

50376
1987
375

50376
1987
375

N° d'ordre : 14

Année 1987

THESE

présentée à

L'INSTITUT INDUSTRIEL DU NORD

en vue de l'obtention du diplôme de

DOCTEUR INGENIEUR

par

PASCALE AUGER

UNE METHODOLOGIE INTEGREE D'AIDE A LA CONCEPTION
DES SYSTEMES DE FABRICATION

Soutenue le 9 Juin 1987 devant la Commission d'Examen

Membres du Jury :

M. BORNE
M. GENTINA
F. FOISSEAU
C. MELIN
M. VOJNOVIC
M. RANDET

Président
Rapporteur
Examinateur
Examinateur
Examinateur
Invité

AVANT-PROPOS

La présente thèse s'est déroulée dans le cadre du projet européen: ESPRIT 623 : "OPERATIONAL CONTROL FOR ROBOT SYSTEM INTEGRATION INTO CIM" dont fait partie RENAULT AUTOMATION.

Je remercie M. RANDET pour m'avoir accueillie à la Direction Technique des Automatismes Avancés de RENAULT AUTOMATION.

Je présente mes plus vifs remerciements à M. VOJNOVIC, responsable du service Etudes et Recherches en Productique, et, chef du projet ESPRIT 623.

Tout au long du développement de ce thème de recherche, il m'a accordé sa confiance et guidé grâce à son expérience et sa vaste culture scientifique.

Je remercie M. GENTINA, Professeur à l'Institut industriel Du Nord, pour avoir assuré le suivi de ce travail de recherche. Ses critiques et conseils ont beaucoup contribué à l'amélioration du présent mémoire.

Je remercie également M. BORNE, Professeur à l'Institut industriel Du Nord, pour avoir accepté de présider ce jury.

Je remercie M. FOISSEAU, ingénieur de recherche au CERT-DERI, pour ses conseils judicieux, concernant la conduite de cette étude.

Je remercie M. MELIN, Professeur à l'Université de Compiègne pour l'honneur qu'il me fait en participant à mon Jury de Thèse.

J'exprime toute ma reconnaissance à MM. DUPUIS, BONTEMPS et QUANTIN de la SERI pour avoir eu la patience de répondre à mes questions et m'avoir fait partager leur compétence quant à la conception de systèmes de fabrication.

J'exprime toute ma gratitude à M. ROGALA, Directeur Technique d'ACMA-ROBOTIQUE, pour l'intérêt qu'il a manifesté à l'égard de ce travail en nous fournissant un avant-projet industriel comme base d'exemple.

Mes remerciements vont également à MM. BONNIER et EHRET de la DPU pour leurs encouragements et leur collaboration quand il s'est agi d'utiliser une telle méthodologie sur des cas industriels réels.

SOMMAIRE

	Page
INTRODUCTION	1
<u>Chapitre 1 - MODELISATION DES SYSTEMES DE FABRICATION</u>	6
<u>1.1. Introduction</u>	7
1.1.1. Présentation de l'exemple d'illustration	7
1.1.2. Utilisation de l'exemple pour illustrer la méthodologie de conception proposée	9
<u>1.2. Conception d'un système de fabrication</u>	10
<u>1.3. Méthodologie de conception des systèmes de fabrication</u>	11
1.3.1. Définition des besoins	11
1.3.2. Evaluation d'une méthode de conception	12
<u>1.4. Quelques méthodes de conception existantes</u>	13
1.4.1. HIPO	14
1.4.2. SADT	15
1.4.3. PERSPECTIVE	17
1.4.4. SARS	19
1.4.5. EPOS	24
1.4.6. Conclusions	28
<u>1.5. Plan de présentation du mémoire</u>	29

<u>Chapitre 2 - UN MODELE CONCEPTUEL POUR LA PRE-ETUDE DES SYSTEMES DE FABRICATION</u>	30
<u>2.1. Introduction</u>	31
<u>2.2. Principales étapes d'un avant-projet</u>	32
<u>2.3. Modèle de cycle de vie</u>	33
2.3.1. Objectifs	33
2.3.2. Modèle de Mac DERMID	33
2.3.3. Structure du modèle de Mac DERMID	35
2.3.4. Modélisation d'une étape d'avant-projet	38
<u>2.4. Modèle de représentation des besoins</u>	40
2.4.1. Objectifs	40
2.4.2. Syntaxe générale retenue	41
2.4.3. Syntaxe détaillée de la spécification des besoins	44
2.4.4. Avantages et inconvénients	49
<u>2.5. Modèle de spécification du système</u>	50
2.5.1. Objectifs	50
2.5.2. Syntaxe générale retenue	51
2.5.3. Spécification d'une fonction générale	53
2.5.4. Spécification d'une action élémentaire	56
2.5.5. Spécification d'un équipement	62
2.5.6. Spécifications graphiques	65
2.5.7. Avantages et inconvénients	66
<u>2.6. Conclusion</u>	68

	Page
Chapitre 3 - METHODOLOGIE D'AIDE A LA CONCEPTION DES SYSTEMES DE FABRICATION	69
<u>3.1. Objectifs</u>	70
<u>3.2. Etapes de la méthodologie</u>	71
<u>3.3. Présentation générale des étapes d'un avant-projet</u>	72
3.2.1. Analyse du Système de Fabrication	72
3.2.2. Definition des Solutions Techniques	74
3.2.3. Evaluation Economique	75
3.2.4. Evaluation de Faisabilité Technique	75
<u>3.4. Démarche du concepteur</u>	76
<u>3.5. Analyse du Système de Fabrication</u>	78
3.5.1. Ecriture du cahier des charges client	78
3.5.2. Illustration du GUIDE CONCEPTEUR	79
3.5.3. Spécification des besoins du concepteur	83
3.5.4. Illustration du GUIDE CONCEPTEUR	85
3.5.5. Définition de l'environnement lié à l'application	86
<u>3.6. Définition des Solutions Techniques</u>	87
3.6.1. Analyse Fonctionnelle Générale	87
3.6.2. Choix des solutions techniques	90
3.6.3. Spécifications fonctionnelles	92
3.6.4. Spécifications opérationnelles	94
3.6.5. Vérifications des contraintes de réalisation	95
3.6.6. Spécifications détaillées	99
3.6.7. Vérifications des spécifications	100
<u>3.7. Conclusion</u>	101

<u>Chapitre 4 - OUTIL D'AIDE A LA CONCEPTION DE SYSTEMES DE FABRICATION</u>	104
<u>4.1. Introduction</u>	105
<u>4.2. Fonctionnalités du système</u>	107
4.2.1. Dialogue homme-machine	108
4.2.2. Traitement d'un projet	110
4.2.3. Traitement d'une version	111
4.2.4. Traitement d'une spécification	112
4.2.5. Choix du dialogue opérateur	114
4.2.6. Gestion de Bases de Données	115
<u>4.3. Architecture fonctionnelle</u>	116
4.3.1. Traitement Dynamique de l'Information (TDI)	117
4.3.2. Interface Opérateur	117
4.3.3. Module de traitement des spécifications	117
4.3.4. Bases de Données externes	117
<u>4.4. Environnement de développement et d'exploitation</u>	120
4.4.1. Architecture matérielle	120
4.4.2. Environnement de développement	121
4.4.3. Environnement d'exploitation	122
4.4.4. Conclusions	123
<u>4.5. Analyse organique</u>	124
4.5.1. Réalisation du module TDI	124
4.5.2. Réalisation du module Interface Opérateur	127
<u>4.6. Conclusion</u>	129

	Page
<u>Chapitre 5 - CONCLUSION GENERALE ET PERSPECTIVES DE DEVELOPPEMENT</u>	130
<u>Principaux résultats</u>	132
<u>De l'ingénierie des systèmes de fabrication vers leur programmation</u>	134
<u>BIBLIOGRAPHIE</u>	135
<u>ANNEXES</u>	140
ANNEXE 1 - Glossaire	141
ANNEXE 2 - Analyse fonctionnelle détaillée de l'outil réalisé	145
ANNEXE 3 - Présentation de l'Analyse Organique selon METHRA	149
ANNEXE 4 - Présentation du module de programmation hors-ligne	153

INTRODUCTION

La recherche systématique d'une plus grande productivité et flexibilité, dans le domaine de l'Industrie Manufacturière, a contribué à une évolution vers une production automatisée.

Or, il n'existe pas, à l'heure actuelle, de moyens de conception efficaces pour estimer précisément les risques techniques et économiques encourus lors de la réalisation d'un système flexible de fabrication.

Une conséquence directe est l'augmentation excessive des coûts de développement des systèmes de fabrication.

Il devient donc impératif d'avoir une démarche plus progressive dans l'approche des solutions. Une pré-étude des systèmes de fabrication, complète et structurée, doit être mise en place.

Celle-ci permettra de définir, avant toute réalisation, les fonctions et les équipements concernés, l'organisation et la gestion de l'atelier, l'évaluation des coûts et des performances.

Dès lors, les outils classiques de l'ingénierie ne suffisent plus. De nouveaux modèles, des méthodes et des outils de conception, adaptés au domaine de la production automatisée, doivent être proposés.

C'est l'objet de ce mémoire.

Le premier objectif de notre travail de recherche a donc été de définir une méthodologie d'aide à la conception des systèmes de fabrication.

Le deuxième objectif de notre travail a été la mise au point d'un outil d'aide à la conception des systèmes de fabrication, reprenant les concepts de la méthodologie développée.

* *

*

L'étude de l'état de l'Art nous a permis d'identifier des méthodes, des langages, et, des outils pour la conception et le développement de systèmes informatiques.

[DEMETROVICS-87, KOCH-82, YEH-80, HOWDEN-81, KRONENTAL-77, HAMMER-79, LUDEWIG-78, TEICHROEW-77/1/2, PARNAS-77].

En raison de l'analogie qui peut être établie entre une information, et, un objet physique, une transposition de ces méthodes et outils pour la conception de systèmes de fabrication a pu être envisagée.

Nous avons donc évalué l'adéquation des méthodes de conception, mises en oeuvre aujourd'hui, qui nous semblaient les plus significatives (HIPO, SADT, PERSPECTIVE, ...).

Celles-ci répondent à des besoins des concepteurs sans toutefois proposer une approche adaptée à une production automatisée [KAMPEN-81].

En effet, chacune de ces méthodes recommande une technique d'analyse et un formalisme d'écriture qui recouvre une et une seule étape de conception. SADT, par exemple, fournit un support de conception pour l'analyse fonctionnelle d'un système.

Dès lors, de part l'hétérogénéité des processus de réflexion et de démarche, il devient difficile de valider l'ensemble des résultats issus de chaque étape de conception.

L'utilisation de telles méthodes, intéressante pour l'étude d'un point particulier se révèle pénalisante lors de la définition complète et structurée d'un système de fabrication.

Des études ont été menées dans le domaine de la production automatisée. Des méthodes ponctuelles ont été définies pour la mise en place d'applications robotiques [KOREN-85], le choix d'un robot [MORGAN-84] ou encore son installation [BERGANT-82, BEHUNIAK-81].

Des recherches plus générales ont porté sur la conception de systèmes de fabrication [BERNUS-79] ou la définition des besoins logiciels pour un système complexe [HENINGER-80, CALVEZ-78].

[FISHER-83] a mis au point un guide pour l'écriture des besoins et l'analyse d'un système.

Aucun des travaux, réalisés jusqu'ici, ne propose une méthodologie qui fournisse une approche intégrée et systématique du processus de conception des systèmes de fabrication.

Dans le cadre de ce travail de recherche, nous nous sommes donc efforcés de répondre aux besoins fondamentaux de l'ingénierie des systèmes de fabrication qui sont :

- 1 - la définition d'un modèle de conception,
- 2 - la mise en place d'une démarche de conception structurée,
- 3 - des outils d'ingénierie pour l'écriture des spécifications, l'analyse syntaxique et la documentation des spécifications.

Un modèle de conception des systèmes de fabrication, a été proposé sur la base des travaux de [Mac DERMID-84, LEHMAN-80] concernant la modélisation de la conception des systèmes informatiques. C'est l'objet du chapitre 2.

La définition d'une démarche de conception structurée a justifié la mise en place d'un GUIDE CONCEPTEUR [AUGER-86, CALVEZ-82]. Celui-ci a été défini dans le chapitre 3.

Enfin, la définition d'un outil intégré d'aide à la conception de systèmes de fabrication, dans le chapitre 4, s'est inspirée des concepts présentés par [HARLOW-86] pour le développement d'une station de travail en ingénierie.

* *

*

Notre étude s'inscrit dans un programme de recherche européen plus vaste. Il s'agit du projet ESPRIT 623 intitulé :

"Operational Control for Robot System Integration into Computer Integrated Manufacturing"

Ce projet a été défini sur la base d'un projet pilote "Design Rules for the Integration of Industrial Robots into CIM systems".

Au cours de cette étude, l'analyse des besoins des utilisateurs et l'état de l'Art en production automatisée, ont permis d'identifier la nécessité de disposer des systèmes suivants :

- système d'ingénierie pour la conception de systèmes de fabrication,
- système de conception de produits en vue d'une fabrication automatisée,
- système de simulation et de programmation de robots basé sur des techniques CAO.

La finalité du projet ESPRIT 623 est de répondre à ces besoins en développant deux outils : un outil d'aide à la conception et un outil de programmation hors-ligne de robots.

L'outil de programmation hors-ligne a pour but de valider, avant toute réalisation, les solutions techniques proposées. Pour ce faire, il construit une représentation virtuelle de ces solutions.

Les partenaires de RENAULT AUTOMATION, dans ce projet de recherche, sont issus du monde universitaire et industriel : KUKA (Augsbourg), U. de KARLSRUHE, U. de BERLIN, U. de GALWAY (Irlande), FIAR (Milan) et les Universités de MADRID, de LISBONNE et d'AMSTERDAM.

MODELISATION DES SYSTEMES DE FABRICATION

(chapitre 1)

1.1. Introduction

Compte-tenu de l'intérêt pratique du travail de recherche développé, les résultats présentés dans ce mémoire illustrent l'étude d'un exemple industriel.

Celui-ci a été naturellement choisi au sein de RENAULT AUTOMATION et dans un souci de comparaison, déjà résolu par les méthodes classiques de conception.

Il s'agit de l'automatisation du procédé de conditionnement de 27 types de pare-brise de voitures de formes différentes. Il est décrit dans le paragraphe 1.1.1.

Nous nous intéresserons au processus général de conception d'un tel système de fabrication afin d'en dégager les points critiques (paragraphe 1.2.).

Nous définirons ensuite, les besoins pour une méthodologie complète de conception des systèmes de fabrication (paragraphe 1.3.).

Dans le paragraphe 1.4., nous présenterons quelques méthodes de conception des systèmes informatiques.
L'une d'entre elles est à l'origine du développement de notre méthodologie.

Enfin, le plan de présentation du mémoire est détaillé dans le paragraphe 1.5.

1.1.1. Présentation de l'exemple d'illustration

Il s'agit de l'automatisation du procédé de conditionnement de 27 types de pare-brise de voitures de formes différentes.

L'opération de conditionnement consiste en :

- l'alimentation du pare-brise,
- la pose d'un joint mousse,
- l'emballage du pare-brise,
- l'évacuation du pare-brise.

La machine d'emballage (désignation technique : faredeluse) est imposée.

La solution technique retenue comprend quatre plateaux tournants (2 en entrée, 2 en sortie) ayant chacun deux containers. Des opérateurs déposent les pare-brise dans les containers.

Deux robots en entrée prennent chacun un pare-brise au moyen d'une ventouse pour les préconditionner.

Cette opération consiste à poser un profilé en mousse agglomérée sur les pare-brise au moyen d'un dispositif technique (poste de recentrage et "ligne d'amenage de joint-mousse").

Les pare-brise sont déposés ensuite sur un tapis roulant pour être acheminé vers la machine d'emballage. Le poste de dépôt étant commun aux deux robots, un interverrouillage a été prévu de façon à ce que les deux robots n'entrent pas en collision. (Une came est fixée sur le socle de chaque robot).

La faredeleuse enveloppe le pare-brise d'un film plastique protecteur, qui a été chauffé auparavant. En se refroidissant, il se rétracte et se tend sur le pare-brise.

Un troisième robot en fin de chaîne dépose les pare-brise emballés, dans les containers de sortie, pour qu'ils soient évacués.

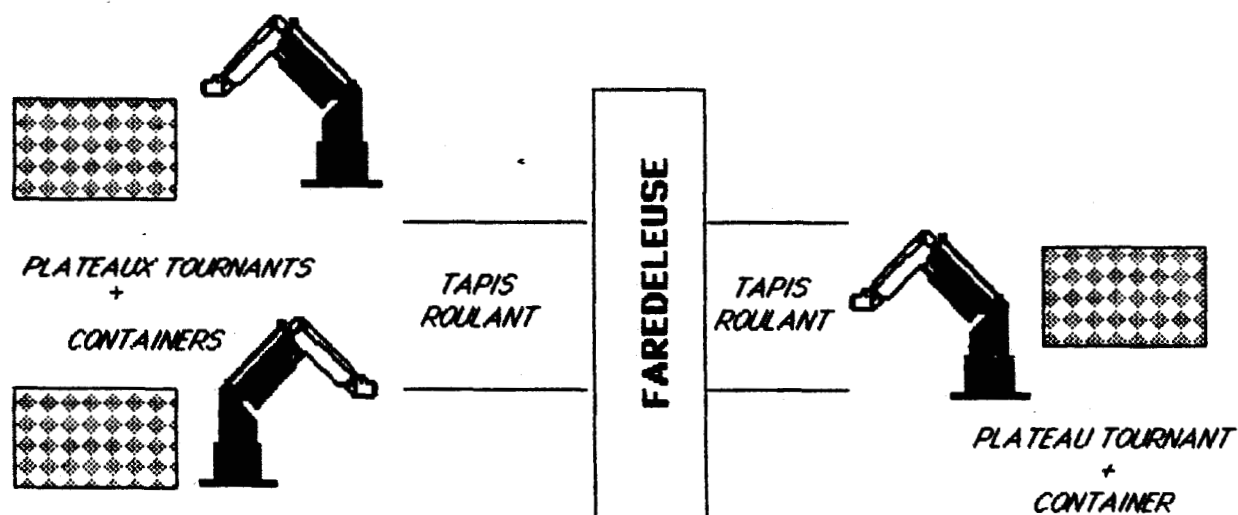


Fig 1 Schéma d'implantation du système de conditionnement

1.1.2. Utilisation de l'exemple pour illustrer la méthodologie de conception proposée

Dans un premier temps, nous avons évalué sur la base de cet exemple l'intérêt des méthodes de conception existantes et leur adéquation au domaine de la Production Automatisée.

Dans un deuxième temps, cet exemple illustrera les points forts et les concepts de la méthodologie que nous avons développée.

1.2. Conception d'un système de fabrication

Le processus de conception d'un système de fabrication a été identifié comme suit :

- une phase d'Avant-Projet ou de Pré-Etude,
- l'étude détaillée de la solution technique retenue,
- la réalisation de cette solution.

Dans le cadre d'un avant-projet, le concepteur propose un système de fabrication clef en main avec des garanties au niveau des coûts et des performances techniques.

L'objectif est d'aller aussi loin que possible dans la phase de définition virtuelle d'un système, et, d'offrir le plus de garanties possibles.

Par conséquent, cette étude aborde, en détail, la spécification des besoins du client et du constructeur.

Sur la base de ceux-ci, le concepteur propose plusieurs architectures possibles pour le système de fabrication à réaliser.

Ces différentes solutions sont spécifiées d'un point de vue fonctionnel (description des tâches des équipements) et opérationnel (description des équipements et de leurs performances techniques).

Les solutions décrites sont alors évaluées sur des critères économiques et techniques.

Toutes études ou solutions trop coûteuses à mettre en oeuvre doivent être écartées à ce niveau de la conception.

La documentation produite en fin d'avant-projet constitue une référence contractuelle pour l'étude détaillée de la solution et sa réalisation.

Par conséquent, l'évolution et le succès de la mise au point d'un système de fabrication dépend de la qualité de la pré-étude réalisée.

Or, une très forte concurrence industrielle contribue à faire baisser les prix et les délais pour soumettre une proposition au client.

Dès lors, la phase d'avant-projet se révèle être une phase de conception critique, pour laquelle une méthodologie et un outil intégré de conception doivent être définis.

1.3. Une méthodologie de conception des systèmes de fabrication

1.3.1. Définition des besoins

Lors de la pré-étude d'un système de fabrication, le concepteur rencontre des problèmes techniques, économiques ou de suivi de projet.

Une méthodologie de conception efficace doit faciliter sa tâche en l'aidant à formaliser son approche, lors de :

- (i) l'étude exhaustive des fonctionnalités du système et de leur implémentation,
- (ii) la définition précise des options ou dérogations par rapport aux exigences du client qui sont souvent matière à litige lors de l'échéance,
- (iii) le contrôle des risques techniques liés au projet,
- (iv) les premières évaluations du comportement dynamique des solutions proposées, qui doivent être menées en parallèle des phases de spécification,
- (v) la garantie de l'adaptabilité de la solution lors des évolutions futures de l'application,
- (vi) la réduction de l'investissement en temps et en argent avant la signature du contrat, sans nuire à la qualité de l'étude,
- (vii) et, enfin, la mise en place d'un suivi de projet efficace.

Une première solution est l'utilisation d'un langage de spécification, indispensable pour la description formelle et structurée d'un système de fabrication.

Une deuxième solution est la mise en place d'un système d'outils, associé à la méthodologie de conception développée.

Ce système d'outils devra offrir une ouverture vers les logiciels d'ingénierie couramment utilisés par les bureaux des méthodes (logiciels de simulation, ...).

Dès lors, la méthodologie de conception développée, en ayant une dimension productive réelle, pourra devenir l'outil journalier des concepteurs de systèmes de fabrication.

1.3.2 Evaluation d'une méthode de conception

Une méthode de conception sera évaluée, dans le cadre de ce mémoire, en fonction des solutions qu'elle propose pour satisfaire les contraintes spécifiques d'un avant-projet.

En outre, elle devra répondre aux besoins traditionnels des concepteurs, qui sont les suivants :

- (i) disposer d'un moyen de description rigoureux avec plusieurs niveaux d'abstraction (langage de spécification)
- (ii) différencier les étapes d'analyse (quoi faire ?) et de résolution (comment faire ?)
- (iii) garantir la complétude des informations décrites,
- (iv) éviter toute ambiguïté,
- (v) détecter facilement et le plus tôt possible les erreurs pour éviter un effet multiplicateur,
- (vi) pouvoir réutiliser facilement des spécifications antérieures.

Tout langage de spécification doit offrir des facilités d'apprentissage, de mise en oeuvre et de mise à jour. Il doit permettre d'intégrer des commentaires, des formalismes textuels ou graphiques.

Cette lisibilité ne doit pas se faire au détriment du degré de formalisme qui garantit d'une part la mise en oeuvre de procédures de vérification automatiques, et, d'autre part, une certaine rigueur dans la formalisation des spécifications.

Un bon compromis doit donc être établi afin d'obtenir une documentation lisible qui soit une référence contractuelle.

La méthode, pour l'écriture des spécifications, doit assurer l'enchaînement des spécifications, une description modulaire et hiérarchisée des problèmes étudiés, ainsi que l'étude de sous-systèmes coopérants.

Enfin, la pré-étude d'un système de fabrication pose le problème de l'intégration de composants hétérogènes. Une bonne méthode de conception doit proposer au concepteur une démarche cohérente et structurée, pour aborder cette intégration.

Les caractéristiques techniques et les performances (cadence, vitesse, ...) des équipements d'un système de fabrication doivent pouvoir être également formalisées.

1.4. Quelques méthodes de conception existantes

Nous avons étudié, d'après les critères ci-dessus, les méthodes de conception existantes qui pourraient être transposées au domaine de l'ingénierie des systèmes de fabrication.

Après un tour d'horizon effectué en Juin 85, nous avons retenu les méthodes de conception suivantes : HIPO, SADT, PERSPECTIVE, SARS et EPOS.

Ces méthodes de conception ont été évaluées sur la base de l'exemple industriel présenté auparavant.

1.4.1. HIPO

HIPO (Hierarchy plus Input Process Output) est une méthode de documentation hiérarchisée fondée sur l'utilisation de diagrammes de structure hiérarchique.

Ceux-ci sont utilisés à trois niveaux de description :

- niveau Système,
- niveau Programme,
- niveau Module.

L'utilisation de HIPO pour la formalisation des fonctions de l'application de conditionnement conduit au diagramme de structure suivant :

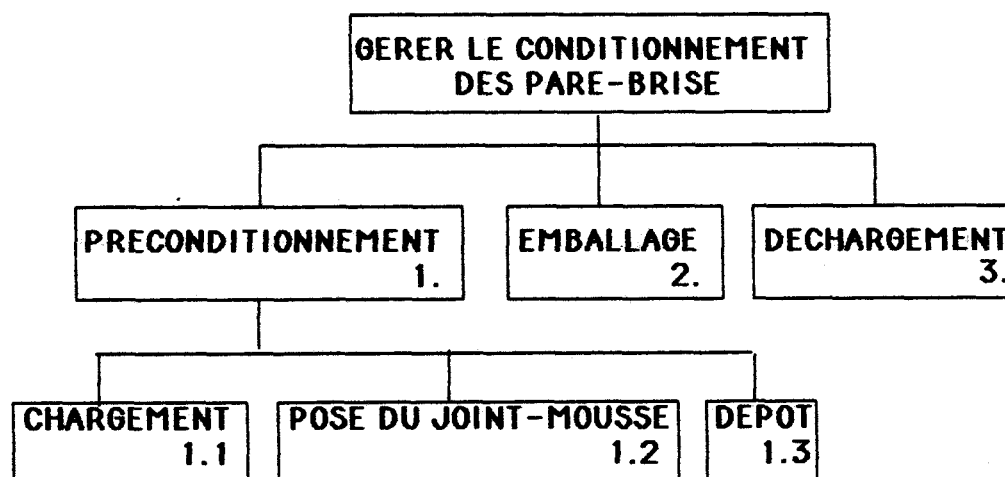


Fig 2 Diagramme de structure de niveau système

Cette méthode ne propose pas de formalisme de spécification, mais simplement une technique de documentation.

Or, l'utilisation d'un langage naturel est en général source d'ambiguïté et d'erreur. En outre, toute procédure de vérification (autre qu'un cycle lecteur-auteur) est difficile à mettre en oeuvre.

La méthode HIPO ne propose pas de règles pour la décomposition successives du système en programmes et modules.

Enfin, son utilisation dans le domaine de la Production Automatisée n'apporte pas de solution pour la description des équipements d'un système de fabrication.

1.4.2. SADT

SADT (Structured Analysis and Design Technique) est une méthode dédiée à l'analyse fonctionnelle d'un système [IGL-76].

Pendant l'analyse, l'accent est mis sur ce que le système est supposé faire à partir d'une description la plus générale possible. La méthode d'approche proposée par SADT est descendante, modulaire, hiérarchique et structurée.

De ce fait, SADT propose un formalisme graphique basé sur deux types de diagrammes. Il s'agit de diagrammes décrivant le système d'une part en termes d'activités (actigrammes) et d'autre part en termes de données (datagrammes).

La description de la fonction de préconditionnement selon le formalisme SADT conduit au diagramme suivant :

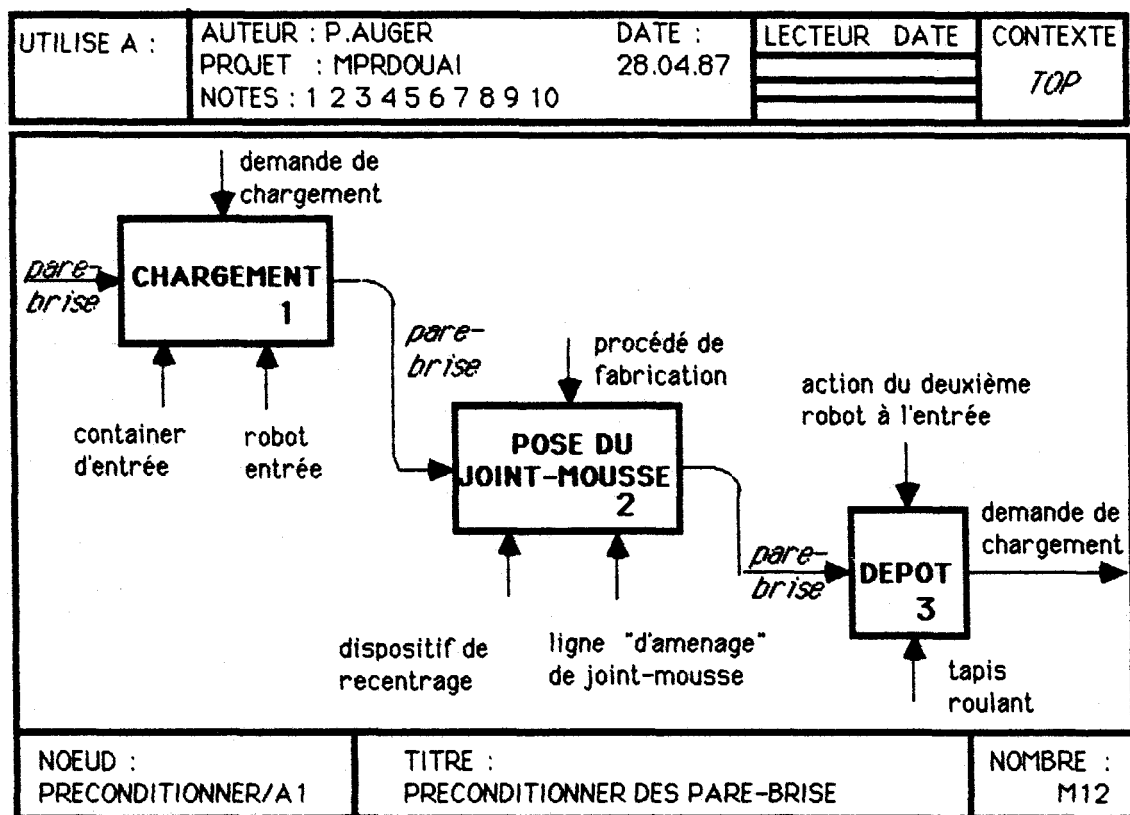


Fig 3 Actigramme de la fonction de Préconditionnement

Une vérification des spécifications graphiques est prévue sous la forme d'un cycle de lecteur-auteur. Toutes les décisions et commentaires sont rédigés et classifiés sur la base d'un schéma d'organisation fourni par SADT.

La pratique d'une telle méthode, si elle s'avère positive sur certains points (rigueur dans la démarche de conception, ...) présente néanmoins des inconvénients majeurs pour une utilisation dans un contexte productique.

Tout d'abord, SADT est une méthode ponctuelle. Elle n'est pas adaptée à la description des équipements des systèmes de fabrication, et, de leurs performances.

La formalisation actigramme-datagramme ne permet pas de rendre compte des besoins du client pour l'automatisation de son application.

Des contraintes de temps de cycle, d'environnement ou d'ordre financier s'expriment difficilement sous la forme de diagrammes.

L'existence de deux modèles graphiques (actigrammes-datagrammes) souligne un manque d'intégrité au niveau de la démarche de conception. De plus, elle nécessite une vérification croisée, difficile à mettre en oeuvre.

Enfin, le cycle lecteur-auteur n'est pas toujours respecté lors de délais de réalisation très courts.

* *

*

D'autres méthodes, dont la portée est plus grande, sont soit disponibles industriellement, soit au niveau de prototypes de recherche.

En réponse aux besoins des concepteurs de programmes, elles proposent une démarche intégrée, rigoureuse, ainsi qu'un support de description des problèmes logiciels pour le cycle complet de la conception.

Dans le cadre de ce travail d'évaluation, nous nous sommes tout particulièrement intéressés aux méthodes PERSPECTIVE, SARS et EPOS.

