

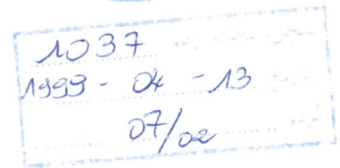
# THESE

DE MAGISTER  
EN ELECTRONIQUE

Option : Signaux et Systèmes

Présentée par

M<sup>r</sup> Khalid HACHEMI



## THEME

**Mise au point d'un logiciel pour la conduite  
et l'ordonnancement d'un atelier flexible**

Soutenue en Avril 1999 devant la commission d'examen :

<u>Président :</u>	F. BEREKSI - REGUIG	M.C.	U.A.B. Tlemcen
<u>Rapporteurs :</u>	N. GHOUALI	Professeur.	U.A.B. Tlemcen
	Z. SARI	C.C.	U.A.B. Tlemcen
<u>Examineurs :</u>	B. CHERKI	M.C.	U.A.B. Tlemcen
	C. KARA TERKI	D <sup>r</sup> Ingénieur.	U.A.B. Tlemcen

## Dédicaces

Je dédie ce travail à :

- Mes parents.
- Mes sœurs : Nassima, Faïza et Karima.
- Mes Frères : Nor-eddine et Ahmed.
- Mes amis.
- Mes professeurs.
- Et à tous ceux qui me sont chères.

*Khalid.*



## Remerciements.

Le travail que nous présentons dans cette thèse, a été effectué au laboratoire de robotique (pôle Bel – horizon et pôle Chetouane). Je remercie l'ensemble des membres de l'équipe qui m'ont amicalement et chaleureusement intégré au sein de l'équipe.

Je tiens tout d'abord à remercier Monsieur Nouredine GHOUALI, Professeur à l'université de Tlemcen, qui ma proposé un domaine de recherche fort intéressant et pour m'avoir orienté et aidé dans mon travail, par ses recommandations et ses précieux conseils malgré toutes ses charges. Je lui en suis très reconnaissant.

Je tiens à remercier également, Monsieur Zaki SARI, chargé de cours à l'université de Tlemcen, pour m'avoir guidé encore une fois, tout au long de ce travail et pour avoir su, répondre à mes attentes et orienter mes investigations.

Je tiens à exprimer mes sincères reconnaissances, à Monsieur F. BEREKSI-REGUIG, Maître de conférences à l'université de Tlemcen, qui m'a fait l'honneur d'accepter la présidence de Jury de cette thèse.

Je voudrais exprimer ma gratitude à Monsieur, B. CHERKI, Maître de conférences à l'université de Tlemcen, pour l'honneur qu'il nous a fait de s'intéresser à ce travail et de le juger.

Je tiens à remercier, Monsieur C. KARA TERKI, Docteur Ingénieur à l'université de Tlemcen, pour avoir accepté en tant que membre de Jury d'examiner le présent travail.

Je tiens à remercier, mes collègues de l'institut d'électronique pour leur soutien et leurs encouragements. Je nommerai, M. HADJILA, M. BOUSAHLA, S. ASOUS, B. OULD-ABBAT, F. DERRAZ, Mme A. CHOUKCHOU-BRAHAM.

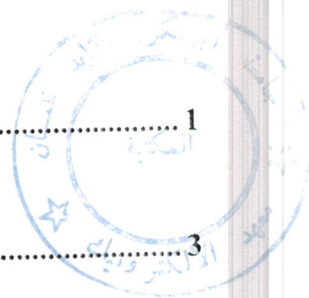
Je tiens ici, à exprimer ma gratitude à Monsieur S. HAMDAN pour son aide, pendant mon séjour au CDTA.

Je tiens à remercier, mon frère NOR-EDDINE, pour son encouragement permanent, et aux moments les plus difficiles. Je tiens à lui en marquer ici toute ma profonde gratitude.

Que toute personne ayant participé de près ou de loin à l'accomplissement de ce travail, trouve ici l'expression de nos profonds remerciements.

Je tiens, pour finir, à remercier toute ma famille pour sa patience, son sacrifice et son encouragement à mon égard.

*Sommaire*



INTRODUCTION.....	1
CHAPITRE 1. LES SYSTEMES FLEXIBLES DE PRODUCTION.....	3
1.1. INTRODUCTION.....	3
1.2. DÉFINITION D'UN SYSTÈME FLEXIBLE DE PRODUCTION.....	3
1.3. LES ÉLÉMENTS DES SYSTÈMES FLEXIBLES DE PRODUCTION.....	4
1.3.1. <i>Les stations de travail</i> .....	4
1.3.2. <i>Équipements de transport dans les Systèmes Flexibles de Production</i> .....	8
1.3.3. <i>Les systèmes de stockage</i> .....	11
1.4. LES SYSTÈMES DE CONTRÔLE.....	12
1.5. LES SYSTÈMES DE COMMUNICATIONS.....	15
1.6. LA FLEXIBILITÉ.....	17
1.7. TYPES DE SYSTÈMES FLEXIBLES DE PRODUCTIONS.....	18
1.8. MESURE DE PERFORMANCES.....	18
CHAPITRE 2. LA CELLULE FLEXIBLE AUTOMATISÉE DE L'E.M.N. : UNE ÉTUDE DE CAS.....	20
2.1. INTRODUCTION.....	20
2.2. PRÉSENTATION DE L'INSTALLATION PHYSIQUE.....	20
2.2.1. <i>Descriptif des produits fabriqués</i> .....	20
2.2.2. <i>Palettes</i> .....	21
2.3. VUE D'ENSEMBLE DE LA CELLULE FLEXIBLE.....	22
2.4. ARCHITECTURE DE L'AUTOMATISME.....	23
2.4.1. <i>Analyse de la partie commande</i> .....	24
2.4.2. <i>Synthèse de la commande</i> .....	26
2.5. CONCLUSION.....	34
CHAPITRE 3. MODELISATION ET SIMULATION DES SYSTEMES A EVENEMENTS DISCRETS.....	35
3.1. SYSTÈMES À ÉVÉNEMENTS DISCRETS.....	35
3.2. QUELQUES CONCEPTS FONDAMENTAUX.....	35
3.2.1. <i>Objet d'un système</i> .....	35
3.2.2. <i>L'état du système</i> .....	36
3.2.3. <i>Événements</i> .....	36
3.2.4. <i>Activité</i> .....	36
3.2.5. <i>Processus</i> .....	36
3.3. MODÉLISATION.....	37
1) <i>La modélisation du type file d'attente</i> .....	37
2) <i>La modélisation par réseau de Petri (RdP)</i> .....	39
3.4. SIMULATION.....	48
3.4.1. <i>Étapes d'une modélisation- simulation</i> .....	48
3.4.2. <i>Les différentes approches de la simulation à événements discrets</i> .....	50
3.5. LA SIMULATION APPLIQUÉE AUX SYSTÈMES FLEXIBLES DE PRODUCTION.....	53
3.5.1. <i>Différents champs d'utilisation de la simulation dans les systèmes de production</i> .....	53
3.5.2. <i>Avantages de la simulation</i> .....	54
3.5.3. <i>Inconvénients de la simulation</i> .....	54
3.5.4. <i>Le besoin à une simulation basée sur l'animation graphique</i> .....	55



3.5.5. Limitations de l'animation graphique dans l'interprétation des résultats.....	56
<b>CHAPITRE 4. CONCEPTION ET DEVELOPPEMENT DU LOGICIEL.....</b>	<b>57</b>
4.1. INTRODUCTION .....	57
4.2. DÉFINITIONS .....	57
4.3. TECHNIQUE DE CONCEPTION PAR OBJET .....	58
4.4. LE SYSTÈME DE DÉVELOPPEMENT DELPHI .....	59
4.5. STRUCTURE GÉNÉRALE DU LOGICIEL.....	59
4.5.1. Bases de données du système.....	60
4.5.2. Simulation.....	60
4.6. SIMULATION AVEC ANIMATION.....	61
4.7. PROGRAMMATION CONCURRENTTE ET PROGRAMMATION TEMPS RÉEL .....	62
4.7.1. Introduction .....	62
4.7.2. Gestion de la simulation animée.....	64
4.7.3. Présentation de quelques problèmes.....	64
4.8. SIMULATION SANS ANIMATION GRAPHIQUE .....	74
4.9. EVOLUTION DE LA SIMULATION .....	75
<b>CHAPITRE 5. FONCTIONS DU LOGICIEL (MANUEL D'UTILISATION) .....</b>	<b>79</b>
5.1. INTRODUCTION .....	79
5.2. MENU PRINCIPAL.....	79
5.3. BASE DE DONNÉES DU SYSTÈME.....	80
5.3.1. Bases de données d'entrée (données du modèle).....	80
5.3.2. Bases de données de sortie (résultats).....	83
5.4. SIMULATION .....	86
5.4.1. Simulation avec animation .....	86
5.4.2. Simulation sans animation.....	89
5.5. ORGANIGRAMMES .....	89
5.6. CONCLUSION.....	93
<b>CHAPITRE 6. APPLICATIONS DU LOGICIEL .....</b>	<b>94</b>
6.1. INTRODUCTION .....	94
6.2. EXEMPLES.....	94
6.2.1. Exemple 1 .....	95
6.2.2. Exemple 2 .....	96
6.2.3. Exemple 3 .....	98
6.2.4. Exemple 4 .....	100
6.3. CONCLUSION.....	101
<b>CONCLUSION.....</b>	<b>102</b>
<b>BIBLIOGRAPHIE .....</b>	<b>104</b>

## *Introduction*



## Introduction

La productique appliquée à tout ce qui concerne la production de biens et de services, est l'une des bases de l'économie des pays industrialisés. Elle constitue la clé de la compétitivité et se fonde sur la production intégrée par ordinateur. La productique regroupant l'ensemble des méthodes et techniques de l'ingénierie de la production, de l'étude à la fabrication, a impliqué un remaniement complet de la manière de produire en fonction des critères que sont la qualité, l'adaptabilité et la compétitivité.

L'évolution des besoins d'une part, et l'évolution des technologies d'autre part, ont présidé à l'apparition de systèmes de plus en plus complexes. Ainsi, et grâce au développement des machines à commandes numériques et aux progrès technologiques, les systèmes de production sont devenus progressivement flexibles, c'est à dire, capables de s'adapter à des fabrications variables et diversifiées.

Malheureusement, ces systèmes sont devenus complexes, ce qui rend difficile leur dimensionnement, leur automatisation et leur gestion. De plus l'expérience montre qu'un investissement en automatisation conduit souvent à remettre en cause le processus de fabrication. Il est devenu donc primordial de maîtriser cette automatisation, en s'insérant dans un contexte économique. Les étapes non productives de la vie d'un système automatisé comme sa conception, sa mise au point, sa maintenance, doivent être raccourcies, rendues plus rationnelles, moins coûteuses. En effet, toute erreur de conception se solde par des dysfonctionnements du système qui doivent être ensuite corrigés à grand frais.

La phase de conception inclut les spécifications fonctionnelles, la modélisation et l'évaluation du comportement du système. Pour aider le concepteur dans ces tâches, la simulation constitue un moyen important, permettant d'assister le concepteur pour la prise des décisions. La simulation numérique est l'expérimentation sur un modèle, c'est une procédure de recherche scientifique qui consiste à réaliser une reproduction artificielle (logicielle) du phénomène que l'on désire étudier. La simulation ainsi considérée peut être motivée, soit par le fait de l'impossibilité de recourir à l'expérimentation directe (contrainte budgétaires, physiques...), soit que l'on ne dispose pas d'outil théorique efficace permettant d'étudier le comportement du système.

De nos jours, la simulation constitue une alternative intéressante pour l'étude des systèmes flexibles de production (FMS). Dans plusieurs travaux, l'utilisation de la simulation est apparue plus favorable par rapport aux méthodes analytiques qui souvent échouent à capturer les interactions complexes d'un FMS particulier [Stecke et Solberg, 81 ; Erickson et al, 87 ; Wu et Wysk, 89]. De plus, la simulation animée est devenue un moyen très important dans l'application de la simulation à un système réel. Cette représentation dynamique de l'exécution du modèle, peut fournir des perceptions précieuses du comportement du système.

Le présent travail, porte sur le développement d'un logiciel de simulation d'une cellule flexible d'assemblage. Cette cellule est un système à événements discret (SED) ;

c'est un système dynamique dont l'espace des états est discret et dont l'évolution est déterminée par l'occurrence d'événements.

Le Grafcet est un puissant outil de description qui offre un grand pouvoir de concision dans la modélisation des systèmes logiques séquentiels. Cet outil va être utilisé avantageusement pour la synthèse des automatismes de fonctionnement du système étudié, et par la suite dans l'implémentation du programme de simulation.

Nous allons aborder dans un premier chapitre, les systèmes flexibles de production et leurs principales caractéristiques. Le chapitre 2 est consacré à l'étude détaillée d'un cas de système flexible de production, il s'agit de la cellule flexible d'assemblage. Les approches utilisées dans la modélisation et la simulation des systèmes à événements discrets ainsi que les champs d'utilisation de la simulation feront l'objet du troisième chapitre.

Le chapitre 4 débute par la présentation de l'approche orientée objet pour la conception des logiciels, et se termine par l'introduction des techniques de la programmation concurrente nécessaire à la réalisation de la simulation. Ceci permettra de mettre en exergue les problèmes soulevés lors de la programmation des systèmes concurrents.

Les fonctions réalisées par le logiciel seront présentées dans le chapitre 5. Finalement l'exploitation du logiciel pour traiter des exemples concrets sera décrite dans un sixième et dernier chapitre.





## 1.1. Introduction

Un système de production a pour but d'apporter une valeur ajoutée partant de matériaux, de pièces, de sous ensembles,... il élabore des produits de valeur supérieure et qui peuvent être :

- Soit des produits finis, directement commercialisés.
- Soit des produits intermédiaires servant à la réalisation des produits finis.

Cependant, dans un monde industriel en pleine évolution, caractérisé par une concurrence de plus en plus accrue, de nouvelles exigences sont apparues ; tel que :

- La recherche de coûts plus bas pour le produit à fabriquer.
- La recherche d'une meilleure qualité du produit.
- La réalisation d'opérations impossibles à contrôler manuellement.
- La diversification des types de produits fabriqués.

C'est au centre de ces préoccupations ayant comme objectif essentiel d'aider à la compétitivité globale du produit, que les Systèmes flexibles de production (Flexible manufacturing systems « FMS ») ont fait leurs apparitions.

Ces systèmes sont conçus pour produire en petite et moyenne quantités et à des coûts minimums, une variété de produits avec des temps de préparation des machines et de changements d'outils minimums. Leurs structures leur permettent de fabriquer une très large gamme de produits et donc d'être moins sujets à un crash économique.

En plus, ces systèmes flexibles de production visent à améliorer la productivité, et à s'adapter à n'importe quel changement des types de produits à fabriquer pour suivre les variations du marché. Ainsi, la flexibilité peut être définie comme la possibilité de s'adapter à un large éventail d'environnements différents.

Contrairement aux systèmes conventionnels de production composés de machines rigides (ne fabriquant qu'un seul type de produit), les systèmes flexibles de production, pour faire face au nouveau concept de diversification de types de produits à manufacturer, et aux fluctuations de leurs demandes, sont équipées de machines à commande numérique flexibles, susceptibles d'opérer sur plusieurs variantes de produits.

## 1.2. Définition d'un système flexible de production

Un atelier flexible permet la production automatique de pièces de types divers et en quantités variables. Les opérateurs n'interviennent pas directement dans le processus de fabrication et limitent essentiellement leurs interventions à l'entretien ; l'ordonnancement de la production est géré par un système informatique [Widmer, 91].

Askin [Askin, 93] définit un système flexible de production (Flexible Manufacturing System, FMS) comme étant un ensemble de machines à commande numérique, de stations de travail connectées par un système de transport automatisé, le tout commandé par un ordinateur.

### 1.3. Les Eléments des systèmes Flexibles de production

Les systèmes flexibles de production peuvent être caractérisés par les points suivants :

- Machines reprogrammables à commande numérique.
- Changement d'outils automatisé.
- Système de transport automatisé.
- Chargement et déchargement automatisé.
- Contrôle coordonné.

Les systèmes flexibles de production nécessitent des machines complexes et chères, mais leur nombre est réduit par rapport à d'autres systèmes de production. Globalement une économie dans le coût et l'espace de l'ordre de 30% peut être réalisée [Askin, 93].

Les systèmes flexibles de production sont constitués d'un certain nombre d'équipement qui peuvent être divisés en quatre grandes familles :

- Les stations de travail, englobant les machines à commande numérique, les systèmes de changement d'outils, les palettes, les supports, les systèmes d'inspection...
- Les systèmes de manutention et de transport, comprenant les systèmes de transport, les robots manipulateurs...
- Les systèmes de stockages, renfermant les aires de stockages, les systèmes de chargement/déchargement.
- Les systèmes de contrôle et de communication.

#### 1.3.1. Les stations de travail

Une station de travail est composée d'une machine à commande numérique, d'un magasin d'outils avec un système automatique de changement d'outils, d'un magasin de palettes et de supports, d'un système automatique de chargement/déchargement des palettes dans la machine, d'un système d'inspection et d'un système de contrôle.

##### a) Les machines

La principale composante des systèmes flexibles de production, est incontestablement la machine. C'est au niveau de la machine que le produit acquiert une valeur ajoutée lors de sa fabrication par usinage (machines outils) ou par transformation (moulage, injection, traitement de surfaces...) :

- Les machines outils, utilisées dans les systèmes flexibles de production, sont en général, des machines à commande numérique. Elles sont commandées par un programme mémorisé dans un contrôleur à base de microprocesseur. Le programme est développé pour chaque produit. En changeant d'outils et de programme de commande, une machine est en mesure de réaliser une grande variété d'opérations sur des produits de forme, de taille et de matériaux différents.



- Les machines d'injection ont un moule, un système d'injection, un creuset chauffé (plastique, zamac, aluminium...). Le moule s'ouvre et se ferme automatiquement, les pièces sont chassées du moule par des éjecteurs, l'injection se fait par un système de piston à haute pression.
- Pour le traitement de surfaces, un ensemble de bacs est installé en chaîne, reliés par un ou plusieurs chariots automatisés. A l'une des extrémités de la chaîne se trouve la station d'entrée/sortie.
- Un nombre important de types de machines existe dans les systèmes de production. Nous citerons à titre d'exemple : les laminoirs, les presses, les fours, les plieuses, les machines d'assemblages, les machines d'inspection. Les machines d'inspection sont conçues pour contrôler la qualité des produits fabriqués. Les produits de qualité insuffisante sont soit rejetés, soit renvoyés vers le système pour des opérations supplémentaires.

#### b) Les palettes et supports

Les palettes et supports sont des composants essentiels d'un système flexible de production. Les palettes sont des pièces qui servent d'interface entre les machines et les produits pendant la fabrication ou l'assemblage. Les produits, solidaires de leur support, se déplacent dans le système de production sur les palettes.

La palette doit pouvoir se déplacer dans tout le système de production, elle doit, donc être compatible avec tous les éléments du système flexible de production. Les palettes sont de forme carrée, rectangulaire ou circulaire, elles peuvent être fabriquées en acier, en bois ou en plastique.

Le support soutient et localise le produit. La forme des supports dépend du type de produit. Les supports dédiés à un produit ne sont pas utilisables pour un autre. De ce fait, les supports provoquent une augmentation des coûts de production, particulièrement pour les produits en petits lots. Des tentatives de fabrications de supports standards, autorisant leur utilisation pour une plus ou moins grande variété de produits, ont permis d'augmenter la flexibilité du système et de diminuer les coûts de productions. L'une des techniques de fabrication de supports flexibles, est de réaliser les supports en assemblant un groupe de pièces de formes standards (formes en V, en □, en T...).

#### c) Changeur automatique d'outils

Un changeur automatique d'outils est un système capable de sélectionner l'outil approprié, pour une opération donnée, à partir d'un magasin d'outils et de le fixer sur la tête porte outils de la machine.

Il est possible d'utiliser des magasins d'outils centralisés pour tout le système. Dans ce cas, des zones tampons de stockage d'outils sont disposées au niveau de chaque station de travail, et un système de transport est utilisé pour transférer les outils du magasin central vers les stations de travail. Ce système présente

l'avantage d'une réduction du nombre total d'outils dans le système, mais nécessite un système de transport plus élaboré et présente des temps de changement plus longs.

#### d) Magasin et chargement/déchargement des palettes

Lorsque les produits arrivent à la station de travail sur les palettes, ils sont réceptionnés dans un magasin en forme de carrousel, de ligne ou autre. Les produits sont ensuite chargés sur la table de travail de machine à l'aide d'un système automatique de chargement/déchargement, qui peut être, par exemple un robot manipulateur. La machine, à l'aide d'un système de détection (code barre...) identifie le produit, charge l'outil et le programme correspondants, et procède à la fabrication.

#### e) Système automatique d'inspection

L'inspection des produits a pour but d'assurer que ces derniers répondent aux spécifications exigées. Un système d'inspection est vital pour la fabrication de produits de qualité.

On peut définir une inspection comme une opération où la totalité ou une partie du produit est comparée avec le produit normalisé. L'inspection automatique nécessite l'intégration de cette dernière dans le cycle de production. Ce qui permet de minimiser les transports et de corriger rapidement les défauts.

La fig. 1.1 donne un modèle RdP (Réseau de Pétri) d'une machine d'inspection dédiée à un produit. La transition  $t$  représente le test. A l'issue de ce test, la transition  $U_1$  est franchie si le produit est conforme. Si la qualité est jugée trop mauvaise, la transition  $U_2$  est franchie. Si la qualité n'est pas suffisante mais peut être améliorée grâce à des opérations supplémentaires, alors la transition  $U_3$  est franchie. Les places  $O_1$ ,  $O_2$  et  $O_3$  représentent respectivement le stock de produits de bonne qualité, le stock de rebuts et le stock de produits à recycler. La machine est libre, en fonctionnement ou en attente d'évacuation du produit testé selon que le jeton se trouve dans les places  $F$ ,  $Q$  ou  $W$ .

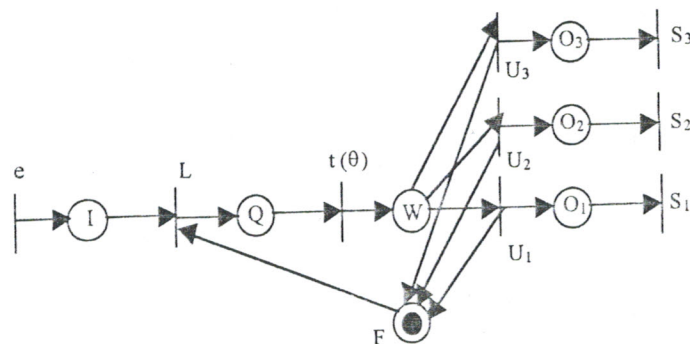


Fig. 1.1 : Réseau de Petri d'une machine d'inspection. [Proth et al, 95]



L'inspection automatique requiert différents types de technologies : informatique, électronique, optique, ultrason, métrologie, photographie... Dans le passé, l'inspection se faisait après la production de tout le lot. Cette méthode provoquait des pertes considérables : des produits défectueux, en plus ou moins grandes quantités, devaient être, soit retravaillés, soit rejetés.

De plus, les corrections du processus de fabrication ne pouvaient être faites que pour les lots suivants. Pour remédier à ces problèmes, deux techniques d'inspection ont été élaborées.

- Inspection pendant l'opération : Cette inspection se fait pendant la fabrication, la pièce est positionnée, sur la table de la machine, et jaugée pendant la fabrication. Cette technique n'est pas très précise, mais elle est souvent suffisante et nécessite un faible investissement de départ. De plus, elle permet d'informer le contrôleur de la machine sur les corrections à apporter au processus de fabrication (usure d'outils, positionnement...).
- Inspection après l'opération : Après l'exécution de l'opération, la pièce est transportée vers une machine de mesure qui vérifie les dimensions, l'état des surfaces..., de la pièce. Après vérification, la pièce est orientée vers la destination appropriée (acceptée, rejetée ou retravaillée). De plus, des informations sont transmises aux machines outils pour la correction des défauts. Cette technique est beaucoup plus précise que la précédente, mais elle nécessite un investissement très important.

#### f) Système de contrôle.

Une station de travail est un système complexe composé de plusieurs équipements interdépendants. Le système de contrôle a pour mission de coordonner et superviser un certain nombre de fonctions dont :

- Le chargement du produit sur machine, l'exécution de l'opération et le déchargement.
- Surveiller et protéger le système des perturbations (mauvais chargement, casse d'outil, casse de support...).
- Veiller au respect des priorités des différents produits.
- Maximiser l'utilisation de la machine et minimiser la quantité d'équipements annexes (outils, supports, palettes...).

Le système de contrôle a d'autres fonctions, parmi elles, il doit surveiller l'usure des outils pour les remplacer avant leurs casses. Ceci est réalisé par l'utilisation de plusieurs détecteurs qui permettent de déceler des :

- Changements dans les dimensions des produits.
- Changements dans la géométrie des outils.
- Augmentations des vibrations et du bruit.
- Détériorations de l'état de surface.
- Changements dans le couple et la température de coupe.

Ces détections se font en ligne et les corrections peuvent intervenir tout de suite après la fin d'une opération.

### 1.3.2. Equipements de transport dans les Systèmes Flexibles de Production

#### a) Introductions.

Les équipements de transport servent à déplacer les produits et éventuellement les outils entre les machines suivant les chemins planifiés pendant la conception du FMS.

Il existe différents moyens de transport et différentes stratégies. On peut par exemple déplacer les produits par petits lots au lieu que ce soit par unités. Le choix de l'une de ces méthodes dépend du système global. On favoriserait le déplacement à l'unité lorsque les distances entre stations sont petites. Par contre, pour de grandes distances, le déplacement par lots convient mieux.

#### b) Classification des équipements de transport.

Il existe plusieurs manières de classifications des équipements de transport [Viswanadham, 92] :

- Suivant le type d'installation (sur le sol, suspendu ou aérien).
- Suivant le type de commande (manuel, contrôlé, automatique).
- Suivant le type de déplacement (Chemins fixes ou routage flexible).
- Suivant le type de fonctionnement :
  - Dans les systèmes synchrones, toutes les charges dans le système, se déplacent simultanément à la même vitesse et à intervalles réguliers.
  - Dans les systèmes asynchrones, les charges se déplacent indépendamment les unes des autres.

#### c) Différents types d'équipements de transport

Il existe différents types d'équipements de transport, avec des degrés d'automatisation très variables. Nous pouvons citer :

- Les convoyeurs : Ils sont fréquemment utilisés en pratique pour le déplacement des produits lourds ou des produits à fort débit. Il existe différents types de convoyeurs : convoyeur à tapis roulant, à chaîne, à rouleau... Ce sont des équipements de transports synchrones. Ils peuvent être en forme de ligne (unidirectionnel ou bidirectionnel) ou en boucle fermée (déplacement continu ou recirculation). Ils sont utilisés dans les chaînes de montages, les carrières, les cimenteries, les aéroports... Un convoyeur peut être décomposé en cantons. Un canton est une portion de convoyeur qui peut être occupée par au plus un produit à la fois. La figure 1.2. donne le modèle réseaux de Pétri (RdP) d'un convoyeur qui comporte quatre cantons. Dans ce modèle, les transitions  $t_1$ ,  $t_2$ ,  $t_3$ ,  $t_4$  et  $t_5$  représentent les déplacements des produits d'un canton à un autre.  $\Delta$  représente le temps nécessaire au déplacement.



Chaque couple de places ( $Q_i, S_i$ ) contient un seul jeton. Le canton  $i$  est libre si le jeton figure dans la place  $Q_i$  et il ne l'est pas si le jeton figure dans  $S_i$ . Le marquage de la figure 1.2 correspond au cas où les cantons 3 et 4 seraient occupés et les cantons 1 et 2 sont libres.

- Les grues et palans :

Ils sont utilisés pour les déplacements verticaux et/ou horizontaux de produits de diverses tailles et poids dans un espace limité. Ils sont utilisés principalement dans les chantiers de construction, les ports...

- Les véhicules industriels :

Les véhicules autoguidés (Automated guided vehicles, AGV) sont les véhicules industriels les plus élaborés avec un degré d'automatisation très important.

- Les robots industriels : Ils sont utilisés comme équipements de chargement/déchargement et/ou déplacement de produits d'un point à un autre dans un espace limité. De plus, ils peuvent être utilisés pour réaliser des opérations simples comme le soudage à point ou la peinture. Ils sont en général commandés numériquement, et sont compatibles avec les machines à commande numérique. Les robots sont classifiés de différentes manières. Parmi elles, la classification par catégorie [Viswanadham, 92] :

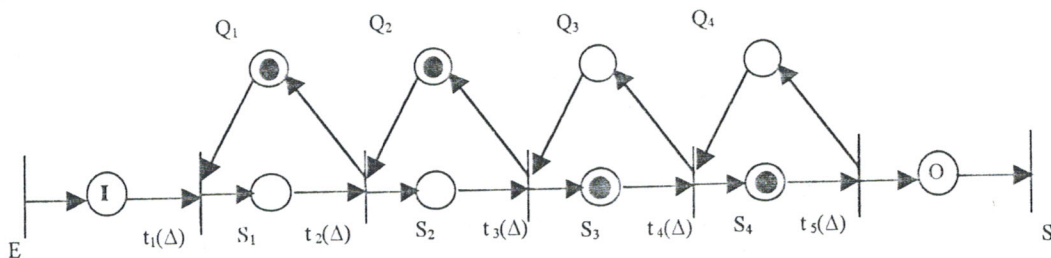


Fig. 1.2. Modèle RdP d'un convoyeur quatre cantons. [Proth et Xie, 95]

- \* Les robots de première catégorie sont des robots à séquence fixe, sans détecteurs, ni puissance de calcul.
- \* Les robots de seconde catégorie sont des robots dotés de détecteurs externes, et d'une certaine puissance de calcul. Cela leur permet de modifier leurs actions en réponse à de petites perturbations. Ils sont capables de distinguer les différents produits (taille, forme...) et de les classer par type.
- \* Les robots de troisième catégorie sont des robots intelligents qui sont capables de collecter des informations à travers des capteurs, de traiter ces informations dans des calculateurs puissants et de prendre les décisions qui s'imposent.

Les robots industriels ont un large éventail d'applications dans les systèmes flexibles de production. Ils peuvent s'adapter à l'environnement à travers leurs capteurs et leurs contrôleurs programmables.

#### d) Les véhicules autoguidés (Automated guided Vehicles).

Les AGV sont des moyens de transport très flexibles. Ils permettent d'assurer le déplacement d'un produit entre un nombre défini de destinations. Les AGV présentent plusieurs avantages.

1. Grandes flexibilité de routage.
2. Possibilité d'automatisation totale.
3. Transport de charges très importantes (jusqu'à 50 tonnes) [Viswanadham, 92].

Il existe différents types d'AGV :

- Les AGV remorqueurs (towing AGVs), qui peuvent tirer des charges sur roulettes, sont utilisés dans les installations où les charges sont volumineuses. Ils sont capables de tracter des charges de l'ordre de 25 tonnes. [Viswanadham, 92]
- Les AGV à charge unitaire (unit load AGVs), sont utilisés pour transporter des charges sur leurs plates-formes. Ils peuvent être équipés de plusieurs accessoires comme des rouleaux, des convoyeurs à tapis, des chariots élévateurs... etc. Des charges transportées de l'ordre de 6 tonnes sont fréquentes. [Viswanadham, 92]
- Les AGV d'assemblages (assembly line AGVs), sont utilisés dans les lignes d'assemblages, ils transportent les produits d'une station d'assemblage à l'autre, jusqu'à la livraison des produits finis à la dernière station
- Les AGV à bras manipulateurs (Robotic arm AGVs), sont des combinaisons d'AGV et de robots. Ils sont utilisés pour les faibles charges. On les rencontre, principalement, dans l'industrie des semi-conducteurs. Leur potentiel d'utilisation est nettement plus important que celui des autres types d'AGV (avantage des AGV et des robots).
- Les AGV pour système de fabrication flexibles (flexible machining system AGVs), sont des AGV qui servent à transporter et servir de supports aux produits. De plus, les outils nécessaires à la fabrication y sont entreposés.  
Dans ce cas, la matière première et les outils sont chargés sur l'AGV, ils sont transportés aux différentes stations de travail pour subir les différentes opérations tout en restant sur l'AGV. Finalement le produit fini est déchargé de l'AGV. Ces véhicules nécessitent une grande précision de positionnement. Ils permettent de réaliser des ateliers flexibles entièrement automatisés.
- Les AGV autonomes (autonomous AGVs), sont les AGV du futur. Ils pourront communiquer avec les autres AGV du système et avec le contrôleur principal. Ils seront dotés de systèmes visuels leur permettant d'éviter les obstacles et de



reconnaître les produits. Ils seront capables de détecter leurs propres dysfonctionnements, de se réparer, de détecter les accidents...

Les AGV sont guidés de différentes manières :

- Electromagnétique, par fil électrique enterré, dans ce cas l'AGV détectera les signaux radio fréquence (RF) transmis dans le fil et pourra suivre le chemin tracé par le fil. Ce système est très utilisé parce que très fiable. Son principal inconvénient est la difficulté d'ajouter ou de modifier les chemins.
- Optique, ici des bandes en couleurs sont peintes ou collées sur sols. Des capteurs optiques permettent de détecter les différences de couleurs et de suivre ces bandes. Facile à installer, ces systèmes présentent l'inconvénient d'être inefficaces dans les environnements sales ou poussiéreux.
- Infrarouge, dans ce cas les véhicules sont équipés de caméras qui détectent des sources de lumière infrarouge aux extrémités des allées. D'installation facile et peu coûteuse, insensibles à la lumière ambiante, ils présentent, néanmoins, l'inconvénient de la sensibilité aux obstacles visuels.

