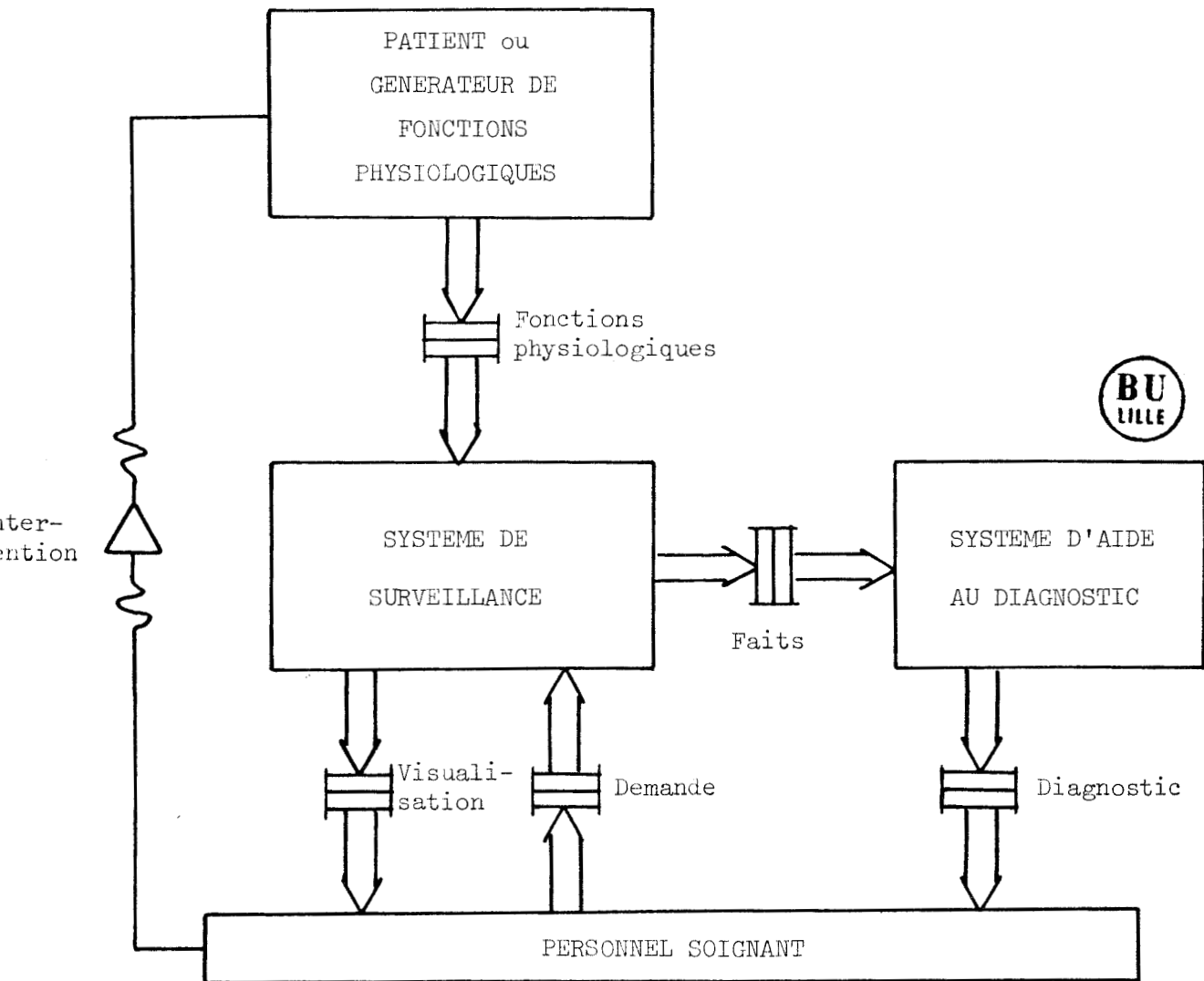
--> Personnel soignant (pouvant "agir" sur le système en sélectionnant les différentes visualisations).

Ce schéma, avec les différentes actions et leurs liens, indique implicitement les trois principales fonctions du système :

--> Analyse de fonctions physiologiques

--> Génération des faits

--> Visualisation des résultats sur demande de l'opérateur.



SYSTEME DE SURVEILLANCE

DANS SON CONTEXTE

figure III.9

Au regard des spécifications fonctionnelles, ce système de surveillance peut être découpé, par décomposition descendante, en différentes actions (figure III.10) :

- > L'Unité de Traitement Local (U.T.L.)
Chargée des traitements propre à chaque signal

- > L'Unité de Traitement Global (U.T.G.)
Chargée des traitements relatifs à l'ensemble des fonctions physiologiques (collecte des résultats, alarmes, génération des faits...)

- > L'Unité de Dialogue (U.D.)
Chargée d'interfacier le système de traitement avec l'opérateur (reconnaissance de la demande, visualisation...)

Chacune de ces trois actions peut, bien évidemment, être décomposée de nouveau.

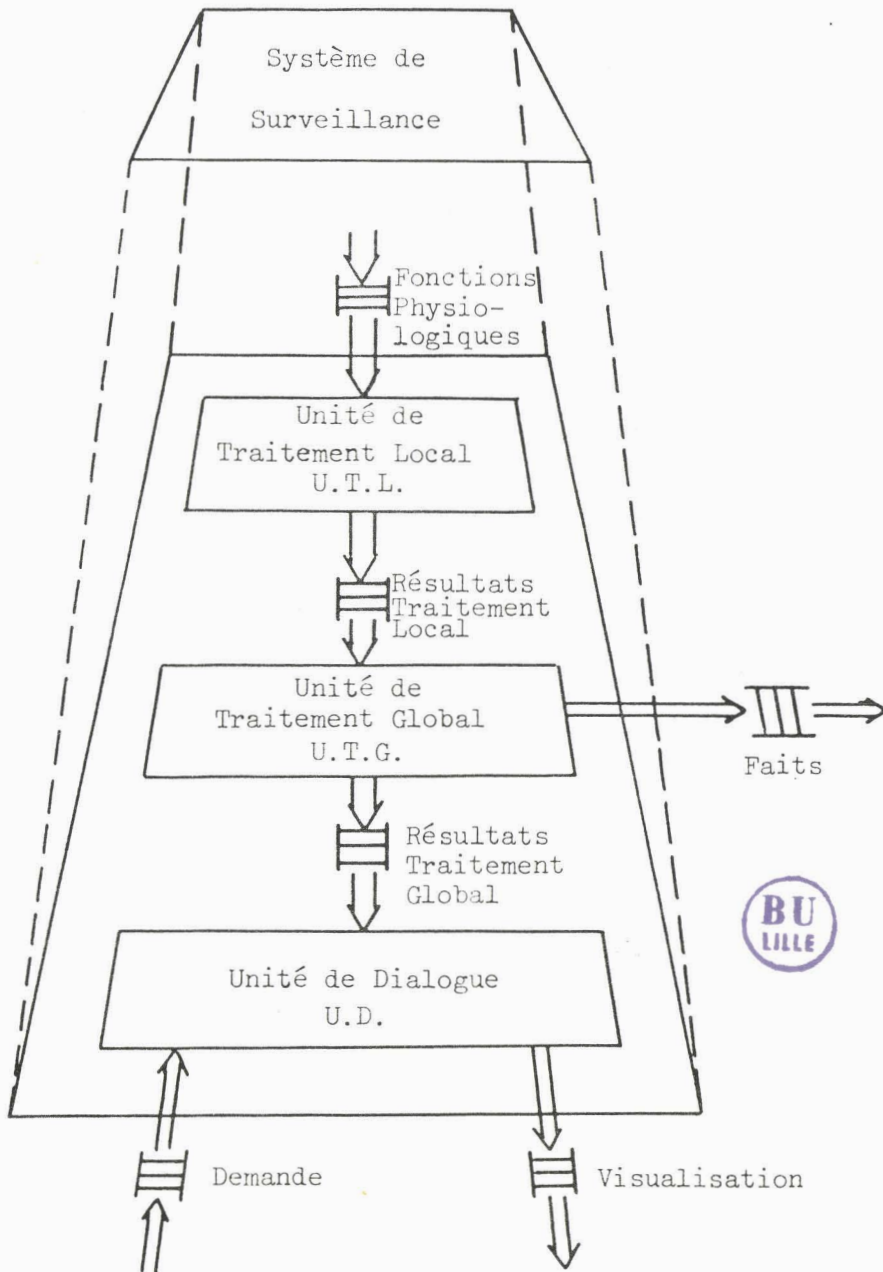


figure III.10

La figure III.11 présente trois niveaux de décomposition de l'unité de traitement local.

L'action dans son ensemble (Premier niveau de décomposition) est décomposée en deux sous actions donnant naissance au deuxième niveau de décomposition :

- > L'unité de traitement local de la fonction cardiovasculaire
- > L'unité de traitement local de la fonction respiratoire.

Chacune de ces sous actions est de nouveau décomposée (troisième niveau de décomposition)

Ainsi l'action " Unité de Traitement Local de la fonction cardiovasculaire " fournit deux sous actions :

- > U.T.L. relative à l'ECG
 - > U.T.L. relative à la PAP ou à la TA
- et l'action " Unité de Traitement Local de la fonction respiratoire " fournit :
- > U.T.L. relative à la pression (PRE)
 - > U.T.L. relative au volume (VOL) et au CO₂.

La contrainte, liée à l'extraction du paramètre VDS, de construire la courbe $CO_2 = f(VOL)$ implique de traiter ces deux signaux par la même action.

De même une décomposition supplémentaire de chaque U.T.L. appliquée à un signal est présentée par les figures III.12 ET III.13.

La figure III.12 est une décomposition valable pour les signaux ECG, PAP, TA, ou PRE tandis que la figure III.13 est une décomposition spécifique au traitement du CO₂ et du VOL.

physiologiques

U.T.L.

PREMIER NIVEAU DE DECOMPOSITION

Résultats

Fonction cardiovasculaire

Fonction respiratoire

DEUXIEME NIVEAU DE DECOMPOSITION

U.T.L. FONCTION CARDIOVASCULAIRE

U.T.L. FONCTION RESPIRATOIRE

Résultats cardiovasculaire

Résultats respiratoire

ECG

SPAP TA

PRE

VOL CO2

TROISIEME NIVEAU DE DECOMPOSITION

U.T.L. E.C.G.

U.T.L. P.A.P. T.A.

U.T.L. PRE

U.T.L. VOL - CO2

Résultats E.C.G.

Résultats P.A.P. T.A.

Résultats PRE

Résultats VOL - CO2



figure III.11

figure III.12

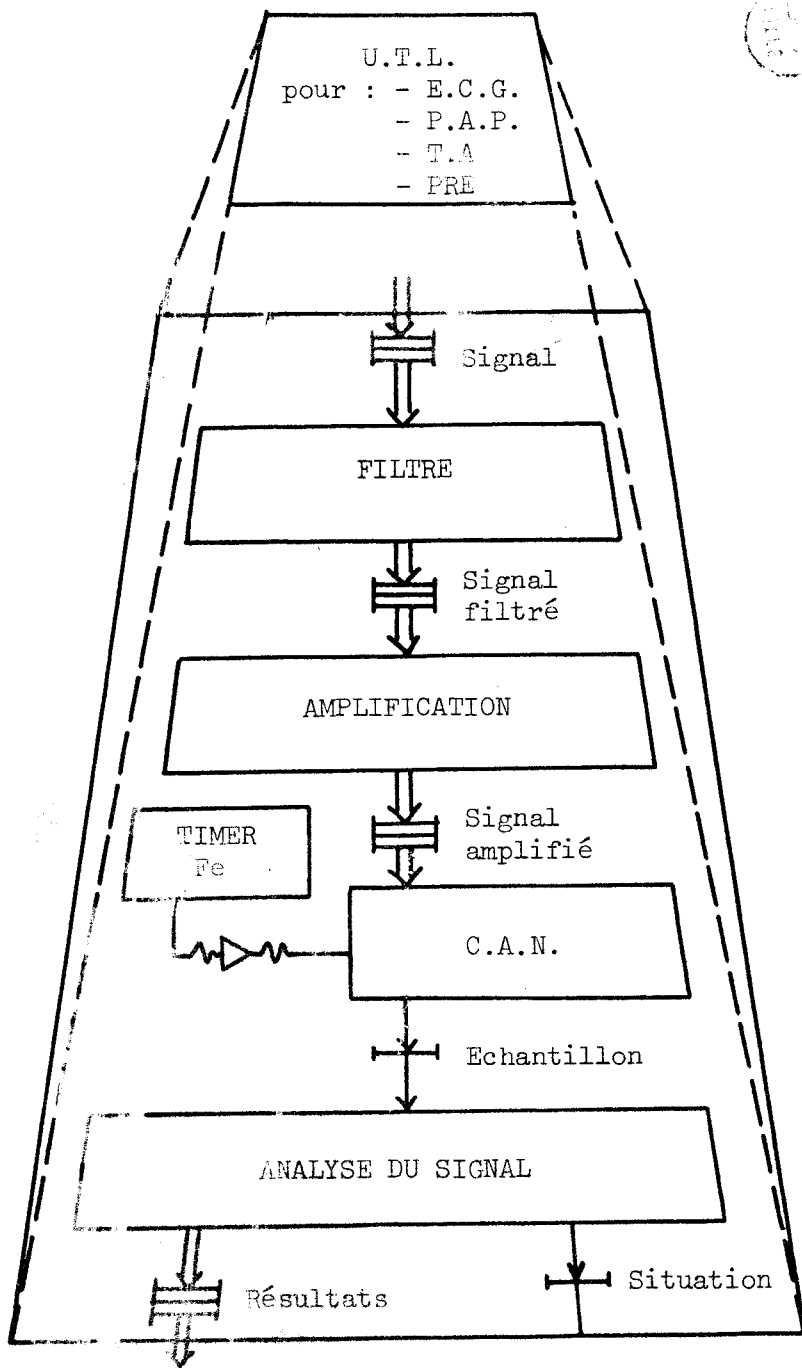
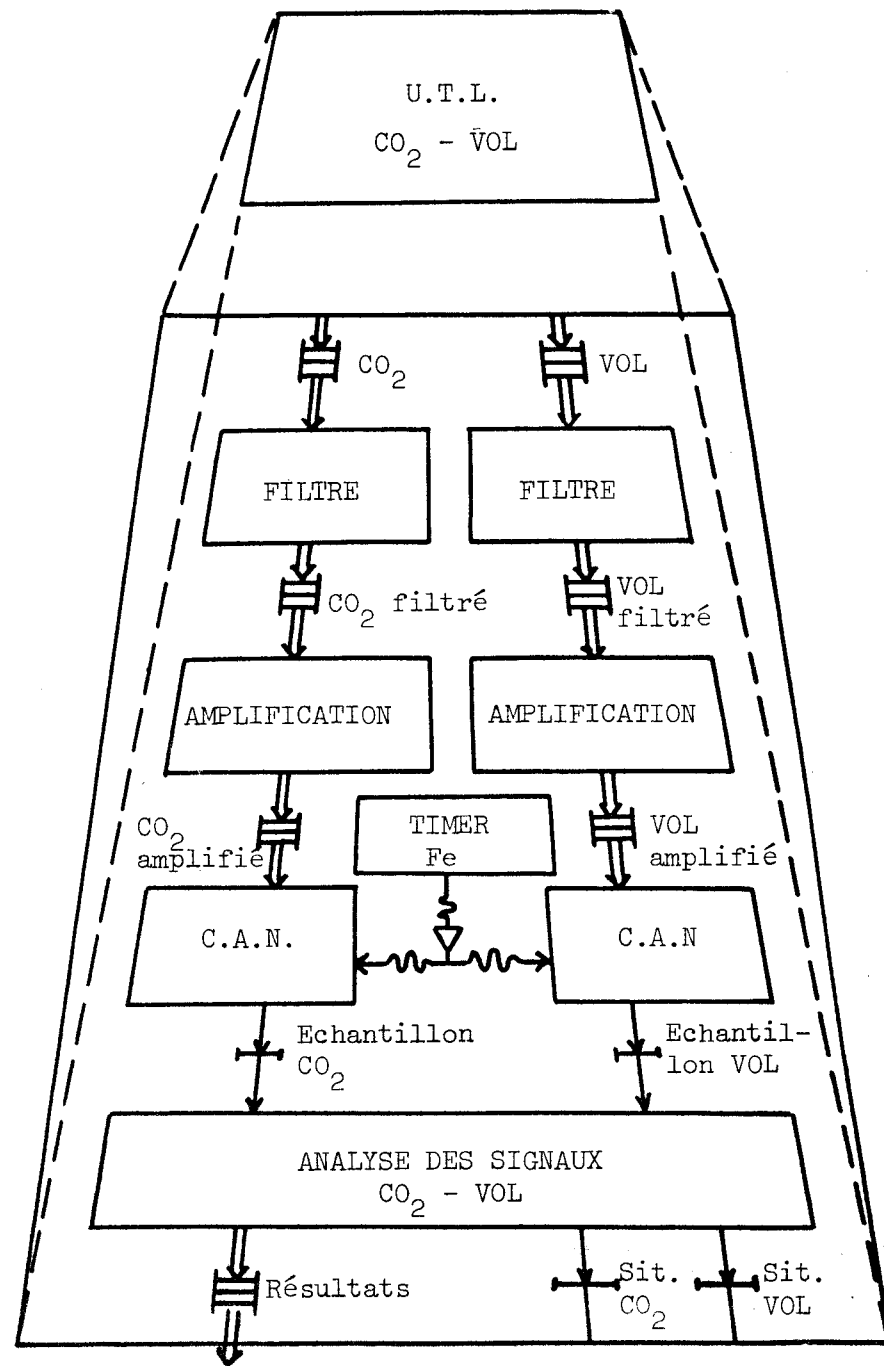


figure III.13



La figure III.14 présente la décomposition de l'Unité de Traitement Global (U.T.G.) avec les trois sous actions principales qui sont :

- > La gestion des situations alimentant la gestion des alarmes.
- > La gestion des résultats.
- > La production des faits.

Les modules " gestion des situations ", " gestion des alarmes " sont repris à la figure III.15 indiquant un niveau de décomposition plus détaillé.

On y trouve une action " Timer " déclenchant l'activité de la " scrutation de l'ensemble des situations " toutes les quinze secondes, une action " Horloge " fournissant l'heure au module " mémorisation des situations " afin qu'il puisse dater ces situations.

Ces situations sont fournies à un module " gestion des alarmes " qui lui même est composé des sous actions " reconnaissance de l'anomalie " et " génération de l'alarme ".

Au regard des situations, l'action " reconnaissance de l'anomalie " déclenche, si il y a lieu, l'action " génération de l'alarme " qui en fonction du type de l'alarme génère l'alarme appropriée (sonore, visuelle ...).

La figure III.16 présente la décomposition descendante du module " gestion des résultats ".

Le dernier niveau de décomposition présente une action " Timer " réglé à quinze minutes qui déclenche le module " scrutation de l'ensemble des résultats ". Celui ci passe ces données au module " mémorisation sur 24 H. des résultats datés " qui, grâce à la variable heure, stocke ces résultats avec une indication de date.

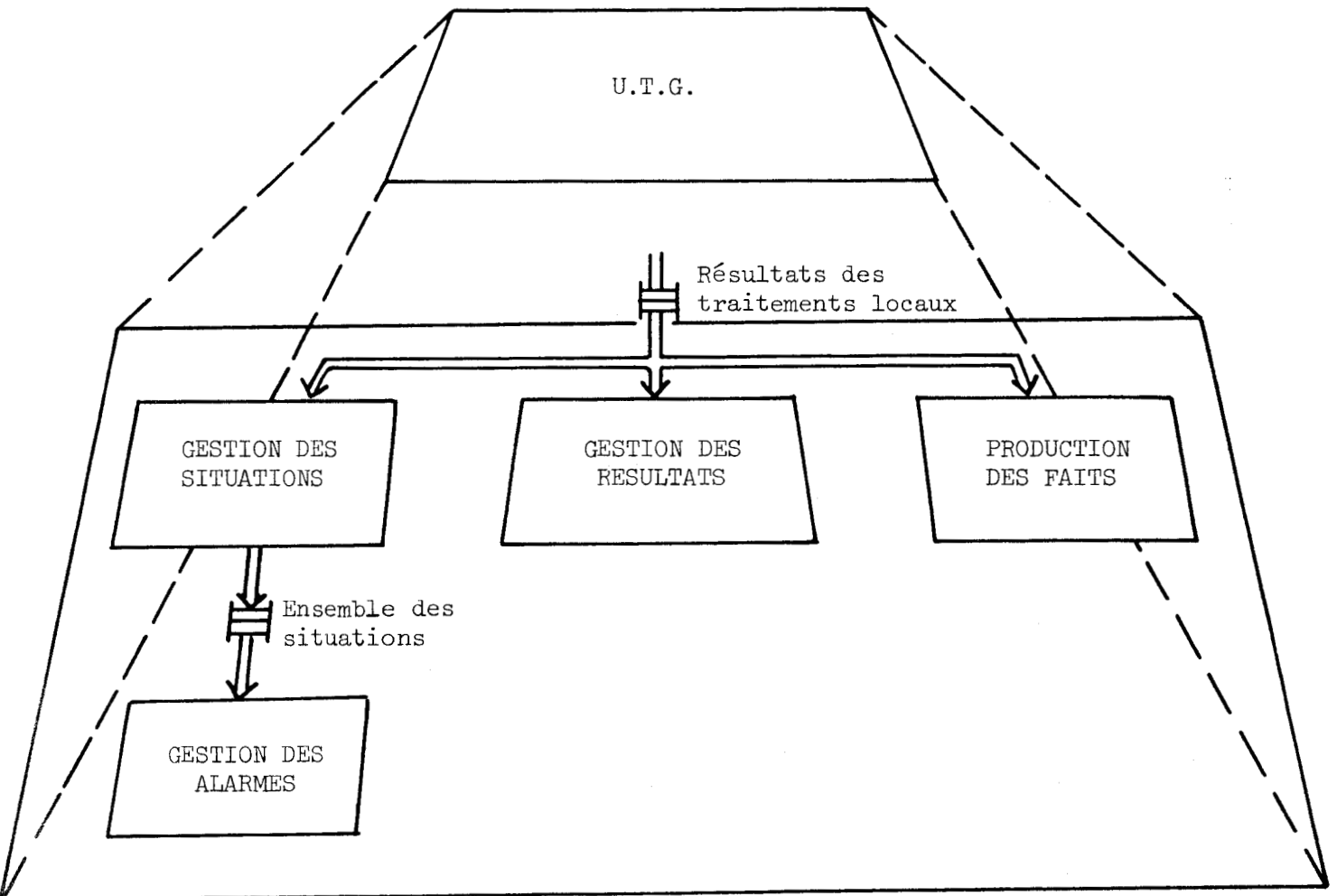
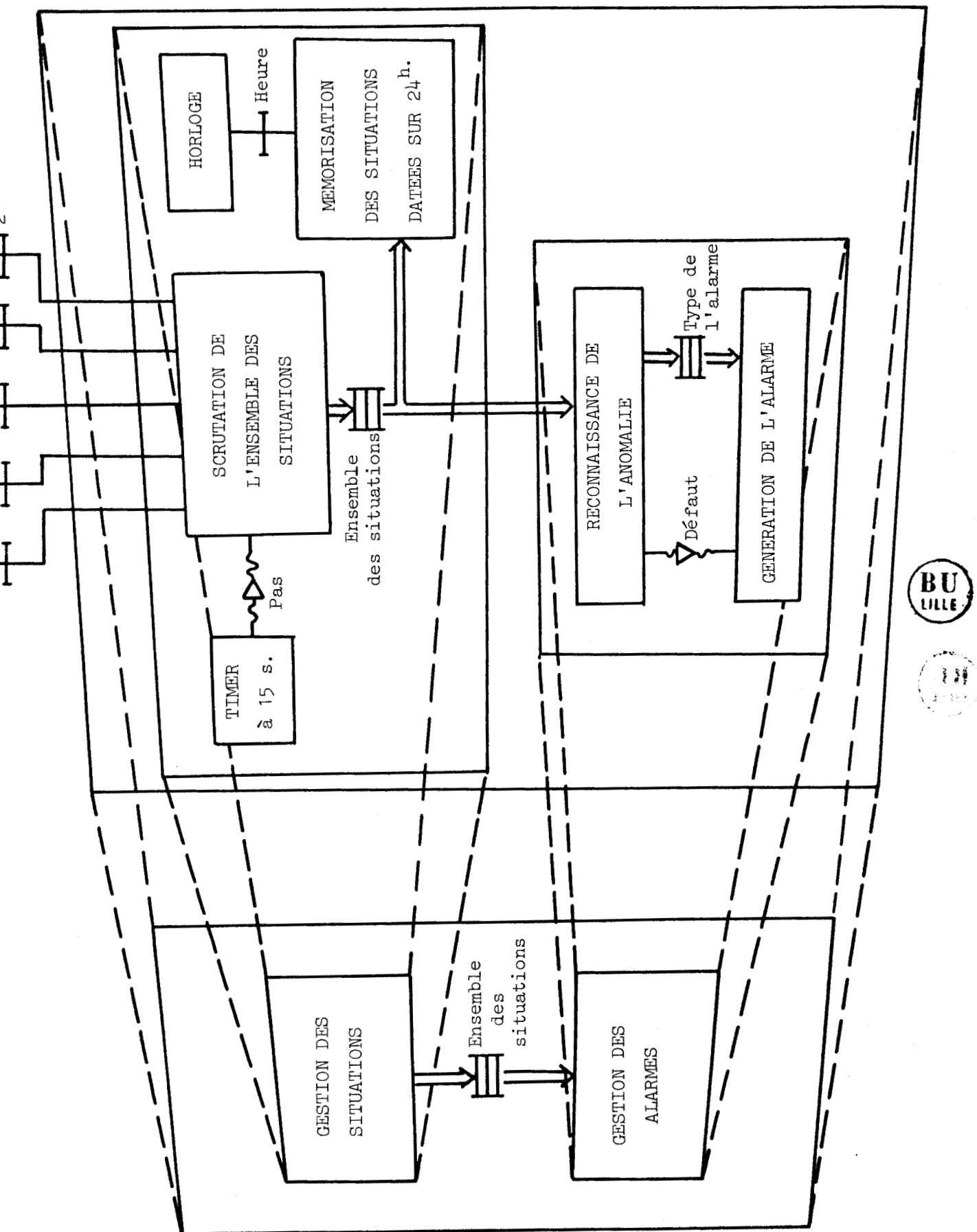


figure III.14

figure III.15



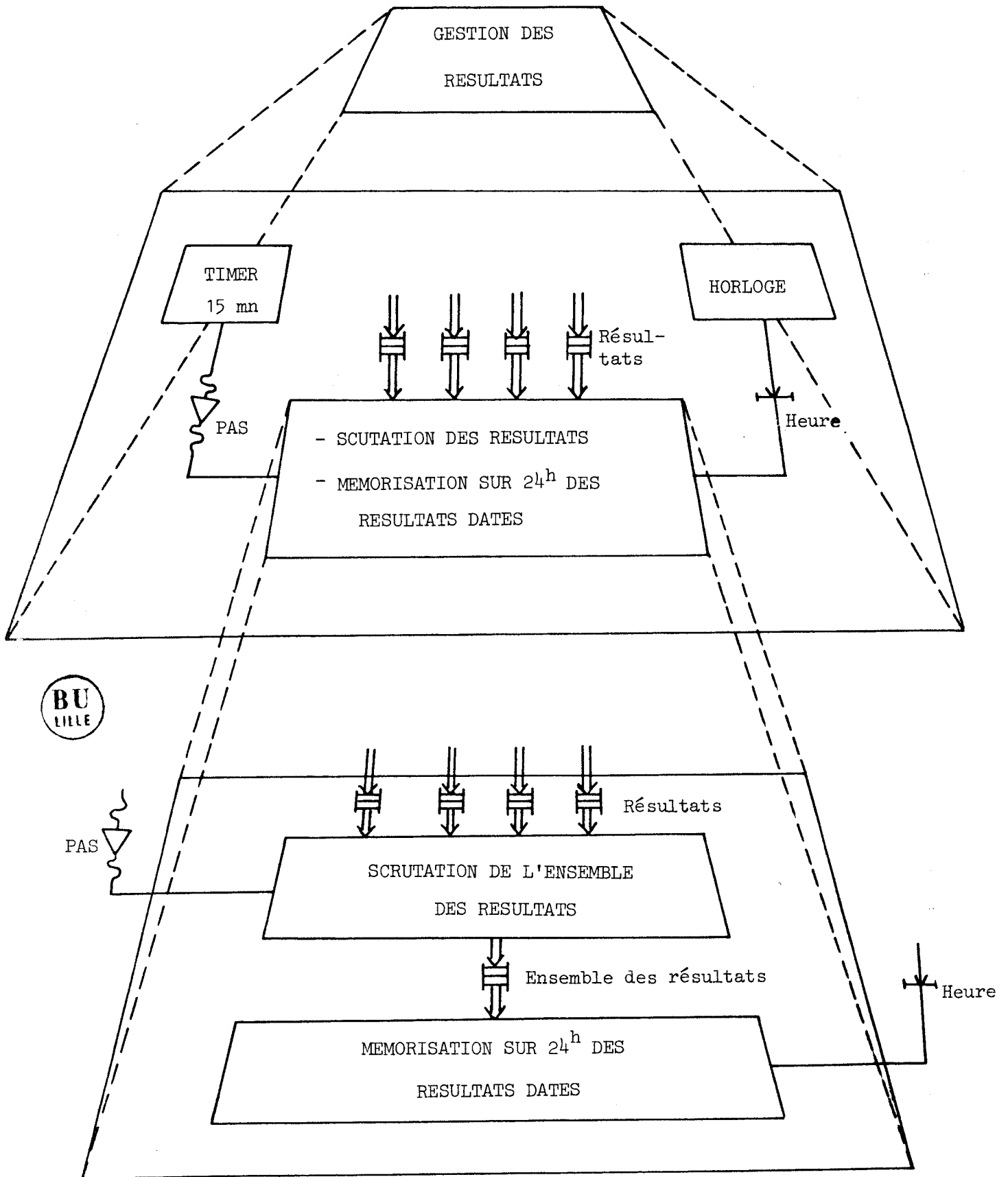


Figure III.16

L'Unité de Dialogue (U.D.) sera décomposée comme le montre la figure III.17.

L'opérateur en appuyant sur une touche déclenche l'action de " reconnaissance de la demande ".

Une fois cette demande reconnue, l'action appropriée est déclenchée (par l'intermédiaire des événements ev1, ev2,...evn).

- Ces actions peuvent être :
- observation des situations
 - observation des résultats
 - visualisation d'un signal
 - etc...

Ces actions fournissent les informations à un module de " mise en forme " de ces informations chargé d'adapter les données pour la visualisation.

La figure III.18 présente un exemple d'action pouvant être déclenchée par le module " reconnaissance de la demande ".

Il s'agit de la fonction " observation de l'état physiologique antérieur de l'ECG ".

Cette action a besoin de connaître la largeur de la fenêtre temporelle d'observation.

Une fois le rappel de tous les histogrammes répertoriés dans cette fenêtre, une action est chargée de lier et de mettre en forme ces données afin de ne fournir qu'un seul histogramme à la représentation.

Bien évidemment, ces décompositions ne sont pas finales.

L'objectif de ce travail n'est pas de montrer toutes les étapes des décompositions mais d'en présenter le principe par quelques exemples.

