

# THÈSE

présentée à

L'UNIVERSITÉ DES SCIENCES ET TECHNOLOGIES DE  
LILLE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ

en

PRODUCTIQUE,  
AUTOMATIQUE ET INFORMATIQUE INDUSTRIELLE

par

LYES KERMAD  
Ingénieur IDN

CONTRIBUTION À LA SUPERVISION ET À LA GESTION  
DES MODES ET DES CONFIGURATIONS DES SYSTÈMES  
FLEXIBLES DE PRODUCTION MANUFACTURIÈRE

Soutenue le 15 janvier 1996 devant le jury composé de  
Messieurs les Professeurs :

C. ANDRÉ	Rapporteur
J.P. BOUREY	Directeur de travail
M. COURVOISIER	Rapporteur
E. CRAYE	Examineur
J.C. GENTINA	Directeur de Thèse
A.E.K. SAHRAOUI	Examineur
M. STAROSWIECKI	Examineur

Cette thèse a été préparée au Laboratoire d'Automatique et d'Informatique Industrielle de Lille,  
CNRS D1440, de l'Ecole Centrale de Lille.

# **AVANT-PROPOS**

*Le travail présenté dans ce mémoire a été effectué au Laboratoire d'Informatique Industrielle de Lille (L.A.I.L.) sous la direction scientifique de Monsieur GENTINA, Professeur à l'Ecole Centrale de Lille, avec la collaboration de Monsieur BOURRY, Professeur à l'Ecole Centrale de Lille. Je voudrais leur témoigner toute ma reconnaissance pour l'aide tant scientifique que morale qu'ils ont su m'apporter tout au long de ce travail.*

*Je tiens également à remercier :*

- *Monsieur SAHRAOUI, Professeur à l'IUT-B de l'Université Toulouse II,*
- *Monsieur STAROSWIECKI, Professeur à l'Université de Lille I,*
- *Monsieur CRAYE, Professeur à l'Ecole Centrale de Lille,*

*pour l'honneur qu'ils me font en examinant ce travail et en acceptant de participer à mon jury de Thèse.*

*Je suis très reconnaissant à :*

- *Monsieur ANDRÉ, Professeur à l'Université de Nice-Sophia Antipolis,*
- *Monsieur COURVOISIER, Professeur à l'INSA de Toulouse,*

*pour l'honneur qu'ils me font en acceptant d'examiner ce travail et d'être les rapporteurs de cette Thèse.*

*Je voudrais adresser une pensée toute particulière à tous les membres du LAII auxquels je suis reconnaissant de leur bonne humeur et de leur aide précieuse tant sur le plan scientifique que sur le plan humain. Je tiens à ce sujet à remercier tout particulièrement Monsieur CRAYE pour sa disponibilité et ses remarques pertinentes qui m'ont aidées à améliorer la qualité de ce mémoire.*

*Enfin, je remercie très sincèrement Monsieur VANGREVENINGE qui a assuré la reprographie de ce mémoire.*

# SOMMAIRE

Introduction générale .....	9
Partie A .....	15
Partie B .....	61
Partie C .....	117
Conclusion générale .....	165
Annexe .....	169
Références bibliographiques .....	189
Table des matières .....	201
Table des figures .....	207

# **INTRODUCTION GÉNÉRALE**

Au cours des dernières décennies, les besoins de consommation n'ont cessé de se développer et d'évoluer. Les produits demandés sont de plus en plus complexes et variés, alors qu'en même temps leur durée de vie diminue.

Ces évolutions ont conduit les industriels à employer des moyens de production flexibles qui peuvent être facilement adaptés à des fabrications variant dans le temps par l'introduction de nouveaux produits, le changement des objectifs de production ou la déficience du procédé ou de la commande.

Cependant, la mise en place des Systèmes Flexibles de Production Manufacturière (SFPM) nécessite des investissements considérables aussi bien sur le plan matériel que sur le plan de la conception du système de contrôle/commande. En effet, l'étude et la mise au point du système de commande est une opération coûteuse à cause de la grande complexité du fonctionnement des SFPM. La réduction du coût de la commande devient alors un impératif. Cette réduction passe par l'automatisation de la démarche de conception, et la création d'outil de validation et de détection d'erreurs. Plusieurs travaux ont été menés dans ce sens au sein de la communauté de recherche française. Citons à ce titre quelques équipes :

- Le Laboratoire d'Automatique et d'Analyse des Systèmes (LAAS) de Toulouse pour ses travaux qui ont développé une hiérarchie modulaire de commande/surveillance en temps réel [Courvoisier et al, 84 et 89], [Sahraoui, 87, 92 et 93] et [Combacau, 91]. [Bonneval, 93] complète ces travaux en proposant des solutions aux problèmes de reprise et de traitement des urgences.
- Le Laboratoire d'Automatique de Grenoble (LAG) pour ses travaux sur la modélisation hybride du système de commande [Long, 92], [Devapriya, 92] et l'évaluation des performances des SFPM [Bouchouch, 92].
- Le laboratoire I3S (Informatique, Signaux et Systèmes de Sophia Antipolis), pour ses travaux sur une approche mixte synchrone et asynchrone d'un système temps réel. Cette approche permet d'intégrer en partie les aspects réactifs des SFPM [André, 91 et 92], [Fancelli, 91]. Ces travaux sont complétés par une approche de conception et de validation des systèmes réactifs. Il s'agit, plus particulièrement, du développement des applications de contrôle du fonctionnement normal, mais également de surveillance et de chargement des modes de fonctionnement en cas d'erreurs.
- Le Laboratoire de Génie Industriel et Logiciel (LGIL) de l'Université Valenciennes et du Hainaut-Cambrésis pour ses travaux sur un langage de haut niveau pour le contrôle/commande de cellules robotisées. Ces travaux menés dans le cadre du projet KARMEN (MIAT/SERICS n° 9023021) ont notamment abouti à la proposition d'une méthode d'exploitation

des systèmes automatisés de production, le MESAP [Parayre, 92].

Le Laboratoire d'Automatique et d'Informatique Industrielle de Lille (LAIL) développe une méthodologie de Conception Assistée depuis la spécification jusqu'à l'implantation sur site des Systèmes de Production en Industrie Manufacturière (CASPAIM). De nombreux travaux ont déjà été réalisés dans le cadre de ce projet : [Corbeel, 79], [Castelain, 87], [Kapusta, 88], [Bourey, 88, et 93], [Craye, 89 et 94], [Cruette, 91], [Bois, 91], [Toguyeni, 92], [ElKhatabi, 93], [Ausfelder, 94], [Amar, 94], [Huvenoit, 95], [Tawegoum, 95] et dernièrement, [Ohl, 95].

Les travaux présentés dans ce mémoire s'inscrivent dans le cadre du projet CASPAIM. Ils proposent d'y intégrer la conception de la supervision et de la gestion des modes et des configurations des SFPM.

Ces travaux s'inscrivent dans la continuité des travaux de S. Bois [Bois, 91] sur la gestion des modes de marches et d'arrêt des ressources de production. Ils sont en aval des travaux présentés par A.K.A. Toguyeni [Toguyeni, 92] et S. ElKhatabi [ElKhatabi, 93] dans le cadre de la surveillance en ligne et de la surveillance réactive des moyens de production d'un atelier. Ils viennent, en outre, compléter les travaux sur la commande développée à l'aide des réseaux de Petri à travers les notions de gamme logique et de gamme opératoire [Cruette, 91], [Amar, 94].

Les résultats présentés dans ce mémoire concernent la gestion des modes et des configurations des SFPM. Nous avons choisi une approche structuro-fonctionnelle [Sembougamoorthy et al, 86] pour modéliser les systèmes de production. Le modèle COD (Chargement, Opération(s), Déchargement) obtenu permet de spécifier à l'aide d'outils synchrones graphiques la surveillance réactive de l'état de disponibilité des ressources et des fonctions du SFPM. Il permet aussi de spécifier et d'activer en temps réel la configuration nécessaire à la réalisation d'un objectif de production en tenant compte des défaillances du procédé.

Ce mémoire est divisé en trois parties :

- La première partie précisera le domaine d'application de notre démarche à travers la définition des caractéristiques des SFPM. La problématique de la supervision et de la gestion des modes sera précisée à travers une spécification fonctionnelle de la conduite des SFPM. Cette spécification montrera la place de la supervision et de la gestion des modes dans le système de conduite. Cette partie présentera enfin les approches les plus connues de la gestion des modes. Elle insistera notamment sur leurs principes, leurs ouvertures et leurs limites avant de présenter l'état d'avancement de ce sujet au sein de l'équipe PFM (Production Flexible Manufacturière) du LAIL.
- En partant des approches de modélisation des SFPM adoptées dans le projet CASPAIM du LAIL (les modèles de la commande de coordination et les modèles de la surveillance), nous

proposerons dans la deuxième partie une approche fonctionnelle de la modélisation des SFPM. Notre objectif sera ici d'aboutir à une vision des SFPM adaptée à la gestion des modes de fonctionnement et des configurations.

- La représentation COD, résultat de la démarche de la modélisation fonctionnelle, sera mise en œuvre dans la troisième partie pour exploiter les SFPM. Les modèles de la coordination utilisés dans CASPAIM seront traduits en arborescence. Le lien avec la surveillance est illustré à travers l'intégration de l'état des ressources physiques pour la détermination de la disponibilité des fonctions du SFPM. Cette dernière partie aboutit à la spécification de la gestion des modes et des configurations à l'aide d'un langage synchrone graphique. Cet outil est adapté à l'expression des mécanismes essentiels de supervision à la fois par l'intégration du flux ascendant des informations liées à la surveillance et du flux descendant des informations liées à la gestion des configurations des SFPM.

# **PARTIE A : PROBLÉMATIQUE DE LA SUPERVISION ET DE LA GESTION DES MODES**

Chapitre I : Les Systèmes de production : hypothèses de travail

Chapitre II : L'intégration des SFPM automatisés

Chapitre III : Spécification fonctionnelle de la conduite d'un SFPM  
[Kermad et al, 93b]

Chapitre IV : La gestion des modes dans la littérature

## INTRODUCTION

La plupart des travaux qui concernent les Systèmes Flexibles de Production Manufacturière (SFPM) traitent d'aspects partiels de la commande et n'abordent que rarement les problèmes d'intégration. Il existe peu d'approches d'intégration de la conception globale du contrôle/commande des SFPM. Dans la plupart des cas, les solutions présentées concernent une catégorie particulière de systèmes avec des hypothèses de travail très fortes [O'Grady, 92]. La littérature sur la surveillance des SFPM, par exemple, décrit l'intégration de la détection, du diagnostic, du pronostic et du recouvrement d'erreur, i.e., des différentes fonctions de la surveillance sans parler des liens avec les autres fonctions du système de contrôle/commande. En effet, la surveillance n'existe pas indépendamment des autres parties de la commande des SFPM. Elle coopère avec la commande et l'ordonnancement. Ces fonctions travaillent sur les données générées par le SFPM.

L'objectif de cette partie est de poser le problème de la supervision et de la gestion des modes d'un SFPM dont elle donnera la définition et précisera la position dans le système de contrôle commande.

Cette partie présentera d'abord les SFPM. Nous préciserons leurs principales caractéristiques. Nous présenterons notamment leur complexité pour justifier la nécessité d'une approche de conception globale et intégrative de leurs systèmes de contrôle/commande.

Ensuite, nous présenterons une spécification fonctionnelle de la conduite des SFPM. Nous y préciserons la place de la supervision et de la gestion des modes dans le système de conduite. Nous aurons en permanence le souci de l'intégration des différentes fonctions que nous présenterons. Nous montrerons notamment leurs liens et leurs frontières mutuelles. Nous traiterons ces problèmes sans a priori sur les solutions que peut choisir un concepteur pour une fonction particulière. Il est le seul à connaître toutes les contraintes et la finalité d'une fonction et peut en conséquence l'implanter.

Enfin, nous parlerons des approches les plus connues de la gestion des modes. Nous en présenterons les principes, les ouvertures et les limites avant de présenter l'état d'avancement de ce sujet au sein de l'équipe PFM (Production Flexible Manufacturière) du LAIL (Laboratoire d'Automatique et d'Informatique industrielle de Lille) dans le cadre du projet CASPAIM (Conception Assistée des Systèmes de Production Automatisée dans l'Industrie Manufacturière). Nous poserons alors la problématique de la supervision et de la gestion des modes.

## **CHAPITRE I : LES SYSTÈMES DE PRODUCTION : HYPOTHÈSES DE TRAVAIL**

D'une manière générale, l'activité productrice met en jeu de façon interactive des individus appartenant à trois populations [Rodde, 89] :

- la population des produits,
- la population des moyens de production,
- la population des opérateurs de production.

Chaque produit, moyen ou opérateur impliqué dans le système de production peut évoluer indépendamment du système. Mais ces acteurs sont liés par l'objectif à atteindre : produire. Nous présenterons cet objectif de production à travers la gestion des modes de collaboration de ces trois acteurs dans le cadre des Systèmes Flexibles de Production Manufacturière (SFPM).

### **I.1. Définition et typologie des systèmes de production**

Les Systèmes Flexibles de Production Manufacturière (SFPM) sont une catégorie particulière de systèmes de production. Ils font l'objet de nombreuses recherches. Nous allons en présenter les principales caractéristiques et montrer leur place au sein de l'ensemble des Systèmes de Production. Nous présenterons une typologie et une classification de ces systèmes de production à travers la notion de flexibilité.

Nous trouvons dans [Woodward, 65] trois classes principales de production sur la base de la taille des lots de produits fabriqués. [Benassy, 87] les classe de la même manière en mettant l'accent sur les techniques de gestion et d'ordonnancement mises en œuvre dans chaque classe. C'est ainsi que sont définies les productions :

- unitaire qui concerne typiquement l'entreprise artisanale, la construction navale ou les chantiers de construction d'ouvrages importants (barrage, centrale électrique...).
- Petite et moyenne série dont les moyens de production sont des machines polyvalentes sur lesquelles passe une grande diversité de produits.
- Grande série (ou production de masse) qui utilise des machines spécifiques généralement automatisées. Ces machines exécutent une ou plusieurs opérations techniques.