### 1.3.3. Les systèmes de stockage

#### a) Introduction

Les systèmes de stockages sont utilisés dans les systèmes de fabrications afin d'absorber les aléas et de réguler la fabrication. Les aires de stockage peuvent, en plus du stockage de produits, servir à d'autres activités, comme l'emballage, le contrôle de qualité. Ils peuvent aller du système complètement automatique au stockage manuel.

#### b) Composants d'une aire de stockage

Une aire de stockage est composée d'un bâtiment, d'un système de rangement (racks, étagères...) d'un mécanisme de transport, de chargement et déchargement, et d'un système de contrôle.

Le bâtiment est en général construit en structure légère supportée par les racks de rangement. Le système de rangement est en général une structure en étagères composée de profilés d'acier verticaux sur quoi se fixent des profilés d'acier horizontaux formant les supports pour les palettes de stockage.

Des convoyeurs gravitationnels peuvent être utilisés comme rack pour minimiser l'espace des allées. D'une extrémité du convoyeur, on insère les produits à stocker et on les ressort de l'autre extrémité. Ces systèmes respectent la loi premier entré premier sorti – FIFO (First in first out).

Les systèmes de portiques roulant, de carrousels...peuvent être utilisés pour faciliter le stockage/déstockage. Des mécanismes de transport automatiques, semi

automatiques ou manuels sont utilisés pour le transport, le chargement et le déchargement des produits : les transpalettes sont souvent utilisées à cet usage, mais dans ce cas, jusqu'à 50% de l'espace des aires de stockage sera réservé aux allées qui doivent être assez large pour permettre aux transpalettes de manœuvrer. [Askin, 93]

### c) Les systèmes automatiques de stockage/déstockage

Les systèmes de stockages les plus élaborés sont les systèmes automatiques de stockage/déstockage (automated storage retrieval systems AS/RS). Ces systèmes, de plus en plus utilisés, sont constitués de magasins de stockage (racks) et d'un convoyeur reliant les machines S/R et les points d'entrée/sortie de l'aire de stockage. [Keserla, 94]

Les magasins de stockages sont constitués de grandes étagères (racks) installées le long d'allée permettant l'accès aux emplacements de stockage.

La machine de stockage/déstockage peut être un portique en forme de rectangle creux, les cotés horizontaux glissent sur des rails fixées le long des allées (l'un au sol, l'autre au plafond). Tandis que sur les cotés verticaux est placé un plateau glissant à deux degrés de liberté (linéaires, vertical et horizontal perpendiculaire à l'allée) permettant de charger et décharger les palettes.

Les convoyeurs reliant les machines S/R sont en général des convoyeurs linéaires soit à tapis roulant soit gravitationnels. A leurs deux extrémités, sont fixés les points d'entrée et de sortie du système de stockage. [Askin, 93]

## 1.4. Les systèmes de contrôle

Le système de contrôle peut être composé d'un ordinateur central de commande, il donne des instructions et reçoit des états des situations de chaque équipement du FMS. Il peut garder en mémoire les gammes d'usinage des produits si les ordinateurs subalternes sont saturés.

C'est le système de contrôle qui décide quand et/ou comment les produits doivent se déplacer entre les différentes machines et être chargés ou déchargés. Un système de contrôle élaboré peut commander et ordonnancer en temps réel le FMS. Il peut, entre autre, décider de remplacer les outils après une certaine durée d'utilisation pour éviter les pertes de qualité et les risques de casse ; il peut vérifier la charge des batteries des AGV... [Askin, 93]

Le système de contrôle élaboré est composé de trois entités principales :

- Les calculateurs, sont des ordinateurs à base de microprocesseurs dotés de mémoires et de puissances de calculs plus ou moins grandes.



- Les logiciels, développés pour chaque type de commande, sont le cœur du système de contrôle. De leurs puissances dépend le degré de performances du système.

- Les bases de données, les systèmes de transmission d'information, les capteurs...représentent la troisième entité.

#### □ Architecture des systèmes de contrôle

Plusieurs structures de contrôle ont été développées pour la commande des FMS. La plus part d'entre elles, sont une variante ou une combinaison de trois structures principales :

1. La structure centralisée : Cette structure est caractérisée par un seul centre de contrôle pour l'élaboration de l'ensemble des décisions. Toutes les informations arrivent dans un contrôleur central qui, après traitement prend les décisions qui s'imposent et les transmet aux différentes composantes du FMS pour les exécuter (Fig. 1.3.).

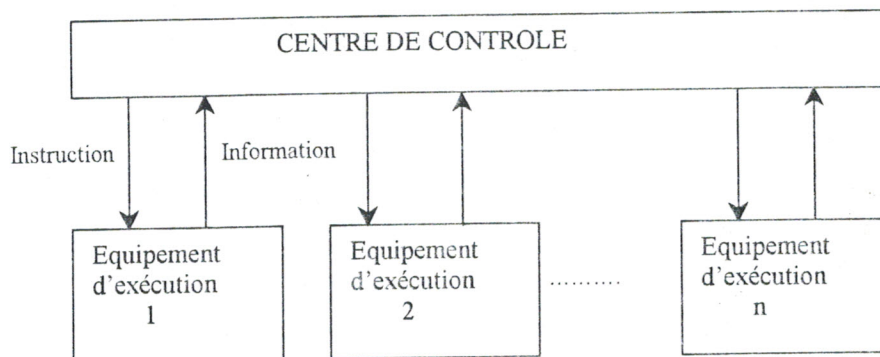


Fig. 1.3. Représentation d'une structure de contrôle centralisée.

Cette structure présente l'avantage de centraliser les décisions et de limiter le nombre de transmissions d'informations. En contre partie, elle présente plusieurs inconvénients dont :

- La limitation de la taille du FMS (imposée par les temps de traitement dans le centre de contrôle).
- La sensibilité aux perturbations (C'est l'ordinateur central qui doit prendre, par exemple, la décision de changer un outil après une casse).
- La difficulté d'extensions (ajout de nouveaux équipements). Le contrôle de chaque nouvel équipement doit se faire à partir de l'ordinateur central. Ce qui nécessite à chaque fois des réaménagements importants des logiciels et des systèmes de communications.

2. La structure hiérarchique : Dans cette structure, il existe plusieurs niveaux de contrôle, chacun recevant des instructions du niveau supérieur et transmettant des instructions au niveau inférieur. De même, les informations étant transmises en sens inverses (fig.1.4).

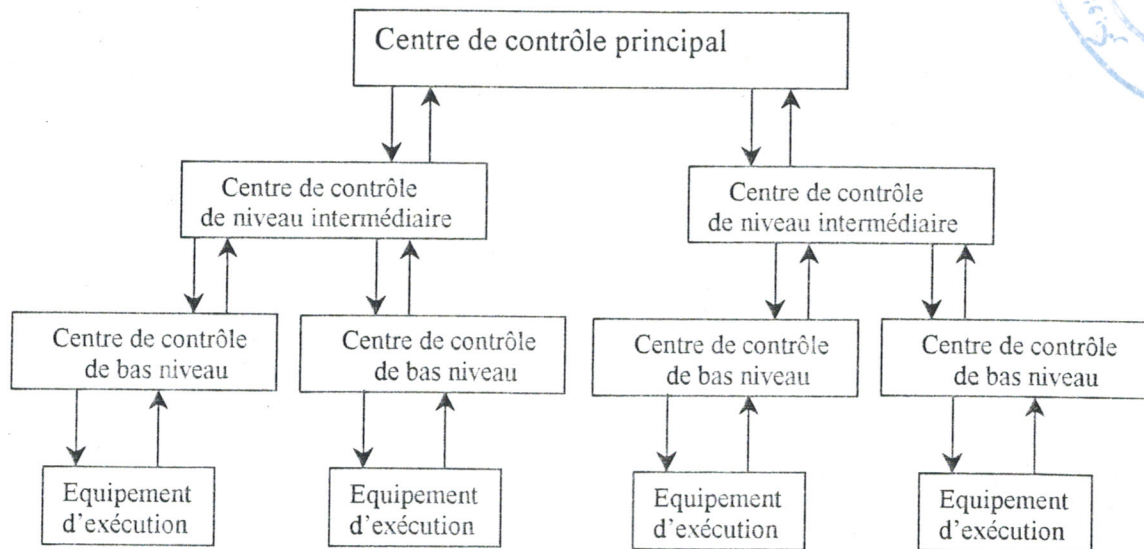


Fig. 1.4. Représentation d'une structure de contrôle hiérarchique à trois niveaux.

Cette structure présente plusieurs avantages dont :

- Le temps de décision est plus faible.
- Chaque niveau possède une certaine autonomie de calcul et de décision.

Par contre elle présente des inconvénients, dont le principal est le manque de coopération entre les centres de contrôle de même niveau.

### 3. La structure distribuée

Cette structure est caractérisée par la disposition de plusieurs centres de contrôle de même priorité. Chaque centre de contrôle, prend ses décisions indépendamment des autres, en fonction de son environnement propre, tout en ayant une coopération (échange d'informations) avec les autres centres de contrôle (fig. 1.5.a.).

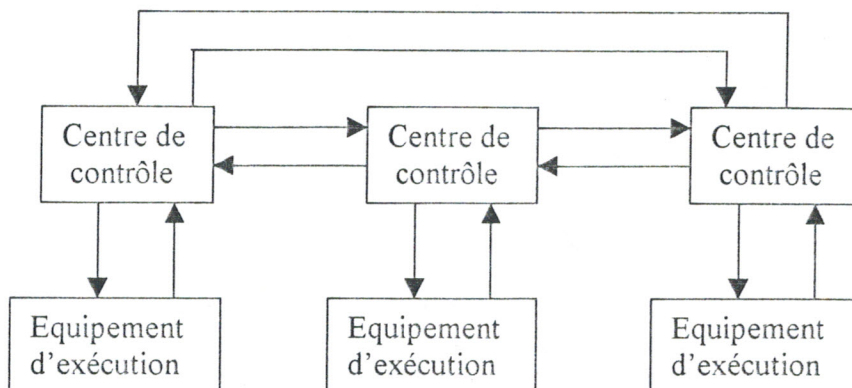


Fig. 1.5.a. Représentation d'une structure de contrôle distribuée.



Cette structure a pour principaux avantages :

- Chaque centre de contrôle peut prendre des décisions en temps réel concernant les équipements sous sa responsabilité.
- Le nombre de décisions par unité de temps est faible.
- La coopération avec les autres centres permet la prise de décision en tenant compte de l'environnement global.

Des structures hybrides, tentant de bénéficier des avantages de chaque structure existente. Par exemple, une structure hiérarchique, où les centres de décisions de chaque niveau pourront

communiquer entre eux tout en ayant plus de liberté dans la prise de décision concernant leur environnement propre, permettra de bénéficier des avantages des trois structures suscitées (fig. 1.5.b).

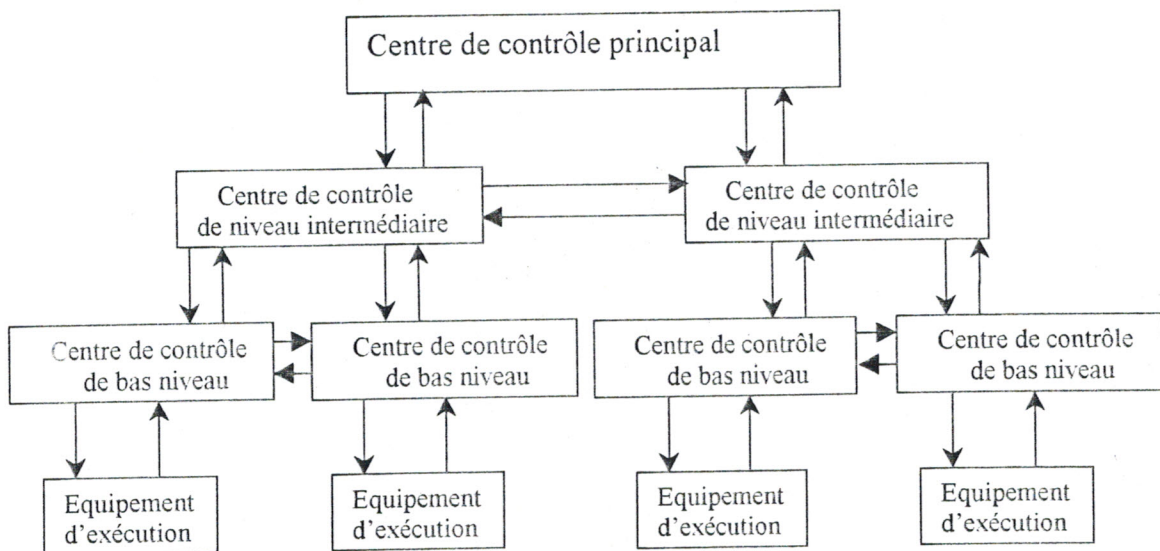


Fig. 1.5.b. Représentation d'une structure de contrôle hybride combinant une structure hiérarchique à trois niveaux et une structure distribuée.

### 1.5. Les systèmes de communications

Pour permettre la coopération entre les différents centres de contrôles, il est nécessaire de prévoir un système de communication entre ces différents centres. Le système de communication permet d'assurer les fonctions suivantes :

- Récupération de programmes ou données d'un centre de contrôle à un centre subalterne.
- Transmission d'informations concernant l'état de la production d'un contrôleur de machine à un centre de contrôle plus important.

- Acquisition des mesures de production (taux de production, temps d'attente, temps de changement, utilisation des machines... etc.).
- Transmission de fichiers à la mémoire de masse concernant l'état de toute les composantes du système (produits, outils, supports, palettes, matières premières... etc.).
- Transfert de données en temps réel, lorsque nécessaire (contrôle des outils, routage des AGV... etc.).

Pour ce faire, il est nécessaire de concevoir un réseau de communication local (Local area network LAN). On distingue deux types de réseaux [Bouteille et al, 97] :

#### 1- Les réseaux hétérogènes :

Les équipements interconnectés sont de nature différentes (automates programmables, commande numérique ou de robot, périphériques...) et nécessitent l'utilisation d'interfaces de communication externes pour assurer la compatibilité matérielle et logicielle des échanges.

#### 2- Les réseaux homogènes :

Les systèmes de contrôle reliés sont de même nature et compatibles entre eux. Ils se raccordent directement sur le médium commun (câble d'interconnexion et bornier passif) par l'intermédiaire de modules coupleurs conçus pour être intégrés directement dans les systèmes de contrôles.

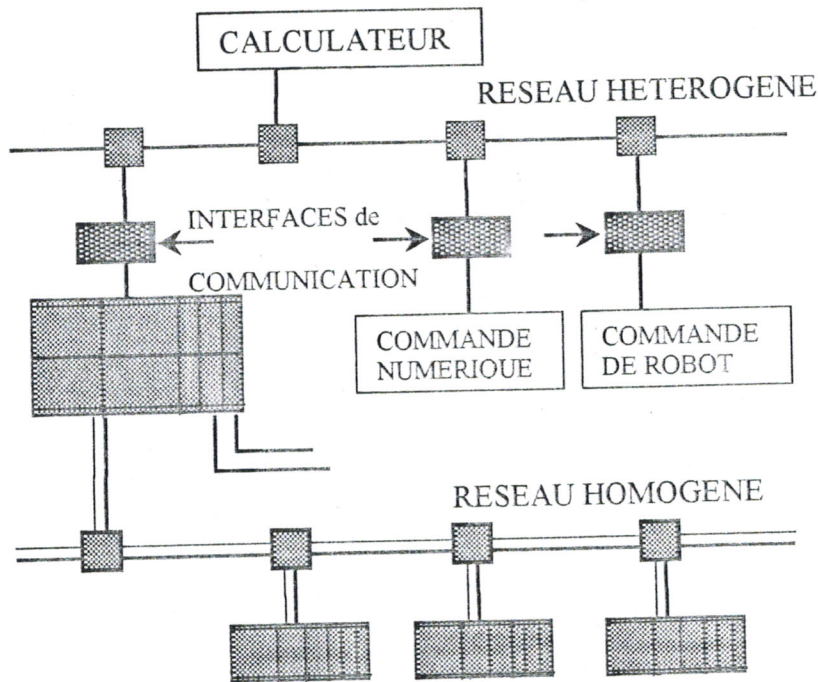


Fig. 1.6. Les deux types de réseaux de communication.

### - Topologies des réseaux de communications.

On trouve principalement trois types de configuration du réseau :

- 1- Topologie en étoile, où un système central est connecté directement à tous les équipements de contrôle.
- 2- Topologie en anneau, où les équipements sont connectés les uns à la suite des autres en boucle fermée.
- 3- Topologie en bus, où chaque équipement de contrôle vient se connecter sur un bus principal.

### 1.6. La flexibilité [Sari et al, 98]

Il existe plusieurs facteurs nécessitant la flexibilité. On peut citer entre autre :

- L'incertitude : qui comporte les incertitudes relatives aux équipements de production (pannes, défaillances,...), les incertitudes relatives aux personnels (erreur de jugement, absentéisme...) et l'incertitude de l'environnement extérieur (fournisseurs, clients, transport...).
- La variabilité de la demande : La demande peut être aléatoire, saisonnière, cyclique...



- Les produits et les technologies ont des durées de vie courtes.
- Le large éventail de produits manufacturés.
- La production à la commande.
- Les courts délais de livraison.

La flexibilité est donc la caractéristique de l'interface entre un système et son environnement extérieur, dans ce cas la flexibilité agit comme un filtre protégeant les systèmes des perturbations extérieurs. Elle peut être définie aussi comme la possibilité de s'adapter à un large éventail d'environnements différents.

### 1.7. Types de systèmes flexibles de productions

Plusieurs classifications de FMS sont données selon les auteurs. Nous citons à titre d'exemple une classification donnée par Askin [Askin, 93], basée principalement sur le nombre de machines à commande numérique (CN) :

- *L'équipement (equipment)*, représente une machine à CN, un robot manipulateur...
- *La station de travail (workstation)*, représente une machine à CN avec son aire de stockage, son système de chargement/déchargement, son changeur d'outils...
- *La cellule flexible (cell)*, représente une ou deux stations de travail.
- *L'atelier flexible (shop)*, représente un ensemble de cellules reliées par un système de transport automatique.
- *L'unité industrielle (facility)*, est l'ensemble de plusieurs ateliers.

Askin pense que dans la plus part des cas un système flexible de production représente, en fait, un atelier flexible.

### 1.8. Mesure de performances

Le but d'un système de production est de faire des profits, il faut fabriquer des produits à faible coût et de bonne qualité pour pouvoir les imposer dans un marché à forte compétition.

De ce fait, pour améliorer un produit, il faut minimiser son coût et augmenter sa qualité. Ceci peut se faire à travers l'optimisation d'un certain nombre de critères appelés critères de performances :

- Le temps de cycle : C'est le temps compris entre l'arrivée de la matière en production et la sortie du produit fini. C'est le temps moyen qu'un produit passe dans le système de



production. Il est égal à la somme des temps de chargement, des temps d'exécutions, des temps d'attente.

Seul le temps d'exécution augmente la valeur du produit, tandis que les autres temps, tout en coûtant de l'argent, ne donnent aucune valeur ajoutée au produit. Il faudra donc minimiser ces temps pour minimiser les coûts de production. Le temps de cycle doit être minimisé autant que possible.

- Les stocks d'en cours : Les stocks d'en cours coûtent de l'argent sans apporter de valeur ajoutée au produit. Ils doivent être minimisés. Néanmoins, il reste nécessaire de conserver des stocks pour palier les éventuelles perturbations externes (approvisionnement défectueux, commande imprévue...) ou internes (pannes).
- Le taux d'utilisation des machines : c'est le rapport entre la somme des temps d'exécutions effectifs et le temps de production.
- Le taux de production : Le taux de production représente la quantité de produits sortant du système par unité de temps, il doit être maximisé.
- La capacité de production : C'est le débit de production maximum pouvant être atteint en tenant compte des contraintes du système. C'est la valeur maximum que peut prendre le taux de production.
- La qualité : La qualité des produits est leur aptitude à répondre aux besoins de l'utilisateur (fiabilité, durée de vie,...), pour assurer cette qualité il faut limiter les rebuts au maximum.

*Chapitre 2*

**Cellule flexible automatisée de l'école des Mines  
de Nantes : une étude de cas**

## 2.1. Introduction

Avant d'aborder la description de la cellule flexible d'assemblage l'objet de notre étude, il nous a paru nécessaire de préciser que nous nous sommes inspiré en grande partie d'un modèle de cellule flexible acquis par l'école des mines de Nantes (France).

L'étude se fera donc sur la base des descriptions fonctionnelles, fournis par les documents de la plate forme. [Schneider, 97a ; Schneider, 97b ; Bulle, 97]

Cependant pour les graficets de commandes et de supervision, nous étions obligés de les concevoir parce qu'ils n'ont pas été fournis avec la documentation de la plate forme.

## 2.2. Présentation de l'installation physique

La cellule flexible de l'école des mines de Nantes (E.M.N.) est un atelier d'assemblage entièrement automatisé. C'est à dire que la fabrication des produits est exécutée par des robots de type bras manipulateur six axes, pilotés par des automates programmables industriels (A.P.I.). Le moyen de transport utilisé est un convoyeur constitué d'actionneurs de type vérins pneumatiques et de tapis chaînés à écailles.

Le stockage et déstockage sont assurés par un robot industriel quatre axes piloté lui aussi par automate.

### 2.2.1. Descriptif des produits fabriqués

La cellule flexible d'assemblage ayant avant tout un rôle pédagogique et de recherche, elle a été conçue dans l'esprit de réaliser des opérations simples sur des produits simples, un des intérêts de l'installation étant d'étudier les temps et quantités de production.

Ces produits sont donc des assemblages de pièces Duplo. Outre l'avantage quant à la simplification de la fabrication, ce type ne génère pas de déchets, et chaque pièce de la nomenclature est réutilisable à l'infini.

Chaque produit est composé au minimum d'une plaque de fond qui supporte l'assemblage des Duplos que l'on vient imbriquer dessus.

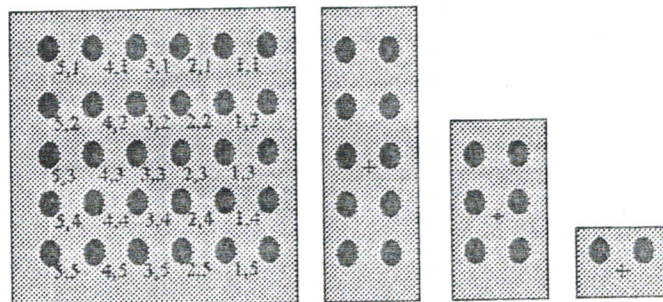


Fig. 2.1.a. Les différents types de Duplos utilisés.

Le repérage des pièces par rapport à la plaque est illustré sur la figure ci dessus (Fig. 2.1.a.). Les Duplos sont repérés par leur centre.



Chaque opération de la gamme étant définie par les coordonnées en x, y et z (de 1 à 5 niveaux de duplos, plus la plaque support) de la pièce devant être posée, son type (petite, moyenne ou longue), sa couleur (jaune ou verte), son orientation (horizontale ou verticale) et la station d'assemblage (SA1 ou SA2) sur laquelle l'opération doit avoir lieu.

Un produit est composé d'un maximum de 15 opérations. Une opération consistant à poser une pièce Duplo. Les paramètres d'une opération (couleur, orientation...) sont transformés par une opération logique en un code qui est compréhensible par les robots d'assemblage.

### 2.2.2. Palettes.

Les produits sont fabriqués sur des palettes transitant sur des tapis roulants et des aiguilleurs. Ces palettes au nombre de 30 contiennent chacune une étiquette inductive permettant de mémoriser les informations utiles à la fabrication (numéro de la palette, état, liste des codes opérations définissant la gamme du produit, temps d'assemblage...). La taille mémoire des étiquettes inductives est suffisante pour stocker d'autres informations si besoin.

Les palettes disposent des éléments mécaniques à leur transition d'une station à une autre et à leur indexation, ainsi que sur leur dessus de 4 pions de positionnement des plaques de fond des produits. La fig. 2.1.b. illustre le contenu d'une étiquette inductive (électronique).

Numéro de la palette	1 mot
N° de commande	1 mot
Etat de la palette	1 mot
Mot de routage – pour définir le prochain aiguillage de la palette -	1 mot
Nombre d'opérations à effectuer sur palette	1 mot
Nombre d'opérations déjà effectuées sur palette	1 mot
Niveau de priorité de la palette	1 mot
Opérations possibles n=1 à 15	X mots
Temps d'assemblage total	1 mot

Fig. 2.1.b. Contenu d'une étiquette inductive.



### 2.3. Vue d'ensemble de la cellule flexible

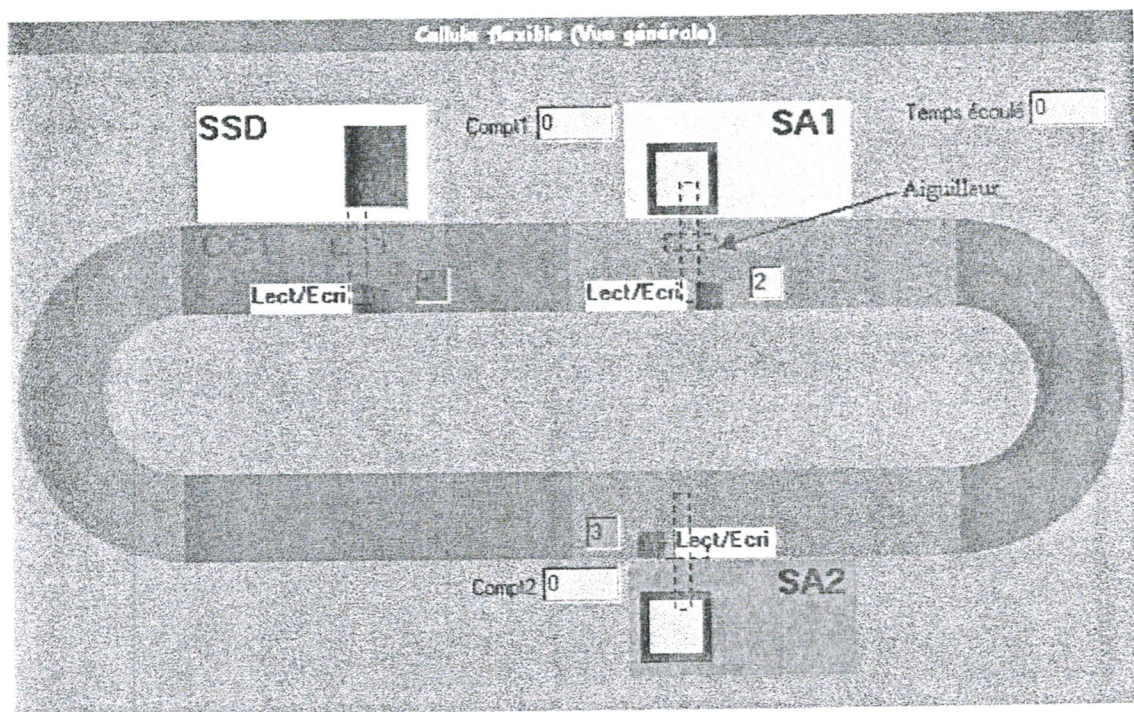


Fig. 2.2. Topologie de la cellule flexible d'assemblage.

La cellule flexible est composée de quatre parties distinctes (Fig. 2.2.) :

- deux stations d'assemblage (SA1 et SA2).
- une station de convoyage et d'aiguillage (CC1).
- une station de stockage - déstockage (SSD).

#### a) Stations d'assemblage

Les deux stations d'assemblage sont identiques, elles sont appelées SA1 et SA2, et peuvent exécuter le même type de travail.

Une station d'assemblage est composée d'un indexeur (celui ci permet de positionner les palettes de manière précise pour les mettre en référence par rapport au robot), d'un bras manipulateur et de sept goulottes de distribution des pièces d'assemblage. La description détaillée d'une station d'assemblage est donnée au paragraphe 2.4.2.b.

#### b) Station de convoyage et d'aiguillage

La station de convoyage et d'aiguillage appelée "CC1" est la partie intelligente de la cellule puisque c'est elle qui se charge d'aiguiller les palettes en fonction de leur numéro



et de leur état. Elle comporte trois aiguilleurs pour envoyer les palettes sur SA1, SA2 ou SSD.

C'est l'automate qui contrôle cette station et qui connaît les positions des palettes à chaque passage par les blocs de lecture - écriture. Les palettes sont lues et /ou écrites à chaque fois qu'elles passent devant l'une des trois antennes inductives de CC1 (blocs de lecture - écriture).

### c) Station de stockage - déstockage

Comme les stations SA1 et SA2, la station de stockage - déstockage SSD comporte un indexeur et un bras manipulateur. Son rôle est de stocker et de déstocker les produits finis ou défectueux. Les produits sont stockés dans 100 cases du magasin dont le robot est situé au centre.

## 2.4. Architecture de l'automatisme

L'automatisme est composé de quatre Automates programmables industriels (API) TSX-37 reliés par réseau Fipway (débit 1Mbit/s). Ils sont rigoureusement identiques, sauf l'automate CC1 qui dispose d'une carte mémoire et d'une carte entrées - sorties supplémentaires. Les automates sont reliés aux capteurs inductifs qui lisent et écrivent les étiquettes magnétiques des palettes par bus UnitelWay. (Voir fig. 2.4.).

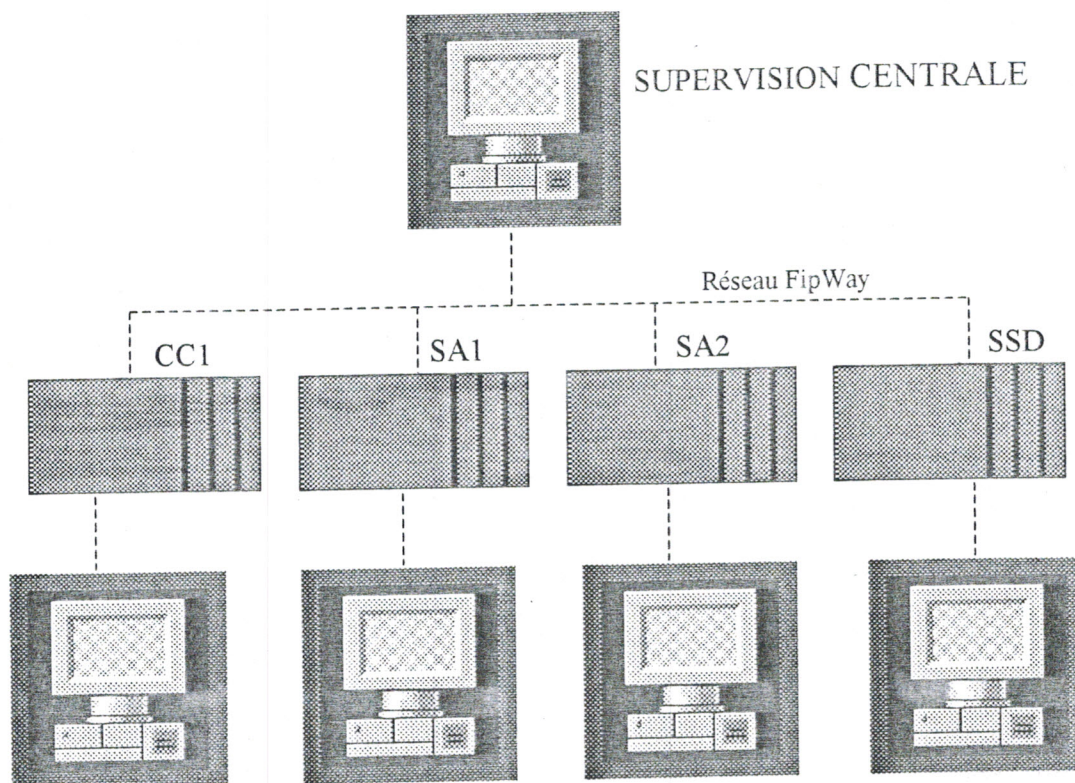


Fig. 2.4. Architecture de l'automatisme de la cellule flexible.



### 2.4.1. Analyse de la partie commande

#### a) Introduction

Rappelons brièvement le principe d'un automatisme, qui comporte deux parties :

- Une partie opérative (P.O.), dont les actionneurs agissent sur le processus automatisé.

- Une partie commande (P.C.) qui coordonne les actions de la partie opérative. (Voir fig. 2.5.).

La partie opérative, est celle qui opère sur la matière d'œuvre et le produit. Elle comporte en général : les outillages (outils de coupe, tête de soudure...) et les actionneurs destinés à mouvoir ces moyens (moteur électrique, vérin pneumatique...).

La partie commande, émet des ordres vers la partie opérative et en reçoit les signaux en retour des capteurs qui rendent compte de l'évolution du système, afin de coordonner ses actions.

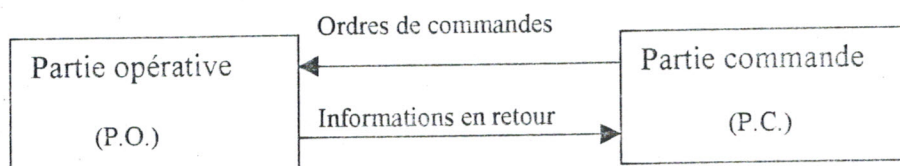


Fig. 2.5. Principe d'un automatisme.