1.4.3. PERSPECTIVE

PERSPECTIVE est un système d'aide au développement et à la réalisation de logiciels temps-réel, proposant aux concepteurs un environnement complet (méthode, langage et outils).
[SYSTEMS DESIGNERS LIMITED-85].

La méthode PERSPECTIVE recommande une démarche de conception modulaire et hiérarchisée basée sur l'analyse des besoins en communication (flux de données).
Le formalisme utilisé est graphique.

Une application est modélisée selon une structure de sous-systèmes, processus et modules communiquant entre eux par des interfaces.
Cette décomposition est structurée dans le but d'identifier les aspects temps-réel (coordination, synchronisation) du logiciel à réaliser.

Les outils informatiques, associés à la méthode PERSPECTIVE, fournissent un support à la construction, la vérification et la mise en oeuvre opérationnelle du logiciel développé.
Il s'agit d'un éditeur, d'un module d'analyse syntaxique des spécifications, et, d'un outil de gestion de projet.

La description du système de fabrication pour le conditionnement d'un pare-brise, selon le formalisme PERSPECTIVE, conduit au diagramme suivant :

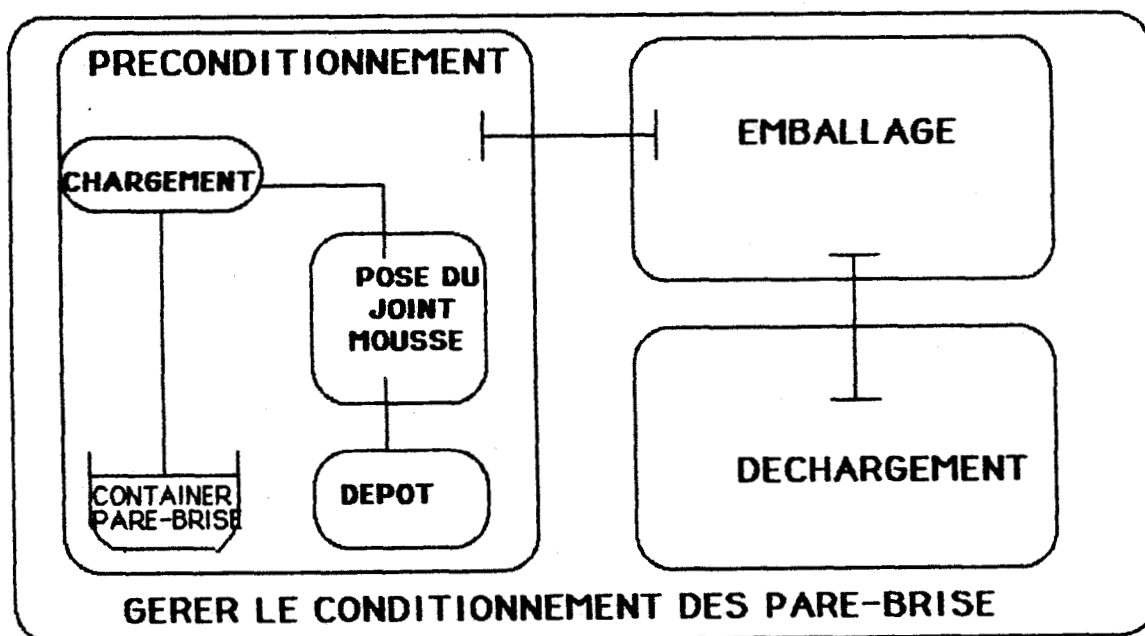


Fig 4 Système de conditionnement des pare-brise

Ce formalisme graphique est intéressant, dans les phases d'analyse et de spécification fonctionnelle.

En effet, il permet de visualiser différentes fonctions à un même niveau de décomposition, ainsi que le flux de données entre ces fonctions.

Par contre, il n'est pas adapté à la spécification des besoins du client et du constructeur pour l'automatisation d'une application. En effet, comment expliciter des contraintes économiques ou techniques ("la température de l'atelier est de soixante degrés Celsius", par exemple) sous la forme de tels diagrammes ?

Les équipements du système de fabrication, leurs performances, sont également difficiles à décrire en terme de tâche ou de module.

Enfin, le support informatique proposé par la méthode PERSPECTIVE n'offre pas l'ouverture souhaitée pour une utilisation productive.

L'outil PERSPECTIVE fait appel à un langage PASCAL spécifique, et, donc à un compilateur spécifique.

Une extension des concepts de cette méthode et des outils associés pour la conception de systèmes de fabrication est difficilement envisageable.

1.4.4. Système SARS

Le système SARS est un système d'aide au développement de logiciels, réunissant une méthode de spécification MARS, un langage LARS, et, des outils informatiques [EPPLE-84, EPPLE-83, KOCH-83, KOCH-82].

1.4.4.1. La méthode MARS

La méthode MARS propose une spécification fonctionnelle, modulaire et descendante basée sur un modèle de Stimulus-Réponse. Un Stimulus représente un événement causal auquel répond un effet (Réponse).

Les fonctions de Stimulus-Réponse proposent un découpage fonctionnel du problème dont les décompositions successives sont modélisées sous la forme de réseaux de Stimulus-Réponse.

1.4.4.2. Le langage LARS

Le langage LARS, associé à la méthode MARS, offre une description formelle des réseaux de Stimulus-Réponse sur la base des trois entités suivantes : NETS-LARS, INTERFACE-LARS, GUARD-LARS.

Les NETS-LARS modélisent le flux de données circulant dans le réseau de Stimulus-Réponse.

L'INTERFACE-LARS traite les données d'entrée-sortie qui définissent les interfaces entre le système et le monde extérieur.

Les GUARD-LARS décrivent les connections logiques entre les interfaces et les réseaux de Stimulus-Réponse.

1.4.4.3. Outils associés à la méthode MARS

Les outils associés à la méthode MARS sont les suivants : un système d'information, un éditeur, un module d'analyse syntaxique des spécifications, et, un système de dialogue.

Le système d'information gère toutes les données globales au moyen d'une base de données relationnelle avec laquelle on communique par une matrice d'accès.

Il existe deux types d'accès au système d'information :

- * un accès direct à travers un langage de requête
- * un guide utilisateur semi-graphique.

L'éditeur offre des modes de représentation graphiques et textuelles en entrée et en sortie.

Un compilateur LARS vérifie la complétude et la cohérence syntaxique de toutes les spécifications.

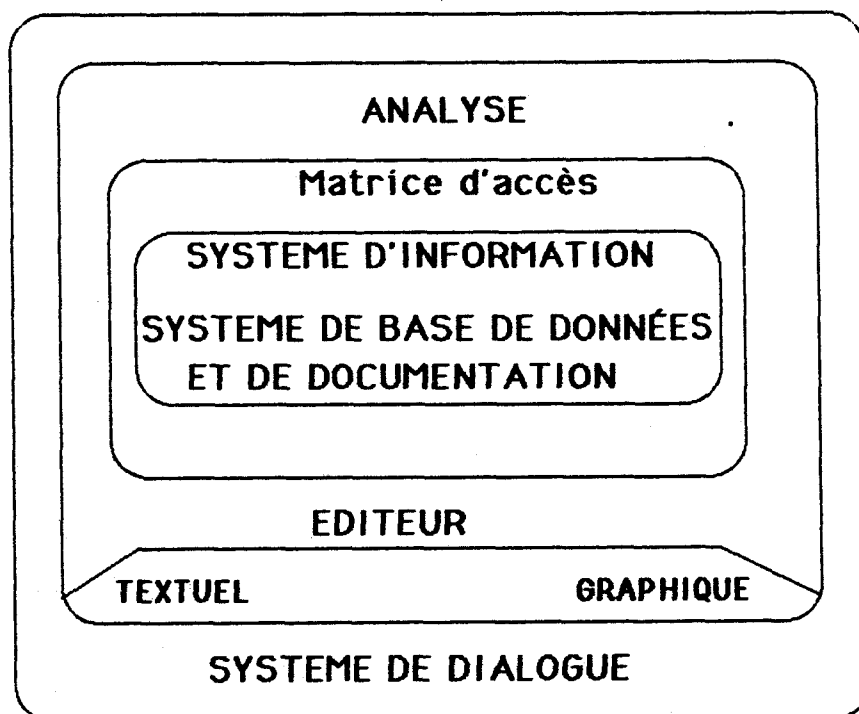
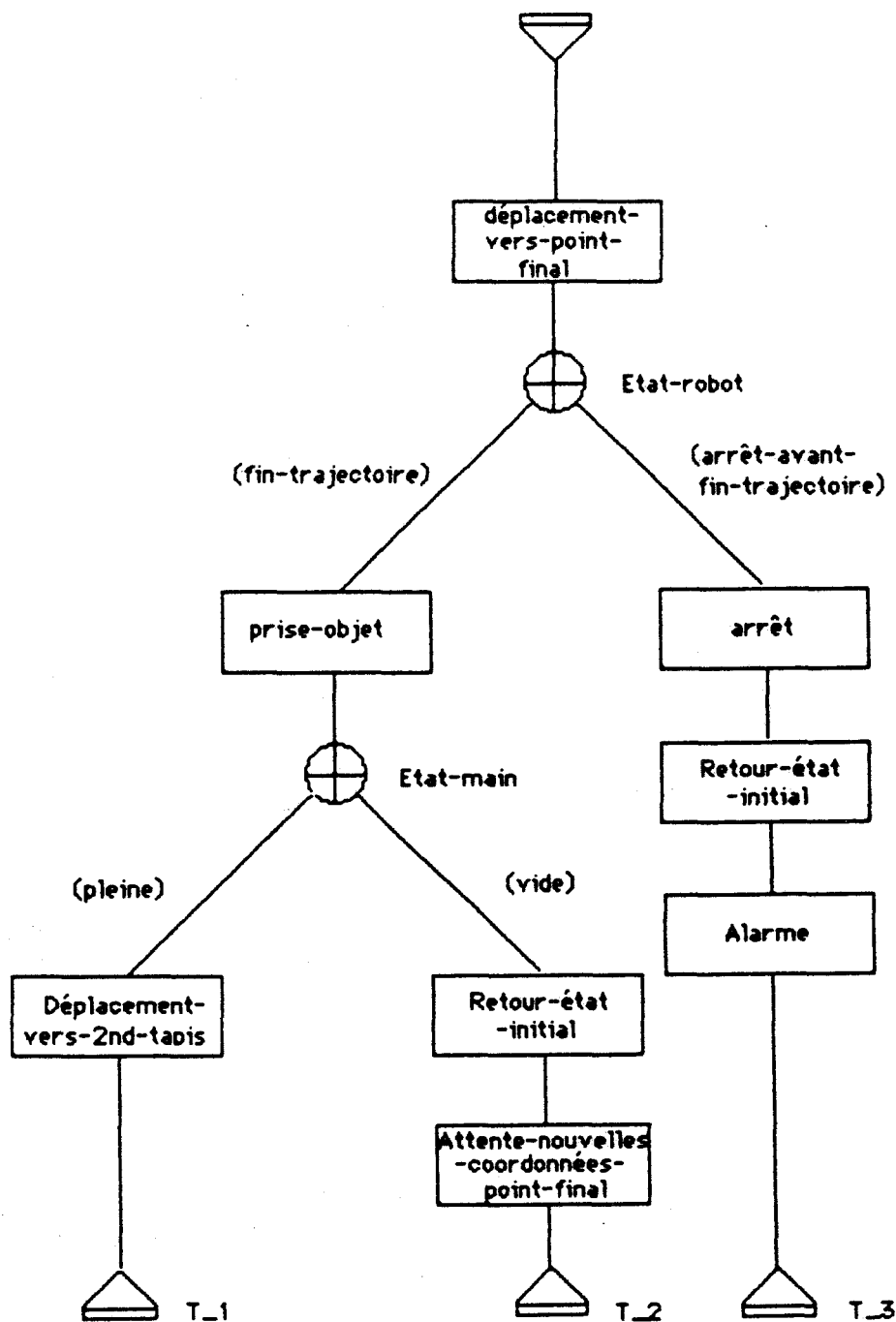


Fig 5 Outils associés à la méthode MARS

1.4.4.4. Illustration de la méthode sur la base de notre exemple

Nous décrirons ici le module logiciel qui assure la saisie d'un pare-brise dans un container, le transport et le dépôt du pare-brise par un robot sur un tapis roulant.

La spécification graphique proposée par la méthode MARS conduit au réseau de Stimulus-Réponse suivant :



La spécification textuelle associée au réseau de Stimulus-Réponse présenté est la suivante :

srm-det DEPLACEMENT-ROBOT

```
structure
alpha DEPLACEMENT_VERS_POINT_FINAL ;
considérer ETAT_ROBOT

    if (FIN_TRAJECTOIRE)
    alpha PRISE_OBJET ;
    considérer ETAT_MAIN

        if (MAIN_PLEINE)
        alpha DEPLACEMENT_VERS_2nd_TAPIS
        terminate T1 ;

        or (MAIN_VIDE)
        alpha RETOUR_ETAT_INITIAL ;
        alpha ATTENTE_NOUVELLE_COORDONNEES
        POINT_FINAL ;
        terminate T2 ;
    end considérer ETAT_MAIN ;

    or (ARRET_AVANT_FIN_TRAJECTOIRE)
    alpha ARRET
    alpha RETOUR_ETAT_INITIAL ;
    alpha ALARME ;
    terminate T3 ;

end considérer ETAT_ROBOT ;
end structure ;
```

end srm-det DEPLACEMENT-ROBOT ;

Fig 7 Spécification textuelle du module de saisie

1.4.4.5. Evaluation du système et conclusions

Le système SARS garantit une démarche rigoureuse du fait de l'utilisation de la méthode MARS ainsi qu'une spécification cohérente et complète des systèmes à processus parallèles.

L'existence d'un traitement graphique des entrées facilite la formalisation des informations.

Enfin, la possibilité d'intégrer des commentaires, et, un traitement automatisé des spécifications font de SARS un outil parfaitement adapté à la modélisation des systèmes temps-réel.

Néanmoins, l'impact du système SARS quant à la spécification d'un système de fabrication reste très ponctuel.

En effet, le langage LARS ainsi que la méthode associée ne prennent pas en compte la spécification opérationnelle du système (c'est à dire la description des équipements et de leurs performances).

Le formalisme textuel et graphique utilisé par LARS n'est pas forcément accessible à tous les lecteurs.

Enfin, le support matériel de SARS (machine SIEMENS) n'est pas courant dans les milieux de l'ingénierie.

1.4.5. Le système EPOS

Le système EPOS offre une méthode de conception et des outils informatiques pour le développement, la validation, la documentation et la gestion de projets informatiques [GPP-85].

1.4.5.1. La méthode EPOS

La méthode EPOS conduit à la réalisation d'un système informatique selon trois étapes :

- la définition des besoins du client et du concepteur,
- la spécification de la solution retenue,
- la mise au point d'un support de programmation sur la base des spécifications rédigées.

Elle recommande également une approche pour la conduite et la gestion des projets.

Cette méthode s'appuie sur trois formalisme de spécification : EPOS-R, EPOS-S et EPOS-P.

1.4.5.2. Les formalismes de spécification EPOS

* LANGUAGE EPOS-Requirement : EPOS-R

EPOS-R formalise les besoins du client et du concepteur selon un langage proche du langage naturel.

Des tables de décision et des mots-clefs permettent de : structurer la description des besoins, d'introduire des procédures de vérification automatiques, et, des références formelles lors de la spécification du système.

* LANGUAGE EPOS-Specification : EPOS-S

EPOS-S est un langage formel de spécification de système, intégrant des commentaires.

Ce langage permet la description des fonctions et des objets physiques d'un système informatique, à un niveau quelconque de décomposition.

Il est basé sur sept entités formelles : MODULE, ACTION, DEVICE, CONDITION, DATA, EVENT et INTERFACE. Chacune d'elles est décrite par un masque de spécification.

* LANGUAGE EPOS-Project : EPOS-P

EPOS-P est un langage de gestion de projet. Il permet de spécifier la structure du projet, les plannings de travail et la répartition des charges entre les membres de l'équipe.

1.4.5.3. Les outils informatiques du système EPOS

Les outils informatiques, que propose le système EPOS, sont un éditeur (en l'occurrence l'éditeur VAX), trois systèmes d'évaluation et un système de communication.

* UTILITAIRE EPOS-Management : EPOS-M

EPOS-M est un utilitaire dédié à la production de documents de gestion (diagrammes PERT), générés à partir des spécifications textuelles en EPOS-P.

* UTILITAIRE EPOS-Analyse : EPOS-A

EPOS-A est un utilitaire dédié à l'analyse syntaxique des spécifications rédigées en EPOS-R et EPOS-S.

Il prend en compte l'étude de la cohérence et de la complétude syntaxique des spécifications.

* UTILITAIRE EPOS-Documentation : EPOS-D

Tout un système de documentation à la fois textuel et graphique est mis à la disposition de l'utilisateur, par le biais de cet utilitaire.

* UTILITAIRE EPOS-Communication : EPOS-C

Ce système de communication assure un accès flexible à tous les autres composants du système EPOS. Il existe trois modes de dialogue correspondant au degré de familiarisation de l'utilisateur avec le système.

1.4.5.4. Illustration de l'utilisation du système EPOS sur la base de notre exemple industriel

Des exemples de spécification, rédigés en EPOS-R et en EPOS-S, ainsi qu'un diagramme hiérarchique, produit par le module de documentation du système EPOS, sont présentés page suivante.

Il s'agit de besoins du client et du concepteur concernant la manipulation des pare-brise. La fonction de préconditionnement et l'équipement qui la réalise sont spécifiés en EPOS-S.

Le diagramme hiérarchique présente les fonctions principales du système de conditionnement des pare-brise (PRECONDITIONNEMENT, EMBALLAGE, DECHARGEMENT).

EPOS-R**SPECIFICATION DES BESOINS CLIENT****2.1. PERFORMANCES FONCTIONNELLES****2.1.1. Critères principaux****REQUIREMENT 2111 (0)**

<impératif fonctionnel>
Le pare-brise doit être tenu
lors de la pose du
joint-mousse.

SPECIFICATION DES BESOINS DU CONCEPTEUR**1. STRATEGIE DE CONCEPTION**

REQUIREMENT 21001 (0) <manipulation des pare-brise>
Le conditionnement des pare-brise et leur manipulation
seront accomplis par un robot X58 et un système de
préhension adapté.

FULFILS REQUIREMENT 2111 (0) PARTLY

EPOS-D**SERER LE CONDITIONNEMENT
DES PARE-BRISSE****A1****PRECONDITIONNEMENT****A 1.1****EMBALLAGE****A 1.2****DECHARGEMENT****A 1.3****CHARGEMENT****A 1.1.1****POSE DU JOINT-MOUSSE****A 1.1.2****DEPOT****A 1.1.3****EPOS-S****A 5.1****ACTION PRECONDITIONNEMENT-1.****DESCRIPTION :**

PURPOSE : "...."

FULFILS : REQUIREMENT 2111 (0),
CONSTRAINT 4101 (0).

DESCRIPTIONEND.

DECOMPOSITION : SET (EVDEPOT1);
WHILE CONDITION-PARE-BRISSE
PRISE1;
POSE-JOINT-MOUSSE1;
RESET (EVDEPOT2);
WAIT FOR EVDEPOT1
THEN DEPOT1
WAITEND;
SET (EVDEPOT2)

OO.

DECOMPOSITIONEND.

TRIGGERED : SYSTEMSTART.

INPUT : AUTORIS-PRISE1.

OUTPUT : SUIVI-POSE-JOINT-MOUSSE1.

PROCESSED : ROBOT1.

ACTIONEND.**OV 1****DEVICE ROBOT1.****DESCRIPTION :**

PURPOSE : "Ce robot permet de prendre le pare-brise dans
un container et le dépose sur le tapis d'entrée
de la foreuse après la pose du joint-mousse".

PERFORMANCE : "Charge maximum de 60 kilogrammes"

FULFILS : REQUIREMENT 3200 (0),
CONSTRAINT 3201 (0).

DESCRIPTIONEND.

COMPONENTS : ARMOIRE-DE-COMMANDE-1,
BRAS-1,
EFFECTEUR-1.

INPUT : ALARME-OPERATEUR-1,
ACTION-OPERATEUR-1.

CONNECTION :
SIGNAL ALARME-OPERATEUR:
ALARME-OPERATEUR<- ARMOIRE-DE-COMMANDE

PERFORMS : PRECONDITIONNEMENT-1

DEVICEEND.

1.4.5.5. Avantages et inconvénients du système EPOS

Le système EPOS, grâce à ses différents modules logiciels recouvre toutes les étapes de spécification d'un système informatique.

C'est la seule méthode de spécification à proposer un formalisme (DEVICE) pour la description des objets physiques.

La cohérence syntaxique des spécifications est assurée par des procédures de vérification automatiques d'EPOS-A.

Du fait de la génération automatique de la documentation graphique, le système EPOS facilite la mise à jour de la documentation et sa maintenance.

Il est disponible sous de nombreux matériels (IBM PC, VAX et MicroVAX, INTEL, SIEMENS), ce qui facilite son introduction dans les bureaux des méthodes.

Néanmoins, le choix du système d'exploitation VMS nécessite une configuration matérielle assez lourde.

Par ailleurs, le formalisme EPOS demande à être adapté à la spécification des systèmes de fabrication.

L'interface utilisateur est encore peu conviviale, n'offrant pas de possibilité d'entrées graphiques à l'inverse de SARS.

Enfin, l'absence de menu en version française peut freiner son utilisation dans les milieux traditionnels de la conception de systèmes de fabrication (Bureaux des méthodes).

Certains de ces inconvénients (absence d'entrées graphiques, indisponibilité sous le système d'exploitation UNIX) sont déjà l'objet d'études et de développements pour les futures versions du système EPOS.

Les autres points ne sont pas critiques quant à la sélection finale d'un outil de spécification.

1.4.6. Conclusions

En raison de son approche intégrée, des facilités de documentation et de mise à jour offertes, du formalisme de spécification qu'il propose, le système EPOS a été finalement retenu.

Notre travail consiste à adapter le système EPOS à la spécification des systèmes de fabrication, qui présentent à la fois des aspects logiciels et matériels.

La méthodologie que nous proposons de développer dans le chapitre 3, pour la conception des systèmes de fabrication est basée sur l'utilisation du système EPOS.

1.5. Plan de présentation du mémoire

Après avoir identifié le système EPOS, comme un outil de spécification potentiel pour la définition des systèmes de fabrication, nous proposons de définir :

- UN MODELE CONCEPTUEL, afin de formaliser le processus de conception. Les extensions des langages de spécification du système EPOS, pour la définition des systèmes de fabrication, sont également présentées dans le chapitre 2.
- LA METHODOLOGIE DE CONCEPTION des systèmes de fabrication. Les règles détaillées de son application sont présentées, dans le chapitre 3, sur la base d'un exemple industriel.
- UN OUTIL D'AIDE A LA CONCEPTION, intégrant à la fois la méthodologie définie et le système EPOS, est présenté dans le chapitre 4.

En guise de conclusion, nous proposons d'étudier les prolongements possibles du travail réalisé.

UN MODELE CONCEPTUEL POUR LA PRE-ETUDE
DES SYSTEMES DE FABRICATION

(chapitre 2)

2.1. Introduction

Une méthodologie de conception, pour la pré-étude des systèmes de fabrication, doit permettre d'aboutir d'une manière efficace, à une solution répondant à toutes les contraintes de réalisation d'un projet.

Dès lors, cette méthodologie doit être fondée sur un modèle conceptuel adéquat pour guider le concepteur dans une approche formelle de la solution.

La définition d'un modèle conceptuel repose :

- (a) d'une part, sur le choix d'un modèle de cycle de vie fournissant une technique globale de description des étapes d'un avant-projet,
- (b) d'autre part, sur la définition des langages de spécification, utilisés pour la représentation des informations.

Le modèle de cycle de vie doit structurer la pré-étude d'un système de fabrication comme une suite ordonnée d'étapes constructives.

Dès lors, il est possible de décrire, de plus en plus précisément le système à réaliser, ainsi que les contraintes d'enchaînement des différentes étapes d'un avant-projet.

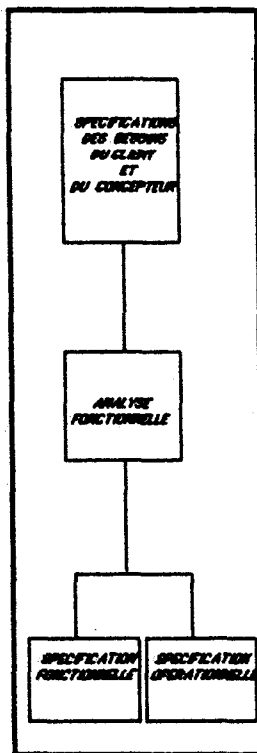
Les modèles de description retenus sont étroitement liés aux étapes mêmes de l'avant-projet, étapes que nous proposons de décrire dans le paragraphe suivant.

2.2. Principales étapes d'un avant-projet

La pré-étude d'un système de fabrication repose sur :

- l'Analyse du Système de Fabrication,
- la Définition de Solutions Techniques,
- une Evaluation Economique,
- une Etude de Faisabilité Technique.

L'entrée d'informations se fait uniquement lors des deux premières étapes, dans le cadre de :



L'Analyse du Système de Fabrication avec :

- (a) La spécification des besoins
qui prend en compte les besoins du client, ainsi que les contraintes et les limites liées à l'expérience du concepteur.

La Définition des Solutions Techniques avec :

- (b) L'analyse fonctionnelle
qui recommande une approche descendante, procédant par niveaux d'abstraction, pour identifier les fonctions générales que devra remplir le système automatisé.
- (c) La spécification fonctionnelle
qui décrit, sur la base des résultats de l'analyse fonctionnelle, les fonctions réalisées par les équipements.
- (d) La spécification opérationnelle
qui décrit les équipements impliqués dans la réalisation des fonctions d'une solution, du point de vue de leurs performances et de leurs caractéristiques techniques.

A partir de ces différents niveaux de description, deux modèles de spécification peuvent être définis.

Le modèle de spécification des besoins est décrit dans le paragraphe 2.4.

Le formalisme de spécification des besoins du système EPOS est à la base de ce modèle, moyennant quelques adaptations pour la prise en compte des problèmes mécaniques.

Le modèle de spécification du système fait l'objet du paragraphe 2.5. Il reprend certains concepts du langage de spécification d'un système proposé par l'outil EPOS.

2.3. Modèle de cycle de vie

2.3.1. Objectifs

La pré-étude, le développement, la réalisation, l'exploitation et la maintenance d'un système de fabrication constituent les différentes phases du cycle de vie d'un tel système.

La description de cette évolution peut être structurée selon un modèle de cycle de vie.

Un tel modèle facilite la description de chaque phase du cycle de vie selon un format cohérent et standard.

En outre, il préconise comment :

- * justifier et fournir un contexte pour les termes utilisés,
- * aborder la description d'une étape du cycle de vie,
- * valider les résultats de cette étape,
- * avoir une référence commune pour comparer les différentes phases d'un cycle de vie.

Dans le domaine du Génie Logiciel, de nombreux modèles de cycle de vie ont déjà été proposés [CALVEZ-82, LEHMAN-80, CALVEZ-78].

Ils fournissent une méthode d'approche et structurent la description de systèmes informatiques. Ce sont des modèles séquentiels, hiérarchiques, basés sur des représentations de graphes, ...

Le modèle de Mac DERMID, formalisant le processus de développement de systèmes informatiques dans un environnement Ada, a été retenu pour les raisons détaillées dans le paragraphe suivant [Mac DERMID-84].

2.3.2. Modèle de Mac DERMID

A l'inverse des autres modèles pré-cités, le modèle de MAC DERMID prend en compte la dynamique de la phase d'avant-projet, avec les aspects suivants :

- * construction d'une étape d'avant-projet,
- * validation d'une étape en liaison avec les objectifs du projet,
- * modification et maintenance d'une étape d'avant-projet,
- * parallélisme des étapes d'avant-projet.

Sa structure générique permet de modéliser aussi bien les phases d'exploitation ou de maintenance.

Son utilisation garantit donc une approche cohérente et standard des différentes phases du cycle de vie d'un système, et, à long

terme la faisabilité d'une approche intégrée.

Sa structure est décrite dans le paragraphe suivant.

2.3.3. Structure du modèle de Mac DERMID

Une étape d'avant-projet, d'après le modèle de Mac DERMID, est décrite sous quatre aspects :

- l'activité de conception ou de pré-étude (la transformation),
- la documentation résultant de cette activité (représentation),
- les procédures de vérification (vérification), et,
- les critères pour la validation finale de cette étape (contrôle).

Le modèle graphique, proposé par Mac DERMID, pour la représentation d'une étape d'avant-projet est donc le suivant :

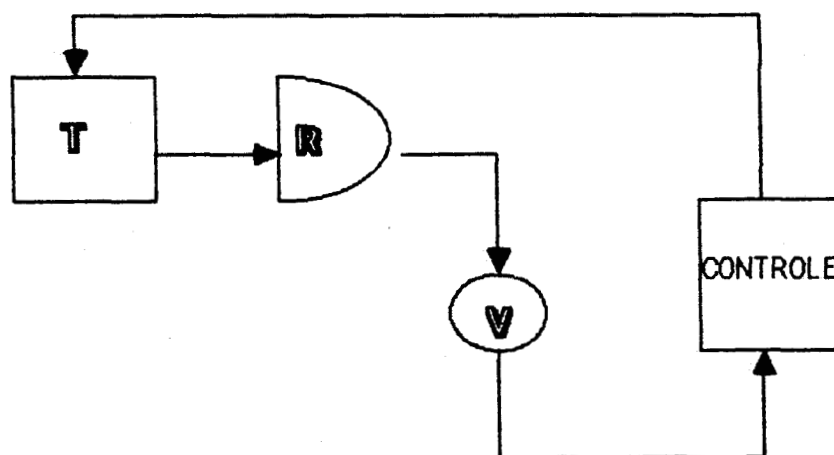


Fig 9 Modélisation d'une étape d'avant-projet

Le cycle complet de la pré-étude d'un système de fabrication est présenté page 37.

(a) La transformation

La notion de transformation, qui est attachée à une étape d'avant-projet, représente le travail accompli pour mener à bien cette étape à partir d'une certaine formulation du problème.

Une transformation peut commencer avant que la précédente ne soit finie, offrant la possibilité de représenter des activités simultanées.

(b) La représentation

A la fin de chaque transformation, correspondant à l'aboutissement d'une étape d'avant-projet, il est nécessaire de produire une documentation synthétisant les résultats obtenus.

La structure et le contenu des documents à fournir sont décrits sous la forme d'une entité appelée "représentation".

(c) La vérification

Un étape d'avant-projet est sanctionnée par une étude critique de la documentation. Cette étude peut être systématisée sous la forme de procédures de vérification standards à appliquer. Celles-ci sont regroupées et décrites dans l'unité de "vérification" de Mac DERMID.

Par exemple, une procédure de vérification peut consister en l'étude de la cohérence et de la complétude d'une représentation. Toutes les informations internes à une représentation doivent être cohérentes, une représentation doit être compatible avec les représentations antérieures. Enfin, tous les cas envisageables, dans le cadre d'un avant-projet, doivent être traités.

(d) Le contrôle

L'entité appelée "contrôle", dans le modèle de Mac DERMID, permet au concepteur de définir les points critiques pour l'évolution du projet.

Un tel modèle offre un cadre formel pour la description de certains problèmes de l'ingénierie d'avant-projet comme le contrôle des informations ou la garantie de l'intégrité des données d'un avant-projet.

Il facilite, en outre, la description de procédures simultanées ou répétitives.