Cette classification selon la taille des lots de produits s'avère insuffisante si on considère les aspects technologiques d'un système de production qui prend davantage en compte les caractéristiques des produits [Rakotoson, 93]. [Froment et al, 84] propose la notion de processus

qu'ils qualifient de continus (pétrochimie, sidérurgie, production d'énergie...) ou de discontinus (mécanique, électronique, fonderie, ...). Cependant la réalité n'est pas aussi tranchée, car un processus peut être partiellement discontinu et partiellement continu. L'exemple le plus souvent mentionné dans la littérature concerne l'industrie agro-alimentaire [Bayart et al, 92]. La production de fromages ou de pots de yaourt comporte une partie continue qui se caractérise par la production de la pâte laitière et de son transport de manière continue. Ensuite la mise en pot par unités élémentaires ainsi que les traitements suivants caractérisent la partie discontinue. On parle alors de processus mixtes et de systèmes hybrides.

[Pourcel, 91] propose aussi une classification des systèmes de production, mais cette fois sur la base de critères relatifs à la demande commerciale (stock ou demande), aux gammes de produits (linéaires, convergentes ou enchevêtrées) et enfin relatifs à la notion de continuité. En effet, pour Pourcel, "un flux est défini comme continu ou discontinu par rapport à la période d'examen du système de décision". Cette définition permet de rattacher le caractère continu-discret attribué à un système de production, non plus au procédé ou produit lui-même, mais au modèle qui est utilisé par la commande.

Nos travaux s'intéressent principalement aux questions liées à la commande et à l'organisation des Systèmes de Production. Ils s'inscrivent dans le cadre du projet CASPAIM (Conception Automatisée des Systèmes de Production Manufacturière) qui adopte pour la commande et la coordination une représentation par modèles à événement discrets du comportement des ressources et des produits. Ces modèles concernent les Systèmes de Production Manufacturière qui sont, de plus, qualifiés de flexible au sens où nous le définissons ci-dessous.

## **I.2. La Flexibilité**

### **I.2.1. Définition**

La flexibilité (ou adaptabilité et souplesse) peut se définir en terme de capacité et de rapidité d'adaptation à une nouvelle situation. Elle est caractérisée, d'une part, par l'étendue et la variété de la demande qu'elle peut satisfaire et, d'autre part, par la facilité et la rapidité de passage d'un état à un autre et d'une configuration à une autre [Ranky, 90].

### **I.2.2. Typologie**

Le contexte actuel de production impose à un système de production de posséder plusieurs types de flexibilités [Sethi et al, 90], [Hutchison, 79]. On trouve dans [Brown et al, 85] huit types de flexibilité :

- **Flexibilité des machines** qui caractérise la capacité d'une machine à changer d'outil et de programme de commande pour passer d'une opération à une autre.

- **Flexibilité de routage** qui caractérise la possibilité d'acheminer un produit d'un point à un autre en suivant des trajets différents. Selon [Shin, 1989], cette flexibilité est :
  - Potentielle si les trajets sont précalculés en régime normal et redéfinis en cas de dysfonctionnement et de dégradation.
  - Active si les transferts sont décidés dynamiquement en fonction de l'état courant du système.
- **Flexibilité de production** qui caractérise l'étendue des gammes de produits différents que peut réaliser le système.
- **Flexibilité de produit** qui exprime la possibilité de fabriquer un nouveau produit.
- **Flexibilité d'ordre des opérations** qui exprime la possibilité de changer l'ordre des opérations qui constituent la gamme d'un produit.
- **Flexibilité d'expansion** qui exprime la possibilité d'étendre la configuration du système et d'élargir ses fonctionnalités de façon modulaire.
- **Flexibilité de volume** qui caractérise la capacité du système à adapter sa cadence de production aux exigences de l'environnement.
- **Flexibilité de mélange** qui exprime la capacité du système à fabriquer simultanément des produits différents avec des flux variables.

### I.2.3. Mesure de la flexibilité

La mesure de la flexibilité est un problème difficile eu égard à sa grande diversité. En effet, si la description qualitative de la flexibilité a été traitée par de nombreux auteurs, les aspects quantitatifs n'ont souvent été qu'effleurés. Pour [Ranky, 1990] la difficulté de mesurer la flexibilité est due à la nécessité de prendre en compte trois paramètres : l'étendue et la variété des produits à fabriquer, le temps et la durée de la reconfiguration du système de production ainsi que son coût, tant à l'exécution qu'à l'implantation.

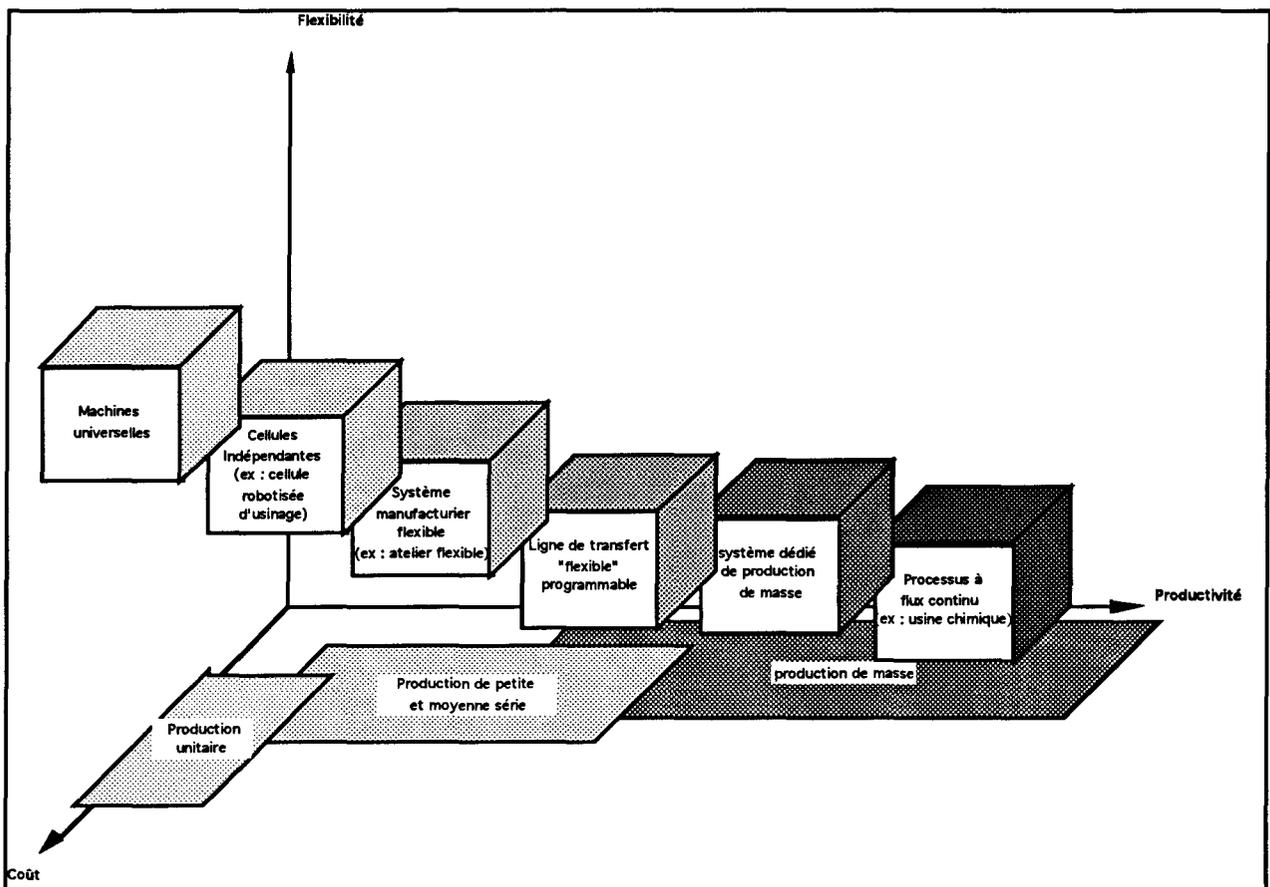
Cette mesure est d'autant plus difficile à appréhender qu'elle concerne à la fois le potentiel d'adaptation d'un système de production et ses performances. Qu'en est-il par exemple d'un système flexible qui ne fabrique qu'un seul produit en raison des exigences du marché ou d'une mauvaise politique de gestion et d'utilisation ?

Enfin, la flexibilité des systèmes de production n'est pas un concept simple que l'on peut considérer isolément [Bonetto, 85]. Elle est liée à la flexibilité de la commercialisation, de la

livraison, des approvisionnements, de la spécification des produits, de la conception...

#### I.2.4. Flexibilité et Productivité : le dilemme.

Pour conclure ce paragraphe sur la flexibilité des systèmes de production, nous présentons la dualité entre la productivité et la flexibilité d'un Système de Production. La figure I.1 montre que la flexibilité a un coût en terme de productivité. Les ateliers flexibles, objet de notre étude, sont dédiés à la production de petites et moyenne série. Ils présentent le meilleur compromis entre flexibilité et productivité. Pour être plus complet, il faudrait prendre en compte un axe supplémentaire qui représenterait l'investissement nécessaire pour chaque type de système de production. Le coût de cet investissement est fonction de la complexité technologique et du niveau des compétences nécessaires à la réalisation, à l'exploitation (maintenance et mise en œuvre) et à l'adaptation.



*Figure I.1 : Productivité, Flexibilité et Coût*

### I.3. Les Systèmes Flexibles de Production Manufacturière (SFPM)

La performance des coûts de production a toujours été le souci majeur de l'industrie manufacturière. Avant 1960, ce souci se traduisait par l'acquisition et l'utilisation de machines mono-axe ou multi-axe, ainsi que des lignes de transfert linéaires ou rotatives. Vers 1960, le marché s'orientait vers les produits fabriqués en petites et moyennes séries. Cette tendance est devenue

largement majoritaire dans les années 1980 et 1990, en particulier dans l'électroménager, la HI-FI, l'automobile, le textile, etc... Dans les années 1960, les machines à commande numérique sont suffisamment fiables et deviennent productives. La première machine à commande numérique directe voit le jour au Japon à la fin de la décennie 1960, et en Europe en 1973. Le premier grand pas vers les SFPM fut franchi au Japon en 1975 avec le premier Centre d'Usinage à commande numérique utilisant des systèmes automatiques de changement d'outils et de chargement/déchargement de palettes. Depuis, et plus particulièrement dans les années 1980, des milliers de SFPM de toutes tailles ont été implantés un peu partout dans le monde.

Un SFPM est donc "un atelier constitué d'un ensemble d'éléments capables de fabriquer un ensemble de pièces par programmation. Ce sont par exemple des machines outils à commande numérique et à changement d'outils automatique avec un système de transport automatique" [Ranky, 1990].

Il n'existe pas de SFPM standard, mais il y a autant de formes différentes que d'applications industrielles. On identifie toutefois deux catégories de base, les lignes de transfert flexibles et les ateliers flexibles :

- Les lignes de transfert flexibles sont le plus souvent des lignes d'assemblage de grande série. La notion de flexibilité s'y restreint souvent à un grand nombre de variantes d'un produit en phase de finition. C'est le cas notamment des chaînes de montage automobile.

- L'atelier flexible est une solution et un outil pour résoudre le problème de fabriquer le bon produit au bon moment en quantités nécessaires et suffisantes et au moindre coût [Barbier et al, 92].

Le Système Flexible de Production Manufacturière qui nous intéresse est "un atelier automatisé et intégré, composé d'un ensemble de machines-outils à commande numérique munies d'une gestion d'outils automatisée. Il dispose de systèmes de métrologie et d'assurance de qualité. Les composants de l'atelier sont reliés par un système de manutention automatisé et l'ensemble est commandé, contrôlé et surveillé par un système informatique" [Schmidt, 89].

Dans un atelier flexible, les opérations de transformation de même type sont parfois regroupées sur un même site de fabrication. Cependant, le regroupement de machines en sous-structure est souvent réalisé par rapport à des critères d'exploitation, c'est-à-dire, par rapport aux produits à fabriquer. Les sous structures (cellules ou îlots) peuvent comporter, dans ce cas, des machines de fonctionnalités différentes [Ferreira et al, 91]. Notons que la décomposition atelier/cellule/station/poste est courante dans l'industrie [Vidril, 95]. Cette décomposition n'est toutefois pas figée. Aucune norme ni consensus ne permettent de définir chacun des niveaux et chaque concepteur organise son système selon ses propres critères [Ranky, 90]. En effet, si l'accord est unanime pour concevoir de façon hiérarchique et structurée une architecture de production et sa

commande, la justification des différents niveaux est souvent empirique et dépend du point de vue adopté.

## **CHAPITRE II : L'INTÉGRATION DES SFPM AUTOMATISÉS**

En 1995, toutes les entreprises petites ou grandes sont sensibilisées aux questions de flexibilité de la production et, plus généralement, à l'automatisation. En effet, ces entreprises sont passées d'une politique de gestion optimisée des stocks à une politique de production à la demande. Elles ont à satisfaire simultanément cinq exigences [Ranky, 90] :

- Augmentation de la productivité.
- Diminution des coûts de production.
- Diminution des stocks et des en-cours.
- Choix du niveau de qualité des produits.
- Réduction de la taille des lots de produits (production unitaire)

Cependant l'automatisation d'un système de production ne suffit pas à assurer la performance d'une entreprise. En effet, cette performance dépend aussi de la conception des produits, de la logistique et de la commercialisation mais surtout de la cohérence de l'intégration des différentes fonctions de l'entreprise. Dès sa conception, l'automatisation du système de production doit tenir compte de la nécessité d'intégrer différents modules.

Deux éléments facilitent la réalisation de cette intégration : les techniques de la productique et le concept CIM (Computer Integrated Manufacturing). La productique intègre les techniques informatisées de production au sein de l'entreprise sans s'attarder sur les liens entre les services. Elle s'attache essentiellement à réaliser une intégration horizontale. Par contre, le concept CIM va plus loin en intégrant toutes ces techniques au sein de l'entreprise et en proposant une intégration verticale. [Barbier et al, 92]

D'autres auteurs définissent le CIM comme un système complexe multicouches qui fournit une aide pour une automatisation intégrée à tous les niveaux de l'entreprise en reliant entre eux des flots d'automatisation (au sens de [Ingersoll Engineers, 88]).