#### b) Le procédé

La cellule flexible d'assemblage est un système automatisé complexe, il est composé de quatre stations SA1, SA2, CC1 et SSD.

Cette cellule permet d'effectuer des opérations d'assemblage sur des palettes. Le déplacement de ces palettes à l'intérieur de la cellule est assuré au moyen d'un convoyeur. Les palettes sont introduites ou retirées de la cellule par la station de stockage - déstockage (SSD). Des aiguilleurs relient le convoyeur aux différentes stations ; ceux-ci permettent de déplacer les palettes entre les différentes stations.

Un robot est associé à chacune des stations d'assemblage et de stockage-déstockage. Les robots permettent d'effectuer des opérations d'assemblage de duplos sur les palettes.

Chaque station comporte de nombreux actionneurs (permettent, par exemple, d'aiguiller les palettes - vérins d'aiguillage-) ainsi que de nombreux capteurs (pouvant, par exemple détecter la présence des palettes).

Les palettes sont équipées avec des étiquettes électroniques (cliquettes inductels). Chaque étiquette contient toutes les informations concernant la palette à laquelle elle est associée (type, routage, opérations déjà effectuées...). Le contenu des étiquettes peut être lu et/ou modifié à divers endroits de l'atelier par des blocs de lecture/écriture.

### c ) Description du problème

Des palettes de différents types (selon la gamme de production affectée), sont introduite dans la cellule (par ordre de priorité) par l'intermédiaire de la station de stockage - déstockage. Chaque palette doit être convoyée d'une station d'assemblage à une autre pour y subir des opérations.

Avant d'être introduite dans le système, chaque palette à un routage qui est préalablement enregistré sur son étiquette électronique.

On impose les spécifications de fonctionnement suivantes :

1) *Routage des palettes* : Une palette introduite dans la cellule, doit subir les opérations inscrites sur son étiquette électronique par les robots des stations d'assemblage correspondantes. Chaque opération a un temps d'exécution défini par la gamme de production. Ainsi, les aiguilleurs doivent être commandés de façon à garantir un routage correct des palettes.

2) *Opérations d'assemblage* : Une palette aiguillée vers une station d'assemblage doit subir des opérations. Il s'agira donc de commander la station d'assemblage de façon à ce que l'opération puisse être effectuée par le robot.

3) *Capacité d'une station d'assemblage* : De façon à garantir la bonne exécution d'une opération d'assemblage, au plus une palette peut être simultanément présente dans la station d'assemblage.

### d ) Analyse du problème

Dans notre problème de commande, nous avons essayé de tirer avantage de la modularité de notre système.

En fait, chaque station est commandée par un automate programmable industriel (API), et supervisée par un PC de supervision locale.

De plus, rappelons que l'étiquette électronique associée à chaque palette, comporte toutes les informations pertinentes concernant le routage de la palette. Ainsi, l'utilisation de ces étiquettes permet de nous affranchir de la nécessité d'un système de commande qui centralise les informations concernant chaque palette : son routage, sa position dans la cellule... etc. Ceci nous permet de séparer notre problème de commande en sous-problèmes indépendants.



Dans notre application, les stations d'assemblage ont un fonctionnement similaire, de même les aiguilleurs des stations d'assemblage sont identiques.

La station de stockage – déstockage SSD possède un fonctionnement différent. Avant de spécifier son fonctionnement, commençons par définir les temps suivants :

- Nous définissons un temps  $t_r$  intrinsèque à la station SSD ; c'est le temps nécessaire au robot pour prendre et transférer une palette.

- Nous définissons un temps  $t_{SSD}$  qui est le temps nécessaire pour effectuer les opérations de déchargement et chargement d'une palette.

Les spécifications de fonctionnement de SSD sont :

1- Au démarrage du système, les palettes sont introduites dans la cellule en fonction de leur niveau de priorité, ceci avec une fréquence de sortie commandable. De ce fait les palettes sont disposées dans la station SSD en une file d'attente.

2- Lorsqu'une palette termine ses opérations, elle rejoint la station de stockage - déstockage SSD.

3- Après son transfert pour être déchargée, une palette peut rejoindre la queue de la file d'attente (FIFO : First In First Out, premier arrivé premier sorti) de la station SSD, et dans ce cas elle devient en attente. Dans le cas contraire (si elle ne rejoint pas la file d'attente) elle devient vide.

4- L'aiguilleur associé à la station SSD, doit donc être commandé de façon à assurer le fonctionnement suscit.

#### 2.4.2. Synthèse de la commande

Etant donné que les spécifications de commandes et de supervision de la cellule flexible n'ont pas été fournies par le constructeur (Schneider pour E.M.N.), il nous a paru intéressant de concevoir une commande de la cellule basée sur l'approche de commande supervisée, développée par Charbonier [Charbonier, 96].

Pour ce faire, nous avons utilisé des modèles Grafset pour la commande et la supervision.

##### a) Concept de commande supervisée

Le concept de la commande supervisée introduit, hiérarchise les rôles dévolus à la commande et à la supervision. La fig. 2.6. illustre le principe de ce concept.

Un procédé couplé à son système de commande est représenté dans la fig. 2.6.a. Les entrées du système de commande sont les sorties du procédé, et vice versa.

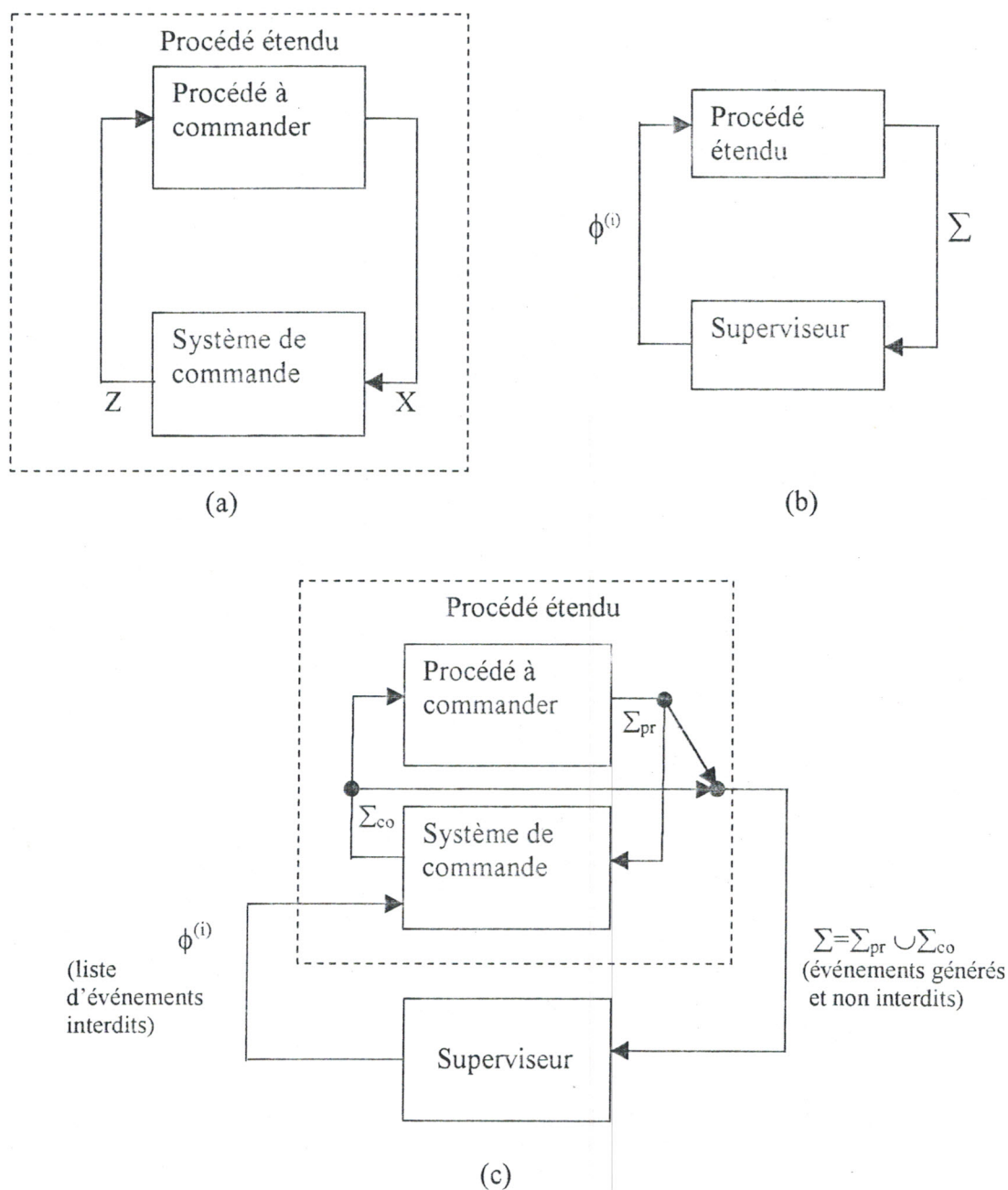


Fig. 2.6. Commande supervisée

- (a) procédé étendu
- (b) schéma de supervision
- (c) schéma de commande supervisée



Le procédé à commander est perçu comme un système à événements discrets "SED" (Voir chapitre 3).

Le système de commande est un SED qui observe l'état du procédé à commander par l'intermédiaire de ses entrées et modifie l'état du procédé par l'intermédiaire de ses sorties.

Les concepts de commande et de supervision sont clairement séparés. Il s'ensuit que la tâche de commande est indépendante de la tâche de supervision, l'avantage ainsi obtenu est que si l'on modifie les spécifications à imposer au niveau de la supervision, le système de commande ne nécessite pas d'être modifié.

### b) Application de la commande supervisée à la station d'assemblage

La fig. 2.7. donne le schéma d'une station d'assemblage.

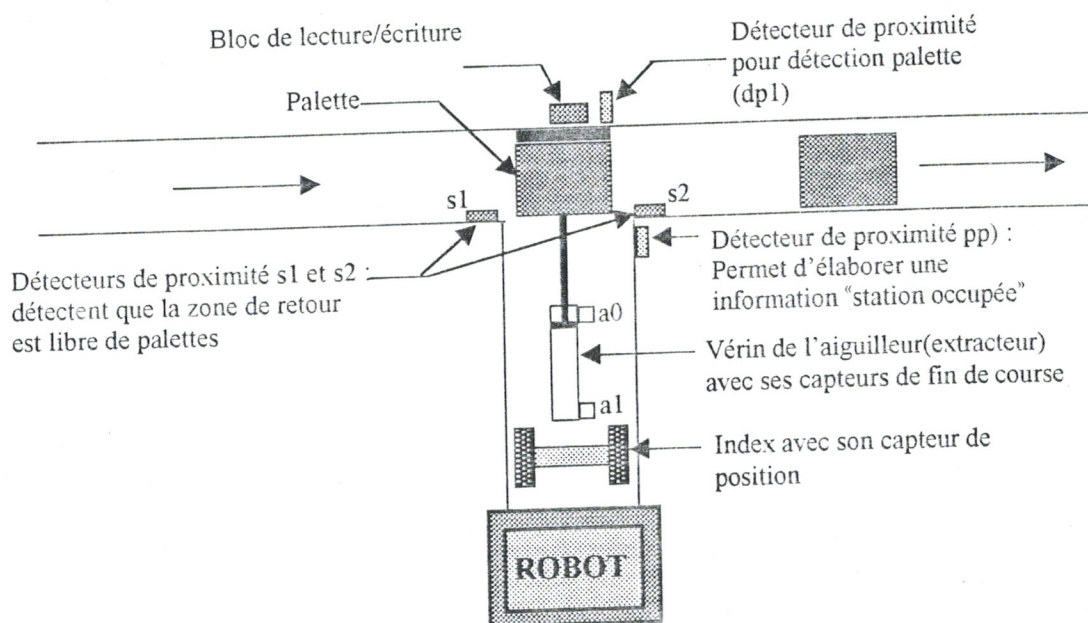


Fig. 2.7. Station d'assemblage.

#### 1) Spécifications de fonctionnement

Aussitôt qu'une palette arrive à un bloc de lecture – écriture, la lecture de l'étiquette est commencée. A la fin de lecture, si le code convient (aiguillage correspondant), l'extracteur déplace la palette vers l'index. L'index lève alors la palette, le robot effectue les opérations d'assemblage. Dès la fin de cette tâche, l'index baisse la palette et l'extracteur ramène la palette vers le convoyeur où une opération d'écriture sur l'étiquette

est faite. Dans ce fonctionnement, nous faisons l'hypothèse que l'extracteur est du côté du convoyeur et que l'index est en position basse à l'instant initial.

## 2) Spécifications de commande

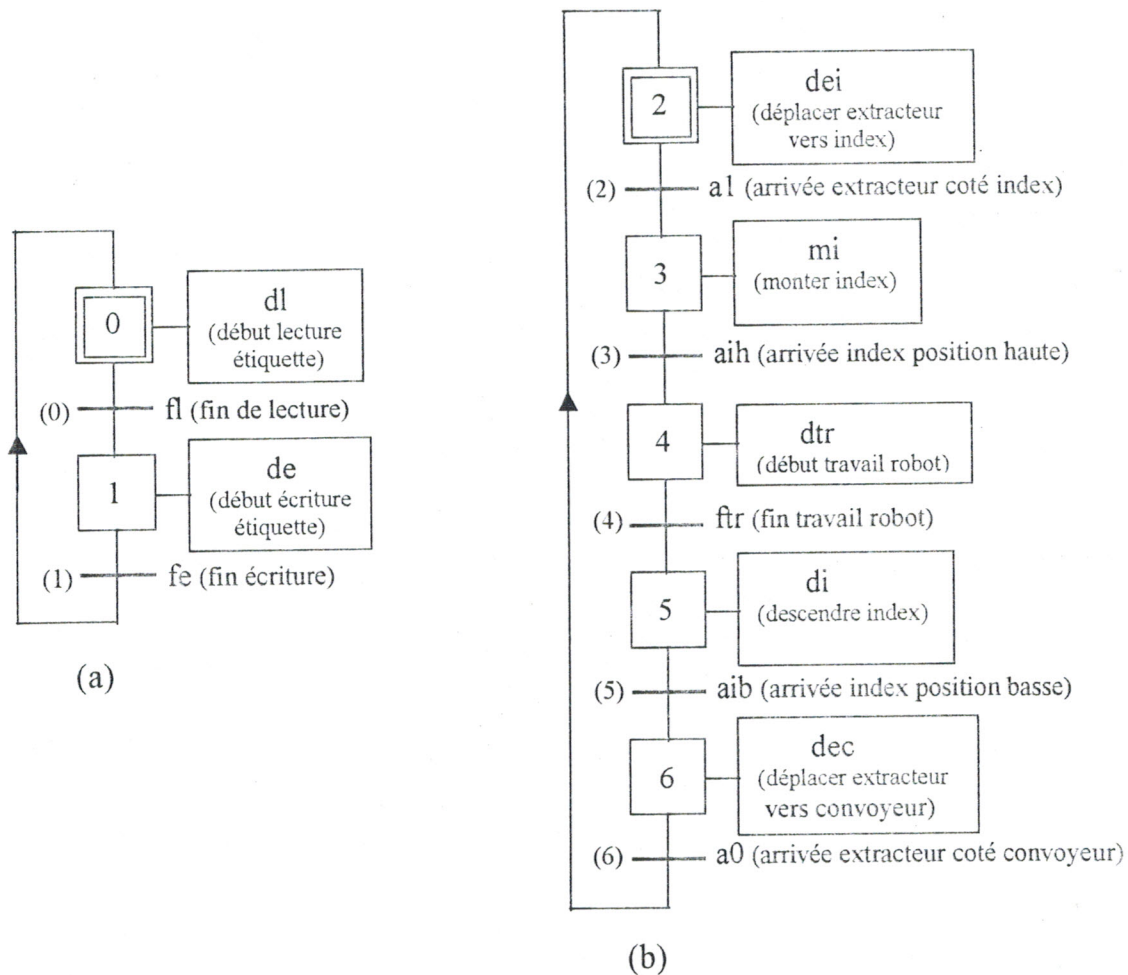


Fig. 2.8. (a) Commande du bloc de lecture/écriture.  
(b) Commande de l'extracteur, de l'index et du robot.

Le grafset de la fig. 2.8.a. permet de commander le bloc de lecture-écriture. A l'instant initial l'étape 0 est activée et un début de lecture sur l'étiquette est ordonné (événement "dl"). Aussitôt que la fin de lecture se produit (événement "fl"), l'étape 1 devient active et l'écriture est alors commencée (événement "de"). La fin de l'écriture ("fe") ramène le grafset dans son état initial (où une lecture recommence).

Le grafset de la fig. 2.8.b. permet de commander l'extracteur, l'index et le robot. A l'instant initial, l'extracteur est déplacé vers l'index ("dei"). A la fin de ce déplacement ("al"), l'index est monté ("mi"). Lorsque l'index arrive en position haute ("aih"), l'opération d'assemblage par le robot est alors amorcée ("dtr"). Puis à la fin de cette opération ("ftr"), l'index est descendu ("di"). Lorsque l'index arrive en position basse



("aib"), l'extracteur est déplacé vers le convoyeur ("dec"). Aussitôt que l'extracteur arrive du côté du convoyeur ("a0"), un nouveau cycle recommence.

### 3) Spécifications de supervision

Notons que les grafjets de commande de la fig. 2.8. ne permettent pas à eux seuls le respect des spécifications de fonctionnement. Par exemple, une opération de lecture ("dl" dans la fig. 2.8.a.) ne doit commencer que lorsqu'une palette est détectée ("dp1"). De plus, l'écriture ("de") ne doit débiter que lorsque l'extracteur revient du côté convoyeur ("a0").

Ces contraintes sont modélisées par le grafjet de supervision de la fig. 2.9.a où les débuts de lecture et d'écriture sont respectivement interdits lorsque les étapes 7 et 8 sont actives.

Le grafjet de la fig. 2.9.b. permet d'interdire le départ de l'extracteur vers l'index ("dei") tant que la lecture sur l'étiquette ("fl") n'est pas terminée ou la station est occupée ou encore le code ne convient pas, c'est à dire, tant que l'étape 9 est active. Le grafjet de la fig. 2.9.c. interdit le départ de l'extracteur vers le convoyeur ("dec") tant que la zone de retour n'est pas libre.

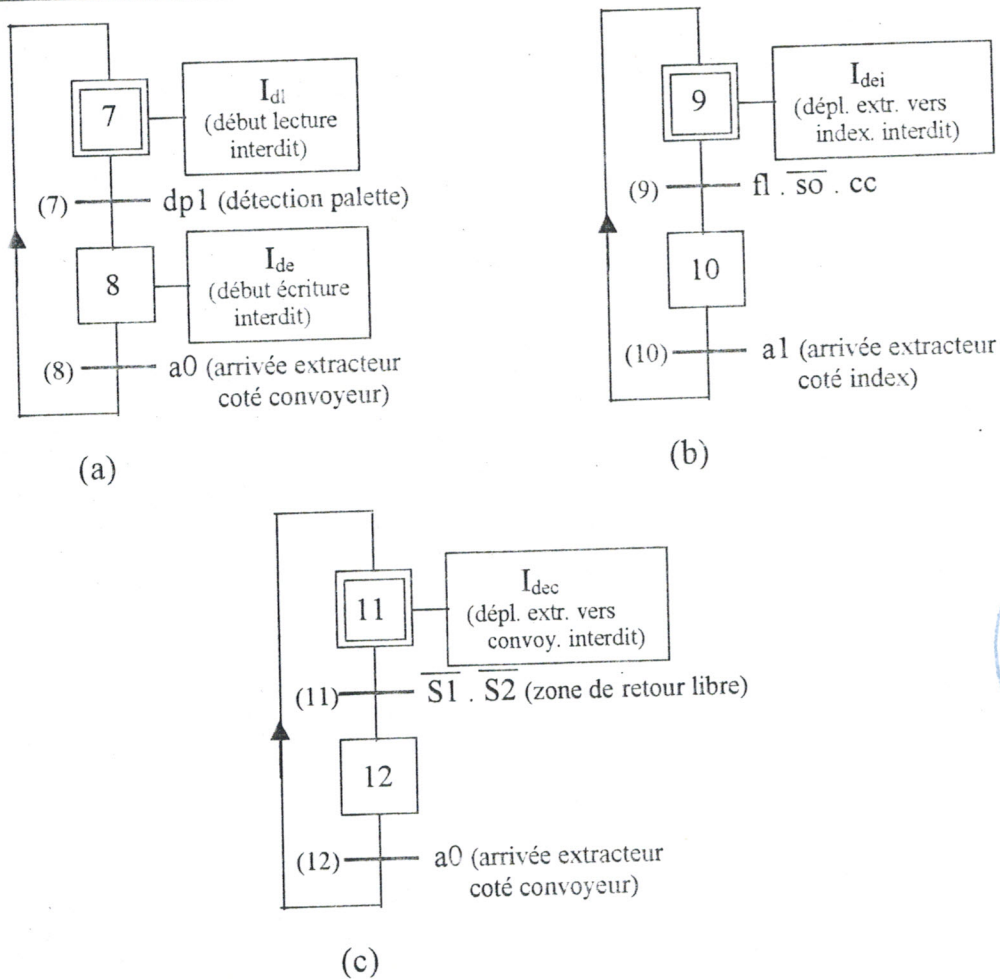


Fig. 2.9. (a) supervision du bloc de lecture/écriture.  
 (b) supervision du déplacement de l'extracteur vers l'index.  
 (c) supervision du retour de l'extracteur vers convoyeur.

Notations :  $so$  : station occupée  
 $cc$  : code convient.

Dans cette démarche les grafjets de la fig. 2.8. définissent les fonctionnements primaires du bloc de lecture- écriture, de l'extracteur, de l'index et du robot. Le rôle de la supervision consiste uniquement à rajouter les quelques contraintes nécessaires au respect du cahier des charges.

La commande supervisée de la station d'assemblage consiste à coupler les grafjets de supervision des fig. 2.9. avec les grafjets de commande de la fig. 2.8. Les grafjets de commande sous supervision sont représentés sur la fig. 2.10.



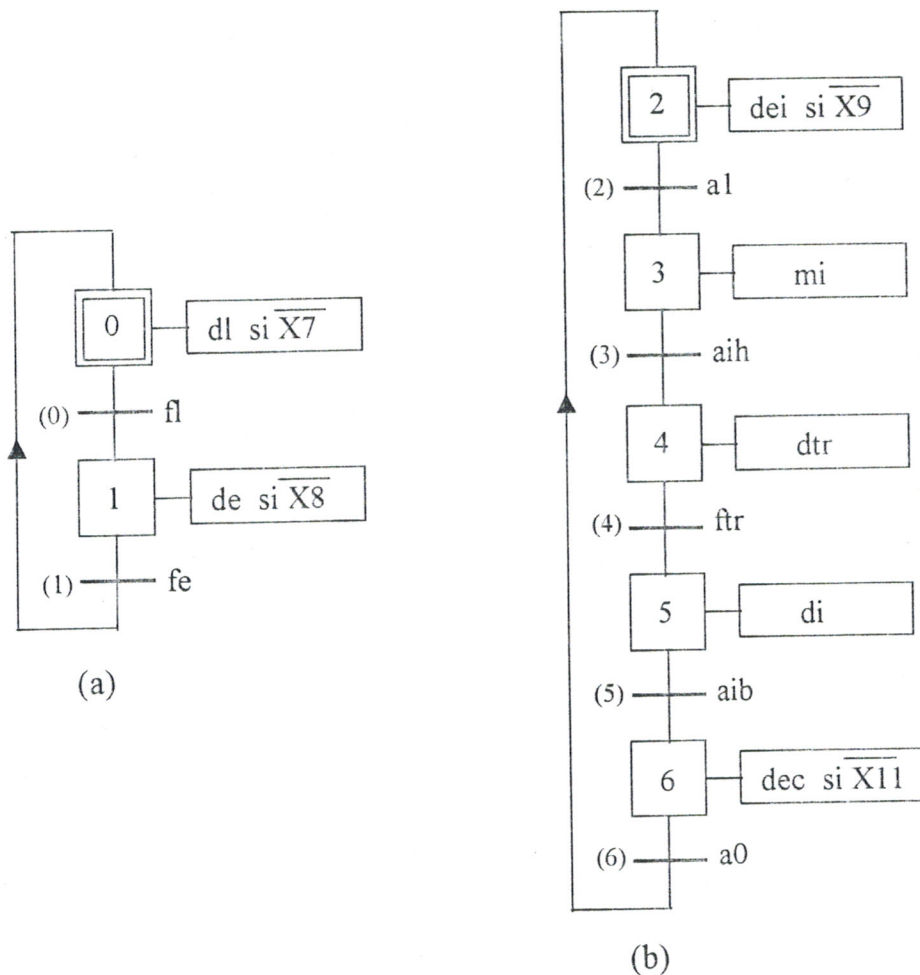


Fig. 2.10. Grafcets de commande sous supervision.  
 (a) bloc de lecture /écriture.  
 (b) extracteur, index et robot.

N.B. : On convient habituellement de désigner par "Xi" l'état actif de l'étape "i".  
 Exemples :  $\underline{X3} = 1$  si l'étape 3 est active  
 $\overline{X6} = 1$  si l'étape 6 est inactive.

Pour assurer le fonctionnement suscité de la station de stockage- déstockage SSD, nous avons envisagé une solution qui utilise deux vérins simple effet pour réaliser l'aiguillage à l'entrée de cette station (Fig. 2.11.).

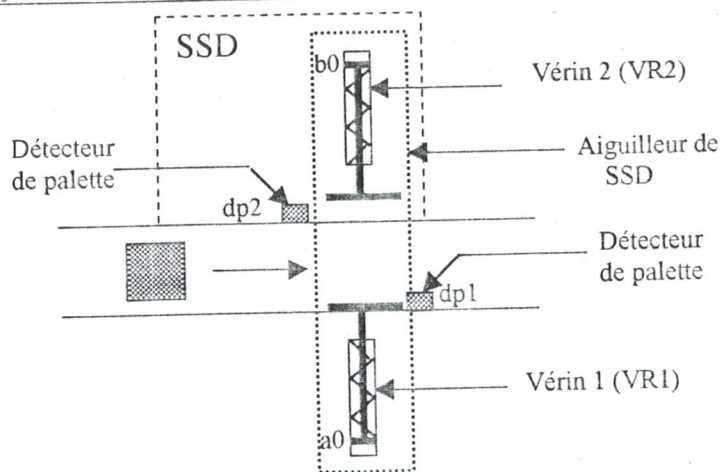


Fig. 2.11. Système d'aiguillage pour SSD.

Le vérin VR1 permet de dériver la palette vers SSD, le vérin VR2 permet d'évacuer la palette vers le convoyeur. La même démarche de commande sous supervision, permet d'obtenir les grafquets de la fig. 2.12.

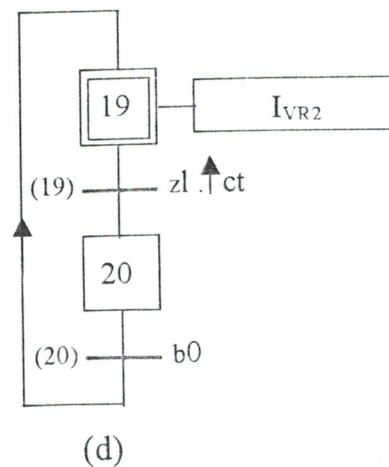
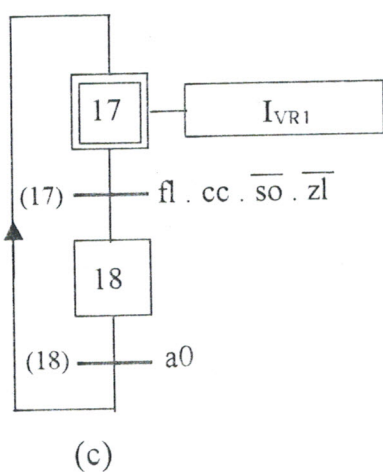
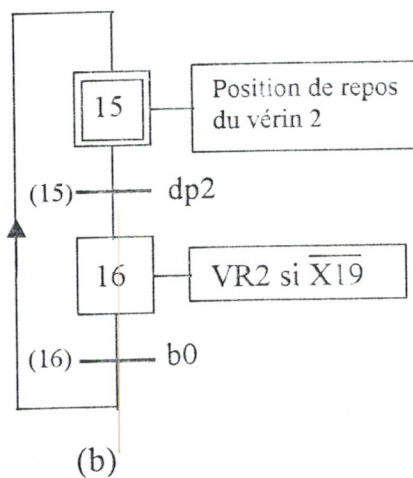
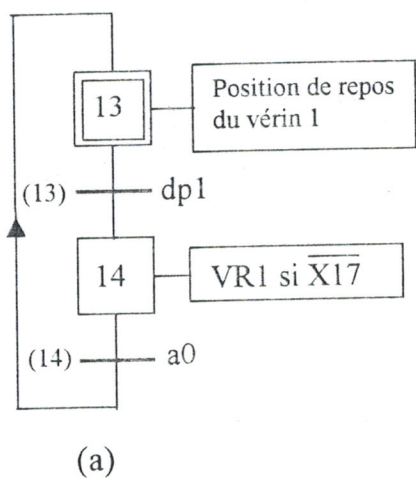


Fig. 2.12. (a) Grafquet de commande sous supervision de VR1.  
 (b) Grafquet de commande sous supervision de VR2.  
 (c) Grafquet de supervision de VR1.  
 (d) Grafquet de supervision de VR2.



Notations : z1 : zone de retour libre.

↑  
ct est un événement déclenché par les tops (fronts montants) d'une horloge de période programmable.

## 2.5. Conclusion

Dans l'approche de commande supervisée qui a été appliquée pour la conception des automatismes de la cellule flexible, le système de commande est composé de plusieurs modules. Chacun de ces modules permet de définir le fonctionnement de base d'une entité du procédé. Les contraintes de fonctionnement que l'on souhaite imposer entre les diverses entités du procédé sont du ressort de la supervision.

Dans notre exemple, la supervision est modulaire, notons que la modularité dans la commande et dans la supervision est rendue, en grande partie, possible par la hiérarchisation des concepts de commande et supervision.

*Chapitre 3*

**Modélisation et simulation des systèmes  
à événements discrets**



### 3.1. Systèmes à événements discrets

Un système à événements discrets (SED) est un système dans lequel l'espace des états est discret. Un tel système est à opposer à un système continu pour lesquels l'état est représenté par des grandeurs qui prennent des valeurs dans un domaine continu. De tels systèmes évoluent conformément à l'arrivée d'événements, caractéristiques de changement d'états du système. [Charbonier, 96]

Exemple : Considérons un ascenseur qui peut prendre 4 états : Bas, en monté, haut, et en descente. Une évolution possible de ce SED est représentée dans le chronogramme de la fig. 3.1.

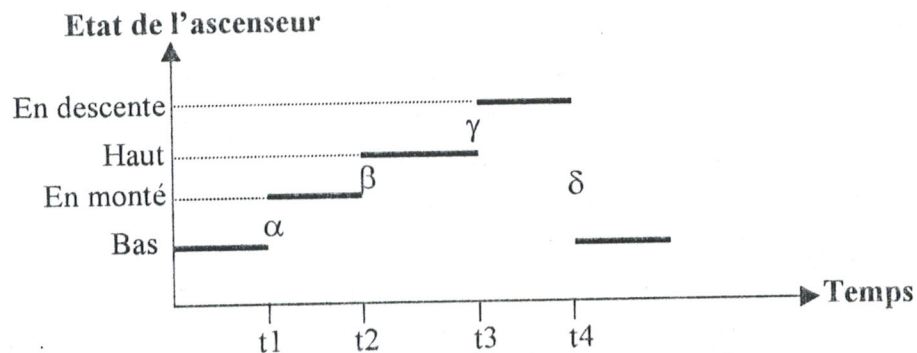


Fig. 3.1. Chronogramme de l'évolution de l'état d'un ascenseur.

Dans son état initial, l'ascenseur est supposé en position basse. A l'instant  $t_1$ , l'événement "début de montée" de l'ascenseur se produit. Cet événement est représenté par le symbole  $\alpha$  dans la fig. 3.1. De façon similaire, le système évolue aux instants  $t_2$ ,  $t_3$  et  $t_4$  sur occurrence des événements "fin montée", "début descente" et «fin descente» représenté respectivement par les symboles  $\beta$ ,  $\gamma$  et  $\delta$ .

Par abus de langage, nous pourrions parler indifféremment du symbole comme de l'événement physique qu'il représente. Ainsi, nous dirons que l'événement  $\alpha$  modélise la transition de l'état «bas» à l'état «en montée».

### 3.2. Quelques concepts fondamentaux

Les systèmes flexibles de production peuvent être considérés comme des systèmes à événements discrets "SED". Avant d'aborder les aspects de la modélisation et de la simulation des SED, nous présentons quelques concepts relatifs aux systèmes.

#### 3.2.1. Objet d'un système

Partant du concept qu'un système est un ensemble d'objets interdépendant unifié pour réaliser une fonction spécifique, l'objet est donc l'élément terminal de cet ensemble ; il est caractérisé par un ou plusieurs attributs auxquels des valeurs peuvent être affectées.