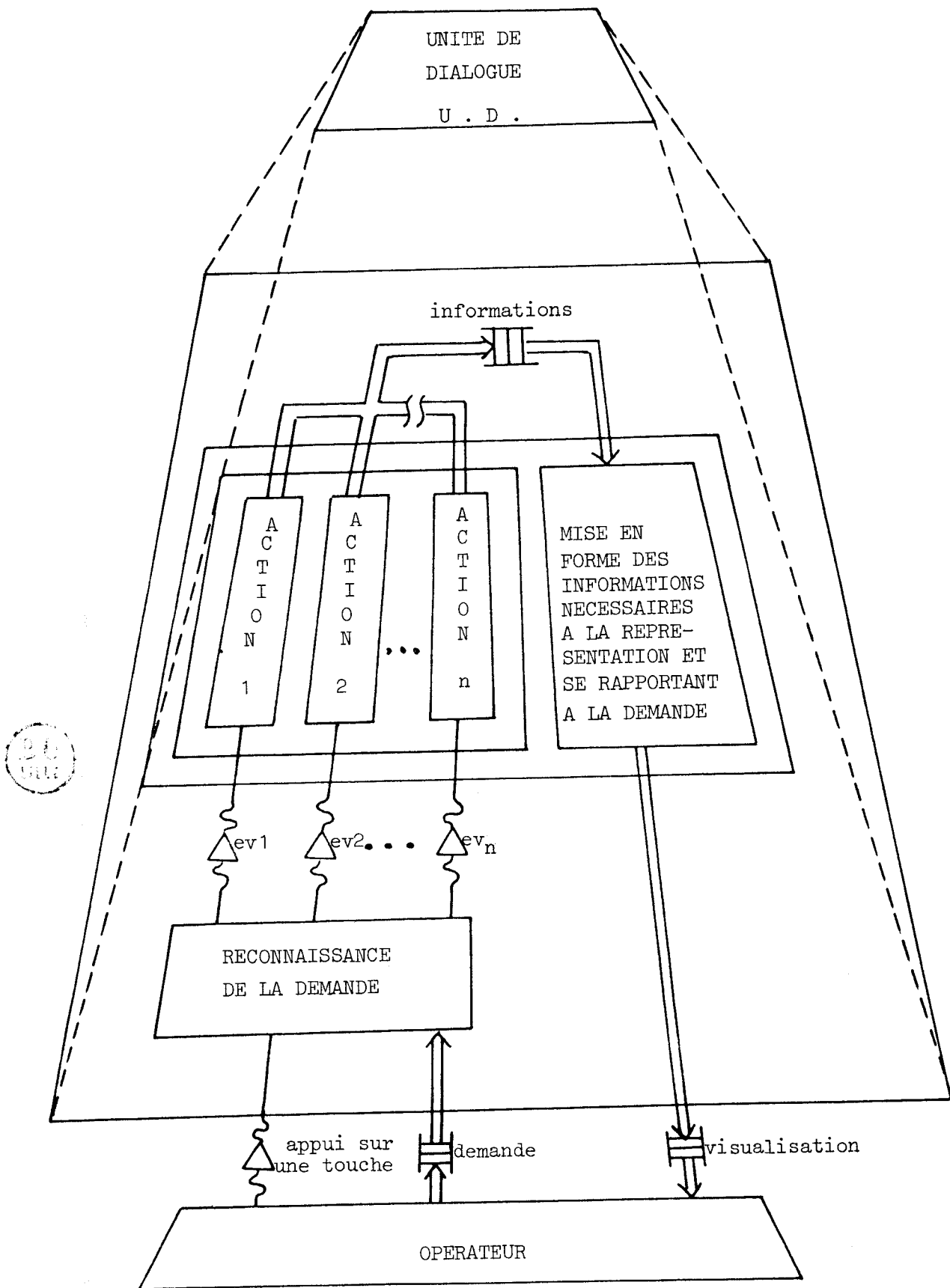


Figure III.17

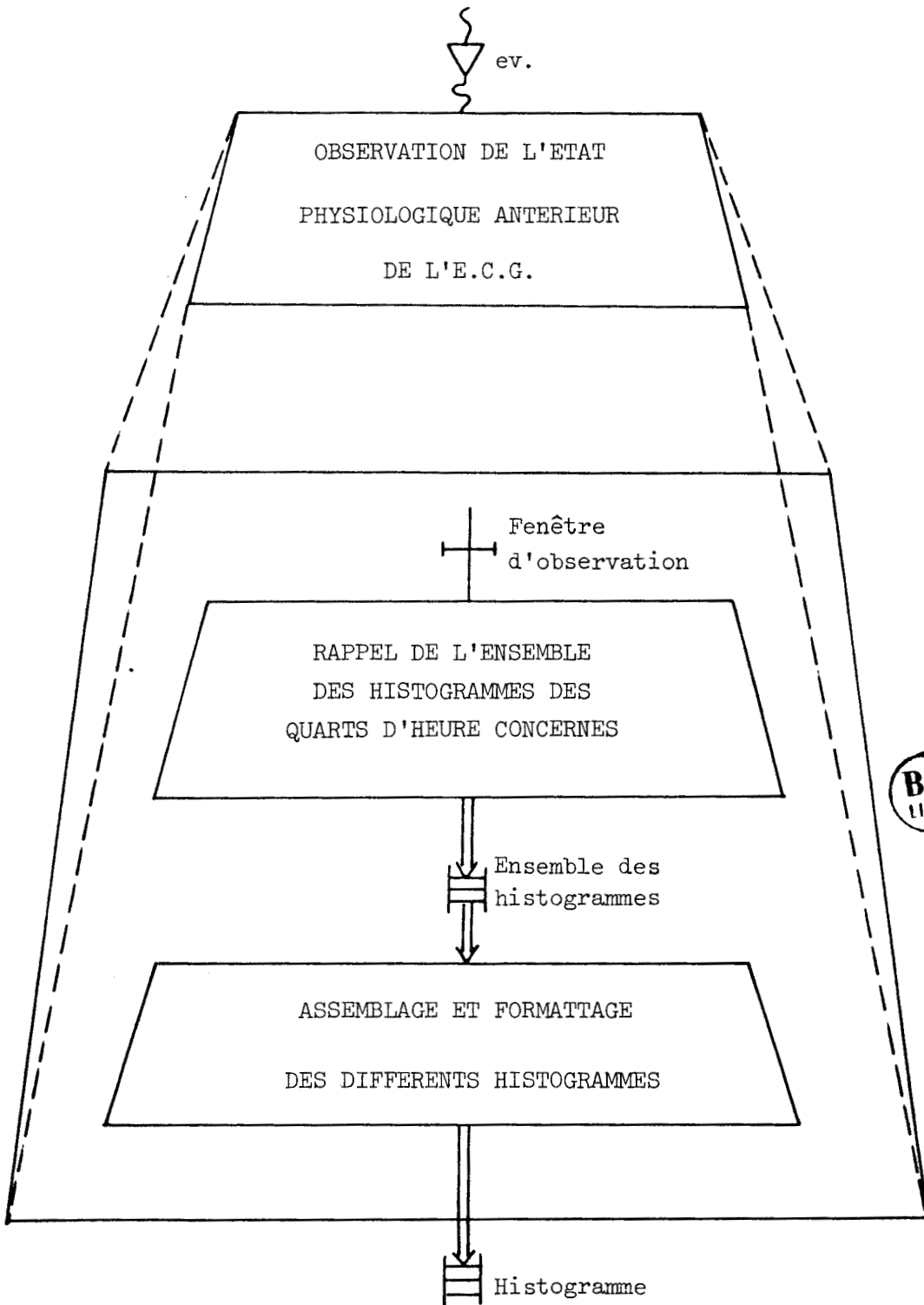


Figure III.18

IV TROISIEME ETAPE : LA DESCRIPTION OPERATOIRE

IV.1 MODELE DE CALVEZ

Le but de cette étape est de détailler le comportement interne des actions élémentaires à partir des spécifications issues de l'étape précédente et des spécifications opératoires.

Comme pour l'étape précédente, l'auteur définit un modèle à respecter.

IV.1.1 MODELE DE COMPORTEMENT DES ACTIONS

Deux types de règles précisent le comportement des actions :

- * les règles d'évolution de l'activité
- * les règles concernant les échanges interaction.

* Règles de description et de comportement

- (1) La description d'une action est décomposée en deux partie :
 - > une partie initialisation exécutée une seule fois lors de la mise en fonctionnement de l'action.
 - > une partie exécution décrivant l'évolution cyclique de l'action.
- (2) Lors de la description fonctionnelle deux types d'action avaient été décrites :
 - > les actions permanentes dont la description retenue sera :

```
Cycle
    < procédures >;
End;
```

- > les actions temporaires dont l'activité est subordonnée à l'existence d'événements ou messages d'entrée :

```
Cycle
    événement 1 : < procédure 1 >;
    :
    :
    événement i : < procédure i >;
    :
    :
    message j   : < procédure j >;
End;
```

Ce modèle implique les caractéristiques suivantes :

- (1) Une action peut être en attente simultanée de plusieurs conditions d'évolution. Lorsqu'une condition devient vraie, le traitement associé est exécuté : il y a exclusion mutuelle entre les traitements.
- (2) L'évolution de l'action est celle d'une machine séquentielle non déterministe car dans le cas de deux conditions vraies simultanément, le seul traitement exécuté est choisi au hasard, l'autre traitement sera exécuté ensuite.
- (3) Chaque traitement associé à un événement est une procédure qui peut se détailler par raffinement (programmation structurée).
- (4) Des conditions supplémentaires d'évolution peuvent être utilisées afin d'améliorer les possibilités de synchronisation interaction, ce qui s'écrit :

```
When
    événement i : < procédure i >;
    message j   : < procédure j >;
End;
```

Cette instruction de contrôle n'autorise que l'exécution d'un seul traitement : si plusieurs conditions sont vraies simultanément il n'y a pas d'effet mémoire comme pour l'instruction Cycle.

* Règles d'échanges interaction

Les synchronisations interaction sont réalisables :
--> soit par génération d'événement (Signal(ev));
--> soit par transfert de message (Send(mess)).

Les échanges de données sont réalisables par les variables partagées et les ports.

IV.1.2 CARACTERISTIQUES DE LA DESCRIPTION OPERATOIRE

- (1) Toutes les propriétés citées pour la structure fonctionnelle sont conservées car les règles pour la description algorithmique sont conformes aux règles de description externe.
- (2) Le fonctionnement de l'application est persistant car en faisant l'hypothèse de durée nulle pour les instructions qui servent à décrire le comportement de chaque action, tous les événements et tous les messages

vont engendrer une évolution des actions qui leurs sont liées.

- (3) Le choix des structures CYCLE et WHEN permet de donner une description structurée des actions.
- (4) L'action élémentaire est une unité de description, de compilation et de répartition.
- (5) Une description sera dite bien définie si :
 - > la description fonctionnelle est bien définie
 - > toutes les entrées de variables sont exploitées et toutes les sorties sont modifiées
 - > l'état de l'application initial est unique.

Il est important que chaque entrée d'événement ou de message induise l'activité de l'action (commandabilité) et que cette activité génère des événements et des messages sur toutes les sorties correspondantes (observabilité).

IV.1.3 DETERMINATION DES ALGORITHMES

Comme aucun choix technologique n'a encore été fait, la détermination des algorithmes se fait à partir des spécifications opératoires sans tenir compte des temps d'exécution.

Une approche structurée est facilitée par l'utilisation des instructions CYCLE et WHEN.

IV.1.4 IMPORTANCE DU LANGAGE DE DESCRIPTION

Le langage de description doit être indépendant de toute technologie, c'est pourquoi l'auteur retient un langage de programmation de haut niveau qui exprime toute la sémantique des variables manipulées, des opérations sur ces variables et de l'évolution des actions.

Une telle description est alors un document spécifiant d'une manière précise et non ambigu la solution pour les étapes de recherche de la structure d'exécution et de la réalisation.

IV.2 APPLICATION AU PROJET

Afin d'illustrer le modèle défini précédemment, nous présentons dans la suite quelques exemples des deux types d'actions :

- * actions permanentes
- * actions temporaires

sous la forme de la description algorithmique proposée par CALVEZ.

* Exemples d'actions permanentes

La figure III.12 présente un Timer générant l'événement PAS déclenchant la conversion analogique numérique du signal (la fréquence d'échantillonnage choisie est de 50 Hz).

La description algorithmique suivante présente cette fonction qui est une action permanente :

```
--> Action TIMER (Sortie : événement PAS)
      Début
          Cycle : attendre (20 ms);
                Signal (PAS);
          End;
      End;
```

Exemple 1

De même la fonction Horloge (figure III.15 et figure III.16) sera décrite de la façon suivante :

```
--> Action HORLOGE (sortie : variable HEURE )
      Début
          HEURE := 00 H 00 MN 00 S; (* partie
                                   initialisation *)
          Cycle : attendre (1 s);
                HEURE := HEURE + 1; (* comptage en
                                   base
                                   sexagésimale *)
          End;
      End;
```

Exemple 2

Ces deux exemples indiquent clairement que l'activité des actions permanentes n'est pas sujette à la présence d'événements ou de messages.

* Exemples d'actions temporaires

Afin de présenter ce type d'action, nous emprunterons deux actions à la figure III.17 :

```
--> Action RECONNAISSANCE DE LA DEMANDE
      ( entrée : - événement APPUI SUR UNE TOUCHE
              - message DEMANDE
        sortie : - événement EV1, EV2, ...EVN)
```

```
Début
  When
    APPUI SUR UNE TOUCHE : lecture (DEMANDE);
    If DEMANDE.ACTION1 Then Signal (EV1);
    If DEMANDE.ACTION2 Then Signal (EV2);
      :           :           :
      :           :           :
    If DEMANDE.ACTIONN Then Signal (EVN);
  End;
End;
```

Exemple 3

```
--> Action OPERATION
      ( entrée : événement EV1, EV2, ...EVN)
      sortie : message INFORMATIONS )
Début
  Cycle
    EV1 : ACTION1; (* Le message INFORMATIONS
    EV2 : ACTION2;   est envoyé par les
      :           :   différentes action *)
      :           :
      :           :
    EVN : ACTIONN;
  End;
End;
```

Exemple 4

La description algorithmique suivante présente une des actions du module OPERATION (figure III.18).

```
--> Action OBSERVATION DE L'ETAT PHYSIOLOGIQUE ANTERIEUR
      ( entrée : - événement EV.
      - variable FENETRE D'OBSERVATION
      sortie : message HISTOGRAMME )
Début
  When
    EV : Début
      Lecture (FENETRE D'OBSERVATION);
      RAPPEL DES HISTOGRAMMES CONCERNES;
      ASSEMBLAGE ET FORMATTAGE DES HISTO.;
      Send (HISTOGRAMMES);
    End;
  End;
End;
```

Exemple 5

Ces différentes actions présentent les principales représentations utilisées par cette description algorithmique :

- (1) L'action TIMER (exemple 1) a pour fonction de générer l'événement PAS toutes les 20 ms, ce qui est indiqué par *Signal(PAS)*, déclenchant ainsi l'activité de l'action CAN.
- (2) L'exemple 2 présente la construction d'une HORLOGE battant la seconde, en mettant à jour toutes les secondes la variable partagée HEURE. Cet exemple indique aussi la distinction faite entre les parties initialisation et exécution.
- (3) Dans l'exemple 3, on montre comment sont générés les différents événements (EV1, EV2, ...EVN) lorsque l'opérateur formule sa demande.
- (4) En fonction de la présence d'un des événements (EV1, EV2, ...EVN), l'exemple 4 indique comment sont activées les actions respectives (par la fonction *Signal(EV.)*).
- (5) L'exemple 5 représente comment est envoyé le message INFORMATIONS par l'intermédiaire de *Send(..)* (dans ce cas le message est HISTOGRAMME).

V QUATRIEME ETAPE : LA STRUCTURE EXECUTIVE

V.1 MODELE DE CALVEZ

Cette quatrième étape a pour but de déterminer la structure exécutive (ou support d'exécution) qui est l'ensemble des ressources et des liens entre celles ci, nécessaires au fonctionnement de l'application.

La description opératoire et les spécifications technologiques servent de données à cette étape délicate et décisive.

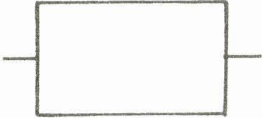

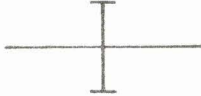

Cette étape doit déterminer la description exécutive composée de la structure d'exécution et de l'intégration, exprimant le rôle de chaque constituant.

Un modèle a été défini pour chacune de ces sous parties.

V.1.1 MODELE DE LA STRUCTURE EXECUTIVE

Par analogie avec la structure fonctionnelle on adopte, pour les représentations, les équivalences suivantes (figure III.19) :

- action <--> processeur :
un processeur étant un organe de transformation de l'information
- événement <--> signalisation :
une signalisation pouvant être des signaux de contrôle, interruption...
- variable partagée <--> mémoire
- port <--> noeud de communication

REPRESENTATION	SIGNIFICATION STRUCTURE FONCTIONNELLE	SIGNIFICATION STRUCTURE EXECUTIVE
	ACTION	PROCESSEUR
	EVENEMENT (Synchronisation)	SIGNALISATION (Signaux de contrôle interruption)
	VARIABLE PARTAGEE (Echanges d'informations)	MEMOIRE
	PORT (dépot de messages)	NOEUD DE COMMUNICATION

BU
LILLE

Figure III.19

Ces constituants doivent satisfaire les règles suivantes :

* Règles pour les processeurs

- > Un processeur en état de fonctionnement est toujours en activité
- > La durée d'exécution des instructions pour un processeur est finie
- > L'activité d'un processeur peut être modifiée sur détection d'une signalisation
- > Un processeur peut accéder au contenu des mémoires qui lui sont liées
- > Un processeur peut générer ses sorties.

* Règles pour les mémoires

- > Les mémoires se chargent de détecter si la variable concernée par une lecture ou une écriture leur appartient
- > La durée d'accès à une mémoire n'est pas nulle mais finie
- > Les contraintes d'intégrité doivent être respectées.

* Règles pour les liaisons

- > Les liaisons doivent assurer la cohérence des messages transmis
- > Dans le cas où plusieurs émetteurs et récepteurs exploitent la même liaison, les problèmes de désignations du correspondant et du message doivent être résolus.

V.1.2 MODELE D'INTEGRATION

La description exécutive est le résultat de l'opération de l'intégration, c'est à dire l'association de la description opératoire à la structure d'exécution.

Afin d'aboutir à une réalisation en état de fonctionnement le modèle d'intégration décrit deux types de règles :

- > Règles de correspondance entre les éléments
- > Règles de correspondance entre les structures.

* Correspondance entre les éléments

- > L'activité d'une action élémentaire sera assurée par un seul processeur du fait de son indivisibilité, de même que les variables partagées et les messages sont considérés indivisibles.
- > Un processeur peut être le support d'exécution de plusieurs actions, de même une mémoire peut contenir plusieurs variables partagées et un noeud de communication pourra véhiculer plusieurs types de messages.
- > Le choix des actions à implanter sur un même processeur doit tenir compte des contraintes de temps et du fait que le processeur doit rester opérationnel (taux d'occupation < 1).

* Correspondance entre structures

La structure fonctionnelle est acceptée par la structure d'exécution si tous les liens de la structure fonctionnelle se trouvent réalisés par les liens de la structure exécutive.

Cette vérification peut être obtenue en construisant une structure fonctionnelle réduite par abstraction et en s'assurant que celle ci est incluse dans la structure d'exécution. Pour chaque lien de la structure fonctionnelle, il faudra au moins une liaison au niveau exécutif.

V.1.3 CARACTERISTIQUES DE LA SOLUTION

- (1) Les principes de raffinement, d'abstraction, et de construction modulaire permettent d'aboutir à une description structurée du support exécutif.
- (2) La description exécutive est bien définie si :
 - > La structure d'exécution est bien définie, c'est à dire si elle est commandable et observable et si les conflits possibles sont pris en charge par la réalisation.
 - > La structure fonctionnelle est acceptée par la structure d'exécution.
 - > Tous les processeurs sont opérationnels.

- (3) La solution n'engendre pas d'interblocage ni de privation car les exécutions sont réalisées en temps fini.

V.1.4 RECHERCHE DE LA STRUCTURE EXECUTIVE

Pour la recherche de la structure exécutive, les spécifications technologiques jouent le rôle de contraintes influençant la solution.

En effet, si le processus à contrôler ne fait pas partie de la structure exécutive, il impose toutefois des contraintes de réalisations par ses entrées/sorties qui sont bien définies : l'interface système de commande - processus est à réaliser !

De même, le processus impose des contraintes de temps pour les commandes. Or la durée d'exécution des instructions des processeurs ne peut plus être considérée comme nulle, elle implique des "retards" qui peuvent engendrer le non respect de celles ci.

Il est donc nécessaire de vérifier ces contraintes de temps lorsqu'une solution est définie. A chaque action deux paramètres interviennent pour cette vérification :

- > FA : la fréquence maximale des activations
- > TA : le temps d'exécution maximum.

* Fréquence maximale d'activation

La fréquence d'activation est indépendante de la technologie mais dépend uniquement de la description opératoire et des spécifications.

La fréquence maximale d'activation d'une action temporaire est déterminée à partir des fréquences d'apparition des événements engendrant une exécution d'un traitement de l'action (cf. annexe 1).

* Temps d'exécution maximum

Le temps d'exécution maximum de l'action est fonction des temps d'exécution des instructions et donc du processeur retenu (cf annexe 1).

Un processeur chargé de l'exécution de n actions sera opérationnel si son taux d'occupation T_0 est inférieur à 1.

Le taux d'occupation peut être évalué par :

$$T_0 = \sum_i FA_i * TA_i$$

avec FA_i : fréquence maximale d'activation de l'action i
 TA_i : temps d'exécution maximum de l'action i.

Pour satisfaire les contraintes de temps on pourra agir :

- > sur le nombre et la nature des actions associées à un processeur
- > sur le temps d'exécution de chaque action et ceci soit en modifiant les algorithmes, soit en changeant de processeur.

V.1.5 POSSIBILITES DE CONSTRUCTION

La structure d'exécution peut s'obtenir selon trois modes :

* Regroupement de structures prédéfinies

Dans ce mode on exploite la structure exécutive d'un ou plusieurs types de matériel existant pour construire des structures plus complexes (par exemple : utilisation des cartes)

* Construction par raffinement

Il n'existe pas toujours de circuit répondant à toutes les fonctions demandées par le processeur. Une opération de décomposition en un ensemble de composants plus élémentaires est alors nécessaire.

* Construction par abstraction

Plusieurs processeurs interconnectés peuvent être remplacés par un seul si ce dernier reste opérationnel, permettant de simplifier la réalisation. Le choix du regroupement peut se faire de manière à réduire la complexité des interconnexions.

V.1.6 CRITERE DE CONSTRUCTION

Certaines spécifications technologiques imposent des critères de construction qui peuvent être :

- > La modularité
- > La minimisation du coût
- > La sûreté de fonctionnement.

* La modularité

La modularité peut être matérielle et/ou logicielle.

Dans le cas du matériel, elle se résoud par l'emploi de sous ensembles tels que des cartes standards.

Pour le logiciel la démarche est similaire au sens où le concepteur cherchera à utiliser des entités logicielles prédéfinies.

* La minimisation du coût

Ce critère est généralement utilisé pour minimiser le coût du matériel pour des productions en série.

On cherche donc à réduire successivement la solution par abstraction en regroupant les processeurs les plus couplés jusqu'à ce que les contraintes de temps ne soient plus respectées.

* Sûreté de fonctionnement

La sûreté de fonctionnement est généralement résolu par l'introduction de redondances au niveau de la structure exécutive et de contraintes d'exclusions de certaines actions.

L'auteur propose dans un premier temps de considérer l'application sans ces redondances, puis de les introduire avec les contraintes d'exclusions associées.

Ces contraintes seront déduites des critères de fonctionnement données dans les spécifications.

V.2 APPLICATION AU PROJET

Les spécifications technologiques servent de données à cette étape de conception. Or pour ce projet, ces spécifications imposent le matériel à utiliser.

De plus, le système doit pouvoir recevoir le traitement de nouvelles fonctions ou signaux sans être obligé de repenser toute sa structure et sa gestion; la modularité représente donc un critère très important de construction.

Ainsi, face à ce critère et aux spécifications technologiques, une carte est associée à chaque signal pour le traitement local et une carte supplémentaire assure l'exécution du traitement global et de l'unité de dialogue (associée à la console - clavier LYME 6000).

Comme les signaux CO2 et VOL nécessitent un traitement commun pour l'extraction du paramètre VDS, il a été décidé de traiter la fonction respiratoire (CO2, VOL, PRE) par une seule carte (SBC 80/24) afin de conserver le critère de modularité au niveau de la fonction.

Toute la gestion du système (Traiement global + Unité de dialogue) est assurée par une SBC 80/24.

La structure exécutive du traitement local de la fonction respiratoire est représentée par la figure III.20.

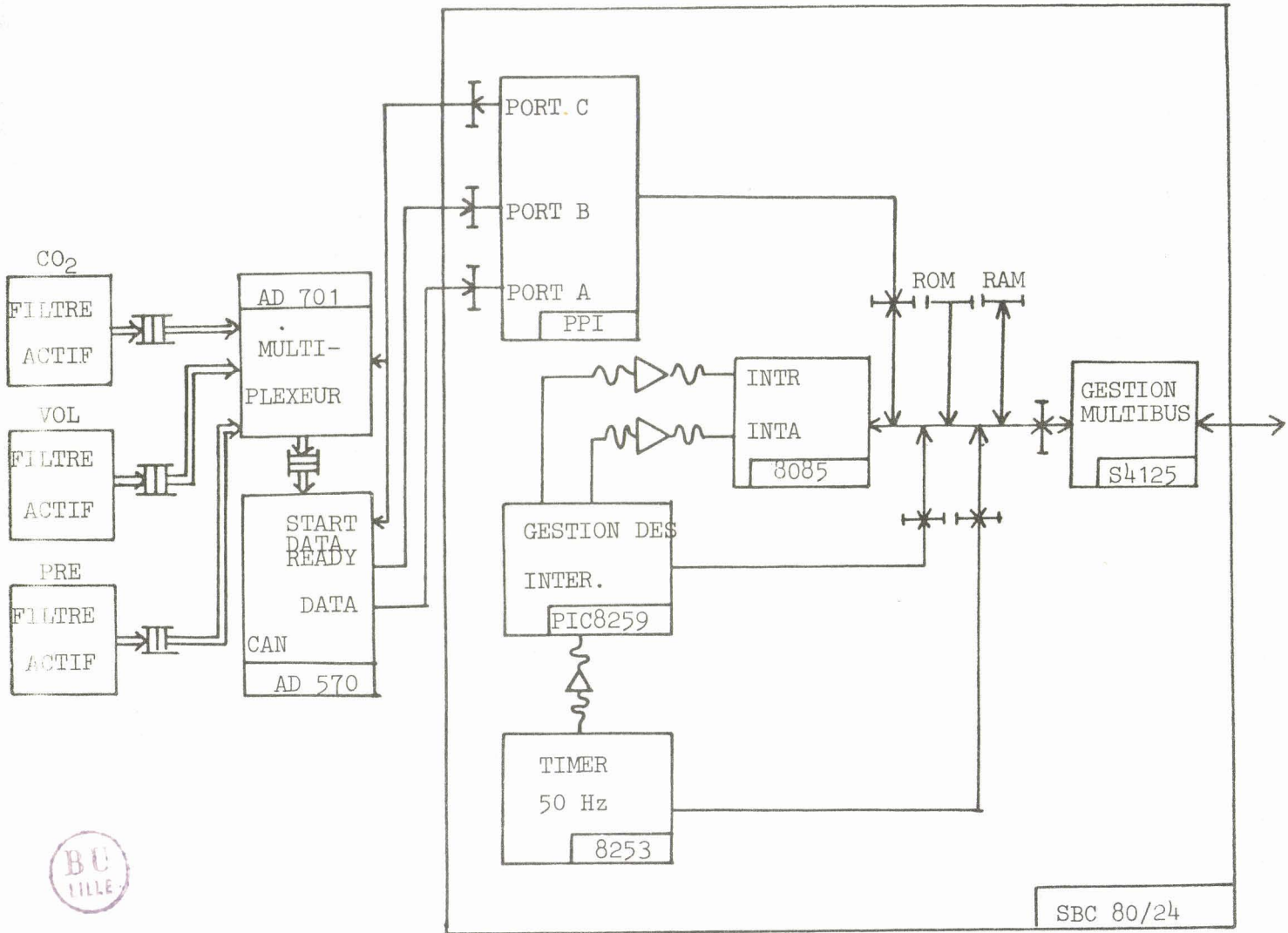


Figure III.20

La structure exécutive intégrant les actions Traitement Global et Unité de Dialogue est représentée par la figure III.21.

L'interface opérateur - système y est schématisée par la description de la console - clavier LYME 6000.

L'interface acquisition (filtre, amplification, conversion analogique, numérique) est réalisée sur une seule carte au format INTEL Multibus afin de pouvoir être intégrée dans le fond de panier.

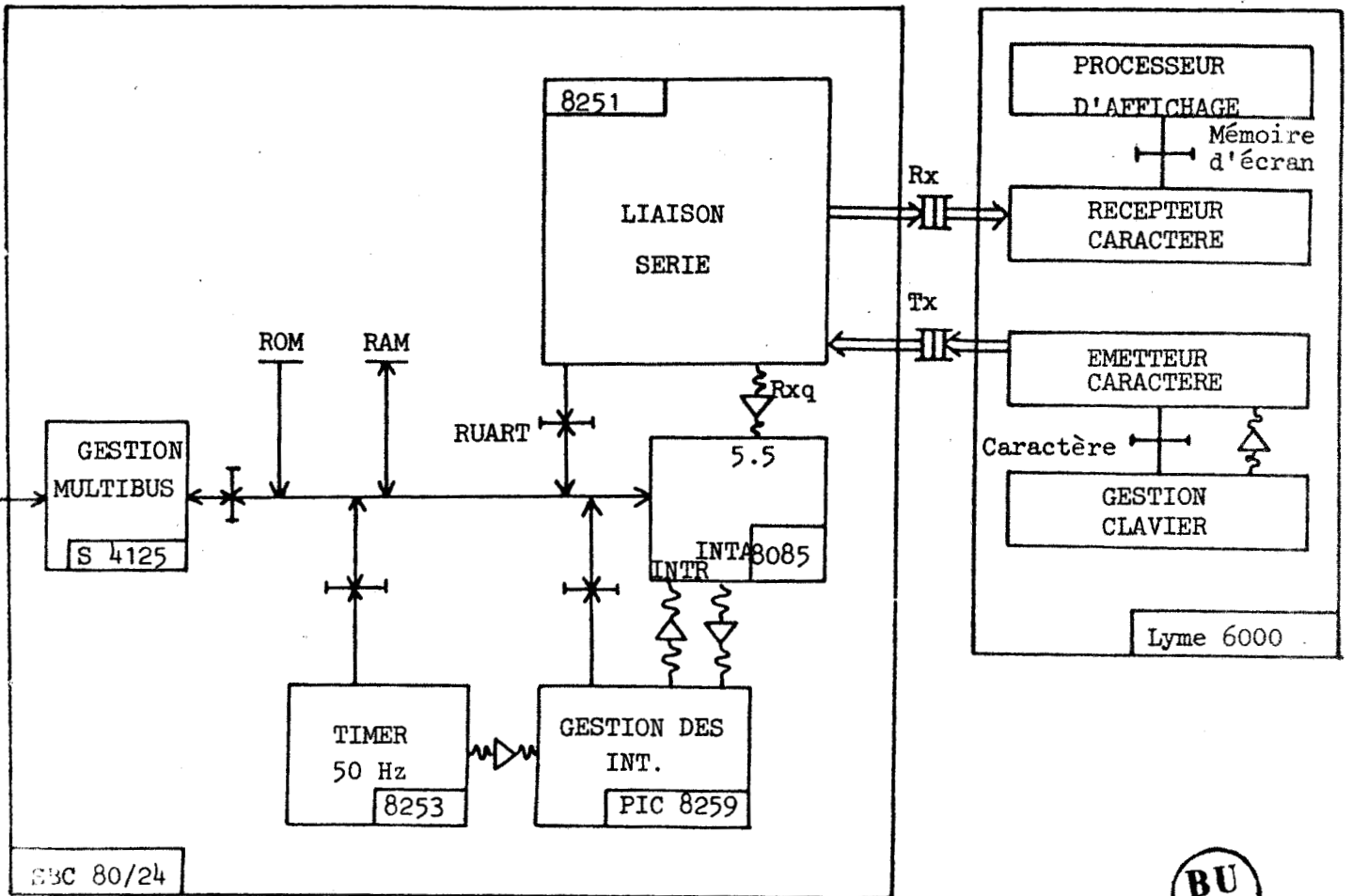


Figure III.21

La structure exécutive du système de surveillance dans son ensemble est représentée figure III.22.

Cette structure multiprocesseurs est du type avec un bus commun qui est le Multibus.