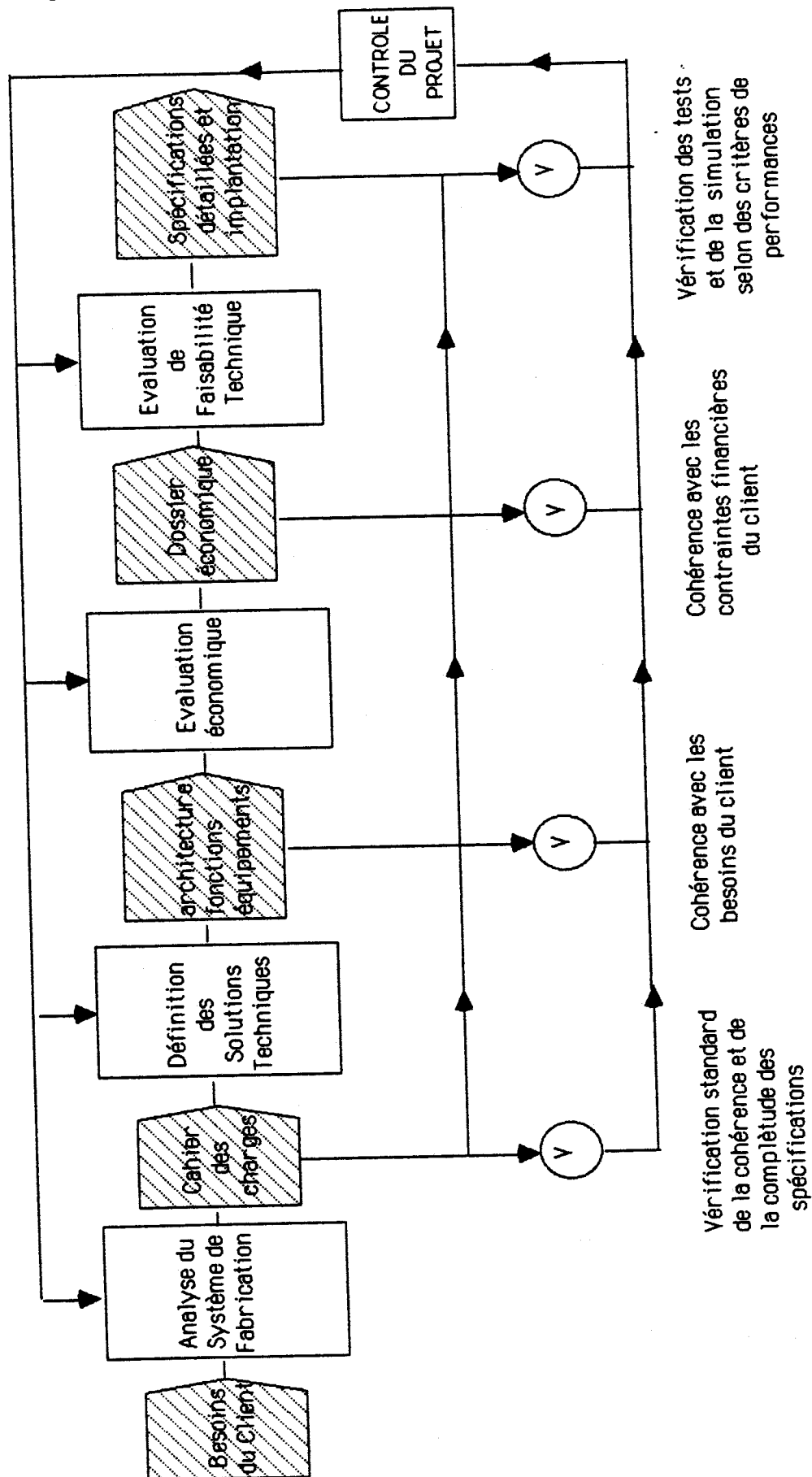


Fig 10 Modèle de MAC DERMID modélisant le processus de conception des systèmes de fabrication

2.3.4. Modélisation d'une étape d'avant-projet

Outre, la modélisation graphique proposée par le modèle de Mac DERMID, une étape d'avant-projet peut être décrite par une fiche de synthèse dont les rubriques principales sont les suivantes :

- * situation de l'étape dans la phase d'avant-projet,
- * document d'entrée,
- * description de l'étude à conduire et des outils à utiliser,
- * description du document à produire,
- * règles de vérification et outils à utiliser pour valider le document de sortie,
- * commentaires.

Cette fiche permet de spécifier précisément chacune des étapes d'un avant-projet, son propos, la forme et la syntaxe à utiliser pour chacune d'elles, ainsi que la détermination des points de contrôle pour une validation de l'étape.

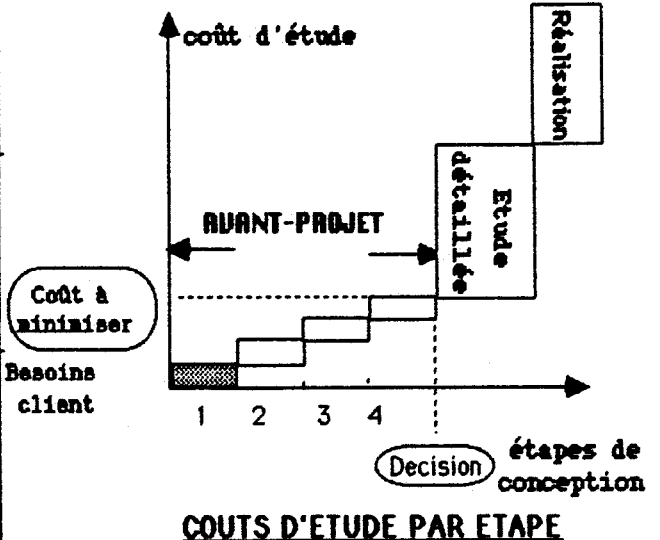
RENAULT AUTOMATION	
PROJET : METHODOLOGIE DE CONCEPTION DES SYSTEMES DE FABRICATION	
PHASE : AVANT-PROJET TACHE : ETAPE D'AVANT-PROJET	
Tâche précédente :	Tâche suivante :
Données d'entrée	Partenaires :
Activité de conception :	Verifications : Procédures: Outils :
Dossier à produire:	Commentaires :

Fig 11 Fiche standard de description d'une étape
 d'Avant-Projet

2.4. Modèle de représentation des besoins

2.4.1. Objectifs

La spécification des besoins comprend trois parties :

- (i) la spécification des besoins du client,
- (ii) la spécification des besoins du concepteur,
- (iii) la définition d'un glossaire de l'application.

Une syntaxe adéquate, prenant en compte ces trois aspects, doit être définie.

Le travail de recherche, réalisé dans ce domaine, s'est appuyé sur le langage de spécification des besoins, les modules d'analyse et de documentation, proposés par le système EPOS.

L'étude d'exemples industriels concrets a permis d'affiner la syntaxe du langage de spécification EPOS-R.

Son utilisation, pour la spécification des besoins, a été définie précisément.

Les résultats et conclusions auxquels nous avons abouti sont présentés dans les paragraphes suivants.

La syntaxe générale retenue ainsi que les différents éléments formels, importants pour la spécification d'un système de fabrication sont décrits.

De manière à éviter toute ambiguïté, les conventions de vocabulaire dans les paragraphes à venir seront les suivantes :

- * le terme de besoin est un terme général désignant l'ensemble des souhaits et contraintes exprimés pour l'automatisation d'une application.
Il se traduit de manière formelle par une spécification.
- * Un élément formel représente un type de besoin (souhait ou contrainte). La syntaxe d'un élément formel est constituée d'un ensemble de parties formelles.

Tous les termes de spécification sont en anglais ; le logiciel EPOS ne proposant pas de version française à l'heure actuelle.

2.4.2. Syntaxe générale retenue

La syntaxe de base, pour la représentation des besoins, est la suivante :

<BESOINS> ::= <SECTION> / <GLOSSAIRE>

<SECTION> ::= <en-tête section> <corps-spécification> (<section>)

<en-tête section> ::= <identificateur> (<abréviation>) <titre>

<identificateur> ::= <entier> (<identificateur>)

(<abréviation>) ::= 'SHORTNAME' <chaîne de caractères>

<titre> ::= <chaîne de caractères>

<corps-spécification> ::= (<texte> / <élément formel> / <concept> /

<partie FULFILS> / <partie SUBSTITUTE>) (<corps-spécification>)

<texte> ::= <chaîne de caractères>

<élément formel> ::= **<REQUIREMENT-CONSTRAINT>** (<catégorie>)

(<chaîne de caractères> **<DECISION-PROCESS>**)

<concept> ::= <terme> (<concept>)

<terme> ::= **'TERM'** <identificateur-terme>

<GLOSSAIRE> ::= <terme> (<REFERENCE>) (<catégorie>)

<définition> (<glossaire>)

Fig 12 Syntaxe générale des besoins

Tous les mots en italique et en caractère gras (REQUIREMENT, CONSTRAINT, FULFILS, ...) constituent la syntaxe de base du langage de spécification des besoins.

Ils sont détaillés dans le paragraphe suivant.

L'exemple page suivante présente un extrait du cahier des charges de l'application de conditionnement des pare-brise où l'on retrouve les trois parties de la spécification des besoins:

- les besoins client exprimés suivant le formalisme <SECTION>,
- les besoins constructeurs exprimés suivant le même formalisme,
- et, le glossaire exprimé suivant le formalisme <GLOSSAIRE>.

Dans la partie glossaire sont définis tous les termes (et non pas uniquement les termes fonctionnels) qui se trouvent dans la partie descriptive de la spécification (besoin client ou constructeur) et qui doivent être définis.

Nous pouvons remarquer, dès à présent, sur la base de cet exemple :

- (i) les références croisées entre la spécification des besoins du client et du concepteur (FULFILS, SUBSTITUTE),
- (ii) les références croisées entre les termes employés dans la spécification des besoins et le glossaire de l'avant-projet (TERM),
- (iii) le lien entre les termes du glossaire et les composants du futur système de fabrication. le mot-clef REFERENCE accolé à un terme du glossaire définit implicitement ce terme comme un élément du système de fabrication à réaliser.

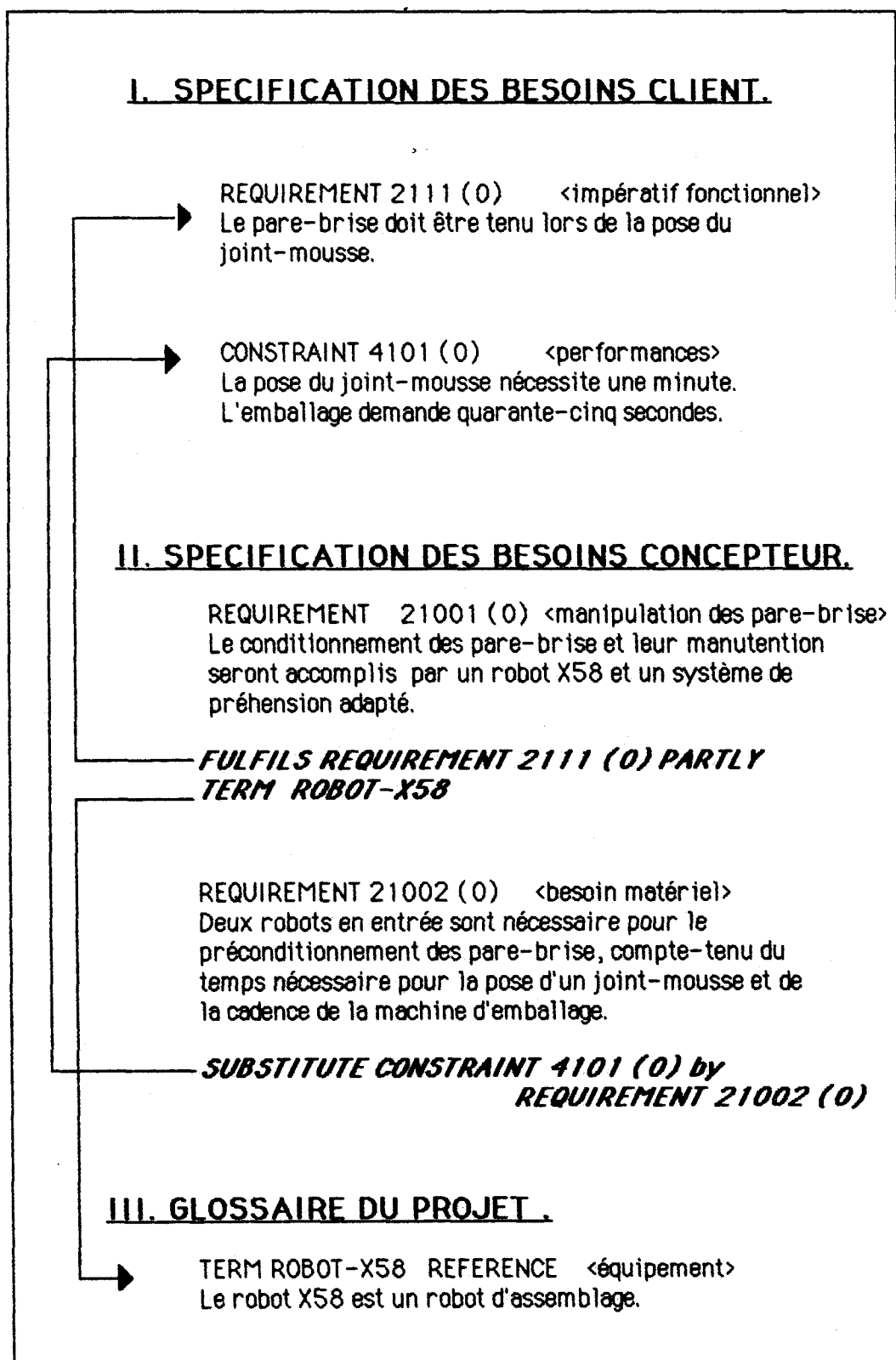


Fig 13 Extrait du cahier des charges de l'application
de conditionnement des pare-brise

2.4.3. Syntaxe détaillée de la spécification des besoins

"REQUIREMENT" et "CONSTRAINT" sont les deux éléments formels du langage de spécification qui permettent d'exprimer à la fois les besoins du client et du concepteur.

"DECISION-PROCESS" est une partie formelle, proposée par le système EPOS pour remplacer la partie textuelle d'un souhait ("REQUIREMENT") ou d'une contrainte ("CONSTRAINT").

"CATEGORY" et "TERM" sont des parties formelles qui permettent d'affiner la description d'un élément formel (souhait ou contrainte).

Enfin, "FULFILLS" et "SUBSTITUTE" décrivent comment certains besoins du client sont satisfaits par des souhaits ou des contraintes du concepteur.

2.4.3.1. REQUIREMENT

Cet élément formel permet de décrire les souhaits du client pour l'automatisation de son application.

REQUIREMENT 2111 (0) <impératifs fonctionnels>
Le pare-brise doit être tenu lors de la pose du joint-mousse.

D'après la syntaxe de Backus-Naur, il est caractérisé par un identificateur (2111 (0) dans le cadre de notre exemple) et une catégorie : impératifs fonctionnels.

2.4.3.2. CONSTRAINT

Cet élément formel permet de décrire les contraintes imposées par le client pour l'automatisation de son application.

La syntaxe est la même que celle d'un "REQUIREMENT".

CONSTRAINT 4101 (0) <performances >
La pose du joint-mousse nécessite une minute. L'emballage demande quarante-cinq secondes.

2.4.3.3. DECISION-PROCESS

Cette partie formelle définit les évolutions possibles d'un système, à partir d'une représentation matricielle.

Les paramètres déterminants pour l'évolution du système sont spécifiés dans la partie haute de la matrice. L'ensemble des états résultant d'une combinaison de ces

paramètres sont décrits dans la partie basse de cette même matrice.

Chaque colonne représente une combinaison précise de ces paramètres à laquelle sont associés les états résultants du système.

Ce formalisme est utilisé, le plus souvent, lors de la spécification d'un fonctionnement en mode dégradé du système automatisé.

En entrée, les équipements, considérés comme critiques, sont toujours spécifiés.

En sortie, le concepteur envisage toutes les procédures à appliquer en cas de panne de l'un de ces équipements.

Dans le cadre de notre exemple industriel, les équipements suivants ont été identifiés :

- deux robots en entrée pour le préconditionnement des pare-brise,
- un robot en sortie de la chaîne d'emballage pour assurer l'évacuation des pare-brise,
- la machine d'emballage qui est imposée par le client.

Par conséquent, des fonctionnements en modes dégradés sont à prévoir en cas de panne de l'un des deux robots en entrée, du robot de sortie ou de la machine d'emballage.

C'est ce que traduit la table de décision page suivante.

Les symboles :

P et M signifient respectivement en Panne et en Marche.
x et - signifient respectivement procédure à appliquer ou à ignorer. Dans ce cas, l'ordre d'exécution de ces procédures est aléatoire.

Par contre, les chiffres dans le tableau indiquent un ordre de priorité, dans l'exécution des procédures.

Enfin, la partie ELSE permet de spécifier une procédure standard à appliquer pour tous les autres cas de panne qui n'auraient pas été explicités.

Par exemple, dans le cas de panne de l'un des deux robots en entrée (colonne 4), la procédure souhaitée par le client, est la suivante :

- déclencher en premier lieu une alarme, et,
- stopper l'activité des deux robots à l'entrée.

DT : Modes dégradés		1	2	3	4	5	ELSE
C1	Robots-entrée	M	P	M	P	P	
C2	Robot-sortie	M	P	P	M	M	
C3	Faredeleuse	M	M	M	M	P	
01	Arrêt général	-	*	1	-	*	-
02	Arrêt-entrée	-	-	-	2	-	*
03	Arrêt-sortie	-	-	-	-	-	*
04	Arrêt-emballage	-	-	-	-	-	*
05	Alarme	-	*	2	1	*	*

Fig 14 Spécification des modes dégradés envisageables

2.4.3.4. Partie CATEGORY

La partie formelle "CATEGORY" permet de définir les classes d'appartenance d'une spécification.

Un exemple de classification des spécifications :

- classe 1 : maintenance,
- classe 2 : économique,
- classe 3 : spécification de type logicielle,
- classe 4 : spécification de type matérielle,

La syntaxe est la suivante : <classe 1, classe 2, ...>

Dans le cadre de notre exemple, la contrainte 4100 (0), classée comme étant une contrainte de type matérielle (classe 4), est spécifiée de la manière suivante :

CONSTRAINT 4100 (0) <contrainte matérielle>

La machine d'emballage existe déjà et doit être réutilisée. La pose du joint mousse doit se faire en maintenant les pare-brise par leur face externe"

TERM faredeleuse

2.4.3.5. Partie TERM

La partie formelle "TERM" permet d'identifier les termes qui nécessitent une définition particulière, lors de la description d'un besoin.

La syntaxe est la suivante : TERM <name>
ainsi que le montre l'exemple précédent.

2.4.3.6. Partie FULFILS

La partie formelle "FULFILS" permet d'identifier les besoins du client qui sont satisfaits par des contraintes ou des choix fait a priori par le concepteur.

La syntaxe est la suivante :
FULFILS <type de la spécification> <identificateur>

REQUIREMENT 21001 (0) <manipulation des pare-brise>
Le conditionnement des pare-brise et leur manutention seront accomplis par un robot X58.

FULFILS REQUIREMENT 2101 (0) PARTLY
TERM ROBOT-X58

Le besoin du client 2101 (0), qui mentionnait la nécessité de prévoir un moyen de manipulation des pare-brise, est satisfait, en partie (le système de préhension reste à définir), par le besoin du constructeur 21001 (0).

Par conséquent, ce besoin du client n'apparaîtra plus lors de la spécification de la solution technique.

2.4.3.7. Partie SUBSTITUTE

La partie formelle "SUBSTITUTE" permet d'identifier les besoins du client, qui sont redondants avec les choix du concepteur.

La syntaxe est la suivante :
SUBSTITUTE <type de la spécification> <identificateur>

REQUIREMENT 21002 (0) <besoin matériel>
Deux robots en entrée sont nécessaires pour le préconditionnement des pare-brise, compte-tenu du temps de pose d'un joint-mousse.

SUBSTITUTE CONSTRAINT 4101 (0) by REQUIREMENT 21002 (0).

Le besoin 21002 (0) a été identifié par le concepteur sur la base des informations du client concernant les temps nécessaires pour la réalisation des opérations de préconditionnement et d'emballage.

Il se substitue à la contrainte du client 4101 (0) qu'il prend en compte implicitement.

2.4.3.8. Partie REFERENCE

La partie formelle "REFERENCE" permet d'identifier les termes du glossaire qui représentent des composants du système de fabrication.

Dans le cadre de notre exemple, le concepteur préconise le choix d'un robot X58 pour la manipulation des pare-brise.

Celui-ci est décrit selon la syntaxe suivante :

TERM ROBOT-X58 REFERENCE <équipement>
Le robot X58 est un robot d'assemblage.

Le robot X58 est donc considéré comme un élément du futur système automatisé. Il devra apparaître en tant que tel dans la liste des équipements qui composent la solution technique.

2.4.4. Avantages et inconvénients

Les avantages d'un tel modèle de spécification des besoins sont directement liés à son aspect semi-formel.

- * Un traitement informatique des procédures de vérification d'ordre syntaxique est possible.
- * Une référence, de la spécification des besoins vers la spécification de la solution technique, est établie avec la partie formelle REFERENCE.
- * Des vérifications croisées entre les spécifications des besoins du client et du concepteur sont possibles grâce aux parties formelles FULFILS et SUBSTITUTE. Toute redondance est ainsi évitée.
- * La rédaction d'un glossaire du projet est prévue.
- * L'intégration de commentaires rend ce langage de spécification des besoins lisible par tous les partenaires impliqués dans la spécification des besoins.
- * La définition de catégories pour une spécification facilite une lecture sélective du cahier des charges selon des critères de priorité ou de type fonctionnel (spécification logicielle, mécanique, ...).

Parmi les inconvénients envisageables, notons la contrainte résultant de l'utilisation d'un formalisme d'écriture même si celui-ci est très souple.

De même, l'existence d'une représentation graphique des besoins faciliteraient leur formalisation et leur compréhension.

2.5. Modèle de spécification du système

2.5.1. Objectifs

L'objectif principal d'un modèle de spécification d'un système de fabrication est de fournir un langage formel dont les éléments permettent de décrire toutes les facettes de ce système automatisé.

Tous les domaines d'application : usinage, soudure ou encore assemblage doivent être pris en compte.

Pour des raisons de lisibilité et de portée, ce modèle doit offrir un formalisme textuel et graphique. Ce dernier devant être aussi varié que possible et équivalent aux entités formelles.

La définition d'un tel modèle a été basé sur le langage de spécification proposé par le système EPOS. Néanmoins, la syntaxe générale retenue est une syntaxe simplifiée par rapport à celle du système EPOS.

Elle a été établie à la suite des cas industriels que nous avons traités. Elle est présentée dans le paragraphe 2.5.2.

2.5.2. Syntaxe générale retenue

Dans le cadre d'un avant-projet, la spécification d'un système de fabrication est relativement succincte, si bien que lors de sa description, seules les trois entités suivantes du système EPOS sont utilisables :

- ACTION MODULE,
- ACTION, et,
- DEVICE.

Le formalisme ACTION-MODULE permet de spécifier les fonctions indépendantes identifiées lors de l'analyse fonctionnelle générale.

ACTION et DEVICE sont utilisés respectivement pour la spécification fonctionnelle et opérationnelle.

Nous avons adapté la définition syntaxique de ces trois entités formelles aux problèmes de la spécification des systèmes de fabrication.

Ce travail est présenté dans les paragraphes 2.5.3. à 2.5.5. sur la base de l'exemple industriel introduit au début de ce mémoire.

La structure syntaxique sous-jacente de toutes ces entités reste celle proposée par le système EPOS :

- description de l'entité en langage naturel,
- décomposition de l'entité,
- relations avec les autres entités.

La partie description est la même pour toutes les entités de spécification. Elle est décrite, une fois pour toute dans ce paragraphe.

Par contre, les parties décomposition et inter-relations sont spécifiques à chaque entité de spécification. Elles seront décrites dans les paragraphes 2.5.3. à 2.5.5.

La partie description (mot-clef : DESCRIPTION) est structurée selon plusieurs mots-clefs (PURPOSE, NOTE, FULFILS, ...) qui fournissent des informations historiques ou descriptives se rattachant à l'entité de spécification étudiée.

Elle permet, grâce à l'élément formel FULFILS d'identifier tous les besoins du client ou du concepteur qui sont satisfaits par l'entité de spécification étudiée.

La structure syntaxique complète de la partie description d'un masque de spécification, d'après la syntaxe de Backus-Naur, est la suivante :

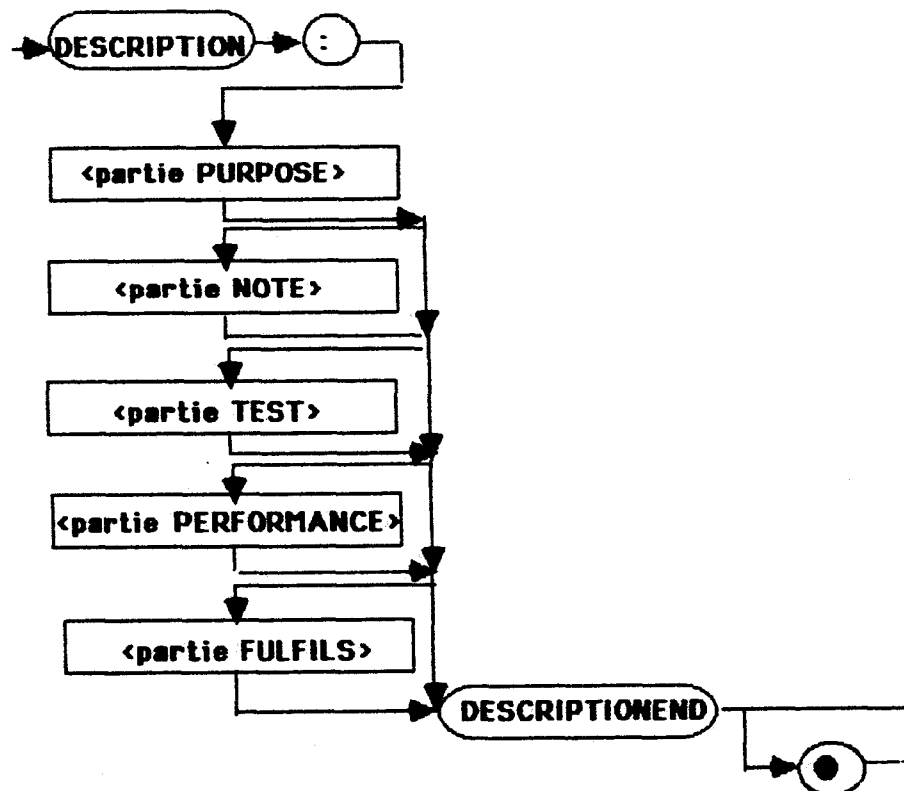


Fig 15 Syntaxe de la partie description d'un masque de spécification

L'élément formel **PURPOSE** permet au concepteur d'introduire des commentaires et une présentation brève de l'entité de spécification (**ACTION-MODULE**, **ACTION** ou **DEVICE**).

L'élément formel **NOTE** introduit des remarques présentant un intérêt particulier pour la suite de l'étude.

La partie **TEST** fournit la liste des tests prévus pour la validation d'une fonction ou d'un équipement décrits d'après les formalismes **ACTION** et **DEVICE**.

Les performance attendues au niveau d'un équipement ou de l'exécution d'une opération sont décrites dans l'élément formel **PERFORMANCE**.

FULFILS permet l'identification des besoins du client ou du concepteur qui sont satisfaits par l'entité de spécification. (vérification croisée des besoins).

2.5.3. Spécification d'une fonction générale : ACTION-MODULE

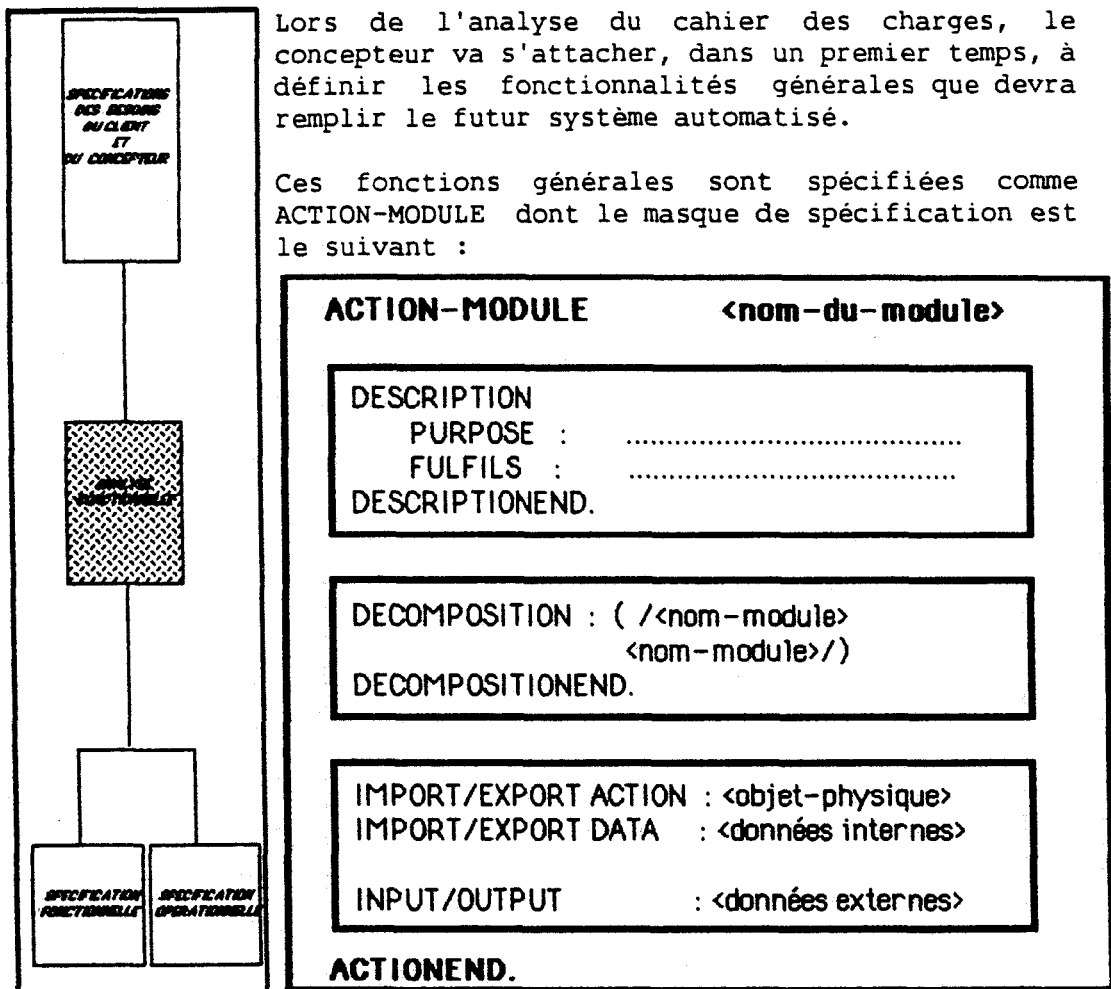


Fig 16 Masque de spécification d'une fonction générale

On retrouve la structure sous-jacente avec ses trois parties : description, décomposition et inter-relations.

La partie décomposition permet d'identifier les modules fils.

Dans la partie inter-relations, IMPORT/EXPORT ACTION représente le flux des objets physiques entre les différentes fonctionnalités.

Cette interprétation résulte d'une extension du formalisme tel que le propose le système EPOS. Elle permet d'assurer, dans notre cas, une spécification cohérente et complète des systèmes de fabrication.

Les données échangées entre les différents modules sont décrites sous la forme d'IMPORT/EXPORT DATA.

INPUT/OUTPUT représente les données échangées avec le milieu extérieur, qui peut être soit un opérateur ou un autre système de fabrication.