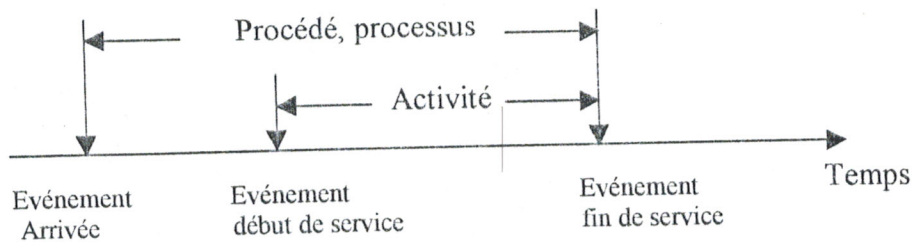


Fig. 3.2. relations entre événement, activité, processus.

### 3.3. Modélisation

Le processus de modélisation s'apparente à une phase de transformation d'un système dans un symbolisme de représentation qui capture les composantes principales qui caractérisent le comportement du système. Suivant l'objectif recherché, ces composantes pourront être différentes dans la mesure où elles sont représentatives de certains aspects du fonctionnement du système. Ce symbolisme de représentation s'appuie également sur des outils théoriques qui garantissent la structure des objets manipulés et les opérations que l'on peut leur appliquer. [Fdida et Pujolle, 89]

Les systèmes flexibles de production sont de plus en plus complexes. De ce fait, le dimensionnement correcte de ces systèmes devient une tâche difficile. Il faut en effet optimiser la plupart du temps, une dizaine de paramètres et le bon sens est devenu tout à fait insuffisant.

Une description fine d'un système permet une représentation fidèle du système, on parle donc d'un modèle réaliste. Or d'autre part, cela conduit à un modèle complexe souvent insoluble par des techniques exactes. Il apparaît donc nécessaire de choisir un compromis permettant d'atteindre l'objectif recherché avec un coût minimum.

#### 3.3.1. Différents types de modélisation

##### 1) La modélisation du type file d'attente

###### a) Introduction

Un objet de type file d'attente est apte à modéliser les phénomènes de consommation de ressource. Le terme ressource désigne ici toute composante physique, logique ou humaine qui peut être nécessaire à la réalisation d'une activité. Une activité est définie par le processus de cheminement d'une entité appelée "client" et qui demande, tout au long de son cycle de vie, l'accès à une succession de ressources indispensables pour sa réalisation (voir fig. 3.3.).



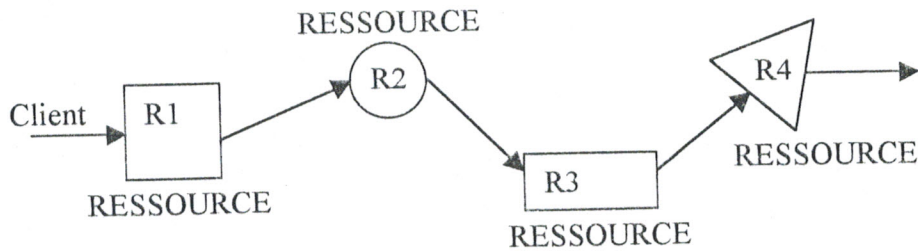


Fig. 3.3. Cheminement d'un client à travers une succession de ressources.

### b) Réseau de files d'attente dans le contexte des systèmes de production

Les systèmes de productions peuvent être représentés par un ensemble de ressources matérielles et logicielles accédées par des tâches ou des clients. Les ressources d'un système étant en nombre limité, les clients vont entrer en concurrence pour l'accès à ces ressources. Il apparaît alors naturel de représenter un tel système par un réseau de file d'attente. Un système de production est considéré comme un réseau de file d'attente où les machines et les opérateurs (ressources) représentent les serveurs et les produits sont les clients (fig. 3.4.).

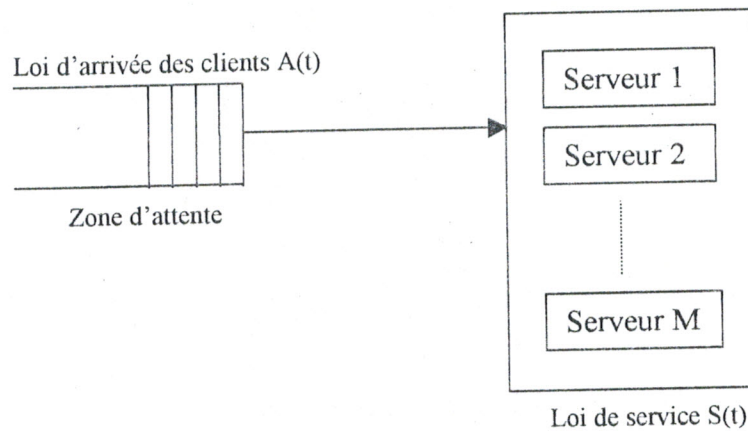


Fig. 3.4. Réseau de file d'attente.

Un client arrive suivant un certain processus  $A(t)$ . Si le guichet de service est libre, le client y entre et il est servi avec un processus  $S(t)$ . S'il n'est pas seul dans la file d'attente à son arrivé, il lui est appliqué une discipline de service  $DS$ . Les plus classiques sont FIFO (First In First Out) ou PARS (Premier Arrivé Premier Sorti) en français. Le nombre de guichets n'est pas obligatoirement unique, il peut y en avoir plusieurs, ce nombre est noté  $C$ . Le nombre de place dans la file d'attente est fini que nous noterons ici  $K$ . Si un client se présente lorsque la file est pleine, il est refusé. Enfin notons  $L$ , le nombre total de clients qui peuvent emprunter la file d'attente. Nous noterons une file d'attente par  $A/S/C/K/L (DS)$ .

Pour arriver à modéliser des systèmes beaucoup plus complexes, une file d'attente n'est pas suffisante, il faut faire appel à des réseaux de files d'attente. [Fdida et Pujolle, 89]

### c) Intérêts et limites.

Malgré l'importance des hypothèses simplificatrices apportées au modèle afin de lui donner les propriétés de résolution nécessaires, la précision des résultats est souvent bonne. Néanmoins, la classe des réseaux de file d'attente que l'on sait résoudre analytiquement est assez réduite car les solutions des équations mathématiques des modèles analytiques ne peuvent être obtenues qu'à condition de vérifier certaines hypothèses d'indépendance concernant les éléments du modèle ; les modèles les plus simples possèdent des solutions exactes sous des hypothèses strictes. [Fdida et Pujolle, 89]

## 2) La modélisation par réseau de Petri (RdP)

### a) définitions de base

Les Réseaux de Petri (RdP) sont des graphes bipartis qui comportent deux types de sommets, à savoir les places qui permettent de modéliser les stocks d'en cours, et les transitions qui représentent les opérations. Ils ont le double avantage de fournir un support graphique naturel qui est d'une aide précieuse pour le concepteur, et de posséder des propriétés analytiques qui permettent souvent une évaluation simple du comportement du système étudié. Dans ce qui suit, nous allons présenter les caractéristiques essentiels des RdP décrites dans la référence [Proth et Xie, 95].

Un RdP est composé de trois éléments de bases :

- Les places : qui peuvent être interprétés dans le contexte des systèmes de production comme des stocks tampons.
- Les transitions : sont interprétés comme des opérations.
- Les jetons : représentent des pièces.

Les places et les transitions sont reliées par des arcs. A chaque arc est associé un **poids** entier positif (la valeur par défaut est égale à 1). Le **marquage** d'un RdP est un vecteur à composantes entières positives ou nulles, de dimension égale au nombre de places. Chaque composante représente le nombre de jetons contenus dans la place correspondante.

De manière formelle, un RdP est un 5-uple  $PN = (P, T, A, W, M_0)$  où :

$P = \{p_1, p_2, \dots, p_n\}$  est un ensemble fini de places,

$T = \{t_1, t_2, \dots, t_q\}$  est un ensemble fini de transitions,

$A \subseteq (P \times T) \cup (T \times P)$  est l'ensemble fini des arcs,

$W : A \rightarrow \{1, 2, \dots\}$  est la fonction poids attachée aux arcs,

$M_0 : P \rightarrow \{0, 1, 2, \dots\}$  est le marquage initial.



Notons que  $P \cap T = \emptyset$ .

Le RdP sans son marquage initial est noté  $N = (P, T, A, W)$ , d'où  $PN = (N, M_0)$ . Le RdP est dit **ordinaire** si tous les poids des arcs sont égaux à 1. Nous notons :

- ${}^0t$  l'ensemble des places d'entrée de la transition  $t$ , c'est-à-dire l'ensemble des places  $p / (p, t) \in A$ .
- $t^0$  l'ensemble des places de sortie de la transition  $t$ , c'est-à-dire l'ensemble des places  $p / (t, p) \in A$ .
- ${}^0p$  l'ensemble des transitions d'entrée de la place  $p$ , c'est-à-dire l'ensemble des transitions  $t / (t, p) \in A$ .
- $p^0$  l'ensemble des transitions de sortie de la place  $p$ , c'est-à-dire l'ensemble des transitions  $t / (p, t) \in A$ .

Une transition  $t$  est dite **tirable** (ou **franchissable**) si :  $\forall p \in {}^0t : M(p) \geq W(p,t)$ .  $M(p)$  est le nombre de jetons contenus dans la place  $p$  et  $W(p,t)$  est le poids attaché à l'arc qui relie  $p$  à  $t$ . Tirer une transition  $t$  consiste à :

- retirer  $W(p,t)$  jetons de toute place  $p \in {}^0t$ ,
- ajouter  $W(t,p)$  jetons dans toute place  $p \in t^0$ .

Le tirage d'une transition  $t$  consiste donc à transformer le marquage initial  $M_0$  du RdP  $PN$  en un marquage  $M$  défini par :

$$M(p) = \begin{cases} M_0(p) - W(p,t) & \text{si } p \in {}^0t \\ M_0(p) + W(p,t) & \text{si } p \in t^0 \\ M_0(p) & \text{si non} \end{cases}$$

L'ensemble des marquages qu'il est possible d'atteindre à partir de  $M_0$  en effectuant un ou plusieurs tirages est noté  $R(M_0)$ . Une transition sans place d'entrée est appelée **transition source**. Une transition sans place de sortie est appelée **transition puits**.

Un RdP est dit **pur** s'il ne contient pas de boucle élémentaire, c'est à dire qu'une place ne peut pas être à la fois une entrée et une sortie de la même transition. Un **RdP à capacité finie** est un RdP dans lequel le marquage de chaque place  $p$  est limité à une quantité  $Q(p)$ . Dans un RdP à capacité finie, une transition  $t$  est tirable si et seulement si, en plus des conditions habituelles, les marquages des places de sortie de  $t$  n'excèdent pas leur capacité après tirage.

Dans un RdP ordinaire (c'est à dire où chaque arc est pondéré à 1), un ensemble  $P_s$  de places est un **verrou** (ou **impasse**) si toute transition qui a une place de sortie dans  $P_s$  a au moins une place d'entrée dans  $P_s$ . Un ensemble  $P_t$  de places est appelé **trappe** ou **piège** si chaque transition qui a une place d'entrée dans  $P_t$  a au moins une place de sortie dans  $P_t$ .

## b) Arbre des marquages atteignables, arbre de recouvrement et graphe de recouvrement

Considérons un RdP  $PN = (N, M_0)$ . L'objet de l'arbre des marquages atteignables est de découvrir tous les marquages que l'on peut atteindre à partir de  $M_0$ . Il s'agit d'arborescences dont les nœuds sont les marquages atteignables à partir de  $M_0$ , et dont chaque arc représente le tirage d'une transition. La racine de l'arborescence représente  $M_0$ .

Exemple : Considérons l'exemple de la fig. 3.5 dans lequel tous les arcs sont pondérés à 1.

Le marquage initial de ce réseau est  $M_0 = [2, 1, 0]$ . Partant de ce marquage, les transitions  $t_1$  et  $t_3$  peuvent être tirées. Ces tirages conduisent respectivement aux marquages  $M_{1,1} = [1, 2, 1]$  et  $M_{1,2} = [3, 0, 0]$  : C'est le premier niveau de l'arbre des

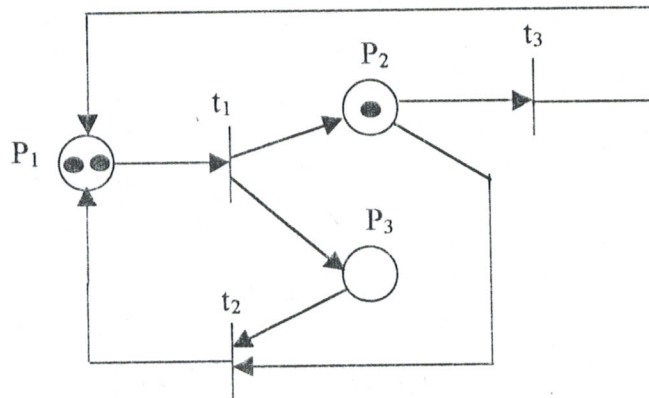


Fig. 3.5. Réseau de Petri général.

marquages atteignables. Partant de  $M_{1,1}$ , il est possible de franchir  $t_1$ ,  $t_2$  ou  $t_3$  qui conduisent respectivement aux marquages  $M_{2,1} = [0, 3, 2]$ ,  $M_{2,2} = [2, 1, 0]$  et  $M_{2,3} = [2, 1, 1]$ . Partant de  $M_{1,2}$ , on ne peut franchir que  $t_1$ , ce qui conduit au marquage  $M_{2,4} = [2, 1, 1]$ , qui est en fait le marquage  $M_{2,3}$ . Nous venons de construire le second niveau de l'arbre des marquages atteignables. Nous procédons de la même manière pour continuer la construction de l'arborescence. Comme nous pouvons le constater, chaque marquage conduit à autant de marquages qu'il y a de transitions tirables. De plus, le même marquage peut se retrouver à différents endroits de l'arborescence. Par conséquent, un arbre des marquages atteignables peut, dans la plupart des cas, se développer indéfiniment.

Pour éviter d'aboutir à un arbre qui se développe indéfiniment, on construit plutôt un **arbre de recouvrement** qui est dérivé de l'arbre des marquages atteignables, moyennant certaines règles de construction, il contient moins d'informations que le précédent mais reste de taille limitée. L'algorithme de construction de l'arbre de recouvrement est donné dans la référence [Proth et Xie, 95]. En construisant cet arbre, il est possible de déterminer si un RdP est **borné** (i.e. l'ensemble  $R(M_0)$  des marquages atteignables à partir de  $M_0$  est fini) ou non. Lorsque le système est



borné, l'arbre de recouvrement donne l'ensemble des états qui peuvent être atteints les uns à partir des autres, et la manière de réaliser ces transformations. A la notion d'arbre de recouvrement est associée la notion de **graphe de recouvrement** qui est un graphe dont tous les nœuds représentant le même marquage sont réduits en un seul nœud.

### c) Matrice d'incidence et équation d'état

La matrice d'incidence  $U = [u_{ij}]$ ,  $i = 1, 2, \dots, n$  ;  $j = 1, 2, \dots, q$ , d'un RdP pur est définie de la manière suivante :

$$u_{ij} = \begin{cases} W(t_j, p_i) & \text{si } t_j \in {}^0 p_i \\ -W(p_i, t_j) & \text{si } t_j \in p_i^0 \\ 0 & \text{si non} \end{cases}$$

$n$  est le nombre de places et  $q$  le nombre de transitions du RdP considéré.  $W(x, y)$  représente le poids de l'arc  $(x, y)$ .

Soit  $M_0$  un marquage initial et  $\sigma$  une séquence de transitions tirées à partir de  $M_0$ . Soit  $M$  le marquage obtenu après avoir tiré la dernière transition de  $\sigma$ . Nous définissons le vecteur de comptage de  $\sigma$  par  $V_\sigma = [v_1, v_2, \dots, v_q]$ , où  $v_i$  est le nombre de fois que la transition  $t_i$  figure dans  $\sigma$ . L'équation d'état s'écrit alors :

$$M^t = (M_0)^t + U \cdot (V_\sigma)^t \quad \text{où } X^t \text{ est la transposée de } X.$$

### d) P- invariant

Par définition, un vecteur  $Z$  de dimension  $n$  ( $n$  est le nombre de places du RdP considéré) et dont les composantes sont entières et non négatives est un p-invariant si :

$$Z \cdot U = 0 \quad (\text{où } U \text{ est la matrice d'incidence du RdP}).$$

#### - Théorème 1

Si  $Z$  est un p-invariant et  $M_0$  est un marquage initial quelconque d'un RdP, alors  $Z \cdot (M_0)^t = Z \cdot M^t$  pour tout  $M \in R(M_0)$  (i.e. pour tout marquage  $M$  atteignable à partir de  $M_0$ ).

#### - Définitions

1. L'ensemble des places qui correspondent aux composantes non nulles d'un p-invariant est appelé **support du p-invariant** considéré. Si  $Z$  est le p-invariant, son support est habituellement noté  $\|Z\|$ .

2. Le support d'un p-invariant est dit **minimal** s'il ne contient aucun ensemble non vide qui soit un support de p-invariant.

3. Nous appellerons **p-invariant minimal** un p-invariant  $Z$  tel qu'il n'existe aucun autre p-invariant ayant toutes ses composantes inférieures ou égales aux composantes correspondantes de  $Z$ .

### - Théorème 2

Tout p-invariant est une combinaison linéaire de p-invariants minimaux.

### e) T- invariant

Par définition, un vecteur  $W$  de dimension  $q$  ( $q$  est le nombre de transitions du RdP considéré) et dont les composantes sont entières et non négatives est un t-invariant si :

$$U \cdot W^t = 0 \text{ où } U \text{ est la matrice d'incidence du RdP.}$$

### - Théorème 3

Soit  $\sigma$  une séquence tirable et  $V_\sigma$  le vecteur de comptage correspondant. Si  $V_\sigma$  est un t-invariant,  $M_0$  le marquage initial et  $M \in R(M_0)$  le marquage obtenu après le tirage de  $\sigma$ , alors  $M = M_0$ .

### - Définitions

On définit le support du t-invariant, support minimal et le t-invariant minimal de la même façon que ceux concernant le p-invariant (dans ce cas le vecteur  $Z$  est remplacé par le vecteur  $W$ ).

### - Théorème 4

Tout t-invariant est une combinaison linéaire de t-invariant minimaux.

### f) Propriétés qualitatives des réseaux de Petri (RdP)

Nous distinguons les propriétés comportementales et les propriétés structurelles. Les premières dépendent à la fois de la structure du RdP et du marquage initial, les secondes dépendent uniquement de la structure du RdP. En termes de systèmes de production, nous pouvons dire que les propriétés comportementales dépendent de la structure du système et de son état initial, alors que les propriétés structurelles ne dépendent que de la structure du système.



## 1- Propriétés comportementales

### 1.1. Atteignabilité

La dynamique d'un système de production modélisé à l'aide d'un RdP est représentée par l'évolution des marquages obtenus, à partir du marquage initial  $M_0$ , par le franchissement de transitions.  $R(M_0)$  représente l'ensemble des marquages qu'il est possible d'atteindre (nous dirons "atteignables") à partir de  $M_0$ .

#### - Définition

Etant donné un RdP  $PN = (P, T, A, W, M_0)$  et un marquage  $M$ , le problème d'atteignabilité consiste à vérifier s'il existe une séquence de transitions  $\sigma$  qui transforme  $M_0$  en  $M$  (i.e.  $M_0 \xrightarrow{\sigma} M$ ) ou plus simplement si  $M \in R(M_0)$ .

Nous allons présenter quelques résultats théoriques qui permettent de construire des algorithmes pour vérifier si un marquage est atteignable.

#### -Théorème 5

- a) Si un RdP  $PN = (P, T, A, W, M_0)$  est borné, un marquage  $M$  est atteignable si et seulement si le graphe des marquages atteignables comporte un nœud représentant  $M$  ;
- b) Si le RdP  $PN = (P, T, A, W, M_0)$  est non borné, il est impossible de vérifier à l'aide d'un graphe de recouvrement si  $M$  est atteignable. On peut seulement vérifier qu'il existe un marquage  $M' \geq M$ .

#### -Théorème 6

Pour tout marquage atteignable  $M$  d'un RdP  $PN = (P, T, A, W, M_0)$  l'équation suivante possède au moins une solution :

$$M^t = (M_0)^t + U \cdot Y^t$$

où  $U$  est la matrice d'incidence de  $PN$  et  $Y = (Y_1, Y_2, \dots, Y_{|T|})$  est un vecteur à composantes entières non négatives. Une solution est le vecteur de comptage de la séquence  $\sigma$  des transitions dont les franchissements mènent de  $M_0$  à  $M$ .

### 1.2. Bornitude

Certaines places d'un RdP, modèle d'un système de production, représentent des zones de stockages. D'autres places contiennent des jetons qui représentent des ressources de fabrication. Il est souvent souhaitable de savoir si le nombre de jetons dans ces places est limité : cela permet de dimensionner le système de production correspondant ou de découvrir certaines erreurs de conception.

### Définitions

- a) Une place  $p$  d'un RdP  $PN = (P, T, A, W, M_0)$  est dite  $k$ -bornée si le nombre de jetons dans cette place ne dépasse jamais  $k$ , i.e.  $M(p) \leq k$ ,  $\forall M \in R(M_0)$ .
- b) Une place  $p$  est dite bornée si elle est  $k$ -bornée pour un certain nombre entier  $k > 0$ .
- c) Un RdP  $PN = (P, T, A, W, M_0)$  est dit  $k$ -borné si le nombre de jetons dans chaque place ne dépasse pas  $k$ . C'est à dire que :  
 $M(p) \leq k, \forall p \in P \text{ et } \forall M \in R(M_0)$ .
- d) Un RdP est dit borné s'il est  $k$ -borné pour un certain nombre entier  $k > 0$ .
- e) Un RdP  $PN = (P, T, A, W, M_0)$  est dit **sain** s'il est 1-borné, i.e.  
 $M(p) \leq 1, \forall p \in P \text{ et } \forall M \in R(M_0)$ .

### Théorème 7

Un RdP  $PN = (P, T, A, W, M_0)$  est borné s'il existe un nombre fini  $k$  tel que pour tout vecteur  $Y = (Y_1, Y_2, \dots, Y_{|T|}) \geq 0$ ,  $M(p) \leq k, \forall p \in P$  avec :  
 $M^t = (M_0)^t + U \cdot Y^t$ , où  $U$  est la matrice d'incidence.

N.B. Ce théorème donne une condition suffisante, mais non nécessaire.

### 1.3. Vivacité et blocage

Dans un système de production, il est fréquent que des activités manufacturières se déroulent en parallèle. Cela exige la synchronisation de ces activités et le partage des ressources. Une mauvaise synchronisation et un partage inadapté peuvent conduire à un blocage total ou partiel des activités. Un système de production bien conçu doit pouvoir éviter ces problèmes. L'étude de la vivacité d'un modèle de RdP garantit l'absence de blocage dans le système de production considéré.

#### - Définitions

- 1- Une transition  $t$  d'un RdP  $PN = (P, T, A, W, M_0)$  est dite **vivante** si elle peut être franchie quel que soit le marquage atteint, i.e. :  
 $\forall M \in R(M_0), \exists M' \in R(M)$  tel que  $t$  soit franchissable pour  $M'$ .
- 2- Un RdP  $PN = (P, T, A, W, M_0)$  est dit **vivant** si chacune de ses transitions est vivante.
- 3- Un marquage atteignable  $M$  d'un RdP  $PN = (P, T, A, W, M_0)$  est un **blocage** si aucune transition n'est franchissable à partir de  $M$ .
- 4- Un RdP  $PN = (P, T, A, W, M_0)$  est dit **sans blocage** si aucun de ses marquages atteignables  $R(M_0)$  n'est un blocage.



Il existe un certain nombre de théorèmes qui permettent de vérifier la vivacité et l'existence de blocages d'un RdP (RdP bornés et RdP non bornés), faisant appel aux résultats de la théorie des graphes. [Proth et Xie, 95]

#### 1.4. Réversibilité et états d'accueil

La gestion des aléas est particulièrement importante pour le bon fonctionnement d'un système de production. En pratique, les aléas sont nombreux et fréquents. Citons par exemple les pannes des machines et des moyens de transport, la rupture des outils, les dysfonctionnements dus à la qualité insuffisante de la matière première, les retards de livraison, etc... Après avoir solutionné ces dysfonctionnements, on se ramène habituellement à un état précis (ou à l'un des états prévus par les concepteurs) qui permet de relancer le système.

En terme de RdP, ce problème a trait à la propriété de la réversibilité et à l'existence d'états d'accueil.

##### - Définitions

1- Un RdP  $PN = (P, T, A, W, M_0)$  est dit **réversible** si l'on peut toujours revenir au marquage initial quel que soit le marquage atteint, i.e. :

$$M_0 \in R(M) \text{ quel que soit } M \in R(M_0).$$

2- Un marquage  $M_a$  d'un RdP  $PN = (P, T, A, W, M_0)$  est dit **état d'accueil** s'il peut être atteint à partir de tous les marquages atteignables, i.e. :

$$M_a \in R(M) \text{ quel que soit } M \in R(M_0).$$

Il existe un certain nombre de théorèmes qui permettent de vérifier la réversibilité et l'existence d'états d'accueil pour un RdP. [Proth et Xie, 95]

#### 1- Propriétés structurelles

Pour un système de production un certain nombre de ressources est nécessaire pour garantir l'absence de blocages. Il est donc important de pouvoir vérifier s'il existe un marquage initial qui garantit la vivacité du RdP.

##### - Définitions

1- Un RdP  $N = (P, T, A, W)$  est dit **structurellement vivant** s'il existe un marquage initial  $M_0$  tel que le RdP marqué  $PN = (N, M_0)$  soit vivant.

2- Un RdP  $N = (P, T, A, W)$  est dit **structurellement borné** si le RdP marqué  $PN = (N, M_0)$  est borné quel que soit le marquage initial  $M_0$ .

3- Un RdP  $N = (P, T, A, W)$  est dit **conservatif** s'il existe un vecteur colonne  $X$  qui associe à chaque place  $p$  un poids entier positif  $x(p) \geq 0$  de telle sorte que :

$$X^t \cdot M = X^t \cdot M_0, \forall M_0 \text{ et } \forall M \in R(M_0), \text{ où } X = [x(p_1), x(p_2), \dots, x(p_q)]^t.$$

4- Un RdP  $N = (P, T, A, W)$  est dit **répétitif** s'il existe un marquage initial  $M_0$  et une séquence  $\sigma$  de transitions franchissables dans laquelle chaque transition apparaît un nombre illimité de fois.

5- Un RdP  $N = (P, T, A, W)$  est dit **consistant** s'il existe un marquage initial  $M_0$  et une séquence  $\sigma$  de transitions franchissables qui contient au moins une fois chaque transition et dont le franchissement conduit à nouveau au marquage initial  $M_0$ , **i.e.**  $M_0 \xrightarrow{\sigma} M_0$  et  $V_\sigma > 0$  où  $V_\sigma$  est le vecteur de comptage de  $\sigma$ .

### - Vérification des propriétés structurelles

#### - Théorème 8

Soient  $N = (P, T, A, W)$  un RdP,  $m = \text{card}(P)$ ,  $n = \text{card}(T)$  et  $U$  la matrice d'incidence. Soient aussi  $X$  un vecteur colonne de dimension  $m$  et  $Y$  un vecteur colonne de dimension  $n$  dont les composantes sont des nombres réels ou entiers, **i.e.**  $X \in |\mathbb{R}^m$  et  $Y \in |\mathbb{R}^n$  (ou  $X \in |\mathbb{N}^m$  et  $Y \in |\mathbb{N}^n$ ).

- a)  $N$  est **structurellement borné** si et seulement si :
  - a.1)  $\exists X > 0, X^t \cdot U \leq 0$  ou de manière équivalente :
  - a.2)  $\nexists Y \geq 0, U \cdot Y > 0$
- b)  $N$  est **conservatif** si et seulement si :
  - b.1)  $\exists X > 0, X^t \cdot U = 0$  ou de manière équivalente :
  - b.2)  $\nexists Y, U \cdot Y > 0$
- c)  $N$  est **répétitif** si et seulement si :
  - c.1)  $\exists Y > 0, U \cdot Y \geq 0$  ou de manière équivalente :
  - c.2)  $\nexists X \geq 0, X^t \cdot U < 0$
- d)  $N$  est **consistant** si et seulement si :
  - d.1)  $\exists Y > 0, U \cdot Y = 0$  ou de manière équivalente :
  - d.2)  $\nexists X, X^t \cdot U > 0$ .

### - Intérêts et limites

Les RdP présentent l'avantage d'être visibles ; en effet on peut souvent donner au réseau une forme et une structure se rapprochant visuellement de celle du système modélisé. Cependant, ceci n'est vrai que si le nombre de places est faible, si non le réseau devient de plus en plus compliqué (de grande taille).



L'utilisation des extensions des RdP, comme les réseaux de Petri colorés, peuvent limiter ces inconvénients mais pas les supprimer, car les RdP complexes ont malheureusement très peu de propriétés analytiques, ce qui a pour conséquence que l'étude du comportement des systèmes à l'aide des modèles obtenus ne peut se faire que par simulation. [Proth et Xie, 95]

### **3.4. Simulation**

La simulation représente un outil très utile pour l'évaluation des performances des systèmes flexibles de production. Nous allons donner deux définitions de la simulation. [Belhadri et Mesbahi, 94]

#### **a) Définition informelle**

La simulation est la reproduction pas à pas, événement par événement, de l'évolution de l'état d'une vue d'un système dans le temps, dictée par la logique de changement d'état ou encore c'est la création d'une image dynamique du système, généralement dans le but d'étudier son comportement à la place du phénomène réel.

#### **b) Définition formelle**

La simulation est le processus de conception d'un modèle mathématique et logique d'un système réel et d'essayer ce modèle dans un ordinateur. C'est à dire traduire la description du modèle dans une forme reconnue, au niveau du langage, par l'ordinateur, qui à son tour exécute cette forme pour fournir des sorties qui ensuite, seront analysées.

#### **3.4.1. Etapes d'une modélisation- simulation**

D'après Pegden [Pegden et al, 90], le processus de modélisation- simulation peut être décomposé en plusieurs étapes (Voir fig. 3.6.) :

Etape 1 : Cette étape permet de préciser le contexte dans lequel vont être effectuées les étapes suivantes. Dans cette étape il faut bien identifier le problème qu'on veut résoudre en spécifiant les objectifs qu'on se fixe et le contexte dans lequel on va opérer.

Etape 2 : Elle consiste en une construction d'un modèle en définissant sa description dynamique qui est la définition de la façon dont un élément du système interagit pour causer des changements d'état du système à travers le temps.

Etape 3 : C'est la traduction du modèle dans une forme acceptable par l'ordinateur.

Etape 4 : C'est la vérification et la validation qui servent à évaluer les performances du modèle.

Etape 5 : C'est l'étape d'expérimentation qui est l'exécution du modèle de simulation pour l'obtention des valeurs de sorties.

Etape 6 : C'est l'analyse des résultats des sorties pour fournir des recommandations pour la résolution de la problématique.

Etape 7 : C'est la dernière étape ; elle est l'implémentation et la documentation de l'utilisation du modèle.

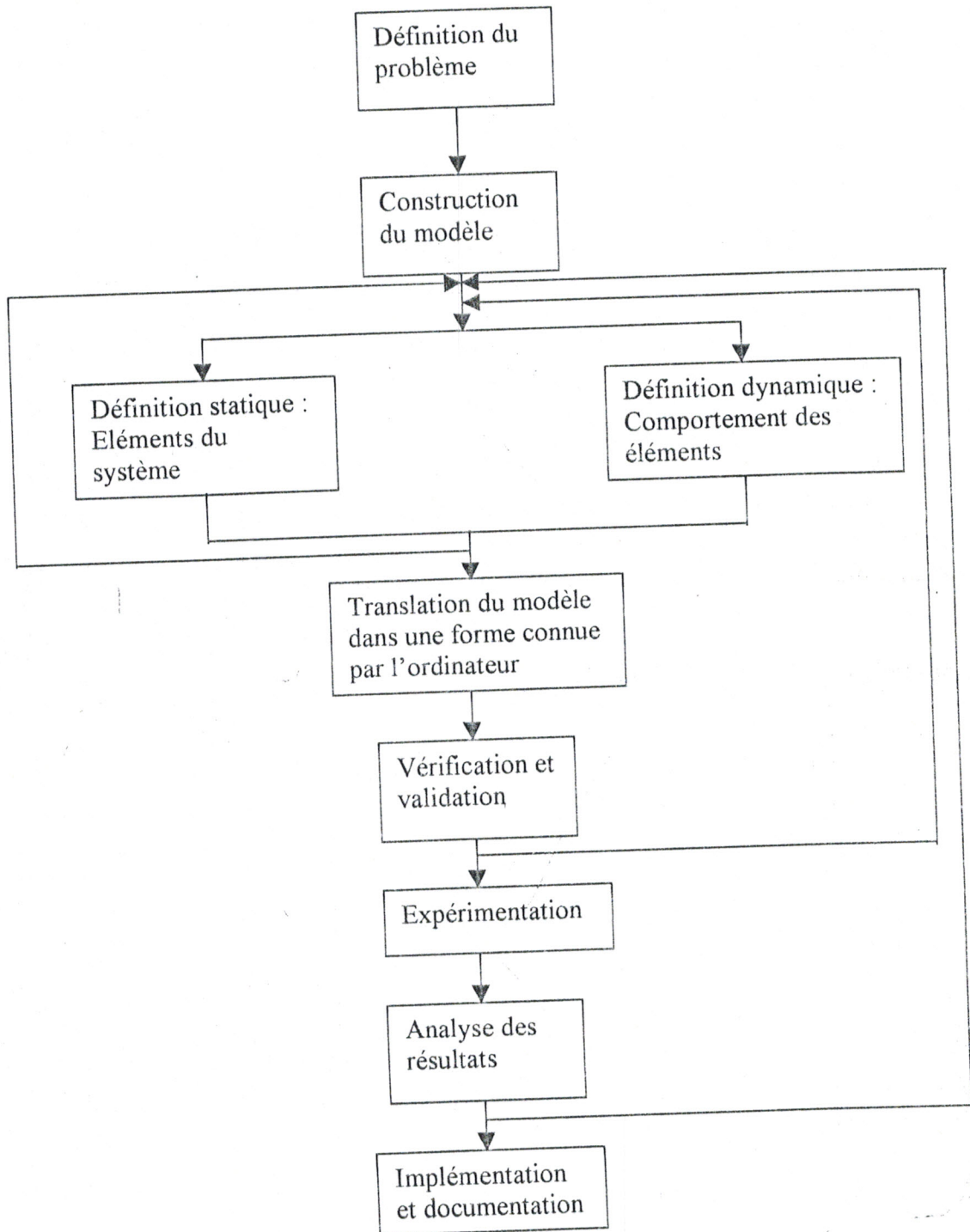


Fig. 3.6. Processus de simulation



### 3.4.2. Les différentes approches de la simulation à événements discrets

Le terme "vue du monde" ou plus précisément "vue de modélisation du monde réel" - (World views), apparaît dans plusieurs publications sur les langages de simulation. Chaque langage de simulation a une vue implicite du monde. Nous pouvons exprimer cela de la manière suivante :

- Le monde consiste en un ensemble d'entités ou transactions qui circulent à travers le système. Ces entités sont décrites, caractérisées et identifiées par leurs attributs.
- Ces entités interagissent avec les ressources et les activités d'une manière conforme à certains conditions qui déterminent la séquence d'interactions.
- Ces interactions sont perçues ou vues comme un système générateur d'événements qui résultent des changements d'états dans le système.