BU
LILLE

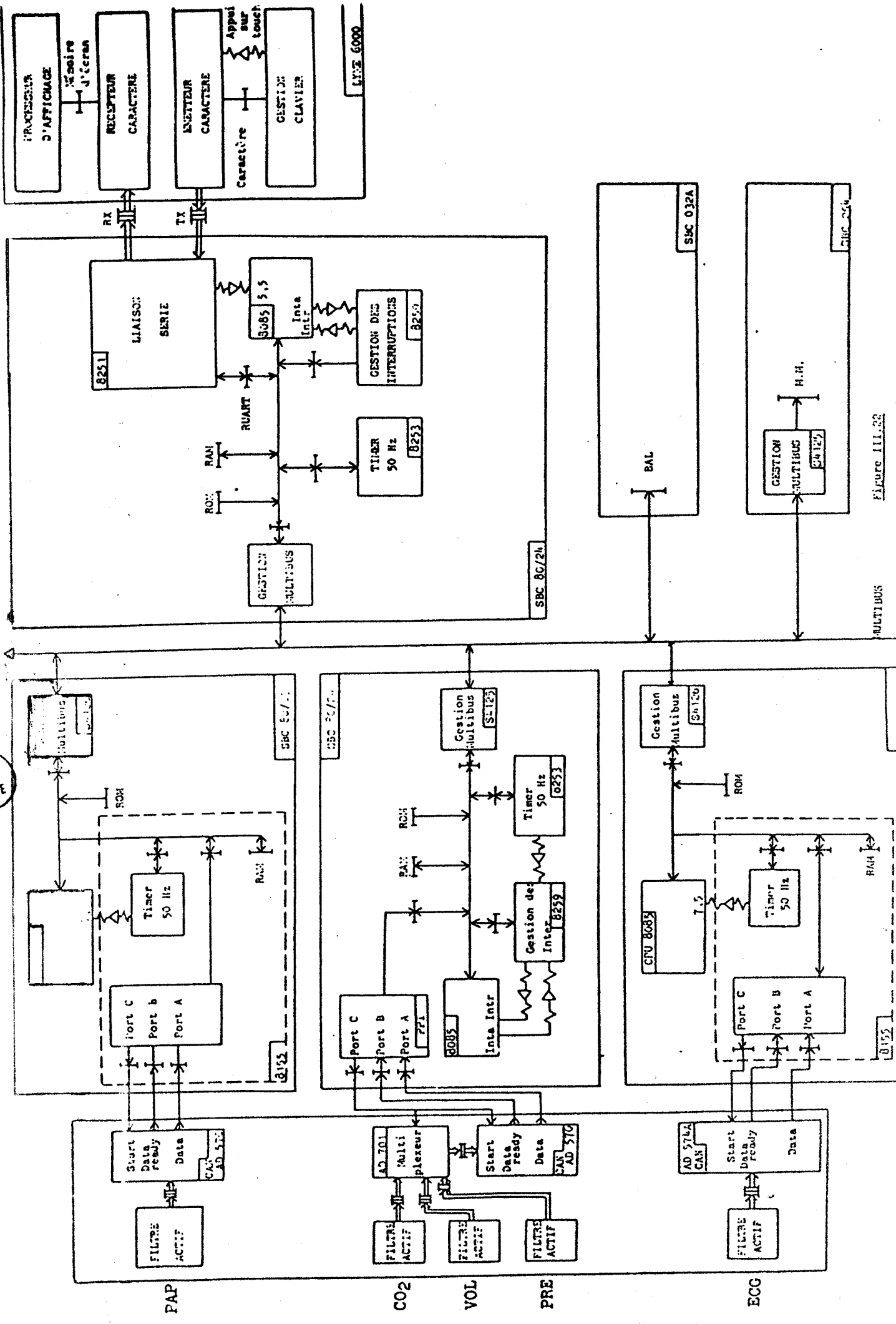


Figure III.22

MULTIBUS

VI DERNIERE ETAPE : LA REALISATION

VI.1 MODELE DE CALVEZ

La description exécutive et certaines spécifications technologiques propre à la réalisation servent de données à cette étape dont le rôle est de produire le matériel, le logiciel, puis l'intégration du logiciel sur le matériel...

VI.1.1 DEVELOPPEMENT PARALLELE DU MATERIEL ET DU LOGICIEL

L'auteur conseille fortement le développement parallèle du matériel et du logiciel afin :

- > de réduire le temps d'exécution
- > de faire intervenir les compétences des spécialistes des deux domaines profitant ainsi d'une influence réciproque.

VI.1.2 REALISATION MATERIELLE

La réalisation matérielle est directement liée à la structure exécutive et au critère de construction adopté.

Dans le cas de la modularité, la réalisation se réduit à l'assemblage de cartes et au développement de l'interface système de commande - processus.

Par contre si on cherche la minimisation du coût toute la construction est à faire ...

Une fois la réalisation matérielle effectuée, il faut s'assurer qu'elle est opérationnelle et qu'elle répond à la demande.

Cette garantie s'effectue au moyen de tests indépendants de l'application.

VI.1.3 REALISATION LOGICIELLE

La réalisation logicielle est l'implantation de la description opératoire sur la réalisation matérielle.

La nature de la description exécutive permet une décomposition en deux parties :

- > la transcription de la description opératoire sur la structure exécutive
- > la réalisation de la structure exécutive à partir de la réalisation matérielle.

Le premier point s'obtient soit par traduction manuelle de la description opératoire, soit par traduction automatique.

Toute transcription peut impliquer des non satisfactions des contraintes de temps. Il faut alors revenir à l'étape de recherche de la structure exécutive et peut être même à l'étape de la recherche des algorithmes.

D'où l'importance d'évaluer très tôt les temps critiques...

Le deuxième point, c'est à dire la réalisation de la structure exécutive à partir de la réalisation matérielle implique certains problèmes :

- (1) Le partage du processeur par plusieurs actions.
Ceci sous entend la notion de moniteur temps réel ou de fonctionnement sous interruptions.
- (2) La cohérence des variables partagées.
Lors de la réalisation, il faut s'assurer des contraintes d'intégrité des variables et donc définir les techniques d'accès.
Ces techniques d'accès dépendent de l'application :
dans un contexte temps réel, il faut assurer le remise à jour des variables très rapidement si on veut respecter toutes les contraintes de temps. De même pour des volumes importants d'informations, un blocage de l'accès trop long peut entraîner une privation.
- (3) La gestion des noeuds de communication.
La technique d'implantation est indépendante de la structure exécutive et du support matériel retenu.
Une des solutions peut être l'utilisation d'un sémaphore communiquant constitué :
--> d'une variable entière représentant la place disponible pour les messages
--> d'une file des messages disponibles
--> d'une file des actions en attente d'un message.
- (4) La prise en compte des événements et des messages.
Le problème se résume par l'implantation des instructions CYCLE et WHEN.

Le matériel se développant parallèlement au logiciel, il faut disposer d'un outil informatique qui émule la future réalisation matérielle pour tester le logiciel :

Il est nécessaire que ce matériel de test implique une intégration la plus directe, si on veut éviter des traductions supplémentaires.

VI.1.4 INTEGRATION

L'intégration, ou implantation du logiciel sur le matériel, doit donner lieu à une série de tests permettant de vérifier la conformité de la solution à la demande.

Si toutes les étapes de la conception ont été abordées avec la rigueur requise, cette phase d'intégration est considérablement réduite.

VI.2 APPLICATION AU PROJET

La réalisation finale aboutit à une structure multiprocesseurs du type bus commun (figure III.23)

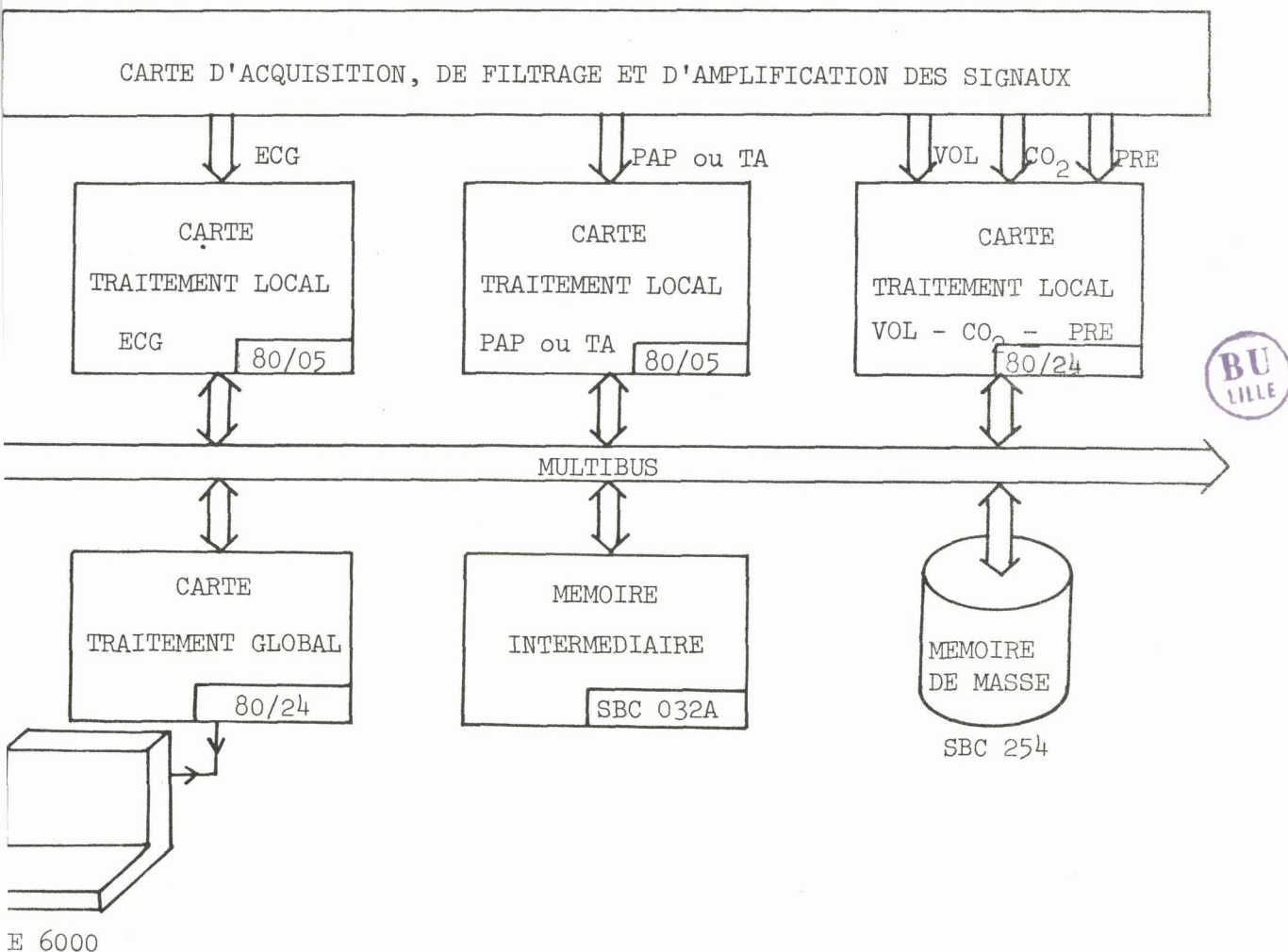


Figure III.23

Cette architecture offre l'avantage de respecter les contraintes de modularité matérielle en allouant un processeur à chaque traitement spécifique à une fonction ou signal physiologique (Traitement Local), et un processeur est chargé du Traitement Global, de gérer l'interface opérateur - système (Unité de Dialogue) et de coordonner l'ensemble (Mémoire de Masse, Mémoire Intermédiaire ...).

VII CONCLUSION

La méthodologie présentée est une méthode complète procédant par étape qui permet d'aboutir, à partir des spécifications, à une réalisation.

Elle repose sur des modèles de description et de comportement simple à comprendre et permet une approche progressive des problèmes de conception.

Pour augmenter l'efficacité de la méthodologie, un outil d'aide à la conception, pour les applications temps réel, est en cours de réalisation /CALV-83/.

Cet outil a pour but de guider et de fournir toutes les informations nécessaires au concepteur durant sa démarche de réflexion.

DEUXIEME PARTIE

PRISE DE DECISION

CHAPITRE IV

LES SYSTEMES EXPERTS

CHAPITRE IV :

LES SYSTEMES EXPERTS

<u>I INTRODUCTION</u>	p.	1
<u>II ARCHITECTURE D'UN SYSTEME EXPERT</u>	p.	1
* <u>Moteur d'inférence</u>	p.	1
* <u>La base de connaissances</u>	p.	2
* <u>La base de faits</u>	p.	2
* <u>Exemple</u>	p.	3
* <u>Les interfaces</u>	p.	3
<u>III CONNAISSANCE - FAITS - MOTEUR D'INFERENCE</u>	p.	3
<u>III.1 DICHOTOMIE CONNAISSANCES - EXPLOITATION</u>	p.	3
<u>III.2 LA CONNAISSANCE</u>	p.	7
<u>III.2.1 Représentation des connaissances</u>	p.	7
* <u>Les règles</u>	p.	7
* <u>Les métarègles</u>	p.	8
<u>III.2.2 Les types de règles de production</u>	p.	9
<u>III.2.3 Coefficient de vraisemblance</u>	p.	11
<u>III.2.4 Propriétés des règles de production</u>	p.	12
<u>III.3 LES FAITS</u>	p.	12
<u>III.4 MOTEUR D'INFERENCE</u>	p.	14
<u>III.4.1 Ordre du moteur d'inférence</u>	p.	14
<u>III.4.2 Utilisation des règles de production</u>	p.	14
<u>III.4.3 Stratégies de raisonnement</u>	p.	15
* <u>Chainage avant</u>	p.	15
* <u>Chainage arrière</u>	p.	17
<u>III.4.4 Cycle de fonctionnement</u>	p.	19
* <u>Phase d'évaluation</u>	p.	19
** <u>La restriction</u>	p.	20
** <u>Le filtrage</u>	p.	21
** <u>La résolution de conflit</u>	p.	21
* <u>Phase d'exécution</u>	p.	22
** <u>Activation des règles</u> en " largeur d'abord ".....	p.	22
** <u>Activation des règles</u> en " profondeur d'abord ".....	p.	23
<u>III.4.5 Efficacité des moteurs d'inférence</u>	p.	24

<u>IV</u>	<u>TYPE D'APPLICATION DES SYSTEMES EXPERTS</u>	p. 25
IV.1	CLASSIFICATION - DIAGNOSTIC	p. 26
IV.2	SURVEILLANCE - CONTROLE	p. 27
IV.3	RECHERCHE HEURISTIQUE	p. 28
IV.4	C.F.A.O.	p. 29
IV.5	E.A.O.	p. 30
IV.6	SYSTEME D'AIDE A LA CONCEPTION DE SYSTEMES EXPERTS	p. 31
<u>V</u>	<u>CONCLUSION</u>	p. 32

I INTRODUCTION

Dans les années soixante, l'Intelligence Artificielle s'était donnée comme objectif la description des mécanismes généraux modélisant le raisonnement humain.

Au regard des résultats peu convaincants (l'exemple de la traduction automatique le montre bien), cet objectif fut peu à peu abandonné comme trop ambitieux, au profit d'un but plus modeste : la modélisation du raisonnement d'un expert dans un domaine précis.

C'est dans cet optique, que depuis une quinzaine d'années, les premiers systèmes experts sont apparus.

Un système expert peut être défini comme un logiciel permettant la résolution d'une certaine catégorie de problèmes, pour lesquels on ne dispose pas d'algorithme répondant de manière efficace à leur résolution.

Pour cela il utilise une grande quantité de connaissances, spécifiques du domaine traité, acquises auprès des experts et un mécanisme de raisonnement proche du leur.

La caractéristique essentielle des systèmes experts est la séparation entre cette Connaissance et les mécanismes chargés de l'exploiter.

II ARCHITECTURE D'UN SYSTEME EXPERT

L'organisation d'un système expert se décompose en trois parties distinctes et élémentaires (Figure IV.1) :

- Le moteur d'inférence ou système de raisonnement
- La base de connaissances ou base de règles
- La base de faits ou espace de travail.

* Moteur d'inférence

Le moteur d'inférence est un mécanisme d'exploitation de la base de connaissances, chargé de la mise en oeuvre du savoir, permettant la " démonstration automatique ".

Il est conçu, le plus souvent, en vue d'applications multiples et doit donc pouvoir être appliqué à divers domaines.

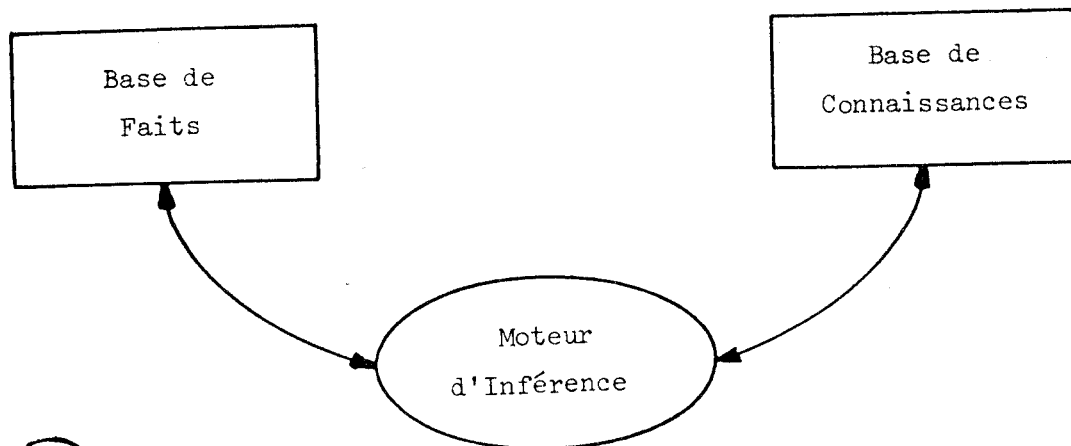


Figure IV.1

* La base de connaissances

La base de connaissances est, par contre, spécifique du domaine étudié.

Elle définit les informations que le système possède sur ce domaine.

Elle contient la connaissance qu'utilise l'expert pour la résolution des problèmes.

Pour cela elle est donc, non seulement formée de la connaissance formelle du domaine, mais aussi des connaissances sur la manière de l'appliquer, traduisant l'expérience et l'intuition de l'expert.

A cet effet, deux catégories sont généralement distinguées :

- la connaissance
- la métaconnaissance (ou connaissance sur la connaissance).

* La base de faits

La base de faits correspond à la description du problème que l'on demande au système expert de résoudre.

Elle constitue pour le contexte de l'application envisagée l'ensemble des données descriptives de ce problème.

Elle contient les faits connus concernant la solution.

* Exemple

- Problème à résoudre par le système expert :

Est ce que les gens sortent ?

- Connaissances sur le domaine :

(1) - *Si le soleil ne brille pas, les gens ne sortent pas.*

(2) - *Si il y a des nuages, le soleil ne brille pas.*

(3) - *Si il pleut, il y a des nuages.*

- Données descriptives du problème :

Il pleut.

Par l'application successive des règles (3), (2) et (1) le moteur d'inférence devra conclure, à partir du fait : "il pleut" que "les gens ne sortent pas", ce qui correspond à résoudre le problème.

* Les interfaces

Bien qu'on ne puisse pas dire que les interfaces soient indispensables à la fonction première des systèmes experts, qui est la résolution de problèmes, elles forment malgré tout un module nécessaire à la convivialité du système.

Elles définissent la communication Homme - Machine.

Elles permettent d'une part l'acquisition des faits, d'autre part, la modification de la base de connaissances par l'expert en fonction des progrès dans le domaine.

De plus, l'interface permet à l'utilisateur de juger de la validité du raisonnement du système, en lui offrant la possibilité d'accès au déroulement des connaissances utilisées par le système pour résoudre le problème. (Figure IV.2)

III CONNAISSANCE - FAITS - MOTEUR D'INFERENCE

III.1 DICHOTOMIE CONNAISSANCES - EXPLOITATION

La caractéristique essentielle des systèmes experts est la séparation très nette entre la connaissance et les méthodes de manipulation de celle-ci, à l'opposé de la programmation traditionnelle.

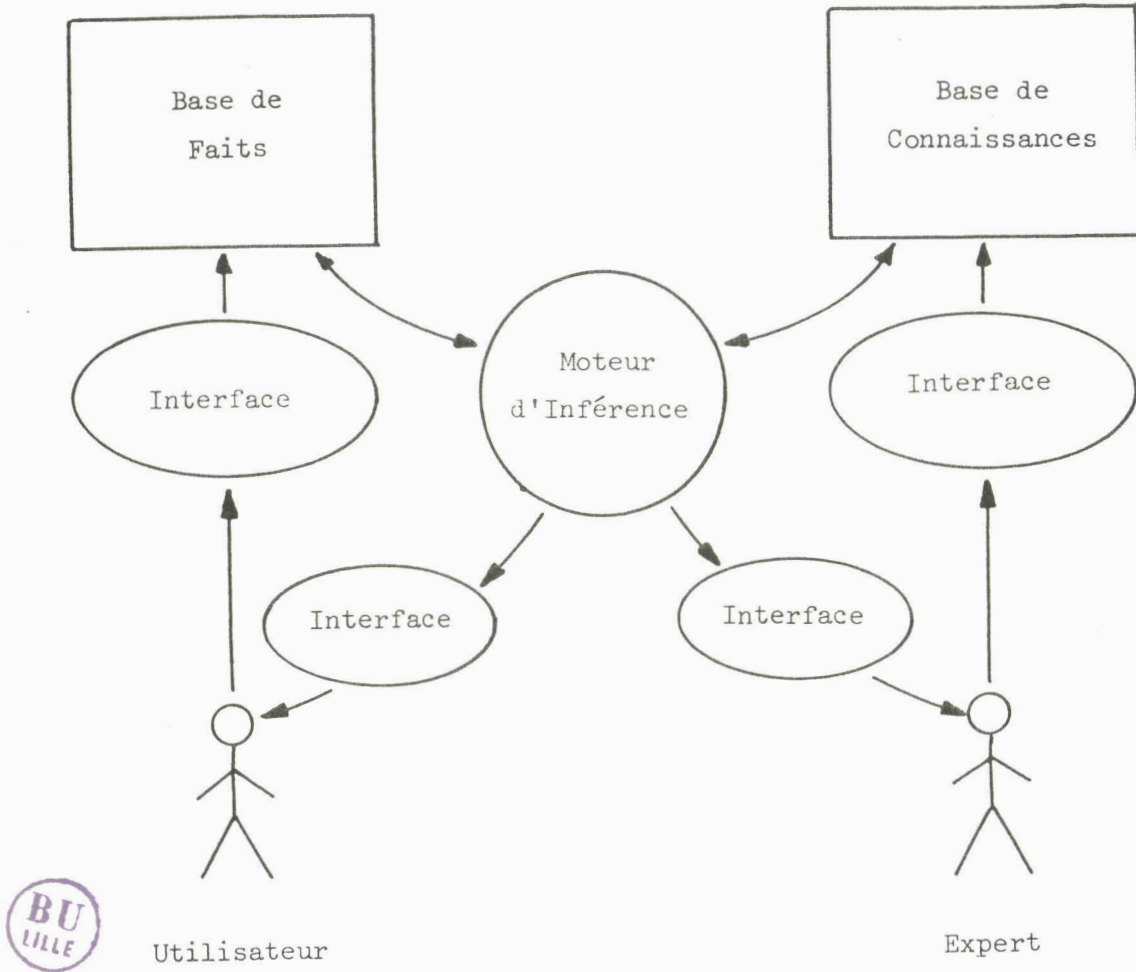


Figure IV.2

Lors de la conception d'un logiciel, répondant à une catégorie de problèmes, la connaissance est traduite dans les instructions et leur ordonnancement : la connaissance est représentée implicitement dans les procédures.

Cet ordonnancement doit être prévu à l'avance d'une manière algorithmique.

Ainsi les données sont toujours traitées d'une manière similaire suivant un cycle figé par l'algorithme de traitement.

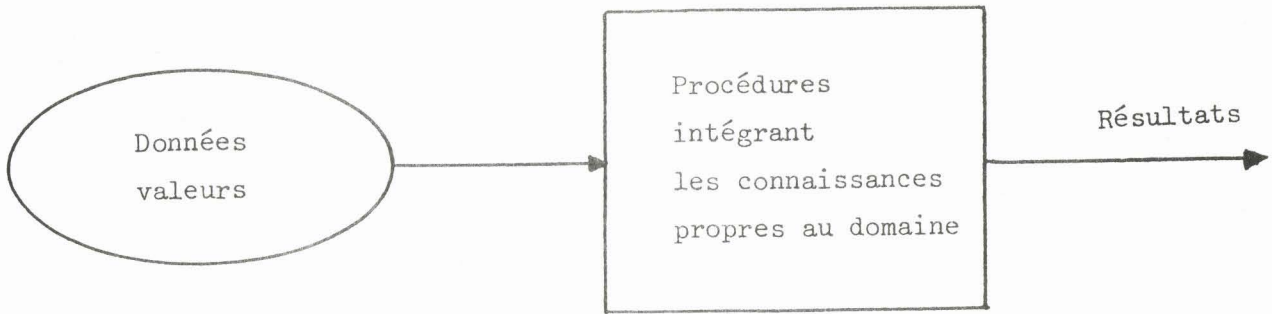


Figure IV.3

La programmation s'intéresse à la résolution de problèmes, bien définis, en recherchant des algorithmes efficaces où sont enfermés toutes les connaissances :

$$\text{Programme} = \text{Algorithme} + \text{Donnée}$$

Dans un système expert, le traitement des données (ou faits) ne passent pas par une séquence rigide d'instructions, mais c'est le mécanisme d'exploitation qui décide, en fonction de ces données, quelles connaissances doivent être appliquées sur un fait pour aboutir à la solution.

Le mécanisme d'exploitation, suivant les situations définies par l'ensemble des données, choisit l'ordre d'application des connaissances.

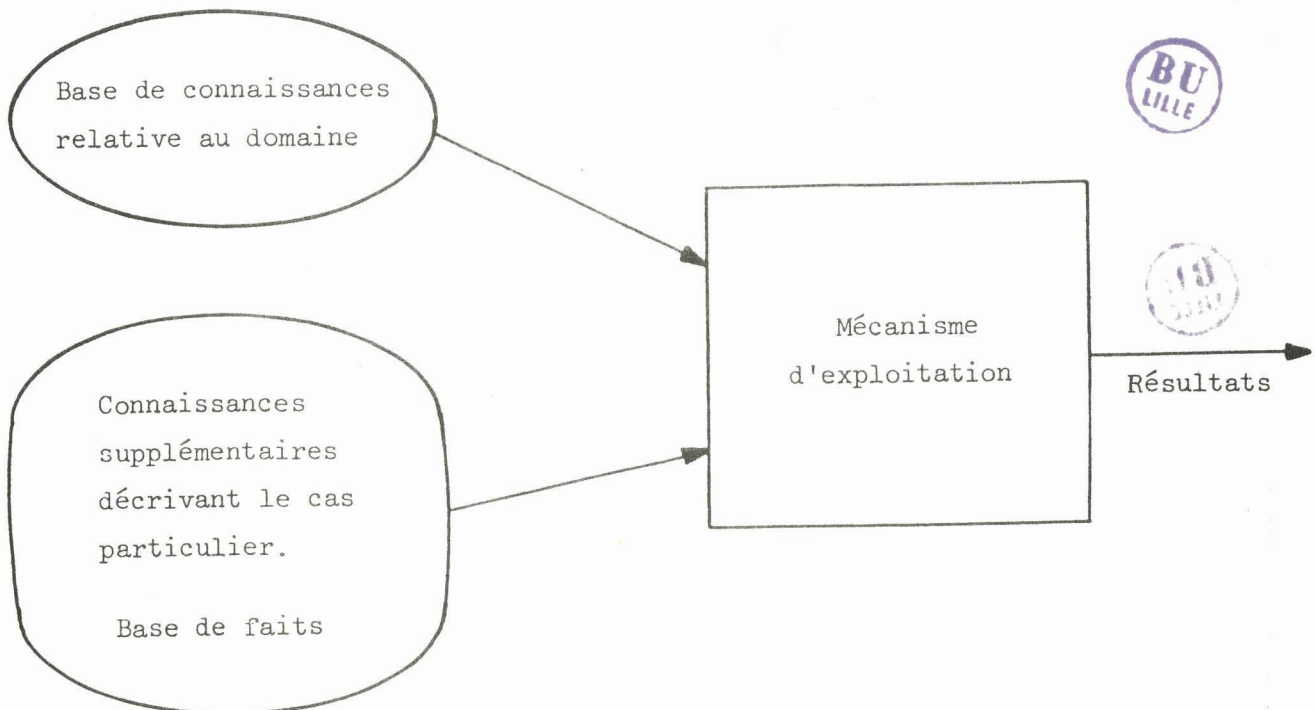


Figure IV.4

Ainsi dans un système expert les connaissances sont extérieures au programme, qui consiste à interpréter des données, présentes dans la base de faits, en fonction des connaissances situées dans la base de connaissances relatives au domaine.

Le mécanisme d'exploitation constitue essentiellement le programme.

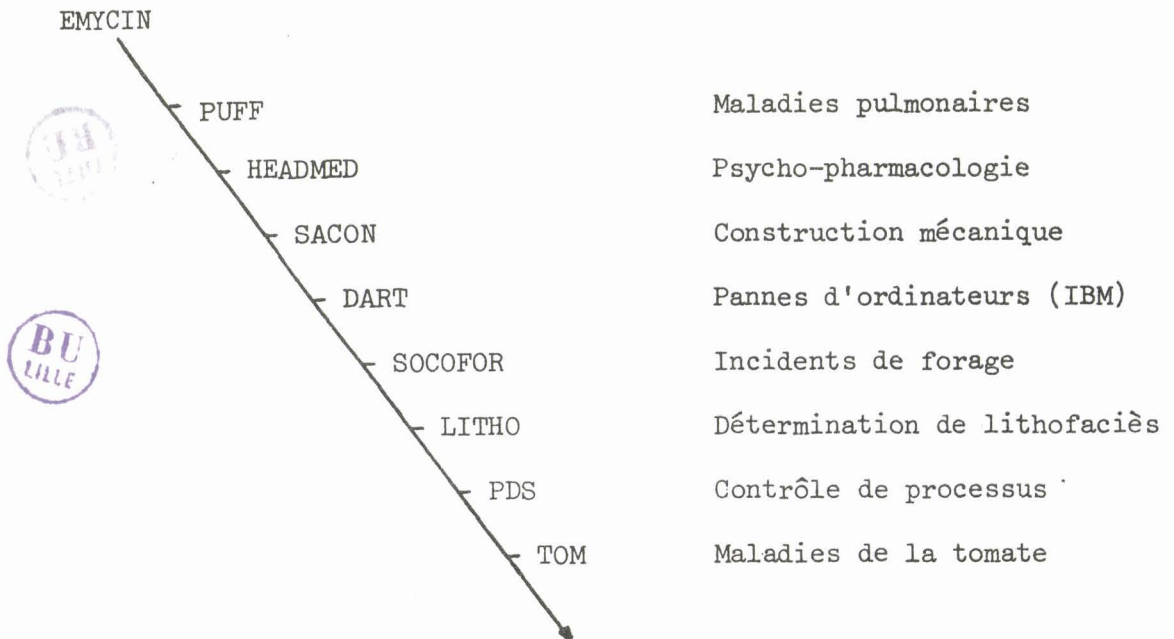
Cette structure offre différents avantages :

- L'évolution de la base de connaissances ne passe pas par un remaniement complet du programme mais par un accès facile à la base de connaissance grâce aux interfaces.

Ainsi elle peut être facilement modifiée en supprimant ou en rectifiant une information fautive ou en ajoutant une nouvelle connaissance.

- Un même moteur d'inférence peut être utilisé pour différentes applications. Il suffit pour cela d'instancier une connaissance au système général.

Par exemple MYCIN /SHOR-76/ a donné naissance à une famille de systèmes experts grâce à son moteur d'inférence EMYCIN /VANM-80/.



- Une même base de connaissances peut être utilisée par des systèmes différents suivants la fonction recherchée.
Par exemple le système GUIDON utilise les connaissances de MYCIN dans un but pédagogique en filtrant la connaissance exprimée par les règles de MYCIN par un jeu de règles pédagogiques /CLAN-79/.
- La justification du raisonnement suivi par le système se fait par la donnée du séquençement des connaissances utilisées.