L'interprétation des IMPORT/EXPORT ACTION et IMPORT/EXPORT DATA dans le cadre de ce mémoire est une interprétation productive que nous avons donné sur la base des exemples traités pour la SERI (affaire CATERPILLAR), ACMA-ROBOTIQUE (affaire MPRDOUAI) et la DPU (ACMA-BEAUCHAMP, CASE-IH).

Dans le cadre de notre exemple, nous avons identifié une fonction générale : "conditionnement des pare-brise". Elle a été formalisée d'après la syntaxe d'un ACTION-MODULE.

L'analyse du système de conditionnement des pare-brise nous a conduit à identifier, au niveau de décomposition inférieur, les fonctions suivantes :

- fonction de préconditionnement (PRECONDITIONNEMENT),
- fonction d'emballage (EMBALLAGE),
- fonction de déchargement (DECHARGEMENT).

La fonction de préconditionnement consiste à poser un profilé en mousse agglomérée sur la périphérie de la vitre (pose du joint-mousse).

Le pare-brise est ensuite déposé sur un tapis roulant pour être acheminé vers la machine d'emballage (faredeleuse).

La fonction d'emballage consiste à envelopper le pare-brise d'un film plastique protecteur. Celui-ci est chauffé. En se refroidissant, il se rétracte et se tend sur le pare-brise.

La fonction de déchargement consiste à évacuer les pare-brise emballés.

L'évolution de la fonction de conditionnement est régie par des données d'entrée/sortie qui correspondent à des actions de l'opérateur (ACTION OPERATEUR).

A 3.1.**ACTION-MODULE CONDITIONNEMENT.****DESCRIPTION :**

PURPOSE : "27 types de pare-brise différents doivent être conditionnés suivant un certain procédé de fabrication".

DESCRIPTIONEND.

DECOMPOSITION : (/PRECONDITIONNEMENT, EMBALLAGE, DECHARGEMENT/).

DECOMPOSITIONEND.

IMPORT-ACTION : PARE-BRISE,

EXPORT-ACTION : PARE-BRISE-CONDITIONNE,

IMPORT-DATA : PAS-DE-PARE-BRISE-CONTAINER-D'ENTREE,

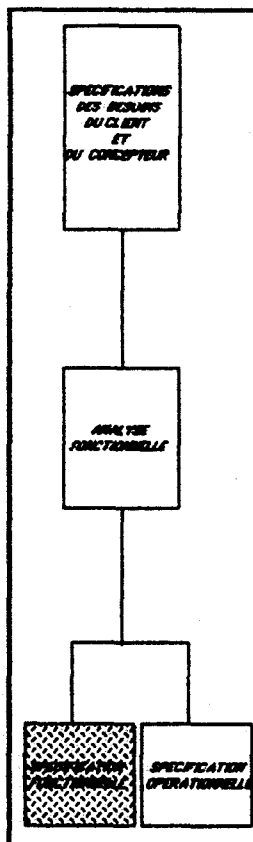
EXPORT-DATA : INFO-PANNE-CONDITIONNEMENT.

INPUT : ACTION-OPERATEUR.

ACTIONEND.

Fig 17 Fonction générale de conditionnement

2.5.4. Spécification d'une action élémentaire : ACTION



Lorsque le concepteur a entièrement analysé les fonctionnalités que doit remplir le système automatisé, il est amené à proposer des solutions technologiques.

Des équipements et les fonctions associées sont décrits selon un niveau de détail suffisant pour pouvoir passer ensuite à la réalisation de cette solution, et, à la programmation des équipements.

Nous avons constaté, sur la base des projets industriels traités, que ces actions élémentaires sont complexes.

Souvent, des contraintes de synchronisation ou de coopération entre équipements sont à prendre en compte.

Le formalisme ACTION que propose le système EPOS est parfaitement adapté à la description de ces contraintes.

La syntaxe générale reste basée sur les trois parties: description, décomposition et inter-relations.

ACTION	<nom-action>
DESCRIPTION PURPOSE : FULFILS : DESCRIPTIONEND. DECOMPOSITION : R1, R2, R3, R4 SYNCHRO : <conditions-de-synchronisation> CYCLE : <cycle-fonction> CODE : <codage-action> DECOMPOSITIONEND. INPUT : <données-entrée> OUTPUT : <données-sortie> PROCESSED : <nom-équipement réalisant la fonction>	
ACTIONEND.	

Fig 18 Masque de spécification d'une action

Dans la partie décomposition, la description d'un flux de commande est possible sur la base des règles syntaxiques R1,R2,R3 et R4, décrites ci-après.

R1 Des actions indépendantes sont spécifiées suivant la syntaxe ci-dessous : DECOMPOSITION : (/ A1 , A2 /)
DECOMPOSITIONEND.

R2 Des actions dépendantes sont spécifiées suivant différentes instructions de contrôle telle que :

IF...THEN...ELSE,
WHILE ... DO,
SWITCH DO IN CASE
IN CASE
NI

R3 Des actions dépendantes et indépendantes ne peuvent être spécifiées à un même niveau de décomposition. Par exemple deux actions indépendantes ne peuvent être parallèles ou séquentielles vis-à-vis d'une troisième action.

R4 Si une action est décomposée en actions dépendantes ; celles-ci ne peuvent être redécomposées en actions indépendantes.

Les contraintes de synchronisation et de cycle peuvent être décrites avec les éléments formels SYNCHRO et CYCLE).

Pour les derniers niveaux de spécification des fonctions des équipements, le codage de ces fonctions (un langage robot, un code équipement, par exemple) peut être introduit sous l'élément formel CODE.

Dans la partie inter-relations, les données échangées entre l'action identifiée, et, l'extérieur sont spécifiées sous l'élément formel : INPUT/OUTPUT.

L'élément formel PROCESSED permet d'indiquer le support de réalisation de l'action identifiée.

Un exemple de spécification mécanique (avec la spécification de la fonction de préconditionnement) et informatique (spécification de la fonction de commande d'un robot) sont présentés pages suivantes.

La fonction de préconditionnement, identifiée en 2.5.3. lors de la décomposition de l'action-module CONDITIONNEMENT dépend des choix techniques qui ont été faits.

En effet, deux robots en entrée préconditionnent des pare-brise et les déposent sur un même tapis roulant.

La spécification de la tâche élémentaire de préconditionnement pour un robot particulier nécessite donc la prise en compte des contraintes de parallélisme.

Le formalisme adapté est celui d'une ACTION.

La fonction de préconditionnement a pour objectif le déchargement, la pose du joint-mousse et le dépôt du pare-brise par le robot 1 sur le tapis roulant.

Elle a été identifiée en réponse au besoin des concepteurs concernant la manipulation des pare-brise (REQUIREMENT 2111(0)).

Elle prend en compte les temps d'exécution des opérations de pose du joint-mousse et d'emballage (CONSTRAINT 4101 (0)).

Le concepteur peut spécifier les tests nécessaires pour valider cette opération ainsi que les performances attendues ou bien faire référence à un document annexe.

Dans le cadre de notre exemple, le concepteur fait référence au document annexe 5645.

L'opération doit être réalisée en une minute quinze secondes.

Elle est déclenchée lorsqu'un pare-brise a déjà été déposé et qu'un autre se trouve en attente dans le plateau d'entrée.

Elle se déroule selon une séquence de six opérations, détaillées dans la partie décomposition (DECOMPOSITION) du masque de spécification.

Tout d'abord, le pare-brise est pris par le robot afin d'assurer la pose du joint-mousse.

Avant de déclencher l'opération de dépôt du pare-brise sur le tapis roulant, le robot attend l'autorisation de dépôt qui est liée à l'activité du robot 2.

La fin de l'opération de dépôt est signalée au robot 2 afin qu'il puisse à son tour avoir accès au tapis roulant.

Ces opérations sont définies et définissent respectivement des informations en entrée et sortie.

Il s'agit de données concernant les possibilités d'accès au tapis, la trajectoire du robot ou des informations de suivi.

Enfin, le support de réalisation de cette fonction de préconditionnement est le robot 1.

La fonction de commande du robot 1, spécifiée ensuite, permet au concepteur de décrire comment il envisage le pilotage du robot1.

Il s'agit de la description d'une fonction logicielle, donc d'une utilisation classique du système de spécification EPOS.

Cette fonction de pilotage comprend une fonction de suivi, de coordination, de gestion de données et de modes dégradés.

Elle est initialisée par un événement SYSTEMSTART. Son évolution est régie par un ensemble de données sur l'état des systèmes de déchargement et d'emballage.

A 5.1

ACTION PRECONDITIONNEMENT-1.

DESCRIPTION :

PURPOSE : "Le déchargement, la pose du joint-mousse, et, le dépôt du pare-brise sont faits par le robot 1".

TEST : "Voir document 5645"

FULFILS : REQUIREMENT 2111 (0),
CONSTRAINT 5301 (0).

DESCRIPTIONEND.

DECOMPOSITION : SET (EVDEPOT1) ;
WHILE CONDITION-PARE-BRISE-PLATEAU1
PRISE1 ;
POSE-JOINT-MOUSSE1 ;
RESET (EVDEPOT2) ;
WAIT FOR EVDEPOT1
THEN DEPOT1
WAITEND ;
SET (EVDEPOT2)

OD.

DECOMPOSITIONEND.

TRIGGERED : SYSTEMSTART.

INPUT : AUTORIS-PRISE1,
AUTORIS-POSE-JOINT-MOUSSE1,
N-AUTORIS-DEPOT1,
TRAJECTOIRE-DEPOT1.

OUTPUT : SUIVI-POSE-JOINT-MOUSSE1,
SUIVI-PRISE1,
SUIVI-DEPOT1.

PROCESSED : ROBOT1.

ACTIONEND.

Fig 19 Spécification de l'action de Préconditionnement

A 5.3

ACTION COMMANDE-ROBOT 1.

DESCRIPTION :

PURPOSE : "Cette fonction assure le contrôle et le suivi du robot1".

DESCRIPTIONEND.

DECOMPOSITION : PARALLEL (SUIVI1, COORDINATION1,
GESTION-DONNES1, MODE-DEGRADE1).

DECOMPOSITIONEND.

TRIGGERED : SYSTEMSTART.

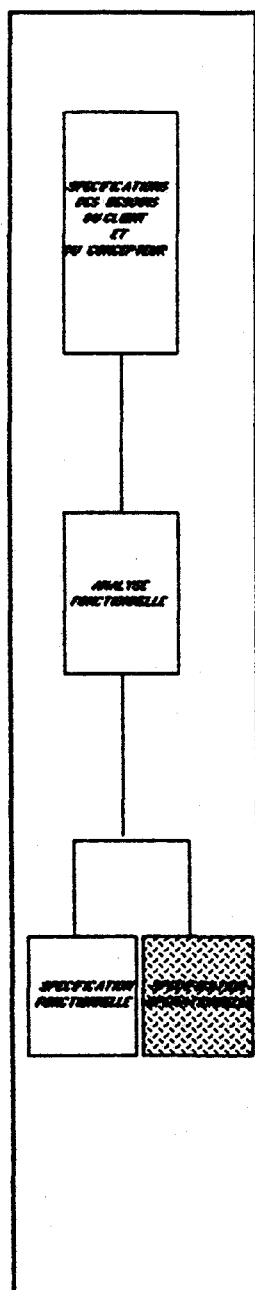
INPUT : SUIVI-PRISE1,
SUIVI-POSE-JOINT-MOUSSE,
SUIVI-DEPOT1,
INFO-PANNE-CHARGEMENT,
INFO-PANNE-EMBALLAGE.

OUTPUT : AUTORIS-PRISE1,
AUTORIS-POSE-JOINT-MOUSSE1,
DEMANDE-DEPOT1,
FICHIER-ROBOT1,
INFO-PANNE-PRECONDITIONNEMENT.

ACTIONEND.

Fig 20 Spécification de l'action de commande du robot 1

2.5.5. Spécification d'un équipement : DEVICE



Lors de la description d'un système de fabrication, le concepteur définit les équipements d'après leurs caractéristiques techniques (performances de fonctionnement, composants de l'équipement : les composants d'un robot, par exemple, sont la pince, le bras, ...).

La description formelle des équipements se fait en utilisant l'entité DEVICE du langage dont la syntaxe est la suivante :

DEVICE	<nom-équipement>
DESCRIPTION PURPOSE : FULFILS : DESCRIPTIONEND.	
COMPONENTS : <nom-équipement> TYPE <catégorie> <nom-équipement> TYPE <catégorie>	
INOUT: données échangées avec l'environnement CONNECTIONS : données internes à l'équipement PERFORMS : fonctions réalisées par l'équipement	
DEVICEEND.	

Fig 21 Masque de spécification d'un équipement

Dans la partie décomposition, l'élément formel COMPONENTS permet de décrire les composants de l'équipement.

Dans la partie inter-relations, les données échangées avec l'environnement, décrites dans l'élément formel INOUT, peuvent être des données opérateur ou encore provenir d'équipements externes au système de fabrication étudié.

Les différentes actions de ses composants sont déclenchées ou produisent des signaux logiques détaillés dans l'élément formel CONNECTIONS.

Enfin, la liste des actions réalisées par l'équipement est spécifiée dans l'élément formel PERFORMS.

Dans le cadre de notre exemple, les contraintes imposés par le client sont les suivantes :

- le poids et la taille des pare-brise à manipuler (CONSTRAINT 3200 (0) et 3201 (0)),
- les contraintes dues à l'environnement (CONSTRAINT 5100)

Elles ont conduit au choix du robot X58 en tant que robot d'entrée.

Celui-ci est décrit page suivante selon ses performances (charge maximum 60 kilogrammes, ...), les besoins qu'il satisfait et ses composants (ARMOIRE-DE-COMMANDE-I, BRAS-I, EFFECTEUR-I).

Le fonctionnement du robot 1 est régi par les actions ou les alarmes lancées par l'opérateur, ainsi que les informations de panne concernant les activités d'emballage et de chargement.

Il accomplit la fonction de préconditionnement-1, qui a été décrite dans le paragraphe précédent (2.5.4.)

DY 1

DEVICE ROBOT1.

DESCRIPTION :

PURPOSE : "Ce robot permet de prendre le pare-brise dans un container et le dépose sur le tapis d'entrée de la faredeleuse après la pose du joint-mousse".

PERFORMANCE : "Charge maximum de 60 kilogrammes"

FULFILS : REQUIREMENT 3200 (0),
CONSTRAINT 3201 (0),
CONSTRAINT 5100 (0).

DESCRIPTIONEND.

COMPONENTS : ARMOIRE-DE-COMMANDE-I,
BRAS-I,
EFFECTEUR-I.

INOUT : ALARME-OPERATEUR-I,
ACTION-OPERATEUR-I,
INFO-PANNE-EMBALLAGE-I,
INFO-PANNE-CHARGEMENT-I,
N-PRES-PARE-BRISE-TAPIS-ENTREE.

CONNECTION :
SIGNAL ALARME-OPERATEUR:
ALARME-OPERATEUR<- ARMOIRE-DE COMMANDE

PERFORMS : PRECONDITIONNEMENT-1

DEVICEEND.

Fig 22 Spécification de l'équipement ROBOT-1

2.5.6. Spécifications graphiques

Le système EPOS propose un outil de documentation graphique, capable de générer automatiquement, sur la base des spécifications écrites, différents types de diagrammes.

Tous ne sont pas utilisables pour la documentation d'une pré-étude d'un système de fabrication.

Sur la base des projets industriels traités et des besoins identifiés dans les bureaux d'études, nous recommandons l'utilisation :

- des diagrammes hiérarchiques,
- des diagrammes de flux de données,
- des diagrammes de flux de commande,
- des réseaux de Petri ordinaires,
- et, des diagrammes d'équipements.

Les diagrammes hiérarchiques décrivent la structure de dépendance existante.

En cours d'étude, ils fournissent un document synthétique des étapes d'analyse et de spécification fonctionnelle. Ils permettent d'estimer précisément l'état d'avancement du projet.

Les diagrammes de flux de données définissent les échanges de données entre les principales fonctions du système (identifiées lors de l'analyse fonctionnelle).

Les diagrammes de flux de commande décrivent de manière précise les conditions d'évolution des fonctions d'un équipement.

Les réseaux de Pétri ordinaires constituent un document indispensable pour la description des contraintes de parallélisme entre les différentes fonctions identifiées lors de la spécification fonctionnelle.

Enfin, les diagrammes d'équipements décrivent un équipement (un robot par exemple), ses composants (bras, pince) et le flux de données (signaux logiques) circulant entre ses composants. Ils représentent une illustration graphique des spécifications opérationnelles.

2.5.7. Avantages et inconvénients

Les avantages d'un tel modèle de spécification des systèmes de fabrication sont liés à son aspect formel.

- * Un système de fabrication est décrit de manière modulaire et hiérarchique.
- * Un traitement automatique des procédures de vérification syntaxique est possible.
- * La vérification croisée entre les besoins et les solutions proposées est systématique (FULFILS). Il est possible d'identifier très rapidement les oublis ou les dérogations par rapport aux exigences du client.
- * Les liens entre les spécifications fonctionnelles et opérationnelles garantissent la description cohérente d'une solution (PROCESSED et PERFORMS).
- * La génération automatique de diagrammes standards comme les diagrammes de flux de commande ou les Réseaux de Pétri ordinaires est prévue.

Il en résulte que l'utilisateur dispose d'un outil convivial offrant des moyens de représentation graphiques et textuels.

La démarche de conception est plus rigoureuse.

Le diagramme page suivante met en évidence trois extraits du dossier de spécification pour l'automatisation d'une application, à savoir :

- l'expression de besoins,
- la description d'une action élémentaire (ACTION), et,
- la description de l'équipement réalisant cette action (DEVICE ROBOT 1).

Les procédures de vérification automatiques mises en oeuvre pour la validation des spécifications (FULFILS, PROCESSED et PERFORMS) sont mises en évidence sur ce diagramme.

EXTRAIT DU CAHIER DES CHARGES CLIENT

REQUIREMENT 2111 (0) <impératif fonctionnel>
Le pare-brise doit être tenu lors de la pose du joint-mousse.

CONSTRAINT 3200 (0) <poids>
Un pare-brise pèse environ dix kilogrammes.

CONSTRAINT 3201 (0) <taille>
Un pare-brise a les dimensions suivantes : 2*1.5 (mètres).

CONSTRAINT 5100 (0) <environnement naturel>
La température dans l'atelier est de 60 degrés Celsius.
Il y a par ailleurs un éclairage très fort.

SPECIFICATIONS FONCTIONNELLES**SPECIFICATIONS OPERATIONNELLES**

5.1

ACTION PRECONDITIONNEMENT**DESCRIPTION :**

PURPOSE : "Le déchargement, la pose du joint-mousse, et, le dépôt du pare-brise sont faits par le robot 1".

TEST : "Voir document 5645"

FULFILS : REQUIREMENT 2111 (0),
CONSTRAINT 4101 (0).

DESCRIPTIONEND.

DECOMPOSITION : SET (EVDEPOT1);
WHILE CONDITION-PARE-BRISE-PLATEAU1
PRISE1;
POSE-JOINT-MOUSSE1;
RESET (EVDEPOT2);
WAIT FOR EVDEPOT1
THEN DEPOT1
WAITEND;
SET (EVDEPOT2)
OD.

DECOMPOSITIONEND.

TRIGGERED : SYSTEMSTART.

INPUT : AUTORIS-PRISE1,
AUTORIS-POSE-JOINT-MOUSSE1,
N-AUTORIS-DEPOT1,
TRAJECTOIRE-DEPOT1.

OUTPUT : SUIVI-POSE-JOINT-MOUSSE1,
SUIVI-PRISE1,
SUIVI-DEPOT1.

PROCESSED : ROBOT1.

ACTIONEND.

DV 1

DEVICE ROBOT1.**DESCRIPTION :**

PURPOSE : "Ce robot permet de prendre le pare-brise dans un container et le dépose sur le tapis d'entrée de la ferdeleuse après la pose du joint-mousse".

PERFORMANCE : "Charge maximum de 60 kilogrammes"

FULFILS : REQUIREMENT 3200 (0),
REQUIREMENT 3201 (0),
CONSTRAINT 5100 (0).

DESCRIPTIONEND.

COMPONENTS : ARMOIRE-DE-COMMANDE-I,
BRAS-I,
EFFECTEUR-I.

INPUT : ALARME-OPERATEUR-I,
ACTION-OPERATEUR-I,
INFO-PANNE-EMBALLAGE-I,
INFO-PANNE-CHARGEMENT-I,
N-PRES-PARE-BRISE-TAPIS-ENTREE.

CONNECTION :
SIGNAL ALARME-OPERATEUR:
ALARME-OPERATEUR(- ARMOIRE-DE COMMANDE

PERFORMS : PRECONDITIONNEMENT-I

DEVICEEND.

Fig 23 Extrait du dossier de spécification du système de
conditionnement des pare-brise

2.6. Conclusion

Le développement d'une méthodologie de conception pour la pré-étude des systèmes de fabrication a justifié, dans un premier temps, la définition d'un modèle conceptuel.

Les besoins identifiées, entre autres l'adéquation au contexte de la Production en Industrie Manufacturière, nous ont conduit à proposer :

(a) d'une part, une technique d'approche et de description de la pré-étude des systèmes de fabrication sous la forme d'un modèle de cycle de vie,

(b) d'autre part, des formalismes d'écriture pour la documentation des différentes étapes de cette pré-étude sous la forme d'un outil de spécification.

La première partie de notre travail de recherche nous a permis de retenir l'outil industriel EPOS et d'envisager son adéquation aux problèmes de fabrication.

De la même manière, l'étude de l'état de l'art nous a amené à étudier le modèle de cycle de vie de Mac DERMID.

Il a été retenu pour la modélisation de la phase d'avant-projet.

Ce modèle conceptuel sert de base pour le présent mémoire et nous permet, maintenant, d'aborder le développement de la méthodologie de conception pour la pré-étude des systèmes de fabrication.

METHODOLOGIE D'AIDE A LA CONCEPTION
DES SYSTEMES DE FABRICATION

(chapitre 3)

3.1. Objectifs

La méthodologie, présentée dans le cadre de ce chapitre, s'appuie sur le modèle de cycle de vie de Mac DERMID et le système EPOS.

Cette méthodologie structure le savoir-faire des sociétés d'ingénierie, sous la forme d'un GUIDE CONCEPTEUR, en fournissant au concepteur une méthode d'analyse, un formalisme d'écriture adapté à ses besoins et des modules logiciels utilisables.

La présentation de ce guide a fait l'objet d'un dossier complet [AUGER-86/1].

La formalisation d'une telle méthodologie vise une meilleure efficacité dans la structuration des solutions proposées.

Enfin, elle facilite la compréhension du problème et de la solution proposée, pour toute l'équipe intervenant après la phase d'avant-projet.

La présentation de cette méthodologie est basée sur le traitement d'un exemple.

Nous avons rédigé un dossier complet des spécifications de la solution technique proposée pour le conditionnement de pare-brise.

Certains extraits sont présentés dans ce chapitre pour illustrer notre propos.

3.2. Etapes de la méthodologie

L'expertise des méthodes de travail de différentes sociétés d'ingénierie nous a conduit à proposer une méthodologie basée sur quatre étapes :

- ANALYSE DU SYSTEME DE FABRICATION,
- DEFINITION DES SOLUTIONS TECHNIQUES,
- EVALUATION ECONOMIQUE,
- EVALUATION DE FAISABILITE TECHNIQUE.

Ces quatre étapes, standards pour la réalisation d'un avant-projet, sont décrites dans les paragraphes 3.3.1 à 3.3.4.

L'approche du concepteur, pour la conduite d'une de ces étapes, est détaillée dans le paragraphe 3.4.

Les deux premières étapes sont détaillées dans les paragraphes 3.5. et 3.6. sur la base de l'exemple industriel présenté au début de ce mémoire.

Nous nous efforcerons de mettre en évidence les parties où devront intervenir les deux dernières étapes d'évaluation sans toutefois les détailler.

Enfin nous terminerons par un bilan de l'apport de cette méthodologie.

3.3 Présentation générale des étapes de la méthodologie de conception

3.3.1. Analyse du Système de Fabrication

L'originalité de la méthodologie que nous proposons réside dans la prise en compte des besoins du client et du concepteur. Elle garantit leur cohérence.

Elle répond aux besoins des bureaux des méthodes en les aidant à structurer leur approche face à des projets de plus en plus complexes de par leur taille et leur variété.

Elle fournit un support pour une gestion correcte des risques économiques et techniques qui sont pris.

L'étude des besoins est facilitée par la mise en place d'une technique d'analyse et de détermination des informations sous la forme d'un GUIDE CONCEPTEUR.

Nous nous sommes inspirés des publications de [BLANCHARD-78], concernant la normalisation d'un cahier des charges.

Dans un premier temps, ce GUIDE recommande au concepteur d'utiliser une liste de thèmes structurés selon plusieurs niveaux de détails. Cette recherche systématique des informations garantit la prise en compte de tous les aspects des problèmes d'automatisation (complétude des spécifications).

Dans un deuxième temps, le GUIDE CONCEPTEUR identifie des contraintes de cohérence à respecter entre les différentes informations d'un cahier des charges.

Le concepteur dispose, donc, à travers ce GUIDE, d'un réseau d'informations à connaître pour l'établissement d'un cahier des charges du client ou du constructeur.

Les contraintes de cohérence entre ces différentes informations lui sont également communiquées. On les désignera par la suite sous le nom de règles de cohérence "sémantique".

Par exemple, le concepteur doit toujours vérifier que les souhaits du client concernant la flexibilité du système de fabrication à réaliser restent cohérents avec le nombre de variantes du produit que ce système devra traiter.

Le Guide Concepteur formalise cette démarche en proposant un thème et une règle de cohérence associée.

thème : "Les critères de flexibilité"

règle : "consulter le nombre de variantes du produit à traiter"

En conclusion, le concepteur est capable d'écrire très vite un cahier des charges en garantissant la complétude et la cohérence des spécifications écrites.

Une autre retombée concrète est une nette amélioration du temps d'étude des cahiers des charges soumis par les clients lors d'un appel d'offres.

Le concepteur détecte plus rapidement l'incompatibilité des différents besoins du client.

3.3.2. Définition des Solutions Techniques

Après l'Analyse du Système de Fabrication, le concepteur doit définir des solutions techniques.

Celui-ci a souvent du mal à faire abstraction de son savoir-faire. Il a tendance à vouloir réutiliser les solutions préconisées auparavant sans étudier les spécificités de son avant-projet à travers une analyse fonctionnelle approfondie.

La méthodologie que nous développons en mettant en place une suite d'étapes pour la définition d'une solution technique oblige le concepteur à plus de rigueur et de méthode dans son approche de la solution finale.

Les étapes recommandées lors de la définition des solutions techniques sont les suivantes :

- une analyse fonctionnelle,
- le choix des solutions techniques sur la base des équipements imposés et de l'expérience du concepteur,
- des spécifications fonctionnelles,
- des spécifications opérationnelles.

La proposition de formalismes différents pour chaque étape de définition de la solution permet au concepteur de bien les différencier.

Par exemple, il arrive souvent que des spécifications fonctionnelles tiennent lieu d'analyse fonctionnelle dans la mesure où le concepteur a déjà une solution en tête.

L'utilisation de la méthodologie que nous avons développé, évite ce type de confusion.

3.3.3. Evaluation Economique

Il existe actuellement plusieurs méthodes d'évaluation économique des systèmes de fabrication [ANCELIN-87, FRAY-87].

Notre propos n'est pas de développer une nouvelle méthode d'évaluation économique mais de rappeler brièvement quelles en sont les étapes principales.

Une estimation rapide des paramètres économiques engagés dans chaque proposition doit être effectuée à partir des coûts de chaque équipement, de la maintenance, des coûts d'exploitation et de l'amortissement escompté.

Pour cela, les premières spécifications techniques des équipements doivent être définies (par exemple dans le contexte robotique le nombre d'axes du robot, ses degrés de liberté, ...)

Les solutions dont le coût dépasse l'enveloppe financière imposée par le client sont abandonnées.

3.3.4. Evaluation de Faisabilité Technique

L'étude de faisabilité intervient après une évaluation économique.

Elle s'intéresse au comportement dynamique et aux performances des solutions retenues. Les configurations proposées sont évaluées essentiellement sur la valeur du temps de cycle et des performances de sortie.

Les problèmes de collisions, de conflits ou d'interminations ainsi que la faisabilité de l'implantation physique du système de fabrication dans l'atelier sont également étudiés.

Dans certains cas critiques, un prototype peut être réalisé, notamment lorsque l'on souhaite vérifier la capacité de préhension d'un outil.

A la fin de cette étape, les ingénieurs concepteurs sont à même de proposer une solution au client.

Cette solution est définie selon des critères de performances et de coût.

3.4. Démarche du concepteur : synthèse de la pré-étude d'un système de fabrication

La pré-étude d'un système de fabrication repose sur la connaissance des différentes étapes d'un avant-projet qui sont : l'analyse du système de fabrication, la définition des solutions et leur évaluation économique et technique.

Les outils de travail du concepteur sont les suivants :

- le MODELE DE CYCLE DE VIE de Mac DERMID,
- le GUIDE CONCEPTEUR,
- le système EPOS.

Le MODELE DE CYCLE DE VIE de Mac Dermid structure l'approche du concepteur lors d'une étape de pré-étude en quatre activités :

- une transformation ou activité de conception,
- une représentation ou écriture et documentation des spécifications,
- une vérification syntaxique des spécifications,
- un contrôle pour la validation de l'étape de spécification.

Le GUIDE CONCEPTEUR, document de base de la méthodologie de conception que nous avons développé, propose une technique d'investigation et d'évaluation des informations liées au projet.

Il représente le savoir-faire des concepteurs pour la conduite d'une étape d'avant-projet.

Le système EPOS est à l'origine des modèles de spécification recommandés dans le chapitre 2.

Il offre un support informatique pour l'analyse syntaxique et la documentation des spécifications.

La démarche et les outils dont dispose le concepteur sont présentés sur le diagramme page suivante.

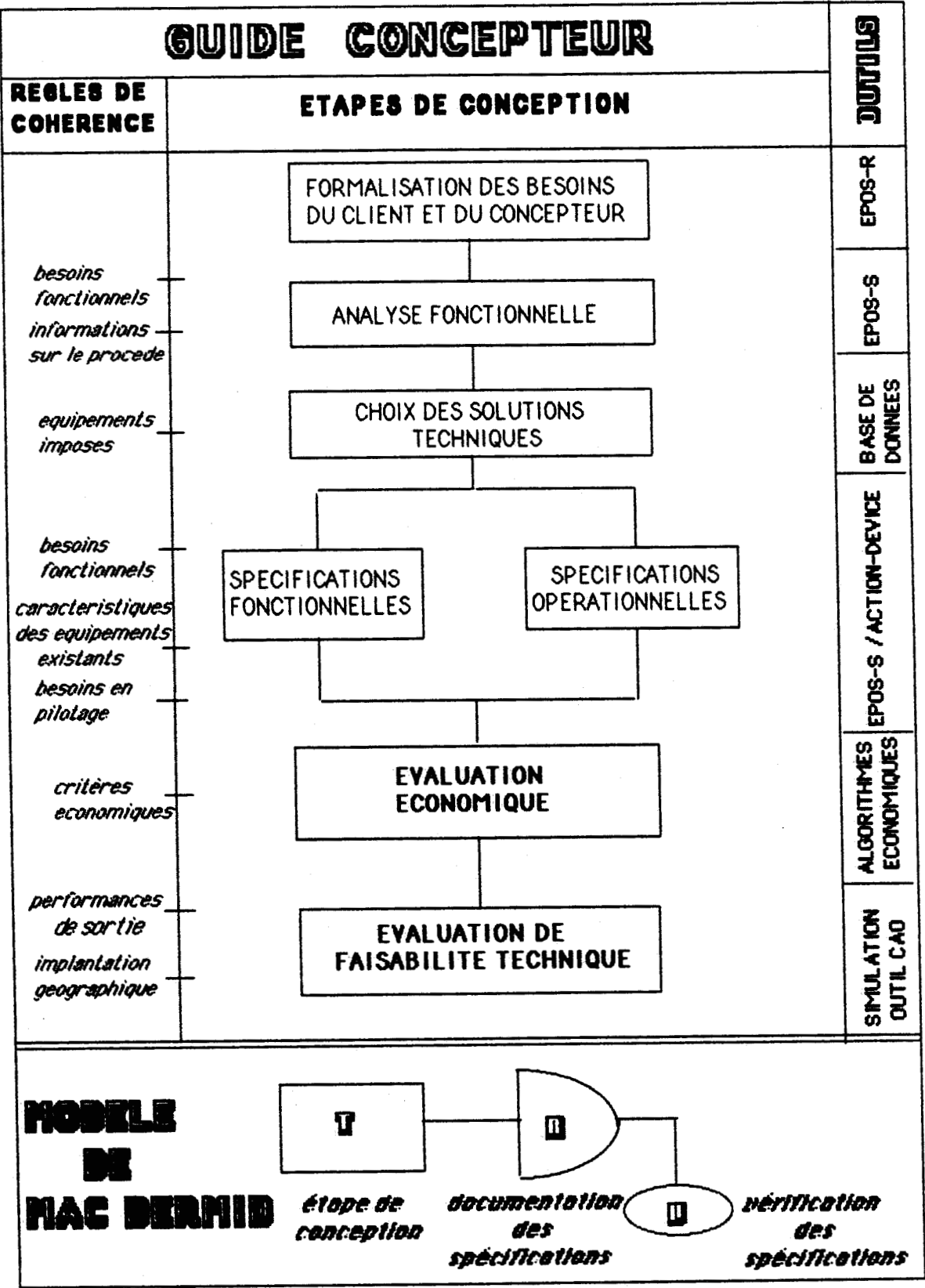


Fig 24 Démarche du concepteur pour la pré-étude de systèmes de fabrication

3.5. Analyse du Système de Fabrication

L' Analyse du Système de Fabrication se décompose en trois étapes.