On distingue trois approches différentes, couramment utilisées pour l'implémentation des mécanismes de cette vue de modélisation : [Pegden et al, 90]

- \* Par gestion (ordonnancement) des événements : "event scheduling".
- \* Par balayage de l'activité : "Activity scanning".
- \* Par interaction de processus : "Process interaction".

Avant d'exposer ces différentes approches, nous définissons deux types d'événements.

- Événement inconditionnel : C'est un événement dont l'exécution dépend de l'occurrence de son temps d'horloge, c'est à dire que l'exécution dépend entièrement du temps.
- Événement conditionnel : L'exécution de ce type d'événements dépend d'autres conditions qui sont fonction de l'état du système.  
Exemple : une ressource occupée.

#### 1) Approche par gestion (ordonnancement) d'événements

Cette stratégie consiste à répertorier une succession d'événements inconditionnels occurrents à des instants discrets dans le temps. L'utilisateur construit un modèle de simulation par :

- La définition de chaque événement occurrent dans le système.  
Ex : arrivée d'une pièce.
- La spécification des causes et des effets de chaque événement.
- La création de mécanismes pour exécuter le changement d'événement.
- Le lien entre un événement et un autre.

Quand une pièce se présente devant un stock d'entrée d'une machine, et que celle-ci est libre, le début de travail peut commencer. La procédure par laquelle les événements de cette nature sont monitorés est appelée "balayage d'activité".

### 3) Approche par interaction de processus

Cette stratégie voit le monde comme un ensemble d'entités en compétition et en coopération modélisé en terme de processus concurrents.

Pour les systèmes de production, les processus de type pièce interagissent en général avec tous les autres processus, ce qui traduit leur cheminement à travers le système.

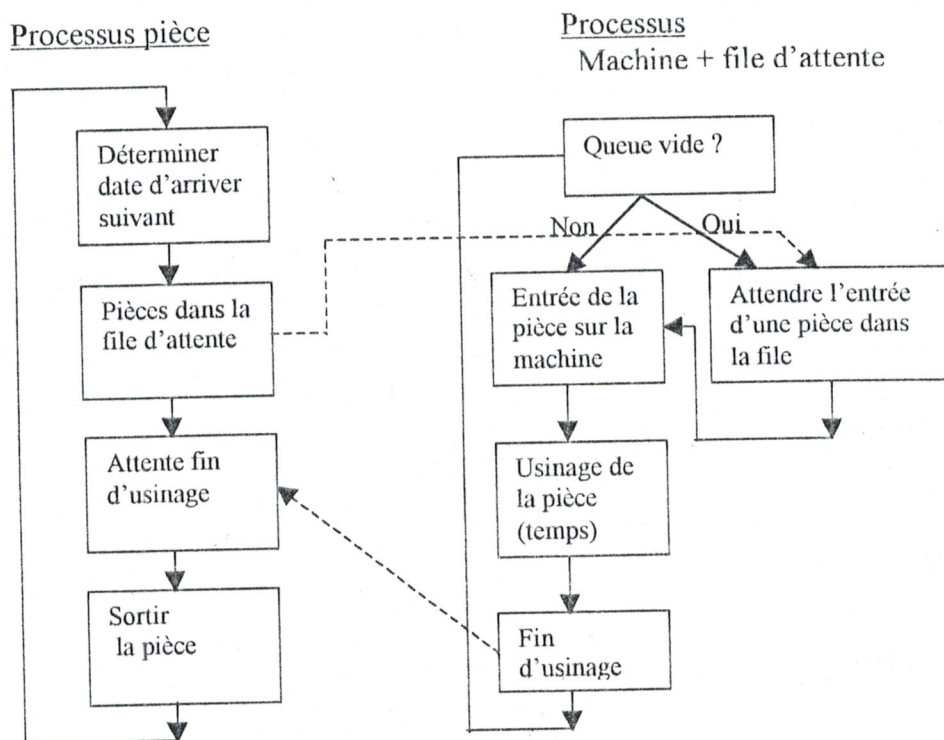


Fig. 3.8. Interaction des processus (pièce, machine + file d'attente).

La modélisation se fait donc en décrivant les déplacements des pièces à travers les différents processus (machine, moyen de manutention...). La fig. 3.8. représente la description de deux processus ; le processus pièce et le processus machine en interaction.

Dans cette approche, on constate que la structure du modèle est proche de la structure du système réel, ce qui facilite la lecture et la validation du modèle. [Belhadri et Mesbahi, 94]



### **3.5. La simulation appliquée aux systèmes flexibles de production**

Essayons tout d'abord de placer la simulation dans le contexte des systèmes de production, ceci permettra de donner une idée sur l'étendu du domaine de la simulation dans ces systèmes.

#### **3.5.1. Différents champs d'utilisation de la simulation dans les systèmes de production.**

La simulation permet de prévoir le comportement des systèmes de production qui sont souvent complexes, en calculant le mouvement et l'interaction des composants du système. De ce fait, la simulation nous fournit la possibilité d'expérimenter sur un modèle plutôt que sur le système réel. Par conséquent elle permet d'envisager les changements adéquats avant l'acquisition de l'installation du système.

On rencontre trois modes distincts d'utilisation de la simulation des systèmes de productions : [Pegden et al, 90]

a) Comme une aide à la conception et l'analyse de la configuration de l'atelier : choix d'équipements, politique de fonctionnement, évaluation des problèmes, ce sont les rôles traditionnels joués par la simulation.

b) Comme moyen d'ordonnancement ; Cette utilisation permet au concepteur d'explorer l'algorithme d'ordonnancement existant et/ou de trouver l'ordonnancement optimal.

c) Comme moyen de contrôle temps- réel (on- line) ; La simulation est activée périodiquement pour lire les données mise à jour (actualisées), exécuter une simulation de l'ordonnancement en tenant compte des nouvelles données. En fonction des résultats, on décide, soit de valider l'algorithme d'ordonnancement, soit de le modifier en conséquence ou encore d'appeler une intervention humaine.

Lorsqu'une simulation est utilisée comme outil de conception, l'étude est motivée par des questions tel que :

- Quelle sera l'objectif de cette conception ? Atteindrait-elle nos objectifs de production ?
- Où sont les goulots d'étranglement ? Que peut-on changer pour améliorer la production ?
- Quel est la meilleure, parmi plusieurs alternatives de conception ? Comment vont changer les performances en fonction du changement des différents paramètres ?
- Comment les défaillances, affecteront - elles les performances ?

Notre travail de simulation s'inscrit principalement dans cette catégorie d'utilisation.

### 3.5.2. Avantages de la simulation

Pegden [Pegden et al, 90] cite quelques avantages de la simulation des systèmes de productions, nous pouvons les résumer comme suit :

- D'abord la simulation est plus facile et plus proche des chefs d'ateliers, par rapport aux modèles analytiques.
- En plus, elle peut avoir plus de crédibilité, parce que son comportement est comparable à celui du système réel ; elle exige moins de suppositions, et par conséquent elle capture plus de caractéristiques du système réel.
- De nouvelles stratégies, procédures de fonctionnements, règles de décisions, structure organisationnelle, flux d'informations... etc., peuvent être explorées sans perturber le système réel.
- On peut étudier le comment et le pourquoi de certains phénomènes occurrents.
- Le temps peut être contrôlé : il peut être compressé, allongé, ce qui nous permet de voir les comportements transitoires.
- La perception des variables et de leurs interactions (responsable de l'affectation des performances) est plus facile.
- Les goulots d'étranglement dans un système, peuvent être identifiés.

### 3.5.3. Inconvénients de la simulation

Malgré les multiples avantages de la simulation, elle n'est pas sans inconvénients, on peut citer :

- La construction du modèle exige un spécialiste, ainsi la qualité de l'analyse dépend de la qualité du modèle et de l'habileté (expérience) du modélisateur. Cette expérience varie largement d'un praticien à un autre.
- Un problème peut se poser dans l'interprétation des résultats de simulation, en particulier pour les modèles stochastiques. En effet, dans un modèle stochastique, il est souvent délicat de déterminer, si une observation faite durant une exécution est due à une corrélation significative dans le système ou due à un comportement aléatoire dans le modèle.
- Parfois la simulation est très coûteuse en temps, des méthodes rapides et approchées utilisant des méthodes analytiques peuvent être préférables.



### 3.5.4. Le besoin à une simulation basée sur l'animation graphique

Nous pouvons être amenés à introduire une simulation animée temps réel, L'animation est intéressante dans la simulation des systèmes de production dans deux phases.

- Phase de vérification et validation du modèle.
- Phase d'exploitation du modèle pour étudier le comportement du système et appréhender les problèmes.

Dans notre travail, la nécessité de la simulation animée est motivée principalement par la seconde phase. Tout de même, nous allons présenter les avantages de la simulation animée pour les deux phases.

#### a) Vérification du modèle

Rappelons que la vérification d'un modèle est le processus qui permet de déterminer si un modèle de simulation s'exécute comme prévu. Ce processus n'assure pas que le modèle représente fidèlement le système réel, mais seulement que le modèle est vide d'erreurs.

La détection des erreurs dans la modélisation est difficile (oublier d'initialiser de variables, de libérer des ressources après l'achèvement d'une opération... etc.). Ces erreurs peuvent être très coûteuses en temps pour les isoler, encore plus elles peuvent devenir indécélables dans le modèle ce qui conduit à des conclusions incorrectes sur les performances du système. [Pegden et al, 90]

Par exemple en regardant une animation, on peut observer qu'une palette peut retenir le contrôle d'une ressource alors qu'elle a quitté la station de travail.

#### b) Validation du modèle

La validation du modèle de simulation consiste à déterminer si le modèle est une approximation suffisamment adéquate du système réel. Cette phase est liée aux objectifs de l'étude par simulation.

Parce que l'animation nous fournit une vue du fonctionnement du système, nous pouvons visualiser l'impact et l'interaction des simplifications faites dans la phase de modélisation. Souvent une simplification qui semble raisonnable durant la phase d'abstraction et de modélisation, devient inadéquate quand elle est visualisée dans une animation graphique. [Pegden et al, 90]

L'animation permet aussi de communiquer rapidement le fonctionnement du système aux clients (chefs d'ateliers) qui connaît le système, mais qui a peu de connaissances en modélisation et en simulation. [Pegden et al, 90 ; Aghetta, 96]

### c) Phase d'exploitation

Parmi les problèmes qui se posent dans la simulation des systèmes flexibles de production (FMS), il y a le problème de l'interprétation des résultats de la simulation. Cette difficulté vient justement de la flexibilité de ces systèmes (capacité de réaction à des changements de différentes natures). Pour palier à ce problème une alternative consiste à recourir à une simulation animée.

Breugnot [Breugnot et al, 91] justifie la nécessité d'une animation par les facteurs suivants :

- Les statistiques sommaires obtenues par simulation ne donnent pas les informations nécessaires pour appréhender le comportement transitoire.
- L'examen d'un listing issu d'un simulateur, pour identifier la localisation d'un problème ou un goulot d'étranglement est suffisamment fastidieux pour justifier une animation.
- Une simulation animée fournit un moyen facile pour vérifier qu'un système se comporte convenablement.

Dimirovski [Dimirovski et al, 94] voit que l'animation rend visuellement tangible le fonctionnement du FMS et révèle les inconsistances ou les erreurs des algorithmes de contrôle et d'ordonnancement.

En outre, l'animation est maintenant considérée comme l'un des meilleurs moyens de communiquer ce type de résultats.

#### **3.5.5. Limitations de l'animation graphique dans l'interprétation des résultats**

Quand une simulation est animée, la vitesse d'exécution de la simulation est usuellement diminuée, pour permettre une visualisation dynamique compréhensible par l'utilisateur ; une simulation sans animation peut rester quelques minutes pour terminer le processus de production, en revanche une simulation animée peut en rester plusieurs jours.

En plus, s'il s'agit d'une simulation statistique (processus stochastiques), il est abusif de tirer des conclusions sur un tel système avec un nombre petit d'échantillons.

Parfois, si on veut voir, par exemple, l'évolution dans le temps d'une longueur d'une file d'attente, la visualisation d'une animation n'est pas la meilleure option, un simple affichage en temps réel de la variable correspondante représente mieux l'information.



*Chapitre 4*

**Conception et développement du logiciel**

## 4.1. Introduction

La conception des gros logiciels pose un certain nombre de problème de conception, de maintenance... etc. De plus on aime avoir des logiciels qui soient fiables, réutilisables, extensible et de qualité. C'est dans ce souci, qu'un nouveau concept de programmation a fait son apparition, c'est la programmation par objet ou programmation orientée objet (POO). Les langages orientés objet sont bâtis autour des concepts de base suivants :

- L'objet.
- La classe.
- L'envoi de messages.
- La hiérarchie des classes et l'héritage qui en découle.

Pour ce faire, nous avons utilisé l'outil de développement Delphi sous Windows 95, qui offre un environnement conforme à l'esprit de la programmation orientée objet POO. L'utilisation de l'analyse fonctionnelle de la supervision présentée au chapitre 2, est nécessaire pour la réalisation du logiciel, puisqu'elle conditionne le fonctionnement global de la cellule.

## 4.2. Définitions [Meyer, 90] [Robert, 89]

**a) L'objet :** L'approche orientée objet utilise un nouveau concept celui d'objet. Un objet est un regroupement en un concept unique des données et des procédures chargées de les exploiter. C'est aussi une portion de connaissance possédant son propre contrôle. Un logiciel conçu par l'approche orientée objet est donc une collection d'objets différents, la connaissance (données et programmes) est distribuée entre chacune de ces entités.

### - Structure d'un objet

Les deux composantes de l'objet sont les données et les procédures.

Les données sont en quelques sorte l'identité (la description) de l'objet, ils représentent son savoir déclaratif, ils peuvent être des : attributs, champs, variables,... etc.

Les procédures (parfois, on rencontre le mot "Méthodes") sont le comportement de l'objet durant la vie du système ; elles représentent son savoir descriptif.

**b) La classe** C'est un regroupement d'objets ayant une même structure de données et un même comportement (procédures) ; C'est donc une sorte de moule à partir duquel on peut fabriquer autant d'exemplaires (objets) que l'on veut. Plus formellement, une classe est définie par un modèle d'un ensemble de données et d'un ensemble de procédures.

**c) L'envoi de message** L'accès et l'utilisation des objets se font par un moyen unique : "l'envoi de messages". L'envoi de messages constitue le moyen de



communication de l'objet avec lequel il interagit avec l'extérieur. La fig. 4.1. représente la structure d'un objet.

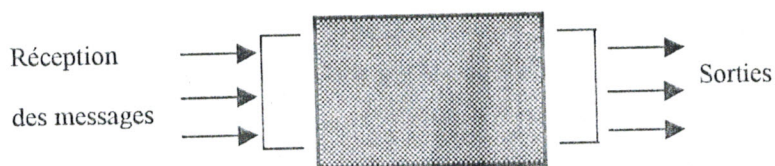


Fig. 4.1. L'objet : une boîte noire

### 4.3. Technique de conception par objet

La résolution d'un problème par l'approche orientée objet repose en premier lieu sur l'expérience du concepteur, mais il existe un processus de développement orienté objet qui consiste-en : [Belhadri et Mesbahi, 94]

- 1) L'identification des classes d'objets qui interviennent dans le monde réel.
- 2) L'identification de leurs actions qui caractérisent leur comportement.
- 3) L'identification des relations entre les différentes classes.
- 4) L'implémentation des classes.

Plusieurs auteurs [zeigler, 87 ; Zeigler, 90 ; Vujosevic, 90] ont décrit la compatibilité entre la programmation orientée objet (POO) et les systèmes à événements discrets (SED).

La modélisation orientée objet qui est l'étape préalable à toute simulation consiste à définir, le système à simuler, par un ensemble de classes d'objets de même caractéristiques. Cette modélisation prend en compte les différentes entités du système réel pour décrire leur comportement. Les classes d'objets sont de ce fait, la représentation informatique des entités physiques matériels, (machines, produits,...) et abstraites (événements, horloge,...) avec leurs attributs. Ces classes d'objets permettent de représenter les caractéristiques du système de production à simuler.

Après la formulation des classes d'objets du système et leurs attributs, on décrit leurs interactions par des méthodes (procédures). La programmation de la logique de changement d'états (événements) consiste à déterminer les objets en activant les méthodes correspondantes par le mécanisme d'envoi de messages, ensuite on fait la synchronisation de ces opérations.

Finalement, nous pouvons dire que la POO se prête bien à la simulation des SED.

#### 4.4. Le système de développement Delphi

##### - Motivation

Delphi est un système de développement d'applications Windows qui possède les propriétés nécessaires pour en faire un outil extrêmement puissant.

En plus Delphi possède un langage de programmation puissant (Pascal Objet) qui offre un large éventail de possibilités pour le concepteur, entre autre la POO. Burda et Farber [Burda et Färber, 96] exposent en détail les performances de Delphi 2.

Pour notre travail de simulation, nous avons opté pour ce système de développement compte tenu de sa richesse en concepts proches de la simulation à événements discrets ainsi que pour l'ergonomie qu'il offre pour la conception de l'interface graphique utilisateur.

#### 4.5. Structure générale du logiciel

Le logiciel que nous avons développé est structuré en plusieurs unités, chacune réalise une fonction donnée. Le synoptique de la figure 4.2 explicite la structure du logiciel avec les unités les plus importantes, en liaison avec la simulation.

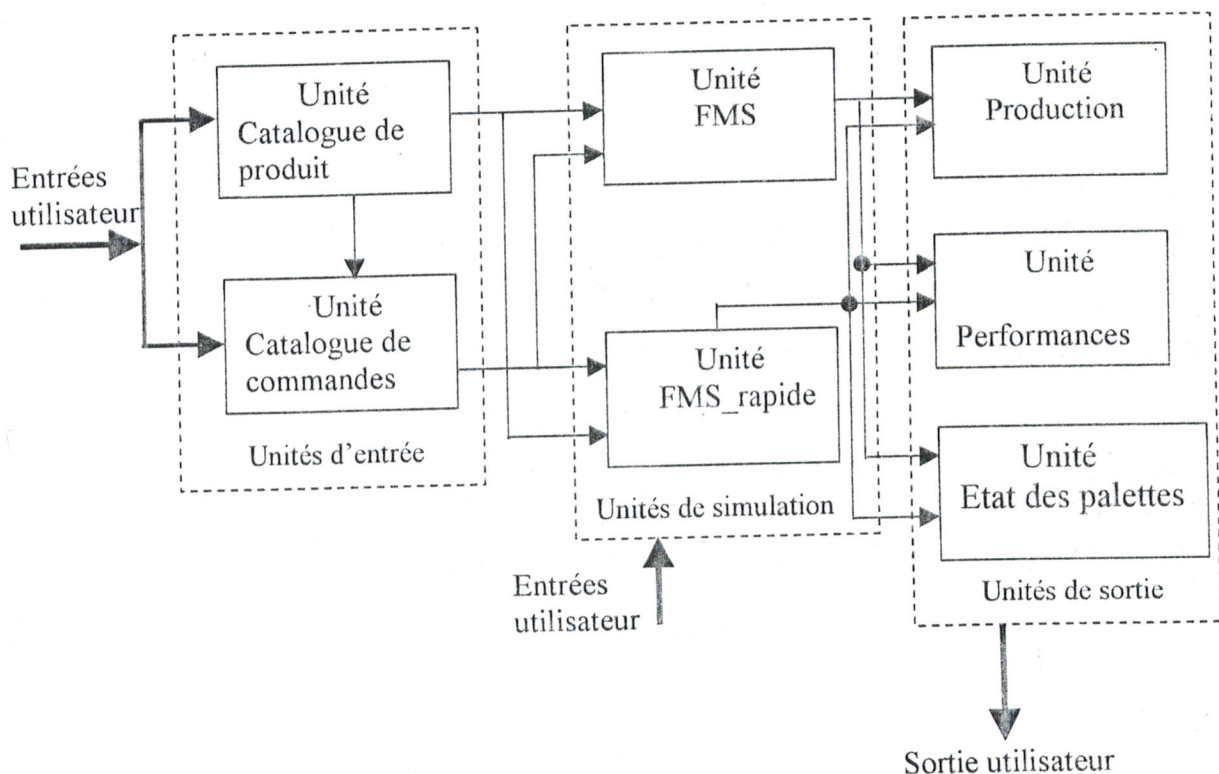


Fig. 4.2. Architecture du logiciel.



### 4.5.1. Bases de données du système

Le logiciel manipule un volume énorme de données, ces données sont lues par le logiciel pour fournir les simulations appropriées. C'est pourquoi, nous avons vu la nécessité de concevoir des bases de données (BdD) qui servent à contenir l'ensemble de données du système. Les résultats de simulation, sont eux aussi représentés dans des bases de données, pour pouvoir les exploiter.

Néanmoins, certaines données relatives à la commande de la cellule (vitesse du convoyeur, fréquence de sortie des palettes,... etc. ) sont définies depuis les fenêtres de simulations. Nous avons conçu les bases de données avec une interface Dbase pour Windows, Les structures de ces BdD seront détaillées au chapitre 5.

- Unité catalogue de produit : Elle commande la BdD catalogue de produit.
- Unité catalogue de commandes : Elle commande la BdD catalogue de commandes.
- Unité FMS : Elle englobe le programme qui permet de générer la simulation animée.
- Unité FMS\_rapide : Elle englobe le programme qui permet de générer la simulation sans animation.
- Unité Performances : Elle commande la BdD "performances".
- Unité Production : Elle commande la BdD "production".
- Unité Palettes : Elle commande la BdD "Etat des palettes".

Les BdD catalogue de produits et catalogue de commandes sont des BdD d'entrée du système, c'est à dire, qu'elles sont chargées de recevoir les données de l'utilisateur. Les unités performances, production et palettes, sont des BdD de sortie du système, c'est à dire qu'elles sont chargées d'afficher les résultats de la simulation.

### 4.5.2. Simulation

La simulation de la cellule regroupe deux types de simulation, la simulation avec animation graphique et la simulation sans animation graphique. Bien que la simulation animée présente plusieurs avantages, il peut être nécessaire de passer à une simulation accélérée, pour avoir les résultats plus rapidement. Pour chaque type de simulation, nous avons prévu une unité correspondante, l'unité "FMS" pour la simulation animée et l'unité "FMS\_rapide" pour la simulation sans animation. L'utilisation de l'une ou l'autre dépend des objectifs de l'étude par simulation.



#### 4.6. Simulation avec animation

C'est l'unité FMS qui permet de générer cette simulation, elle est représentée par une interface graphique, qui visualise la cellule flexible, avec tous les paramètres de commande. La première étape dans la conception de cette unité était, d'abord de construire les graphismes qui représentent les différentes composantes du système, à savoir :

- Les deux stations d'assemblage SA1 et SA2.
- La station de stockage-déstockage SSD.
- La station de convoyage et d'aiguillage CC1.
- Les palettes.
- Le panneau de contrôle avec ses différentes commandes.

Pour ce faire, nous avons exploité les fonctions graphiques offertes par Delphi, pour le dessin de ces différentes composantes. Pour la conception du convoyeur, nous l'avons décomposé en six tronçons colorés d'une façon alternée, et ceci afin de pouvoir mettre en évidence l'animation graphique (voir Fig. 4.3). Chaque tronçon est commandé séparément, avec la même vitesse. La difficulté était donc de synchroniser les mouvements de ces différents tronçons, surtout que deux de ces tronçons sont en forme de coude.

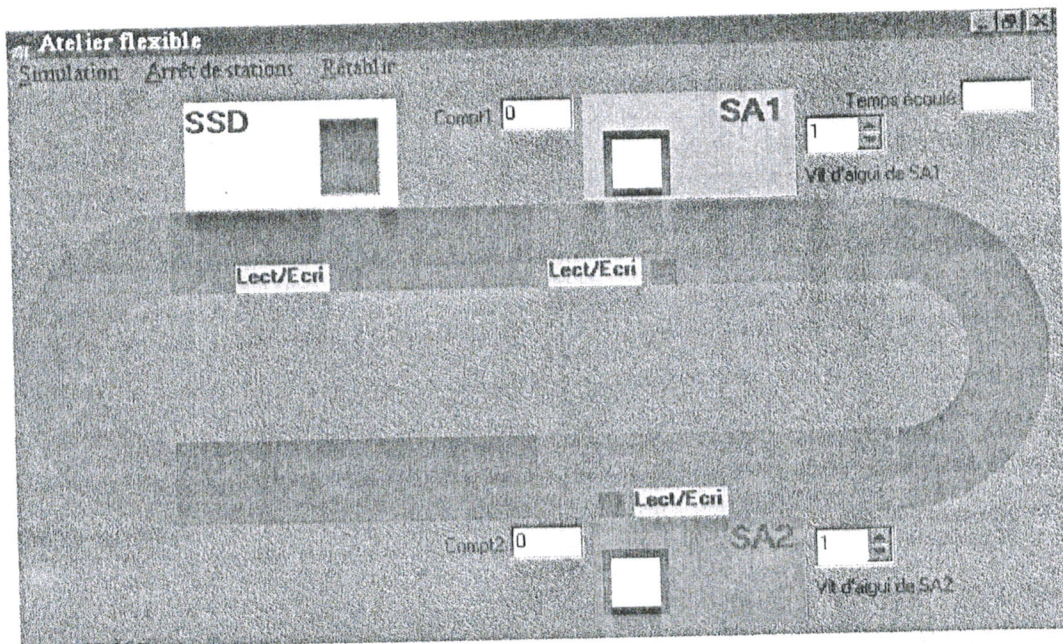


Fig. 4.3. Constituants de la cellule flexible.

Pour la création des palettes, nous avons rencontré certains problèmes, qui sont exigés par les facteurs suivants :



- Les palettes doivent être identifiées et localisées, pour pouvoir suivre leurs mouvements dans la cellule (simulation de l'étiquette électronique).

- La possibilité de créer un grand nombre de palettes.

- la possibilité de donner l'animation aux palettes ainsi créées, pour refléter leur routage à travers la cellule.

Pour répondre à ces contraintes, nous avons exploité les possibilités de la programmation orientée objet (POO). En effet, la solution consiste à créer une classe d'objets "palettes", qui regroupe toutes les caractéristiques nécessaires. Par conséquent, pour l'obtention d'une palette, il suffit de créer un objet du type palette par le mécanisme d'instanciation. Cependant et puisque, Delphi utilise le principe de jeu de construction ; nous disposons pour cela d'une palette richement garnie de composants (objets) qui sont en quelque sorte les pièces du jeu de construction, nous avons exploité une classe d'objets (qui présente des caractéristiques répondant aux exigences) en la détournant de son usage, pour être adapté à de nouvelles fonctions. Cette technique de conception nous a permis de s'affranchir d'un grand nombre de problèmes qui pourraient survenir par l'utilisation d'une programmation classique. Ainsi, nous avons créé 100 palettes (au lieu de 30 que contient la cellule réelle), et ceci pour permettre des simulations plus importantes.

Une fois la phase de représentation graphique des composantes de la cellule terminée, il faut concevoir le programme qui permet de réaliser l'animation graphique de ces composantes. Cette étape très cruciale dans le développement du logiciel, a révélé des problèmes de programmation qui peuvent être résumés par les deux facteurs suivants :

- Le caractère temps réel de l'évolution de la simulation animée : la simulation doit pouvoir refléter le fonctionnement réel du système, qui est soumis à des contraintes temporelles.

- Les interactions entre les différentes composantes du système, qui mettent en évidence les aspects de concurrence et de conflit.

Pour prendre en compte ces aspects et faire face à toutes les situations de conflit, il s'avérait indispensable de faire appel aux techniques de la programmation concurrente.

#### **4.7. Programmation concurrente et programmation temps réel**

##### **4.7.1. Introduction**

La programmation concurrente est caractérisée par l'introduction de la notion d'interruption envoyée par un périphérique au processeur. La programmation temps réel est relativement proche de la programmation concurrente.

- Un programme en temps réel, tout comme un programme concurrent, gère des périphériques.



- Les techniques de la programmation concurrente interviennent dans la programmation en temps réel.

Un programme en temps réel comporte toutefois une dimension supplémentaire : celle de contrôle d'un système externe [Computer, 84 ; Spector, 84] (ce qui entraîne de la part du programme, des contraintes de temps).

Considérons la station d'assemblage de la cellule flexible. Les périphériques sont constitués dans ce cas par des capteurs de présences et de proximités des palettes (périphériques d'entrée) et des actionneurs (périphériques de sortie) permettant le déplacement des palettes (aiguilleurs). Le programme peut avoir pour rôle de lire le contenu de l'étiquette électronique d'une palette, décider éventuellement de l'aiguiller vers une station d'assemblage. Pour remplir cette fonction, le programme doit réagir en temps réel (autrement dit dans un délai maximum à ne pas dépasser) à toute présence de palette au niveau de l'entrée d'une station, par l'envoi de la commande à l'aiguilleur. On retrouve cette caractéristique dans tous les programmes en temps réel :

- Le programme reçoit des informations du système à contrôler.
- Le programme réagit par l'envoi de commandes au système à contrôler.

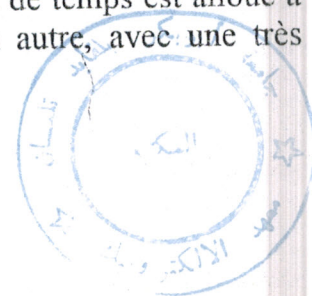
Pour reproduire une simulation fidèle du système réel, et vu la nature temps réel de l'évolution du système réel, il faut introduire dans le programme les techniques de la programmation concurrente et temps réel.

En effet, les palettes qui sont introduites dans la cellule flexible ne sont pas complètement indépendantes les unes des autres. Au contraire, il se peut par exemple que plusieurs d'entre elles désirent agir sur une même variable ou accéder à une même ressource. Une telle situation risque de conduire à des erreurs et doit être contrôlée.

Il est donc clair que les palettes évoluent dans le temps d'une façon parallèle et simultanée. Pour répondre à ce besoin de parallélisme, Il était nécessaire de concevoir le programme de simulation comme un programme concurrent. Cette conception fait intervenir la notion de processus. Un processus peut être comparé à un programme séquentiel qui exécute des instructions l'une après l'autre. La juxtaposition de plusieurs processus va permettre de répondre au besoin de parallélisme.

Toutefois, un vrai parallélisme n'est possible que lorsque nous disposons de plusieurs processeurs, dans le cas contraire, nous réalisons un pseudo-parallélisme, qui est caractérisé par le fait que : [Schiper, 86]

- A tout instant seul un processus est en exécution.
- La commutation de processus, c'est à dire le passage de l'exécution d'un processus à un autre, s'effectue à l'insu des processus. Un quantum de temps est alloué à chaque processus (ou tâche), puis on passe d'un processus à un autre, avec une très grande vitesse. Ainsi on émule un fonctionnement en multitâche.