Ainsi un système expert peut être caractérisé par :

Système expert = Connaissance + Faits + Moteur d'inférence

III.2 LA CONNAISSANCE

La base de connaissances représente le " Savoir " d'un système expert et en cela sa composition est déterminante dans l'efficacité d'un système expert : elle doit contenir une connaissance aussi précise, complète et détaillée que celle de l'expert.

Il est donc évident que la collaboration de ce dernier lors de la conception d'un système expert correspond à un point crucial.

Or si l'expert doit formuler sa connaissance et son expérience, il faut lui donner le moyen de l'exprimer facilement.

De plus, sa formulation n'est généralement pas adaptée à une exploitation automatique par un moteur d'inférence.

C'est le rôle du cogniticien de conduire l'expert à formaliser son savoir et de le transcrire dans une représentation utilisable par le système expert.

III.2.1 Représentation des connaissances

* Les règles

Le formalisme le plus couramment adopté pour la représentation des connaissances est celui des règles de productions (appelées aussi règle situation - action) *.

- - - - -

* D'autres formalismes sont également utilisés en Intelligence Artificielle pour représenter la connaissance tels que les réseaux sémantiques ou les frames.

Nous nous limiterons dans ce travail qu'à l'approche la plus employée des règles de production.

Ces règles de production sont représentées sous la forme :

" Si A Alors B " (1)

C'est à dire si la situation (ou condition ou prémisse) A est vérifiée alors l'action (ou conclusion) B peut être exprimée.

Les situations expriment les conditions d'application de la règle, les hypothèses devant être vérifiées pour tirer la conclusion B.

Dans la règle (1), de même que A peut représenter un certain nombre de situations, B peut exprimer un certain nombre de conclusions :

" Si A1 et A2 et ... et An Alors B1 et ... et Bm ".

La représentation des connaissances sous forme de règles de production définit ainsi des unités granulaires de connaissances appelées " *Granules* ou *Fragments* ".

L'ensemble des règles de productions constitue la base de connaissances, et à la différence de la programmation traditionnelle, ne représente pas une séquence d'instructions mais une collection d'entités de connaissance que le mécanisme d'exploitation utilise ou non en fonction du problème traité.

La connaissance, ainsi exprimée, est spécifiée sous une forme modulaire et déclarative : cette notion de représentation déclarative de la connaissance constitue une caractéristique essentielle des systèmes experts.

En principe, la connaissance n'a pas à être ordonnée, c'est au mécanisme d'exploitation qu'il appartient de déterminer l'ordre d'exploitation des règles.

* Métarègles

Certaines règles décrivent l'expérience, ou l'intuition de l'expert : ce sont les métarègles traduisant la *Métaconnaissance*.

Ces métarègles peuvent exprimer des stratégies de choix entre les règles et guident le système dans sa démarche de résolution en sélectionnant les règles à utiliser en priorité.

MYCIN /SHOR-76/ système expert destiné au diagnostic médical et plus particulièrement à l'analyse d'infections microbiennes, utilise de telles métarègles :

La première métarègle concerne le but recherché :
" Proposer une thérapie pour un patient ".

Métarègle :

Si l'on recherche une thérapie
Alors dans cet ordre, considérer les règles qui permettent de :

- 1 - Acquérir les informations cliniques sur le patient.
- 2 - Trouver quels organismes, s'il en existe, sont causes de l'infection.
- 3 - Identifier les organismes les plus vraisemblables.
- 4 - Trouver tous les médicaments potentiellement utiles.
- 5 - Choisir les plus adaptés en plus petit nombre.

Ainsi les métarègles jouent le rôle de fil d'Ariane dans la résolution de problème : elles permettent d'agencer, d'organiser l'ordre d'exécution de l'ensemble des règles.

III.2.2 LES TYPES DE REGLES DE PRODUCTION

Les règles de production peuvent être exprimées suivant deux types :

- La logique des propositions
- La logique des prédicats (logique du premier ordre).

Dans le calcul des propositions, les règles ne contiennent pas de variables, elles s'appliquent directement aux faits : prémisses et conclusions représentent des faits.

Si on applique ce formalisme au syllogisme d'Aristote, il se traduira par :

Si Les hommes sont mortels
et si Socrate est un homme
Alors Socrate est mortel

Par contre la logique des prédicats autorise l'utilisation de variables et de quantificateurs, permettant aux règles de s'appliquer à une classe de faits.

Le syllogisme précédent s'écrit dans ce formalisme :

Si Homme (x)
Alors Mortel (x) (1)

signifiant : « Si x est un homme Alors x est mortel ».

Il suffit donc d'appliquer cette règle au fait initial " Homme (Socrate) " : « Socrate est un homme » pour déduire la conclusion " Mortel (Socrate) " : « Socrate est mortel ».

Considérons l'exemple suivant :

Si Père (x) = y
et Mère (y) = z
Alors Grand-mère (x) = z

qui signifie « Si y est le père de x et si z est la mère de y Alors z est la grand-mère de x ».

Cette proposition représente une règle générale susceptible de fournir des informations de manière implicite /GOND-83/.

Ce dernier exemple caractérise mieux les possibilités de la logique du premier ordre.

En effet, dans le premier exemple, la variable ne s'adressait qu'à une seule classe de faits ou entités, il pouvait donc être décrit sans faire intervenir de variables.

La règle écrite dans la logique des prédicats :

Si Animal (x)
et si Bipède (x)
et si Sans plume (x)
Alors Humain (x)

peut se traduire aisément dans la logique des propositions car elle ne fait intervenir qu'une seule variable :

Si Individu est un animal
et si Individu est un bipède
et si Individu est sans plume
Alors Individu est un humain

Ainsi l'intérêt d'une règle écrite dans la logique des prédicats est qu'elle s'applique à une classe de faits, tandis qu'une règle, dans la logique des propositions, s'adresse à un seul fait.

Le calcul des prédicats offrent donc une puissance et une souplesse de représentation que n'a pas le calcul des propositions.

III.2.3 COEFFICIENT DE VRAISEMBLANCE

Le savoir humain que l'on cherche à représenter dans la base de connaissances d'un système expert est souvent entâché d'incertitude et d'imprécision.

L'incertitude porte sur la vérité ou la fausseté des propositions (exemple : *Il n'est pas certain qu'il fasse beau demain*), tandis que l'imprécision a trait au contenu de l'information; on tient alors compte du fait que les mots décrivant le savoir n'ont pas forcément une interprétation unique et précise, mais recouvrent un ensemble de valeurs, de paramètres sous-jacents dont on ne connaît souvent qu'approximativement les limites (exemple : *Cette personne est jeune*) /DUBO-85/.

Pour traduire cette imprécision ou cette incertitude, on affecte un coefficient de vraisemblance à chaque règle.

Ce coefficient peut revêtir différentes significations /PINS-81/ :

- Probabilité.
- Certitude (qu'a l'expert que le granule est vrai).
- Intérêt (de l'utilisation du granule à résoudre rapidement le problème).
- Acceptabilité (du granule dans le domaine traité par un ensemble d'experts).

La combinaison des coefficients de vraisemblance est le principal problème lié à leur utilisation car elle peut faire appel à différentes théories mathématiques suivant leur signification :

- théorie bayésienne
- logique floue
- théorie de la décision.

MYCIN utilise des règles telles :

Si la coloration est GRAM NEGAT
et si la morphologie en BATONNET
et si le patient est un HOTE A RISQUE
Alors on suggère (0.6) que c'est un PSEUDOMAS.

Où apparaît un coefficient de crédibilité, appelé aussi facteur d'atténuation. Il peut prendre les valeurs comprises entre -1 et +1, plus il est proche de +1 plus on affirme la conclusion.

III.2.4 PROPRIETES DES REGLES DE PRODUCTION

Les règles de production ont deux propriétés essentielles :

- Leur indépendance
- Leur nature déclarative.

- * Une règle ne pouvant faire référence à une autre, les règles sont indépendantes les unes des autres.

Ceci a l'avantage de pouvoir facilement ajuster la connaissance par addition, modification ou destruction de règles sans craindre d'interférence ou d'effets de bord sur les autres règles.

- * Les règles de production expriment une relation entre des faits sans prévaloir sur leurs applications futures, un mécanisme de raisonnement étant chargé de décider en fonction du contenu de la base de faits de déclencher telle ou telle règle.
Chaque règle représente ainsi une entité de connaissance.

III.3 LES FAITS

L'ensemble des données relatives au problème traité et fournies au système expert par l'utilisateur constitue la base de faits initiale : c'est l'énoncé du problème.

Ces faits vont pouvoir à partir de certaines règles inférer de nouveaux faits.

On appelle inférence la conséquence logique d'un ensemble de règles au regard des faits présents dans la base de faits.

* Exemple :

- R1 : Si il pleut Alors il y a des nuages
- R2 : Si il y a des nuages Alors le soleil ne brille pas
- R3 : Si le soleil ne brille pas Alors les gens ne sortent pas

Le fait « les gens ne sortent pas », conséquence logique des règles R1, R2, R3 et du fait « il pleut », pourra être inféré si le fait « il pleut » est présent dans la base de faits.

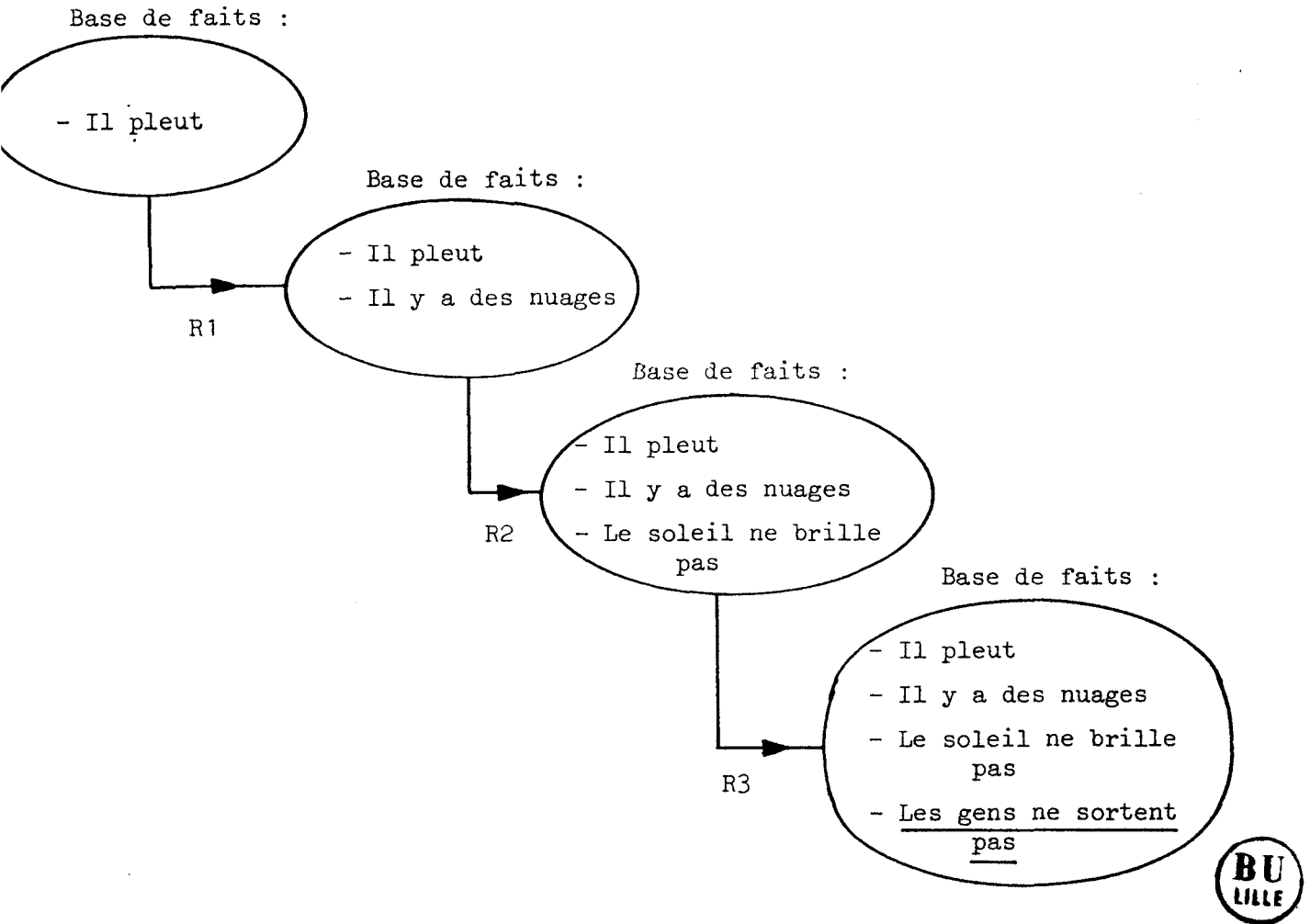


Figure IV.6

En effet si la base de faits contient la donnée « il pleut » par l'activation de la règle R1 le fait « il y a des nuages » sera validé et placé dans la base de faits. Puis celle-ci s'incrémentera des faits « le soleil ne brille pas » et « les gens ne sortent pas » par l'activation successive des règles R2 et R3 . (Figure IV.6)

Au fur et à mesure de l'activation des règles et de l'inférence des faits, la base de faits s'enrichit de données supplémentaires permettant d'affiner la connaissance du problème traité.

C'est dans ce sens que la base de faits correspond pour le système expert à un espace de travail à dimension variable puisqu'il contient les faits fournis par l'utilisateur ainsi que les faits inférés ou déduits de son raisonnement. /GROS-85/

La base de faits représente l'état courant de la connaissance du cas traité.

III.4 MOTEUR D'INFERENCE

Le moteur d'inférence est chargé d'appliquer les fragments de connaissance, situés dans la base de connaissances, aux cas particulier décrit par la base de faits, afin d'inférer de nouveaux faits puis d'aboutir à la résolution du problème.

III.4.1 ORDRE DU MOTEUR D'INFERENCE

Nous avons vu que les fragments de connaissance, représentés sous forme de règle de production, pouvaient faire intervenir deux formalismes de logique :

- la logique des proposition
- la logique des prédicats.

L'utilisation de l'une ou l'autre définit ce qu'on appelle couramment l'ordre du moteur d'inférence :

Si on applique la logique des propositions on parle de moteur d'inférence d'ordre zéro, par contre l'utilisation de la logique des prédicats implique un moteur d'inférence d'ordre un.

La notion d'ordre est directement liée à la logique mathématique où on parle d'ordre zéro pour le calcul propositionnel et de logique du premier ordre pour celui des prédicats.

III.4.2 UTILISATIONS DES REGLES DE PRODUCTION

Le formalisme des règles de production, ne limite pas leur utilisation dans le sens direct, mais la négation de la proposition peut également être appliquée.

Deux modes sont ainsi définies :

- le " modus ponens " :
Si $A \rightarrow B$ et si A est vrai
Alors on peut conclure que B est vrai.
- le " modus tollens " :
Si $A \rightarrow B$ et si B est faux
Alors on peut conclure que A est faux.

Ainsi LAURIERE /LAUR-82b/ à partir de l'exemple :

« *Tout homme est faillible* »

décrit plusieurs utilisations de cette règle dont la plus immédiate est :

- 1) - « *Une entité est faillible dès que je peux prouver que c'est un homme* »

L'utilisation d'une métaconnaissance donne une deuxième solution :

- 2) - « *Si je veux prouver que quelque chose est faillible, un sous but possible est de montrer que c'est un être humain* »

Alors que ces deux énoncés utilisent le " modus ponens " l'utilisation du " modus tollens " donnerait les deux formes duales :

- 3) - « *Si une entité est infallible, ce ne peut être un homme* »
- 4) - « *Si je désire prouver que quelque chose n'est pas un homme, je peux tenter de montrer qu'elle est infallible* »

III.4.3 STRATEGIES DE RAISONNEMENT

La manière dont les règles sont déclenchées définissent le type du moteur d'inférence. Les deux mécanismes les plus connus sont :

- le chaînage avant
- le chaînage arrière.

* Chaînage avant ou mode progressif ou raisonnement guidé par les faits

En chaînage avant on cherche à déclencher les règles dont les prémisses sont vérifiées dans la base de faits et on y ajoute alors les conclusions.

Cette résolution est poursuivie tant qu'il existe des règles activables ou qu'une solution est proposée.

* Exemple

Soit la base de règles :

- R1 : F et C -> L
- R2 : B et K -> L
- R3 : I et N -> L
- R4 : B et M -> F
- R5 : D et E -> F
- R6 : H -> C
- R7 : A et D -> K
- R8 : J et G -> N
- R9 : A et B -> J

Cette base de règles revient à développer l'arborescence de la Figure IV.7 .

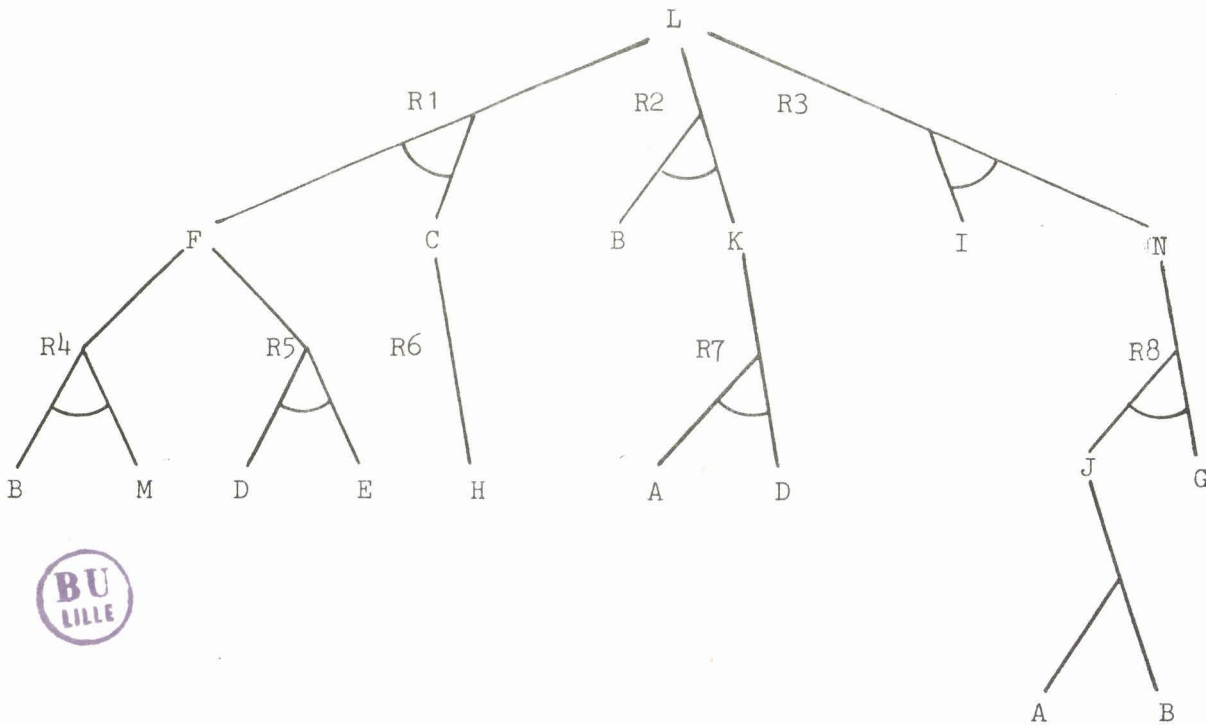


Figure IV.7

Le but du système est de déduire si le fait L est vrai à partir de la base de fait : (M, B, A, D)

Toutes les règles applicables vont donc inférer de nouveaux faits.

On aura ainsi la construction d'un arbre partant des feuilles (Figure IV.8) et une base de faits qui s'incrémentera :

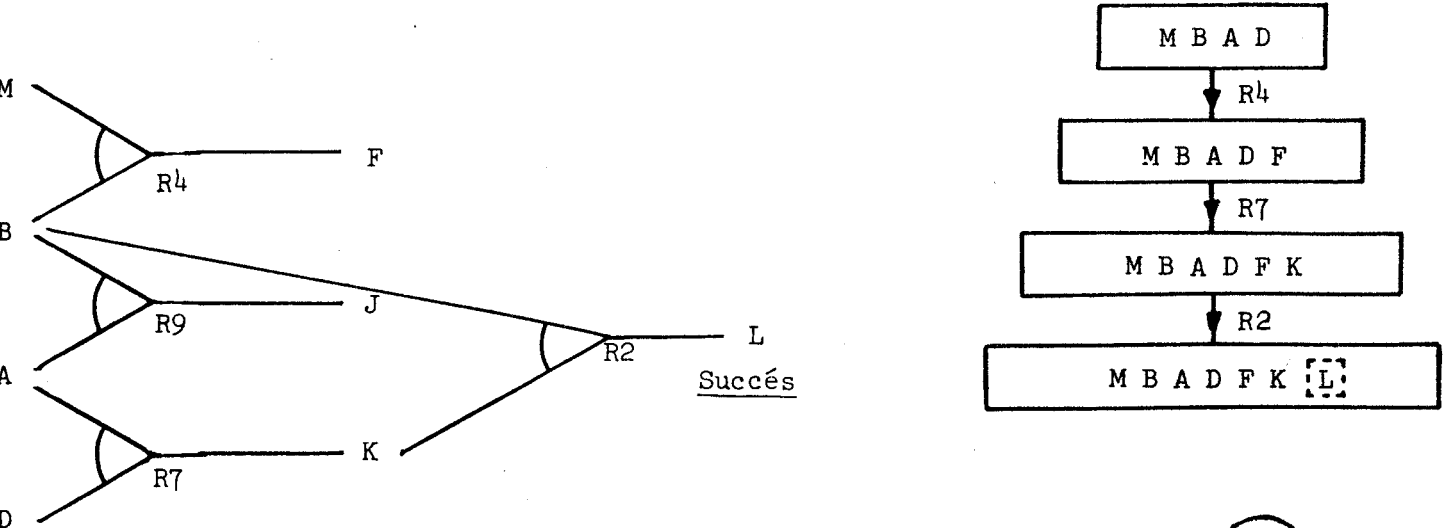


Figure IV.8



La règle R4 inférera le fait F (inutile pour la démonstration). La règle R7 fournira le fait K et grâce à cette inférence, le fait L sera déduit par la règle R2.

La règle R9 était applicable car les faits A et B étaient vrais mais comme le fait L a été déduit avant qu'elle ne soit choisie par le moteur d'inférence, l'inférence du fait J n'a pas eu lieu.

* Chainage arriére ou mode régressif ou raisonnement guidé par les buts

En chaînage arriére on cherche à démontrer si un fait est vérifié. Le moteur d'inférence s'intéresse donc aux règles dont les conclusions comportent ce fait.

Lorsqu'une règle répond à ce critère les prémisses de cette règle deviennent alors des sous buts à vérifier. Si ils font partie de la base de faits alors le but est vérifié. Par contre si un des faits n'est pas vrai alors le système le considère comme nouveau but et il reitère ce procédé tant qu'il a pu prouver la solution ou la nier.

* Reprenons l'exemple précédent

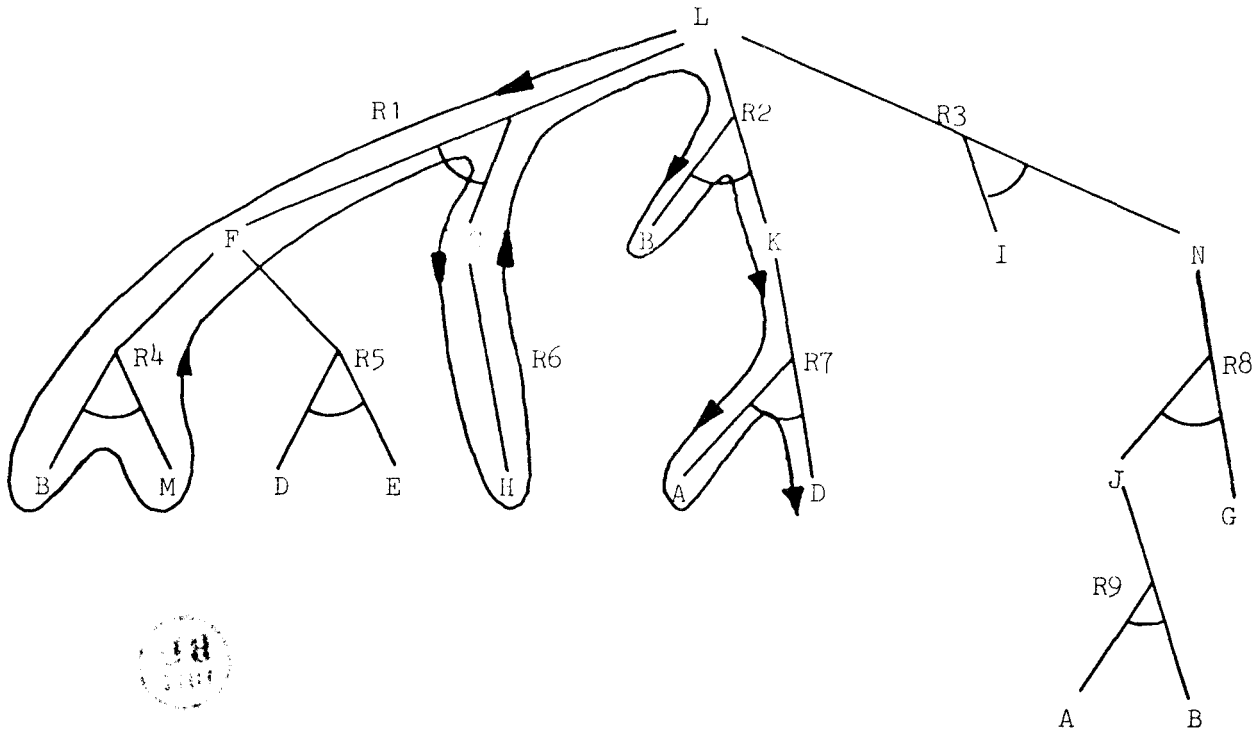


Figure IV.9

L appartient à la partie conclusion de R1, donc les faits F et C deviennent des sous buts à vérifier.

F sera vérifié par la règle R4, du fait C reste le seul but à vérifier. C étant la conclusion de R6, H est un but à vérifier.

Comme H n'appartient pas à la base de faits et est un noeud terminal, cette branche de l'arbre ne permet pas de conclure et donc le système passera à un chemin voisin.

La règle R2 fournit deux nouveaux sous buts (ou buts courants) B et K, B est vérifié donc K est le but à atteindre. La règle R7 validera ce but étant donné que A et D appartiennent à la base de faits.

De ce fait le but K est vrai donc L est vérifié.

En chaînage avant, un fait qu'il soit faux ou inconnu représente la même chose pour le système.

En effet, si $A \rightarrow B$ et si A est faux on ne peut tirer aucune conclusion sur la véracité de B .

Par contre en chaînage arrière, un fait faux prend toute sa signification :

si $A \rightarrow B$ et si B est faux on peut conclure que le fait A est faux.

Ainsi en chaînage avant, le système travaille sur une logique binaire (deux états : Vrai et Faux ou Inconnu) tandis qu'en chaînage arrière la logique est ternaire (trois états : Vrai et Faux et Inconnu).

En chaînage avant, le moteur d'inférence procède par saturation de la base de connaissances : il infère tous les faits pouvant être déduits à partir de la base de faits.

De ce fait il présente le risque d'explosion combinatoire qui n'existe pas dans le mode en chaînage arrière.

Par contre le chaînage arrière est friand en place mémoire (il doit "empiler" tous les sous buts à vérifier) et il est coûteux en temps.

III.4.4 CYCLE DE FONCTIONNEMENT

Le moteur d'inférence chargé d'appliquer les règles activables en fonction de l'état courant de la base de faits, contrôle l'ordre d'exécution des règles.

Pour cela il doit : (figure IV.10)

- évaluer la règle à activer
- à l'issu de cette phase d'évaluation, exécuter la règle
- réitérer ce cycle de base tant qu'une condition d'arrêt n'a pas été détectée (solution trouvée, ensemble des règles activables vide ...).

* La phase d'évaluation

Trois étapes sont à distinguer dans cette phase :

- La restriction
- Le filtrage
- La résolution de conflit.

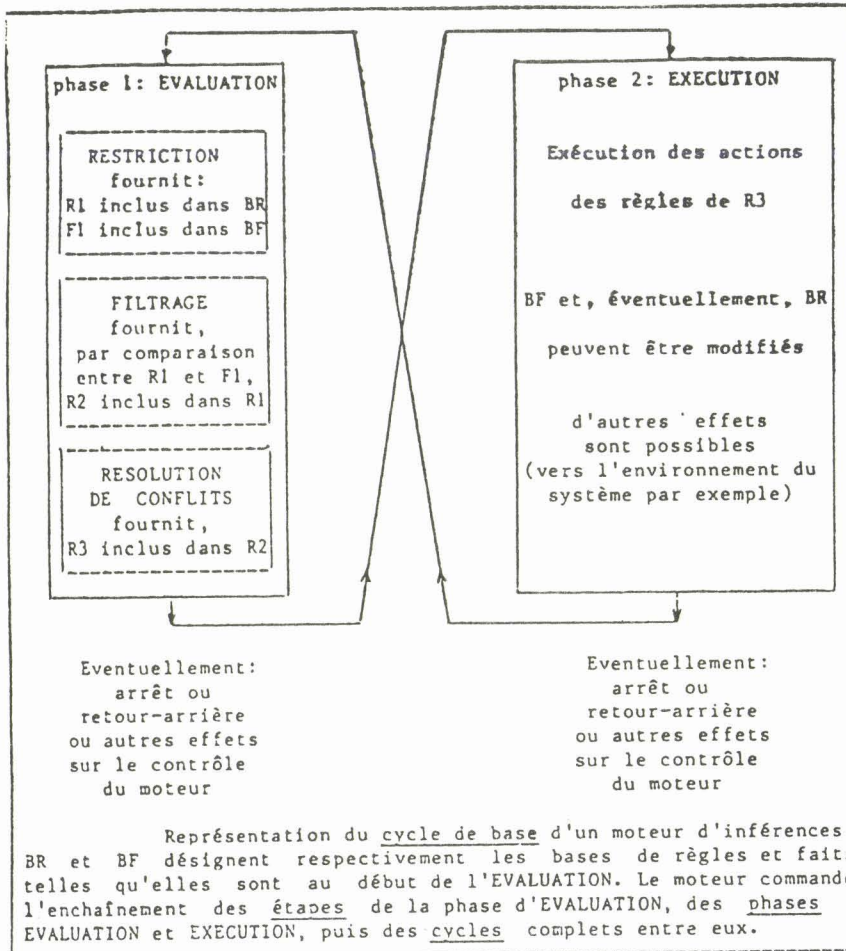


Figure IV.10/FARR-85/

** La restriction

Cette étape sélectionne en fonction de l'état de la base de faits (BF) et de la base de règles (BR), un sous ensemble de BF (F1) et un sous ensemble R1 de BR qui sont à prendre en compte en priorité pour l'étape de filtrage.

Par exemple, pour un système expert d'aide au diagnostic médical, il serait inutile d'envisager les règles et les faits concernant un mal de tête dans le cas où le patient souffre d'une douleur au pied.

La restriction cherche donc à privilégier un ensemble de faits et/ou de règles face à leur ensemble respectif : elle vise à distinguer des terrains d'applications.

** Le filtrage

Face au sous ensemble F1 (chainage avant) et/ou d'un ensemble de faits à confirmer (chainage arrière), le moteur d'inférence va isoler un ensemble de règles R2 (inclus dans R1) dont les conditions d'activations sont satisfaites.

Cet ensemble R2 est appelé " ensemble de conflit ".

** Résolution de conflit

Le moteur d'inférence doit déterminer, à partir de R2, l'ensemble R3 des règles à appliquer effectivement.

Différentes solutions sont choisies suivant les systèmes :

- L'ensemble des règles est appliqué exhaustivement (utilisé pour certains diagnostics afin de ne pas oublier un fait ou une règle pouvant entraîner une erreur par omission)
- Choix de la première règle rencontrée (oblige l'expert et le cognaticien à définir un ordre dans la base de connaissances et il y a donc une perte partielle de la nature déclarative des connaissances car elles ne peuvent plus être rentrées en " vrac ")

Mais le plus souvent la résolution de conflit est confiée à l'application d'un critère de sélection qui peut définir :

- Une priorité sur les règles et sur les faits
- Un facteur discriminant d'une règle ou d'un fait sur leur ensemble respectif
- La fiabilité d'une règle
- Le coût de l'emploi d'une règle
- ...