### **II.1. L'environnement du Contrôle/commande des SFPM**

Nous avons vu qu'un Système Flexible de Production Manufacturière peut être défini comme un système de fabrication programmable au sens le plus large. Il traite à la fois des données de haut niveau sur une informatique distribuée et des flux automatisés de matières. Il utilise des ressources de transformation commandées par une architecture multicouche intégrée [Ranky, 90].

L'environnement changeant de l'entreprise introduit des incertitudes liées aux fluctuations du marché dans la planification de la production. Les SFPM permettent de faire face à ces incertitudes puisqu'ils permettent la fabrication d'une grande variété de produits en quantités variables. Mais la flexibilité potentielle d'un équipement ne peut être exploitée que si le système de contrôle est lui-

même flexible : la flexibilité d'un SFPM ne doit donc pas être réduite par sa commande [Slack, 88]. Une commande bien conçue doit être robuste à des changements de flux de production et à des modifications mineures sur les produits (flexibilité à court terme) et permet l'adaptation à des changements d'équipements ou de produits moyennant des modifications locales grâce à sa modularité et sa transparence (flexibilité à long terme).

Le système de contrôle/commande doit notamment tenir compte des trois spécificités suivantes :

- \* les décisions en ligne telles que le pilotage et l'ordonnancement doivent être prises en temps réel afin de garantir la production, l'optimisation des flux et la meilleure charge des machines,

- \* la sécurité des hommes, des machines et des produits doit être assurée par des fonctions de surveillance et de supervision,

- \* la qualité des produits doit être garantie.

Le contrôle/commande traite un flux d'informations descendant véhiculant les décisions de ses niveaux hiérarchiques. Parallèlement, il prend en compte un flux d'informations ascendant issu des capteurs du système physique de production.

L'importance du traitement de l'information dans un SFPM justifie pleinement une approche globale du système. C'est le meilleur moyen de tenir compte des relations et des contraintes entre la flexibilité, le contrôle de production et les décisions de haut niveau relatives à des changements de l'environnement de la production.

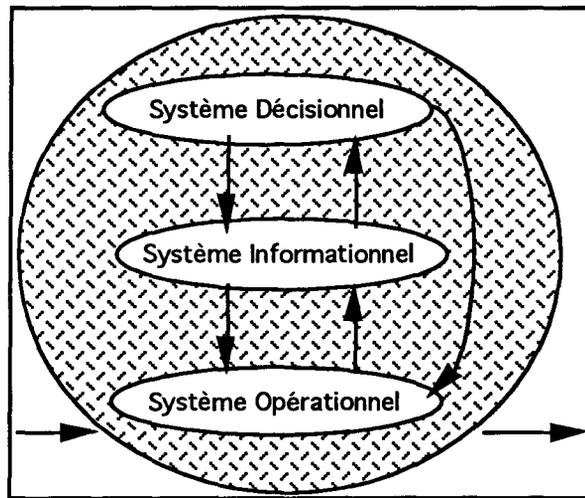
## II.2. L'intégration

La complexité croissante des systèmes de production requiert la modélisation de leur fonctionnement et leur comportement afin de les comprendre et de les intégrer. La simple identification des éléments d'un système complexe et leurs traitements isolés conduisent à une perte d'information et à des difficultés à appréhender leur fonctionnement [Le Moigne, 90]

L'approche systémique propose une modélisation des systèmes complexes selon un "modèle canonique" de neuf niveaux [Le Moigne, 90]. Le sixième niveau de cette forme de modélisation identifie pour tout système complexe trois couches (figure II.1).

Le **système décisionnel** (ensemble d'algorithmes, d'heuristiques, ...) élabore des stratégies intelligentes basées sur des données fournies par le système d'information. Le **système d'information** (base de données, format de données, ...) traite des données venant du système décisionnel vers le système opérant dont il enregistre le comportement pour le communiquer au système décisionnel. Le **système opérant** (ressources de production, de transport, ...) exécute des

ordres et informe le système d'information sur son état réel.



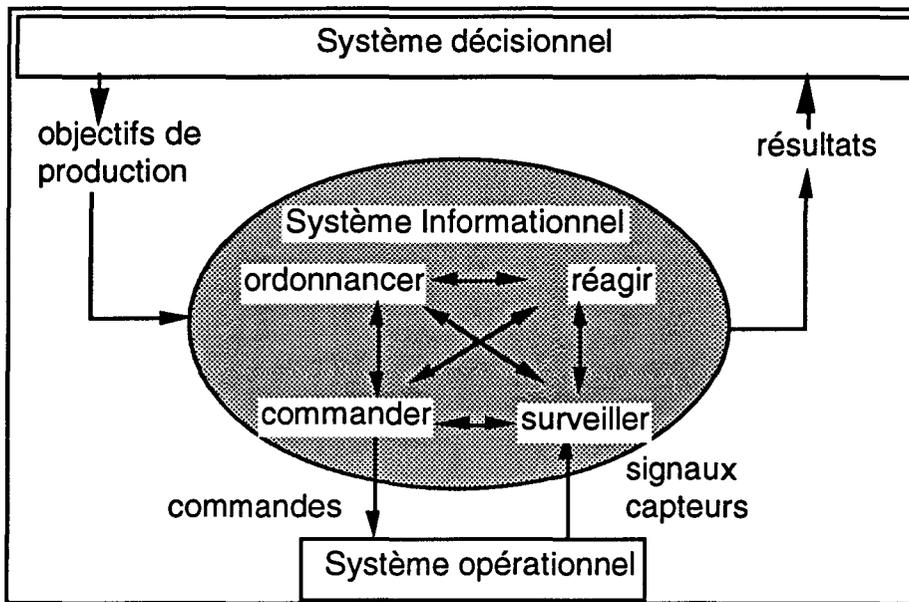
*Figure II.1 : Le modèle canonique OID [Le Moigne, 1990]*

L'application de cette approche générale aux SFPM et à leur environnement, permet de rapprocher le système décisionnel de la gestion de production qui décide des produits à fabriquer en fonction des requêtes du marché, de leurs dates dues etc.... Le système opératoire est le SFPM composé de machines à commande numérique, de robots et d'AGV ("Automatic Guided Vehicule" pour chariots filoguidés), ainsi que leurs actionneurs et leurs capteurs. Le système d'information correspond au système de contrôle : il traite exclusivement des informations de diverses natures soumises à des contraintes temporelles différentes.

Comme l'indique la description générale du système d'information, l'enregistrement et la mémorisation du comportement du système opérant complexe est d'une importance cruciale. Les modèles de comportement sont à la base du système de contrôle. Ces modèles de comportement représenteront nécessairement le SFPM sous des points de vue différents pour fournir les informations traitées dans chaque fonction du système de commande.

Ainsi, le système de contrôle/commande atteint lui-même un certain degré de complexité dû au fait que les mêmes informations sont traitées dans le système sous des aspects différents. La commande de coordination, par exemple, coordonne les ressources de production afin de garantir la bonne exécution des ordres programmés. Mais si un échec d'exécution est détecté par la surveillance, les fonctions d'ordonnancement et de coordination doivent en être informées. L'ordonnancement ne doit plus tenir compte de la machine défaillante et la commande de coordination doit empêcher l'utilisation de cette ressource ainsi que de celles qui lui sont liées. Les deux fonctions ont leurs propres points de vue du système opérant et leurs modèles respectifs nécessitent des informations dans des formats spécifiques. Cet exemple montre le lien complexe entre trois fonctions ; beaucoup d'autres exemples similaires pourraient être trouvés.

En général, nous pouvons représenter les fonctions du contrôle/commande d'un SFPM et leurs interactions globales par la figure II.2 [Archimède, 91].



*Figure II.2 : Les fonctions de commande d'un SFPM.*

La complexité des systèmes de contrôle nécessite une méthode de conception qui prend en compte les problèmes d'intégration dès les phases initiales. Dans le domaine du génie des systèmes, les différentes étapes d'une telle méthode de conception sont habituellement décrites par le cycle de vie suivant [Verdin, 91], [Daenzer, 87].

- \* analyse des besoins : définition du problème et spécification des besoins (Ce qui doit être fait)
- \* conception : définition de solutions différentes (Comment peut-il être fait ?) et choix d'une solution
- \* réalisation
- \* exploitation

Ce cycle de vie suggère une procédure séquentielle, mais en réalité, chaque phase a son cycle de vie propre et la démarche est fortement itérative. La spécification fonctionnelle du système de contrôle/commande présentée dans ce paragraphe, précise ce qui doit être fait ainsi que les exigences et les questions à résoudre dans les étapes suivantes. Nous cherchons une représentation claire des relations entre les fonctions du système puisque ces relations sont fondamentales pour sa bonne marche. Pour certaines fonctions, nous évoquerons à titre d'illustration des solutions qui sont du ressort de la conception, où le choix de la représentation et/ou la méthode de modélisation est déjà fait.

Dans la phase de définition du problème, nous devons intégrer des exigences de qualité

[ISO9000]. La qualité d'un modèle du système de contrôle/commande peut être jugé par les critères suivants [Booch, 91] :

- \* les modifications n'accroissent pas la complexité du système original (modifiabilité).
- \* le modèle du système de contrôle/commande exploite les ressources disponibles de façon optimale à l'égard des décisions de la gestion (efficacité).
- \* le modèle du système de contrôle/commande est exempt d'erreurs de conception. Il est capable de traiter des défaillances et des baisses de performance de manière sûre (fiabilité).
- \* le modèle du système de contrôle/commande est facile à comprendre. Il représente exactement notre connaissance du monde (transparence).

Le domaine du génie logiciel a défini certains principes pour aider à respecter les qualités citées ci-dessus [Booch, 91].

\* l'abstraction et l'encapsulation d'informations : l'information disponible pour un niveau donnée est filtrée et cachée pour les niveaux d'abstraction plus élevés afin de leur éviter une surcharge d'information.

\* modularité et localisation : l'objectif est la conception d'un ensemble de modules le plus faiblement couplés en assurant pour chaque module, la plus grande cohésion de ses éléments.

\* l'uniformité, l'intégralité et la validabilité : les spécifications des modules doivent employer le même langage. Leur fonctionnement doit être entièrement décrit et une validation de la spécification doit être possible. Notons que la nécessité d'un langage de spécification unique est contestée par des travaux récents sur les approches multiformalisme [Sahraoui, 95].

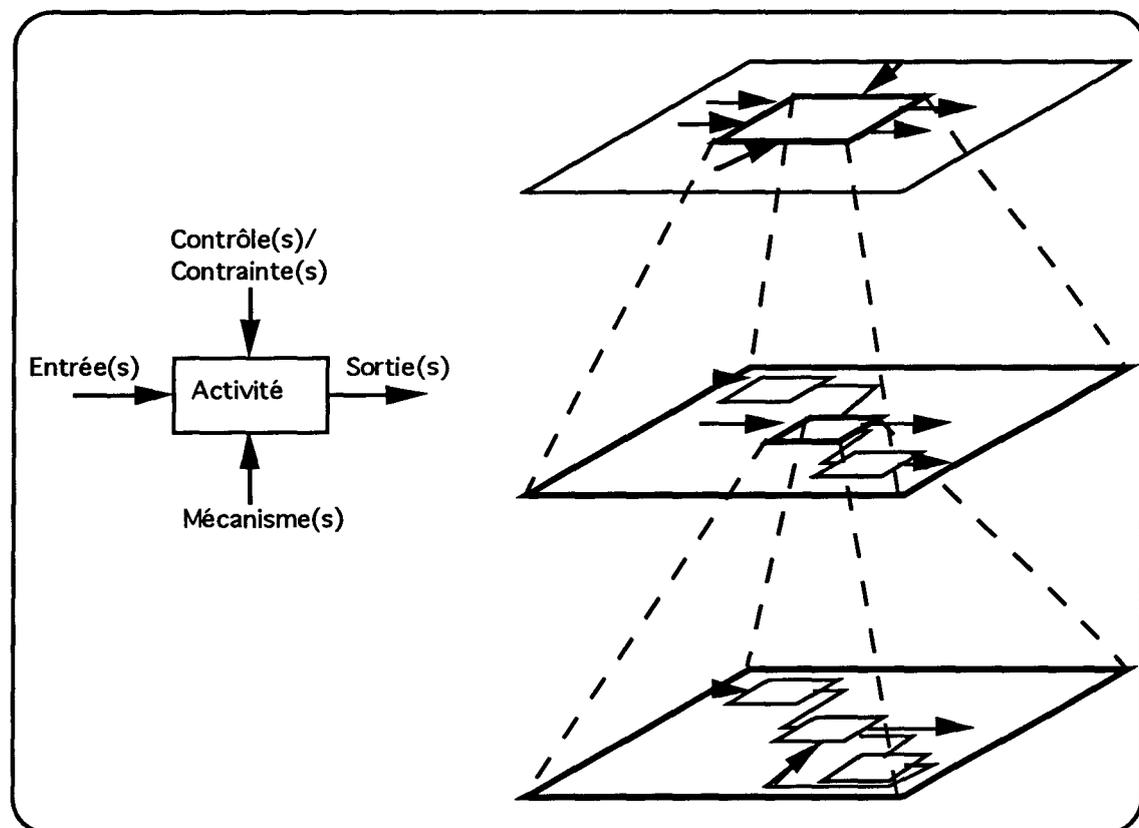
La complexité des SFPM requiert donc une approche globale pour le développement du système de contrôle/commande. Cette approche doit être systématique afin de respecter les exigences de qualité. Elle doit utiliser un formalisme capable de représenter notamment les communications entre les modules.

### **CHAPITRE III : SPÉCIFICATION FONCTIONNELLE DE LA CONDUITE D'UN SFPM [KERMAD ET AL., 93B]**

Pour spécifier, définir et organiser le système de contrôle/commande des SFPM, nous employons le modèle statique SADT [I.C.A.M.]. Ce modèle fournit un cadre pour bâtir les modèles dynamiques employés pour contrôler le système.