### 4.7.2. Gestion de la simulation animée

Pour la génération d'une simulation du système, les différents processus ont besoin d'un moyen de cadencer leur fonctionnement dans le temps, pour ce faire, nous avons été amenées à introduire des composants *horloge* ou "Timer", produisant des interruptions à intervalles réguliers et qui permettent de commander les différentes fonctions du système. Pour cela nous avons utilisé sept timer qui remplissent les fonctions suivantes :

- Un timer timer1, qui sert à commander le mouvement du convoyeur et des palettes, nous avons choisit une période de 50ms, permettant d'obtenir une sensation visuelle satisfaisante.
- Un timer timer2, qui permet de commander les blocs de lecture/écriture, lors du passage d'une palette. La période de ce timer est de 220ms, activant un clignotement de ces blocs, pour indiquer à l'utilisateur l'opération de lecture/écriture.
- Un timer timer3 de période 1s, qui sert à compter les temps d'exécutions des opérations s'effectuant dans la station SA1.
- Un timer timer4 de période 1s, qui sert à compter les temps d'exécutions des opérations s'effectuant dans la station SA2.
- Un timer timer5 de période 1s, qui sert à compter les temps d'exécutions des opérations s'effectuant dans la station SSD.
- Un timer timer6, qui permet de scruter l'état des zones de retour des palettes, nous avons choisit une période de 50ms pour avoir des informations au rythme de l'évolution du système.
- Un timer timer7 de période commandable, servant à commander la fréquence de sortie des palettes.

### 4.7.3. Présentation de quelques problèmes

Nous allons présenter, deux problèmes typiques à la programmation concurrente appliqués à notre cas de système. Le premier problème est celui de l'exclusion mutuelle, le second représente le mécanisme de synchronisation.

#### a) Exposé du problème 1

Supposons pour simplifier, deux palettes P1 et P2 qui doivent effectuer des opérations successivement dans la station d'assemblage SA1 puis dans la station SA2. Une situation peut se présenter : il se pourrait que la palette P1 accède à la station SA1 pour subir des opérations d'assemblages. Avant que la palette P1 n'ait fini ses opérations, la palette p2 accède à la station SA1, ce qui conduit à un conflit (parce qu'une station n'opère que sur une palette à la fois).

Pour éviter une telle situation, il faut empêcher l'une des palettes d'utiliser la station, pendant que l'autre palette l'occupe. Il s'agit de réaliser ce qu'on appelle une *exclusion mutuelle* sur l'accès à la station.

## b) Solution au problème 1

Dans l'exemple précédent, on désire l'exclusion mutuelle sur l'accès à la station d'assemblage. Dans ce cas la station d'assemblage est appelée, une *ressource critique* : une ressource critique est un objet qui doit être accédé en exclusion mutuelle. Il existe plusieurs moyens de réaliser l'exclusion mutuelle, nous pouvons citer brièvement :

\* Exclusion mutuelle par attente active : Dans cette stratégie, l'état est manipulé par deux procédures ; procédure début et procédure fin.

La procédure début, annonce la demande d'entrer à la ressource critique.

La procédure fin, annonce la libération de la ressource.

Pendant qu'un processus détient la ressource, un autre processus voulant accéder à la même ressource, fait de l'attente active, c'est à dire à chaque fois il teste l'état de la ressource, si celle-ci est occupée, il attend jusqu'à sa libération.

Bien qu'il soit possible de réaliser l'exclusion mutuelle avec cette approche, en pratique, il n'est en effet pas réaliste de faire de l'attente active (c'est à dire évaluer continuellement une condition jusqu'à ce qu'elle soit vraie) : le gaspillage du processeur qui en résulterait n'est pas tolérable [Schiper, 86].

**N.B.** Les algorithmes qui seront présentés, sont exprimés en utilisant la syntaxe du langage Pascal. Cependant, dans le but de faciliter la lecture des algorithmes, nous avons pris certaines libertés par rapport à la syntaxe originale de Pascal.

\* Exclusion mutuelle à l'aide de verrous : Nous allons voir maintenant un autre outil permettant d'implémenter une exclusion mutuelle sans faire de l'attente active. Il s'agit du verrou. Un verrou V est une variable du type suivant :

```
Type Verrou = record
    Etat : (ouvert, ferme) ;
    En_attente : liste de processus en attente
End record ;
```

Un verrou peut donc être ouvert ou fermé (initialement il est ouvert). Le verrou contient également une liste d'attente ; c'est la liste des processus devant un verrou fermé. Un processus manipule un verrou grâce aux deux procédures verrouiller et déverrouiller définies ainsi (voir fig. 4.4.) :

```
Procédure Verrouiller (Var V : Verrou) ;
Code
if V.etat = ouvert then V.etat := ferme
    else
        bloquer le processus dans la liste "V.en_attente"
end if
end verrouiller ;
```



**Procédure** deverrouiller (Var V : Verrou) ;

Code

**if** liste "V.en\_attente" non vide **then**

    débloquer un processus en attente dans la liste "V.en\_attente"

**else**

        V.etat :=ouvert

**end if**

**end** deverrouiller ;

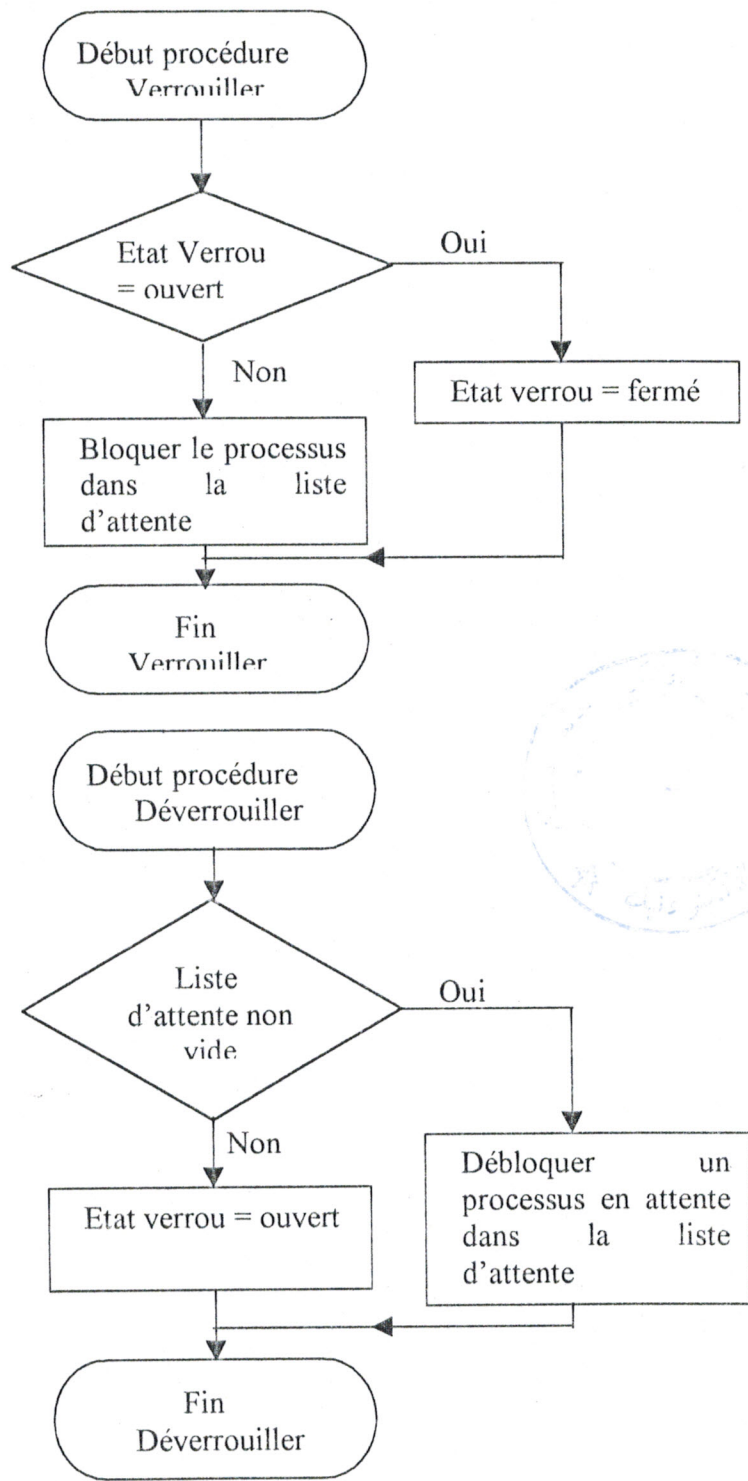


Fig. 4.4. Organigrammes des procédures verrouiller et déverrouiller.

En utilisant les verrous, l'accès exclusif à une station s'exprime ainsi :



**Processus P1 (\* pellette P1 \*)**

Code

```

.....
verrouiller (V) ;
code correspondant ; (*Accès à la station*)
deverrouiller (V) ;

```

```

end P1 ;

```

Le processus P2 (palette P2) s'exprime de façon analogue. Pour notre travail, nous avons adopté une variante de cette solution pour réaliser les exclusions mutuelles.

Il existe encore d'autres outils qui permettent l'exclusion mutuelle à savoir :

- Exclusion mutuelle par masquage d'interruption.
- Exclusion mutuelle à l'aide de sémaphores.

**c) Exposé du problème 2**

Considérons le fonctionnement de la station de stockage- déstockage (SSD) :

Une palette arrivant à la station SSD doit être en premier lieu transférée par le robot pour que l'opération de déchargement puisse être entamée. Les palettes ainsi déchargées, pourront ensuite rejoindre la file d'attente (en attente d'être introduites de nouveau dans la cellule). La fig. 4.5. Schématise un tel fonctionnement.

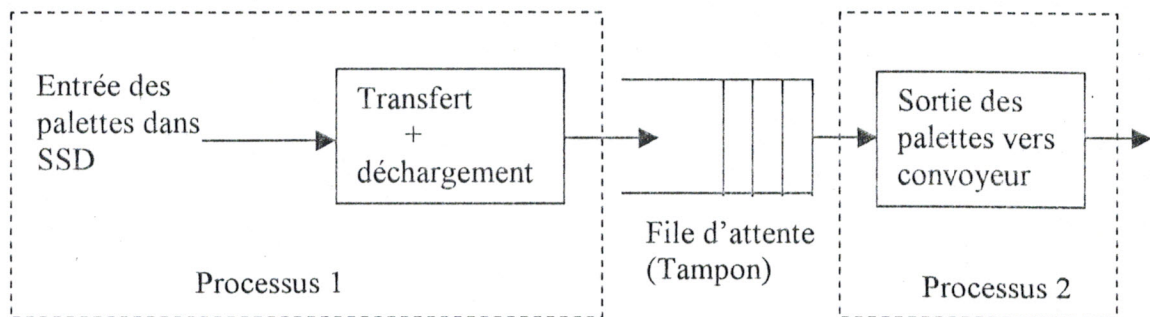


Fig. 4.5. Fonctionnement de SSD.

Le problème consiste à *synchroniser* entre ces deux processus 1 et 2. Ce problème peut être ramené à un problème du type producteur- consommateur (Producer- consumer). En effet, la communication entre les deux processus se passe selon le schéma suivant :

- Le producteur (Processus 1) génère des palettes et les dépose dans une zone commune aux deux processus (on suppose une zone commune d'une taille de  $n$  palettes,  $n > 0$ ).

- Le consommateur (processus 2) prélève les palettes et les utilise.

- Les palettes, dans la zone commune (tampon), sont gérées en file d'attente.

Dans le cas général, il faut résoudre le problème de synchronisation suivant :

- Le producteur (processus 1) doit être mis en attente s'il veut déposer une palette alors que la file d'attente est pleine.
- Le consommateur (processus 2) doit être mis en attente s'il veut prélever une palette alors que la file d'attente est vide.
- Le producteur (processus 1) en attente doit être débloqué dès que le consommateur prélève une palette (puisque à ce moment la file d'attente n'est plus pleine).
- Le consommateur (processus 2) en attente doit être débloqué dès que le producteur dépose une palette (puisque à ce moment la file d'attente n'est plus vide).

#### d) Solution au problème 2

Pour résoudre ce problème, il existe différents outils de synchronisation : [Schiper, 86 ; Tschirhart, 90]

- Synchronisation à l'aide d'événements.
- Synchronisation à l'aide de sémaphores.
- Synchronisation à l'aide de moniteurs.
- Synchronisation à l'aide de données : la boîte aux lettres.
- Synchronisation par rendez - vous.

Néanmoins, dans cette section, nous allons présenter une solution utilisant des "moniteurs" qui a été adopté pour l'implémentation du programme de synchronisation.

Définition : La notion de *moniteur* a été proposée par C.A.R. Hoare en 1974 [Hoare, 74]. Un moniteur est une unité syntaxique analogue à un module ; il permet de regrouper des variables ainsi que les procédures agissant sur ces variables. Mais en plus le moniteur assure l'exclusion mutuelle sur les procédures déclarées dans le moniteur. La synchronisation s'exprime à l'intérieur d'un moniteur grâce à des *signaux*. Un signal, peut être manipulé par les procédures *wait* et *send*.

Soit S un signal : l'exécution de S.wait bloque (inconditionnellement) un processus. Le processus bloqué est dit en attente du signal S. Il est débloqué par l'exécution de S.send. Plusieurs processus peuvent être en attente du même signal : l'exécution de s.send n'en débloque qu'un seul, à savoir le premier s'étant mis en attente du signal (la liste d'attente est gérée en queue).

L'application de cet outil à notre problème, nous a conduit à introduire deux signaux :

- Le signal nonplein, qui transporte la relation  $nb\_pal < n$  (où  $nb\_pal$  indique le nombre de palettes contenues dans la file d'attente, et  $n$  la capacité de la file d'attente).



- Le signal nonvide, qui transporte la relation  $nb\_pal > 0$ .

Les palettes sont gérées en file d'attente. La fig. 4.6, montre comment utiliser un tableau pour la gestion en queue de ces palettes. La variable *inpt* indique l'emplacement pour l'insertion du prochain numéro de palette. La variable *outpt* indique l'emplacement du prochain numéro de palette à prélever. Les variables *inpt* et *outpt* sont incrémentées modulo la taille du tableau : le successeur de l'emplacement *n* du tableau est l'emplacement 1.

La fig. 4.7. présente la solution à ce problème. Si le producteur (processus1) veut déposer une palette alors que la file d'attente est pleine, il se met en attente du signal non plein (ligne 16, fig. 4.7.). Après avoir déposé la palette, le producteur envoie le signal nonvide (ligne 21), puisque à ce moment la condition  $nb\_pal > 0$  est vraie. De même, si le consommateur (processus 2) veut prélever une palette alors que la file d'attente est vide, il se met en attente du signal nonvide (ligne 26).

Après avoir consommé une palette, le consommateur envoie le signal nonplein (ligne 31), puisque à ce moment  $nb\_pal < n$ .

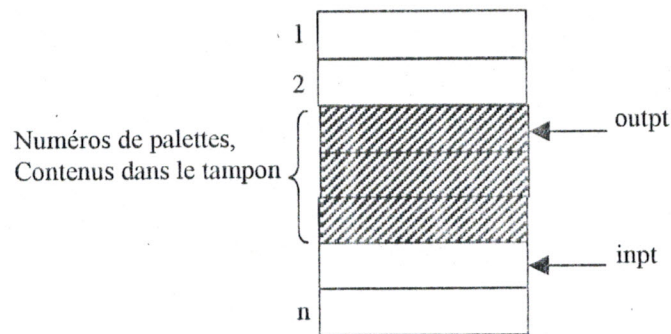


Fig. 4.6. Gestion d'un tampon circulaire.

```

1  Unit tampon ;
2  Defines déposer, prelever ;

3  Const
4      n=100 ; (* taille de la file d'attente *)
5  Var
6      tamp : array[1..n] of Integer ; (*représente un tampon contenant les
                                          les numéros des palettes *)
7      nb_pal : Integer ;
8          (* nombre de palettes contenues dans le tampon *)
9      inpt, outpt : 1.. n ;

10 Signal
11     nonplein ; (* transporte la condition "nb_pal < n" *)
12     nonvide ; (* transporte la condition "nb_pal > 0" *)

13 Procedure déposer (Num_pal : Integer) ;

```

```
14   Begin déposer
15       If nb_pal = n then (* tampon plein*)
16           nonplein.wait ;
17       end If ;
18       tamp[inpt] := Num_pal ; (* dépose le numéro de la palette*)
19       inpt := inpt mod n + 1 ; (* mod : reste de la division *)
20       nb_pal := nb_pal + 1 ;
21       nonvide.send (* puisque nb_pal>0 *)
22   end déposer ;

23   Procedure prelever (Var Num_pal : Integer) ;
24   Begin prelever

25       If nb_pal = 0 then (* tampon vide*)
26           nonvide.wait ;

27       end If ;
28       num_pal := tamp[outpt] ; (* prélève le numéro de la palette *)
29       outpt := outpt mod n + 1 ;
30       nb_pal := nb_pal - 1 ;
31       nonplein.send (* puisque nb_pal<n *)
32   end prelever
33   Initialization (* Initialisation *)
34       nb_pal := 0 ;
35       Inpt := 1 ; outpt := 1
36   End tampon ;
```

Fig. 4.7. Problème de synchronisation résolu à l'aide d'un moniteur.

Les figures fig.4.8, fig.4.9 et fig.4.10 représentent, respectivement, les organigrammes des procédures déposer, prélever et l'activité des processus producteur et consommateur.



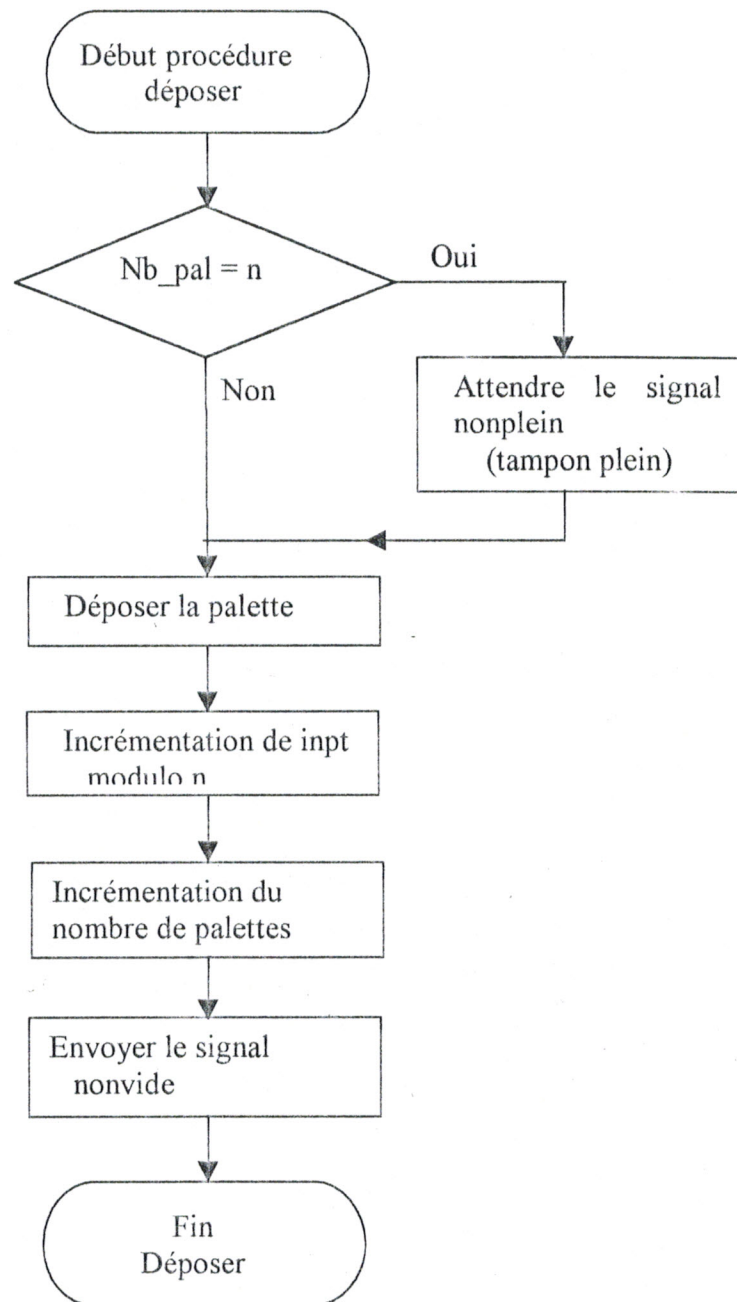


Fig. 4.8. Organigramme de la procédure déposer.

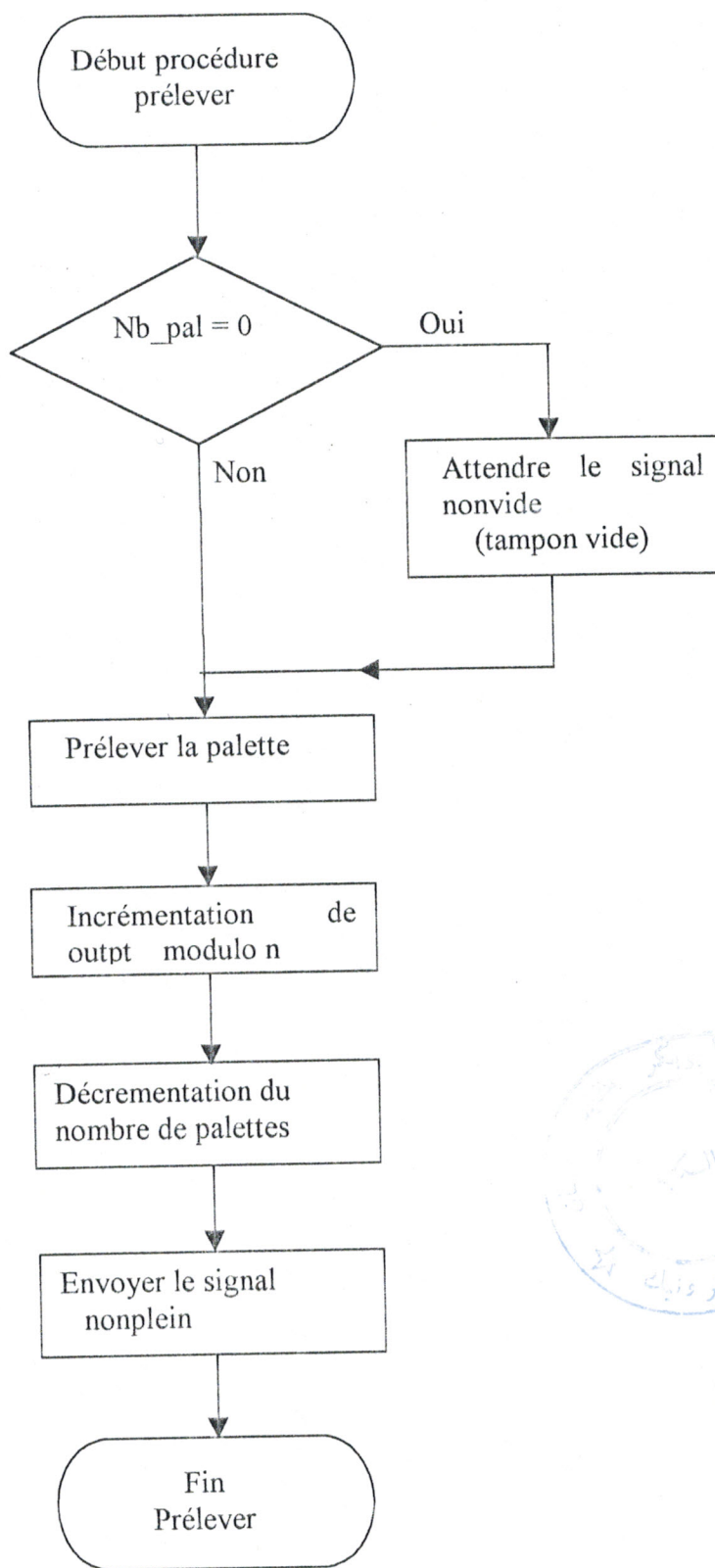


Fig. 4.9. Organigramme de la procédure prélever.



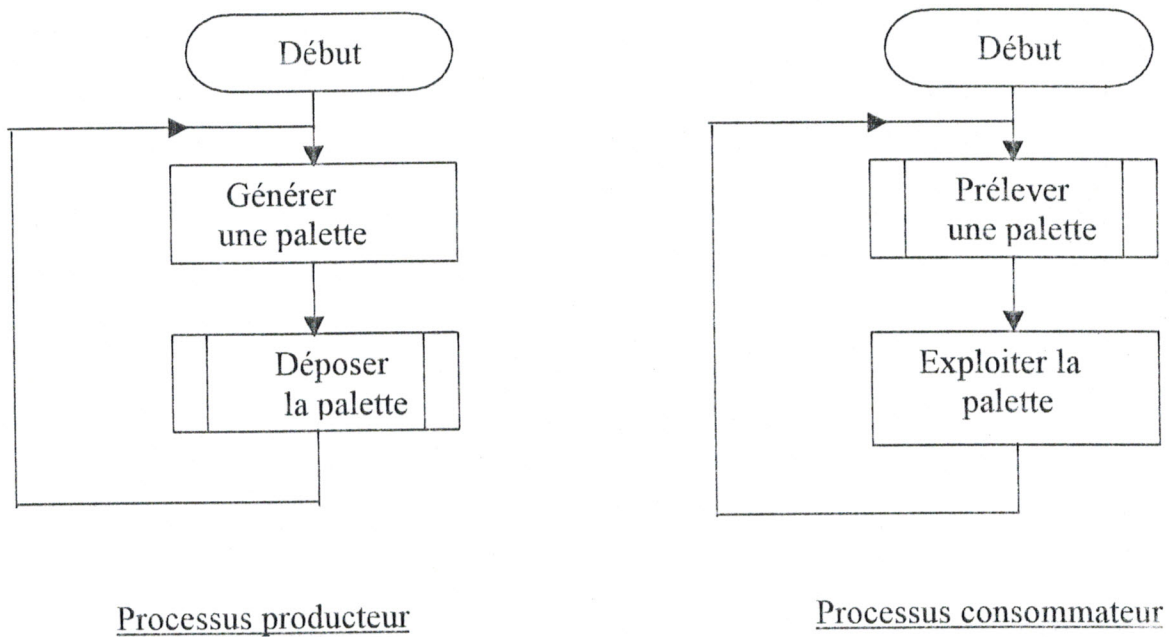


Fig. 4.10. Activité des processus producteur et consommateur.

#### 4.8. Simulation sans animation graphique

L'exécution d'une simulation sans animation est nécessaire, quand on veut accélérer une simulation. Cette fonctionnalité s'avère utile lorsque l'on est sûr que le système fonctionne sans problème et on veut évaluer ses performances sans passer par une animation qui est souvent coûteuse en temps d'exécution. Cela suppose donc, la suppression de la majorité de la partie graphique représentant les constituants de la cellule.

Un raisonnement naturel, conduit à une simulation qui utilise la même structure du programme que celle de la simulation animée, avec une modification qui consiste à diminuer proportionnellement les périodes des différents timer utilisés. Ainsi, si on diminue la période des timer ayant comme période 50ms à 1ms, et ceux ayant comme période 1s à 20ms, on obtiendra une simulation 50 fois plus rapide.

Malheureusement, l'application de cette méthode, n'a pas donné des résultats satisfaisants, du fait que le composant timer offert par delphi, ne permettait pas d'aller à des périodes assez faibles. Par conséquent, cette approche s'est révélé mal adapté à la réalisation de la simulation sans animation.

#### **- Solution proposée**

Le problème à résoudre donc est de pouvoir diviser les bases du temps du système pour accélérer la simulation. L'idée de base de cette solution est de remplacer la structure

des timer, par l'utilisation de boucles de comptage. En effet, les timer de période de 50ms seront remplacés par une procédure Horloge1, qui implémente l'incrémement d'un compteur :

```
Procédure Horloge1;
begin
  Compteur1 := Compteur1 + 1 ; {Compteur1 est une variable entière globale}
end;
```

Pour remplacer les timer de période 1s, la solution consiste à implémenter une procédure Horloge2, qui incrémente un compteur "compteur2", puis divise ce compteur par 20. De cette façon on obtient une fréquence de commutation 20 fois moins, ce qui est demandé.

```
procedure Horloge2 ;
begin
  Compteur2 := Compteur2 + 1; {compteur2, variable entière globale}
  Compteur3 := Compteur2 div 20; { compteur3, variable entière globale}
end;
```

Le programme principal intègre ces deux procédures, dans une boucle de comptage comme suit :

```
Repeat
  Horloge1 ;
  Horloge2 ;
  {code}
Until (condition d'arrêt de simulation = True)
```

Cette approche nous a permis de réduire considérablement le temps de simulation, avec des résultats proches de ceux de la simulation avec animation.

#### 4.9. Evolution de la simulation

Le simulateur permet de suivre l'évolution de l'état d'avancement de la production, ceci est caractérisé par des variables qui représentent les différentes quantités de produits. Une fois qu'une palette termine la fabrication d'un produit, elle doit incrémenter le compteur de produits correspondant au type affecté. Ces compteurs sont structurés dans un tableau de variable entières "Compt\_Prod[Num\_type]". Une première question qui peut être posée est : comment le simulateur prend-il la décision de réutiliser ou de libérer la palette qui vient de terminer la fabrication d'un produit ?

La réponse à cette question passe par l'introduction d'un tableau de variable Type\_satisfait [Num\_type] du type booléen "Boolean", tester la condition de satisfaction (ligne 2), puis d'affecter la valeur vrai (true) à la variable Type\_satisfait [Num\_type] si cette condition est remplie ou la valeur Fausse (False) dans le cas contraire (ligne 3 et 4).



```

1  IF
2  Compt_Prod[Num_type]>=Nbre_prod_type[Num_type]-NbrePaltyp[Num_type]+1)
3  then type_satisfait[Num_type]:=True;
4  else type_satisfait[Num_type]:=False;

```

La variable  $Nbre\_prod\_type[Num\_type]$ , représente le nombre de produit de type  $Num\_type$  demandé. La variable  $NbrePaltyp[Num\_type]$  représente le nombre de palettes affectées au type de produit  $Num\_type$ .

**N.B.** Cette condition correspond au cas ou :

$$Nbre\_prod\_type[Num\_type] \geq NbrePaltyp[Num\_type].$$

Dans le cas contraire, le nombre de palettes utilisé doit être diminué, pour avoir :  $NbrePaltyp[Num\_type] = Nbre\_prod\_type[Num\_type]$ , dans ce cas, la palette qui termine son produit sera systématiquement libérée.

#### a) Démonstration de la condition de satisfaction

Pour démontrer la condition précédente, considérons la première palette notée P1 qui vient de terminer la fabrication d'un produit du type  $Num\_type$ , tel qu'à partir de ce moment le type de produit affecté est considéré comme satisfait. Dans ce cas, tous les autres palettes du même type sont utilisées (car la palette P1, est la première palette susceptible d'être libérée), On note  $N$  ce nombre de palettes (hormis la palette P1), il s'ensuit que :

$$N = NbrePaltyp[Num\_type] - 1$$

La production du type  $Num\_type$  est considérée comme étant satisfaite si pour chaque produit demandé, une palette est affectée. Ainsi, la production sera satisfaite, dès que l'égalité (1) serait vraie.

$$N = Nbre\_prod\_type[Num\_type] - Compt\_Prod[Num\_type] \dots\dots\dots (1)$$

D'où en remplaçant  $N$  par sa valeur :

$$Compt\_Prod[Num\_type] = Nbre\_prod\_type[Num\_type] - NbrePaltyp[Num\_type] + 1$$

Cette égalité présente la condition limite de satisfaction, donc dans le cas général, on aura :

$$Compt\_Prod[Num\_type] \geq Nbre\_prod\_type[Num\_type] - NbrePaltyp[Num\_type] + 1$$

### b) Test d'arrêt de la simulation

La simulation doit être arrêtée, dès que la production de tous les produits sera terminée, Ceci se traduit par la vérification de toutes les égalités :

$$\text{Compteur\_Prod}[\text{Num\_type}] = \text{Nbre\_prod\_type}[\text{Num\_Type}]$$

avec  $\text{Num\_type} = 1.. \text{Nbre\_type}$ , tel que  $\text{Nbre\_type}$  est le nombre de types de produits fabriqués. La fig. 4.11. donne l'organigramme d'une fonction `Detect_fin` de type booléen, qui permet de déterminer la condition d'arrêt de la simulation. Le numéro de type de produit est lu à partir d'un tableau `Tab1`, contenant les numéros des types de produits utilisés, `Dtct` est un tableau de variables booléennes.