Le contrôle par métaconnaissance est sans conteste le plus riche car le mécanisme de sélection est alors mis dans le même formalisme que les règles et ces métarègles, ainsi définies, sont alors manipulées par le moteur d'inférence.

De plus leur modification se fait au même titre que celle des règles, sans intervenir sur le programme. (Un exemple de métarègle a été présentée au III.2.1 avec la première métarègle de MYCIN).

En outre la possibilité d'inclure des métarègles permet à un expert d'ajouter des connaissances moins formelles relevant de son expérience, de son intuition, voir de ses convictions caractéristiques de son " savoir-faire ".

** La phase d'exécution

Dans la phase d'exécution, le moteur d'inférence est chargé d'activer les règles sélectionnées dans R3.

Activer une règle consiste à développer son arborescence d'activation dans laquelle les noeuds correspondent aux différentes instanciations des conditions.

Si R3 est vide, le moteur d'inférence peut s'arrêter, il est dit à régime de contrôle irrévocable (irrevocable control regime) /NILS-80/.

D'autres moteurs reviennent sur une résolution de conflit antérieur, ils opèrent en retour-arrière (backtracking).

Dans le cas où R3 possède plus d'un élément, différentes stratégies sont considérées pour activer les règles.

- Activation des règles en " Largeur d'abord "

Dans ce cas on s'attache, au regard de la base de faits complète, à activer toutes les règles susceptibles de l'être :

Un premier niveau est développé puis les autres niveaux sont construits au fur et à mesure.

Exemple :

Soit la base de règles :

R1 :	F et C	->	L
R2 :	B et K	->	L
R3 :	I et N	->	L
R4 :	B et M	->	F
R5 :	D et E	->	F
R6 :	H	->	C
R7 :	A et D	->	K
R8 :	J et G	->	N
R9 :	A et B	->	J

et la base de faits initiale : (B, M, A, D)

Le but est de déduire si le fait L est vrai ou non.

Le premier ensemble de règles applicables est (R4, R7, R9).

Les faits inférés seront donc F, J et K et ainsi la base de faits devient (B, M, A, D, F, J, K)

Lorsque toutes les règles sont appliquées, on reconsidère la base de règles afin de déterminer les règles activables pour le niveau supérieur.

R2 permet d'inférer L. (figure IV.11)

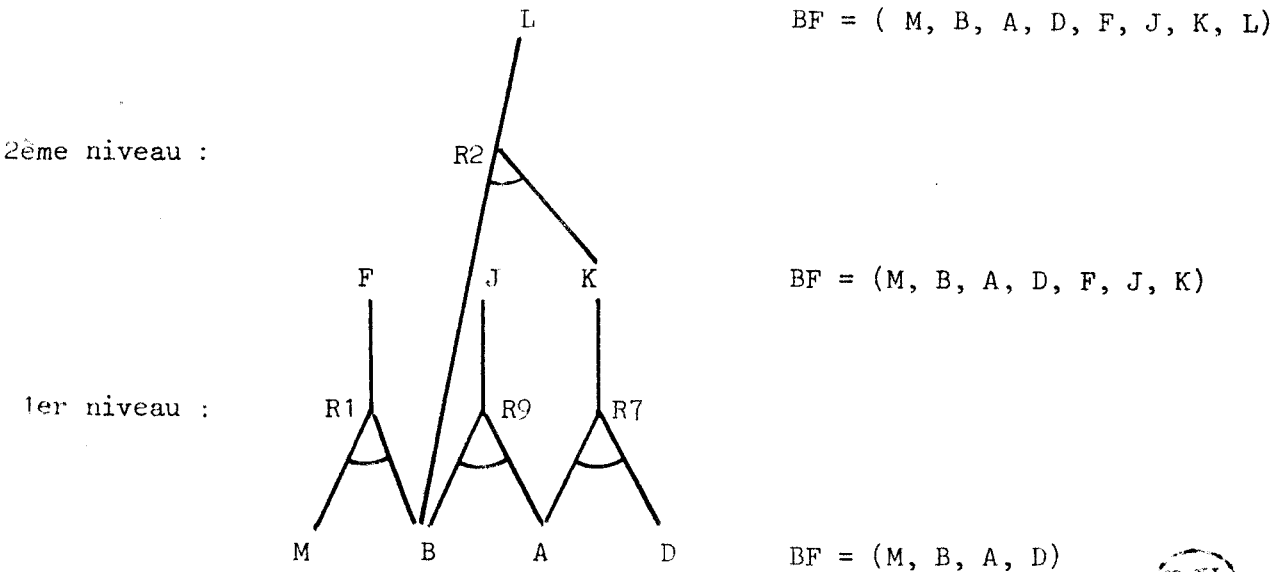


Figure IV.11



Tous les faits du niveau $n + 1$ ne peuvent être inférés tant que tous ceux du niveau n ne l'ont été.

- Activation des règles en " Profondeur d'abord "

Dans ce cas on s'attache à un fait et on regarde quelles sont les règles applicables. A partir de ce fait particulier on développe l'arborescence d'activation qu'il induit et de la famille de faits qui lui est associée.

Cette famille est la collection des faits reliés, à ce fait élu, par leur appartenance commune aux conditions d'application des règles activables et/ou parce qu'ils font partie des conclusions de ces règles qui ont été activées.

A partir de l'exemple précédent, si on choisit le fait A, son arborescence d'activation est : (figure IV.12)

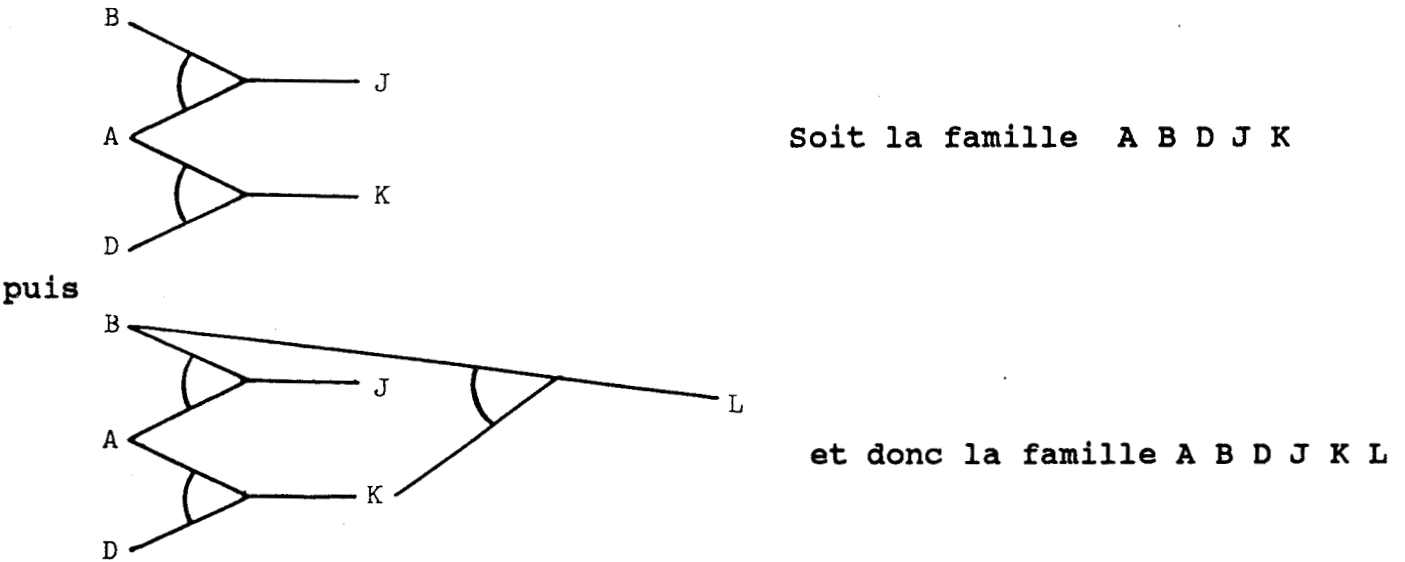


Figure IV.12



L'exemple développé pour illustrer le chaînage arrière (figure IV.9) fonctionne aussi en profondeur d'abord : on essaie de vérifier la véracité des sous buts jusqu'aux noeuds terminaux de l'arborescence avant de parcourir un autre chemin vérifiant le but premier.

III.4.5 EFFICACITE DES MOTEURS D'INFERENCE

Un moteur d'inférence est défini par différentes caractéristiques :

- Ordre du moteur (ordre zéro ou ordre un : dépend de la représentation des connaissances adoptée)
- Mode de résolution de conflit (basé en général sur le caractère discriminant des règles à résoudre rapidement le problème)
- La possibilité de retour-arrière
- Le type d'activation des règles (Largeur d'abord ou Profondeur d'abord)

Ces différents principes définissent, en partie, les performances d'un moteur d'inférence et peuvent donner lieu à des aménagements très variés en fonction de l'application.

Cependant le temps d'exécution d'un cycle de base (figure IV.10) est défini en nombre d'inférences par seconde (ou LIPS = Logical Inférence Per Seconde).

Le nombre de LIPS constitue un critère d'efficacité des moteurs d'inférence et il est dans l'objectif de nombreuses équipes de recherche de tenter de l'augmenter.

De même il est dans l'intention du projet japonais de " Cinquième Génération " d'ordinateur de concevoir des circuits et des architectures de machines permettant d'atteindre des vitesses de l'ordre du million voir du milliard de LIPS...

IV TYPE D'APPLICATION DES SYSTEMES EXPERTS

C'est vers la fin des années soixante, par l'intermédiaire des systèmes experts, que l'Intelligence Artificielle est sortie des laboratoires de recherche pour " s'attaquer " à des problèmes d'intérêts pratiques : tel DENDRAL /FEIG-71/ dont le domaine d'application est la chimie organique, ou MYCIN /SHOR-76/ pour le diagnostic médical.

En effet les domaines qui nécessitent une grande quantité de connaissances, pouvant évoluer, et dont certaines font appel à des heuristiques plus qu'à des lois scientifiques (et donc difficiles à formaliser sous un type algorithmique) permettant à un expert de traduire son " savoir faire ", sont des terrains privilégiés pour les systèmes experts.

Comme nous l'avons vu précédemment, les caractéristiques essentielles des systèmes experts sont :

- Une grande quantité de connaissances sur un domaine précis
- Une connaissance sur la manière de les utiliser (métaconnaissance par exemple)
- Les connaissances sont découpées sous formes de fragments indépendants et exprimées de manière déclarative.
- L'indépendance des connaissances et du mécanisme de manipulation
- Le caractère évolutif de la base de connaissances
- La stratégie de résolution fonction du problème à traiter et non figée dans un algorithme

- La possibilité d'intervention de l'homme dans le raisonnement.

Ces différentes caractéristiques permettent de deviner les diverses applications où les systèmes experts sont susceptibles d'intervenir.

On peut tenter de les regrouper en différentes catégories :

- * Classification, diagnostic
- * Surveillance-contrôle
- * Recherche heuristique de solution
- * C.F.A.O.
- * E.A.O.

IV.1 CLASSIFICATION - DIAGNOSTIC

On peut définir un diagnostic comme étant la classification d'un objet dans un événement, une situation, à partir d'informations incertaines concernant ces caractéristiques.

Dans ce sens, le diagnostic fait généralement appel à la théorie de la décision ou aux statistiques, obligeant l'estimation de fonctions probabilistes.

Les systèmes experts abordent ces problèmes de diagnostics en substituant l'estimation de ces fonctions par la connaissance et le jugement des experts humains.

Le domaine médical est un terrain riche pour l'applications des systèmes experts pour l'aide au diagnostic.

INTERNIST /POPL-75/ est un programme d'aide au diagnostic couvrant environ 80 % de la médecine interne. Il s'intéresse à environ 4000 manifestations de 500 maladies différentes, ce qui en fait l'application la plus ambitieuse (une manifestation peut être un signe, un symptôme, un résultat d'examen ou un autre diagnostic).

INTERNIST a montré ses capacités en diagnosticant correctement des cas considérés comme particulièrement difficiles.

De même MYCIN /SHOR-76/ est un système d'aide au diagnostic des infections bactériennes et à la sélection d'une thérapeutique appropriée pour un malade. Il a été

conçu pour aider les médecins non spécialistes dans la prescription des antibiotiques.

Il utilise environ 500 règles de production auxquelles sont associées un facteur de crédibilité. Les informations sont fournies par l'utilisateur au fur et à mesure que le système les réclame.

MYCIN a été à la base de plusieurs applications tel :

PUFF /AIKI-80/ interprétation des résultats d'exploration fonctionnelle respiratoire.

CLOT /BENN-80/ application au problème de coagulation

HEADMAD /HEIS-78/ appliqué à la psychopharmacologie en pratique psychiatrique.

D'autres domaines que la médecine utilise les systèmes experts pour les diagnostics.

Ainsi /PARC-82/ présente un système expert pour une aide au diagnostic sur réacteurs à neutrons, DART /BENN-81/ est un système expert pour pannes de matériels et de systèmes ...

PROSPECTOR /KONO-79/, spécialisé dans la recherche prétérolière et la prospection minière, a découvert autour du Mount Tolman (état de Washington) deux mines séparées de molybdène.

IV.2 SURVEILLANCE - CONTROLE

VM /FAGA-79/ (Ventilator Monitoring) est un système expert d'aide au personnel soignant en réanimation, il interprète des données quantitatives issues des signaux des différents capteurs dans le cadre d'une unité de soins intensifs. Ces données sont utilisées pour piloter un respirateur artificiel.

Il a quatre tâches principales :

- détecter les erreurs de mesure
- générer les alarmes
- établir un bilan
- proposer une thérapeutique

La connaissance est représentée sous forme de règles de production semblables à celles de MYCIN.

SU/X /NII -78/ est un système expert permettant l'identification et la localisation des objets émetteurs dans l'espace.

PATREC MDX /MITT-80/ est une aide intelligente à la gestion de base de données.

ARGOS II /CAYR-79/ est un système expert pour le contrôle de robots de troisième génération. Il est constitué d'un générateur de plans d'actions et d'un moniteur d'exécution des plans.

IV.3 RECHERCHE HEURISTIQUE

La théorie des graphes, des jeux font bien souvent appel à une recherche heuristique de la solution.

Généralement ces problèmes sont caractérisés par l'existence de deux méthodes systématiques :

- * Une générant les solutions candidates
- * L'autre testant la validité de ces solutions.

Dans bien des cas le nombre de solutions est si grand qu'une recherche exhaustive n'est pas possible. Aussi fait-on appel à une recherche heuristique c'est à dire à une stratégie permettant de réduire considérablement le nombre de solutions sans en compromettre la qualité.

Dans ce sens, les systèmes experts trouvent dans ces domaines des potentialités d'applications de par l'utilisation des métaconnaissances.

DENDRAL /FEIG-71/, un des premiers systèmes experts, utilise des connaissances pour limiter ses recherches.

DENDRAL est un système expert générant des représentations possibles de molécules organiques à partir des données d'un spectromètre de masse, d'un RMN et de contraintes fournies par l'utilisateur.

Le programme tourne suivant un cycle :

- 1) à partir des résultats, extraire des hypothèses générales sur la solution,
- 2) génération systématique des structures satisfaisant ces hypothèses,

- 3) test des structures proposées en prédisant les spectrographes de masse et en rejetant celles qui ne correspondent pas aux résultats.

METADENDRAL /BUCH-78/ fut conçu pour améliorer les performances de DENDRAL.

A partir d'exemples de spectrographes il doit fournir un ensemble de règles devant être intégré à DENDRAL : c'est donc un système expert produisant une base de connaissances.

METADENDRAL a non seulement fournit un ensemble de règles équivalent à celui formalisé par les experts mais aussi a trouvé des règles nouvelles appréciées par les experts en chimie organique.

Le programme de WATERMAN /WATE-70/ est appliqué à l'apprentissage d'heuristique au poker, ces heuristiques, sous forme de règles de production, sont séparées du programme qui doit, en cours de partie, les corriger ou en construire de nouvelles.

IV.4 CONCEPTION ET FABRICATION ASSISTEE PAR ORDINATEUR

La Conception et la Fabrication Assistée par Ordinateur (CFAO) constitue un domaine important des systèmes experts.

Dans cet optique EL /SUSS-75/ est un système expert pour l'aide à la conception en circuits électriques. Il utilise des règles de production pour représenter les principes généraux telle la loi d'Ohm, ou les caractéristiques des composants, permettant ainsi l'analyse des circuits.

De même l'utilisateur peut modifier le circuit et voir les conséquences du changement sans devoir réanalyser le circuit entier.

GARI /DESC-81/ est un système expert d'aide à la conception de gammes d'usinages à partir de règles pouvant exprimer des conseils fournis par des spécialistes.

R1 /DERM-81/ est un système expert d'aide à la configuration d'ordinateurs, conçu chez Digital Equipment Corp. pour suggérer des configurations complètes des systèmes VAX 11 répondant aux demandes de l'utilisateur.

IV.4 ENSEIGNEMENT ASSISTE PAR ORDINATEUR

Les systèmes traditionnels d'Enseignement Assisté par Ordinateur (EAO) possèdent une solution programmée une fois pour toute et donc sont incapables de répondre aux questions de l'étudiant, si celles-ci n'ont pas été prévues dans l'algorithme.

Un système d'EAO ne peut être efficace que s'il possède la compétence des experts du domaine qu'il enseigne. C'est pourquoi grâce à leur raisonnement "calqué" sur celui de l'expert, à leur possibilité de l'expliquer et à leur grande quantité de connaissances les systèmes experts peuvent être appliqués efficacement dans l'EAO.

De plus les métaconnaissances, traduisant le savoir faire de l'expert, c'est à dire la stratégie d'utilisation des connaissances et de résolution du problème, permettent aux systèmes experts d'enseigner la compétence du spécialiste; ce qui est primordial au sens où bien souvent les étudiants possèdent les connaissances nécessaires mais sont incapables de les mettre en oeuvre.

En outre certains domaines nécessitent d'enseigner des modes de raisonnement, ou de résolution de problèmes tels les mathématiques, la médecine ...

GUIDON /CLAN-79/ est un système expert qui exploite la connaissance de MYCIN pour enseigner à la fois les signes et les stratégies de diagnostic.

La base de connaissances est augmentée d'environ 200 règles dites *pédagogiques* permettant de guider le dialogue avec l'étudiant et de présenter les stratégies de diagnostic.

La liste présentée en annexe regroupe suivant les catégories présentées ci-dessus différents systèmes experts existants. Cette liste n'est évidemment pas exhaustive, étant donné le nombre considérable de systèmes experts, elle a pour but de fournir quelques références aux personnes intéressées.

Cette liste présente deux systèmes experts dits de *simulation* :

* le système ACT /ANDE-79/ conçu pour la simulation du comportement humain à partir d'expériences en psychologie

* le système POLITICS /CARB-78/ simulant le comportement humain face aux événements politiques.

De même une classe de système d'aide à l'acquisition de connaissances et à la conception de systèmes experts est présentée.

IV.6 SYSTEME D'AIDE A LA CONCEPTION DE SYSTEME EXPERT

De nombreux systèmes ont été mis au point afin de faciliter la conception des systèmes experts.

Entre l'acquisition manuelle des connaissances et l'apprentissage automatique, les chercheurs ont développé des outils facilitant la phase d'acquisition des connaissances.

TEIRESIAS /DAVI-79/ est un de ces systèmes, il supervise, d'une manière interactive, la construction de la base de connaissances d'un système expert. En dialogant avec l'expert, le système utilise la connaissance, l'examine et permet d'en apprécier les conséquences.

KAS /REBO-79/, Knowledge Acquisition System, est le noyau de PROSPECTOR. IL permet de créer, de modifier ou détruire des connaissances, il protège l'utilisateur des erreurs syntaxiques et conserve une trace des changements à compléter.

Par une exécution contrôlée du programme, l'utilisateur a accès aux effets des modifications.

KAS représente ainsi un puissant outil de développement d'une base de connaissances.

Lors de la conception d'un système expert on peut avoir recours à un noyau existant et alors implanter la base de connaissances du domaine à étudier, mais on peut vouloir réaliser un système expert complet.

Ainsi des langages ont été créés afin d'en faciliter la construction.

AGE /NII-78b/ met à la disposition du concepteur des modules indépendants et préprogrammés pour implanter la base de connaissances, l'interpréteur et la base de faits, (par exemple pour l'interpréteur, il fournit des procédures sur le chaînage avant ou le chaînage arrière).

De même d'autres systèmes, comme ROSIE /WATE-79/, HEARSAY III /BALZ-80/, sont prévus pour faciliter la mise en oeuvre de systèmes experts dans des domaines et des fonctions variés.

V CONCLUSION

Les systèmes experts, grâce à leurs caractéristiques, permettent de s'adapter d'une manière efficace à des fonctions très diverses.

Cette efficacité réside essentiellement dans la qualité des connaissances qu'ils possèdent, ce qui nécessite, pour le développement des systèmes experts, une collaboration active entre l'expert et le concepteur.

Le transfert des connaissances de l'expert vers une " machine " doit être accepté par le spécialiste sans craindre d'y perdre un certain pouvoir.

Il lui est plus facile d'accepter une machine traitant des tâches dites pénibles et peu motivantes qu'un système capable de résoudre des problèmes dont il détient la " *Compétence Intellectuelle* ".

L'objectif du concepteur consiste à présenter les systèmes experts non pas comme des rivaux mais comme étant des simples outils d'aide à leur fonction, au même titre que les autres applications informatiques afin que les experts ne se sentent plus " menacés " par ces logiciels.

De plus il faut qu'il sache que la responsabilité de la décision finale sera toujours de leur compétence et qu'ils gardent l'entière maîtrise de la base de connaissances.

En effet, un des avantages des systèmes experts est la séparation des connaissances et du mécanisme de raisonnement et donc la possibilité d'écarter momentanément, lors de la conception, certains points où l'expert désire avoir l'avis de confrères, contrairement à la programmation classique nécessitant une analyse exhaustive du problème.

Cependant l'utilisation de la programmation traditionnelle ne doit pas être écartée car il existe, pour certains problèmes, des solutions algorithmiques répondant parfaitement et de manière satisfaisante aux problèmes posés.

Toutefois l'utilisation de systèmes experts paraît bien adaptée aux domaines nécessitant une grande quantité de

connaissances ainsi qu'une compétence quant à l'utilisation de celles-ci en fonction des cas particuliers à traiter.

CHAPITRE V

SYSTEME EXPERT EN LIGNE

CONCEPTION D'UNE STATION DE SURVEILLANCE
ET DE PRISE DE DECISION EN TEMPS REEL

CHAPITRE V :
SYSTEME EXPERT EN LIGNE
CONCEPTION D'UNE STATION DE SURVEILLANCE
ET DE PRISE DE DECISION TEMPS REEL

<u>I POSITION DU PROBLEME</u>p.	1
<u>I.1 UTILISATION D'UNE TECHNOLOGIE PLUS RAPIDE</u>p.	1
<u>I.2 STRUCTURATION DE LA BASE DE CONNAISSANCE</u>p.	1
<u>I.3 AMELIORATION DES ALGORITHMES DE RESOLUTION</u>p.	2
<u>I.4 INTRODUCTION DE PARALLELISME</u>p.	2
<u>II ETUDE DE L'EFFICACITE D'UN SYSTEME EXPERT</u>p.	2
<u>II.1 PRESENTATION DE LA STRUCTURE INTERNE DES CONNAISSANCES</u>p.	3
<u>II.2 TEMPS DE REPONSE PAR RAPPORT A L'ORDRE DES REGLES</u> ..p.	3
<u>II.3 TEMPS DE REPONSE EN FONCTION DU NOMBRE DE REGLES</u> ...p.	7
<u>III AMELIORATIONS APORTEES AU MOTEUR D'INFERENCE</u>p.	10
<u>III.1 AMELIORATION DU CHAINAGE AVANT</u>p.	10
<u>III.1.1 SUPPRESSION DES FAITS VERIFIES</u>p.	10
<u>III.1.2 SUPPRESSION DES REGLES APPLIQUEES</u>p.	11
<u>III.1.3 SUPPRESSION DES FAITS ET DES REGLES APPLIQUES</u>p.	11
<u>III.1.4 REMARQUE</u>p.	12
<u>III.2 AMELIORATION DU CHAINAGE ARRIERE</u>p.	13
<u>III.2.1 MEMORISATION DES FAITS CONFIRMES</u>p.	13
<u>III.3 AMELIORATION DUE A L'UTILISATION D'UNE TECHNOLOGIE PLUS RAPIDE</u>p.	14
<u>III.4 CONCLUSION</u>p.	15

<u>IV PERSPECTIVES</u>	p. 16
<u>IV.1 UTILISATION D'UNE MEILLEURE STRUCTURE INTERNE</u>	p. 16
<u>IV.2 INTRODUCTION DE PARALLELISME</u>	p. 20
<u>IV.3 NOTION DE TEMPS</u>	p. 20
<u>V CONCLUSION : SYSTEME EXPERT EN LIGNE</u>	p. 23

I POSITION DU PROBLEME

Lorsque la littérature parle de l'efficacité des systèmes experts, elle évoque généralement ses aptitudes à émettre un diagnostic correct ou à formuler son raisonnement.

Ainsi NILSON, /NILS-71/, qualifie un système expert de complet lorsqu'il est capable de trouver une solution si elle existe et d'admissible si il trouve la "meilleure" solution.

Cependant la rapidité d'exécution des systèmes experts représente une qualité primordiale lorsque ceux ci doivent être utilisés dans des applications temps réels (par exemple pour le contrôle de processus /KNIC-85/).

Différentes méthodes peuvent être abordées pour accroître cette rapidité d'exécution des systèmes de production :

- (1) L'utilisation d'une technologie plus rapide.
- (2) La structuration de la base de connaissance.
- (3) La conception de meilleurs algorithmes de résolution.
- (4) La mise en place d'une structure parallèle.

I.1 L'UTILISATION D'UNE TECHNOLOGIE PLUS RAPIDE

L'utilisation d'une technologie plus rapide fait appel non seulement au choix du matériel utilisé mais aussi à la conception d'architecture de machines spécifiques (projet japonais de "5ème génération" d'ordinateurs).

Cette phase vise à réduire le temps d'exécution du cycle de base d'un moteur d'inférence.

I.2 STRUCTURATION DE LA BASE DE CONNAISSANCE

Le formalisme sous forme de règles de production est la représentation externe de la connaissance facilitant son intégration. Elle représente, par analogie avec la programmation classique, le fichier source du programme.

Afin que le moteur d'inférence utilise une représentation interne efficace, cette base de connaissance est souvent compilée.

La compilation de la base de connaissance consiste à établir des liens directs entre les règles et les termes (prémises et conclusions) afin de structurer la connaissance sous forme de liste et donc d'éviter un accès à

la base de règles selon un parcours totalement séquentiel /VIGN-85/.

I.3 AMELIORATION DES ALGORITHMES DE RESOLUTION

Dans le cycle de base du moteur d'inférence, l'étape de filtrage, ayant pour rôle de sélectionner l'ensemble des règles applicables en fonction de l'état courant de la base de faits, représente la phase la plus coûteuse en temps (environ 90 % pour le système OPS5 /FORG-81/).

Cette étape de filtrage ou de sélection des règles candidates peut être optimisée :

--> soit par une étape de restriction en distinguant des classes de données ou de règles.

Dans le système ARGOS, /FARR-80/, la base de faits est partitionnée en une base de faits à établir et une base de faits établis. Ainsi le premier filtre des déclencheurs de règles ne porte que sur la base de faits à établir.

De plus un indicateur est associé à chaque règle précisant si la règle est active ou inhibée.

--> soit par une amélioration des algorithmes de résolution.

I.4 INTRODUCTION DE PARALLELISME

/GUPT-85/ décrit les différentes sortes de parallélisme pouvant être utilisées afin d'améliorer la vitesse d'exécution.

Les parallélismes peuvent être introduits dans les trois grandes étapes du cycle de base :

- > Le filtrage
- > La résolution de conflit
- > L'activation.

II ETUDE DE L'EFFICACITE D'UN SYSTEME EXPERT

Le système expert étudié a été présenté dans /FERB-84/. C'est un moteur d'inférence d'ordre zéro écrit en pascal fonctionnant aussi bien en chaînage avant qu'en chaînage arrière.

Les règles n'autorisent donc pas l'utilisation de variables et sont écrites dans le formalisme :

SI Fait1 Et Fait2 Et ... Et Faitm
Alors Faitp Et ... Et Faitr

II.1 PRESENTATION DE LA STRUCTURE INTERNE DES CONNAISSANCES

En début d'exécution, le programme accède au fichier BASE contenant la base de faits et la base de règles, afin de compiler cette base de connaissance et donc de construire la structure interne (cf figure V.1) sur laquelle travaille le moteur d'inférence.

L'insertion d'un nouvel élément dans la liste se faisant en tête de liste, les règles et les faits écrits en premier dans le fichier base seront placés en queue de liste.

L'utilisation de liste et donc de pointeurs, permet d'optimiser la place mais surtout d'accélérer le mécanisme d'unification des règles sur la base en ne comparant que des pointeurs et non des chaînes de caractères.

Ceci permet d'utiliser des faits de longueur quelconque (limitée malgré tout à 80 caractères) sans pénaliser l'étape de filtrage.

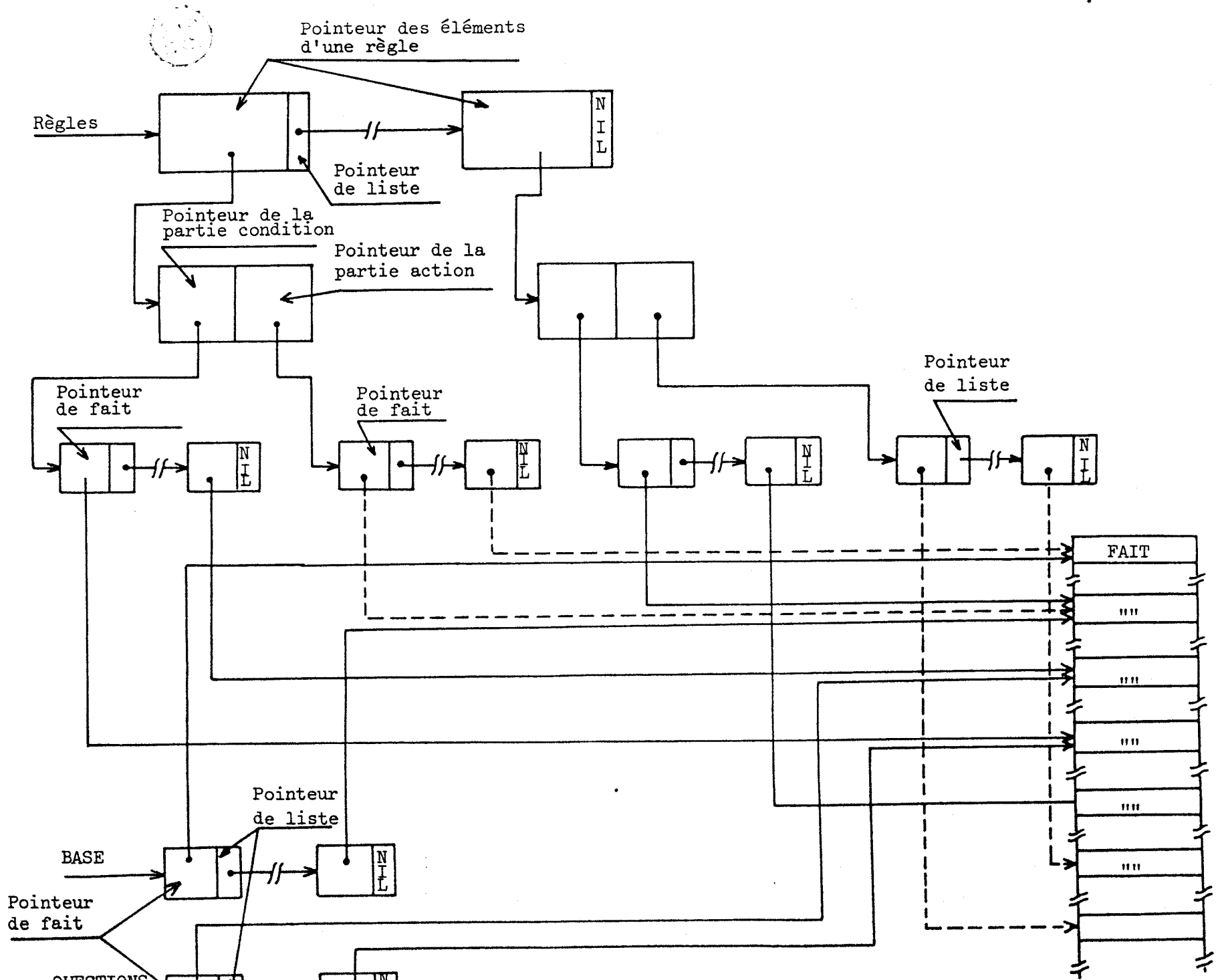
Lorsque l'opérateur utilise le système en chaînage arrière, c'est à dire qu'il cherche à confirmer un fait, le moteur d'inférence peut lui demander des renseignements sur la véracité d'un fait.