On trouvera dans [O'Grady, 92] une étude similaire mais moins détaillée qui met en avant les avantages d'une approche orientée objet de la modélisation, en partant d'une architecture de commande classique limitée à des cellules de production. [Millerat, 95] propose une approche de spécification du contrôle des centrales nucléaires fondée sur l'analyse de la valeur et qui pourrait être appliquée au domaine manufacturier. [Archimede, 91] présente une large description des approches de structuration de la commande proposées dans la littérature.

#### **III.1. Le formalisme SADT pour la spécification fonctionnelle**



*Figure III.1 : Actigrammes, définitions et décomposition.*

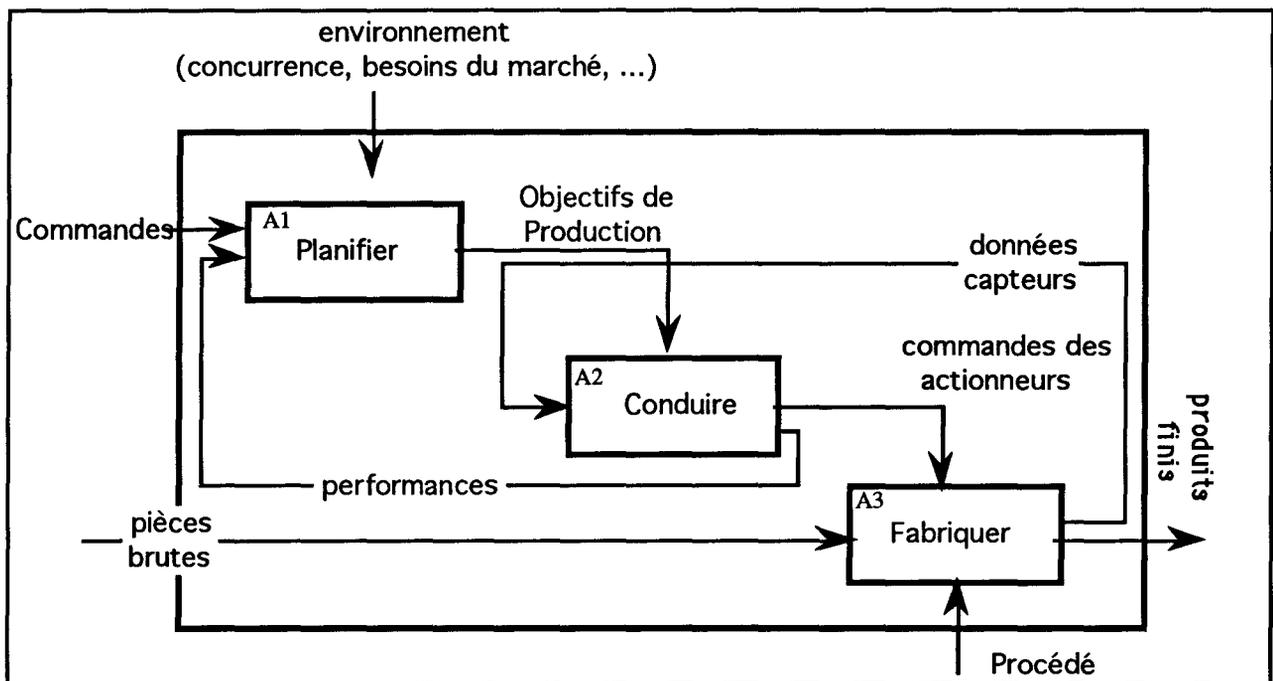
SADT est une méthode graphique de modélisation. Un modèle SADT complet est un ensemble de diagrammes qui illustrent les rapports entre les fonctions et les entités du système modélisé. Les diagrammes sont positionnés dans une structure arborescente qui peut être vue comme une

collection de descriptions de systèmes coordonnés [Lissander, 90].

Les modèles SADT sont bien adaptés à la spécification fonctionnelle des systèmes de contrôle/commande des SFPM, car ils décrivent les relations et les traitements d'informations des constituants d'un système. Nous utilisons dans cette partie la version "IDEF0" de SADT qui n'emploie que les actigrammes. En effet, les datagrammes de SADT ne sont utiles que pour vérifier la cohérence d'une spécification en donnant un modèle dual de celui que nous proposons ici.

### III.2. L'environnement de la conduite

Le système de contrôle/commande des SFPM est un système d'information. L'activité "conduire le système" échange des données avec la gestion de production et le système opérant, tous deux représentés par leurs activités dans la figure III.2.



*Figure III.2 : L'environnement de la conduite.*

#### **III.2.1. La planification de la production**

La planification de la production définit, sur un horizon donné, des objectifs de production en fonction de la demande du marché, des contraintes financières, des décisions stratégiques, des capacités du système et de la disponibilité matérielle pour ne citer que ces quelques paramètres de décision.

Dans les systèmes à flux tirés (PULL), les objectifs de production sont traduits par un calcul de stocks qui définit les quantités maximales et minimales de produits dans les stocks intermédiaires.

Dans un système à flux poussés (PUSH), les objectifs de production sont précisés par un ensemble de produits (commandes), leurs dates dues et leurs dates de début au plus tôt [Chassang, 83]. Le plan de production peut être plus ou moins détaillé. Il peut indiquer, par exemple, un routage spécial pour certains produits. Si un plan détaillé réduit la complexité du problème d'ordonnancement, il réduit aussi la réactivité du système puisqu'il élimine certains degrés de flexibilité de l'activité de pilotage.

La planification de la production travaille habituellement sur un horizon glissant. A chaque mise à jour, les performances du système fournies par l'activité de pilotage qui indiquent le temps, les écarts de qualité et de quantité du cycle de production précédent sont pris en compte.

### **III.2.2. La fabrication**

La fabrication décrit l'activité du système opérant qui transforme des matières premières et des produits semi-finis en produits finis en consommant de l'énergie. Les actionneurs reçoivent des ordres de l'activité de pilotage et les capteurs émettent des signaux qui sont traités comme des données dans la fonction de pilotage.

Le système opérant peut être structuré en groupes de ressources autonomes [Fu et al, 92], telles que les cellules flexibles de production, les stations de travail, etc. Suivant l'approche générale de boîte noire, nous supposons que chaque niveau structurel voit seulement ses composants de niveau inférieur. Ainsi, une cellule est composée de stations de travail et d'un système de transport. Une station de travail est composée de machines et manipulateurs, mais une cellule peut aussi être vue comme un ensemble de machines et un ensemble d'outils de transport si un groupement n'est pas utile.

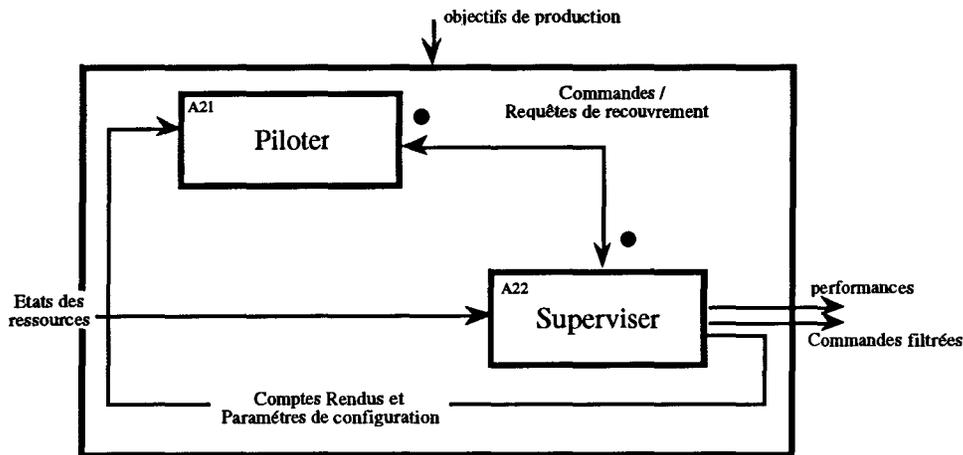
### **III.3. La conduite du système**

C'est la fonction qui, d'une part, transforme les objectifs de production en ordres pour le procédé et, d'autre part, garantit un comportement sauf du système. Le résultat est un flux bidirectionnel d'informations. Les informations descendantes indiquent ce qui doit être fait, et les informations ascendantes représentent les événements créés par des opérations exécutées ou par un comportement imprévu du système.

La communication entre la conduite et le procédé est essentiellement un échange d'ordre et de signaux capteurs déterministes. Elle est soumise à de fortes contraintes de type "temps réel" [André et al, 93].

La communication entre la conduite et la planification concerne des objectifs de production à atteindre et des performances réalisées. Ces informations résultent de calculs et d'analyses de signaux effectués sur un intervalle de temps correspondant à l'horizon de travail de la gestion de

production. La conduite présente à la fois un caractère synchrone et déterministe et un caractère asynchrone et transformationnel ou déterministe.

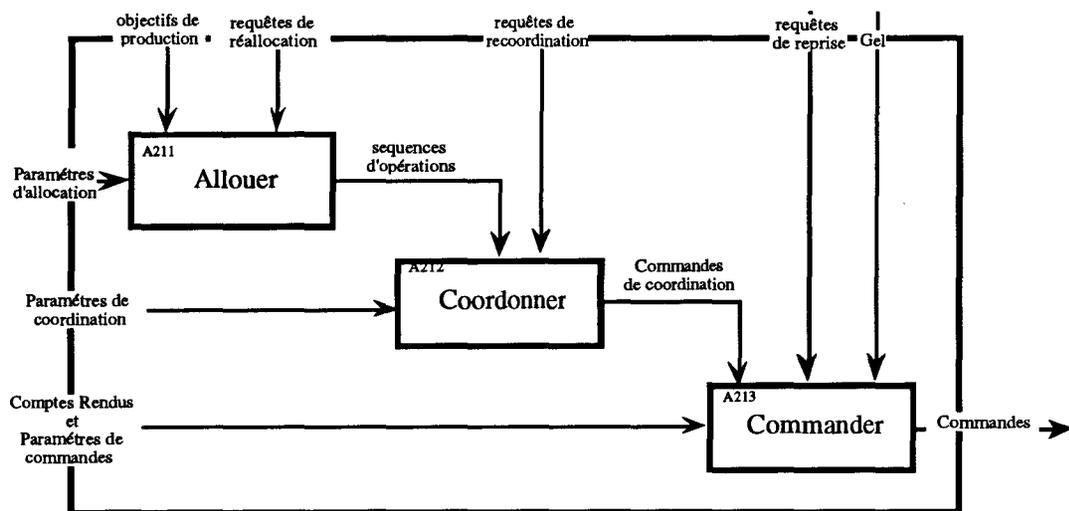


*Figure III.3 : l'activité "Conduire".*

La figure III.3 explicite le premier niveau de décomposition de l'activité de conduite. L'activité concernée par le flux descendant d'informations est appelée "pilote", le flux ascendant d'informations est traité dans la fonction "superviser".

Dans les paragraphes suivants, nous développerons d'abord l'activité de pilotage dans une stratégie de production en flux poussé. Ensuite, nous présenterons une décomposition de la fonction de supervision.

### III.3.1. Le pilotage du système



*Figure III.4 : L'activité "Pilote"*

La hiérarchisation du pilotage, essentielle dans la gestion du système de production, a été

abordée et traitée par de nombreux auteurs : [Benassy, 87], [Carlier et al, 88], [Hammadi, 91] et bien d'autres. La figure III.4 met en évidence une décomposition de la fonction "conduire le système" en trois activités :

- \* l'allocation des ressources
- \* la coordination des ressources
- \* la commande des ressources

#### L'allocation des ressources

Une synthèse de plusieurs approches des problèmes d'ordonnancement et d'allocation des ressources est présentée dans [Archimede, 91]. Il en ressort que le développement de l'activité "Allouer" ne peut se faire qu'en choisissant une politique d'ordonnancement fondée sur la décomposition des activités d'allocation et de planification.

La fonction d'allocation des ressources définit un programme, où chaque ordre est exécuté en réalisant une variante du produit. Chaque opération de cette variante est attribuée à une ressource pour toute sa durée. L'allocation des ressources est souvent un problème difficile. Plusieurs approches de décomposition sont proposées dans la littérature. Nous en présentons brièvement deux.

- La décomposition spatiale est une approche décentralisée qui répond au souci d'augmenter l'autonomie des cellules. Elle tient compte des propriétés spécifiques des SFPM notamment leur organisation en unités autonomes. Cette approche affecte les opérations sur des sous-ensembles de machines de l'atelier afin de réduire la complexité de l'ordonnancement [Ferreira et al, 91]. Les sous-ensembles de l'atelier sont identifiés soit, à partir de groupes d'opérations définies pour une famille de produits (groupement en îlots), soit, à partir de l'organisation de l'atelier.
- la décomposition hiérarchique ou fonctionnelle alloue en premier un sous-ensemble de ressources, le goulot d'étranglement du système ou les machines de transformation par exemple. Un deuxième module d'affectation affecte les machines restantes. On réduit ainsi l'ensemble des ressources et des opérations à allouer [Archimede, 91]. D'autres approches hiérarchiques groupent les machines de fonctionnalité identique en réduisant ainsi l'ensemble des ressources.

Dans la plupart des approches, le choix des variantes de produits est déjà fait quand l'ordre de production est défini [Roubellat et al, 88]. Cette démarche réduit la complexité du problème mais l'ordonnancement y perd en réactivité. En effet, si une ressource nécessaire à la variante prédéfinie d'un produit est hors service, l'ordre correspondant ne peut pas être ordonnancé et un retour à la planification est nécessaire.

Beaucoup d'autres solutions sont possibles : les solutions statiques (règles d'ordonnancement), allocation sur requête (plusieurs produits requièrent en même temps la même ressource) et toutes les combinaisons des approches mentionnées précédemment.

Dans le cadre du projet CASPAIM, [Hammadi, 91] propose une allocation prévisionnelle des transformations (ordonnancement), dans un premier temps, puis une allocation des ressources de transport dans un deuxième temps.

Le modèle d'allocation des ressources de transformation représente le processus de fabrication comme une séquence d'opérations à réaliser sur un produit par les ressources de production. L'allocation des moyens de transport met en œuvre des mécanismes de résolution comme les allocations prévisionnelles [Fu et al, 92] ou des allocations par calcul de priorité [Tawegoum, 95].