Dans un premier temps, le concepteur procède à la spécification exhaustive des besoins du client.

Ensuite, il définit les contraintes dues à son savoir-faire, ou celles induites par les constructeurs et fournisseurs.

Enfin, il met en place un glossaire du projet.

3.5.1. GUIDE CONCEPTEUR pour l'écriture du cahier des charges client

Dans le cadre de notre exemple, nous avons été amené à rédiger un cahier des charges du client sur la base du GUIDE CONCEPTEUR.

Ce cahier des charges définit les besoins du client selon 9 thèmes d'informations :

(i) Les objectifs de l'automatisation

Le contexte de l'avant-projet, les contraintes techniques ou économiques du client ainsi que le procédé de fabrication existant sont décrits rapidement.

(ii) Les performances requises

Les fonctionnalités souhaitées, ainsi que les performances de sortie (cadence, temps de cycle) attendues sont définies dans ce chapitre.

(iii) Les données techniques

D'après les informations du client, le concepteur spécifie, de manière exhaustive, le procédé de fabrication existant ainsi que le produit à traiter par le système de fabrication.

(iv) Les contraintes client

Le concepteur prend connaissance des matériels, logiciels ou standards imposés par le client.

(v) Les contraintes opérationnelles

Le concepteur décrit l'environnement et le site d'accueil du futur système de fabrication.

(vi) Les critères de flexibilité

Le concepteur définit les souhaits du client, concernant l'évolution du système de fabrication.

(vii) Les besoins au niveau d'un système de commande

Le concepteur spécifie tous les besoins nécessitant la mise en place d'un système de commande associé au système de

fabrication.

(viii) Les contraintes économiques

Les contraintes économiques (coût, amortissement) sont détaillées dans ce chapitre.

(ix) Les besoins en contrôle-qualité

Le concepteur détaille les contraintes de tolérance, les procédures de recyclage des pièces défectueuses, ou encore le type de document de contrôle qualité à produire.

3.5.2. Illustration du GUIDE CONCEPTEUR sur la base de l'exemple industriel

Un extrait du cahier des charges est fourni pages suivantes.

Dans cet extrait, les performances fonctionnelles, les interfaces avec la production, les contraintes du client et les contraintes opérationnelles sont définies.

Le formalisme utilisé est le modèle de spécification des besoins recommandé dans le chapitre 2.

La documentation est générée automatiquement par le système EPOS. On peut constater qu'elle reste accessible aux lecteurs.

:	:	:	:
:RENAULT AUTOMATION	: REQUIRED PERFORMANCES	: PAGE: F 2-1	:
:	:	:	:
:PROJET :MPRDOUAI	:	: DATE: 5-14-86:	:
:UTILISATEUR:PLANNING	:	:	:
:	:	:	:

2. PERFORMANCES REQUISES

2.1. PERFORMANCES FONCTIONNELLES

2.1.1. CRITERES PRINCIPAUX

REQUIREMENT 2110 (0) : <impératif fonctionnel>

Le chargement et déchargement des pare-brise, ainsi que leur évacuation doivent être pris en charge par le système de fabrication.

REQUIREMENT 2111 (0) : <impératif fonctionnel>

Le pare-brise doit être tenu lors de la pose du joint-mousse.

2.1.2. SOUHAITS

REQUIREMENT 2120 (1) : <critère de moindre importance>

Il faudrait prévoir un système de convoyage des pare-brise jusqu'à la faredeleuse.

REQUIREMENT 2121 (1) : <limite du processus automatisé>

Le chargement et le déchargement des pare-brise doit se faire sans apporter des modifications aux plateaux tournants existants.

TERM plateau tournant

2.2. PERFORMANCES DE SORTIE

REQUIREMENT 2200 (0) : <cadence>

Le système de fabrication doit être capable de traiter 500 pare-brise par jour.

...

:	:	:	:
:RENAULT AUTOMATION	: DONNEES TECHNIQUES	: PAGE: F 3-3	:
:	:	:	:
:PROJET :MPRDOUAI	:	: DATE: 5-14-86:	:
:UTILISATEUR:PLANNING	:	:	:
:	:	:	:

3.1.3. INTERFACES AVEC LA PRODUCTION

CONSTRAINT 3130 (0) : <interface mécanique>
 Les pare-brise arrivent par containers, toutes les dix minutes,
 en présentant leur face externe.

3.2. DESCRIPTION DU PRODUIT

Nom du produit : pare-brise. Il y a vingt-sept types de pare-brise différents à traiter. Il est prévu d'avoir des modèles plus grands pour autobus. Ils pèseront 20 kilogrammes.

CONSTRAINT 3200 (0) : <poids>
 Un pare-brise pèse environ 10 kilogrammes.

CONSTRAINT 3201 (0) : <taille>
 Un pare-brise a les dimensions suivantes : 2 * 1.5 (mètres)

CONSTRAINT 3202 (0) : <forme>
 Un pare-brise a une forme bombée.

4.2. STANDARDS, METHODES ET DOCUMENTATION IMPOSES

CONSTRAINT 4200 (0) : <standard de documentation>
 La proposition d'avant-projet devra être rédigée sur un traitement de texte compatible IBM.

4.3. CONTRAINTES PERSONNEL

CONSTRAINT 4300 (1) : <nombre de personnes après automatisation>
 Après automatisation, 2 personnes seront nécessaires pour charger et décharger les containers. Une autre personne devra assurer la surveillance du système de fabrication.

CONSTRAINT 4301 (0) : <qualification du personnel>
 Niveau de qualification requis :deux ouvriers, un technicien.

:	:	:	:
:RENAULT AUTOMATION	:CONTRAINTES OPERATIONNELLES:	PAGE: F 5-1	:
:	:	:	:
:PROJET :MPRDOUAI	:	:DATE:5-14-86:	:
:UTILISATEUR:PLANNING	:	:	:
:	:	:	:

5. CONTRAINTES OPERATIONNELLES

5.1. ENVIRONNEMENT

CONSTRAINT 5100 (0) : <environnement naturel>

La température dans l'atelier est de 60 degrés Celsius. Il y a par ailleurs un éclairage très fort.

...

3.5.3. GUIDE CONCEPTEUR pour la spécification des besoins du concepteur

La finalité de toute pré-étude est de proposer au client des solutions standards et fiables.

Le concepteur a alors souvent tendance à préconiser des solutions toutes faites ou qu'il a déjà éprouvées sur des projets antérieurs.

L'objectif du GUIDE CONCEPTEUR est d'éviter cet écueil en l'aidant à répertorier les risques techniques et économiques de son avant-projet.

Pour ce faire, la méthodologie développée propose une analyse descendante et structurée.

Le concepteur doit savoir répondre à une liste de questions avant de concevoir le système de fabrication.

Ces questions sont les suivantes :

(i) Quels sont les objectifs de conception ?

Le concepteur doit définir la stratégie de production pour le système de fabrication.

Il doit préciser notamment quels sont les facteurs de conception pris en compte : un temps de circulation des pièces (flux des pièces) optimisé, un engagement maximum des machines ou encore la disponibilité et l'interchangeabilité des outillages.

(ii) Quels sont les choix conceptuels ?

Une première approche globale doit permettre au concepteur de définir le degré d'automatisation du procédé et son intégration dans un système plus large.

Les opérations amont et aval ainsi qu'une estimation du nombre d'opérateurs manuels doivent être définies.

Les facilités d'accès prévues pour la maintenance des équipements, la gestion des outillages ainsi que l'introduction de terminaux opérateurs doivent être également spécifiées.

Pour une étude plus détaillée, les choix conceptuels dépendent de la compétence et du savoir-faire du concepteur.

S'il s'agit d'un constructeur de robot, il aura tendance à préconiser dès le début un choix de robot.
Dans le cas d'un constructeur informatique, il fournira la base du matériel informatique qui lui semble indispensable.

(iii) Critères de choix pour les fournisseurs

Le concepteur établit un planning à long-terme du projet, dans l'hypothèse d'un accord contractuel.
Des propositions pour les échelonnements de paiement, les dates de révision du projet ainsi que la recette de l'installation et le plan de formation du personnel sont définis.

3.5.4. Illustration du GUIDE CONCEPTEUR sur la base de l'exemple industriel

Dans le cadre de notre exemple, le choix du robot a été défini sur la base du cahier des charges du client en prenant en compte les informations de :

- 1 - Contraintes au niveau de la saisie des pièces (taille, forme poids, nombre de variantes) :

REQUIREMENT 3200 (0) <poids>

Les pare-brise à manipuler pèsent 10 kilogrammes.

REQUIREMENT 3201 (0) <taille>

Les pare-brise à manipuler ont une taille de 2*1,5 mètres.

- 2 - Contraintes opérationnelles pour un bon fonctionnement du robot :

CONSTRAINT 5100 (0) <environnement naturel>

La température ambiante est de 60 degrés Celsius. Il y a un fort éclairage.

- 3 - Performances (vitesse, cadence, ...) attendues :

REQUIREMENT 2200 (0) <cadence>

Les cadences à respecter sont de 500 pare-brise par jour.

Ces différentes contraintes définissent le profil du robot idéal pour le système de conditionnement de pare-brise.

La température de fonctionnement de ce robot idéal doit être supérieure à 60 degrés. Sa capacité de charge doit être au moins de dix kilogrammes.

Sur la base de ces paramètres caractéristiques, nous avons étudié l'ensemble des robots réels dont nous disposons.

La comparaison entre leurs spécifications et celles du robot idéal nous a conduit à la sélection du robot X58.

3.5.5. Définition de l'environnement lié à l'application

La spécification des besoins du client ou du constructeur sont définis sur la base d'une technologie et d'un savoir-faire.

La mise en place d'un glossaire de termes est nécessaire pour assurer une bonne compréhension entre les différents partenaires d'un avant-projet.

La définition d'un tel glossaire est obtenu après l'analyse des spécifications rédigées et la description des termes ou abréviations qui y sont employées.

Le formalisme utilisé a été décrit dans le paragraphe 2.4. : modèle de spécification des besoins.

Il s'agit de l'élément formel TERM suivi du nom du terme à définir et de sa définition en langage naturel.

3.6. Définition des Solutions Techniques

Dans ce paragraphe, nous présentons en détail, les différentes étapes de la pré-étude, qui amènent le concepteur à la définition d'une solution technique.

3.6.1. Analyse Fonctionnelle Générale

Cette première étape consiste à identifier les fonctionnalités du futur système de fabrication.

Le concepteur procède selon une démarche hiérarchisée, descendante et modulaire.

Pour chaque niveau d'analyse, le GUIDE CONCEPTEUR recommande une technique d'analyse des besoins du client pour l'identification des fonctions générales du futur système de fabrication.

La solution est alors obtenue par un processus de réflexion en deux phases :

- décomposition en fonctions,
- construction de la structure hiérarchique et de la représentation fonctionnelle à partir d'actions-modules.

La décomposition fonctionnelle doit être effectuée en évaluant l'ensemble des fonctions qui permettent d'assurer le déroulement des opérations souhaitées.

Cette décomposition ne doit pas être nécessairement calquée sur le procédé de réalisation.

Le niveau de départ de la construction peut être quelconque. Il correspond à la première approche obtenue à partir de la spécification des besoins fonctionnels.

La décomposition de chaque fonction et sa représentation sous la forme d'ACTION-MODULE doit être poursuivie tant que des choix technologiques ne s'avèrent pas nécessaires pour poursuivre la décomposition.

Dans le cadre de notre exemple, l'analyse des besoins fonctionnels pour le conditionnement d'un pare-brise, et l'utilisation du GUIDE CONCEPTEUR nous a conduit à identifier la structure fonctionnelle suivante :

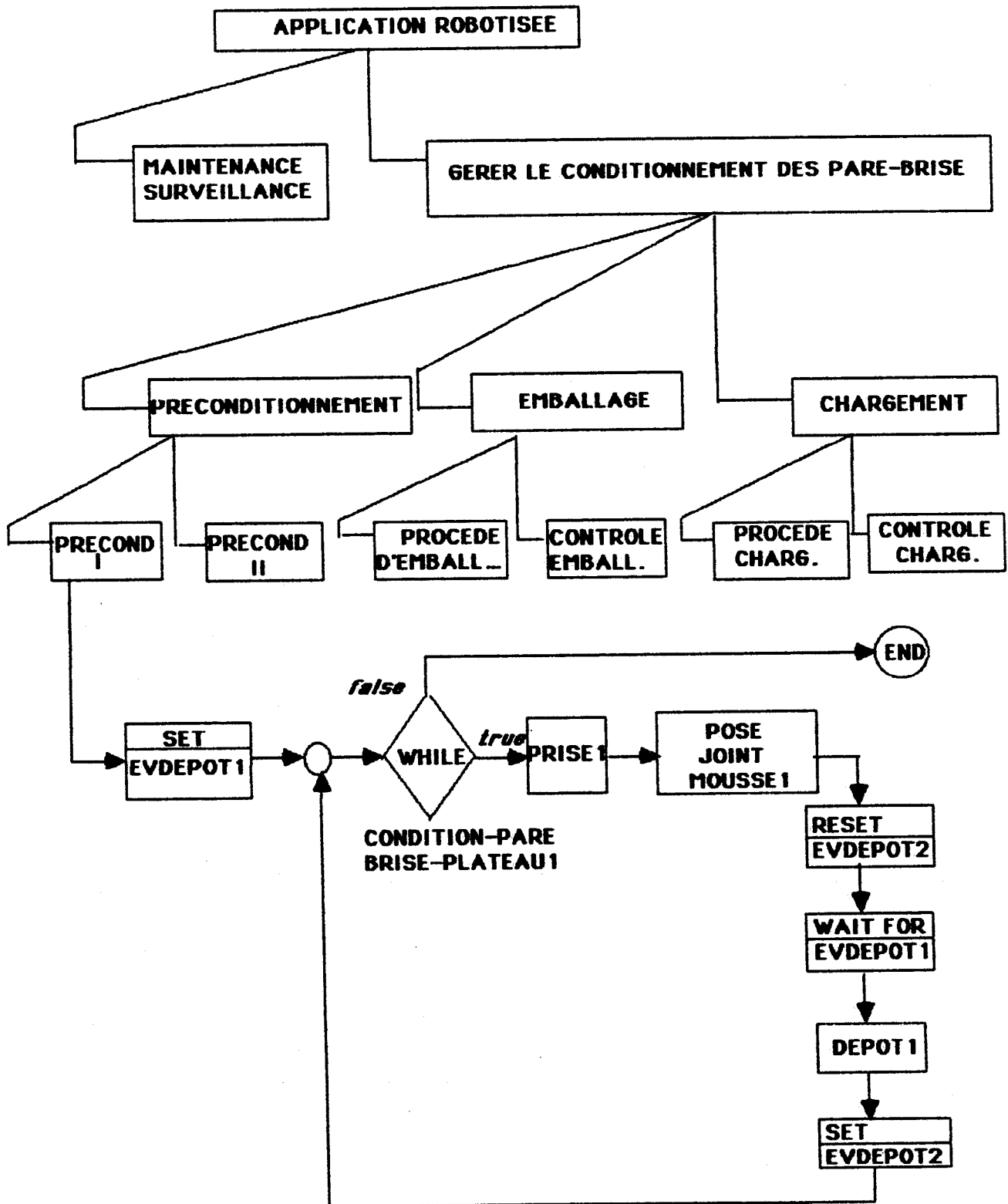


Fig 25 Diagramme hiérarchique

Trois fonctions principales ont été identifiées à un premier niveau de décomposition :

- une fonction de préconditionnement,
- une fonction d'emballage,
- et, une fonction de déchargement.

Ces trois fonctions ont été spécifiées, selon le formalisme d'un action module.

Un diagramme de flux de données nous permet de visualiser l'échange de données entre ces fonctions :

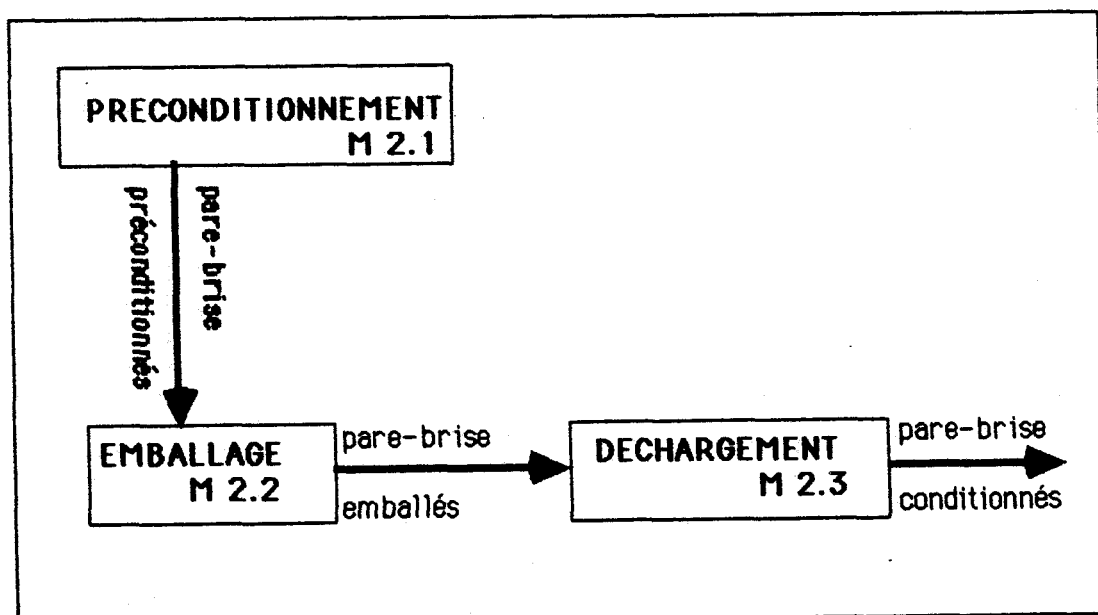


Fig 26 Diagramme de flux de données

3.6.2. Choix des solutions techniques

Dès la spécification des besoins, certains choix technologiques ont été faits a priori.

Ceci, pour deux raisons essentielles : le client désire réutiliser une partie de son matériel ou bien l'expérience du concepteur impose certaines options techniques.

La définition complète d'une solution s'obtient :

- en établissant la liste des équipements nécessaires pour assurer la réalisation des fonctions identifiées lors de l'étape précédente,
- en réutilisant les équipements déjà existants ou en suivant les choix préconisés par le constructeur,
- en définissant leur implantation géographique.

3.6.2.1. Choix des équipements

Le choix des équipements consiste à sélectionner, dans un premier temps, différents types et marques de machines pour la réalisation des opérations de traitement du produit.

Ce choix est souvent imposé. Par contre, la définition de l'outillage de chacune des machines est à la charge du concepteur.

Deux alternatives sont possibles : soit définir des machines dédiées à une opération spécifique, soit avoir des machines équipées d'un outillage standard.

Le choix de l'une de ces alternatives dépend du type des pièces et des opérations à réaliser ainsi que du nombre de variantes à traiter.

Si le nombre des variantes est important, une gamme opératoire commune est définie aussi précisément que possible pour obtenir un nombre minimum d'opérations spécifiques et donc de machines spécifiques.

Ensuite, le choix d'un système de transport et de la stratégie de transport est à faire en fonction des contraintes de poids, de volume des pièces et de leur trajet dans l'atelier.

Enfin, le choix des équipements informatiques dépend des habitudes du client et des standards industriels.

Le milieu ambiant (atelier ou salle climatisée), le volume des acquisitions, le volume nécessaire pour le traitement des données et leur stockage sont autant de critères d'évaluation.

Dans le cadre de notre exemple, la faredeleuse était imposée.

Du fait de la simplicité de la fonction de transport (acheminement des pare-brise à partir des postes de préconditionnement vers la faredeleuse et évacuation des pare-brise), la solution du tapis roulant a été retenue.

Le choix du robot X58 a été défini précédemment.

Enfin, le volume des informations ne nécessitent pas un système de commande spécifique.

3.6.2.2. Configurations proposées

Les configurations doivent être définies dans un souci d'optimisation du trajet des pièces. L'implantation des machines dépend donc des opérations qu'elles réalisent.

Dans le cadre de notre exemple, tous les équipements réalisant l'une des trois fonctions : préconditionnement, emballage et déchargement ont été regroupés.

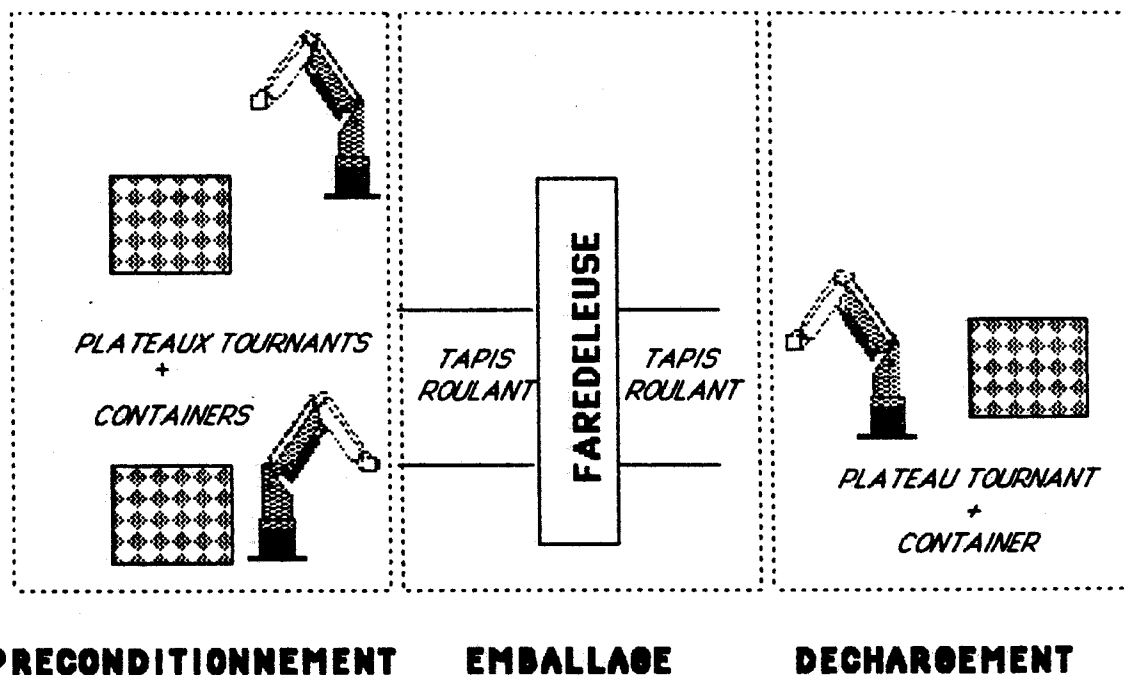


Fig 27 Configuration retenue pour le système de conditionnement des pare-brise

3.6.3. Spécifications fonctionnelles

Les spécifications fonctionnelles décrivent les différentes fonctions réalisées par chacun des équipements du système de fabrication.

Les contraintes de synchronisation entre ces différentes fonctions sont également prises en compte.

Les dernières fonctions identifiées lors de l'analyse fonctionnelle sous la forme d'ACTION-MODULE(s) sont le point de départ des spécifications fonctionnelles.

Les spécifications opérationnelles (définition des équipements) sont menées en parallèle.

Les fonctions sont formalisées en terme d'ACTION(s), dont le masque de spécification décrit les échanges entre fonctions, les contraintes de synchronisation et le codage pour les fonctions les plus fines dans un langage élémentaire (langage robot ou code équipement).

La décomposition est limitée à deux ou trois niveaux de détail dans le cadre d'un avant-projet.

A chaque niveau, et, pour chaque fonction, les besoins du cahier des charges, qui sont satisfaits, sont répertoriés sous l'entité formelle FULFILS.

Pour la réalisation d'un système de commande, le concepteur doit identifier les fonctions logicielles fournies par chaque équipement. Il s'efforcera de mettre en valeur les aspects standards d'un tel développement.

Dans le cadre de notre exemple, la présence de deux robots en entrée introduit des contraintes de synchronisation, que l'on peut identifier sur le réseau de Pétri ordinaire page suivante.

Les deux fonctions de préconditionnement (PRECONDITIONNEMENT-1 et PRECONDITIONNEMENT-2) sont décrites.

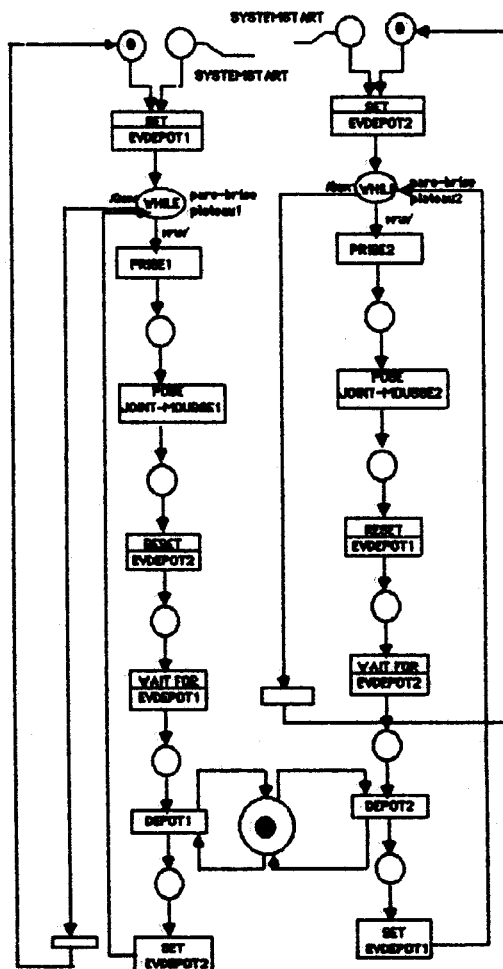


Fig 28 Réseaux de Pétri ordinaires

3.6.4. Spécifications opérationnelles

Les équipements informatiques et mécaniques sont décrits selon leurs performances de fonctionnement, les principales caractéristiques géométriques des différents équipements et leurs spécificités.

Cette description sera formalisée selon le masque de spécification DEVICE que nous avons adapté à ce type de description.

Pour chaque équipement décrit, le GUIDE CONCEPTEUR recommande la vérification des contraintes d'installation, des contraintes liées à la pièce manipulée.

Il met également en évidence le lien entre les informations (caractéristiques géométriques, performances) listées sous l'élément formel : FEATURES, et des instructions géométriques pour la modélisation de l'équipement.

Le concepteur produit une documentation des spécifications opérationnelles.

Dans le cadre de la mise en place d'un système de commande du système de fabrication, il adjoint une fiche synthétique où sont consignés :

- le nombre et les références des ordinateurs fournis ainsi que des périphériques associés,
- le détail des équipements de sauvegarde.

Dans le cadre de notre exemple, le robot X58 a été retenu. Il a été spécifié sous la forme d'un DEVICE dans le paragraphe 2.5.5.

3.6.5. Vérifications des contraintes de réalisation

Les solutions définies, et, décrites de manière statique à partir de leurs fonctions de base et de leurs équipements, sont validées dans le cadre d'un avant-projet selon cinq critères :

- 1 - le degré de flexibilité de la solution retenue,
- 2 - le temps de cycle,
- 3 - les facilités d'accès aux points-clefs de la cellule par le robot, dans le cadre d'une application robotisée
- 4 - l'évaluation des performances du système de commande,
- 5 - le fonctionnement en modes dégradés et le traitement des pannes.

3.6.5.1. Degré de flexibilité

Le concepteur doit étudier deux aspects :

- Quel est le degré de flexibilité du système de fabrication ?
- Quel devrait être le degré de flexibilité de ce système compte-tenu du cahier des charges ?

L'évaluation du degré de flexibilité du système de fabrication est basé sur :

- le type de lots à traiter et la période de traitement (flexibilité statique du procédé)
- le nombre de variantes à traiter en continu à partir d'un profil type (flexibilité dynamique du procédé),
- l'interchangeabilité des outils et des machines lors d'une panne, (flexibilité de l'outil),
- le degré d'adaptation des équipements aux évolutions de la ligne du produit (au niveau formes, poids et opérations à réaliser).

Suivant le type de produit à traiter (variantes, poids, formes, opérations requises et évolutions futures probables), le système de fabrication doit être plus ou moins flexible.

Dans le cadre de notre exemple, le nombre de variantes du produit est élevé (27 types de pare-brise), une grande flexibilité du procédé et des outils est donc nécessaire.

Par contre, si le produit avait eu un long cycle de vie sur le marché et un petit nombre de variantes, il aurait fallu un degré d'adaptation de l'équipement élevé.

C'est au concepteur qu'il incombe d'évaluer la flexibilité requise suivant les besoins répertoriés, et, si la ou les solutions envisagées offrent un tel type de flexibilité.

Le degré de flexibilité du système de commande est étroitement lié au système de fabrication réalisé.

Il s'estime en fonction des facilités de reconfiguration d'un tel système et de la modularité du logiciel développé.

Les capacités d'évolution du matériel informatique retenu, et, du logiciel de commande des machines unitaires, la solution retenue pour la transmission des données (réseau ou point-à-point) sont également des facteurs de flexibilité.

Une structure de réseau tend à banaliser les transferts de données sans nécessairement privilégier un "maître" des échanges qui pourrait à terme constituer un goulot d'étranglement pour l'échange des données.

Des critères mécaniques comme les évolutions futures du produit et du nombre de variantes à traiter vont influencer l'évolution du parc-machine, et donc le volume d'informations à traiter ainsi que les connexions au niveau de l'atelier.

Une solution de transmission des données par réseau est, alors, plus flexible que du point-à-point dans le cas d'un agrandissement du parc-machine.

3.6.5.2. Temps de cycle

L'emploi de machines physiques suppose des contraintes d'interfaçage de ces machines avec le milieu extérieur.

La principale de ces contraintes est le temps de cycle nécessaire pour la réalisation d'un produit.

Il est évalué, d'après le temps imposé par les machines pour effectuer leurs opérations, mais aussi, par l'amont et l'aval du processus.