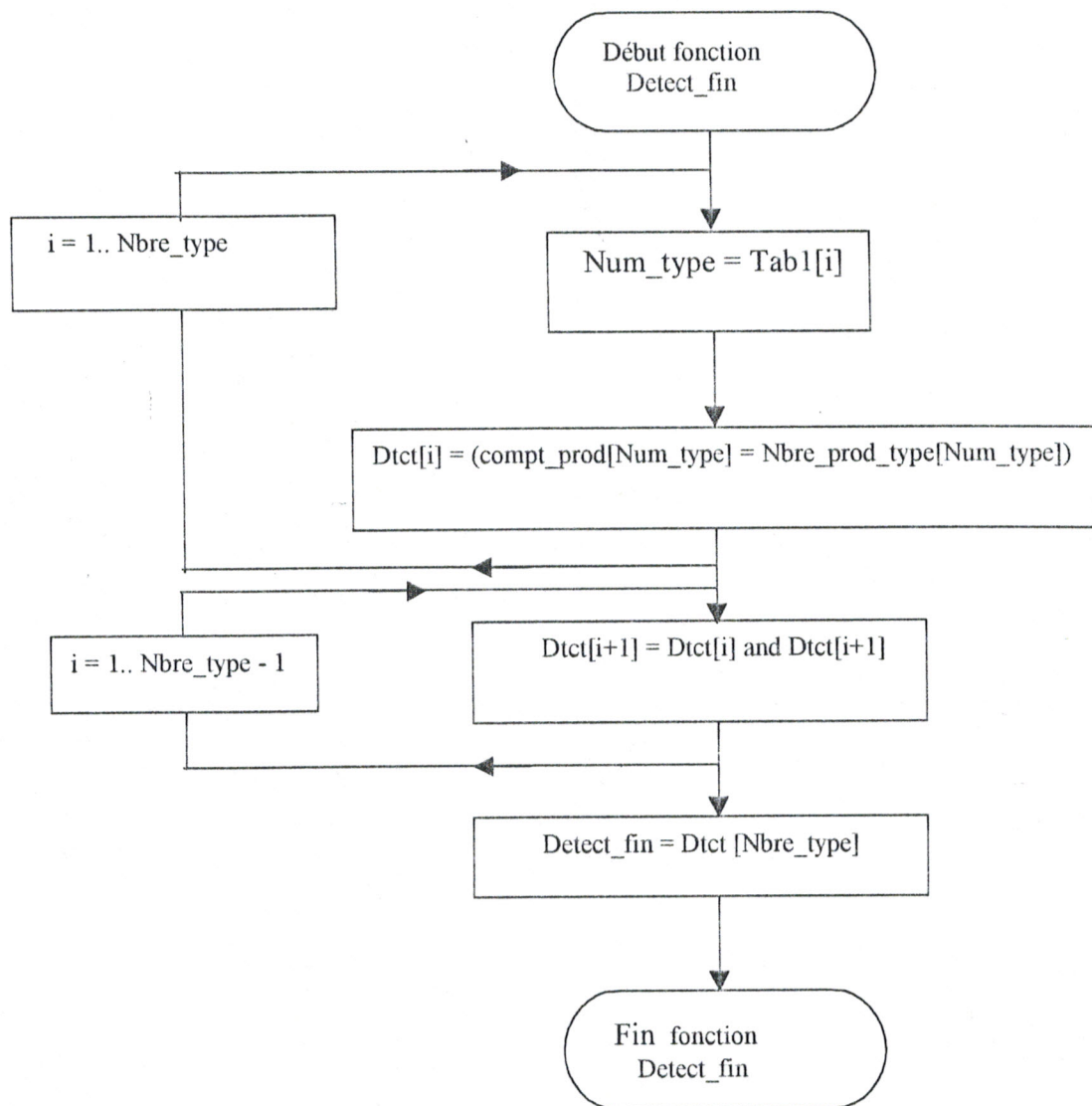


Fig. 4.11. Organigramme de la fonction `Detect_fin`.



### c) Calcul des performances du système

- Taux de production : C'est le nombre de produit sortant du système par unité de temps, il est défini par :

$$\text{Taux de production [Type]} = \frac{\text{Nombre de produits fabriqués[Type]}}{\text{Temps écoulé}}$$

- Temps de cycle : C'est le temps moyen compris entre l'arrivée de la matière en production et la sortie du produit fini, le temps de cycle moyen pour un produit est défini par :

$$\text{Temps de cycle moyen [Type]} = \frac{\sum \text{Temps de cycle pour chaque produit[type]}}{\text{Nombre de produit[type]}}$$

- Taux d'utilisation d'une station d'assemblage : C'est le rapport entre la somme des temps d'exécutions effectifs et le temps écoulé.

$$\text{Taux d'utilisation} = \frac{\sum \text{Temps d'exécutions}}{\text{temps écoulé}}$$

- Taux de charge du convoyeur moyen : C'est la moyenne des taux de charge instantanés, tel que le taux de charge instantané est défini par :

$$\text{Taux de charge instantané} = \frac{\text{Nombre de palettes dans le convoyeur}}{\text{Capacité du convoyeur}}$$

## 5.1. Introduction

Le logiciel que nous avons développé traite de la simulation (avec et sans animation graphique) de la cellule flexible d'assemblage (E.M.N.). Ce logiciel vise à répondre aux objectifs suivants :

- Simuler le fonctionnement de la cellule flexible d'assemblage.
- Offrir à l'utilisateur une grande interactivité avec la simulation, tant au niveau du choix des paramètres, qu'au niveau de l'évaluation des performances du système.
- La possibilité de tester une grande variété de stratégies de contrôles.
- Faire le diagnostic des dysfonctionnements (goulot d'étranglement, arrêts...), et proposer en conséquence des solutions aux problèmes.

Ce logiciel se compose, de plusieurs modules fonctionnels interactifs dont nous allons faire le détail dans ce qui suit.

## 5.2. Menu principal

Dès l'appel au logiciel (Cyber flexi), un menu principal s'affiche, offrant ainsi l'accès à plusieurs menus, comme le montre la fig. 5.1.

Le menu *Introduction*, regroupe deux commandes ; la commande "présentation" qui

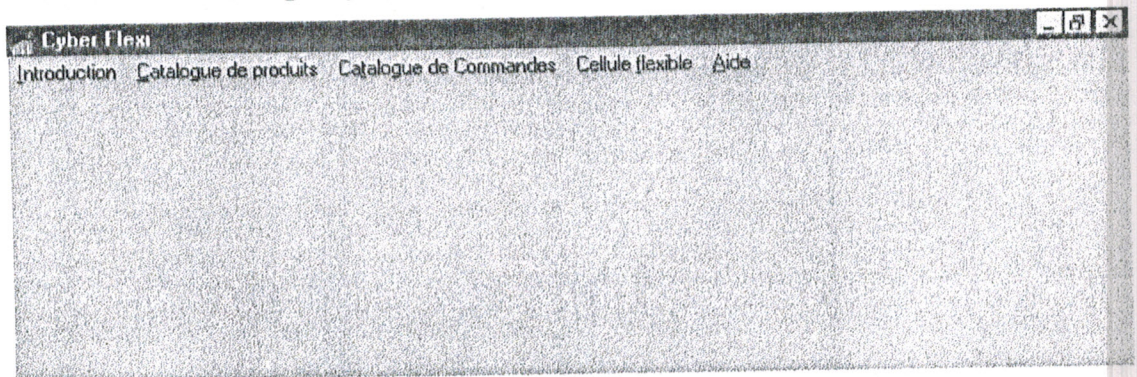


Fig. 5.1. Menu principal.

permet de donner une fenêtre de présentation du logiciel et la commande "Quitter" qui permet de quitter le logiciel.

Le menu *catalogue de produit*, contient une commande Explorer, qui ouvre la BDD catalogue de produit. Le menu *catalogue de commandes*, contient également une commande Explorer, qui ouvre la BDD catalogue de produit.

Le menu *cellule flexible*, regroupe plusieurs sous menus, comme le montre la fig. 5.2. Le menu *Aide* est un menu classique.

Nous allons dans ce qui suit, détailler le fonctionnement de ces différents menus.



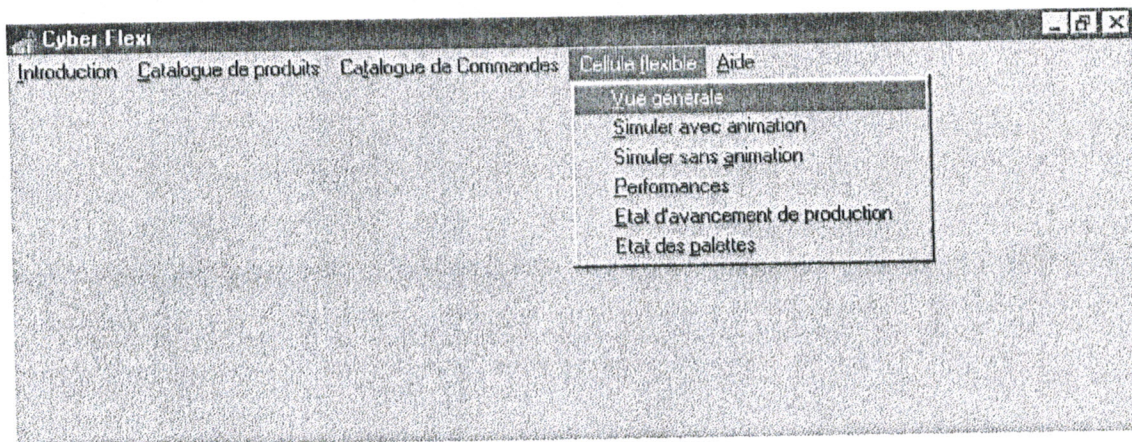


Fig. 5.2. Contenu du menu cellule flexible.

### **5.3. Base de données du système**

Comme nous l'avons dit dans le chapitre précédent, ce logiciel, manipule un volume énorme de données, ces données sont lues par le logiciel pour fournir les simulations appropriées. La conception des bases de données BDD est une solution qui permet de répondre convenablement à nos besoins. Ces BDD regroupent des bases de données d'entrée, c'est à dire qu'elles fournissent des données au système, et des BDD de sortie, c'est à dire de résultats de simulation. Par ailleurs, certaines données relatives à la commande de la cellule (vitesse du convoyeur, fréquence de sortie des palettes,... etc.) sont définis depuis les fenêtres de simulations.

#### **5.3.1. Bases de données d'entrée (données du modèle)**

##### **a) Catalogue de produits**

Ce catalogue est représenté par une base de données (BDD) qui contient les différents produits définis ainsi que leur gamme de fabrication. La fig. 5.3. représente la structure de cette BDD.





Type de Produit	1
Station N°1	1
Station N°2	2
Station N°3	1
Temps 1	7
Temps 2	2
Temps 3	7
Temps TRA	3
Temps SSD	7
Palettes	010203
Priorité	0

Fig. 5.3. Catalogue de produits.

Cette Bdd contient plusieurs champs :

1- type de produit : contient le numéro de type du produit.

2- station N°1, station N°2 et station N°3 contiennent les numéros des stations utilisées pour fabriquer le produit. Cela mérite quelques explications :

Un produit peut parcourir pour sa fabrication, les deux stations d'assemblage SA1 et SA2, avec la possibilité de revenir de nouveau à la première station d'assemblage, ce qui conduit à un routage en trois phases (SA1 SA2 SA1 ou SA2 SA1 SA2) au maximum.

Exemple 1 : Un produit peut passer par la station SA1 pour subir des opérations, puis par la station SA2 et enfin revenir à la station SA1 pour terminer ses opérations.

Quand une station n'est pas utilisée pour la fabrication du produit, on l'affecte la valeur 0.

Exemple 2 : Un produit qui nécessite le passage unique par la station SA1 est décrit par :

Station N°1=1  
 Station N°2=0  
 Station N°3=0

3- Les temps temps1, temps2 et temps3 correspondent aux différents temps d'exécutions sur les différentes stations utilisées.



4- Les temps TRA et temps SSD représentent, respectivement, le temps de transfert et le temps de déchargement- chargement ; ils varient en fonction du type de produit.

5- Le champ palettes désigne les palettes affectées au type de produit correspondant. Les numéros des palettes sont codés sur 2 digits chacune, allant de "01" pour la première palette jusqu'à "00" pour la palette N°100.

6- Le champ priorité affecte un niveau de priorité au type de produit.

Cette base de donnée est munie d'un navigateur de base de données pour faciliter sa manipulation. Ainsi nous pouvons se déplacer d'un enregistrement à un autre, modifier ou supprimer un enregistrement, insérer un nouvel enregistrement... etc. Ce catalogue est accessible depuis la commande *Catalogue de produit/Explorer* du menu principal.

### b) Catalogue de commandes

Ce catalogue est représenté par une BdD qui contient la description de toutes les commandes. La fig. 5.4. représente la structure de cette base de données.

NUMCOM	PROD_TYP1	PROD_TYP2	PROD_TYP3	PROD_TYP4	PROD_TYP5	PROD_TYP6	PROD_TYP7	PROD_L
1	15	0	14	10	10	10	0	
2	20	15	15	0	0	0	0	
3	5	10	15	0	0	0	0	
4	10	10	15	30	30	56	19	
5	26	16	44	0	9	17	0	
6	12	18	20	21	10	16	36	
7	0	62	25	5	0	0	0	
8	44	8	40	50	22	15	61	
9	10	14	60	11	19	46	16	
10	14	0	5	0	41	0	25	
11	0	10	0	5	8	22	0	
12	18	12	15	8	10	0	0	

Fig. 5.4. Structure d'un catalogue de commande  
Composé de 12 commandes.

Nous signalons, que cette BdD est crée dynamiquement en fonction du catalogue de produits, en d'autres termes elle doit afficher un nombre de champs (représentants les

différentes quantités de produits demandés) variable en fonction du nombre de types de produits contenus dans le catalogue de produits. En plus du navigateur de BdD, elle est munie aussi d'un bouton "remplir/valider", qui permet de remplir et valider le catalogue, d'un bouton "tout supprimer" qui permet de supprimer tout le contenu du catalogue. De ce fait et vu les désagréments que peut induire une fausse manœuvre de ce bouton, une action sur ce bouton, affiche d'abord une boîte de dialogue qui propose soit de confirmer la suppression, soit de l'annuler.

Depuis la fenêtre représentant cette BdD, nous pouvons aussi ouvrir un catalogue, détruire un catalogue ou enregistrer un catalogue. Différents menus sont prévus pour cela. Nous pouvons choisir le numéro de la commande active, c'est à dire celle qui doit être exécutée, dans la fig. 5.4. c'est la commande N°1 qui est active.

Cette base de donnée est munie d'un navigateur de base de données pour faciliter sa manipulation. Ce catalogue est accessible depuis la commande *Catalogue de commandes/Explorer* du menu principal.

### c) Données propres à la commande de la cellule flexible

L'utilisateur peut agir sur la commande de la cellule flexible, par le choix des paramètres suivants, depuis la fenêtre de simulation :

- La vitesse du convoyeur.
- La période de sortie des palettes.
- La vitesse d'aiguillage vers SA1.
- La vitesse d'aiguillage vers SA2.

Les fenêtres de simulation (avec et sans animation) sont accessibles depuis la commande *Cellule flexible/Simuler avec animation* et *Cellule flexible/Simuler sans animation* du menu principal.

### 5.3.2. Bases de données de sortie (résultats)

#### a) Performances

Cette BdD permet de visualiser les grandeurs de performances du système à savoir :

- Le taux de production pour chaque type de produit.
- Le Temps de cycle minimal pour chaque type de produit.
- Le temps de cycle moyen pour chaque type de produit.



- La capacité de production pour chaque type de produit.
- Le taux d'utilisation de la station d'assemblage SA1.
- Le taux d'utilisation de la station d'assemblage SA2.
- Le taux de charge du convoyeur.

La fig. 5.5. représente la structure de la fenêtre représentant cette Bdd.

Fig. 5.5. Structure de la Bdd de performances.

En plus du navigateur, cette Bdd est munie d'un bouton *Visualiser*, qui permet d'afficher les résultats, en cours de simulation (On peut quitter momentanément la simulation animée et consulter cette Bdd), ce qui donne la possibilité de suivre l'évolution des performances du système en fonction du temps. Cette fenêtre est accessible depuis la commande *Cellule flexible/Performances* du menu principal.

#### b) Etat d'avancement de la production

Cette Bdd regroupe la quantité de produits demandée, la quantité de produits fabriquée et le pourcentage de production pour chaque type de produit. Elle est structurée selon la fig. 5.6.

En plus du navigateur, cette BdD est munie d'un bouton *Résultats*, qui permet d'afficher les résultats de productions, en cours de simulation, ce qui donne la possibilité de suivre l'évolution de la production du système en fonction du temps, d'un champ *temps correspondant* qui affiche le temps correspondant aux résultats affichés. Cette fenêtre est accessible depuis la commande *Cellule flexible/Etat d'avancement de production* du menu principal.

Type de produit	Quantité demandée	Quantité fabriquée	Pourcentage
1	12	6	50
2	10	8	80
3	15	15	100

Temps correspondant 415

Résultats

Fermer

Fig. 5.6. Structure de la BdD représentant l'état d'avancement de la production avec une commande comportant trois types de produits.

### c) Etat des palettes

Cette BdD affiche les stations parcourues (opérations subies) et les stations restantes (opérations restantes) pour chaque palette, et par conséquent l'état de la palette qui peut prendre les valeurs suivantes (Fig.5.7.a et Fig. 5.7.b.) :



Numéro de palette	Opérations subies	Opérations restantes	Etat
1	SA1	SA2 SA1	En cours
2	Aucune	SA1 SA2 SA1	En cours
3	Aucune	SA1 SA2 SA1	En cours
4	Aucune	SA1 SA2	En cours
5	Aucune	SA1 SA2	En cours
6	Aucune	SA1 SA2	En cours
7	SA2 SA1 SA2	Aucune	Terminé
8	Aucune	SA2 SA1 SA2	En cours

Buttons: Résultats, Fermer

Fig. 5.7.a. La BdD Etats de palettes

Etat de la palette	Signification
En cours	C'est à dire qu'elle est en cours de production et le produit affecté n'est pas terminé.
Terminé	C'est à dire que le produit affecté est terminé.
En attente	C'est à dire qu'elle est en attente (dans SSD) d'être introduite dans la cellule.
Vide	La palette n'est plus utilisée.

Fig. 5.7.b. Les états possibles d'une palette.

**Remarque :** Ces BdD de résultats sont mis à jour à chaque consultation de la BdD, en fonction de l'évolution de la production.

## 5.4. Simulation

### 5.4.1. Simulation avec animation

Cette simulation est visualisée par une fenêtre, accessible à partir de la commande du menu, *Cellule flexible/Simuler avec animation*. Elle fournit une interface graphique facilitant le dialogue utilisateur- simulation. Cette fenêtre est commandée par un module "FMS" qui reçoit un grand nombre de données des BdD suscitées (catalogue de produit et catalogue de commandes). Les autres données sont directement entrées par le biais de



plusieurs boîtes d'édition et boîtes d'édition défilantes ("Spinedit"). Comme on peut le voir sur la fig. 5.8, cette fenêtre dispose aussi d'un panneau de contrôle qui rassemble :

- Un bouton de commande marche/arrêt de la cellule.
- Un "spinedit" qui reçoit la vitesse du convoyeur.
- Une boîte d'édition qui affiche le nombre de palettes utilisées.
- Une boîte d'édition qui affiche le nombre de palettes en attente dans la station SSD.
- Un bouton d'initialisation, qui sert à initialiser les données de la simulation.

Nous avons prévu aussi, un menu "*mis à jour*", qui permet de mettre à jour des données modifiées en amont, en d'autres termes, nous pouvons faire des modifications dans le catalogue de commande, en augmentant la quantité demandée d'un produit et/ou ajouter la fabrication d'un nouveau type de produit.

Pour pouvoir évaluer l'impact d'un arrêt d'une station ; soit un arrêt volontaire pour une opération maintenance préventive ou un arrêt induit par une panne, nous avons prévu deux menus structurés comme suit (fig. 5.9.).

L'arrêt unique de SA1 ou SA2 est activé, respectivement, par les commandes *arrêt SA1* et *arrêt SA2*. Un arrêt associé, c'est à dire que les deux stations sont en arrêt est activé par la commande *arrêt associé (SA1, SA2)*.

Pour rétablir le fonctionnement des stations en arrêt, le menu *Rétablir* fonctionne d'une manière identique à celle du menu *Arrêt*. L'arrêt d'une station (panne par exemple) est modélisé en rendant celle ci indisponible pendant la totalité de la durée de la panne.



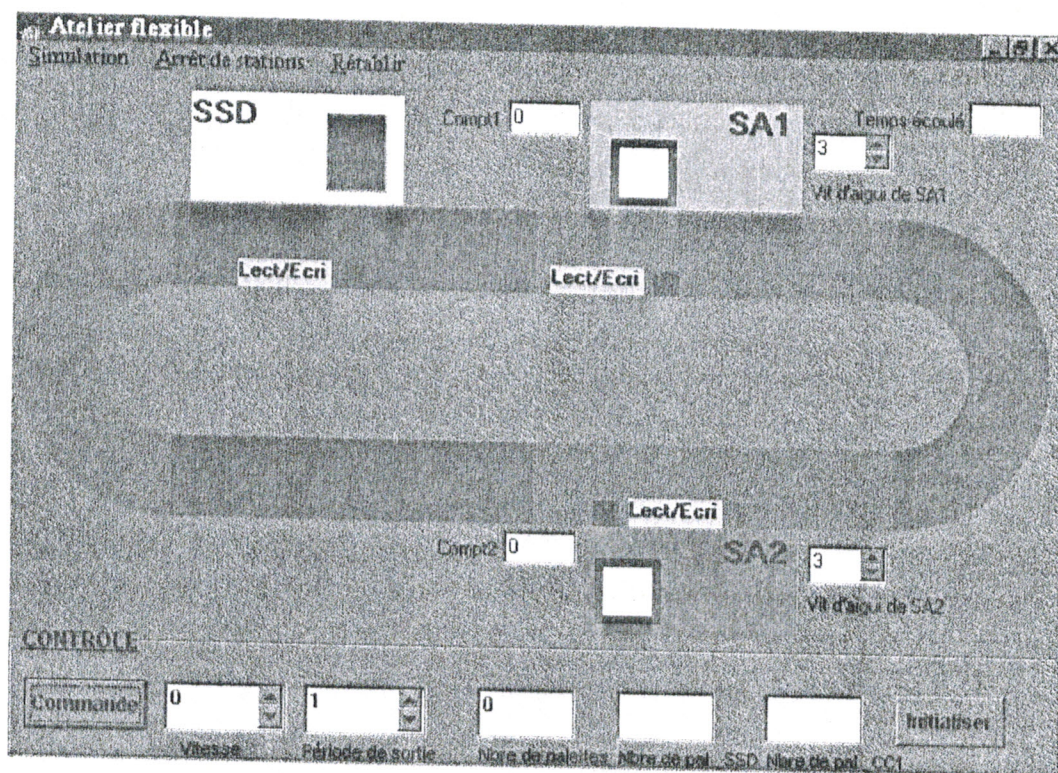


Fig. 5.8. Fenêtre de la simulation animée.

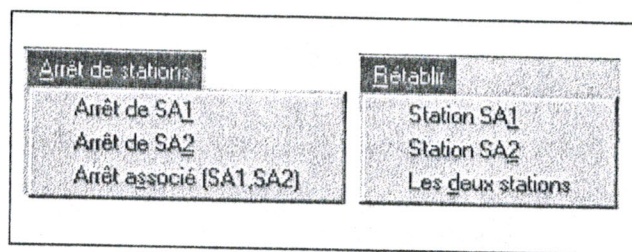


Fig. 5.9. Structure des menus Arrêt et Rétablir.

Cette interface graphique permet d'assister l'utilisateur au déroulement de la simulation par :

- La visualisation du déroulement de la simulation : Les différents événements, changements d'état du système et le temps courant de la simulation doivent être visibles pour l'utilisateur.

- Le déclenchement d'effets visuels lors d'événements particuliers (exécution d'une opération, opération de lecture-écriture, achèvement d'une opération...) afin d'attirer l'attention de l'utilisateur et de bien assimiler le déroulement de la simulation.

Exemple : pour montrer qu'une opération a été exécutée sur une palette, un changement de couleur de la palette est opéré, ainsi elle prendra la couleur de la station d'assemblage parcourue.



- La possibilité de terminer à volonté le déroulement de la simulation.

### 5.4.2. Simulation sans animation

Pour pouvoir effectuer une simulation sans passer par une animation graphique, nous avons prévu le menu *simuler sans animation*. C'est une simulation accélérée, l'entrée des données se fait d'une manière identique à celle de la simulation animée. Elle est commandée par un module (unité) appelé "*FMS\_rapide*" (voir fig.5.10.).

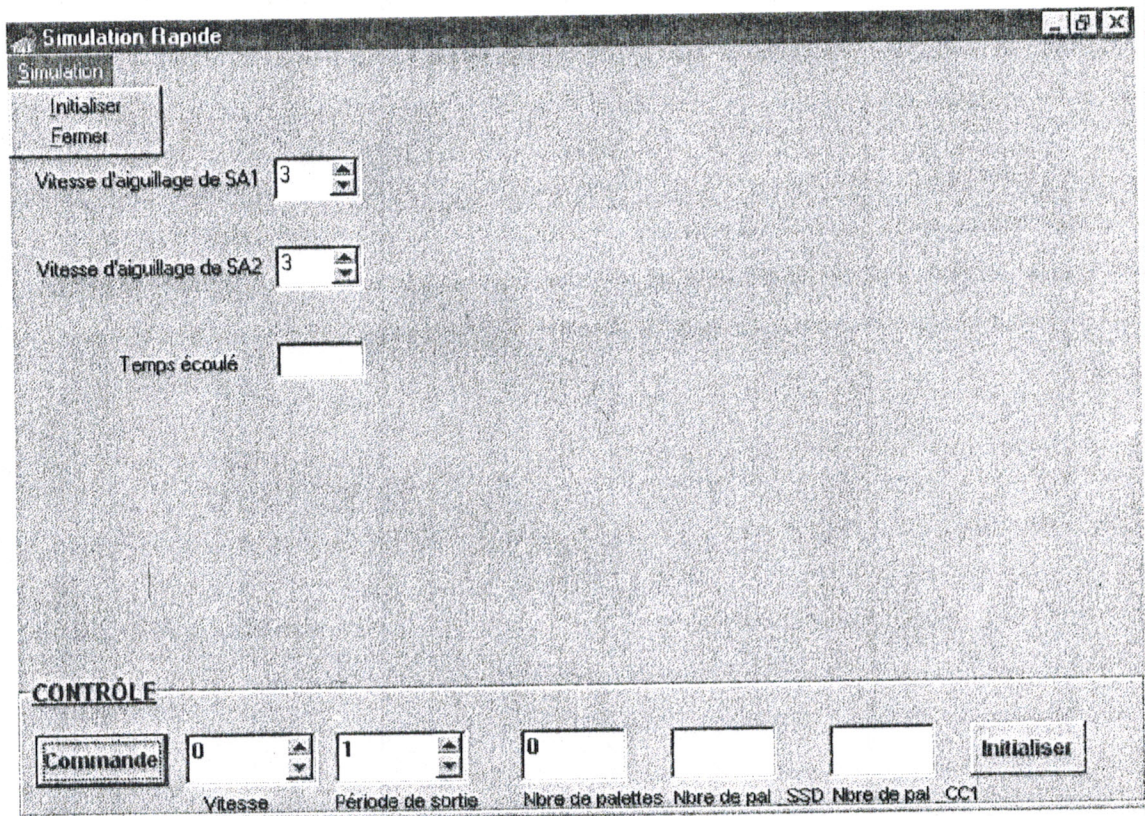


Fig. 5.10. Fenêtre de la simulation sans animation.

## 5.5. Organigrammes

La fig. 5.11. représente l'organigramme du menu principal avec les principaux menus (sans les autres menus classiques : Introduction et Aide). Les organigrammes des figures fig. 5.12 et fig. 5.13, décrivent respectivement les fonctionnements des menus catalogue de commandes et cellule flexible.





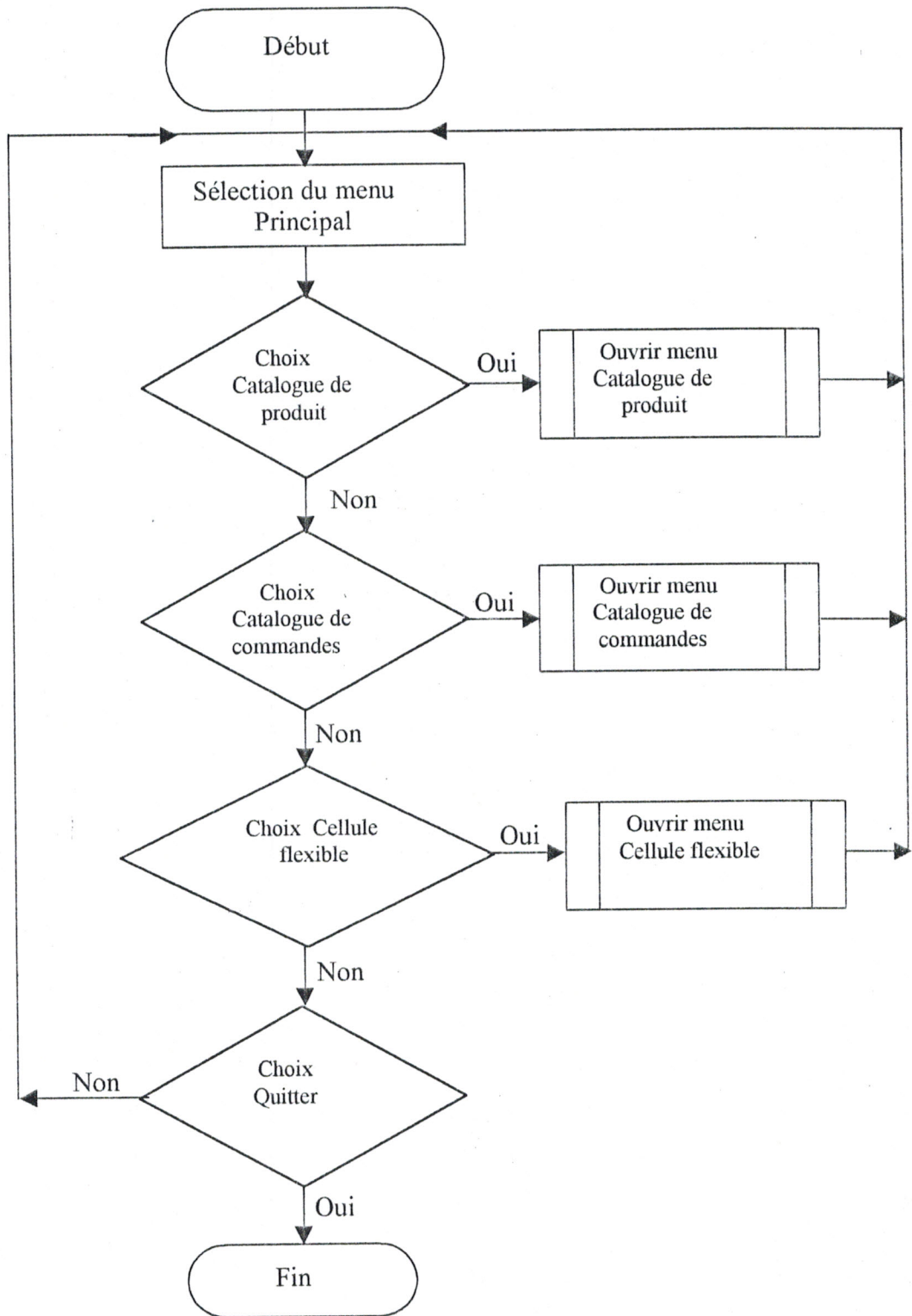


Fig. 5.11. Menu principal

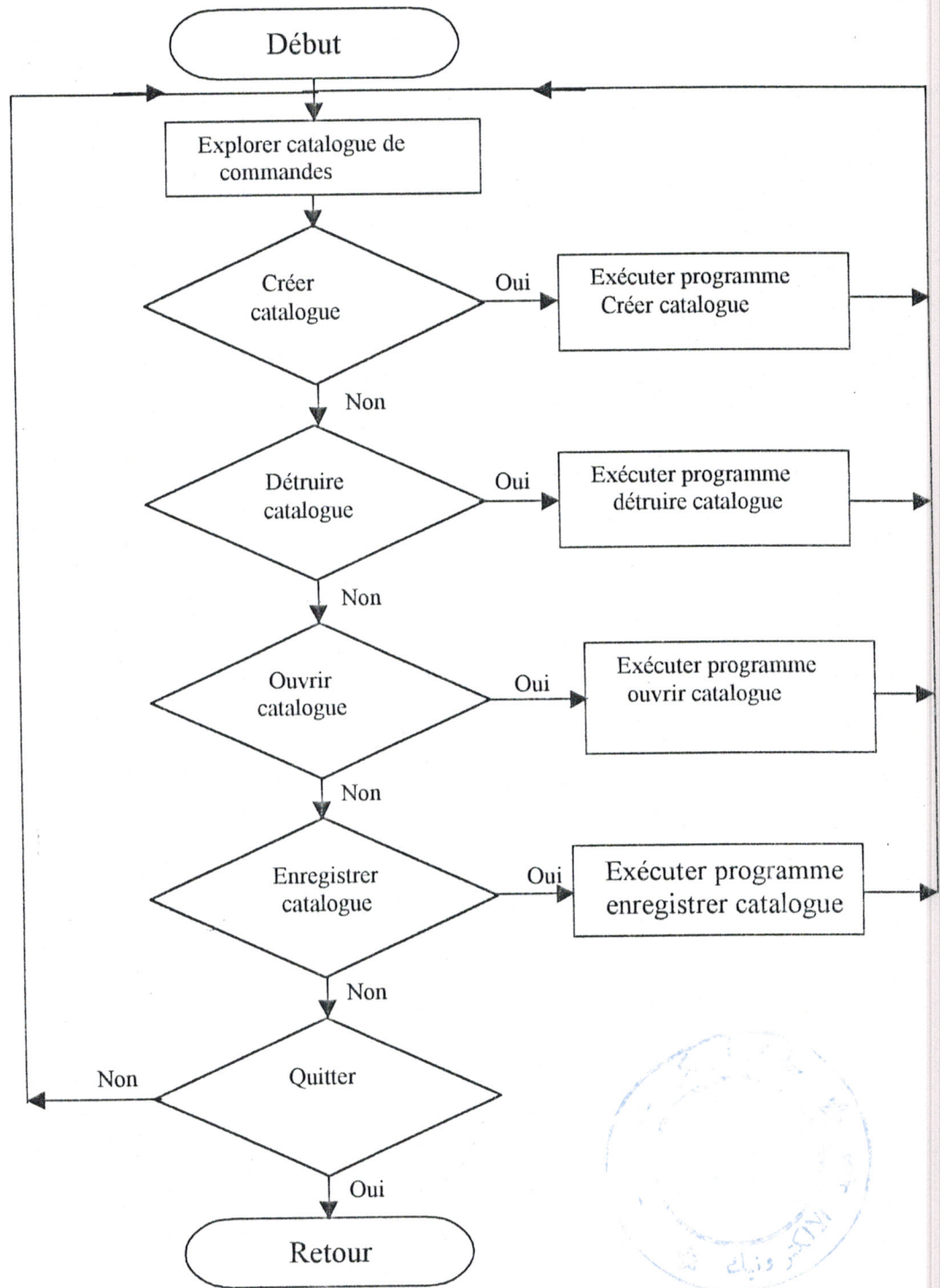


Fig. 5.12. Menu catalogue de commandes.