### La Coordination des ressources

L'allocation des ressources propose une séquence de transports et de transformations pour chaque produit, la coordination des opérations ordonnancées pour chaque produit et la coordination des ressources qui les exécutent sont les tâches principales de la commande de coordination.

La coordination des ressources de production devient cruciale durant les opérations de transfert, car les ressources qui échangent des produits doivent être synchronisées. Les modèles utilisés par l'activité de coordination doivent permettre l'expression de la synchronisation des procédés ainsi que leurs évolutions parallèles, parce qu'habituellement le système fabrique plusieurs produits en même temps. En outre, le modèle de la commande ne doit pas évoluer avant la confirmation de la bonne exécution des ordres donnés (sécurité de la commande de coordination) [Sahraoui, 87].

Le projet CASPAIM propose un modèle réseau de Petri hiérarchique pour la commande de coordination qui permet une modélisation du parallélisme et de la synchronisation [Ausfelder, 92] en respectant les contraintes de sécurité.

### La commande des ressources

La fonction de coordination aboutit à l'identification des ressources et de leurs commandes. L'activité "Commander" exécute l'ordre et transmet le changement d'état demandé à l'activité de supervision. Cette activité garantit que la commande du procédé est conforme à son état réel. Des filtres sont proposés comme modèles afin de détecter les ordres erronés [Alanche et al, 86], [Elkhatabi et al, 92]. L'approche présentée par S. Elkhatabi [Elkhatabi, 93] et développée pour le projet CASPAIM, utilise des modèles de comportement des ressources qui intègrent des contraintes opérationnelles et fonctionnelles. Le modèle de comportement est traduit en ESTEREL [Berry et al, 87], un langage de programmation synchrone, et tient compte des contraintes temps-réels fortes à l'interface entre le procédé et le système de contrôle/commande.

Les trois fonctions décrites dans ce paragraphe peuvent aussi bien être activées par une requête de recouvrement formulée par la fonction de supervision. L'arrêt d'urgence, considéré comme un recouvrement particulier, est développé afin de montrer son caractère unique.

### Conclusion

Nous avons illustré à travers les activités "allouer", "coordonner" et "commander" les différents aspects de l'information traitée par la fonction de conduite du système.

Les travaux proposés dans le projet CASPAIM considère jusqu'à présent une conduite hiérarchisée sur deux niveaux : le procédé et la commande. L'allocation affecte les opérations et détermine l'ordre de passage des produits sur les ressources de l'atelier. Cette fonction utilise des modèles qui représentent à la fois les contraintes sur les produits et les machines. La coordination s'intéresse aux produits et à la synchronisation des ressources. Les modèles utilisés sont associés à la représentation de la séquence des opérations à réaliser sur un produit ou à la représentation du comportement spécifique de chaque ressource. Enfin, la commande des ressources émet des ordres élémentaires aux actionneurs de la partie opérative représentée indépendamment des produits.

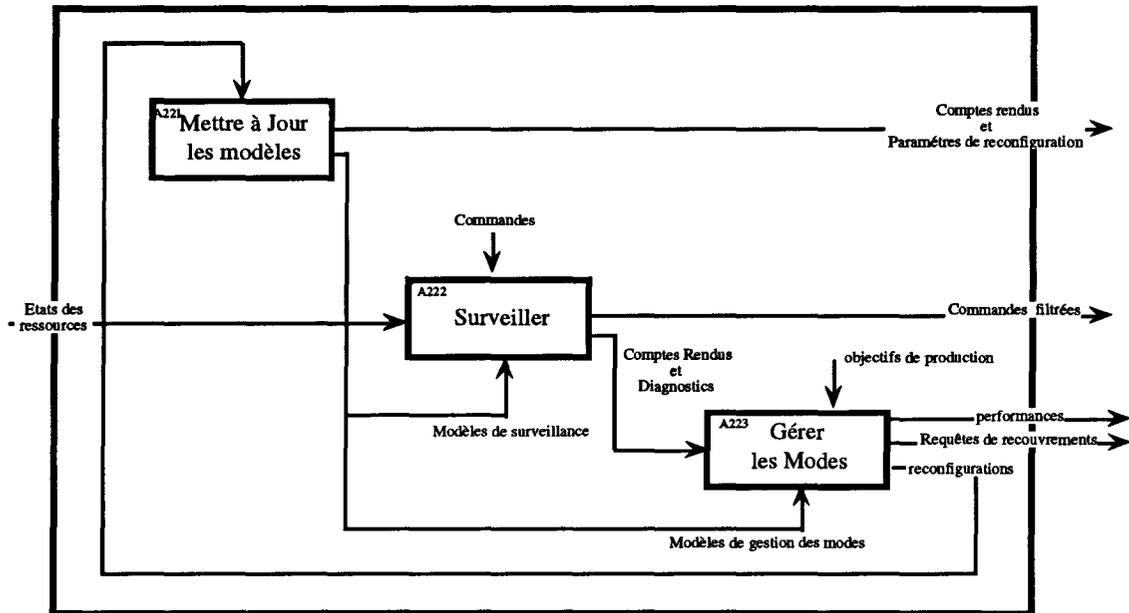
### **III.3.2. La supervision**

Un système de supervision optimise la production, le fonctionnement et la sécurité des systèmes automatisés dans un contexte de production spécifique. En effet, il intègre plusieurs sous-systèmes [Milot, 88] :

- la surveillance,
- la gestion des modes de marche,
- l'interface Homme-Machine.

Nous n'aborderons pas dans cette partie les problèmes liés à l'ergonomie et en particulier des interfaces hommes-machines. Le lecteur intéressé par cet aspect pourra se reporter aux travaux du LAMIH [Milot, 88]. Par contre, nous ajouterons à la supervision une fonction de mise à jour des modèles du SFPM. En effet, les fonctions de supervision et de pilotage travaillent sur des modèles qui doivent être mis à jour afin de fournir une image fiable du procédé. Nous distinguons deux types de mise à jour des modèles : la mise à jour interne et la mise à jour d'états. Chaque activité modifie les paramètres des modèles qu'elle utilise ; c'est la mise à jour interne. La mise à jour d'état concerne les modèles de toutes les activités : le système opérant doit changer son état à l'échéance d'une commande ; la surveillance établit son état réel. Une entité du procédé est représentée dans des formats différents dépendant de l'activité concernée. L'état de la même entité physique doit être identique dans tous les modèles et les modèles représentant une entité doivent correspondre à son état réel. L'information sur l'état doit être cohérente et fiable.

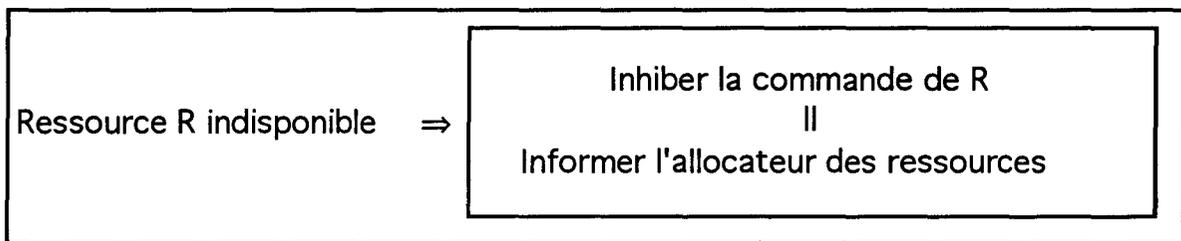
Nous supposons tout au long de ce document que la mise à jour interne des modèles est gérée par chaque activité. Nous parlerons donc uniquement de l'activité de mise à jour des états.



*Figure III.5 : L'activité "Superviser"*

### La mise à jour des modèles

Cette activité doit assurer la cohérence et la fiabilité de l'information sur l'état réel du système et de ses composants dans les différents modèles employés par les activités. Nous pouvons illustrer ce propos à l'aide de l'exemple suivant :



Si la surveillance détecte l'indisponibilité d'une ressource de production, la commande ne doit plus la solliciter, son modèle doit donc être mis à jour au plus tôt. La fonction d'allocation des ressources doit simultanément être informée que cette ressource est hors service pour ne plus en tenir compte.

L'activité de mise à jour des modèles doit assurer ce traitement d'information ainsi que sa cohérence temporelle, i.e., une ressource qui est hors service dans un modèle d'activité, doit être instantanément hors service dans tous les autres modèles d'activité (principe de diffusion).

L'activité de mise à jour des modèles ne doit pas être confondue avec une activité de mise à jour

d'une base de données, car la représentation des modèles dans les bases de données nécessite un choix préliminaire d'implantation. Or, ce choix fait partie de la phase de mise-en-oeuvre du cycle de vie.

Les activités de la conduite travaillent sur des modèles du procédé qui n'ont pas nécessairement le même degré d'abstraction. Certaines fonctions de surveillance utilisent des Objets Commandables Élémentaires (OCE), la fonction d'allocation peut allouer un ensemble de ressources (= machines virtuelles), l'activité de mise à jour des modèles utilise elle-même des représentations différentes du procédé. Ces modèles représentent les liens de dépendance entre différentes entités virtuelles ou structurelles du système. La décomposition structurelle du système présentée par S. Bois [Bois, 91] que nous aborderons ultérieurement apporte les premiers éléments de solutions.

L'activité de mise à jour des modèles travaille dans deux directions : du niveau des OCE vers les niveaux cellule et système et dans le sens inverse du niveau système vers le niveau OCE. Cette mise à jour doit pouvoir être effectuée à chaque niveau. Une machine est par ailleurs composée de différents OCE. Si l'un d'eux change d'état, la machine change d'état. De même, si la machine change d'état, tous ses composants sont susceptibles d'être remis à jour. Nous disons que la machine et ses OCE sont dans une configuration de dépendance conjonctive.

De même que la fonction de surveillance, l'activité de mise à jour des modèles n'est pas seulement concernée par les modèles des ressources de production. L'état d'un produit, précisant par exemple, sa localisation réelle, sa marge restante, les opérations exécutées, et sa qualité doit être représenté et diffusé dans le système de contrôle/commande.

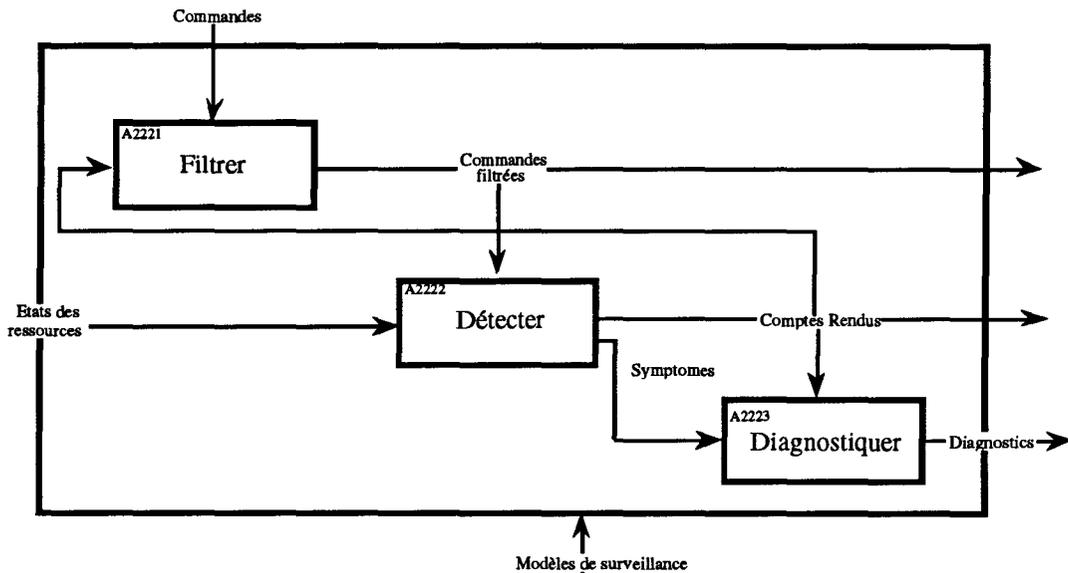
### La surveillance

La figure III.6 présente une décomposition de l'activité de surveillance fondée sur les travaux de A.E.K. Saharoui [Sahraoui, 87], M. Combacau [Combacau, 91], A. Toguyeni [Toguyeni, 92] et S. Elkhatabi [Elkhatabi, 93].

Dans le cadre d'une surveillance séparée de la commande [Sahraoui, 87], la détection compare le comportement attendu du système avec son comportement réel [Combacau, 91]. Les résultats de cette comparaison sont ensuite transmis à la commande et, en cas d'erreur, à l'activité de diagnostic (et de localisation) sous forme de symptômes.

Le projet CASPAIM a développé deux approches de la surveillance. Une première approche, proposée dans [Toguyeni, 92], modélise le comportement des éléments du système de production par Réseaux de Petri Temporisés qui évoluent en fonction de signaux des capteurs attendus, inattendus ou en retard et réalise la détection et le diagnostic des défaillances du procédé. Une

deuxième approche, proposée dans [Elkhatabi, 93] s'intéressent aux "Objets Commandables Elémentaires" (OCE). Cette approche utilise des modèles d'automates implémentés en ESTEREL et qui représentent le comportement des OCE pour filtrer les commandes et assurer la sûreté de fonctionnement du procédé.