Si l'opérateur répond que ce fait est vrai, alors il est placé dans la liste BASE (représentant la liste des faits vrais) sinon il est placé dans la liste des faits QUESTION. Ceci permet au système de savoir qu'un fait est considéré comme faux et non comme inconnu.

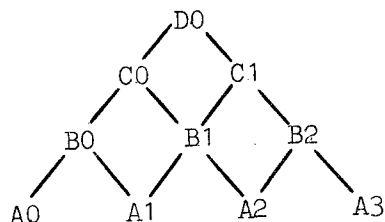
II.2 TEMPS DE REPONSE PAR RAPPORT A L'ORDRE DES REGLES

Le choix de la règle à activer se fait suivant le critère de sélection : "La règle à activer est la première règle applicable rencontrée".

Si ce critère de sélection n'influe pas sur le résultat final, par contre il modifie le temps de réponse en fonction de l'ordre des règles.



En effet considérons la base de règles suivantes :

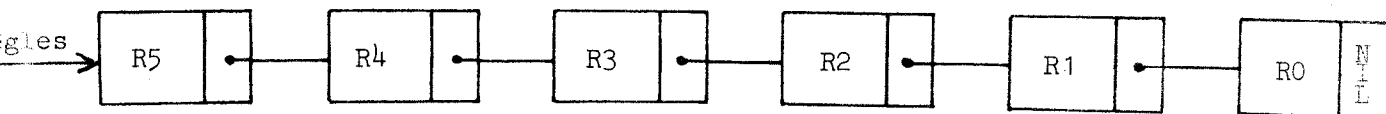


- R0 : si C0 et C1 alors D0
- R1 : si B1 et B2 alors C1
- R2 : si B0 et B1 alors C0
- R3 : si A3 et A2 alors B2
- R4 : si A2 et A1 alors B1
- R5 : si A1 et A0 alors B0

et la base de faits initiale suivante (A0, A1, A2, A3).

Le fait D0 peut être vérifié par l'activation de toutes les règles.

Supposons que ces règles soient chaînées dans la configuration suivante :



Le système va tester en premier lieu si R5 s'applique, c'est à dire il teste si A0 et A1 sont vérifiés, comme ils appartiennent à la base de faits, R5 est activée et donc B0 est placé dans la base de fait.

Le système se réinitialisant en début de chaîne, le pointeur courant pointe de nouveau R5. Le moteur d'inférence teste donc une nouvelle fois l'applicabilité de R5, R5 est donc activée une nouvelle fois mais comme B0 appartient déjà à la base de faits, il continue sur le test de R4...etc...

Donc lorsqu'une règle est applicable, le moteur d'inférence l'active et compare, en dernier lieu, si la partie condition appartient ou non, dans son intégrité, à la base de faits.

Si on s'intéresse au nombre de fois que chaque règle est activée avant de trouver le fait D0, on aboutit au tableau 1.

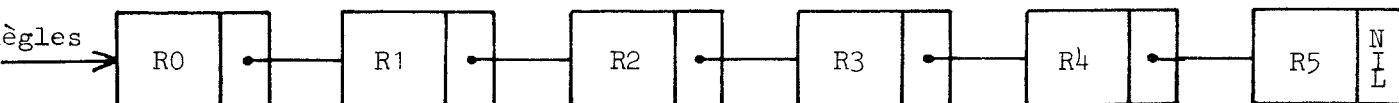


REGLES	R0	R1	R2	R3	R4	R5
Nombre d'activations	1	2	3	4	5	6

Tableau 1

Soit un total de 21 activations.

Si les règles sont chaînées en sens inverse :



Le moteur d'inférence teste en premier R0, comme C0 et C1 sont faux, il passe à la règle suivante, même chose pour R1 et R2, R3 s'applique, il place ainsi B2 dans la base de faits puis se réinitialise sur R0...etc...

La construction du nombre d'activation par règle est donnée par le tableau 2 :



REGLES	R0	R1	R2	R3	R4	R5
Nombre d'activations	1	3	1	3	2	1

Tableau 2

Soit un total de 11 activations.

On s'aperçoit, sur cet exemple simple, que le moteur d'inférence a activé 11 règles dans le deuxième cas plutôt que 21 dans le premier.

Il y a donc un gain de temps appréciable qui se retrouve dans le test effectué en utilisant une "arborescence" de connaissance similaire à l'exemple précédent mais portant sur un ensemble de 30 règles (correspondant à BASE1 de la figure V.2).

Dans le premier cas le système met 15 secondes pour conclure sur le fait de niveau le plus haut (J0) et 10 secondes dans le deuxième cas.

En chaînage arrière ce système est plus lent dans le deuxième cas.

En effet si, dans le même contexte, il doit confirmer le fait de niveau le plus haut, soit J0, il met 35 secondes à résoudre le problème dans le premier cas et 43 secondes dans le deuxième cas.

Ceci peut s'expliquer par le fait qu'en chaînage arrière le raisonnement est guidé par les buts tandis qu'en chaînage avant il est guidé par les faits.

Ainsi, bien que l'auteur précise que la connaissance peut être intégrée en "vrac", il est nécessaire de préciser que cette notion de vrac est réelle quant à la solution finale, mais que l'ordre intervient dans le temps de réponse du système.

II.3 TEMPS DE REPONSE EN FONCTION DU NOMBRE DE REGLES

Un facteur très important dans le temps de réponse du système est évidemment la consistance de la base de connaissance ou le nombre de règles que le système doit traiter.

Afin d'étudier les performances de ce moteur d'inférence, nous avons testé celui ci avec 4 bases de connaissance différentes, aussi bien en chaînage avant qu'en chaînage arrière, en positionnant les faits du niveau le plus bas dans la base de faits et en mesurant le temps nécessaire à déduire (chaînage avant) ou à confirmer (chaînage arrière) le fait de niveau le plus haut.

Le test a porté sur 4 bases de connaissances possédant respectivement :

- > 30 règles (BASE1)
- > 51 règles (BASE2)
- > 72 règles (BASE3)
- > 93 règles (BASE4) (cf. figure V.2).

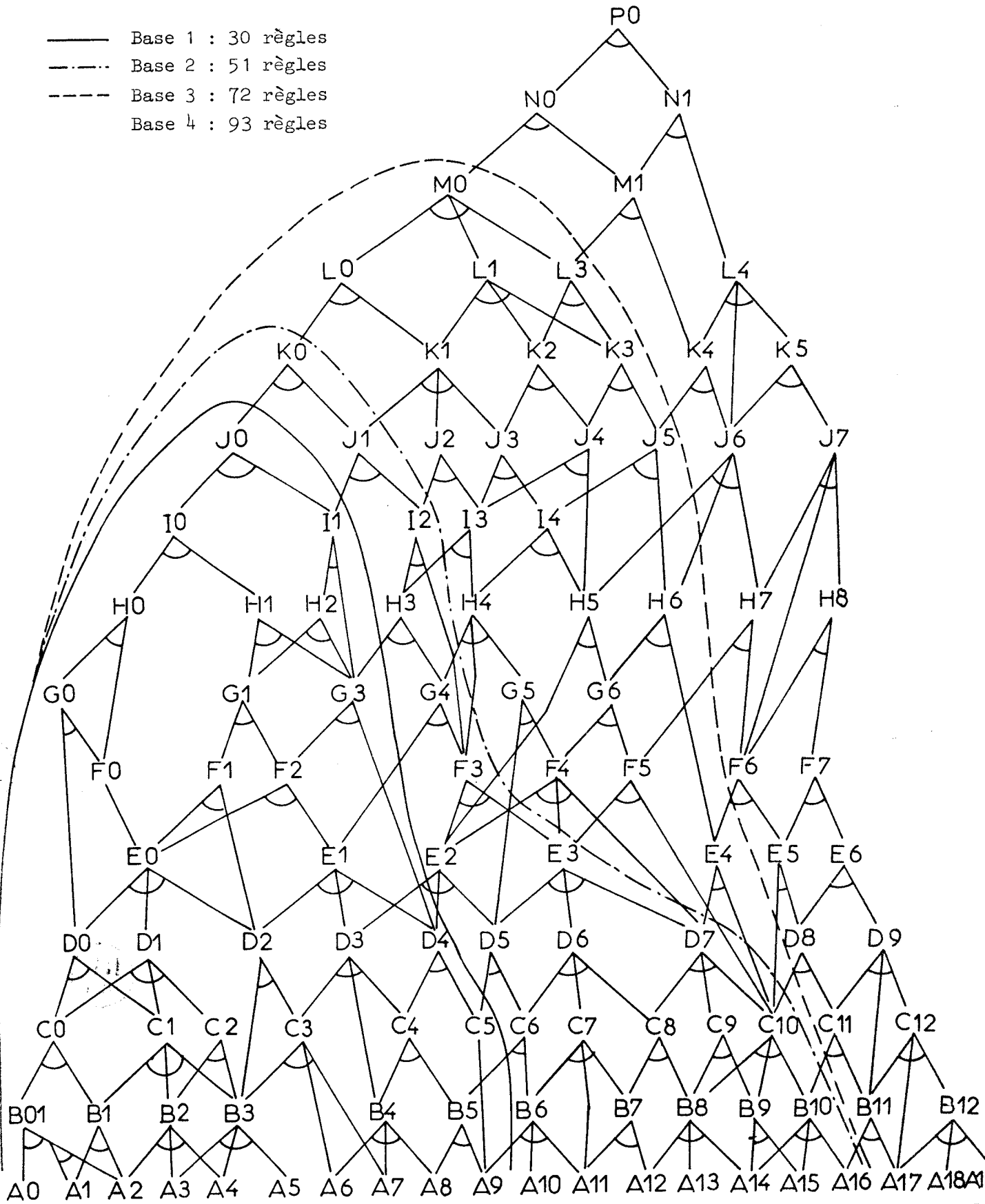
Le tableau 3 indique les temps de réponse en chaînage avant et chaînage arrière et les figures V.3 et V.4 fournissent les courbes de temps de réponse en fonction du nombre de règles.

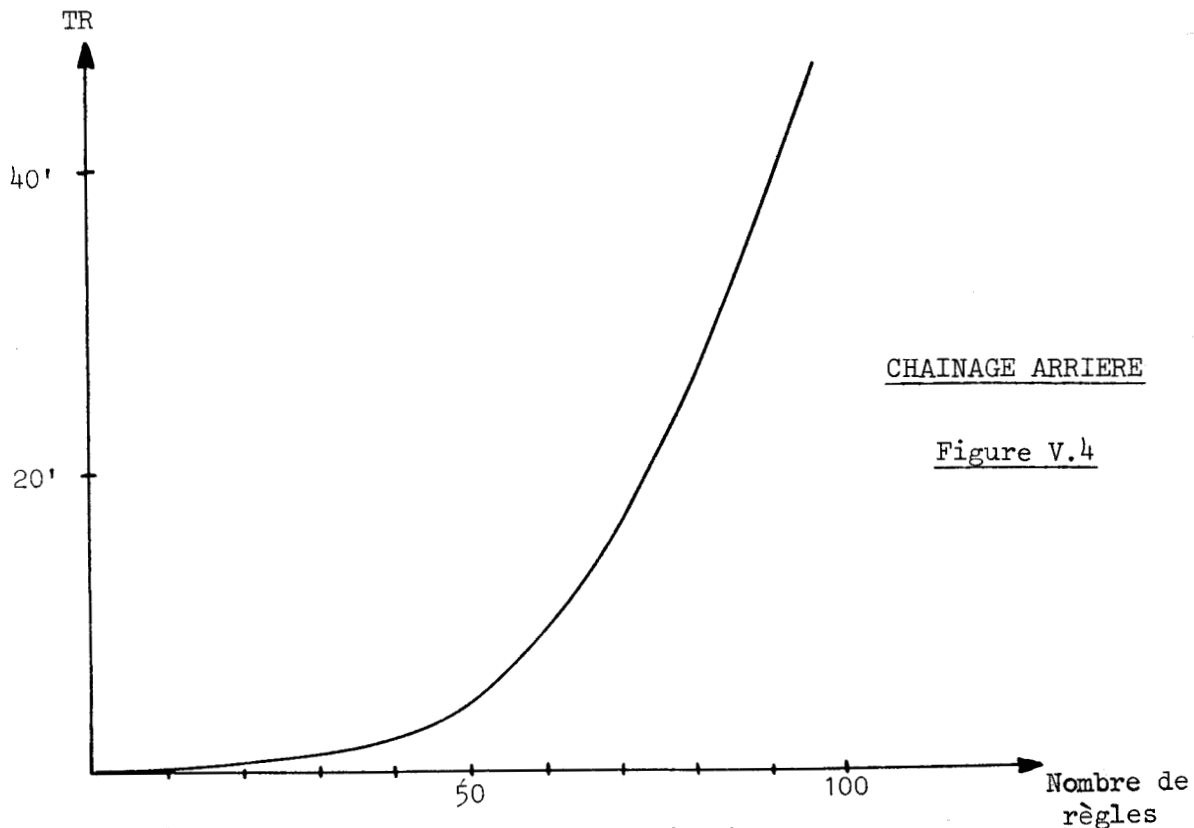
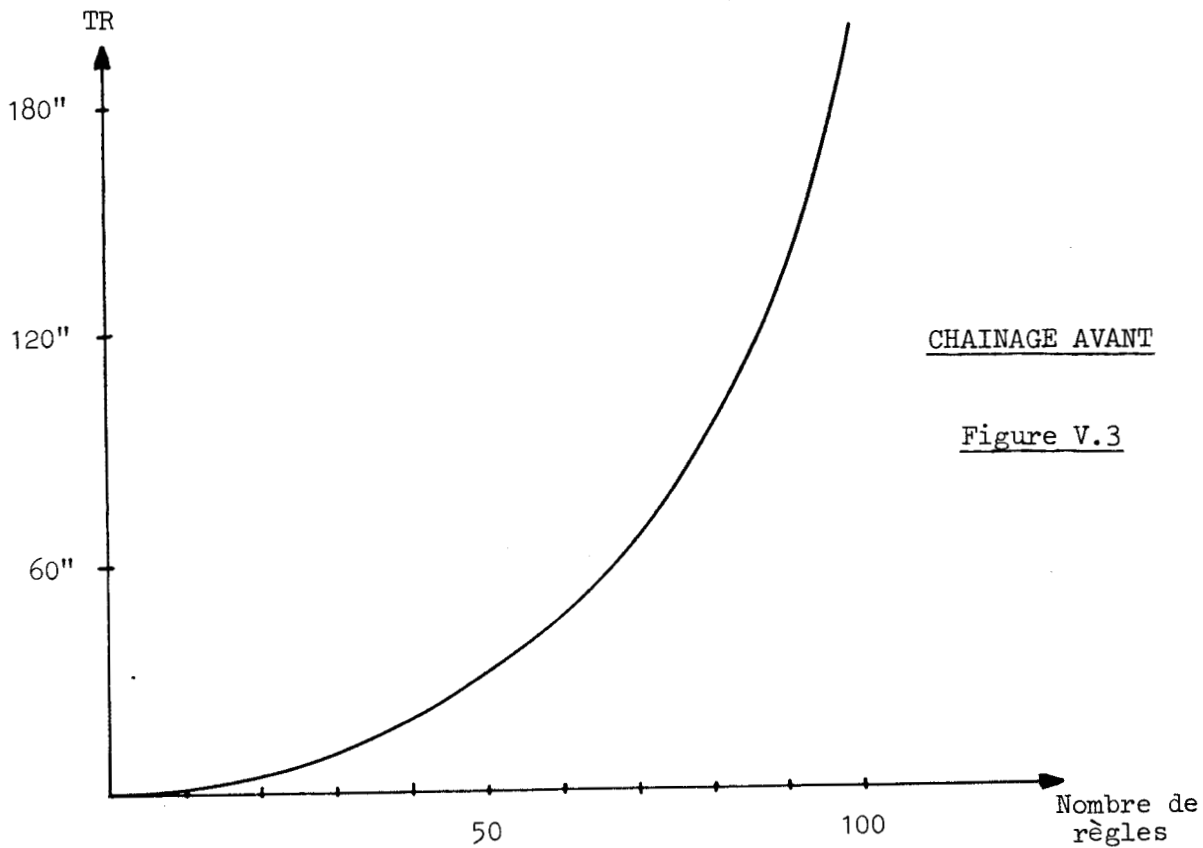
BASE	BASE 1	BASE 2	BASE 3	BASE 4
Chaînage avant	10"	45"	1' 23"	3' 05"
Chaînage arrière	43"	1' 37"	18' 30"	41'



Tableau 3

Figure V.2





L'analyse de ces deux courbes indique que le temps de réponse du système est très sensible au nombre de règles et ceci beaucoup plus en chaînage arrière qu'en chaînage avant.

De plus un temps de 41 minutes pour une base de connaissance de 93 règles est un délai que peu d'utilisateurs auront la patience d'attendre...

C'est pourquoi différentes améliorations ont été apportées à ce moteur d'inférence afin d'en augmenter les performances aussi bien en chaînage avant qu'en chaînage arrière.

III AMELIORATIONS APORTEES AU MOTEUR D'INFERENCE

III.1 AMELIORATION DU CHAINAGE AVANT

III.1.1 SUPPRESSION DES FAITS APPLIQUES

En chaînage avant, lorsque le moteur d'inférence parcourt la base de règles, il teste les faits appartenant à la partie condition des règles.

Or une règle de la forme :

Si A et B alors C (1)

ne peut être activée que si A et B sont vrais (s'ils appartiennent à la liste BASE).

Donc si A est vrai et si B est inconnu, la règle n'est pas activable. Pourtant à un premier passage le moteur d'inférence a pu prouver que A est vrai.

Or il est possible que l'activation d'une règle déduise le fait B. De ce fait (1) devient activable, mais le moteur doit tester à nouveau la véracité de A qu'il avait prouvée précédemment.

C'est pourquoi, à chaque règle possédant un fait appartenant à la partie condition et existant dans la base de faits, il est possible de supprimer ce fait sachant qu'il est vrai et qu'il est donc inutile de le retester à chaque cycle.

Cette suppression des faits appliqués amène le gain de temps, par rapport au système initial, décrit dans le tableau 4.

BASE	BASE 1	BASE 2	BASE 3	BASE 4
Système initial	10"	45"	1' 23"	3' 05"
Système amélioré	8"	28"	1' 03"	1' 59"

Tableau 4

Cette suppression des faits intervient sur l'étape de filtrage des règles.

III.1.2 SUPPRESSION DES REGLES APPLIQUEES

Dans la même optique que précédemment, une règle ayant apporté sa connaissance par son activation n'est plus utile à la résolution du problème : elle ne peut qu'apporter une connaissance que le système possède.

Or le moteur d'inférence dans le parcours de la base de règles teste l'applicabilité de toutes les règles, même si elles ont déjà été activées.

Si elle est applicable, il active cette règle et il teste la base de faits afin de savoir si la partie conclusion appartient déjà à cette base de faits.

L'intégration d'un algorithme de suppression des règles appliquées apporte le gain de temps donné par le tableau 5.

BASE	BASE 1	BASE 2	BASE 3	BASE 4
Système initial	10"	45"	1' 23"	3' 05"
Système amélioré	8"	22"	54"	1' 35"

Tableau 5



III.1.3 SUPPRESSION DES FAITS ET DES REGLES APPLIQUES

La suppression des faits et des règles appliqués apporte le gain de temps décrit dans le tableau 6.

BASE	BASE 1	BASE 2	BASE 3	BASE 4
Système initial	10"	45"	1' 23"	3' 05"
Système amélioré	7"	21"	53"	1' 34"

Tableau 6

La conjonction des deux améliorations apporte un léger gain de temps par rapport à la suppression des règles.

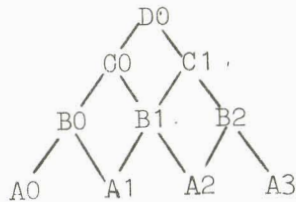
III.1.4 REMARQUE

Dans le moteur d'inférence initial, lorsque la base de connaissance est chaînée tel que les règles de niveau bas sont en tête de liste, le système est plus lent que si elles sont placées en queue de liste.

Dans le système avec la suppression des règles et des faits la conclusion inverse est vérifiée :

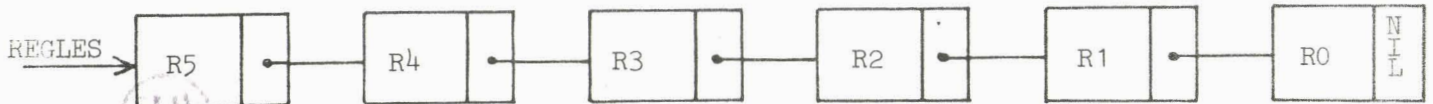
En considérant la base de connaissance de 30 règles et si les règles de niveau haut sont en tête le système met 7 secondes à déduire J0 tandis que si elles se retrouvent en queue de liste il ne met que 3 secondes.

Considérons l'exemple précédent :



- R0 : si C0 et C1 alors D0
- R1 : si B1 et B2 alors C1
- R2 : si B0 et B1 alors C0
- R3 : si A3 et A2 alors B2
- R4 : si A2 et A1 alors B1
- R5 : si A1 et A0 alors B0

Supposons que ces règles soient chaînées dans la configuration suivante :



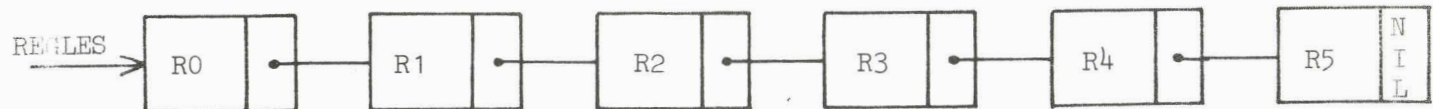
R5 est applicable, elle est donc activée puis détruite.

Le pointeur courant se réinitialise en tête de liste, il pointe donc R4 qui est activable, donc activée puis détruite ...etc...

Dans ce cas, chaque règle est donc activée une seule fois.



Par contre pour la configuration suivante :



R0, R1, R2 sont testées afin de savoir si elles sont activables, comme elles ne le sont pas, le système poursuit

sa recherche, R3 est activée et détruite, le pointeur se réinitialise en tête de liste et reteste R0, R1, R2...etc...

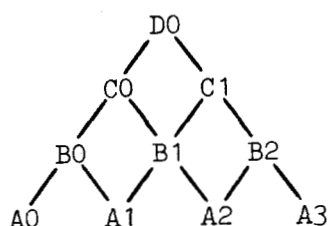
Le moteur d'inférence est donc plus rapide dans la première configuration.

III.2 AMELIORATION DU CHAINAGE ARRIERE

III.2.1 MEMORISATION DES FAITS CONFIRMES

Le système initial ne positionne pas dans la base de faits les faits qu'il confirme, en chaînage arrière.

Ainsi pour la base de connaissance suivante :



- R0 : si C0 et C1 alors D0
- R1 : si B1 et B2 alors C1
- R2 : si B0 et B1 alors C0
- R3 : si A3 et A2 alors B2
- R4 : si A2 et A1 alors B1
- R5 : si A1 et A0 alors B0

Lorsqu'il doit confirmer D0, il faut qu'il confirme C0 et C1, puis B0, B1, B2.

Comme B1 appartient à la partie condition de R1 et R2, le moteur va justifier deux fois B1 par l'application de la règle R1.

Ce doublon peut être évité en plaçant dans la base de faits, les faits que le moteur confirme.

Cette amélioration amène le gain de temps indiqué dans le tableau 7.

BASE	BASE 1	BASE 2	BASE 3	BASE 4
Système initial	43 "	1' 37"	18' 30"	41'
Système amélioré	3"	8"	12"	18"



Tableau 7

III.3 AMELIORATION DUE A L'UTILISATION D'UNE TECHNOLOGIE PLUS RAPIDE

Les temps de réponse, fournis précédemment, sont les résultats de tests opérés sur un MICRAL-30 (compatible IBM-PC) avec un programme écrit sous MS-PASCAL.

L'utilisation de TURBO-PASCAL, sur un même matériel, a permis de diminuer considérablement les temps de réponse.

Le tableau 8 donne les résultats pour le chaînage avant et le tableau 9 pour le chaînage arrière (le système initial a été testé seulement sous MS-PASCAL).

BASE	BASE 1	BASE 2	BASE 3	BASE 4
Système initial	10 "	45 "	1' 23"	3' 05"
Système amélioré sous MS-PASCAL	7 "	21 "	53 "	1' 34"
Système amélioré sous TURBO-PASCAL	3 "	9 "	19 "	38 "

Tableau 8

BASE	BASE 1	BASE 2	BASE 3	BASE 4
Système initial	43 "	1' 37"	18' 30"	41 "
Système amélioré sous MS-PASCAL	3 "	8 "	12 "	18 "
Système amélioré sous TURBO-PASCAL	2 "	4 "	6 "	8 "

Tableau 9

Compte tenu que la structure du moteur d'inférence est essentiellement basée sur la comparaison de pointeurs et que le MICRAL-30 est doté d'un microprocesseur INTEL 8088 ("faux" 16 bits), l'utilisation d'un microordinateur, muni d'un véritable 16 bits, diminuerait considérablement les temps d'exécution.

Dans ce sens, le système a été testé sur un IBM PC/AT muni d'un microprocesseur INTEL 80286.

Les gains de temps apportés par l'utilisation de cette technologie plus rapide donne les résultats indiqués dans le tableau 10.

Seul les résultats des tests opérés sur la BASE4 (93 règles) sont fournis.

	Système sous MS-PASCAL		Système sous TURBO-PASCAL	
	MICRAL-30	IBM PC/AT	MICRAL-30	IBM PC/AT
Chaînage avant	1' 34 "	35 "	38 "	15 "
Chaînage arrière	18 "	8 "	8 "	6 "

Tableau 10



III.4 CONCLUSION

Les améliorations apportées au moteur d'inférence ont permis de diminuer considérablement les temps d'exécution et donc d'en accroître l'efficacité.

Ces améliorations portent sur deux points essentiels :

- > L'utilisation de meilleurs algorithmes :
 - (1) suppression des faits et des règles appliqués
 - (2) mémorisation des faits confirmés.

- > L'utilisation d'une technologie plus rapide :
 - (1) l'emploi de TURBO-PASCAL
 - (2) l'utilisation d'un véritable 16 bits tel le 80286 d'INTEL.

Ces améliorations permettent de passer, pour la BASE4 (93 règles), d'une résolution demandant 3 minutes 5 secondes à un temps de réponse de 15 secondes en chaînage avant, et de 41 minutes à 6 secondes en chaînage arrière...

La suite de ce chapitre décrit des méthodes susceptibles d'améliorer le système, par l'emploi d'une meilleure structure interne des connaissances afin de pouvoir connecter le système à des processus temps réel.

IV PERSPECTIVES

IV.1 UTILISATION D'UNE MEILLEURE STRUCTURE INTERNE

Dans le moteur d'inférence précédent, pour décider si une règle est applicable, le système vérifie que la partie condition est vraie.

Pour cela il vérifie que chaque fait de la condition appartient à la base de faits, ce qui oblige à parcourir la liste BASE tant qu'il n'a pas trouvé ces faits.

Si un de ces faits n'est pas vrai, il balaie cette liste dans son ensemble pour conclure que le fait n'appartient pas à la base de faits donc que la règle n'est pas applicable.

En chaînage arrière, la confirmation d'un fait oblige le parcourt de la liste QUESTION afin de connaître si un fait n'a pas été déclaré faux précédemment.

Lorsqu'une règle est activée, le système cherche à placer la partie condition dans la base de faits.

Or si un ou plusieurs des termes de la conclusion appartient déjà à la base de faits, il est évident qu'il ne faut pas le réécrire dans cette liste.

Ceci oblige le système à tester l'ensemble de la base de faits et le cas échéant de créer un élément dans cette liste.

Ainsi l'activation d'une règle implique des parcours multiples de la base de faits causant des pertes de temps.

Il est possible de créer une structure interne des connaissances, par une compilation différente, où une liste contiendrait l'ensemble des faits déclarés aussi bien dans la base de faits que dans la base de règles (cf. figure V.5).

Un indicateur serait associé à chaque fait précisant si il est :

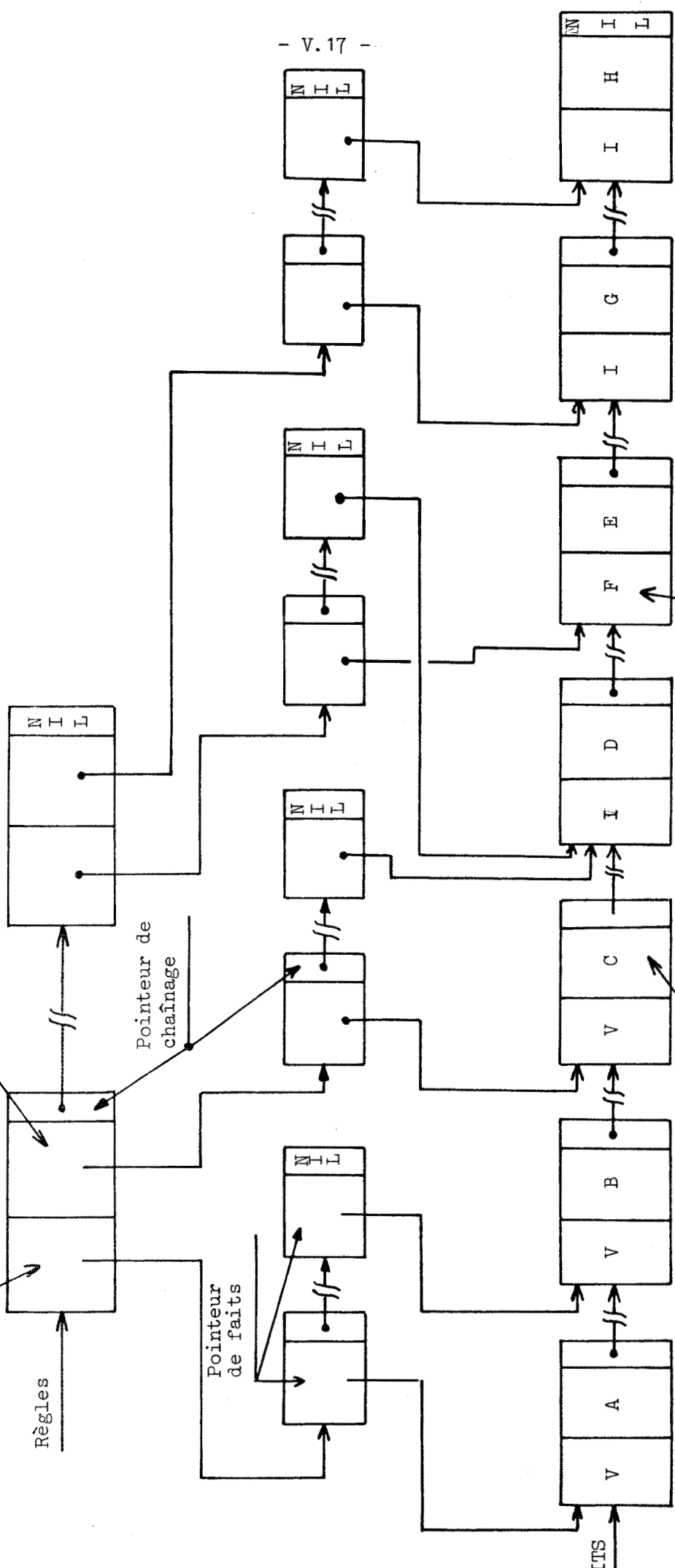
- > vrai
- > faux
- > inconnu.

La liste REGLE, dans ce cas, est composée de cellules chaînées entre elles, possédant chacune trois éléments :

- > Un pointeur de la partie condition.