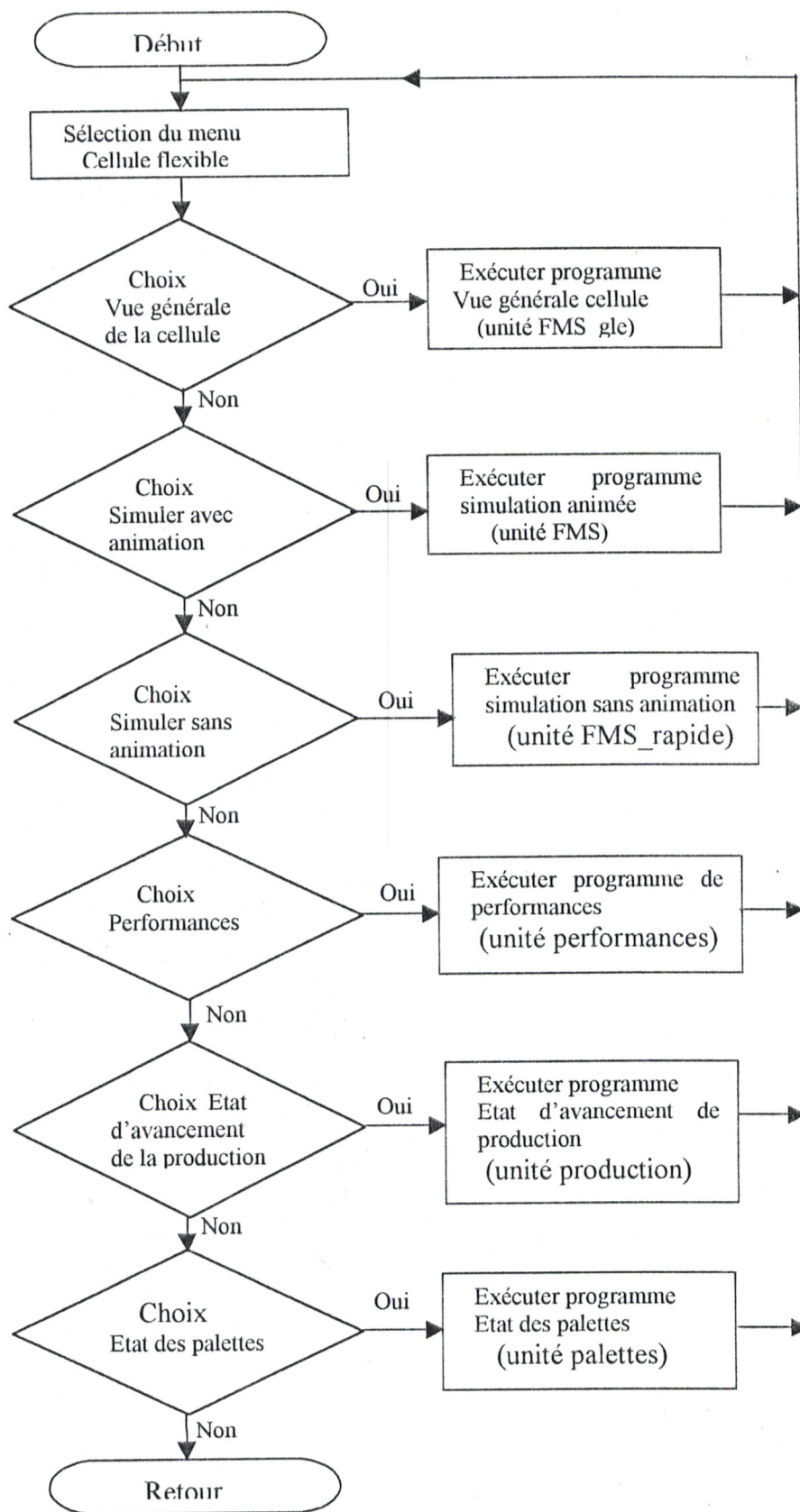


Fig. 5.13. Menu cellule flexible.

## **5.6. Conclusion**

Le logiciel développé est structuré en plusieurs modules (unités), chacun assurant une fonction précise. Bon nombre de menus et de raccourcis clavier, autorisent un accès rapide aux fonctions souhaitées, un grand effort a été effectué pour dégager l'utilisateur de tout souci de manipulation, l'interactivité avec la simulation était au centre des objectifs fixés.



*Chapitre 6*

**Applications du logiciel**

## 6.1. Introduction

Le logiciel développé, offre de nombreuses possibilités pour l'utilisateur. A travers ce chapitre nous allons présenter quelques exemples de simulations qui permettent d'évaluer les performances nécessaires d'un système et par la suite de prendre les décisions qui s'imposent pour améliorer les performances.

Le premier exemple traite un cas de fonctionnement normal, nous allons essayer de modifier certains paramètres et de voir l'influence de ceux ci sur les performances du système. Le deuxième exemple traite une commande plus importante (220 produits). Ensuite, nous présenterons, un cas de fonctionnement anormal (avec goulot d'étranglement). Dans cet exemple, nous allons nous rendre compte qu'un problème peut avoir plusieurs solutions. Le dernier exemple traite une simulation avec arrêt de station, ceci permettra d'évaluer l'impact de cet arrêt sur la production.

## 6.2. Exemples

Nous considérons un catalogue de produits avec 10 types de produits différents. Les gammes de production ainsi que les palettes affectées aux types de produits sont répertoriées comme le montre la fig. 6.1.

Type de produit	Station N°1	Station N°2	Station N°3	Temps1	Temps2	Temps3	Temps transfert	Temps SSD	Palettes	Priorité
1	1	2	1	7	2	7	3	7	010203	0
2	1	2	0	4	3	0	5	3	040506	1
3	2	1	2	10	3	5	5	2	070809 10	2
4	2	1	0	11	5	0	5	3	111213 14	3
5	1	0	0	12	0	0	5	3	1516	4
6	2	0	0	8	0	0	6	3	171819 202122	5
7	2	1	2	8	3	7	2	5	232425 26	0
8	2	1	0	10	5	0	2	5	272829 303132	1
9	2	0	0	6	0	0	5	2	333435 36	2
10	2	1	0	15	7	0	2	5	373839 404142	10

Fig. 6.1. Catalogue de produits.

Notons que l'unité de temps utilisée est la seconde (s) et les vitesses sont données en unités réduites.



### 6.2.1. Exemple 1

Considérons une commande composée de trois types de produits à fabriquer, le tableau suivant résume cette commande (fig. 6.2.) :

N° de commande	Quantité de produit du type 1	Quantité de produit du type 2	Quantité de produit du type 3
1	5	10	15

Fig. 6.2. Quantités de produits de la commande N°1.

En choisissant les paramètres de contrôle suivants :

- Vitesse du convoyeur =  $V_{\text{conv}} = 4$  ;
- Vitesse d'aiguillage SA1 = Vitesse d'aiguillage SA2 = 6 ;
- Fréquence de sortie des palettes = 1 palette/seconde.

L'exécution d'une simulation nous donne les résultats de la fig. 6.3, représentant les performances du système.

Type de produit	Taux de production	Temps de cycle minimal	Temps de cycle moyen	Capacité de production
1	$6 \cdot 10^{-3}$	61	409	$6,25 \cdot 10^{-2}$
2	$1,19 \cdot 10^{-2}$	34	240	$1,66 \cdot 10^{-1}$
3	$1,79 \cdot 10^{-2}$	65	135	$5,88 \cdot 10^{-2}$

Le temps mis par la production	840
Le taux d'utilisation de SA1	$1,85 \cdot 10^{-1} = 18,5\%$
Le taux d'utilisation de SA2	$3,15 \cdot 10^{-1} = 31,5\%$
Le taux de charge du convoyeur	$3,73 \cdot 10^{-1} = 37,3\%$

Fig. 6.3. Performance du système (avec  $V_{\text{conv}} = 4$ ).

En changeant la vitesse du convoyeur à  $V_{\text{conv}} = 5$ , les performances changent selon la fig. 6.4.

Type de produit	Taux de production	Temps de cycle minimal	Temps de cycle moyen	Capacité de production
1	$7,64 \cdot 10^{-3}$	53	194	$6,25 \cdot 10^{-2}$
2	$1,53 \cdot 10^{-2}$	30	126	$1,66 \cdot 10^{-1}$
3	$2,29 \cdot 10^{-2}$	57	159	$5,88 \cdot 10^{-2}$

Le temps mis par la production	654
Le taux d'utilisation de SA1	$2,37 \cdot 10^{-1} = 23,7\%$
Le taux d'utilisation de SA2	$4,05 \cdot 10^{-1} = 40,5\%$
Le taux de charge du convoyeur	$3,33 \cdot 10^{-1} = 33,3\%$

Fig. 6.4. Performance du système (avec  $V_{\text{conv}} = 5$ ).

### - Evaluations

Comme nous pouvons le constater par la comparaison des résultats des tableaux de la fig. 6.3. avec ceux de la fig. 6.4, nous obtenons dans le deuxième cas des taux de production meilleurs (augmentation de 127%). De même, les temps de cycles moyens sont améliorés ; le temps de cycle moyen des produits de type 1 est diminué d'un rapport de 52%, celui du type 2 d'un rapport de 47% et celui du type 3 d'un rapport de 18%.

Les taux d'utilisations des deux stations SA1 et SA2 ont augmenté, respectivement, de 5,2% et 9%. Enfin, la durée de la production a diminué de 22%.

A travers cet exemple, nous pouvons mettre en évidence l'influence d'un paramètre, en l'occurrence la vitesse du convoyeur (l'augmentation de la vitesse a permis d'accélérer le processus de production) sur les performances globales du système.

### 6.2.2. Exemple 2

Considérons une commande composée de 220 produits (fig. 6.5), le nombre de palettes utilisées est de 17 palettes.

N° de commande	Quantité de produit du type 1	Quantité de produit du type 4	Quantité de produit du type 6	Quantité de produit du type 9
2	50	50	50	70

Fig. 6.5. Quantités de produits de la commande N°2.

En choisissant les paramètres de contrôle suivants :

- Vitesse du convoyeur =  $V_{\text{conv}} = 6$  ;
- Vitesse d'aiguillage SA1 = 8 ;
- Vitesse d'aiguillage SA2 = 7 ;





- Fréquence de sortie des palettes = 1 palette/seconde.

L'exécution d'une simulation nous donne les résultats de la fig. 6.6, représentant les performances du système.

Type de produit	Taux de production	Temps de cycle minimal	Temps de cycle moyen	Capacité de production
1	$1,72 \cdot 10^{-2}$	47	153	$6,25 \cdot 10^{-2}$
4	$1,72 \cdot 10^{-2}$	48	165	$7,69 \cdot 10^{-2}$
6	$1,72 \cdot 10^{-2}$	29	151	$1 \cdot 10^{-1}$
9	$2,40 \cdot 10^{-2}$	26	129	$1,25 \cdot 10^{-1}$

Le temps mis par la production	2906
Le taux d'utilisation de SA1	$3,27 \cdot 10^{-1} = 32,7\%$
Le taux d'utilisation de SA2	$5,06 \cdot 10^{-1} = 50,6\%$
Le taux de charge du convoyeur	$5,46 \cdot 10^{-1} = 54,6\%$

Fig. 6.6. Performances du système.

### - Evaluations

En comparant les taux de production avec les capacités de production (la capacité est la valeur maximale que peut prendre le taux de production) de chaque type de produit, on obtient des rendements allant de 19,2% pour le type 9 à 27,52% pour le type 1.

La comparaison entre les temps de cycles minimums et les temps de cycles moyens pour chaque type de produit, montre que :

- la fabrication d'un produit de type 1 reste en moyenne 3,25 fois le temps de cycle minimal (qui correspond à des temps d'attente nulles).
- la fabrication d'un produit de type 4 reste en moyenne 3,43 fois le temps de cycle minimal.
- la fabrication d'un produit de type 6 reste en moyenne 5,20 fois le temps de cycle minimal.
- la fabrication d'un produit de type 9 reste en moyenne 4,96 fois le temps de cycle minimal.

Cela est expliqué par le fait qu'une palette attend dans la plus part des cas, la libération de sa station de destination, ce qui augmente le temps d'attente et par conséquent le temps de cycle du produit.

### 6.2.3. Exemple 3

Dans cette application, nous allons exposer un problème très rencontré dans les systèmes flexibles de production ; c'est le problème des goulots d'étranglement "Bottleneck".

Considérons une commande constituée de 6 types de produits comme le montre le tableau de la fig. 6.7.

N° de commande	Quantité de produit du type 1	Quantité de produit du type 2	Quantité de produit du type 3	Quantité de produit du type 4	Quantité de produit du type 5	Quantité de produit du type 10
3	18	12	15	8	10	14

Fig. 6.7. Quantités de produits de la commande N°3.

En choisissant les paramètres de contrôle suivants :

- Vitesse du convoyeur =  $V_{conv} = 3$  ;
- Vitesse d'aiguillage SA1 = Vitesse d'aiguillage SA2 = 10 ;
- Fréquence de sortie des palettes = 1 palette/seconde.

L'exécution de la simulation, met en évidence un goulot d'étranglement qui bloque le système à partir de l'instant  $t = 343$  s.

L'état d'avancement de la production se bloque à l'état suivant (fig. 6.8.) :

Type de produit	Quantité demandée	Quantité fabriquée	Pourcentage
1	18	0	0%
2	12	0	0%
3	15	3	20%
4	8	2	25%
5	10	3	30%
10	14	2	14,29%

Fig. 6.8. Etat d'avancement de la production.

Le taux de charge du convoyeur instantané (à ce moment là) est égal à 1, c'est à dire plein à 100%. Le problème des goulots d'étranglement est un problème qu'il faut empêcher de se produire, parce qu'il conduit à un blocage du système de production et par conséquent, cause des pertes considérables.



## - Solutions

Comme nous allons le constater, ce problème peut avoir plusieurs solutions proposées.

### a) Solution 1

Comme ce problème est dû à une congestion du convoyeur, nous pouvons par exemple, augmenter la période de sortie des palettes, et ceci pour empêcher la saturation du convoyeur et par la suite le goulot d'étranglement.

En choisissant une période de sortie égale à 2 secondes (C'est à dire une fréquence de 1 palette/2 secondes), nous obtenons les résultats suivants (fig. 6.9.) :

Type de produit	Taux de production	Temps de cycle minimal	Temps de cycle moyen	Capacité de production
1	$1,03 \cdot 10^{-2}$	73	258	$6,25 \cdot 10^{-2}$
2	$6,85 \cdot 10^{-3}$	39	326	$1,16 \cdot 10^{-1}$
3	$8,56 \cdot 10^{-3}$	77	274	$5,88 \cdot 10^{-2}$
4	$4,56 \cdot 10^{-3}$	74	383	$7,69 \cdot 10^{-2}$
5	$5,70 \cdot 10^{-3}$	44	206	$7,14 \cdot 10^{-2}$
10	$8 \cdot 10^{-3}$	77	301	$5,88 \cdot 10^{-2}$

Le temps mis par la production	1753
Le taux d'utilisation de SA1	$3,44 \cdot 10^{-1} = 34,4\%$
Le taux d'utilisation de SA2	$3,39 \cdot 10^{-1} = 33,9\%$
Le taux de charge du convoyeur	$6,09 \cdot 10^{-1} = 60,9\%$

Fig. 6.9. Performances du système.

Si nous augmentons encore la période de sortie des palettes (diminuer la fréquence de sortie) à 1 palette/3 secondes, les performances seront améliorées, comme le montre la fig. 6.10.

Type de produit	Taux de production	Temps de cycle minimal	Temps de cycle moyen	Capacité de production
1	$1,16 \cdot 10^{-2}$	73	214	$6,25 \cdot 10^{-2}$
2	$7,74 \cdot 10^{-3}$	39	253	$1,16 \cdot 10^{-1}$
3	$9,67 \cdot 10^{-3}$	77	282	$5,88 \cdot 10^{-2}$
4	$5,16 \cdot 10^{-3}$	74	217	$7,69 \cdot 10^{-2}$
5	$6,45 \cdot 10^{-3}$	44	72	$7,14 \cdot 10^{-2}$
10	$9,03 \cdot 10^{-3}$	77	187	$5,88 \cdot 10^{-2}$

Le temps mis par la production	1551
Le taux d'utilisation de SA1	$3,89 \cdot 10^{-1} = 38,9\%$
Le taux d'utilisation de SA2	$3,84 \cdot 10^{-1} = 38,4\%$
Le taux de charge du convoyeur	$5,03 \cdot 10^{-1} = 50,3\%$

Fig. 6.10. Performances du système.

**b) Solution 2**

Si nous gardons une fréquence de sortie de 1 palette/3 secondes et nous augmentons la vitesse du convoyeur à  $V_{\text{conv}} = 5$ , nous obtenons des résultats meilleurs, notamment la réduction de la durée globale de production (voir fig. 6.11.).

Type de produit	Taux de production	Temps de cycle minimal	Temps de cycle moyen	Capacité de production
1	$1,50 \cdot 10^{-2}$	52	133	$6,25 \cdot 10^{-2}$
2	$1 \cdot 10^{-2}$	29	122	$1,16 \cdot 10^{-1}$
3	$1,25 \cdot 10^{-2}$	56	196	$5,88 \cdot 10^{-2}$
4	$6,65 \cdot 10^{-3}$	53	289	$7,69 \cdot 10^{-2}$
5	$8,31 \cdot 10^{-3}$	34	55	$7,14 \cdot 10^{-2}$
10	$1,16 \cdot 10^{-2}$	56	136	$5,88 \cdot 10^{-2}$

Le temps mis par la production	1203
Le taux d'utilisation de SA1	$5,01 \cdot 10^{-1} = 50,1\%$
Le taux d'utilisation de SA2	$4,95 \cdot 10^{-1} = 49,5\%$
Le taux de charge du convoyeur	$4,57 \cdot 10^{-1} = 45,7\%$

Fig. 6.11. Performances du système.

**6.2.4. Exemple 4**

Dans cette application nous allons examiner l'impact de l'arrêt d'une station sur les performances globales du système.

Reprenons l'exemple de la première application avec les mêmes paramètres de contrôle (le cas de  $V_{\text{conv}} = 5$ ), et essayons cette fois-ci d'induire une panne (arrêt) de la station SA1, de durée de 40 s, survenant à l'instant  $t = 57$  s. Après simulation nous avons obtenu les résultats suivants (Fig. 6.12.):



Type de produit	Taux de production	Temps de cycle minimal	Temps de cycle moyen	Capacité de production
1	$6,60 \cdot 10^{-3}$	53	226	$6,25 \cdot 10^{-2}$
2	$1,32 \cdot 10^{-2}$	30	217	$1,66 \cdot 10^{-1}$
3	$1,98 \cdot 10^{-2}$	57	164	$5,88 \cdot 10^{-2}$

Le temps mis par la production	757
Le taux d'utilisation de SA1	$2,05 \cdot 10^{-1} = 20,5\%$
Le taux d'utilisation de SA2	$3,50 \cdot 10^{-1} = 35\%$
Le taux de charge du convoyeur	$3,62 \cdot 10^{-1} = 36,2\%$

Fig. 6. 12. Performances du système.

Nous remarquons, qu'après mise en route de la station en arrêt, l'état d'avancement de la production est rattrapée progressivement. Le retard cumulé par cette panne est estimé à 103 secondes, soit environ 15% du temps nécessaire à la production sans arrêt de la station.

### 6.3. Conclusion

Les exemples suscités, ont permis par le biais de la simulation, d'évaluer les performances du système, en fonction de la variation des différents paramètres.

Le problème de goulot d'étranglement a été présenté, c'est un problème assez pénalisant et qu'il faut le résoudre. Nous nous sommes rendu compte qu'il peut exister plusieurs solutions possibles à ce problème.

Finalement, nous avons présenté un cas d'arrêt de station, et évalué les répercussions de celui-ci sur le fonctionnement de la cellule, il est possible de choisir la date ainsi que la durée de cet arrêt. L'ajout d'un état supplémentaire de palettes ("En défaut"), constitue une perspective intéressante, pour pouvoir également simuler l'impact de défaillances des palettes.



## ***Conclusion***



## Conclusion

Dans le présent travail, nous nous sommes proposés de développer un logiciel de simulation d'une cellule flexible d'assemblage. Le recours à la simulation vise à répondre à des problèmes de complexité et de dimensionnement des systèmes flexibles de production. Pour la réalisation de ce logiciel, nous avons utilisé le système de développement Delphi sous Windows 95, qui offre un environnement de programmation orientée objet, très efficace pour le développement des logiciels modernes.

Par ailleurs, nous avons adapté l'approche de commande supervisée à la conception des automatismes de la cellule flexible, qui vont par la suite contribuer pour l'implémentation des programmes de simulation. Dans cette approche, le système de commande permet de définir le fonctionnement primaire du procédé, tandis que les contraintes liées au fonctionnement de celui-ci sont du ressort de la supervision. L'application de cette approche nous a permis de décomposer le système en plusieurs modules commandés séparément. Cette modularité dans la commande, sera traduite par une modularité équivalente dans le programme de simulation, ce qui apporte une simplification dans l'analyse du problème.

Le Grafcet est un outil qui permet de modéliser de façon concise la partie séquentielle des automatismes du système, le concept de commande supervisée s'en servira avantageusement.

Deux types de simulations, ont été présentées dans cette thèse :

- La simulation avec animation graphique, qui est orientée vers le diagnostic et l'analyse des problèmes, pouvant survenir dans le système.
- La simulation sans animation graphique, qui permet de réduire le temps de simulation et qui est de ce fait, orientée vers l'évaluation des performances du système.

En fait, la comparaison entre les deux types de simulation à travers plusieurs exemples a permis de tirer les remarques suivantes :

- La simulation sans animation permet de donner plus rapidement les résultats, qui sont assez proches de ceux obtenus par simulation avec animation. En effet l'exécution d'une commande de 200 produits, reste près de 53 minutes dans une simulation animée, tandis que la simulation sans animation réduit la durée d'exécution à environ 4 minutes, ce qui correspond à une réduction de 13 fois.
- Dans une simulation animée, plus elle est importante (le nombre de palettes mis en jeu est important  $> 15$ ), plus on constate une diminution de la vitesse de simulation. Cela est induit par l'occupation importante



de la mémoire, causée par la création d'objets graphiques (qui sont très gourmands en ressources mémoire).

Par conséquent, Cela nous conduit à dire que lorsque nous sommes amené à faire une étude par simulation, Il ne faudrait pas utiliser d'emblée une simulation animée pour évaluer les performances du système, elle doit être utilisée après une simulation sans animation pour analyser les problèmes qui pourront survenir.

L'évolution temps réel de la simulation, nous a conduit à introduire la programmation concurrente qui offre un ensemble de techniques et d'algorithmes, permettant de résoudre les problèmes de concurrence et de synchronisation inhérents aux systèmes flexibles de production. Nous avons constaté à travers ce travail à quel point, ces techniques permettaient de surmonter les phénomènes de conflits et de concurrences.

Notons que ce logiciel, permet de simuler le fonctionnement de la cellule flexible, en considérant la gamme d'un produit comme une succession d'opérations (sur les différentes stations), nécessitant chacune un temps d'exécution, le déroulement des opérations élémentaires (assemblage de Duplos) n'est pas explicité. Il serait donc intéressant, d'ajouter une unité qui permet de simuler le fonctionnement interne des stations d'assemblage. Cette unité permettra de lire la liste des codes opérations définissant la gamme du produit et de générer le code correspondant aux déplacements du robot, en tenant compte des spécifications dictées par le type des duplos utilisés ainsi que leurs positionnements (coordonnées, orientation).

Pour le système que nous avons considéré, il s'agit de stations d'assemblage ne nécessitant pas un changement d'outils pour le robot, entre deux opérations successives. De ce fait, le robot est capable d'opérer sur plusieurs types de palettes, sans changement d'outils. Cependant, dans un système flexible de production générale, ce n'est pas toujours le cas. En effet les stations de travail disposent de machines flexibles pouvant effectuer une multitude d'opérations (fraisage, soudure,...) nécessitant un changement important d'outils, qui peut prendre un temps non négligeable, il s'ensuit que le robot ne pourra pas opérer sur deux palettes de types différents, sans consommer un temps indispensable pour le changement d'outils.

L'intégration des temps de changement d'outils pour chaque type de produit, permettra d'étendre la simulation à d'autres systèmes flexibles de production. D'autre part, cela suppose l'introduction des modifications sur les règles de décision car le changement d'outils peut poser des problèmes de flexibilité. Afin d'obtenir la souplesse de production nécessaire, une solution consiste à traiter la production par compagnes successives, où chaque compagne regroupe des produits identiques ou des produits dont les gammes sont proches (ne nécessitant pas un grand changement d'outils).

L'exploitation du logiciel, pour traiter des applications a permis d'une part de montrer l'influence des différents paramètres sur les performances du système et d'autres part de proposer des solutions améliorant ces performances. Enfin, nous pensons qu'il serait intéressant de pouvoir utiliser ce logiciel pour tester différentes stratégies de contrôle et par la suite de faire la synthèse d'un algorithme de pilotage optimale de la cellule.



## **BIBLIOGRAPHIE**

## BIBLIOGRAPHIE

- [Aghetta, 96] L.A. AGHETTA, "Utilisation de la simulation en aide au pilotage : une étude de cas à la SNECMA", RAIRO-APII- Journal Européen des systèmes Automatisés (JESA), Volume 30-n°7, pages 945 à 962, 1996.
- [Askin, 93] R.G. ASKIN, C.R. STANDRIGE, "Modeling and Analysis of Manufacturing Systems", John Wiley and Sons, 1993.
- [Belhadri et Mesbahi, 94] A. BELHADRI, L. MESBAHI, "Modélisation et simulation des systèmes de production par l'approche de SLAM II (MSSP3)", PFE. Institut d'Informatique, USTO (Oran), 1994.
- [Bouteille et al, 97] D. BOUTEILLE, N. BOUTEILLE, S. CHANTREUIL, R. COLLOT, J.P. FRACHET, H. LE GRAS, C. MERLAUD, J. SELOSSE, A. SFAR, "Les automatismes programmables", CEPAUDES-EDITIONS, 2<sup>e</sup> édition, 1997.
- [Breugnot et al, 91] D. BREUGNOT, M. GOURGAND, D. Hill, P. Kellert, "GAME : An Object-oriented Approach to Computer Animation in Flexible Manufacturing System Modelling ". Proceedings of 24<sup>th</sup> annual Simulation Symposium, pages. 217-227, 1991.
- [Bulle, 97] T. BULLE, "Supervision d'un atelier flexible automatisé", Projet industriel à Schneider Electric S.A., Ecole des Mines de Nantes, 1997.
- [Burda et Färber, 96] A. BURDA, G. FÄRBER, "Borland Delphi 2", Micro Application, Ed, 1996.
- [Charbonier, 96] F. CHARBONIER, "Commande supervisée des systèmes à événements discrets", thèse de doctorat de l'Institut National Polytechnique de Grenoble (INPG, FRANCE), 1996.
- [Computer, 84] "Software for Industrial Process Control", Computer, 14, 2, février 1984.
- [Dimirovski et al, 94] G.M. DIMIROVSKI, P.C. RADOJICIC, R.M. MARKOVIC, "Modelling, Control and Animated Simulation of complex processes in Robotised FMS", IECON'94, 20<sup>th</sup> International Conference on Industrial Electronics, Control and Instrumentation, Vol 2, pages 1141-6, 1994.



- [Erickson et al, 87] C. ERICKSON, A. VANDENBERGE, T. MILES, "Simulation, Animation, and shop-Floor Control", In Proceedings of the 1987 Winter Simulation Conference, ED. H. GRANT, W.D. KELTON, A. THESEN, 649-652- Atlanta, GA, 1987.
- [Fdida et Pujolle, 89] S. FDIDA, G.PUJOLLE, "Modèles de systèmes et de Réseaux", tome 1 : Performance, Editions EYROLLES, Paris, 1989.
- [Hoare, 74] C.A.R. HOARE, "Monitors : An operating System Structuring Concept", CACM, 17, 10, pages 549-557, octobre 1974.
- [Kersela, 94] A. KESERLA, B.A. PETERS, "An Analysis of Dual Shuttle Automated Storage/Retrieval Systems", Working paper, Dept. Of Industrial Engineering, Texas and A and M University, 1994.
- [Meyer, 90] M. B. MEYER, "Conception et programmation par objets", Inter Editions, Paris, 1990.
- [Pegden et al, 90] C.D. PEGDEN, R.E. SHANNON, R.P.SADOWSKI, -Systems Modeling Corporation-, "Introduction to Simulation using SIMAN", McGraw- Hill, Inc, 1990.
- [Proth et Xie, 95] J.M. PROTH, X. XIE, "Les réseaux de Petri pour la conception et la gestion des systèmes de production", Editions MASSON, 1995.
- [Robert, 89] T.L. Robert, "Programmation orientée objet en Turbo Pascal", SYBEX, 1989.
- [Sari et al, 98] Z. SARI, H. CHETTO, N. GHOUALI, "Analyse de la Flexibilité des Systèmes Automatisés de Production", Proceedings du CNP'98, 1<sup>er</sup> Colloque National sur la Productique, Tizi-Ouzou, Algérie, pages 41-46, mai 1998.
- [Schiper, 86] A. SCHIPER, "Programmation Concurrente", Presse Polytechniques Romandes, Lausanne, 1986.
- [Schneider, 97a] SCHNEIDER ELECTRIC, "Analyse fonctionnelle de la supervision de la cellule flexible de l'école des Mines de Nantes (FRANCE)", 1997.
- [Schneider, 97b] SCHNEIDER ELECTRIC, "Documentation de la cellule flexible (E.M.N.)", 1997.
- [Spector, 84] A. SPECTOR, "Les Logiciels de Commande de Processus", Pour la Science, 28, pages 62-77, novembre 1984.

- [Stecke et solberg, 81] K.E. STECKE, J.J. SOLBERG, "Loading and Control Procedure for Flexible Manufacturing systems", International Journal of Production Research, No. 19, 481-490, 1981.
- [Tschirhart, 90] D. TSCHIRHART, "Commande en temps réel", DUNOD, Informatique Industriel, 1990.
- [Viswanadham, 92] N. VISWANADHAM, Y. NARAHARI, "Performance Modeling of Automated Manufacturing Systems", Printice Hall, 1992.
- [Vujosevic, 90] R. VUJOSEVIC, "Object-oriented Visual Interactive Simulation", Proceedings of the 1990 Winter Simulation Conference, pages 490-8, 1990.
- [Widmer, 91] M. WIDMER, "Modèles mathématiques pour une gestion efficace des ateliers flexibles", collection META, LAVOISIER TEC DOC, 1991.
- [Wu et Wysk, 89] D.S. WU, R.A. WYSK, "An Application Of Discret Event Simulation to On-Line Control and Scheduling in Flexible Manufacturing", International Journal of Production Research, vol. 27, 1603-1624, 1989.
- [Zeigler, 87] B.P. ZEIGLER, "Hierarchical, Modular Discret Event Modelling in an Object Environment", Simulation 49, 5, pages 219-230, 1987.
- [Zeigler, 90] B.P. ZEIGLER, "Object-oriented Simulation with Hierarchical Modular Models", Academic Press, San Diego, CA, 1990.



## Résumé

Ce travail porte sur le développement d'un logiciel de simulation, pour la conduite d'une cellule flexible d'assemblage.

Le logiciel traite deux types de simulations ; la simulation avec animation graphique et la simulation sans animation graphique. La simulation animée est orientée vers le diagnostic et l'analyse des problèmes, pouvant survenir dans le système. La simulation sans animation permet de réduire le temps de simulation, elle est orientée vers l'évaluation des performances du système.

Nous avons, adapté l'approche de commande supervisée, pour la conception des automatismes de la cellule flexible, en utilisant l'outil Grafcet. Par l'utilisation des grafquets de commandes sous supervision et l'application des techniques de la programmation concurrente, nous avons pu résoudre les problèmes de concurrences et de synchronisation inhérents au système.

L'exploitation du logiciel, pour traiter des applications a permis d'une part de montrer l'influence des différents paramètres sur les performances du systèmes et d'autres part de proposer des solutions améliorant ces performances. Les impacts des goulots d'étranglement ainsi que l'arrêt de stations ont été abordés.

### Mots clés :

Systemes flexibles de production (FMS), Simulation à événements discrets, Programmation orientée Objet (POO), Ordonnancement, programmation concurrente, Simulation animée, commande supervisée, Productique.