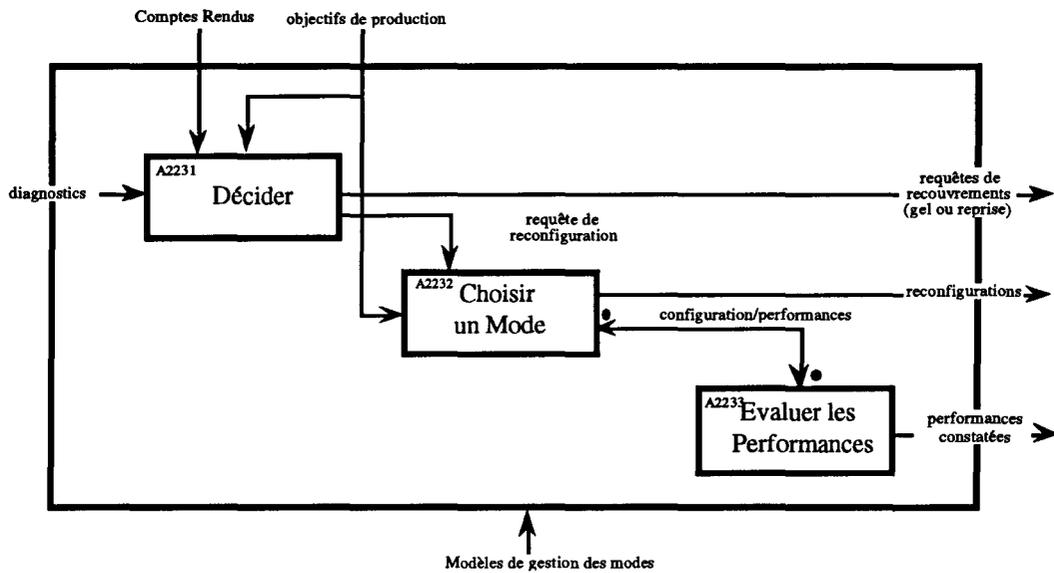
*Figure III.6 : l'activité "Surveiller".*

Quand un comportement anormal est détecté, les symptômes sont transmis au diagnostic. L'activité de diagnostic localise et identifie les composants défectueux. A. Toguyeni [Toguyeni, 92] a présenté une fonction de diagnostic qui travaille sur un modèle fonctionnel du procédé.

Les activités de la fonction de surveillance concernent non seulement les ressources de production mais aussi les produits. Les produits finis doivent être identifiés, les défauts de qualité ou les retards doivent être détectés et les performances réelles du système doivent être calculées. Les causes des défauts de qualité et des retards doivent être localisées, leurs conséquences doivent être évaluées et enfin une décision (réordonnancement, par exemple) doit être prise. Comme nous pouvons le voir, les activités sont les mêmes. Seuls les modèles sont différents.

### La gestion des modes

Cette fonction sélectionne la configuration du système qui concilie les objectifs de production d'une part et l'état courant du système d'autre part. Cette fonction doit décider des actions à mener dès l'occurrence d'un événement que ce soit dans le système ou dans les objectifs de production.



*Figure III.7 : L'activité "Gérer les Modes"*

L'activité "Décider" définit les actions de recouvrement en fonction des comptes rendus et des diagnostics issus de la surveillance et de l'état du système ainsi que de la réalisation des objectifs visés. L'activité "Décider" assure, d'une part, la sécurité du système et, d'autre part, elle augmente sa disponibilité.

L'occurrence d'une erreur sur le procédé déclenche la mise en œuvre d'actions de recouvrement en fonction de la gravité de l'erreur. Les actions de recouvrement modifient la commande en modifiant les paramètres de ses modèles afin de mettre en œuvre les stratégies de maintien de la production. Les actions de recouvrement sont réparties en trois familles [Sahraoui, 87], [Craye, 89], [Sahraoui, 92] :

- gel,
- reprise,
- reconfiguration.

Le **gel** de la commande interdit d'envoyer certains ordres au procédé. Ce verrouillage de la commande porte exclusivement sur les modèles de la commande. Ces modèles sont mis à jour pour ne faire apparaître que les solutions réalisables et intègrent, par conséquent, un mécanisme de verrouillage traduisant le gel d'une partie de la commande. Ce mécanisme de verrouillage peut s'exprimer par une condition fonctionnelle d'envoi de commande (prédicat) ; le verrou étant une condition supplémentaire [Sahraoui, 92].

La **reprise** consiste à ramener le système dans l'état stable précédent l'erreur que nous appellerons point de reprise [Ausfelder, 94]. Cette possibilité de recouvrement est conditionnée par l'existence d'actions réversibles dans le procédé. Nous distinguons deux types de points de reprise :

les points de reprises d'une ressource et les points de reprise d'un produit.

La **reconfiguration** permet le remplacement d'un module de commande par un autre en cas d'anomalie. Cette reconfiguration peut être statique ou dynamique [Sahraoui, 92]. La reconfiguration dynamique correspond à un traitement d'exception à la suite d'une perturbation dans le procédé. Ce traitement est équivalent à l'exécution d'une séquence possible de commandes. La reconfiguration dynamique est alors un reparamétrage de la commande. Des stratégies opportunes de reconfiguration et de choix de modes permettent l'exploitation de la flexibilité potentielle du système et de maintenir la production.

### Conclusion

L'importance de la flexibilité d'un système de production apparaît clairement à travers la spécification de l'activité "Gérer les modes". La modélisation de la flexibilité et son exploitation par les mécanismes de la gestion des modes et de recouvrement sont intimement liées. Il est évident que la disponibilité du système dépend essentiellement de la réalisation de cette activité.

La spécification fonctionnelle du contrôle/commande des SFPM présentée dans ce chapitre est le cadre général du travail de recherche du projet CASPAIM au LAIL.

Les solutions brièvement décrites pour certaines activités requièrent la cohérence des différents modèles. c'est dans ce sens que nous évoquons le concept d'intégration.

Notre spécification ne vise pas une automatisation complète du système de contrôle. Nous n'appréhendons aucune solution ni le fait qu'un opérateur humain puisse exécuter une ou plusieurs des fonctions décrites. Nous n'avons pas, non plus, tenu compte des contraintes de coût liées à la conception d'un système de contrôle/commande.

## **CHAPITRE IV : LA GESTION DES MODES DANS LA LITTÉRATURE**

La problématique de la gestion des modes d'un SFPM apparaît dans les blocages et les modes de marche non compatibles et la sécurité de fonctionnement des installations. La sécurité de fonctionnement est étroitement liée à la notion de mode de marche dans les normes [Afn, 86] et [CEE, 91]. Ces normes précisent en particulier l'état "Arrêté" et distinguent les commandes d'arrêt normal, les commandes d'arrêt de sécurité et les commandes d'arrêt d'urgence. Elles précisent également, les conditions de passage d'un état d'arrêt à un état de marche.

En ce qui concerne les problèmes de blocage et de modes de marche non compatibles, il s'agit de traiter des problèmes :

- de reconfiguration qui interviennent en cas de changement de production. La procédure de reconfiguration indique la marche à suivre pour clore la production en cours, pour réorganiser la cellule et enfin pour la reconditionner pour une reprise de la production au mieux (critère de temps de charge ...) [Sahraoui, 87].
- de fonctionnement en mode dégradé : le modèle doit décrire de quelle manière et sur quels critères l'atelier passe en mode dégradé.
- de reprise et de réintégration : quand une machine, suite à une panne ou une modification, est hors ligne, il faut la réintégrer dans sa cellule après réparation. Donc pour passer d'un état dégradé à un état de production normale, il est nécessaire d'amener le système dans un état spécifique, compatible avec la réintégration de la machine dans la cellule. Cet état peut être caractérisé par une position particulière, par un état (ex : hors énergie) de la partie opérative, par un état de la partie commande (marche, arrêt...), par une configuration spécifique des dispositifs de sécurité (logiciel et/ou matériel)... Il est nécessaire de restituer au système de fabrication un état permettant cette reprise.

Le Gemma est la première approche de formalisation de la gestion des modes de marche. Il précise les différents modes de marche et d'arrêt pour un ensemble "Partie Opérative + Partie Commande " unique. Les concepts du Gemma ont été la base de plusieurs travaux que nous rappelons ci-après.

### **IV.1. Le GEMMA**

Le GEMMA est un ensemble de spécifications proposées par l'ADEPA [ADEPA, 79], [ADEPA, 92] dans le cadre de la gestion des modes de marche d'un Système Automatisé de Production (SAP). Il distingue trois familles de procédures :

- Les procédures de fonctionnement qui englobent les modes de production normale, de marches de préparation, de clôture, de vérification et de test.
- Les procédures d'arrêt qui décrivent les modes conduisant à un arrêt du système pour des raisons extérieures. On distingue les modes suivants :
  - Arrêt dans l'état initial
  - Arrêt demandé en fin de cycle
  - Arrêt demandé dans un état déterminé
  - Arrêt obtenu
  - Préparation pour remise en route après défaillance
  - Mise de la Partie Opérative dans l'état initial
  - Mise de la Partie Opérative dans un état déterminé
- les procédures suite à une défaillance qui regroupent les modes conduisant à un arrêt du système à cause de défaillances. On y retrouve les modes :
  - Arrêt d'Urgence
  - Diagnostic et/ou traitement des défaillances
  - Production tout de même

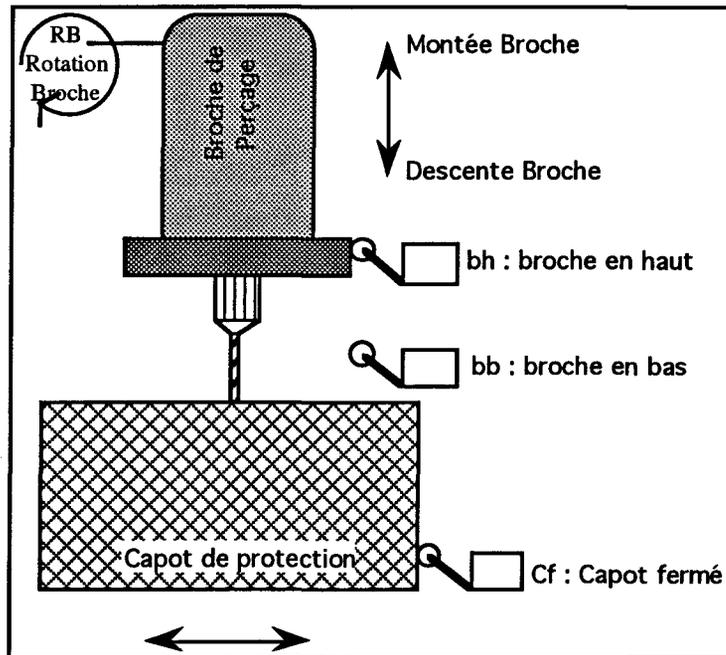
Ce guide est adapté à la définition des modes de marches d'un SAP analysé d'un point de vue global : une machine-outil automatisée, par exemple. Dans ce contexte, le GEMMA permet :

- une définition globale des marches souhaitées et une identification formelle des différents états que peut prendre l'équipement,
- la formalisation des conditions de passage d'un état à l'autre, notamment du point de vue du dialogue opérateur/équipement,
- l'expression des contraintes globales à respecter (contraintes de sécurité, de conduite, de maintenance).

Le GEMMA présente l'avantage de spécifier les modes de marche sous la forme d'un ensemble d'états mutuellement exclusifs, ce qui permet de structurer la commande de l'équipement. Ainsi, on pourra associer à chaque mode de marche une commande spécifique. Par exemple, si la commande

est conçue en Grafcet, chaque mode permettra de valider un ensemble de graphes de commande. L'exemple simplifié du perçage semi-automatique de la figure IV.1 extrait de [ADEPA, 92] illustre cette approche.

La partie opérative est composée d'une broche de perçage commandée automatiquement en rotation, en descente et en montée. Le montage ou le démontage des pièces ainsi que la mise en place du capot de protection.



*Figure IV.1 : Perçage semi-automatique [ADEPA, 92]*

On prévoit deux modes principaux :

- Mode Automatique : états 1 et 2.
- Mode Défaillant : états 3 et 4.

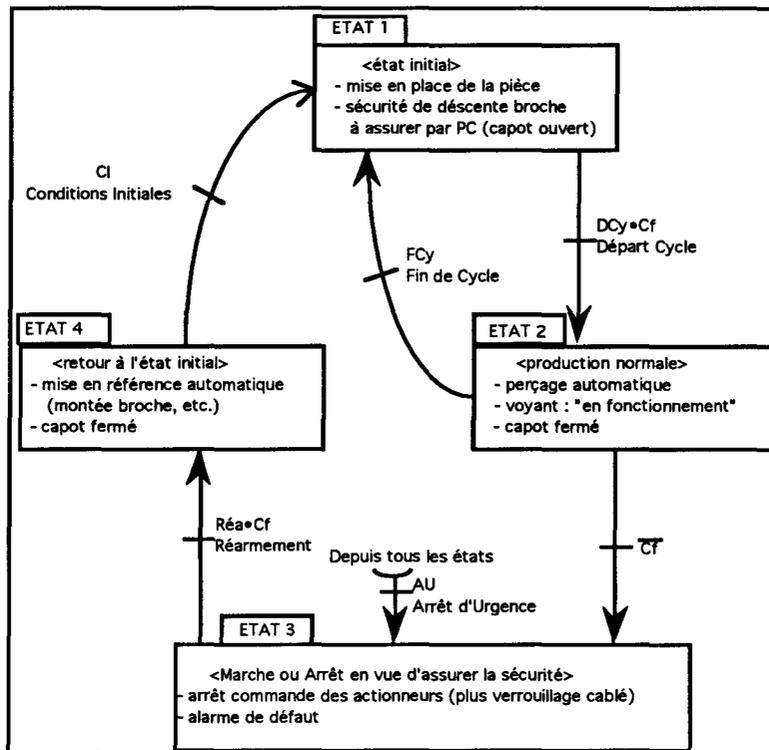
Etat 1 : la mise en place de la pièce est possible, la partie commande assurant la sécurité vis à vis de la descente de la broche tant que le capot est ouvert.

Etat 2 : le bouton "départ cycle" permet le passage à l'état 2 et le perçage automatique. La fin du cycle de perçage provoque le retour à l'état 1. L'ouverture du capot dans cet état mène à l'état 3.

Etat 3 : "arrêt d'urgence" mène à l'état 3 depuis tous les autres. La commande des actionneurs est suspendue en plus du verrouillage câblé directement sur l'organe de service.

Etat 4 : c'est la remise en référence de l'équipement automatique par l'information "Réarmement" quand le capot est fermé. L'équipement atteint l'état 1, dès que les conditions

initiales sont réalisées.



*Figure IV.2 : Représentation des états du système de perçage*

L'application de ces principes à l'exemple du perçage semi-automatique est représentée par le Grafset global (figure IV.3) structuré autour des modes de marches. Il comporte les Grafset partiels suivants :

GMM, de niveau hiérarchique supérieur, assure la Gestion des Modes de Marches en forçant les deux Grafset de niveau inférieur GPN et GIP.