Celui ci pointe la liste des faits condition de la règle.

partie condition
partie action



Valeur littérale du fait (chaîne de caractères)

Indicateur de vérité :
- vrai
- faux
- inconnu

Exemple :

- R1: Si A et .. et B Alors C et .. et D
- R2: Si E et .. et D Alors G et .. et H

R2 ne peut s'appliquer car E est faux

Figure V.5



--> Un pointeur de la partie action.
Il pointe la liste des faits conclusion.

--> Un pointeur pointant la règle suivante.

De même la liste FAITS possède la même structure où chaque cellule est aussi composée de trois éléments :

--> Un indicateur précisant si le fait est vrai, faux ou inconnu.

--> Une chaîne de caractères représentant le fait littéral.

--> Un pointeur de chaîne pointant le fait suivant.

Cette structure évite des recherches exhaustives à divers point de vue :

--> Les faits de la partie condition ou action d'une règle pointe directement leur valeur logique.

--> Ils pointent directement leur valeur littéral.

--> Une seule liste indique si le fait est vrai, faux ou inconnu, évitant les parcours multiples des listes BASE et QUESTION du moteur précédent.

De plus cette structure facilite la déclaration des faits faux dans la base de faits et leur pleine utilisation.

En effet, un fait inconnu est potentiellement vrai, donc une règle possédant ce fait dans sa partie condition est susceptible d'être appliquée.

Par contre si elle possède un fait faux, cette règle est non activable, la "connaissance" renfermée dans cette règle est donc inexploitable pour le cas présent et peut donc être détruite.

Dans ce système, une règle est donc détruite dans deux cas :

--> La règle a été appliquée

--> La règle est non applicable.

L'intérêt majeur de cette structure est que chaque fait est lié directement à sa véracité sans que l'on ait à parcourir la base de faits, et à comparer ni des pointeurs ni des chaînes de caractères.

Ceci, avec les procédures de destruction des règles, devrait minimiser les temps d'exécution.

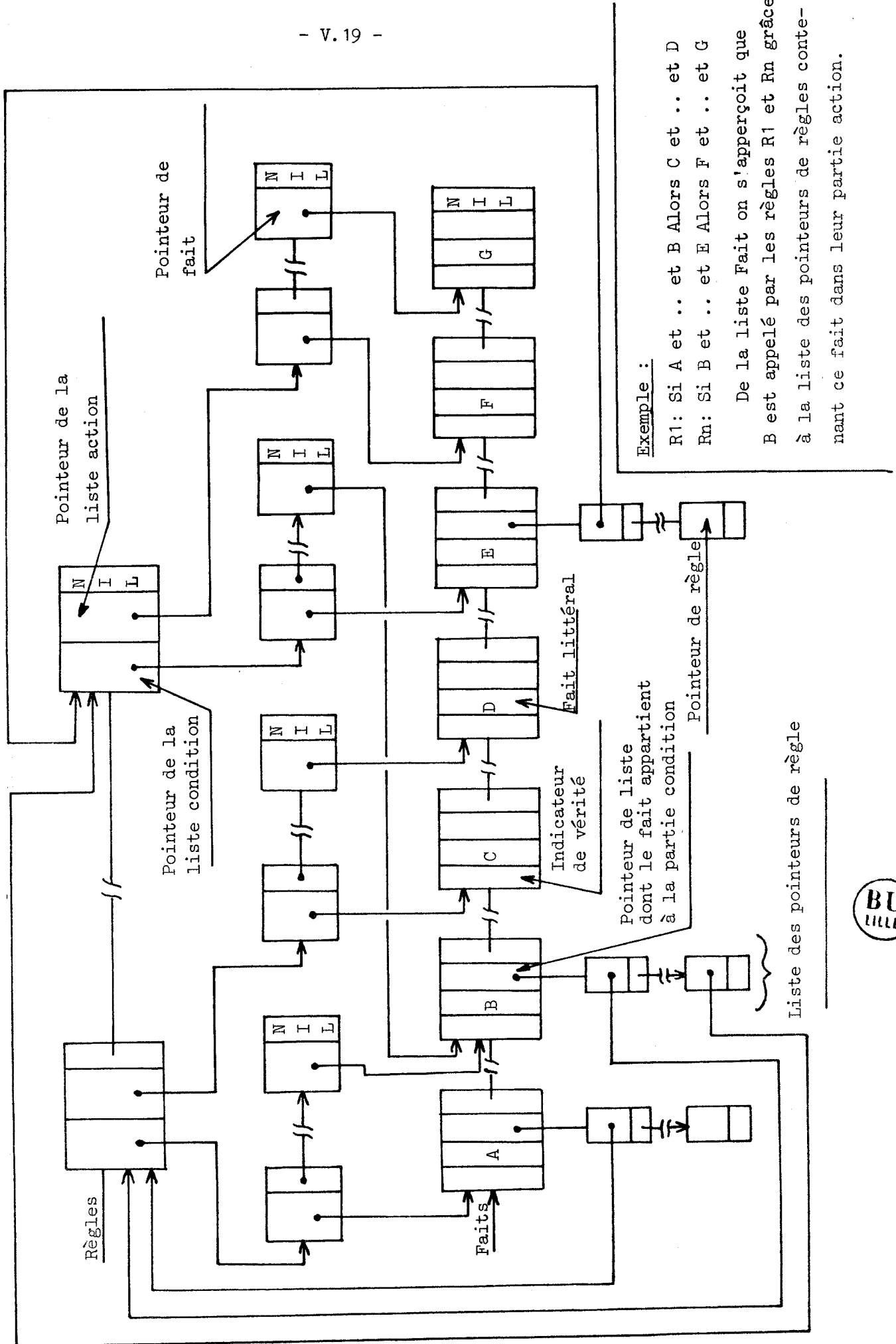


Figure V.7



De plus l'accès à la valeur littérale des faits dans la liste FAITS, permet à un processus externe de pouvoir écrire aisément la valeur logique de ceux ci.

Ce qui permet de connecter à ce système expert, un générateur de faits alimentant "en ligne" sa base de faits.

IV.2 INTRODUCTION DE PARALLELISME

La structure interne des connaissances, précédemment décrite, peut être complétée comme le montre la figure V.7.

Dans cette structure les cellules de la liste FAIT sont composées de quatre éléments :

- > Un indicateur précisant si le fait est vrai, faux ou inconnu.
- > Une chaîne de caractères représentant le fait littéral.
- > Un pointeur de liste contenant l'ensemble des règles dont le fait appartient à la partie condition.
- > Un pointeur de chaîne pointant le fait suivant.

Ainsi, par le parcourt de la liste FAIT, le système peut accéder à l'ensemble des règles appelant un fait dans sa partie condition, ce qui permet :

- > de supprimer les faits vrais dans les parties condition et action des règles, car un fait n'a plus à être prouvé s'il l'est déjà.
- > de supprimer les règles possédant un fait faux dans leur partie condition.

De plus ces procédures peuvent être confiées à une tâche fonctionnant en parallèle avec le moteur d'inférence.

Cette tâche, en diminuant l'étape de filtrage du système sans le perturber dans son exécution, diminuerait les temps de réponse de celui ci.

IV.3 NOTION DE TEMPS

Dans le contrôle de processus les données et les mesures opérées sur ce processus sont dynamiques : l'état d'un processus peut évoluer durant l'analyse.

Cette contrainte oblige les systèmes experts de contrôle/conduite de processus à gérer l'aspect temporel et évolutif de l'information.

Or le formalisme des connaissances sous forme de règles de production et de faits implique une représentation factuelle de l'information où l'intégration de la notion de temps se révèle très difficile.

C'est pourquoi dans les systèmes experts traditionnels, faits et connaissances sont statiques : il n'est pas envisagé que l'état d'un processus puisse changer en cours d'analyse :

lorsque la validité d'une donnée est établie elle est immuable.

Pour pallier cet inconvénient, tout en gardant la notion factuelle de l'information, il est possible de structurer l'univers des connaissances en hiérarchisant le savoir tant au niveau de l'expertise du domaine que du cas traité (cf. figure V.8).

Cette structure permet d'intégrer, implicitement, la notion de temps dans les faits, en considérant la notion d'évolution de l'information.

En effet le couplage d'un système expert à un système de surveillance de processus alimentant en ligne la base de faits en données caractérisant l'état de ce processus à un instant t , permet au moteur d'inférence de conclure sur l'état de ce processus à cet instant, en fonction des connaissances contenues dans une base de règles dite "statique".

Si la base de faits de ce système expert est alimentée suivant une cadence définie par la vitesse d'inférence du moteur et du rythme des observations du générateur de faits, les conclusions fournissent des informations caractérisant l'état du processus en fonction du temps.

A partir de ces informations il est possible de conclure sur l'évolution du processus.

Pour cela une base de faits est chargée avec les faits caractérisant son état à l'instant t_0 et avec les faits caractérisant celui de l'instant t_1 (ces différents faits représentant en quelque sorte l'histoire des données).

Un deuxième moteur d'inférence permet, à partir de cette base de faits et d'une base de règles dite "d'évolution", de définir le caractère dynamique de l'état du processus en terme d'évolution.

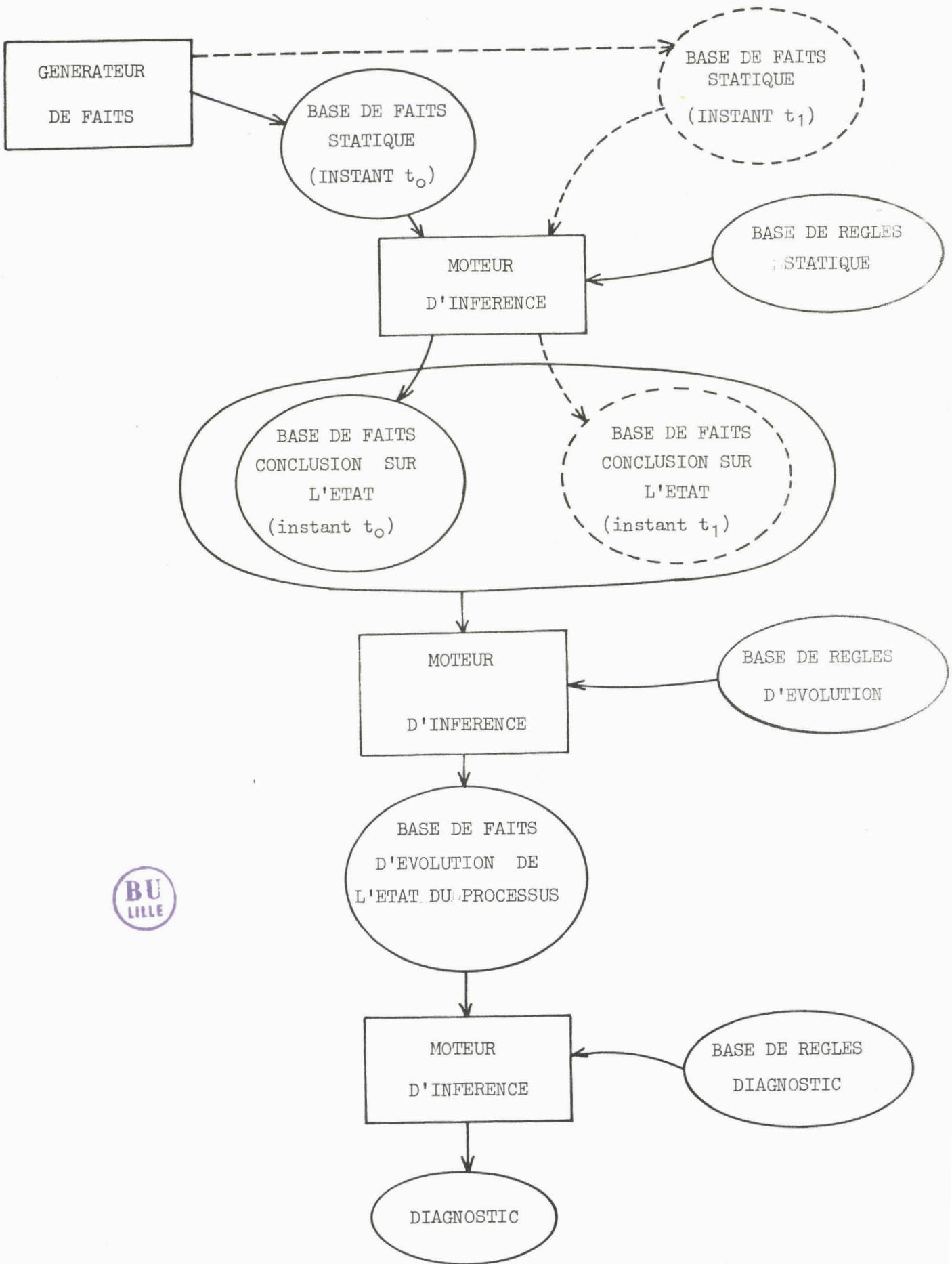


Figure V.8

A partir de la définition de l'évolution du processus et en fonction d'une base de règle "diagnostic", un troisième moteur d'inférence permet de conclure en proposant un diagnostic intégrant la notion de temps.

Les structures présentées en figure V.5 et V.6 se prêtent bien à cette hiérarchisation de la connaissance et de l'inférence dans le sens où un système expert alimente la base de faits du système expert suivant.

Cette structuration de la connaissance et de l'inférence permet d'étendre les champs d'application des systèmes experts en offrant la possibilité de traiter les aspects temporels et évolutifs de l'information rencontrés dans le contrôle de processus.

V CONCLUSION : SYSTEME EXPERT EN LIGNE

Ce chapitre a présenté différentes améliorations qu'il est possible de réaliser sur un moteur d'inférence afin de diminuer les temps de réponse d'un système expert.

Ces améliorations portent :

- > sur l'utilisation d'une technologie plus rapide :
 - (1) compilation du programme du moteur d'inférence sous TURBO-PASCAL (*)
 - (2) l'utilisation d'un microordinateur plus rapide (*)
- > sur l'utilisation de meilleurs algorithmes :
 - (1) suppression des faits vérifiés (*)
 - (2) suppression des règles appliquées (*)
 - (3) suppression des règles non applicables
 - (4) mémorisation des faits confirmés en chaînage arrière (*)
- > sur la structure interne de la base de connaissance par une meilleure technique de compilation :
 - (1) diminution du nombre de listes à parcourir
 - (2) faits directement reliés à leur valeur logique
- > sur l'introduction de parallélisme :
 - (1) conception d'une tâche ayant pour fonction de supprimer les faits vérifiés et les règles non activables (règle possédant un fait faux dans sa partie condition).
 - (2) structuration de la base de faits et de la base de connaissance en niveaux hiérarchiques autorisant un parallélisme dans le fonctionnement des niveaux.

Certaines de ces techniques ont été testées (marquées par un astérisque) et ont permis de diminuer considérablement les temps de réponse.

Ainsi pour une même base de connaissances (possédant une centaine de règles) la résolution est passée de 3 minutes 5 secondes à 15 secondes en chaînage avant et de 41 minutes à 6 secondes en chaînage arrière.

Les autres techniques non testées devraient permettre également un gain de temps et il serait donc intéressant de les mettre en oeuvre.

De même la conception d'une structure interne de la base de faits et de la base de connaissance en différents niveaux hiérarchiques (cf. figure V.8) permet l'introduction de l'aspect temporel et évolutif de l'information dans les faits et dans la connaissance.

Cet aspect temporel, formulé en terme d'évolution, de tendance, permet de conserver la notion de connaissance factuelle et donc de garder le formalisme de la logique des propositions pour représenter l'expertise du domaine.

La possibilité d'introduire une notion de temps dans la connaissance représente une grande qualité pour un système expert.

En effet dans la surveillance et/ou la conduite de processus, l'information peut évoluer en cours d'analyse.

Or la plupart des systèmes experts ne permettent pas la prise en compte de cette évolution, car ils raisonnent sur une base de faits où la valeur de vérité d'une donnée ne peut être modifiée si elle a déjà été établie.

De plus la structure interne des connaissances décrite figure V.5 (pour un niveau) permet à un dispositif externe d'alimenter la base de faits, ce qui autorise ainsi le contrôle/conduite de processus.

La plupart des systèmes experts en place ne sont pas intégrés à un ensemble automatisé générant les faits et ceci à notre avis pour plusieurs raisons :

--> Taille importante des bases de faits et de connaissance qui ne permettent pas des applications portables et des remises à jour rapides

--> Lenteur des mécanismes d'inférence

et également parce que la plupart des applications actuelles des systèmes experts, et plus particulièrement au domaine médical, nécessitent plus l'intégration de bases de

connaissances importantes (MYCIN : 500 règles) que des temps de réponse rapides.

Cependant l'expérience en cours de réalisation tend à montrer que dans le cas où la base de connaissance peut s'exprimer en un nombre de règles relativement faible (une centaine) et que l'ensemble des faits fournis par le générateur de faits est également restreint, il est possible de conclure sur l'état du patient au rythme des observations.

Ainsi le couplage du générateur de faits décrit en première partie au système expert aboutit à l'organisation présentée par le synoptique (figure V.9) /COUV-85b/.

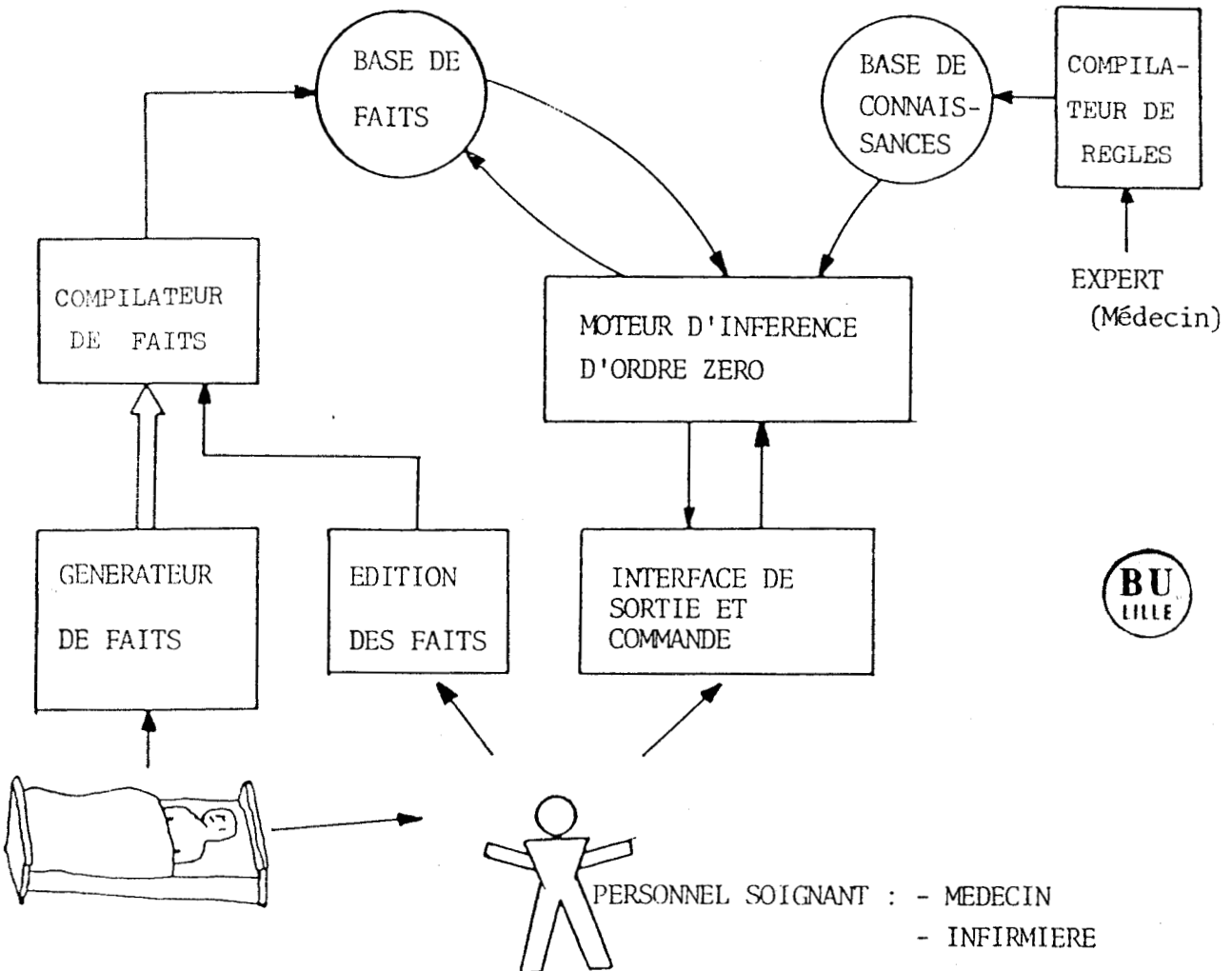


Figure V.9

La base de faits est perpétuellement remise à jour soit par le générateur de faits toutes les 15 secondes, soit par l'opérateur pour apporter des informations qui ne sont pas fournies par ce générateur.

L'ensemble de ces faits sont traités par un compilateur de faits qui transforme la représentation des informations dans le but d'une manipulation plus rapide par le moteur d'inférence.

La base de connaissance est générée par l'expert au travers d'un compilateur de règles organisant la connaissance suivant la structure arborescente développée précédemment, sur laquelle travaille le moteur d'inférence.

Le moteur d'inférence est un dispositif qui peut répondre aux commandes suivantes :

- > Déduction et explication de l'état du patient et de son évolution.
- > Confirmation d'hypothèse proposée par l'utilisateur.
- > Complément d'informations dans une procédure de vérification d'hypothèses.
- > Explication du raisonnement par l'indication de l'ordre des règles activées.



CONCLUSION GENERALE

Le travail que nous avons présenté conduit à définir le concept de base d'une chaîne d'observation de processus complexes et de prise de décision à partir des états observés, dans un contexte d'exécution en temps réel.

Cette chaîne de traitement de l'information peut être décomposée en deux parties :

- > D'une part la constitution d'un observateur qui permet, en temps réel, d'appréhender les informations brutes issues d'un processus, et d'en extraire les paramètres caractérisant son état.
- > D'autre part, la mise en place d'un système d'aide à la décision qui traite en ligne les résultats de l'observation afin de proposer un diagnostic en temps réel.

La réalisation d'un observateur numérique complexe nécessite l'adoption d'une méthode de conception de systèmes de contrôle/commande.

Si la méthode développée par CALVEZ /CALV-82/ et servant de fil conducteur à la conception du système d'observation présenté au chapitre III fournit une réponse aux nombreux problèmes de conception des systèmes de commande en temps réel, nous pensons qu'il est intéressant de développer et d'affiner le concept de "boîtes noires" décrit par les différents modules en y intégrant des critères de mesure qualitatifs et quantitatifs de la solution proposée.

En effet, dans la méthode proposée par CALVEZ, il n'est développé aucun moyen de juger de la qualité de la solution si ce n'est qu'elle répond au problème posé et qu'elle satisfait les contraintes de temps.

Ces critères de qualité peuvent s'appuyer sur la méthode de conception de logiciels complexes décrit dans /MEIL-80/.

En effet, YOURDON et ses collaborateurs se sont intéressés à la conception structurée qui est une approche descendante permettant de diminuer la complexité des systèmes en les décomposant en boîtes noires et en les hiérarchisant (à la manière de la méthode de CALVEZ).

Cette approche offre en outre un ensemble de critères qui permet d'évaluer la qualité de la solution proposée.

La décomposition descendante développée par YOURDON met en évidence deux critères essentiels :

- > Le critère de couplage :
qui définit le degré de dépendance entre deux modules.
- > Le critère de cohérence :
qui représente la mesure de l'association fonctionnelle d'éléments dans un module; un élément pouvant être une instruction ou un groupe d'instructions ou un appel à un module.

Le but de la conception structurée est de réaliser des modules dont les éléments sont fortement et véritablement reliés aux autres éléments du même module (cohérence) mais indépendants des autres modules (couplage).

Le respect de ces critères de qualité lors de l'utilisation de la méthode de conception de CALVEZ permettrait d'affiner la définition de modules à couplage serré en considérant des facteurs autres que le respect des contraintes de temps.

Ainsi la synthèse de ces deux méthodes en définirait une nouvelle qui faciliterait le travail de conception et, de par son jeu de critères de qualité, définirait des modules cohérents et à faible couplage minimisant les retours-arrière dans la succession des différentes étapes lors de la conception des systèmes de commande en temps réel.

L'étape suivante de cette étude est consacrée aux traitements en ligne des observations au moyen d'un système expert afin de répondre au problème de l'aide à la décision.

Néanmoins la difficulté essentielle de ce type d'application des systèmes d'aide à la décision réside en deux points :

- > Le couplage en ligne du système expert au générateur de faits constitué par le système d'observation.
- > Le problème des temps de résolution de ces systèmes.

En effet la principale contrainte qui ralentit l'utilisation des systèmes experts dans les applications temps réel est le problème de temps de réponse.

C'est pourquoi nous nous sommes intéressés à améliorer les performances d'un moteur d'inférence d'ordre zéro s'appliquant à une base de connaissances peu importante.

L'étape suivante ou le prolongement de cette étude ne doit pas se limiter à ces améliorations.

La diminution des temps de réponse des systèmes experts définit plusieurs axes d'étude présentés au chapitre V.

Ces axes sont :

- > L'utilisation d'une meilleure structure interne des connaissances par le biais de la compilation de la base de règles et de la base de connaissances.
- > L'introduction de tâches parallèles dans le cycle de résolution.
- > La structuration de l'univers des connaissances en hiérarchisant le savoir tant au niveau de l'expertise du domaine que du cas traité.

De plus le dernier axe autorise l'introduction de la notion de temps dans les faits ce qui permet de traiter des données temporelles et donc de prendre en compte l'évolution du processus observé, ce qui représente un principe primordial dans la surveillance en continu d'un processus.

L'étude qui a été réalisée trouve son support dans l'observation d'informations électrophysiologiques dans le contexte de la réanimation.

Toutefois le domaine d'application de cette chaîne observation-prise de décision ne se limite pas au contexte biomédical mais peut se généraliser au domaine très vaste que constitue la surveillance ou le contrôle en continu des processus.

ANNEXES

Annexe 1 : Vérification du caractère opérationnel pour les processeurs /CALV-82/

Un processeur est dit opérationnel si son taux d'occupation à tout instant est inférieur à 1 et si toutes les contraintes de temps pour les actions se trouvent respectées.

Une telle évaluation est faite à partir des fréquences d'activation et les temps d'exécution de ces actions par le processeur retenu pour l'implantation.

I) Temps d'exécution maximum pour une action : cas le plus défavorable

Considérons une action temporaire activable par n événements acceptés en entrée. A chaque événement E_i , est associé une région gardée RG_i dont la durée d'exécution dans le cas le plus défavorable est T_{RGi} .

Les régions gardées sont exécutées en exclusion mutuelle.

La durée minimale entre deux apparitions consécutives d'un événement E_i est : $1/F_{Ei}$. Les événements sont supposés totalement asynchrones.

Pour une seule action, il s'agit de déterminer, dans le cas le plus défavorable, le taux d'occupation du processeur chargé de son exécution.

C'est un problème d'ordonnancement de plusieurs processus cycliques, sur une ressource non requérable de manière à garantir l'exclusion mutuelle. Des résultats présentés dans /LIU-73/, /LABE-74/, /LEIN-80/, sont applicables à ce problème.

Tout d'abord, le cas le plus défavorable apparaît lorsqu'une région gardée est activée, alors que toutes les autres viennent d'être activées (simultanéité de tous les événements). Ensuite, pour une politique d'ordonnancement : première activation --> première exécution - entre régions gardées (solution qui évite la privation), la date de fin d'exécution de la dernière région gardée par rapport à l'instant d'apparition simultanée de tous les événements est

$$T_A = \sum_{i=1, n} T_{RGi}$$

Pour éviter toute perte d'événements, il faut s'assurer que :

$$\forall_i : 1/F_{Ei} > T_A$$

Le cas le plus défavorable concerne la région gardée R_{Gi} telle que :

$$1/F_{Ei} = \min_{\forall j = 1, n} (1/F_{Ej})$$

Le taux d'occupation maximum du processeur chargé de l'exécution de l'action est donc :

$$T_{Omax} = \sum_{i=1, n} T_{Rgi} * \max_i (F_{Ei})$$

En respectant cette condition, l'instant de fin d'exécution de toute région gardée, par rapport à l'instant d'activation, est toujours :

$$D_{Rgi} < \sum_{j=1, n} T_{RGj}$$

II) Taux d'occupation maximum d'un processeur pour plusieurs actions

Compte tenu des résultats donnés au paragraphe précédent, une action A_i sera considérée activée à la fréquence $F_{Ai} = (\max_{\text{sur } j} F_{Ej})$ pour i .

Pour n actions exécutées sur un processeur requérable, il faut s'assurer que : $T_{Omax} = \sum_{i=1, n} F_{Ai} * T_{Ai} < 1$.

Il faut ensuite choisir une politique d'ordonnancement de manière à satisfaire toutes les contraintes de temps.

Dans /LIU-73/ et /LABE-74/, il est démontré que, pour un ordonnancement à priorité fixe, l'utilisation du processeur est optimal si on choisit :

$$\text{priorité } A_i > \text{priorité } A_j \text{ pour } F_{Ai} > F_{Aj} \quad \forall i, j$$

Calvez considère que cette solution simple est satisfaisante. En effet, il n'est pas utile de mieux optimiser l'utilisation des microprocesseurs, car le prix est faible, et toute autre solution se traduit par une plus grande complexité de l'ordonnanceur et donc introduit des retards plus importants à l'exécution des actions.

III) Evaluation des retards maximums à l'exécution pour la vérification des contraintes de temps

Pour l'ordonnancement à priorité fixe, supposons les priorités telles que :

$$\text{priorité } A_1 > \dots \text{priorité } A_i > \dots \text{priorité } A_n$$

Soient - RD_i le retard maximum pour le début d'exécution de l'action A_i
- RF_i le retard maximum pour la fin d'exécution de l'action A_i

$$\begin{array}{ll} \text{Pour } A_1 : RD_1 = 0 & RF_1 = TA_1 \\ A_2 : RD_2 = TA_1 & RF_2 = TA_2 : (1 - TO_1) \\ \vdots & \\ \vdots & \\ A_i : RD_i = TA_{i-1} : (1 - \sum_{j=1}^{i-1} TO_j) & RF_i = TA_i : (1 - \sum_{j=1}^{i-1} TO_j) \end{array}$$

Ces valeurs permettent de vérifier pour chaque action le respect des contraintes de temps.

IV) Evaluation du temps d'exécution maximum pour une région gardée

Le respect des contraintes de temps se déduit de l'évaluation des temps d'exécution pour chaque région gardée.

Chaque région gardée est purement séquentielle. Des lectures et écritures de variables partagées ou des attentes sur événements ou messages peuvent introduire des retards importants.

Pour les événements et messages, il faut prendre la durée d'attente la plus longue.

La durée d'exploitation d'une variable partagée est fonction du mécanisme retenu pour assurer l'accès en mutuelle exclusion.

Il est montré dans /KAIS-81/ qu'il existe un tournoi conflictuel entre l'ordonnancement à priorité fixe et l'exclusion mutuelle.

Soient n actions exécutées par un processeur et se partageant une même variable. Si l'action A_n , la moins prioritaire, accède à la variable partagée, et que toutes les autres actions deviennent actives, A_n est interrompue. Toutes les autres actions sont exécutées jusqu'à l'accès à la section critique, puis A_n est poursuivie pour la libérer. Les autres actions peuvent alors achever leur exécution.

Le temps d'exécution d'une région peut donc devenir important et ainsi impliquer un non-respect des contraintes de temps. Le remède proposé /KAIS-81/ consiste à modifier temporairement la priorité de l'action lorsqu'elle entre en section critique.

--> L'invalidation des interruptions est efficace, mais certaines autres contraintes de temps peuvent ne plus être satisfaites. Le retard d'accès maximum est alors égal dans ce cas à la durée de la plus longue section critique implantée sur le processeur.

--> L'augmentation de la priorité à une valeur supérieure à celles de toutes les actions utilisant la section critique est un bon compromis, mais nécessite l'implantation de mécanismes de synchronisation spécifiques.

ANNEXE 2 :

Liste de systèmes experts existants, classés par catégorie d'application avec l'indication du domaine d'application et la référence bibliographique.