Sa valeur est déterminante dans le choix d'une solution. Elle doit être évaluée le plus précisément possible avant la définition complète d'une solution.

Le but du GUIDE CONCEPTEUR, dans ce cadre, est de rappeler les performances souhaitées par le client et qui seront les critères de choix d'une solution.

Un outil CAO évalue le temps de cycle des applications

robotiques. Un outil de simulation de type SLAM permettra la modélisation des machines et plus globalement des performances du système de fabrication.

3.6.5.3. Facilités d'accès

Dans le cas d'une cellule robotisée, le concepteur doit pouvoir garantir des facilités d'accès à tous les points-clefs de la cellule conçue.

Cette évaluation repose sur l'adéquation de l'enveloppe de travail des robots et des opérations prévues comme le chargement/déchargement d'une ou plusieurs machines.

Ce type de vérification se fait facilement lorsque l'on a recours à un outil CAO.

3.6.5.4. Evaluation des performances du système de commande

Le concepteur doit arriver à un taux de disponibilité de 80%, pour pouvoir garantir de bonnes performances du système de fabrication en phase d'exploitation.

Ce taux de disponibilité est évalué en fonction de la charge du système (nombre, type, longueur des informations) et de la fréquence des échanges.

Les temps d'acquisition, la capacité de traitement des données, et, la taille mémoire du système de commande associé doivent être estimés.

3.6.5.5. Fonctionnement en modes dégradés

Le fonctionnement en mode dégradé constitue un point délicat de la définition d'un système de commande. Une analyse descendante est recommandée.

Le but du GUIDE CONCEPTEUR, dans ce cas, est de fournir au concepteur le moyen d'identifier et de décrire les configurations de panne et les modes dégradés associés.

Pour ce faire, l'analyse du concepteur est guidée par une liste de questions de référence auxquelles il doit répondre.

Dans le cas d'une seule machine en panne, le concepteur doit envisager un fonctionnement en mode dégradé sur la base des questions suivantes :

- (a) S'agit-il d'une machine spéciale ?
- (b) Dispose-t-elle d'un outillage spécifique que n'ont pas les autres machines de l'atelier ?

- (c) Y- a-t'il des opérateurs chargés de la surveillance ?
- (d) Quels sont les stocks prévus pour les outillages ?
- (e) Quel est le volume de pièces qui devait être traité par la machine ?
- (f) Quelles sont les possibilités de gestion de stocks tampons ?
- (g) Quelles sont les contraintes au niveau du système de commande pour un déroutage systématique des pièces ?
- (h) La génération de gammes de déroutage pour une machine est-elle possible sans conduire à une explosion combinatoire ?
- (i) Quelles sont les capacités d'optimisation du système de commande pour un déroutage systématique des pièces ?

Prendre en compte le cas de plusieurs machines en panne simultanément et définir un fonctionnement en mode dégradé est en général très compliqué (explosion combinatoire !).

Cela n'a pas été envisagé dans ce travail.

3.6.5.6. Traitement des pannes

Le concepteur doit étudier différentes procédures de redémarrage possibles en cas de panne du système de commande.

Traditionnellement, les procédures de redémarrage sont de deux types : des démarrages à chaud, ou, des démarrages à froid.

Avant de choisir l'une de ces alternative, le GUIDE CONCEPTEUR recommande, d'abord, de se poser les questions suivantes :

- Quel est le coût de la solution ?
- Quel doit être le niveau de restitution des informations ?
- Quel est le coût de la mise en oeuvre ?
- Quelle sera la charge du réseau (dans le cas d'un couplage machine-ordinateur de type réseau) ?

3.6.6. Spécifications détaillées

Les spécifications détaillées ont pour objectif d'affiner les spécifications fonctionnelles sur des points critiques qui sont apparus lors de l'évaluation technique.

Il peut s'agir de problèmes de synchronisation ou d'agencement spatial.

A ce stade, certaines solutions techniques peuvent être définitivement écartées dans la mesure où elles ne satisfont pas des besoins du client.

Pour les solutions restantes, le concepteur procédera à une estimation économique. Il devra prendre en compte notamment :

- le coût de gestion des stocks tampons,
- le coût de chaque équipement,
- l'installation sur site,
- la documentation,
- la formation,
- la maintenance.

Pour l'évaluation du système de commande associé interviendront également :

- le coût des logiciels standards (licence, droit de recopie),
- le coût du développement de logiciels spécifiques (H/MOIS)

Les solutions dont le coût total dépassera l'enveloppe financière prévue par le client seront abandonnées ou repensées.

Le concepteur est alors à même de proposer une ou éventuellement deux solutions techniques. Elles seront détaillées afin de constituer un dossier complet d'avant-projet.

3.6.7. Vérifications des spécifications

A chaque étape de la définition du système de conditionnement des pare-brise, le concepteur a été amené à vérifier les spécifications rédigées.

Ces vérifications au cours de la pré-étude d'un système de fabrication sont indispensables pour éviter une trop grande divergence du projet par rapport aux objectifs initiaux de l'automatisation.

Néanmoins, avant de faire une proposition ferme au client, le concepteur doit procéder à une dernière vérification.

Elle consiste à étudier si tous les besoins exprimés par le client ou le concepteur ont été pris en compte à un niveau quelconque de la spécification.

Cette vérification croisée, en fin de spécification, est recommandée par toutes les méthodologies existantes.

L'utilisation d'un formalisme de spécification permet un traitement informatique des vérifications croisées entre le cahier des charges et les spécifications du système automatisé.

Ce traitement s'effectue sur la base d'une comparaison systématique entre tous les besoins répertoriés dans le cahier des charges et ceux listés sous l'élément FULFILS de toutes les spécifications de la solution technique.

L'utilisation du système EPOS comme support de la méthodologie et le traitement informatique qu'il propose apporte donc un plus indéniable dans la mise en oeuvre des vérifications croisées.

Les besoins qui ne figurent à aucun niveau de la spécification de la solution technique n'ont pas été pris en compte.

Le concepteur a alors deux alternatives : affiner les spécifications de la solution pour prendre en compte ces besoins ou bien présenter la solution telle quelle avec des dérogations par rapport au cahier des charges du client initial.

3.7. Conclusion

A partir du point de vue analytique offert par le modèle conceptuel, nous avons proposé une méthodologie intégrée pour la conception d'un système de fabrication. Celle-ci est présentée page 102.

Sur la base des besoins du client, elle permet d'aboutir à la définition complète d'une solution, des fonctions remplies et des équipements retenus, structurant la pré-étude d'un système de fabrication en quatre étapes :

- l'Analyse du Système de Fabrication,
- la Définition des Solutions Techniques,
- l'Evaluation Economique,
- l'Evaluation de Faisabilité Technique.

Les deux dernières étapes permettent d'évaluer les solutions proposées.

Dans le cadre d'un GUIDE CONCEPTEUR, des conseils et des règles de cohérence, à respecter pour l'écriture des spécifications sont définis.

La méthodologie reste toutefois ouverte à l'introduction de règles supplémentaires, induites par le savoir-faire du concepteur.

Cette méthodologie est indispensable pour des réalisations importantes, car elle permet de réduire la recherche de solutions par intuition.

Elle justifie rationnellement les choix retenus par une validation de la solution finale en fonction des besoins du client et du concepteur.

Elle permet une conception progressive, en différenciant très nettement les phases de spécification des besoins, d'analyse fonctionnelle, de spécification fonctionnelle et opérationnelle.

Chacune de ces phases fait l'objet d'une documentation claire, précise et structurée.

Par conséquent, elle facilite la mise en place d'une organisation de l'étude où interviennent simultanément plusieurs groupes de travail.

L'ensemble de la documentation retrace l'historique de l'étude, ce qui facilite par la suite l'évolution ou l'adaptation du système de fabrication aux besoins du client.

Par ailleurs, le support informatique existant (le système EPOS) sur lequel s'appuie la méthodologie montre que la définition complète et rigoureuse d'une solution technique peut se faire avec des langages de haut-niveau.

La pérennité de la méthodologie est une caractéristique difficile à évaluer.

Néanmoins son indépendance à la fois par rapport au support informatique utilisé (le système EPOS), et, à la technologie de réalisation offre certaines garanties.

L'intérêt d'une telle méthodologie a été évaluée sur des avant-projets industriels de taille importante. Pour de petites applications, son utilisation est moins critique.

Nous avons pu constater la réelle efficacité du GUIDE CONCEPTEUR, lors de la définition des besoins du client et du concepteur, notamment dans la phase de validation des spécifications (cohérence et complétude).

Les résultats et les retombées au niveau de l'utilisation d'une méthodologie dépendent toutefois des qualités du concepteur.

Avant de devenir l'outil indispensable du concepteur, une telle méthodologie doit offrir la plus grande souplesse au niveau de l'apprentissage et de la mise en oeuvre.

C'est dans cet objectif que nous avons développé un outil d'aide à la conception.

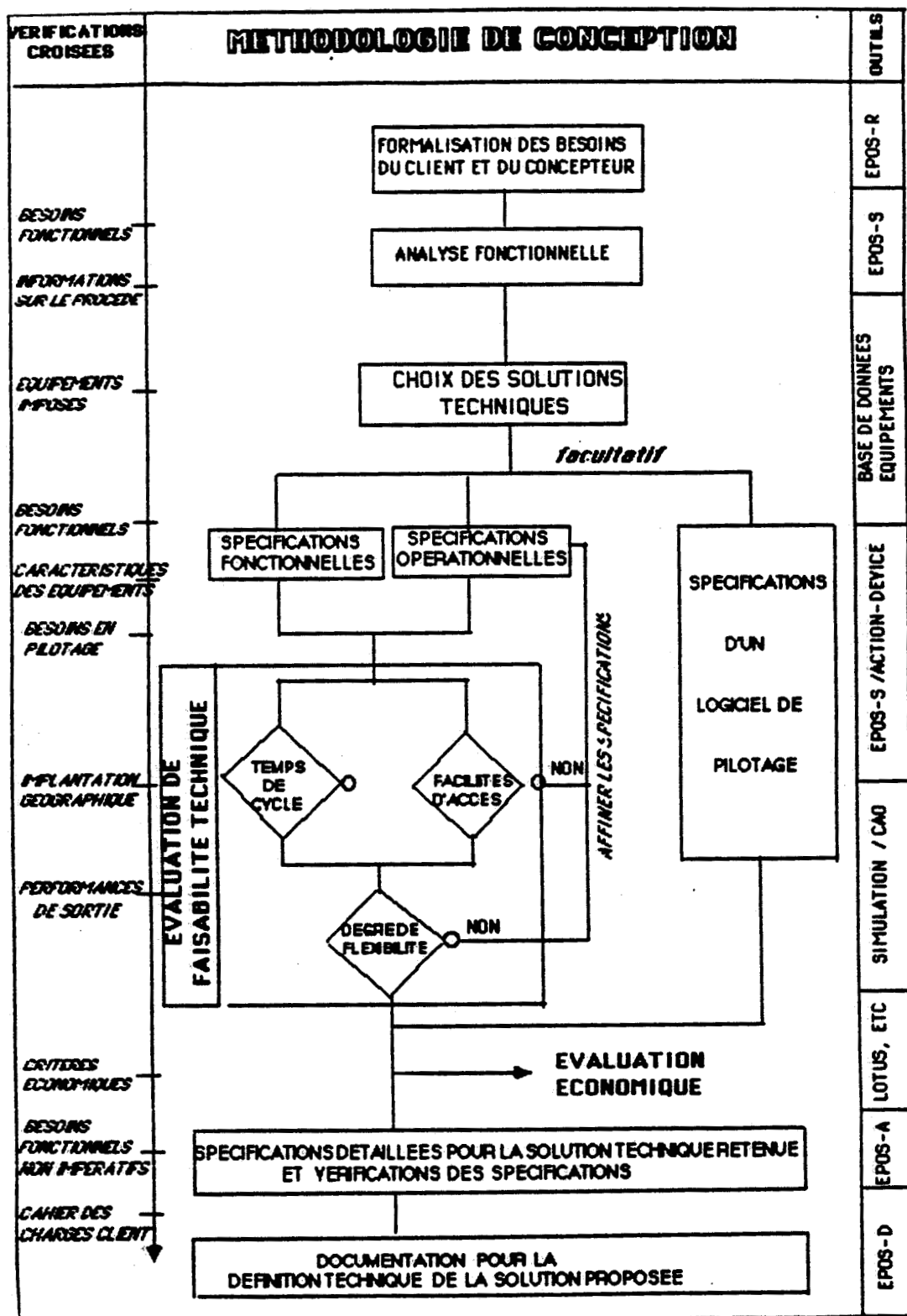


Fig 29 Méthologie de conception des systèmes de fabrication

OUTIL D'AIDE A LA CONCEPTION
DES SYSTEMES de FABRICATION

(chapitre 4)

4.1. Introduction

L'analyse des besoins en ingénierie et l'étude de cas industriels nous a permis de constater :

- (i) d'une part, la variété des cas de conception qui peuvent se présenter,
- (ii) et d'autre part, l'absence de formalisation pour une approche structurée de ces différents cas.

Une première solution est la mise au point d'un GUIDE CONCEPTEUR, tel qu'il a été décrit dans le chapitre 3.

Il fournit une technique d'investigation des informations qui est systématique ainsi que des recommandations pour les formalismes d'écriture à utiliser lors de la description d'une étape de conception.

Néanmoins, cette approche n'est pas suffisante pour maîtriser le volume des spécifications pour certains avant-projets. L'utilisation d'un outil informatique s'impose.

Le système EPOS, joint au GUIDE CONCEPTEUR, est une deuxième solution.

Celle-ci offre au concepteur une approche structurée et un traitement informatique des spécifications qui simplifient la conduite des avant-projets de taille importante.

Néanmoins, il est apparu que cette solution n'est pas entièrement satisfaisante, lors de l'étude de cas industriels complexes.

En effet, le système EPOS n'est pas directement utilisable par les concepteurs de systèmes de fabrication. La formation nécessaire (connaissance des langages de spécification) est pénalisante pour l'utilisation d'un tel outil, dans un bureau des méthodes.

La mise au point d'un véritable outil d'aide à la conception qui soit, à la fois convivial, et, ouvert aux différents problèmes de conception des systèmes de fabrication est donc nécessaire.

En réponse à ces besoins de convivialité, nous avons conçu et réalisé une interface concepteur dialoguant avec le système EPOS.

Cette interface est une méta-structure qui facilite la définition des systèmes de fabrication sur la base d'un GUIDE CONCEPTEUR et du système EPOS.

Elle laisse cependant la possibilité à l'utilisateur de faire évoluer le GUIDE CONCEPTEUR selon ses besoins. En effet, de nouvelles versions du GUIDE CONCEPTEUR pour la réalisation d'un type d'application spécifique peuvent être introduites.

Par conséquent, l'outil logiciel développé propose :

- (i) un support informatique du GUIDE CONCEPTEUR dans son état actuel et une interface vers le système EPOS, mais aussi,
- (ii) l'introduction de l'expérience du concepteur sous la forme d'une nouvelle version du GUIDE CONCEPTEUR actuel.

L'existence d'un tel support informatique facilitera la mise en oeuvre de la méthodologie que nous avons développée. Il réduira les problèmes de mise à jour de la documentation de par l'existence d'un traitement de texte convivial et de facilités graphiques.

Il guidera et fournira toutes les informations nécessaires au concepteur durant sa démarche de réflexion.

Pour des commodités d'écriture, nous désignerons ce logiciel sous l'abréviation PACAV : Prototype d'Aide à la Conception d'Avant-projet.

Ce chapitre a pour but de présenter :

- (a) d'une part les fonctionnalités du logiciel d'un point de vue utilisateur (paragraphe 4.2.) ainsi que l'architecture fonctionnelle retenue (paragraphe 4.3.),
- (b) et, d'autre part, l'environnement de développement et d'exploitation ainsi que les étapes principales de l'analyse organique du logiciel (paragraphes 4.4. et 4.5.).

4.2. Fonctionnalités du système

Les fonctionnalités du logiciel PACAV, en tant que prototype 1987, sont des fonctions de spécification jointe à une fonction de communication de haut-niveau (interface concepteur) pour la conduite d'un avant-projet.

Les fonctions de spécification, c'est à dire la formalisation, l'analyse et la documentation des spécifications sont des fonctions standards offertes par le système EPOS.

Les fonctionnalités offertes par l'interface concepteur ont été définies d'après les besoins des bureaux des méthodes et des sociétés d'ingénierie. Il s'agit essentiellement :

- d'un dialogue homme-machine convivial,
- du traitement interactif d'un avant-projet,
- de l'introduction d'une nouvelle version du GUIDE CONCEPTEUR
- de la gestion des projets et guides concepteurs existants.

Ces fonctionnalités sont détaillées dans les paragraphes suivants sur la base d'un exemple.

4.2.1. Dialogue Homme-machine

Le dialogue homme-machine est en anglais, en raison de la dimension européenne du projet de recherche ESPRIT.

Par contre, les informations entrées par l'utilisateur peuvent être rédigées en français.

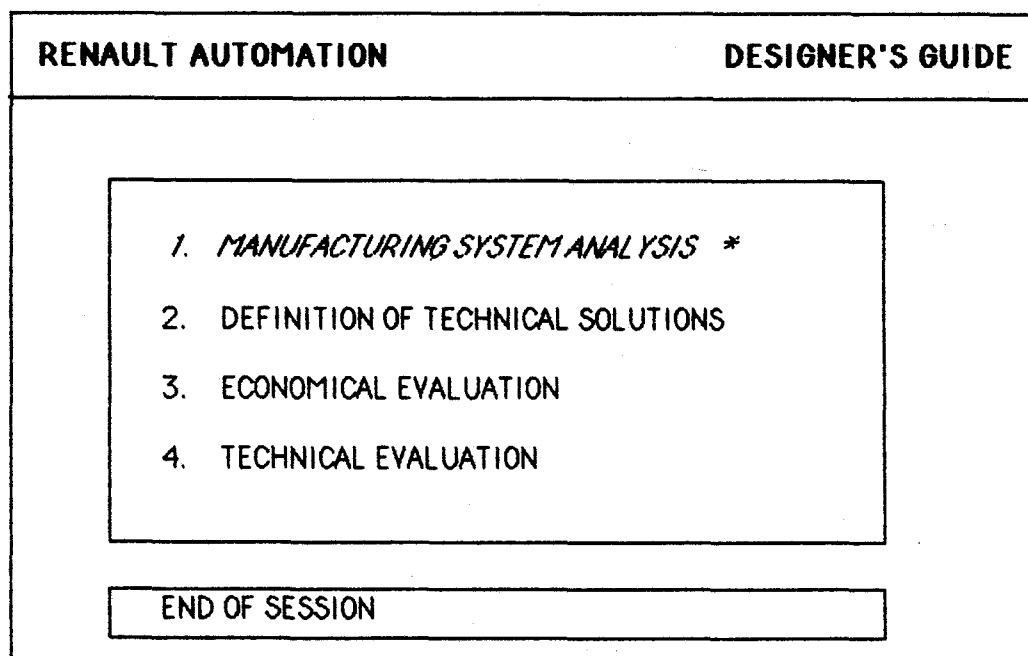
Il est construit sur la base du GUIDE CONCEPTEUR, présenté dans le chapitre 3.

* Initialisation

Le dialogue homme-machine propose un enchaînement de menus qui guident le concepteur lors de la pré-étude d'un système de fabrication.

L'initialisation de ce dialogue consiste à définir une démarche de conception basée sur les quatre étapes qui ont été présentées dans le chapitre 3.

Dans le cadre de notre exemple, qui concerne l'automatisation du procédé de conditionnement d'un pare-brise, le concepteur analyse tout d'abord le système de fabrication.



Cette analyse comprend à la fois la définition des besoins du client et du concepteur.

Dans le cadre de notre exemple, nous nous intéresserons à la spécification des besoins du client.

* Evolution du dialogue

La conduite d'une de ces étapes d'avant-projet nécessite :

- (i) la définition de l'avant-projet et de la version sur laquelle on va travailler,
- (ii) la définition de la structure du dialogue homme-machine, en choisissant le GUIDE CONCEPTEUR approprié au type d'application traitée,
- (iii) la connaissance de l'existant (c'est à dire les avant-projets réalisés et les structures de dialogue déjà mises en place).

Sur la base de ces contraintes, le menu qui apparaît à la suite de la sélection d'une de ces étapes propose trois fonctions standards : PROJECT, STRUCTURE et GENERAL.

RENAULT AUTOMATION		DESIGNER'S GUIDE	
PROJECT	STRUCTURE	GENERAL	END

La fonction PROJECT permet de définir l'avant-projet et la version de travail ainsi que les fonctions de manipulation associées.

L'enchaînement des menus à partir de la sélection de la fonction PROJECT est présenté dans les paragraphes 4.2.2. à 4.2.4.

La fonction STRUCTURE définit les différentes structures de dialogue, qui sont basées sur les GUIDE CONCEPTEURS introduits. Elle est détaillée en 4.2.5.

La fonction GENERAL présente les informations concernant la gestion des bases de données projets et structures. Elle est détaillée en 4.2.6.

Ces trois fonctions de travail : PROJET, STRUCTURE ou GENERAL sont disponibles à l'utilisateur tout au long du dialogue. Le niveau de dialogue inférieur apparaît à l'opérateur selon un formalisme inspiré de celui du traitement interactif mis en place sur le MACINTOSH.

4.2.2. Traitement d'un avant-projet

Lors de la sélection d'un avant-projet, l'utilisateur voit apparaître la liste des avant-projets existants ainsi que l'option "nouveau projet".

Notre exemple industriel est l'avant-projet ACMA que l'on sélectionne :

RENAULT AUTOMATION		DESIGNER'S GUIDE	
<i>PROJECT *</i>	STRUCTURE	GENERAL	END
New project <i>ACMA PROJECT *</i> DOUAI PROJECT 			

Le choix d'une version du projet ou d'une structure (qui est un GUIDE CONCEPTEUR spécifique) se fait de la même manière.

Un avant-projet peut être consulté, détruit ou bien étudié sur la base d'une de ses versions.

L'opérateur choisit l'une de ces trois possibilités :

RENAULT AUTOMATION		DESIGNER'S GUIDE	
<i>PROJECT *</i>	STRUCTURE	GENERAL	END
1. Info consultation 2. <i>Version processing *</i> 3. Project deletion 4. END			

4.2.3. Traitement d'une version

Le concepteur doit, tout d'abord, identifier la version sur laquelle il souhaite travailler.

Cette version peut être consultée, analysée d'un point de vue syntaxique, documentée ou bien mise à jour.

Dans le cadre de notre exemple, le concepteur souhaite introduire de nouvelles spécifications pour la réalisation du système de conditionnement des pare-brise.

Il sélectionne la ligne 6 "Specification processing", afin de créer une nouvelle version de l'avant-projet ACMA (le nom de la version "V3" est affiché en haut de l'écran).

RENAULT AUTOMATION ACMA-PROJECT DESIGNER'S GUIDE			
PROJECT *	STRUCTURE	GENERAL	END
1. Consultation 2. Analysis 3. Documentation 4. Classification 5. Lexicon management 6. <i>Specification processing *</i> 7. Version deletion			

4.2.4. Traitement d'une spécification

L'écriture des spécifications est guidée par les fonctions proposées en en-tête du menu.

L'utilisateur peut faire appel à la fonction AGENDA pour l'écriture des besoins du client.

Cette fonction propose un thème de spécification (BESOINS FONCTIONNELS) et un formalisme associé (REQUIREMENT).

L'utilisateur entre une première information : "le pare-brise doit être tenu lors de la pose du joint-mousse".

Les règles de cohérence sémantique qui sont décrites dans le GUIDE CONCEPTEUR pour chaque thème se traduisent par des messages d'aide qui apparaissent en bas de l'écran.

Dans le cadre de notre exemple, un message d'aide, pour la spécification des besoins fonctionnels, apparaît en bas de l'écran :

"N'oubliez pas de remplir le constituant 123, si des besoins en contrôle-qualité ont été identifiés".

RENAULT AUTOMATION		ACMA PROJECT		DESIGNER'S GUIDE	
AGENDA *	OPEN	CONSULT	INFO	VERSION	END
BESOINS FONCTIONNELS REQUIREMENT <i><impératif fonctionnel></i> "Le pare-brise doit être tenu lors de la pose du joint-mousse"					
INSERT	REQ	CONS	DEC PROC	TERM	TEXT
<i>N'oubliez pas de remplir le constituant 123, si des besoins en contrôle-qualité ont été identifiés.</i>					

Les autres fonctions disponibles sont les suivantes :

- OPEN : permet au concepteur de remplir librement le dossier de spécification d'un avant-projet, en proposant une liste de thèmes et le formalisme de spécification associé.
- CONSULT : permet la consultation des spécifications déjà rédigées.
- INFO : fournit des informations sur l'état d'avancement du dossier de spécification (quelles sont les spécifications complètes, ...).
- VERSION : enregistre la version courante d'un avant-projet.

4.2.5. Choix du dialogue opérateur : Introduction de l'expertise du concepteur

L'utilisateur est en mesure de sélectionner le dialogue homme-machine.

Initialement, le logiciel propose un dialogue basé sur le GUIDE CONCEPTEUR que nous avons développé.

Néanmoins, ce dialogue peut se révéler inadapté pour le traitement d'un avant-projet spécifique.

Dans ce cas, l'utilisateur a la possibilité d'introduire son propre savoir-faire sous la forme d'une nouvelle version du GUIDE CONCEPTEUR.

Sur la base de ces informations, un nouveau dialogue homme-machine est proposé.

Lors de la sélection du thème STRUCTURE, l'opérateur voit apparaître les différentes possibilités de gestion du dialogue homme-machine.

RENAULT AUTOMATION		DESIGNER'S GUIDE	
PROJECT	STRUCTURE *	GENERAL	END
STANDARD STRUCTURE 1..... Select a structure or 1. Consultation 2. New name 3. Updating 4. Deletion 5. END Return			

Le concepteur a donc la possibilité de créer une nouvelle structure de dialogue, de la consulter, de la mettre à jour ou de la détruire.

4.2.6. Gestion de Bases de Données

Les avant-projets et les structures de dialogue ainsi créés doivent pouvoir être stockés dans des Bases de Données distinctes.

L'opérateur y a accès avec la fonction GENERAL :

RENAULT AUTOMATION		ACMA-PROJECT	DESIGNER'S GUIDE
PROJECT	STRUCTURE	GENERAL *	END
1. PROJECTS DB 2. STRUCTURE DB RETURN			

Toutes les recommandations et menus liés aux concepts méthodologiques suivant lesquels le concepteur élabore un dossier d'avant-projet font partie des informations dites structurelles ainsi que l'expertise introduite par le concepteur lui-même.

Toutes ces informations sont associées à une structure de dialogue et stockés dans la Base de Données Structures .

La Base de Données Projets regroupe tous les avant-projets qui ont déjà été traités.

4.3. Architecture fonctionnelle

Les fonctions requises nous conduisent à identifier trois modules fonctionnels et trois bases de données externes.

De manière générale, les données échangées entre ces différents modules et système sont des chaînes de caractères, des commandes ou des clefs correspondant à un choix de l'opérateur dans le menu affiché.

L'entrée des données se fait au niveau du terminal opérateur.

Chaque module fonctionnel, présenté dans le diagramme ci-dessous, a fait l'objet d'une analyse fonctionnelle détaillée (ANNEXE 2).

les spécifications du logiciel PACAV, selon la méthode de développement METHRA ont fait l'objet d'un dossier complet [AUGER-86/2].

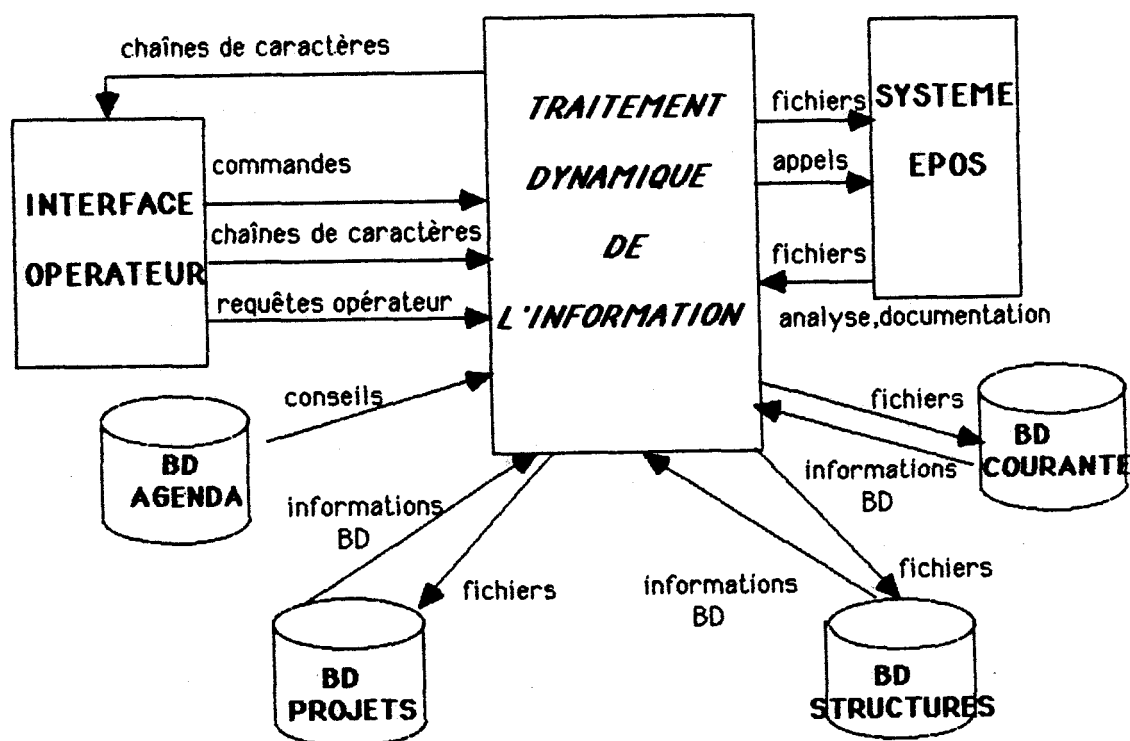


Fig 30 Architecture fonctionnelle de l'outil PACAV

4.3.1. Traitement Dynamique de l'Information (TDI)

Ce module fonctionnel a pour objectif d'assurer la rédaction d'un dossier d'avant-projet.

Toutes les fonctions de traitement d'un avant-projet identifiées précédemment (consulter, détruire, ...) sont regroupées dans ce module.

Il initialise et supervise les fonctions de gestion de bases de données, le dialogue opérateur et les appels vers le module de traitement des spécifications.

Il gère, également, avec l'Interface Opérateur, l'introduction de l'expertise du concepteur qui sera à la base d'un nouveau dialogue. Il recommande un formalisme spécifique pour l'entrée d'informations et propose à l'utilisateur une liste d'actions à entreprendre.

4.3.2. Interface Opérateur

Ce module fonctionnel centralise toutes les fonctions de gestion du dialogue.

Il prend en charge l'affichage des différents menus et écrans associés, lit les données saisies à l'écran par l'opérateur et les envoie vers le module de Traitement Dynamique de l'Information.

4.3.3. Module de traitement des spécifications

Ce dernier module fonctionnel regroupe toutes les fonctions d'analyse, de classification et de documentation d'un dossier de spécification.

Un glossaire de l'avant-projet est également défini.

4.3.4. Bases de Données externes

La Base de Données Projets comprend tous les avant-projets, et, les versions associées qui ont été traités.

La Base de Données Structures stocke l'expertise du concepteur sous la forme d'une structure.

La Base de Données Courante contient toutes les informations sur la version courante du dossier d'un avant-projet (quelles sont les spécifications complètes, ...).

Les recommandations pour l'écriture des spécifications, provenant du GUIDE CONCEPTEUR, sont enregistrées sous la forme d'un agenda.