GPN réalise le cycle automatique de perçage et Gère la Production Normale dès que l'étape 2 (de GMM) est active.

GIP exécute la remise en référence et Gère l'Initialisation de la Partie Opérative sur activation de l'étape 4 (de GMM).

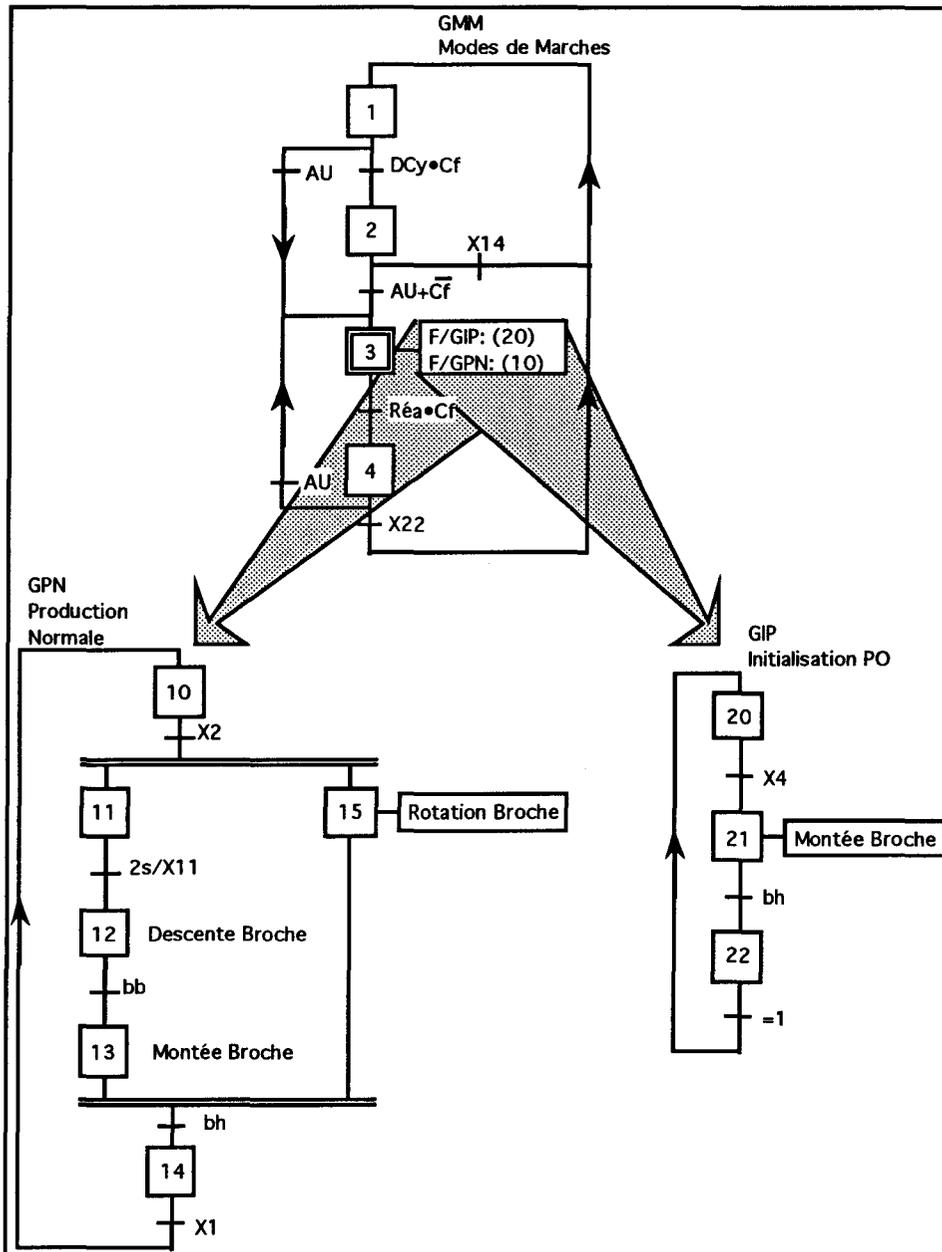


Figure IV.3 : Structuration de la commande du point de vue des modes de marche [ADEPA, 92].

#### IV.1.1. Limites du Gemma et solutions envisagées

Le GEMMA ne permet que l'expression des modes d'une machine ; il n'est pas prévu pour des systèmes composés. Ses principales limites sont l'unicité de mode et son inadéquation avec une commande hiérarchisée. Il est fortement lié à une mise en œuvre de la commande en Grafset dont les caractéristiques hiérarchiques sont limitées.

Les modes d'un SFFPM doivent être vus sous un angle moins restrictif que celui préconisé par le GEMMA. Nous avons retenu de ce dernier la notion de famille de modes que nous avons étendue. Dans chaque famille de modes, nous retrouvons la notion d'états exclusifs. Par contre, ces familles

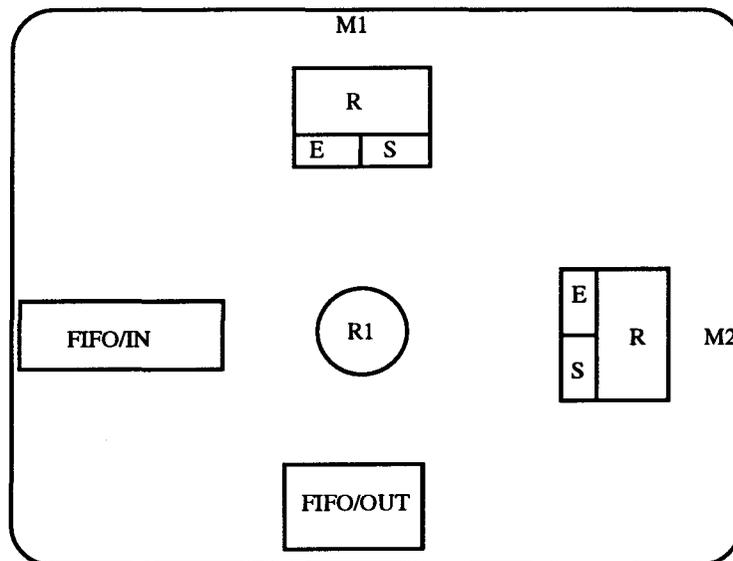
évoluent, a priori, en parallèle. Cependant elles peuvent être contraintes dans certains cas déterminés par le concepteur. Ainsi, nous identifions les familles suivantes :

- **Modes de marche** : automatique, Cycle par cycle ou pas à pas.
- **Modes d'arrêt** : hors tension, en fin de cycle, dans un état déterminé. En outre, la position d'arrêt pour chaque mode peut être précisée.
- **Modes de fonctionnement** : normal, dégradé ou hors service.
- **Modes d'exploitation ou d'Utilisation** : test, production ou maintenance. Le mode d'exploitation en production peut lui-même être décomposé en sous modes : préparation, production effective, clôture. Il est à noter ici une règle simple pour conduire la composition des modes. La règle consiste à privilégier par ordre de priorité les points de vue production, puis test, et enfin maintenance. Ainsi, une unité composée sera en production si au moins un de ces composants est en production. Par contre, cette unité sera en maintenance, si tous ses composants le sont.
- **Modes opératoires** : ils spécifient le type d'opérations effectuées sur les produits.

Le modèle que nous proposons enrichit en informations le GEMMA. En effet, un SFPM peut, par exemple, être en mode d'exploitation production/préparation et être parallèlement en mode de fonctionnement dégradé. Cela signifie qu'une défaillance est survenue au cours de la marche de préparation. Cet état du système est impossible à modéliser avec le GEMMA classique.

En effet, le GEMMA ne permet pas de prendre en compte de manière systématique la hiérarchie qui caractérise la conception des SFPM. En effet, la complexité qui caractérise un SFPM nécessite sa structuration et sa hiérarchisation en regroupant ses constituants en entités de plus en plus agrégées. L'interprétation d'un mode du système par rapport aux modes de ses composants pose alors problème. L'unicité du mode actif du GEMMA devient difficile à appréhender et à représenter. L'exemple de la figure IV.4 va nous permettre d'illustrer ces problèmes. Nous y trouvons deux machines M1 et M2 servies par un robot R1 et alimentées en pièces par une file d'entrée (FIFO-IN) et évacuées par une file de sortie (FIFO-OUT).

Une première approche, descendante, consisterait à effectuer une analyse des modes du système puis de ses composants. Supposons que le système soit dans l'état "production tout de même". Que signifierait cela du point de vue des machines ? Plusieurs interprétations sont possibles : soit l'une des deux machines (M1 ou M2) est dans l'état "production tout de même". Soit la machine M1 (respectivement M2) est en "marche ou arrêt en vue d'assurer la sécurité" tandis que la machine M2 (respectivement M1) est en "production normale". On pourrait ainsi multiplier les interprétations.



*Figure IV.4 : Les machines M1-M2*

Une deuxième approche, ascendante, consisterait à établir des modes de marche pour chaque composant (machine) du SFPM et déduire par composition les modes de marches du système global. Cette approche n'est pas évidente à réaliser, car elle doit intégrer le point de vue du concepteur. Ce point de vue dépend de l'utilisation et des objectifs en exploitation de ces modes de marche : mise en route et arrêts des machines, préemption, recouvrement des défaillances, etc...

#### **IV.2. Approche du Laboratoire d'Automatique Industrielle de l'I.S.M.C.M. [Anakök, 87]**

Une approche systémique permet de définir la finalité et l'environnement d'un SAP. Yéva Anakök décrit une organisation interne des SAP en niveaux hiérarchiques en interaction et montre la dualité entre le SAP et le produit. Cette dualité conduit à deux approches complémentaires : une approche par des "objectifs produits" et une approche par des "objectifs systèmes".

La première approche par des "objectifs produits" permet de déterminer les moyens de production et d'organiser la commande en partant des caractéristiques du produit et de ses phases d'élaboration. Cette approche ascendante permet de créer des organisations de plus en plus abstraites et complexes, jusqu'à la définition des "objectifs systèmes". L'approche par des "objectifs systèmes" permet de définir les contraintes et les liens d'exclusion et d'évolution entre les stades et les modes du SAP et de gérer, en conséquence, son organisation dynamique.

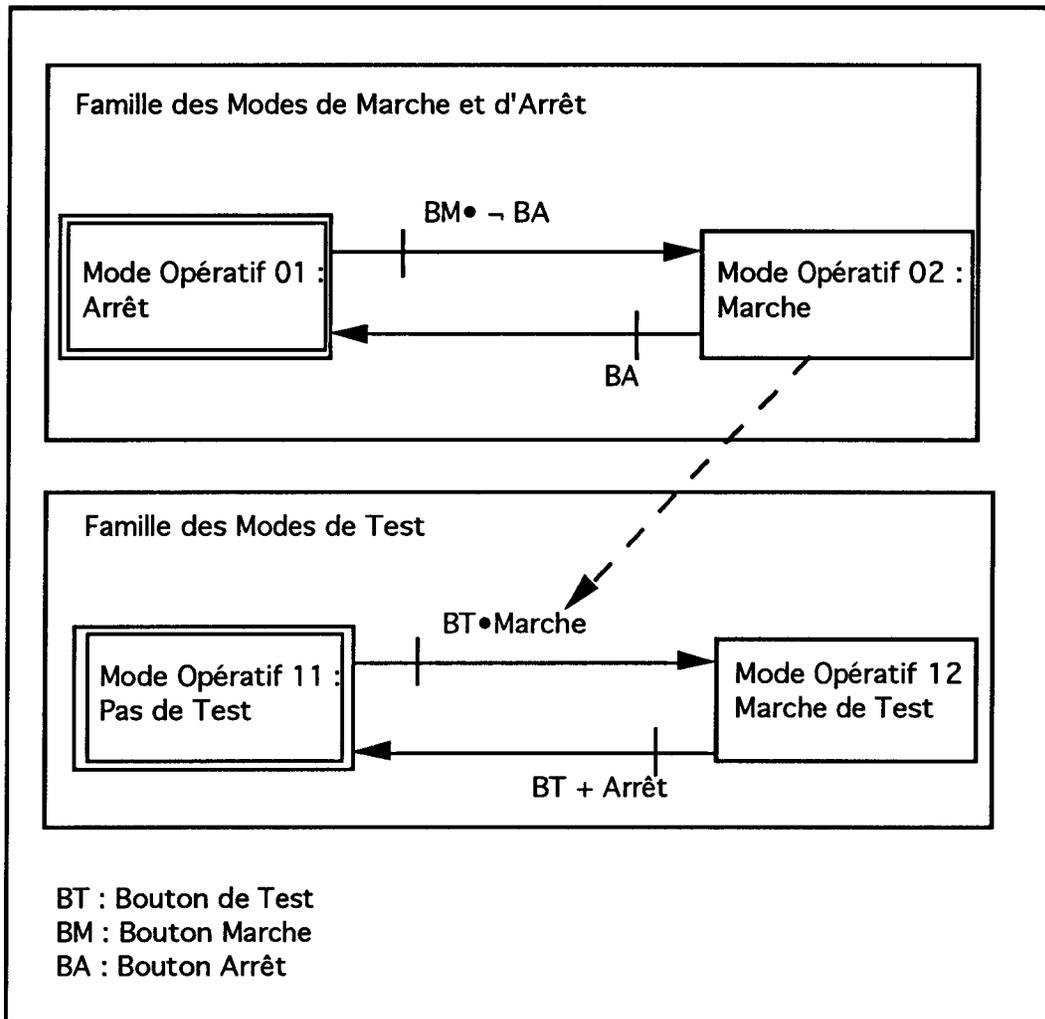
Le LAI de l'ISMCM [Anakök, 87] introduit les notions de modes opératifs inférieurs et de modes opératifs supérieurs. Un mode opératif inférieur exprime la marche, l'arrêt ou le paramètre de fonctionnement d'un ensemble d'opérateurs appartenant à une organisation. Dans l'exemple de la figure IV.6, les étapes du Grafset représentent les modes opératifs inférieurs. L'évolution d'un mode à l'autre (réceptivité) est conditionnée par les informations que fournit le Niveau Inférieur du SAP.