CLASSIFICATION - DIAGNOSTIC

DOM	DOMAINE D'APPLICATION	REFERENCE
BEL	Médecine	/PATI-81/
BTENDING	Suivie d'anesthésie	/MILL-83/
ADACEUS	Médecine	/POPL-82/
ASNET	Médecine	/KULI-82/
ESSOL	Reconnaissance géotechnique	/AYEL-84/
LOT	Problème de coagulation	/BENN-80/
ART	Maintenance	/BENN-81/
OUET	Maintenance logicielle	/FOUE-83/
EADMED	Psychopharmacologie - Pratique psychiatrique	/HEIS-78/
EMOCAD	Médecine	/ANDR-84/
INTERNIST	Médecine	/POPL-75/
EDIC	Médecine	/SCHW-84/
YCIN	Médecine	/SHOR-76/
NCOCIN	Médecine	/SHOR-81/
ARCY	Nucléaire	/PARC-82/
IP	Médecine	/PAUK-77/
ROSPECTOR	Géologie	/KONO-79/
ROTIS	Médecine	/SOUL-81/
UFF	Médecine	/AIKI-80/
AM	Médecine	/GASC-81/
ELF	Médecine	/KOHL-84/
PHINX	Médecine	/FIES-84/
ROPICAID	Médecine	/AUVE-84/
PLAIN	Thérapeutique	/SWAR-83/



SURVEILLANCE - CONTROLE

NOM	DOMAINE D'APPLICATION	REFERENCE
ANA	Robotique (gestion des stocks)	/DERM-79/
ARGOS II	Robotique (simulation, contrôle)	/CAYR-79/
DIPMETER	Géologie : analyse de signaux	/DAVI-81/
LITHO	Géologie : analyse de signaux	/BONN-82/
PATREC/MDX	Aide à la gestion de base de données	/MITT-80/
PEACE	Electronique : analyse de circuit	/DINC-79/
RESEDA	Histoire : analyse de document	/ZARR-81/
SU/X - HASP	Identification et localisation d'objet émetteur dans l'espace	/NII -78/
V.M.	Surveillance en réanimation	/FAGA-80/

RECHERCHE HEURISTIQUE

NOM	DOMAINE D'APPLICATION	REFERENCE
AM	Mathématiques : démonstration automatique	/LENA-76/
CRYSSALYS	Médecine	/ENGE-79/
DENDRAL	Chimie organique	/FEIG-71/
MACSYMA	Mathématiques : manipulation de symbole	/MATH-74/
METADENDRAL	Chimie organique	/BUCH-78/
MOLGEN	Biologie moléculaire, génétique	/STEF-78/
PARADISE	Jeu : échecs	/WILK-79/
PASTRE	Mathématiques : théorie des ensembles démonstration de théorèmes	/PAST-78/
POPESCU	Jeu : bridge	/POPE-82/
QUINLAN	Jeu : bridge	/WILK-79/
SECHS	Chimie	/WIPK-76/
SIMMIAS	Recherche pétrolière	/LAUR-81/
SYNCHEM	Chimie	/GELE-77/
WATERMAN	Jeu : poker	/WATE-70/



CFAO (Conception - Fabrication Assistées par Ordinateur)

NOM	DOMAINE D'APPLICATION	REFERENCE
APE	Informatique	/BART-81/
AARON	Production de dessin	/COHE-79/
EL	Electronique	/SUSS-75/
GARI	Gamme d'usinage	/DESC-81/
NASL	Electronique	/DERM-78/
NUDGE	Elaboration emploi du temps	/GOLD-77/
PECOS	Informatique	/BARS-79/
PSI	Informatique : programmation automatique	/GREE-76/
R1	Informatique	/DERM-81/
RITA	Conception assistée	/ANDE-76/
SACON	Mécanique : ingénieur conseil	/MICH-80/
SAFE	Informatique : programmation automatique	/BALZ-74/
TROPIC	Architecture - Transformateur électrique	/LATO-77/

EAO (Enseignement Assisté par Ordinateur)

NOM	DOMAINE D'APPLICATION	REFERENCE
BACON 3	Physique	/LANG-79/
BUGGY	Mathématiques	/BROW-78/
GUIDON	Médecine	/CLAN-79/
MECHO	Mécanique	/BUND-79/
REINART	Médecine : aide à l'éducation des malades	/BETA-84/
SOPHIE	Electronique	/BROW-75/



SIMULATION

NOM	DOMAINE D'APPLICATION	REFERENCE
ACT	Simulation : expérience en psychologie	/ANDE-79/
POLITICS	Simulation : face à des événements politiques	/CARB-78/

AIDE A LA CONCEPTION DE SYSTEMES EXPERTS

NOM	DOMAINE D'APPLICATION	REFERENCE
AGE	Aide à la construction de systèmes experts	/NII-78b/
EMYCIN	Aide à l'acquisition de connaissances (moteur nu)	/VANM-79/
EXPERT	Diagnostic. Aide à l'acquisition de connaissances	/WEIS-79/
HEARSAY III	Compréhension automatique de la parole	/BALZ-80/
KAS	Aide à l'acquisition de connaissances	/REBO-79/
MEDIT	Aide à l'acquisition de connaissances	/SCHW-84/
METALOG	Moteur nu	/DINC-80/
OPS - OPS.5	Moteur nu	/FORG-77/
PROLOG	Moteur et langage	/COLM-77/
PS	Moteur nu	/RYCH-76/
ROSIE	Aide à la construction de systèmes experts	/WATE-79/
SNARK	Moteur	/LAUR-82/
TANGO	Moteur d'inférence	/CORD-82/
TEIRESIAS	Aide à l'acquisition de connaissances	/DAVI-79/



BIBLIOGRAPHIE

BIBLIOGRAPHIE

- /ADI -84/ AGENCE DE L'INFORMATIQUE
"Les ateliers de génie logiciel"
Génie logiciel N. 1, 1984.
- /AIKI-80/ AIKINS J.S.
"Prototypes and production rules : a knowledge
representation for computer consultations."
Technical Report Memo HPP-80-17, Stanford Heuristic
Programming Project, August 1980.
- /ANDE-76/ ANDERSON R., GILLOGLY J.
"Rand intelligent terminal agent RITA design and
philosophy"
R 1809 ARPA, 1976
- /ANDE-79/ ANDERSON J., KLINE P.
"A learning system and its psychological implication"
Proc. IJCAI 6, 1979, p 16-21.
- /ANDR-84/ ANDREUX J.P., ALHENC-GELAS M.
"HEMOCAD: aide au diagnostic et à l'enseignement en
hémostase"
Actes des Journées Médecine Informatique 1984
IIRIAM Marseille, 1984
- /AUVE-84/ AUVERT B., VAN LOOK F., LE THI HUONG DU, AEGERTER P.,
GILBOS V., EMMANUELLI X., BOSSEAU J.F., BOUTIN P.,
LANDRE M.F., PHILIPPE A.M.
"Conception d'un système général d'aide à la
décision médicale pour les infirmiers des
pays en voie de développement "
Actes des Journées Médecine Informatique 1984
IIRIAM Marseille, 1984
- /AYEL-84/ AYEL, LAURENT, SOUTIF
"CESSOL: Un système expert pour définir des campagnes
de reconnaissances de sol.
4ème Congrès RF. IA. Paris, Janvier 1984

- /BALZ-74/ BALZER R. et Al.
"Domain independent automatic programming "
ISI/RR-73-14, Information Science Institute, March 74.
- /BALZ-80/ BALZER R., ERMAN L.D., LONDON P., WILLIAMS C.
"HEARSAY-III: A domain independent framework for expert systems"
Proceeding of the First Annual National Conference on Artificial Intelligence, 1980, p 108-110
- /BARS-79/ BARSTOW D.R.
"An experiment in knowledge based automatic programming"
Artificial Intelligence 12, 1979, p 73-119
- /BART-81/ BARTELS U., OTTHOFF W., RAWLEFS P.
"APE: a system for automatic programming from abstract specifications of data types and algorithms"
Proc. IJCAI 7, 1981, p 1037-1043
- /BENN-80/ BENNET J.S., GOLDMAN D.
"CLOT: a knowledge based consultant for bleeding disorders"
Memo HPP-80-7, 1980
- /BENN-81/ BENNET J.S., HOLLANDER C.R.
"DART: an expert system for computer fault diagnosis"
Proc. IJCAI 7, 1981, p 843-845
- /BETA-84/ BETTAILLE H., MASSOTE A.M., MAURY M.
"REINART: système expert appliqué au rein artificiel"
Actes des Journées Médecine Informatique 1984
IIRIAM Marseille, 1984
- /BLAN-78/ BLANCHARD M.
"Rapport de la commission de la normalisation de la représentation du cahier des charges d'un automatisme logique."
Automatisme tome 23, N. 34, Mars-Avril 78, p 66-83
- /BONN-82/ BONNET A.
"LITHO: un système expert en géologie"
Actes des 4èmes journées francophones
IMAG, Université de Genève, 1982

- /BRAR-80/ BRARD P., TOULOTTE J.M.
"L'automate programmable avec implantation direct de graphes fonctionnels est-il une solution d'avenir ?"
Le Nouvel automatisme - Janv. Fev. 1980, p 39-43
- /BROW-75/ BROWN J.S., BURTON R.
"Multiple representations of knowledge for tutorial reasoning "
In Bobrow et Collins (eds).
Representation and Understanding Academic Press
New York, 1975
- /BROW-78/ BROWN J.S., BURTON R.
"Diagnostic models for procedural bugs in basic mathematical skills"
Cognitive Science 2, 1978, p 155-192
- /BUCH-78/ BUCHANAN B., FEIGENBAUM E.
"DENDRAL and METADENDRAL"
Artificial Intelligence 11, 1978, p 5-24
- /BUND-79/ BUNDY A., BYRD L., LUGER G., MELLISH C., MILNE R., PALMER M.
"MECHO: a program to solve mechanics problems"
Dept of AIU of Edinburgh, Working Paper 50, 1979
p 1-104
- /CALV-82/ CALVEZ J.P.
"Une méthodologie de conception des multi-micro-ordinateurs pour les applications en temps réel"
Thèse d'Etat - ENSM - Nantes, 6 Nov. 82.
- /CALV-83/ CALVEZ J.P., THOMAS Y.
"Méthodologie de conception pour les multi-micro-ordinateurs de commande des processus complexes."
Rapport final ATP N. 9.81.63, Mai 83, Nantes
- /CARB-78/ CARBONNEL J. Jr
"POLITICS: automated ideological reasoning"
Cognitive Science 2, 1978, p 27-51
- /CAYR-79/ CAYROL M., FADE B., FARRENY H.
"Objets formels et liaisons d'attributs dans ARGOS II"
2ème Congrès AFCET de reconnaissances de formes et intelligence artificielle, TOULOUSE, 1979

- /CHAM-79/ CHAMBRIN M.C.
"Réalisation à l'aide de microprocesseurs d'un système de surveillance de l'efficacité ventilatoire en réanimation respiratoire."
Thèse Docteur Ingénieur, USTL, 10 Déc. 79
- /CHAN-83/ CHANGEUX J.P.
"L'homme neuronal"
Ed. Fayard, Coll. Le temps des sciences, 1983
- /CHAU-79/ CHAUDOUARD C.
"Système temps réel pour commande de processus."
Thèse 3ème cycle , ENSM, Nantes, 19 Déc. 79
- /CLAN-79/ CLANCEY W.J.
"Tutoring rules for guiding a case method dialog"
Int. J. of Man Machine Studies 11, 1979, p 25-49
- /COHE-79/ COHEN H.
"What is an image"
Proc. AJCAI 6, 1979, p 1028-1057
- /COLM-77/ COLMERAUER A.
"Programmation en logique du premier ordre"
Actes des journées "La Compréhension" IRIA 77
- /CONN-80/ CONNOR M.F.
"Structured analysis and design technic SADT."
Portfollio 32.04.02 System development Management
Anerbach publishers, 1980.
- /CORD-82/ CORDIER M.O., ROUSSET M.C.
"TANGO: moteur d'inférences pour un système expert avec variables"
Publication du GR 22 30, Utilisation de Connaissances Déclaratives, 1982, p 83-117
- /COUV-85/ COUVREUR M., DUPONT D.
"Expert system on line - Building of a real time control unit for a patient in intensive care"
MIMI' 85, Sant Felliù, June 25-28 1985

- /COUV-85b/ COUVREUR M., DUPONT D.
"Système expert en ligne - Application au traitement
et à l'interprétation de signaux d'origines médicales"
4ème Symposium Franco-Soviétique de G.B.M.
Toulouse, 16/20 Sept. 1985
- /DAVI-79/ DAVIS R.
"Interactive transfer of expertise : acquisition of
new inference rules"
Artificial Intelligence, vol. 12, Aout 79, p 121-157
- /DAVI-81/ DAVIS R., AUSTIN H., CARLBON I., FRAWLEY B.,
PRUCHNICK P., SNEIDERMAN R., GILREATH J.A.
"The DIPMETER advisor: interpretation of geological
signals"
Proc. IJCAI 7, 1981, p 846-849
- /DEMA-78/ DE MARCO T.
"Structured analysis and system specification"
New York : Yourdon Press, 1978
- /DERM-78/ Mc DERMOTT D.
"Planning and acting"
Cognitive Science 2, 1978, p 71-109
- /DERM-79/ Mc DERMOTT J.
"Learning to use analogies"
Proc. IJCAI 6, 1979, p 568-576
- /DERM-81/ Mc DERMOTT J., STEELE B.
"Extending a knowledge-based system to deal with ad-hoc
constraints"
Proc. IJCAI 7, 1981, p 824-828
- /DESC-81/ DESCOTTES Y.
"GARI: Un système expert pour la conception des gammes
d'usinages"
Thèse 3eme Cycle, IMAG, Grenoble 1, 1981
- /DIAZ-82/ DIAZ M.
"Modeling and analysis of communication and cooperation
protocols using PETRINET based models ."
COMPUTER NETWORKS, Vol. 6, N. 6, Déc. 82, p 419-441

- /DINC-79/ DINCBAS M.
"PROLOG et un exemple de programmes expert écrit en PROLOG"
Doc. CERT 2/3/22, 1979
- /DINC-80/ DINCBAS M.
"Le système de résolution de problèmes METALOG"
Rapport CERT/DERI 3146 convention DRET 79, 1246
Toulouse, 1980
- /DOD -82/ DOD software initiative
"Le logiciel au coeur des moyens tactiques et stratégiques "
TSI, Vol 1, N. 6, 1982, p 511,515.
- /DOMI-84/ DOMINE F., DUPONT D., MEHADJI M., CHAMBRIN M.C.,
COUVREUR M., CHOPIN C.
"Conception d'un appareillage intégré de surveillance individuelle des malades en réanimation."
Actes 2ème Forum Jeunes Chercheurs, LILLE, Sept. 84
Paru dans Innov. Technol. Biol. Med.
Vol. 6, N. 3, 1985
- /DUBO-85/ DUBOIS D., PRADE M.
"Introduction aux systèmes experts de gestion"
Le traitement de l'imprécision et de l'incertitude dans les modèles de raisonnement des experts
EYROLLES, Paris , 1985.
- /ENGE-79/ ENGELMORE R., TERRY A.
"Structure an function of the CRYSSALIS system"
Proc. IJCAI 6, 1979, p 250-256
- /FAGA-79/ FAGAN L., KUNZ J., FEIGENBAUM E., OSBORN J.
"Representation of dynamic clinical knowledge: Measurement interpretation in the intensive care unit"
Proc. IJCAI 6, 1979, p 260-262
- /FAGA-80/ FAGAN L.M.
"VM: representating time-dependent relations in a medical setting"
PhD thesis, Stanford University, June 1980

- /FARR-80/ FARRENY H.
"Un système pour l'expression et la réalisation de problèmes orienté vers le contrôle de robots"
Thèse d'Etat, Université Toulouse III, Sept. 1980
- /FARR-85/ FARRENY H.
"Les systèmes experts : principes et exemples"
CEPADUES Editions , TOULOUSE , 1985.
- /FATH-83/ FATHI E.T., KRIEGER M.
"Multiple Microprocessor systems : What, Why, When"
IEEE Computer, Mars 83, p 23-32.
- /FEIG-71/ FEIGENBAUM E.A., BUCHANAN B.G., LEDERSBERG J.
"On generality and problem solving: a case study using the DENDRAL program"
Machine Intelligence 6, B. MELTZER and D. MICHIE (eds)
New York American Elsevier, 1971, p 165-190
- /FERB-84/ FERBER J.
"Les systèmes experts (II) : Un moteur d'inférence en Pascal"
Microsystème, Mai 1984
- /FIES-83/ FIESCHI M.
"SPHINX: Un système expert d'aide à la décision en médecine."
Thèse de doctorat d'état en biologie humaine, 1983
Fac. de Médecine de l 'Université d'Aix Marseille II
- /FIES-84/ FIESCHI M.
"Intelligence artificielle en médecine: des systèmes experts"
Ed. Masson, 1984
- /FORG-77/ FORGY C., Mc DERMOTT J.
"OPS : a domain independent production system"
Proc. IJCAI 5, 1977, p 833-839
- /FORG-81/ FORGY C.L.
"OPS'5 user's manual"
Technical Report CMU-CS-81-135
Carnegie-Mellon University, Pittsburgh, 1981

- /FOUE-83/ FOUET J.M.
"Un système de compréhension de programmes"
Globule 4, Mai 1983
- /GASC-81/ GASCUEL O.
"SAM: Un programme général d'aide à la décision médicale
structurant automatiquement ses connaissances"
Congrès AFCET-RFIA Nancy, 1981
- /GELE-77/ GELERNTER H.L. et Al
"Empirical explorations of SYNCHEM"
Science, 9 Sept. 1977, p 1041-1049
- /GEMM- / Document ADEPA
"Le GEMMA, Guide d'Etude des Modes de Marches et
d'Arrêts."
- /GOLD-77/ GOLDSTEIN I., ROBERTS R.
"NUDGE: A knowledge based scheduling program"
Proc. IJCAI 5, 1977, p 257-263
- /GOND-83/ GONDRAN M , LALEUX J.C.
"La représentation des connaissances en fiabilité "
Note EDF 4600/02, 1983.
- /GRAF-77/ Rapport final de la commission AFCET
"Normalisation de la représentation du cahier des
charges d'un automatisme logique"
Automatique et Informatique Industrielle, N. 61 62
Nov/Déc 1977.
- /GREE-76/ GREEN, CORDELL
"The design of the PSI program synthesis system"
Proc. of the Second International Conference on
Software Engineering, 1976
- /GROS-85/ GROS J., BERNAD C.
"Droit -système expert et aide à la décision "
5ème journée internationale:
Les systèmes experts et leurs applications
Avignon, Mai 85, p 777-836.

- /GUPT-85/ GUPTA A.
"Parallelism in production systems : the sources
and the expected speed up"
5èmes Journées Internationales : Les systèmes experts
et leurs applications, Avignon, Mai 1985, p 25-57
- /HEIS-78/ HEISER J.F., BROOKS R.E.
"Design considerations for a clinical psychopharmacology
advisor"
Proc. 2nd Annual Symposium Computer Applications In
Medical Care, 1978, p 278-285
- /KEIS-81/ KAISER C.
"De l'utilisation de la priorité en présence
d'exclusion mutuelle"
Rapport de Recherche, N. 84, INRIA, Juil. 81
- /KNIC-85/ KNICKERBOCKER C.G., MOORE R.L., HAWKINSON L.B.,
LEVIN M.E.
"The PICON expert system for process control"
5èmes Journées Internationales : Les systèmes experts
et leurs applications, Avignon, Mai 1985, p 59-66
- /KOHL-84/ KOHLER F., MONCHOVET S., GROUSSIN M., MARTIN J.
"SELF: un générateur simple d'aide à la décision
médicale, application de quelques outils logiques"
Actes des Journées Médecine Informatique 1984
IIRIAM Marseille, 1984
- /KONO-79/ KONOLIDGE K.
"An inference net compiler for the PROSPECTOR rule
based consultation system"
Proc. IJCAI 6, 1979, p 487-489
- /KRAK-82/ KRAKOWIAK S.
"Systèmes intégrés de production de logiciel :
Concepts et réalisations"
TSI, Vol. 1, N. 3, 1982, p 187-200.
- /KULI-82/ KULIKOWSKI C.A., WEISS S.M.
"Representation of expert knowledge for consultation :
the CASNET and EXPERT projects"
In Szolovits P. editor, Artificial Intelligence in
Medecine, Westview Press
1892, AAAS Selected Symposium 51


- /LABE-74/ LABETOUILLE J.
"Ordonnancement des processus temps réels sur une
ressource pré-emptive"
IRIA, Rapport de Recherche, N. 74, Mai 1974
- /LANG-79/ LANGLEY P.
"Rediscovering physics with BACON-3"
Proc. IJCAI 6, 1979, p 505-507
- /LATO-77/ LATOMBE J.C.
"Une application de l'Intelligence Artificielle à la
Conception Assistée par Ordinateur"
Thèse d'Etat, Grenoble, 1977
- /LAUR-81/ LAURIERE J.L., PERROT A.
"Représentation et utilisation de connaissance dans
l'industrie pétrolière"
Journées d'études AFCET sur les Systèmes Expert
Avignon, 1981
- /LAUR-82/ LAURIERE J.L.
"Le système SNARK : symbolic normalized acquisition and
representation of knowledge"
Rapport Institut de Programmation 427, 1982
- /LAUR-82b/ LAURIERE J.L.
"Représentation et utilisations des connaissances
Première partie : Les systèmes experts
Deuxième partie : Représentation des connaissances"
T.S.I., Vol. 1, N. 1 & 2, 1982
- /LECE-84/ LECERF J.M.
"Conception d'une structure logicielle redondante en
vue d'améliorer la sureté des prises de décisions en
temps réel. Application à l'analyse automatique des
signaux physiologiques."
Thèse 3ème cycle USTL, Oct. 1984.
- /LEIN-80/ LEINBAUGH D.W.
"Guaranted response times in a hard-real-time
environment"
IEEE transaction on software engineering
Vol. SE-6, N. 1, Jan. 1980, p 85-91

- /LENA-76/ LENA D.B.
"AM: an interfacial intelligence approach to discovery
in mathematics as heuristic search"
Stanford University, Computer Sc. Dept., AIM 286
July 1976
- /LISS-83/ LISSANDRE M., LACIER P., SKALLI A.
"SAS : un système d'assistance aux spécifications"
Actes des journées BICRE 83, CAP D'AGDE, Oct. 84
p 46-67.
- /LIU -73/ LIU C.L., LAYLAND J.W.
"Scheduling algorithms for multi-programming
in a hard-real-time environment"
J.A.CM., Vol. 20, Jan. 1973, p 46-61
- /LONC-83/ LONCHAMP J.
"Structuration logique en terme d'agents communicants
des applications réparties de commande/contrôle
de processus"
Actes des journées BIGRE 83, CAP D'AGDE, Oct. 83
p 255-273
- /MATH-74/ The MATHLAB Groupe
"The MACSYMA reference manual"
The Mathlab groupe, MIT, Sept. 1974
- /MEIL-80/ MEILIR PAGE-JONES
"The practical guide to structured systems design"
New York : Yourdon Press, 1980.
- /MICH-80/ MICHIE D. (Edit.)
"Expert system in the microelectronic age"
Edinburgh Univ. Press, 1980
- /MILL-83/ MILLER P.L.
"Medical plan analysis by computer"
In Proc. MEDINFO 83.
Van Bommel, Ball, Wigertz (ed.), North Holland, 1983
- /MITT-80/ MITTAL S., CHAN DRASE KARAN B.
"Conceptual representation of patient data bases"
Journal of Medical System, Vol. 4, N. 2
1980, p 169-185

- /MOAL-81/ MOALLA M., DAVID R.
"Extension du GRAFCET pour la représentation de
systèmes temps réel complexes"
RAIRO Automatique /systems analysis and control
Vol. 15, N. 2, 1981, p 159-191.
- /MORI-74/ MORIN G.
"Le champ central"
Ed. E.S.F., 1974
- /MYER-78/ MYERS G.
"Composite/Structured Design"
New York : Van Nostrand Reinhold, 1978
- /NII -78/ NII H.P., FEIGENBAUM E.A.
"Rule based understanding of signals"
Pattern Directed Inference Systems, Waterman et
Hayes-Roth (Ed.), New York : Academic Press, 1978
- /NII-78b/ NII H., ALELLO N.
"AGE : Attempt to generalize"
Work. Paper 5, Stanford, 1978
- /NILS-71/ NILSON N.J.
"Problem solving methods in artificial intelligence"
San Francisco : Mc Graw-Hill, 1971
- /NILS-80/ NILSSON N.
"Principles of Artificial Intelligence"
Tioga Publishing Company, 1980
- /PARC-82/ PARCY J.P.
"Un système expert en diagnostic sur réacteurs à
neutrons rapides"
Thèse 3ème cycle, Université de Marseille-Luniny
1982
- /PAST-78/ PASTRE D.
"Automatic theorem proving in set theory"
Artificial Intelligence 10, 1, 1978, p 1-27

- /PATI-81/ PATIL R.S.
"Causal representation of patient illness for electrolyte and acid-base diagnosis"
Technical Report TR 267
MIT, Laboratory for Computer Sciences, 1981
- /PAUK-77/ PAUKER S.G., SZOLOVITS P.
"Analizing and simulating taking the history of the present illness : context formation"
In Schneider et Sagwall Hein (ed.), Computational Linguistics in Medecine, North Holland, 1977
- /PIGN-81/ PIGNON D.
"Enquête au sein des centrales nucléaires "
Ed F.Nathan, Dossier 90, 1981.
- /POPE-82/ POPESCU R.
"Jeu de la carte au bridge"
Publication du GR 22 n 30, Utilisation de Connaissances Déclaratives, 1982, p 339-360
- /POPL-75/ POPLE H.E. Jr
"DIALOG: a model of diagnosis logic for internal medicine"
Proc. IJCAI 4, Tbilissi, 1975
- /POPL-82/ POPLE H.E. Jr
"Heuristic methods for imposing structure on ill structured problems : the structuring of medical diagnostics"
Artificial Intelligence in Medecine, Szolovits P. (Ed)
AAAS Selected Symposium 51, 1982
- /REBO-79/ REBOH R.
"The knowledge acquisition system. A computer-based consultation for mineral exploration"
R. O. DUDA et AL, Final Report, SRI Project 6415
Artificial Intelligence Center, SRI International
Menlo Park CA, Sept. 1979
- /ROUC-83/ ROUCAIROL G.
"Représentation du parallelisme et modèles de systèmes"
Bulletin de liaison de la recherche en informatique et automatique, N. 83 , 1983.

- /RYCH-76/ RYCHENER M.D.
"Production systems as a programming language for
artificial intelligence applications"
PhD Carnegie Mellon University, Vol. 1 et 2, 1976
- /SCEP-82/ Projet SCEPTRE
"Proposition de standard noyau d'exécutif temps réel"
Rapport BNI n 26/2, Sept. 82.
- /SCHW-84/ SCHWEIN P., HAB B.
"MEDIC: aide à la santé primaire là ou il n'y a pas
de médecins.
MEDIT: aide les médecins à rédiger les données pour
MEDIC"
Actes des Journées Médecine Informatique 1984
IIRIAM Marseille, 1984
- /SIRI-84/ S.I.R.I.C.
"Allô...Moi? Ici les autres
ou quand il s'agit de se comprendre"
Ed. Empirika, Coll. La vie quotidienne à la lumière
du fonctionnement du cerveau, 1984
- /SHOR-76/ SHORTLIFFE E.H.
"Computer based medical consultations : MYCIN"
Americain Elsevier, New York, 1976
- /SHOR-81/ SHORTLIFFE E.H., CARLISLE SLOTT A., BISCHOFF M.B.,
CAMPBELLE A.B., VAN MELLE W., TACOB B.C.D.
"ONCOCIN: an expert system for oncology protocole
management"
Proc. IJCAI 7, 1981, p 876-981
- /SOUL-81/ SOULA G.
"Aide à la décision en logique floue.
Application en médecine"
Thèse Doctorat d'Etat en Biologie Humaine
Université d'Aix Marseille, 4 Mars 1981
- /STEF-78/ STEFIK M.J.
"Inferring DNA structures from segmentation data"
Artificial Intelligence, Vol. 11, 1978, p 85-114

- /SUSS-75/ SUSSMAN G., STALLMAN R.
"Heuristic tecnics in computer-aided circuit analysis"
IEEE Transactions on Circuits and Systems
CAS 22, 11, 1975, p 857-865
- /SWAR-83/ SWARTOUT W.R.
"XPLAIN: a system for creating and explaining expert
consulting programs"
Artificial Intelligence 21, 1983, p 285-325
- /TACO-79/ TACONET B, CHOLLOT B.
"Programmation du grafcet sur automate à langage
logique, à relais ou booléen."
Nouvel automatisme - Janv. Fev. 79, p 41-45
- /VANM-79/ VAN MELLE W.
"A domain-independent production rule system for
consultation programs"
Proc. IJCAI 6, 1979, p 923-925
- /VANM-80/ VAN MELLE W.
"A domain-independent system that aids in constructing
knowledge-based consultation programs"
PhD. Disertation, Comp. Sc. Dep.
Stanford University, 1980.
- /VASS-82/ VASSEUR C.
"La notion d'événement dans les systèmes dynamiques de
détection, classification temps réel et applications à
la conception d'une instrumentation distribuée."
Thèse d'Etat, U.S.T.L. 1, 1982.
- /VERJ-83/ VERJUS J.P.
"Parallelisme et synchronisation "
Bulletin de liaison de la recherche en informatique
et automatique, N. 83, 1983.
- /VIGN-85/ VIGNARD P.
"CRIQUET Version 2, Un outil de base pour construire
des systèmes experts"
Rapport de recherche N. 380, INRIA, Mars 1985
- /WATE-70/ WATERMAN D.
"Generalization and learning of heuristics"
Artificial Intelligence 1, 1970, p 121-170
- 

- /WATE-79/ WATERMAN D.A.
"User-oriented systems for capturing expertise : a rule based approach"
Expert Systems in the Microelectronic Age
Michie D. (ed.), Edinbrough : Edinbrough University Press, 1979, p 26-34
- /WEISS-79/ WEISS S.M., KULIKOWSKI C.A.
"EXPERT : a system for developping consultations models"
Proc. IJCAI 6, 1979, p 942-947
- /WILK-79/ WILKINS D.
"Using plans in chess"
Proc. IJCAI 6, 1979, p 960-967
- /WIPK-76/ WIPKE W.T.
"Computer planning of research in organic chemistry"
Computer in Chemical Education and Reseach
Ludena E.V., Sabelli N.H. and Wahl A.C. (Ed.)
New York : Plenum Press, 1976, p 381-391
- /YOUR-78/ YOURDON E., CONSTANTINE L.
"Structured design : fundamentals of a discipline of computer."
Program and system design
New York : Yourdon Press , 1980.
- /ZARR-81/ ZARRI G.P.
"Building the inference component of an historical information retrieval system"
Proc. IJCAI 7, 1981, p 401-408
- /ZELK-78/ ZELKOWITZ M.V.
"Perspectives on software engineering "
ACM Computing Surveys, Vol. 10, N. 2 , Janv. 78



0

RESUME

Cette étude vise à définir les spécifications d'une chaîne d'observation et de prise de décision dans un contexte d'exécution en temps réel.

La première partie (chapitre 1, 2 et 3) donne les spécifications du module d'observation, selon le schéma suivant :

Le premier chapitre définit le cadre du travail et précise la notion d'observation et les différents modes associés ainsi que le mécanisme de prise de décision.

Le deuxième chapitre fournit quelques outils et méthodes de conception de systèmes de contrôle/commande numériques.

Enfin, le troisième chapitre développe et explique l'utilisation de la méthode conçue par CALVEZ en prenant pour application la conception d'un observateur de signaux physiologiques.

La deuxième partie (chap. 4 et 5) donne, quant à elle, les mécanismes associés au dispositif de prise de décision.

Le chapitre 4 aborde l'organisation générale d'un système expert ainsi que les critères permettant de l'insérer dans une chaîne automatique d'observation et de prise de décision. Finalement, le dernier chapitre décrit un système conçu autour d'un moteur d'inférence d'ordre 0. Les performances temps réel du système sont ensuite analysées dans le cadre d'un couplage avec le générateur de faits constitué de l'observateur défini dans la première partie.

MOTS CLES : OBSERVATION - PRISE DE DECISION -- SYSTEME EXPERT TEMPS REEL