Compte tenu de la position centrale du module de Traitement Dynamique de l'Information, nous détaillerons le flux des données à partir des entrées et des sorties de ce module fonctionnel.

De manière générale, l'échange des données avec l'Interface Opérateur consiste en l'exécution des fonctions sélectionnées par l'opérateur et la restitution des données enregistrées pour une version d'un avant-projet.

L'échange des données avec le système EPOS consiste en des commandes d'exécution des fonctions de spécification.

I. ENTREES

En provenance de l'Interface Opérateur sont fournies des chaînes de caractères, des commandes ou des requêtes opérateur.

Les chaînes de caractères sont des informations entrées par l'opérateur lors de l'écriture du dossier de spécification.

Il peut s'agir de l'état d'une spécification (complète ou incomplète), de son contenu, du nom du projet ou de la version.

A ce niveau, l'expertise du concepteur peut être introduite sous la forme d'un GUIDE CONCEPTEUR.

Les commandes sont des fonctions sélectionnées par l'opérateur dans le menu affiché.

Les requêtes Opérateur peuvent être des questions de l'opérateur à propos d'informations générales comme la date de création, l'auteur d'un projet, d'une version ou d'une structure.

En provenance du module de traitement des spécifications sont fournis, soit des fichiers d'erreur résultant de l'analyse des spécifications, soit des fichiers de documentation.

Les trois Bases de Données (Projets, Structures et Courante) fournissent des informations générales sur la gestion des éléments qu'elles contiennent.

Dans le cas de la base de données Projets sont traitées toutes les informations historiques et descriptives liées à la création d'un avant-projet.

Toutes les informations sur les structures déjà existantes (noeuds du dialogue, noeuds terminaux, règles de cohérence) sont fournis par la Base de données Structures.

La Base de Données courante fournit toutes les informations historiques et descriptives qui se rapportent à la version courante de l'avant-projet traité.

Il peut s'agir du contenu d'une spécification, des commentaires associés, de la liste des thèmes de spécification déjà traités ou de ceux qui restent à traiter.
Les fichiers échangés avec le module TDI sont les différentes formes de la version courante .

Enfin la Base de Données AGENDA fournit uniquement des recommandations pour guider l'utilisateur dans l'écriture des spécifications.

II. SORTIES

Le module Interface Opérateur reçoit des chaînes de caractères. Il s'agit d'informations qui ont été saisies auparavant par l'opérateur ou encore des messages d'aide, d'erreurs et les résultats du traitement des spécifications.

Le module de traitement des spécifications reçoit des appels pour le lancement des procédures d'analyse ou de documentation des spécifications, ainsi que le fichier contenant une version du dossier de spécification à traiter.

Les trois Bases de Données (Projets, Structures et Courante) reçoivent des fichiers contenant respectivement une version du dossier de spécification à figer, une structure à enregistrer, la version courante du dossier de spécification.

4.4. Environnement de développement et d'exploitation4.4.1. Architecture matérielle

L'environnement matériel pour réaliser le prototype 1987 est un MICROVAX/VMS, un terminal d'entrée-sortie VT220 et les périphériques graphiques suivants : une console TEKTRONIX 41-B-15 et une table traçante HP.

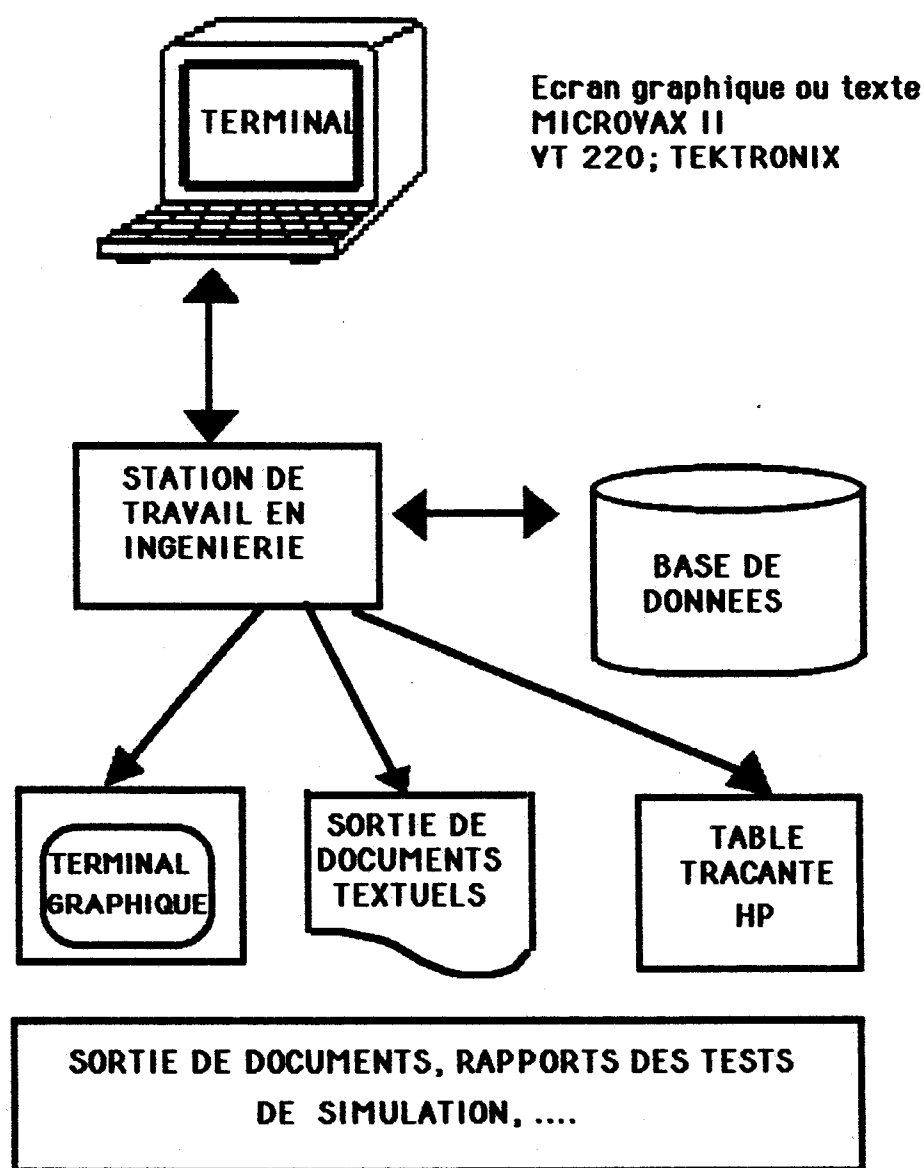


Fig 31 Architecture matérielle de l'outil PACAV

4.4.2. Environnement de développement

L'environnement de développement qui a conduit à la réalisation du logiciel PACAV consiste en :

- la méthode de développement METHRA,
- le traitement de texte WPS-PLUS pour la rédaction des spécifications du logiciel,
- le langage VAX-LISP et le debugger associé pour la réalisation du module de traitement dynamique de l'information,
- le langage PASCAL pour la réalisation du module de gestion d'écran (interface opérateur),
- l'environnement de programmation VMS pour l'interfaçage entre le programme maître en LISP et le système EPOS.

Nous avons mis en oeuvre la méthode de développement METHRA, afin d'assurer une meilleure qualité des logiciels développés au sein de la Division des Etudes de Commandes.

Cette méthode s'inspire des concepts d'une analyse descendante, modulaire et structurée.

Les phases d'analyse de METHRA sont orientées, de manière générale, vers la spécification de logiciels temps-réel, multi-tâches avec des contraintes d'ouverture et le codage dans des langages de haut-niveau (Pascal, Modula).

Nous distinguons dans METHRA huit phases pour la conception d'un logiciel :

- Analyse des Besoins,
- Analyse Fonctionnelle,
- Analyse Organique Générale,
- Analyse Organique Détaillée,
- Codage,
- Tests Unitaires,
- Intégration,
- Validation.

Chacune de ces étapes fait l'objet d'un document écrit dont la structure est définie par METHRA.

Par exemple, le dossier d'Analyse Fonctionnelle est rédigé en partie en langage naturel, et, selon un formalisme graphique proche de SADT.

L'Analyse Organique fait appel à un formalisme graphique spécifique détaillé dans l'annexe A3.

VAX-LISP a été retenu en final, comme support de réalisation du module de traitement dynamique de l'information en raison de sa structure de données (p-listes). Il est défini selon le standard COMMON-LISP et commercialisé par DIGITAL.

La structure de données qu'il propose, est bien adaptée à la modélisation des thèmes d'une liste d'informations et de règles de cohérence associées.

4.4.3. Environnement d'exploitation

L'environnement d'exploitation comprend l'utilitaire FMS, (FORMS MANAGEMENT SYSTEM), le système EPOS ainsi que l'environnement de programmation VMS.

Les raisons du choix du système EPOS ont été détaillées dans le chapitre 1.

La réalisation du module fonctionnel "Interface Opérateur", a fait appel à des utilitaires de gestion de dialogue. Nous avons étudié trois utilitaires disponibles sous MICROVAX/VMS, à savoir : TDMS, FMS et SMG.

FMS (Forms Management System) a été retenu en raison d'une mise en oeuvre plus aisée pour des fonctionnalités équivalentes, et, ses facilités d'interfaçage avec Pascal.

Enfin, le choix de l'environnement de programmation VMS est lié aux contraintes de l'architecture matérielle qui a été retenue.

4.4.4. Conclusions

Devant l'intérêt des concepteurs pour un outil intégré d'aide à la conception, nous avons réalisé une première maquette basée sur des techniques d'Intelligence Artificielle.

L'architecture matérielle et l'environnement de développement qui ont été retenus sont standards, dans le but de faciliter l'évaluation de cette maquette.

Ils ont largement contribué à une approche logicielle structurée et modulaire.

Néanmoins, la solution retenue pour la gestion d'écran devra être revue pour la réalisation d'une deuxième version du prototype.

L'utilisation d'un terminal VT220 et la procédure de définition des écrans imposée par FMS rendent difficile la mise en oeuvre de solutions à multi-fenêtrage.

L'opérateur a des difficultés pour faire évoluer sa gestion d'écran et redéfinir son écran de travail (découpage en nouvelles zones, redéfinition de la taille des fenêtres).

Cette première expérience de programmation en langage interprété pourrait être également à la base d'une réflexion originale sur la nature des méthodes de développement de logiciels d'Intelligence Artificielle.

La spécification d'un module en LISP nous a permis de constater la nécessité d'adapter, sur quelques points, la méthode de développement METHRA.

Les phases d'analyse organique (générale et détaillée) devront être revues.

Une modélisation du logiciel sous la forme de systèmes, modules et tâches n'est pas toujours très bien adaptée.

De même, la phase de codage qui est prévue dans toutes les méthodes de développement n'a plus lieu d'être.

Par contre, une bonne documentation du programme est nécessaire, du fait de l'enchaînement et de l'imbrication des différentes fonctions LISP.

4.5. Analyse organique

Nous allons détailler quelque peu les mécanismes de programmation utilisés pour réaliser les deux modules fonctionnels identifiés précédemment : le module de Traitement Dynamique de l'Information et le module Interface Opérateur.

4.5.1. Réalisation du module TDI

L'analyse organique conduit à structurer le programme selon les modules suivants :

- fonctions du module TDI,
- bibliothèque de procédures de commande,
- bibliothèque de routines externes et d'alien-structures
- bibliothèque de fonctions de gestion de bases de données,
- programme maître.

Les interactions entre ces différents modules sont détaillées sur le diagramme page suivante.

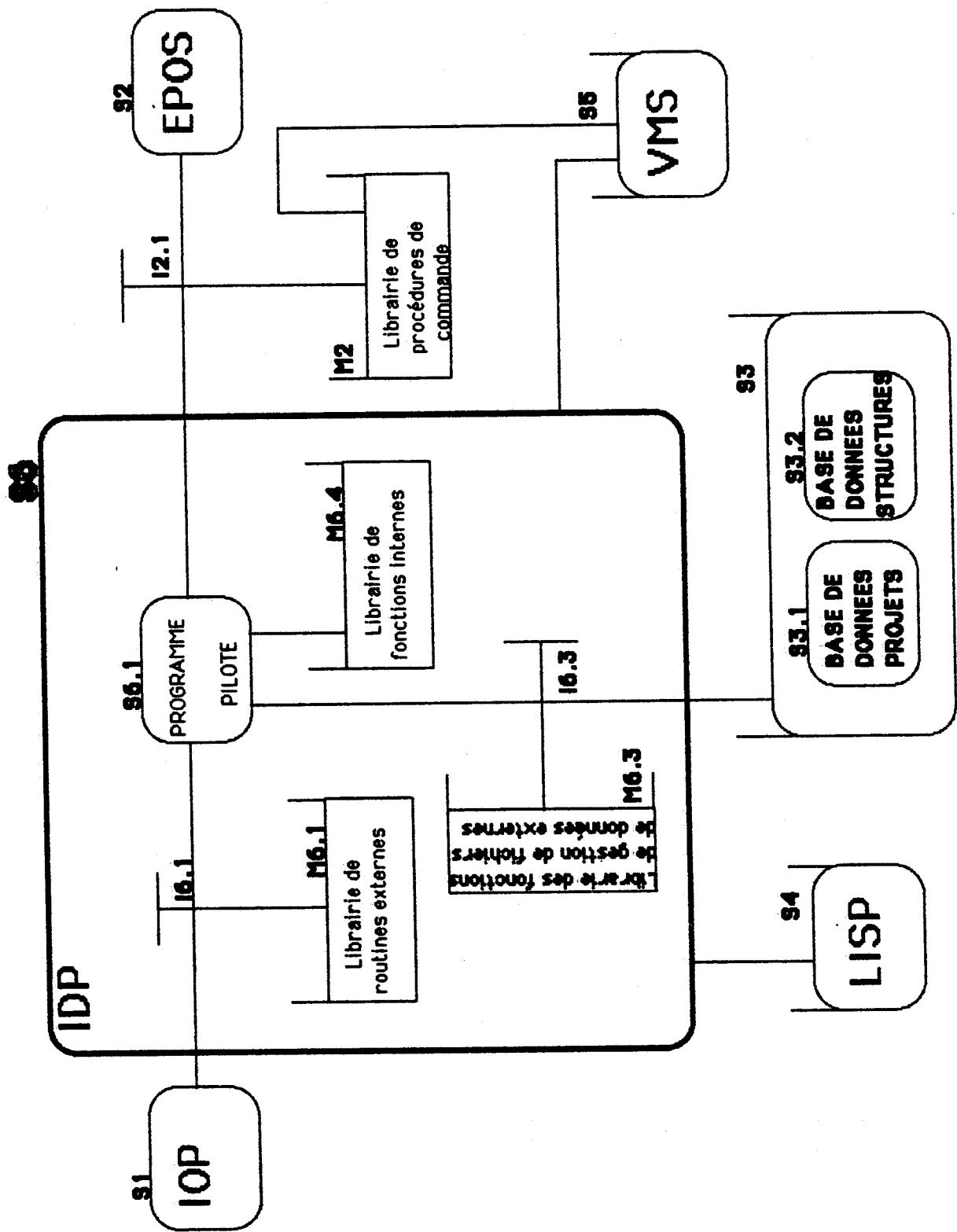


Fig 32 Architecture organique du module TDI

4.5.1.1. Fonctions du module TDI

Les fonctions de manipulation des spécifications comme "ouvrir une spécification", la "consulter", suivre l'"agenda" sont des fonctions LISP. Elles gèrent la Base de Données AGENDA et la Base de Données courante.

4.5.1.2. Bibliothèque de procédures de commande

La bibliothèque de procédures de commande permet de gérer, à partir du programme LISP les appels au système EPOS et les envois de données dans un format compréhensible par celui-ci.

Une procédure de commande est un fichier spécifique écrit en langage de commande DIGITAL (DCL).

L'appel d'une fonction spécifique du système EPOS nécessite une procédure de commande et un fichier de données associé.

Dans ce fichier de données sont stockées toutes les réponses qu'aurait fait un utilisateur en interactif lors du déroulement du menu du système EPOS.

4.5.1.3. Bibliothèque de routines externes et d'alien-structures

L'échange des données entre les modules de Traitement Dynamique de l'Information et l'Interface Opérateur fait appel à une structure de programmation particulière. Il s'agit de faire dialoguer un programme PASCAL (compilé) et un programme LISP (interprété).

Un programme LISP peut appeler des procédures compilées ou des fonctions PASCAL en les considérant comme des routines externes. Le schéma inverse est impossible. Par conséquent, l'échange des données est basé sur les mécanismes LISP dédiés à l'appel de procédures compilées (routines externes et alien-structures).

4.5.1.4. Bibliothèque de fonctions de gestion de bases de données

Cette bibliothèque de fonctions LISP gère l'utilisation des deux bases de données externes : la Base de Données Projets et la Base de Données Structures.

4.5.1.5. Programme maître

Il initialise les variables globales utilisées ainsi que les appels vers les fonctions du module TDI, les routines externes qui ont été définies, et, le système EPOS.

4.5.2. Réalisation du module Interface Opérateur

L'analyse organique du module Interface Opérateur réalisé par l'utilitaire FMS conduit à la décomposition suivante :

- gestion du dialogue (GEST DIAL),
- gestion de la visualisation (GEST VISU),
- interface avec le module TDI (I/TDI).

Les interdépendances entre ces trois modules sont décrites dans le diagramme ci-dessous

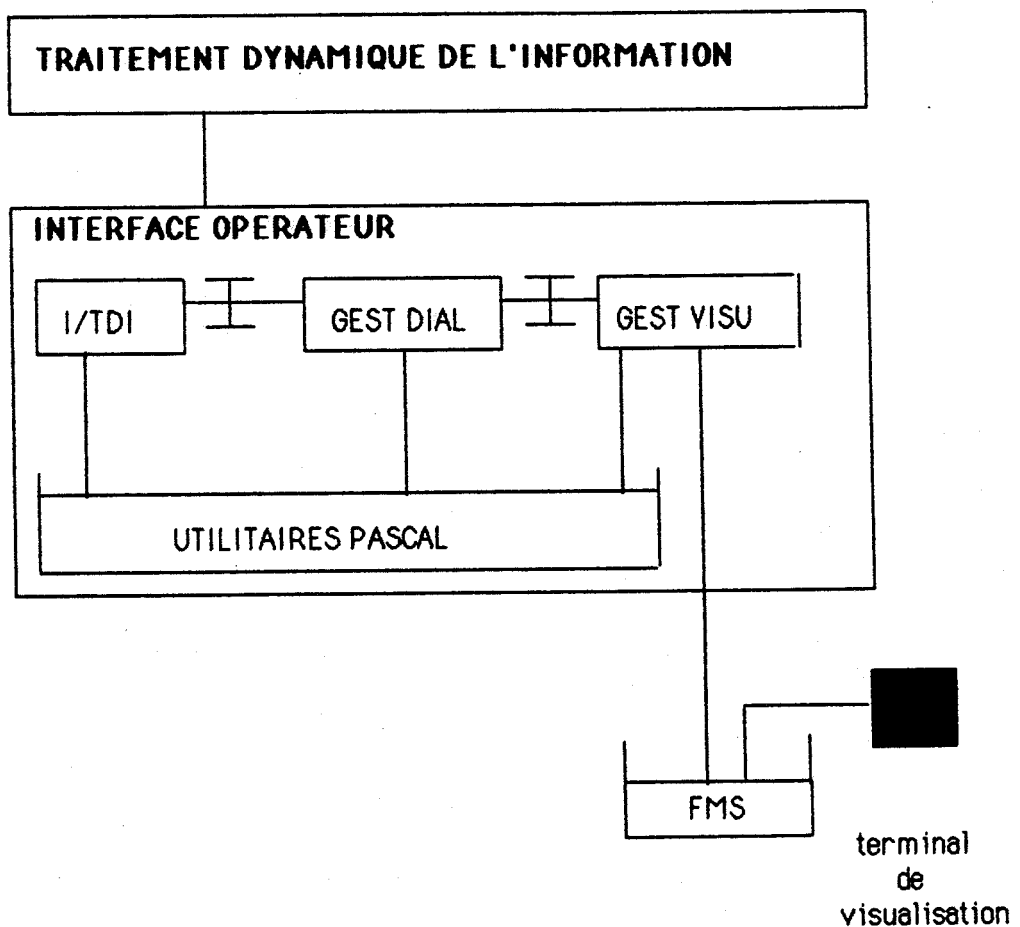


Fig 33 Architecture organique du module Interface
Opérateur

4.5.2.1. Gestion du dialogue

Ce module PASCAL assure la mise à jour, l'enregistrement du dialogue et des données saisies à l'écran.

La nouvelle transaction homme-machine est également définie à ce niveau.

4.5.2.2. Gestion de la visualisation

Tous les écrans à afficher lors de l'évolution du dialogue sont définis dans ce module.

Il est réalisé en Pascal et fait appel au FORM DRIVER de l'utilitaire FMS.

4.5.2.3. Interface vers le module TDI

L'interface vers le module TDI est constituée de toutes les procédures PASCAL déclarées comme globales et qui sont des points d'entrée vers le module LISP.

Un exemple de procédure globale est la transmission du choix de l'opérateur dans un menu vers le module TDI.

4.6. Conclusion

Deux utilisateurs pilotes : les Méthodes Centrales de la Régie Renault et l'Aérospatiale évaluent cet outil d'aide à la conception sur des avant-projets internes.

De premières expériences sur des cas industriels concrets nous ont déjà amené à constater la nécessité de prévoir un dimensionnement suffisant des bases de données projets et structures.

Un temps de réponse plus performant est également souhaitable, compte-tenu des performances de la première version réalisée.

Par contre, la convivialité de cet outil par rapport aux autres outils classiques d'ingénierie a trouvé un écho favorable auprès des utilisateurs.

Le déroulement de l'agenda est géré en fonction des thèmes sélectionnées par l'utilisateur.

Des règles de cohérence provenant du GUIDE CONCEPTEUR sont associées à chacun de ces thèmes, et, apparaissent sous la forme de messages d'aide lors de la sélection de l'un de ces thèmes.

Cette gestion s'inspire des techniques d'intelligence artificielle (demons) qui permettent le déclenchement automatique de certaines procédures après des événements précis.

En outre, la standardisation de l'architecture matérielle et des moyens d'exploitation offre l'avantage d'un investissement minimum pour l'acquisition du logiciel et la formation de l'utilisateur.

Enfin, la mise en place d'une méta-structure permettant d'introduire l'expertise du concepteur assure la pérennité d'un tel outil et son utilisation dans des domaines très divers de la conception des systèmes de fabrication.

Ce logiciel est donc ouvert, adaptable à différents types de projets et à caractère évolutif.

CONCLUSION GENERALE
en terme de
PERSPECTIVES DE DEVELOPPEMENT

(chapitre 5)

L'utilisation de méthodologies est désormais reconnue comme indispensable dans le domaine de la Production Automatisée, pour faire face à la complexité des projets traités.

Cette complexité peut être de nature très diverse (organisationnelle, humaine, technique).

Dans ce mémoire, notre objectif essentiel a été la présentation d'une méthodologie de conception des systèmes de fabrication de taille industrielle.

Elle représente la synthèse des recherches et expériences que nous avons entrepris dans ce domaine et détaillé dans le chapitre 3.

A travers cet exposé, nous nous sommes efforcés de répondre clairement aux questions essentielles que se pose tout concepteur, lors de l'étude d'un système de fabrication, en proposant également un outil d'aide à la conception adapté à ses besoins.

L'approche choisie pour l'analyse des besoins des concepteurs, la formalisation d'une méthodologie sur la base d'un modèle conceptuel, et, le modèle conceptuel retenu sont généraux.

Nous pensons qu'ils peuvent être une bonne base de travail pour la résolution d'autres problèmes de conception.

De même, la structure et les concepts fonctionnels mis en oeuvre pour la réalisation de notre logiciel peuvent être repris pour la définition d'un outil analogue pour un domaine d'application différent (conception de circuits intégrés VLSI, par exemple).

Principaux résultats

Dans le cadre de ce travail de recherche, nous nous sommes efforcés d'avoir une approche rationnelle et prospective dans le domaine nouveau et en plein essor de la Production Automatisée.

Cette démarche a été initialisée par l'identification des besoins des concepteurs dans ce domaine. Notre réponse a été structurée sur la base de concepts informatiques transposés dans un contexte productique.

L'originalité de notre démarche a consisté à choisir un modèle conceptuel pour la formalisation du processus de conception.

Ce modèle est capable d'induire une compréhension progressive des réalisations.

Un autre apport significatif résulte de la réalisation d'un véritable outil d'aide à la conception.

En effet, cet outil satisfait les besoins d'une approche intégrée devenue impérative pour faire face à la complexité et à la taille croissante des projets traités.

Il gère la pré-étude de systèmes de fabrication sur la base de l'expertise du concepteur et d'un ensemble de procédures de vérifications.

Enfin, il a été conçu selon une approche modulaire et systématique du fait de l'utilisation de la méthode de développement METHRA.

Sa maintenance et surtout son adaptation aux besoins futurs des concepteurs en seront simplifiées.

Les utilisateurs potentiels d'un tel outil d'aide à la conception sont les sociétés d'ingénierie ou les bureaux des méthodes des grandes entreprises.

Ceux-ci s'orientent, de plus en plus, vers une démarche méthodique et intégrée, et, l'utilisation d'outils logiciels d'aide et de validation de la conception.

DEUX TYPES D'EXTENSIONS DU PROTOTYPE peuvent être alors envisagées :

- 1 - une extension du prototype réalisé, de manière à couvrir les étapes de développement et de réalisation d'un système de fabrication.

Des travaux ont déjà été faits au laboratoire d'Informatique Industrielle de l'I.D.N pour la définition d'un modèle de conception détaillée et son évaluation dynamique [CASTELAIN-87].

- 2 - une extension des fonctionnalités du logiciel développé en restant dans le cadre de la pré-étude d'un système de fabrication.

DANS CE DERNIER CAS DE FIGURE, LES EVOLUTIONS SUIVANTES peuvent être envisagées, à partir du prototype réalisé :

- 2 - a) l'intégration du savoir-faire des concepteurs dans des domaines hautement spécialisés,
- 2 - b) l'intégration d'outils d'ingénierie, autres que le système EPOS.

La première solution conduit à une expertise des différents métiers de la conception (assemblage, soudure, ...).

Cette expertise est formalisée selon un GUIDE CONCEPTEUR, et, introduite dans l'outil d'aide à la conception sous la forme d'une nouvelle structure de dialogue.

La seconde solution consiste à intégrer différents outils de validation utilisés par les concepteurs d'aujourd'hui.

Un logiciel de simulation SLAM, un outil de programmation hors-ligne ou encore des algorithmes économiques pourraient faire l'objet d'une telle intégration.

Ce type de développement contribuera, à long terme à la mise au point d'une station de travail pour la conception des systèmes de fabrication.

Ces outils permettront de définir rapidement et d'optimiser la solution technique proposée, en fonction du cahier des charges de l'application.

De l'ingénierie des systèmes de fabrication vers leur programmation :l'évolution de l'outil d'aide à la conception

Les résultats essentiels au terme de deux années de recherche dans le cadre du projet ESPRIT sont les suivants :

- A - la mise au point d'un premier OUTIL D'INGENIERIE sous la forme d'un logiciel d'aide à la conception que nous venons de présenter dans ce mémoire.
- B - la mise au point d'un MODULE DE PROGRAMMATION HORS-LIGNE mono-poste dédié à la modélisation, la programmation hors-ligne et la simulation d'une cellule robotisée d'assemblage. Il est présenté en ANNEXE 4.

Par conséquent, la première étape prospective, qui répond aux besoins des concepteurs lors des pré-études des systèmes de fabrication, concerne l'intégration du module de programmation hors-ligne.

Il permettrait de valider les solutions, définies sur la base du GUIDE CONCEPTEUR, et, spécifiées selon les formalismes EPOS, d'après des critères dynamiques (temps de cycle, performances).

Des études préliminaires ont été conduites pour prouver la faisabilité d'une telle réalisation.

D'autres travaux, menés en parallèle, ont pour finalité le développement d'outils d'ingénierie. Ceux-ci pourraient être intégrés dans le logiciel que nous avons réalisé.

L'Université de GALWAY développe un outil de simulation basé sur le langage SLAM, ainsi qu'un système expert de génération de gammes d'assemblage.

L'Université de STUTTGART développe avec la collaboration de RENAULT AUTOMATION, un système expert d'aide à la décision pour le choix du (ou des) robot(s) intervenant dans la réalisation d'un système de fabrication.

BIBLIOGRAPHIE

[ANCELIN - 87]

B. ANCELIN, CH. FRAY

"L'évaluation économique des systèmes de production"

Conférence INRIA, Systèmes de Production, Avril 1987.

[AUGER - 86/1]

P. AUGER, D. VOJNOVIC

"Designer's Guide : a methodology for the design of production units"

RENAULT AUTOMATION, Avril 1986.

[AUGER - 86/2]

P. AUGER, D. VOJNOVIC

"Designer's Guide Prototype System"

Functional requirements, conceptual requirements, functional analysis and organic analysis

RENAULT AUTOMATION, Septembre 1986.

[BERNUS - 79]

BERNUS and HAVANY

"Design of Integrated manufacturing Systems"

Computers in Industry, Vol.1, N°1, Juillet 1979.

[BLANCHARD - 78]

M. BLANCHARD

"Rapport de la commission de normalisation de la représentation du cahier des charges"

Automatisme - tome XXIII - N°3-4 - Mars-Avril 1978.

[BEHUNIAK - 81]

J. A. BEHUNIAK

"Planning the Successful Robot Installation"

Robot". Today, vol.3 N°2, 1981.

[BERGANT - 82]

J. M. BERGANT

"Achieving a Successful First Robot Installation"

Robot". Today, vol.4 N°1, Février 1982.

[CALVEZ - 82]

J. P. CALVEZ

"Une méthodologie de conception des systèmes multi-coordonateurs pour les applications de commande en temps-réel"

Université de Nantes, 1982.

[CALVEZ - 78]

J. P. CALVEZ, G. THIBAUT
"A model of functional structuration of complex systems"
Université de Nantes, 1978.

[CAMPAGNE - 87]

J. P. CAMPAGNE, C. CAPLAT
"Elaboration automatique de gammes d'assemblage"
Conférence INRIA, Systèmes de Production, Avril 1987.

[CASTELAIN - 87]

E. CASTELAIN
"Modélisation et Simulation interactive de Cellules de
Production Flexibles dans l'Industrie Manufacturière"
Université de LILLE I, Février 1987.

[DEMETROVICS - 87]

J. DEMETROVICS, E. KNUTH, P. RADO.
"Computer-Aided Specification Techniques"
World Scientific, Series in Computer Science, Vol.1,
1985.

[DEMETROVICS - 82]

J. DEMETROVICS, E. KNUTH, P. RADO, Hungarian Academy of
Sciences
"Specification Meta-Systems". IEEE, 1982.

[EPPLE - 84]

W. K. EPPLE
"Rechnerunterstützte Spezifikation von
Prozessautomatisierungssystemen"
RTP-Softwaretechnik, 1984.

[EPPLE - 83]

W. K. EPPLE, M. D. HAGEMANN, M. K. KLUMP, U. REMBOLD
"The Use of Graphics Aids for Requirements Specification
of Process Control Systems"
IEEE, 1983.

[FISHER - 83]

MARTIN A. FISHER
"Engineering Specifications Writing Guide ,
An authoritative reference for Planning, Writing and
Administering" .
Prentice-Hall, Inc., Englewoods Cliffs, 1983.

[GIRALDI - 87]

J. C. GIRALDI, J. C. TARONDEAU
"Technologies flexibles et organisation du travail"
Conférence INRIA, Systèmes de Production, Avril 1987.

[GPP - 85]

GPP, Gesellschaft für Prozessrechner-programmierung mbH
EPOS MANUALS, München, 1985.

[HAMMER-79]

M. HAMMER and G. RUTH
"Automating the Software Development Process"
Research Directions in Software Technology, P.Wegner,
ed., MIT Press, Cambridge, Mass., and London, England, 1979.

[HARLOW - 86]

JUSTIN E. HARLOW III
"What every engineer should know about Engineering Workstations".
Department of Electrical and Computer Engineering,
University of Cincinnati, Cincinnati, Ohio,
Editor William H. Middendorf, vol.16, p.147, 1986.

[HENINGER - 80]

K. L. HENINGER
"Specifying Software Requirements for Complex Systems :
New techniques and their Applications "
IEEE Trans. Software Engineering, Vol. SE-6, N°1, Jan 1980.

[HOWDEN - 81]

"Contemporary Software Development Environments"
ACM Software Engineering Notes, Vol.6, N°4, Août 1981.

[IGL - 76]

SOFTECH SADT course workbook, Vol.1, 2, 3.
Softech., Inc., 460 Totten Road Waltham, Massachusetts
Août 1976.

[KAMPEN - 81]

G. KAMPEN
"Experience using Automated Software Tools in the
Specification and design of a Large Aerospace
Application"
Proc. Conf. Use of Computer Aids in Systems Development,
System Designers Limited, U.K. 1981.

[KOCH - 83]

G. R. KOCH, W. K. EPPLÉ
"A System for application oriented requirements
specification". University of Karlsruhe, 1983.

[KOCH - 82]

G. R. KOCH
"Application Oriented Specifications for process
automation". University of Karlsruhe, 1982.

[KOCH - 82]

G. R. KOCH
"A Guide to Application Oriented Requirements
Specifications" .
Report of the Technical Committee N°11 of the European
workshop of industrial computer systems, 1982.

[KOREN - 85]

Y. KOREN
"Robotics for engineers". Israel Institute of Technology,
McGraw-Hill Book Company, 1985.

[KRONENTAL - 77]

M. KRONENTAL
"Contribution au développement d'une méthode de
conception des systèmes informatiques"
Thèse 3ème cycle-INPG-1977.

[KUSIAK - 87]

A. KUSIAK
"Knowledge Engineering and optimization in automated
manufacturing systems"
Conférence INRIA, Systèmes de Production, Avril 1987.

[LEHMAN - 80]

M. M. LEHMAN
"Programs, Life cycles and Laws of Software evolutions"
Proc.IEEE, Vol.68,1980.

[LUDEWIG - 78]

J. LUDEWIG, W. STRENG
"Methods and Tools for Software Specification and
Design, A Survey" .
EWICS, TC7, Paper N°149, Zurich, Avril 1978.

[LUSS - 87]

H. LUSS, M. FARBER, CHANG-SUNG YU
"Assembly line design tools : line balancing and line
layout"
Conférence INRIA, Systèmes de Production, Avril 1987.

[MAC DERMID - 84]

J. Mac DERMID, System designers Ltd, Fleet
K. RIPKEN, TECSI - Software, Paris
"Life cycle support in the ADA environment"
The Ada Companion Series, Cambridge University Press,
p.247, 1984.

[MILLER - 81]

T. J. MILLER and B. J. TAYLOR
"A System Requirements Methodology"
IEEE Electro 81 Conf.Proc., New York, Avril 1981.

[MORGAN - 84]

C. MORGAN
"Robots Planning and Implementation" .
IFS Publications Ltd., UK, Springer-Verlag, p.195, 1984.

[MOSER - 87]

J. M. MOSER, R. STEPOURJINE, J. VIAL
"Méthodologie de conception d'une cellule flexible
d'usinage "
Conférence INRIA, Systèmes de Production, Avril 1987.

[PARNAS - 77]

D. L. PARNAS
"The Use of Precise Specifications in the Development
Software"
B. Gilchrist (ed.), Information Processing 77, North
Holland Publishing Company, Amsterdam 1977.

[SIMON - 62]

H. A. SIMON
" The Architecture of Complexity"
Proc. American Philosophical Society, Vol.106 N°6,
Décembre 1962.

[SYSTEMS DESIGNERS LIMITED - 85]

DEC VAX-11 PERSPECTIVE,
Language Reference Manual,
Command Reference Manual,
Programmers Guide.
Pembroke Broadway, Camberley, Surrey. Mai 1985.

[TEICHROEW - 77]

D. TEICHROEW, E. A. HERSCHLEY III, Y. YAMAMOTO
"Computer-Aided Software Development, Part2"
Software Reliability, Infotech Int'ltd, Maidenhead,
Berkshire, England, 1977.

[TEICHROEW - 77]

D. TEICHROEW and E. A. HERSCHLEY III,
" PSL/PSA : A Computer-Aided Technique for Structured
Documentation and Analysis of Information Processing
Systems" .
IEEE Trans. Software Engineering, Vol. SE-3 N°1, Janvier
1977.

[VERNADAT - 87]

F. VERNADAT, A. DILEVA, D. BIZIER
"From enterprise analysis to conceptual database design
in production systems"
Conférence INRIA, Systèmes de Production, Avril 1987.

[YEH - 80]

R. T. YEH, P. ZAVE
"Specifying Software Requirements" .
Proc. IEEE, Vol. 68, 1980.

ANNEXES

ANNEXE 1 : GLOSSAIRE DES TERMES EMPLOYES DANS LE MEMOIRE

Action élémentaire :

Synonyme de tâche, fonction, action. Désigne ce que fait un équipement du système de fabrication [REF CH2].

Action générale :

Fonction générale du système de fabrication identifiée lors de l'analyse du procédé de fabrication [REF CH2].

Avant-projet :

Phase du cycle de vie d'un système de fabrication. La phase d'avant-projet nécessite l'analyse du système de fabrication, la définition des solutions techniques, leur évaluation économique et technique. Synonyme : Pré-étude [REF CH1].

Besoin :

Un besoin est un terme général désignant l'ensemble des souhaits (performances fonctionnelles) et des contraintes (budget économique) qui ont été exprimés pour l'automatisation d'une application [REF CH2].

Client :

Personne ou groupe de personnes qui financent l'avant-projet et qui décident des besoins en général [REF CH1].

Cohérence :

Absence de contradictions dans les spécifications définies.
[REF CH2]

Complétude:

Prise en compte de tous les aspects d'un ensemble de spécifications [REF CH2].

Concepteur:

Personne ou groupe de personnes de la société d'ingénierie ou du bureau des méthodes qui répondent à l'appel d'offres d'un client et proposent un système de fabrication [REF CH1].

Conception :

Processus de réflexion aboutissant à la définition complète d'un produit. Dans le cadre de la conception d'un système de fabrication, la conception regroupe deux phases : l'avant-projet et le développement d'un système de fabrication [REF CH1].

Cycle de vie :

Séquence des phases de la vie d'un produit au sens large du terme. Dans le cadre de notre étude, nous nous intéressons au cycle de vie d'un système de fabrication [REF CH2].

Développement :

Se reporter à la définition d'une étude détaillée [REF CH2].

Elément formel :

Représente un type de besoin (souhait ou contrainte) [REF CH2].

Etape :

La phase d'un cycle de vie est constituée d'un ensemble d'étapes. La phase de pré-étude d'un système de fabrication comprend quatre étapes : l'Analyse du Système de Fabrication, la Définition des Solutions Techniques, l'Evaluation Economique et l'Evaluation de Faisabilité Technique [REF CH2].

Etude détaillée :

Spécifications détaillées d'un système de fabrication sur la base du dossier de pré-étude. Synonyme développement [REF CH1].

GUIDE CONCEPTEUR :

Document de base de la méthodologie de conception des systèmes de fabrication.

Il propose une technique d'investigation des informations (liste de thèmes), et, une technique d'évaluation de ces informations (règles de cohérence entre ces thèmes) [REF CH3].

Ingénierie :

Ensemble de modèles, de méthodes et d'outils de conception pour la définition d'un système automatisé optimal dans un contexte industriel [REF INTRODUCTION].

Ingénierie d'avant-projet :

Modèle, méthode et outils d'aide à la conception pour la pré-étude des système de fabrication.

Méthode :

Elle fournit une séquence d'étapes pour guider l'écriture des spécifications du système. [REF INTRODUCTION]

Modèle :

Il fournit une représentation formelle d'un processus. Il est introduit pour faciliter l'analyse du processus. [REF INTRODUCTION].

Modèle de cycle de vie :

Un modèle du cycle de vie structure la description des différentes phases de la vie d'un produit (en l'occurrence un système de fabrication) [REF CH2].

Modèle conceptuel :

Un modèle conceptuel se compose d'un modèle de cycle de vie et de modèles de description des informations (langages de spécification) [REF CH2].

Objet :
Un objet peut être un besoin ou un élément de la solution technique (action générale, action élémentaire, équipement).
[REF CH2]

Outil :
C'est un moyen ou un utilitaire permettant d'être plus performant dans l'accomplissement d'une tâche.
[REF INTRODUCTION]

Partie formelle :
La syntaxe d'un élément formel est constituée d'un ensemble de parties formelles [REF CH2].

Phase :
Décrit une partie du cycle de vie d'un produit. Dans le cadre du cycle de vie d'un système de fabrication, il y a cinq phases : la pré-étude, le développement, la réalisation, l'exploitation et la maintenance [REF CH2].

Pré-étude :
Se reporter à la définition d'un avant-projet.
[REF INTRODUCTION]

Production :
La production recouvre deux activités la conception du système de fabrication, du produit, et, des gammes de fabrication, la fabrication (usiner, assembler, traiter) [REF INTRODUCTION].

Productique :
Synonyme Production Intégrée par Ordinateur (PIO).
La productique concerne les sciences et les techniques de la production.
Les sciences de la production (ingénierie) sont les connaissances, concepts, modèles, méthodes intervenant dans l'analyse, la conception, et la conduite des systèmes de production.
Les techniques de production : concernent les supports technologiques intervenant dans la mise en oeuvre des systèmes de production (mécanique, électronique, ...) [REF CH1].

Spécification :
Description d'un objet en mettant en évidence ses propriétés principales [REF CH1].

Spécification fonctionnelle :
Description des fonctions supportées par les équipements du système de fabrication [REF CH2].

Spécification opérationnelle :
Description des équipements du système de production en fonction de leurs composants, performances, ... [REF CH2].

Station de travail :

Système intégré d'outils informatiques. Dans le domaine de l'ingénierie d'avant-projet, une station de travail est un système intégré d'outils pour la pré-étude des systèmes de fabrication [REF INTRODUCTION].

Système de commande :

Système informatique gérant le pilotage des logiciels de commande de machines unitaires. Un système de commande est associé à un système de fabrication [REF CH3].

Système de fabrication :

Réalise des opérations de transformation sur un produit (usinage, assemblage, ...). Un système de fabrication se compose d'un ensemble de machines, d'automates programmables et de robots [REF CH1].

Système de production :

Ensemble de moyens organisés autour d'une activité de production.

ANNEXE 2 :

ANALYSE FONCTIONNELLE DETAILLEE DES MODULES de TRAITEMENT DYNAMIQUE
DE L'INFORMATION (TDI) ET de l'INTERFACE OPERATEUR (IOP).

1. Description détaillée du module de TDI

L'analyse des fonctionnalités du module TDI, présentées dans le paragraphe 4.5.1., ont conduit à identifier plusieurs modules fonctionnels au niveau de décomposition inférieur.

Ces modules sont les suivants :

- module de gestion de la Base de Données Projet,
- module de sortie de l'information,
- module d'initialisation,
- module de traitement d'un cahier des charges courant,
- module de traitement de l'agenda,
- module de traitement d'erreurs,
- module d'acquisition de l'expertise du concepteur,
- module de traitement des données pour le système de spécification.

Le diagramme suivant permet de visualiser le flux d'informations entre ces différents modules.

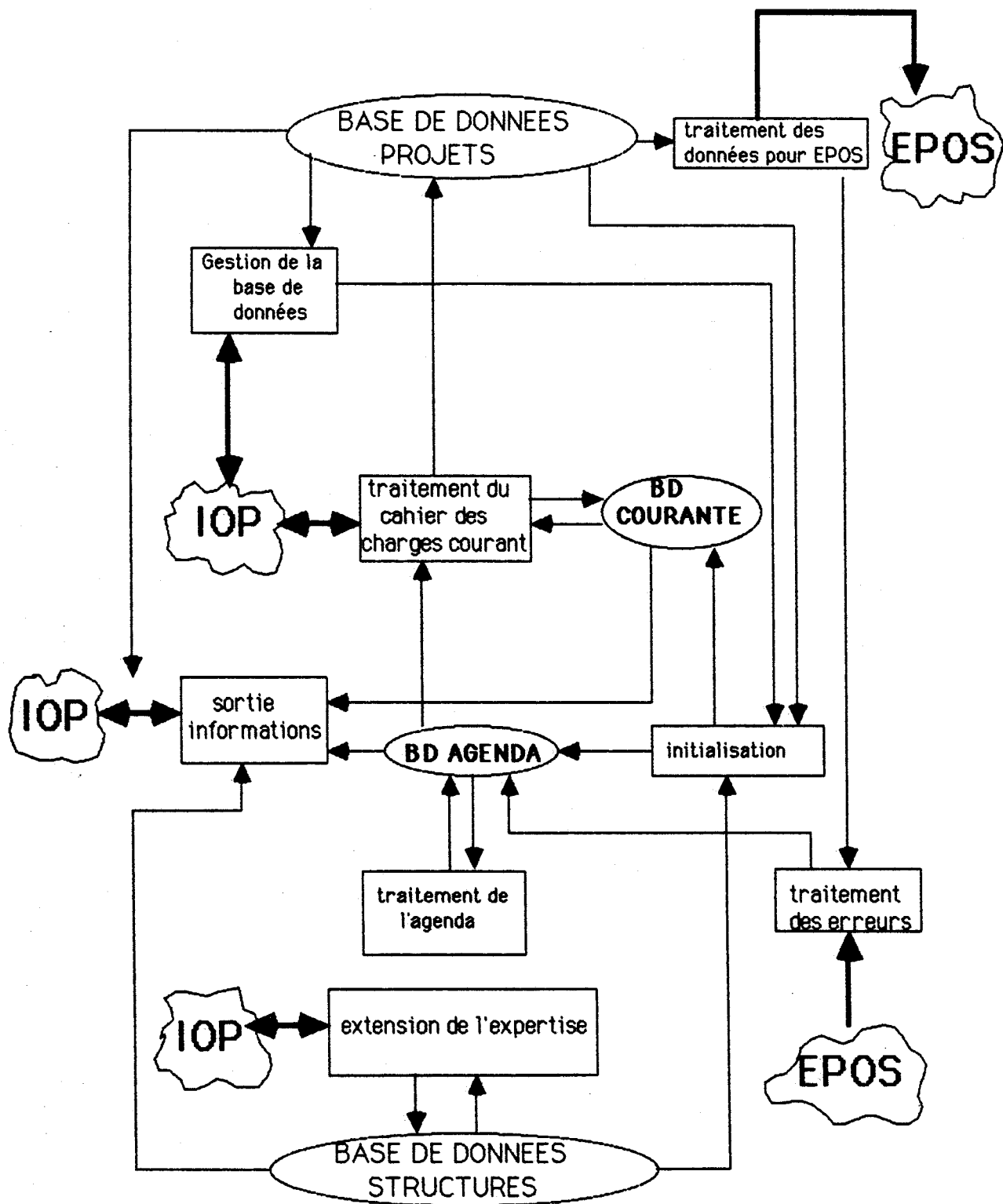


Fig 33 Architecture fonctionnelle du module TDI

2. Description détaillée du module fonctionnel Interface Opérateur

Le module Interface Opérateur a été décomposé en trois modules fonctionnels, après l'analyse des fonctions générales identifiées dans le paragraphe 4.5.2.

Il s'agit des modules suivants :

- module de gestion d'écran,
- module d'interface avec TDI,
- module de gestion du dialogue.

2.1. Module de gestion d'écran

Ce module gère l'interface homme-machine qui consiste en : l'affichage des écrans, la lecture des données entrées par l'opérateur et l'envoi de ces données vers le module de gestion du dialogue.

2.2. Module d'interface vers TDI

Ce module est responsable de l'interface TDI/IOP.
Les choix de l'opérateur et les données saisies à l'écran sont envoyées vers le module TDI qui les traitent.
Les accusés de réception provenant du module de Traitement Dynamique de l'Information sont également pris en compte.

2.3. Module de gestion du dialogue

Ce module est responsable de l'initialisation du dialogue opérateur et du contrôle des données saisies à l'écran par l'opérateur.

Le diagramme, page suivante, représente le flux de données circulant entre ces différents modules.

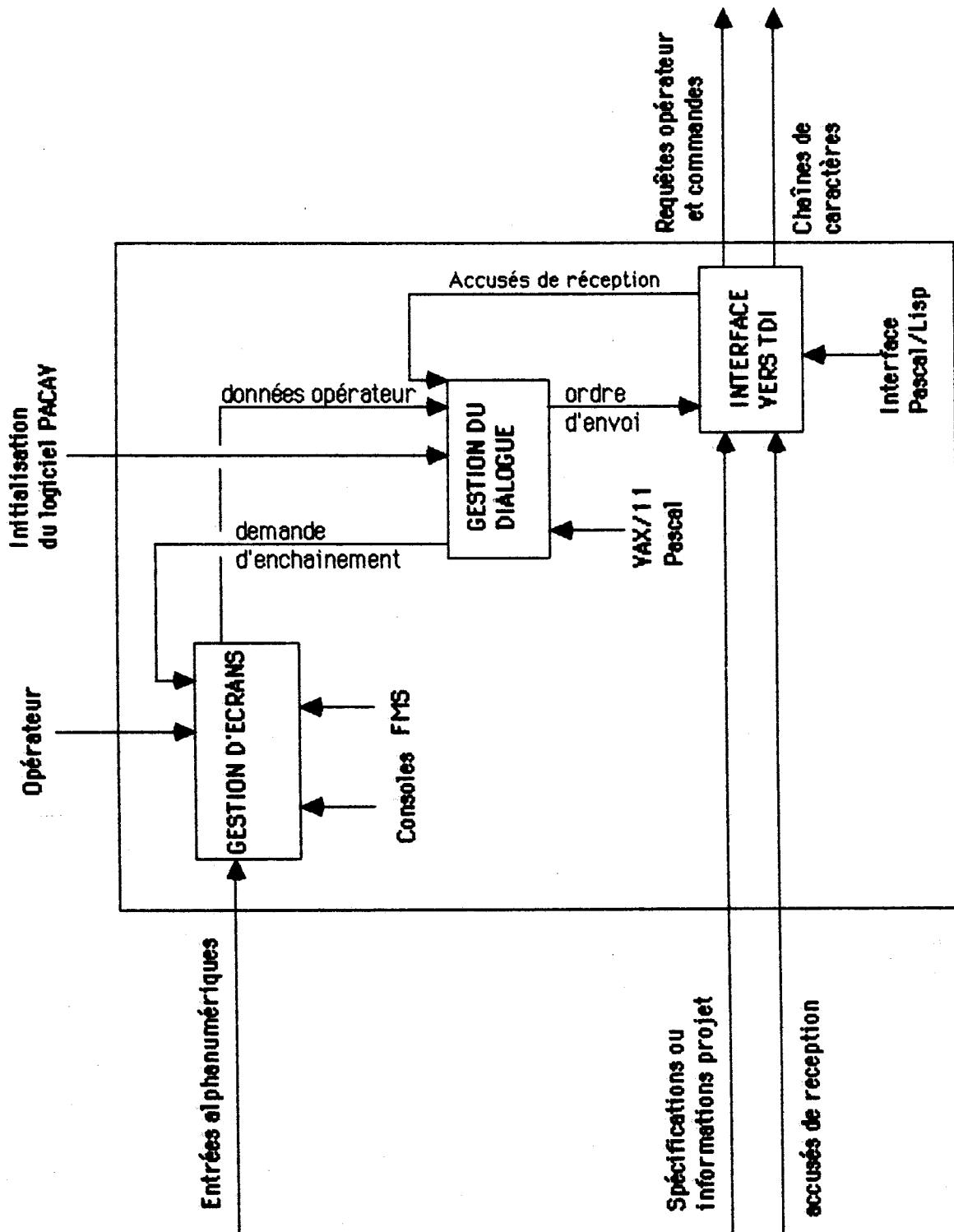


Fig 34 Architecture fonctionnelle du module IOP

ANNEXE 3 :

PRESENTATION DE L'ANALYSE ORGANIQUE SUIVANT LE FORMALISME METHRA1. Buts de l'analyse organique

Le but de l'analyse organique est double :

- guider le concepteur dans la phase d'analyse en lui fournissant un formalisme et des concepts,
- produire une documentation synthétique du logiciel permettant à des réalisateurs qui n'ont pas participé aux phases d'analyse précédentes de s'intégrer plus facilement.

2. Intervenants

Les intervenants au niveau de l'analyse organique constituent le noyau de l'équipe de réalisation. Ils prennent ensuite la responsabilité d'une équipe pour les phases d'analyse organique détaillée et de codage.

3. Principes de l'analyse organique

L'analyse organique s'appuie sur une méthode de décomposition itérative fondée sur un formalisme graphique. Les différents éléments de décomposition sont alors considérés de deux points de vue :

- spécification externe, et,
- conception.

3.1. Principes de décomposition

3.1.1. Eléments de décomposition d'un système

Les éléments de décomposition d'un système sont les suivants :

- interface,
- sous-système,
- tâche,
- module,
- périphérique.

* L'INTERFACE :

est le moyen unique de représentation des communications entre les autres éléments de METHRA.

Elle est toujours réalisée par un sous-système ou un module. METHRA distingue deux types d'interfaces.

L'interface de type canal est dédiée à la communication, l'interface de type accumulateur regroupe tous les autres traitements de données.

Les différents objets accessibles par une interface sont des constantes, des types, des variables, des procédures et des fonctions.

* LE SOUS-SYSTEME :

C'est une unité de structuration. Un sous-système se décompose en sous-systèmes, modules ou tâches.

Le sous-système de premier niveau est un sous-système particulier qui n'est pas issu de la décomposition supérieure. Il est qualifié de système.

* LA TACHE :

C'est l'un des éléments terminaux de la décomposition. Elle représente tous les éléments qui ont une activité autonome par opposition au module qui est passif.

Une tâche communique avec les autres tâches par des interfaces. Une tâche peut donc utiliser des modules ou des sous-systèmes (qui réalisent des interfaces).

* LE MODULE :

C'est l'autre élément terminal de la décomposition. Le module est passif. Sa seule vocation est de réaliser une interface qui sera utilisée par des sous-systèmes, des tâches ou d'autres modules.

* LE PERIPHERIQUE :

Un périphérique représente un système physique. Il peut apparaître au niveau supérieur de la décomposition ou être encapsulé dans un sous-système.

3.1.2. Règles de construction des diagrammes

Il ne faut pas confondre les diagrammes METHRA avec des diagrammes de flux de données. Les liaisons entre les différents éléments indiquent des dépendances. Par contre le diagramme peut être repris pour préciser des flux de données et des synchronisations.

Il existe une règle fondamentale, pour la spécification de logiciels temps-réel : deux sous-systèmes ne peuvent pas avoir de relations sans passer par une interface.

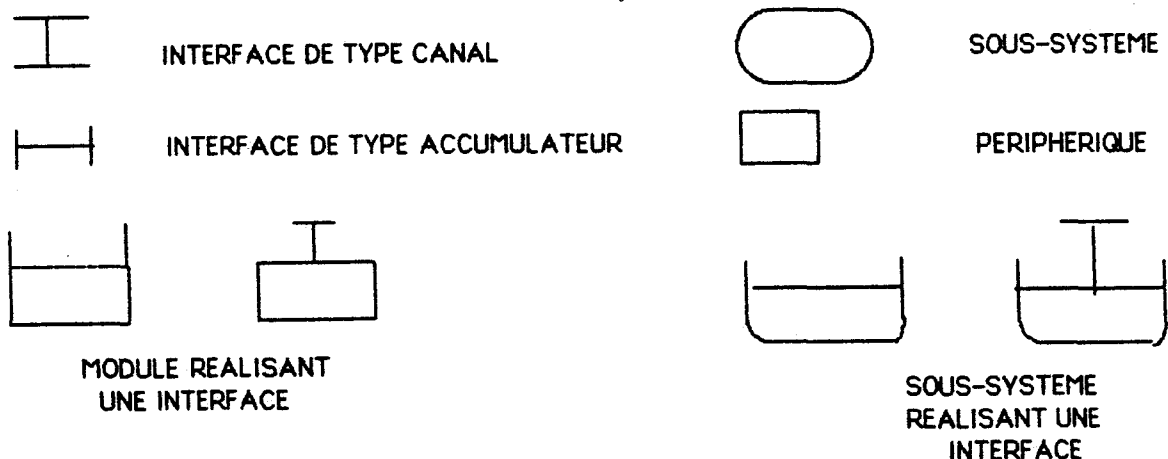
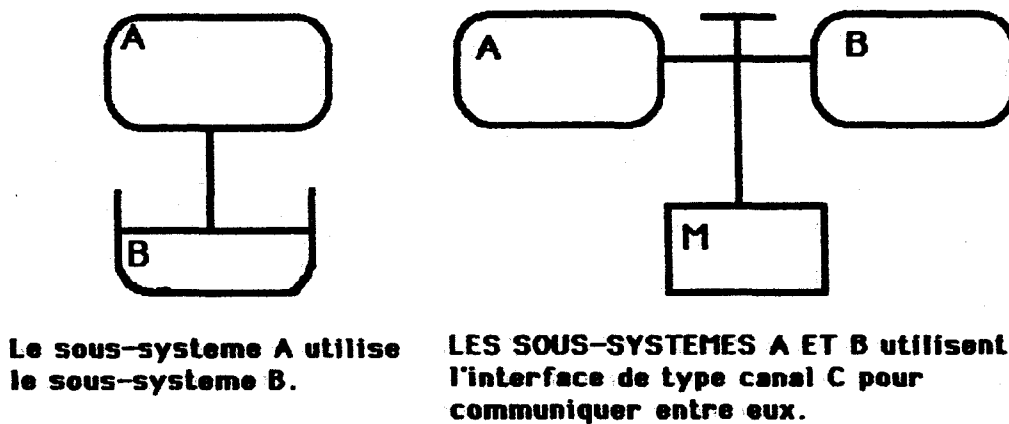
Deux types de relations sont alors possibles :

- si l'un des sous-systèmes réalise une interface de type accumulateur, l'autre sous-système peut utiliser cette interface.
- si deux sous-systèmes doivent échanger des informations sans que l'un soit au service de l'autre, ils doivent utiliser une interface de type canal pour communiquer. Cette interface est elle-même réalisée par un module ou un autre sous-système.

La distinction entre ces deux possibilités reste subjective. Le choix de l'une d'entre elles est un des points clés de l'analyse organique.

Une interface ne se limite pas à la définition de procédures et de fonctions. Elle contient des éléments tels que des constantes, des types et des variables. Une interface peut aussi dépendre d'une autre interface. Un formalisme est prévu à cet effet.

Si une interface est utilisée par beaucoup d'éléments, il peut être délicat de tracer toutes les liaisons. Un système de renvoi par étiquette doit être utilisé.

REPRESENTATION GRAPHIQUE DES DIFFERENTS ELEMENTS DE LA DECOMPOSITIONEXEMPLES DE DIAGRAMMES METHRAFig 35 Formalisme graphique METHRA

ANNEXE 4

PRESENTATION DU MODULE DE PROGRAMMATION HORS-LIGNE1. Introduction

La finalité du module de programmation hors-ligne, réalisé dans le cadre du projet ESPRIT, est la représentation virtuelle d'une cellule robotisée.

Cela consiste à définir graphiquement la cellule, développer des programmes de robots, pour ensuite les exécuter, les vérifier et les télécharger dans un contrôleur robot réel.

Les éléments de la cellule robotisée sont modélisés d'après un outil CAO (en l'occurrence EUCLID).

Seuls les aspects géométriques et cinématiques sont pris en compte.

2. Structure du module robotique

Il est composé de cinq modules logiciels réalisant la modélisation, l'arrangement, la programmation, l'émulation et la simulation d'une cellule robotisée d'assemblage.

2.1. Module de modélisation

Ce module permet de représenter numériquement les éléments de la cellule (robot, pince, composants).

2.2. Module d'arrangement

Ce module permet de décrire la configuration de la cellule en spécifiant la position relative des objets les uns par rapport aux autres.

2.3. Module de programmation

Le module de programmation permet de décrire les tâches des équipements dans un langage robotique de haut-niveau appelé ROLS.

Il a une syntaxe PASCAL avec des instructions robotiques spécifiques pour la description d'opérations d'assemblage.

2.4. L'émulation

Le module d'émulation permet l'exécution du programme robot.

2.5. Module de simulation

Ce module permet de visualiser le comportement dynamique de la cellule et de ses équipements. En particulier, le concepteur peut vérifier l'absence de collisions, visualiser la trajectoire et estimer la valeur du temps de cycle.

L'enchaînement de ces modules est détaillé sur le diagramme ci-dessous :

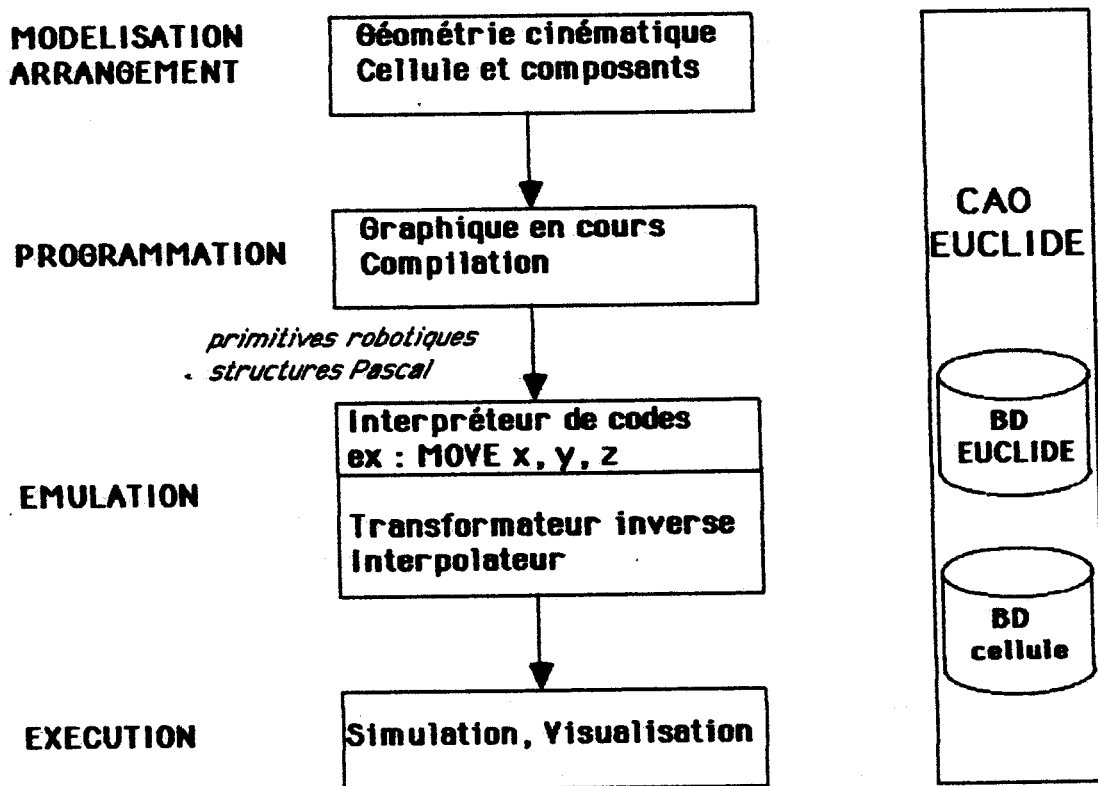


Fig 36 Module de programmation hors-ligne

*

*

*