Un mode opératif supérieur exprime un objectif global du SAP : choix ou mise en activité d'une organisation. Il est défini au Niveau Supérieur de la Partie Commande (PC).

Une PC multi-organisations, multifonctions et multitâches apparaît donc nécessaire. Mais cette PC à plusieurs degrés de liberté doit se comporter comme un tout cohérent vis-à-vis de l'extérieur. Il s'agit de garantir l'unicité (au sens de cohérence et de non-contradiction) de ses objectifs. La notion de famille de modes inspirée du GEMMA peut répondre à cet impératif.

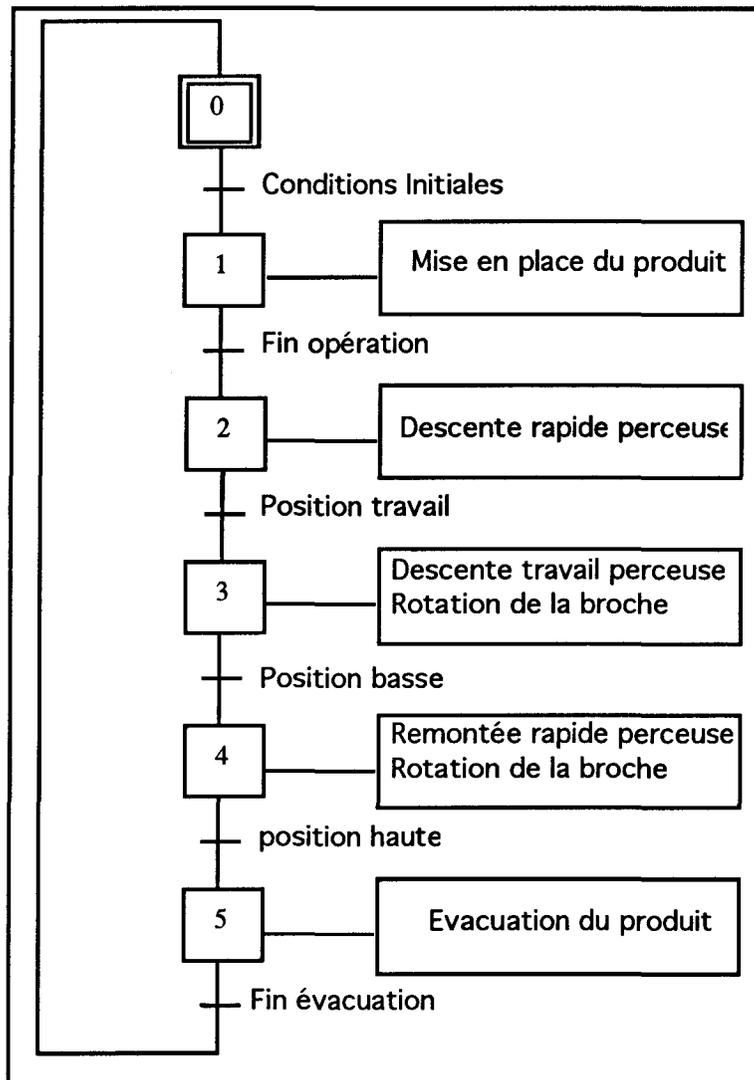
Une Famille de Modes est vue comme un ensemble dont les éléments commandent et qualifient les mêmes organes opératifs, selon un même critère ; chaque élément proposant une organisation différente. Un SAP, en fonction de ses spécifications et de sa complexité peut avoir une ou plusieurs Familles de Modes. Les critères de regroupement ne sont pas universels et varient d'un système à un autre.

La situation d'une Famille de modes dépend parfois de celle d'une autre : il existe donc des liens structurels et hiérarchiques entre elles. En cas de dépendance entre familles de modes, la Famille supérieure commande et "force la situation des autres et les familles inférieurs "suivent" ou "observent" leur supérieure. Ainsi, dans l'exemple de la figure IV.5, la Marche de Test ne peut être obtenue et maintenue que si le SAP est en Marche.

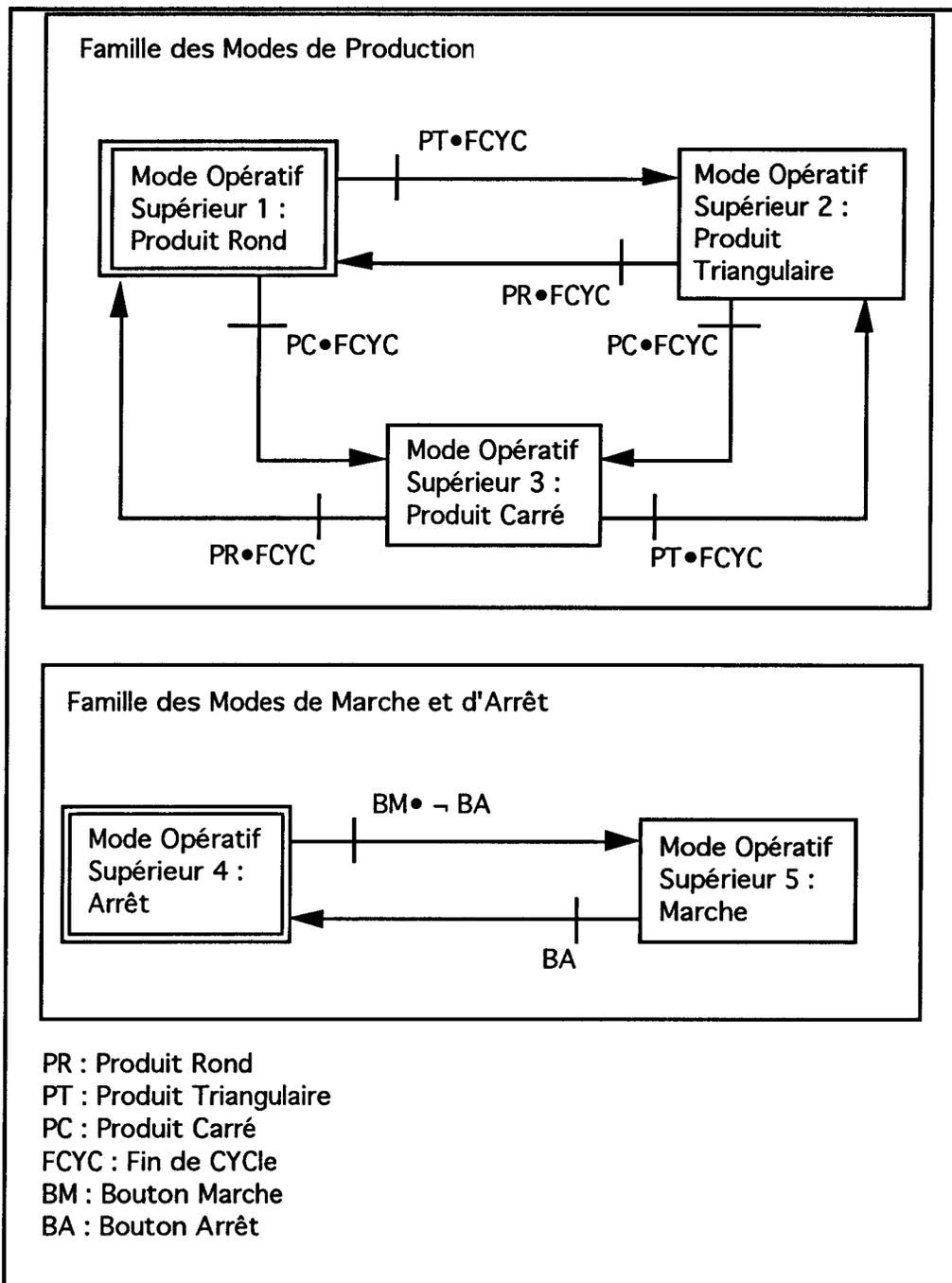


*Figure IV.5 : Hiérarchie Structurelle entre les familles de Modes*

Les familles de modes permettent de garantir l'unicité de modes du SAP. Il s'agit de traiter les états conflictuels en hiérarchisant les décisions et les actions et non les activités. C'est, notamment, le cas des modes opératifs appartenant à des familles distinctes et sélectionnés simultanément. Yéva Anakök attribue à chaque centre de décision une partie opérative virtuelle. Tous commandent ainsi leur partie et un système d'arbitrage réalise la synthèse des actions virtuelles en établissant des relations logiques ou hiérarchiques entre les différentes décisions concernant un même organe opératif.



*Figure IV.6 : Une Organisation du centre d'usinage : perçage simple*



*Figure IV.7 : Exemple de Familles de Modes*

#### IV.2.1. Limites

Cette approche des Systèmes Automatisés de Production (SAP) est fondée sur la connaissance fonctionnelle des machines de production, de leurs produits et de leur environnement dans le cadre d'un certain nombre d'hypothèses restrictives. En effet, les SAP considérés sont limités à une machine ou à une unité-machine (un ensemble de postes unis par le transfert synchrone des produits). En outre, les moyens de production sont analysés dans une perspective de commande au

sens large sans se préoccuper ni des méthodes d'élaboration des produits et de choix des moyens de production ni des problèmes liés à la défaillance des organes opératifs du SAP.

En outre, l'analyse par des "objectifs systèmes" suppose implicitement que l'ensemble des interactions entre le système et les produits, ainsi que les modes opératoires sont prédéterminés. Elle néglige les liens de coopération et de concurrence entre les moyens de production. Pourtant, ces liens peuvent mettre en cause la définition des fonctions et même parfois, celle des objectifs du SFPM.

#### **IV.3. L'approche du Laboratoire de Génie Industriel et Logiciel de l'Université de Valenciennes et du Hainaut-Cambrésis : le MESAP [Parayre, 92]**

Les objectifs du MESAP sont d'intégrer pour un SAP :

- Le système de commande, avec les modes de programmation, de réglage, de maintenance, de diagnostic, de marche automatique, pas à pas et cycle par cycle.
- Le système de communication, avec les modes "En ligne", "Hors ligne" et "En défaut".
- Le système de contrôle/capteurs intelligents qui ont leur propre fonctionnement avec les modes de non-fonctionnement, de surveillance et de diagnostic.
- La partie opérative, avec les modes "à l'arrêt", "en fonctionnement" et "en défaillance".
- La production pour les modules de conduite d'atelier, avec les modes "En production", "Hors production" et "Hors délais".

La modélisation d'un SAP par MESAP introduit trois niveaux d'abstraction :

- le niveau Pièce qui décrit le flux de pièces circulant dans la cellule ainsi que leurs gammes opératoires (telles qu'elles sont définies au LAIL [Amar, 94]),
- le niveau Composant qui décrit tous les composants (constituants matériels ou logiciels de base) de la cellule sans aucune hiérarchisation,
- le niveau Application qui décrit tous les logiciels. Une application tient compte de la tâche qui lui a été impartie lors de la phase de conception, des modes de marche du composant sur lequel elle a été implantée et des différents éléments qu'elle peut superviser.

Le niveau pièce est modélisé avec le formalisme "extended SADT" et les niveaux Composant et Application sont structurés en Unités Fonctionnelles modélisées avec des Réseaux de Petri Interprétés et Colorés. Ces UF décrivent le comportement des modes de marche d'une partie du

composant ou de l'application en distinguant différentes familles de modes de marche (Arrêt, Fonctionnement, Défaillance...).

#### **IV.3.1. Limites**

La proposition MESAP ne propose pas de solution pour obtenir une structuration correcte d'un SFPM. En outre, elle considère que les modes de marche d'une application ne sont pas directement fonction des modes de marche des applications de niveaux inférieurs.

#### **IV.4. Approche du Laboratoire d'Automatique de Nantes [Biland, 94]**

Le LAN [Biland, 94] propose une modélisation du comportement d'un Système Automatisé de Production du point de vue des modes de marche d'une fonction donnée de ce SAP sous la forme d'un automate fini déterministe. Il est possible de regrouper ces SAP modélisés par des automates finis déterministes pour obtenir un SAP composé. Compte tenu des interactions technologiques et des contraintes de fonctionnement, une succession d'opérations formelles permet de calculer l'automate fini et déterministe du comportement constaté du SAP composé. Il est alors possible de faire une abstraction de ce comportement constaté pour offrir au niveau supérieur une vision correspondant aux besoins de ce niveau pour remplir son rôle dans la fonction considérée (commande, maintenance, supervision, ...).

L'approche du LAN propose un outil de manipulation d'automates finis et déterministes. Cette approche souffre de l'absence d'éléments méthodologique, notamment pour la partition des états du système agrégé. En effet, il n'aide pas le concepteur à définir et à représenter les interactions et les contraintes de fonctionnement entre les SAP. Le concepteur n'est pas non plus aidé dans la détermination de la vision que peut avoir un niveau  $i+1$  du regroupement, au niveau  $i$ , des SAP du niveau  $i-1$ .

#### **IV.5. L'approche du LAIL**

##### **IV.5.1. L'équipe ASPIC**

L'équipe ASPIC se place dans un cadre plus général que celui des SAP pour proposer une modélisation d'un système sous la forme d'objet communicants [Staroswiecki, 93], [Bayart, 94]. Une modélisation générique de ces objets leur assure des qualités d'interopérabilité et d'interchangeabilité. D'un point de vue externe, un objet assure un certain nombre de services dont on peut demander l'exécution à l'aide de requêtes.

Un objet présente plusieurs modes d'utilisation. Un **mode d'utilisation** est un sous ensemble des services offerts par l'objet. Un mode d'utilisation constitue un moyen de structuration du fonctionnement d'un instrument intelligent, car il sépare l'ensemble des services en trois classes :

- Les services qui appartiennent au contexte du mode en cours qui sont systématiquement et continuellement exécutés tant que l'on reste dans ce mode.
- Les services qui peuvent être demandés au moyen de requêtes.
- Les services qui n'appartiennent pas au mode en cours et qui ne peuvent pas être sollicités.

La réalisation d'un service nécessite l'utilisation de ressources qui peuvent être disponibles ou non. Il y aura donc plusieurs états du service (nominal, dégradé ...) suivant l'état des ressources, on parle alors du mode de marche d'un service. L'équipe ASPIC appelle **mode de marche** d'un instrument intelligent la combinaison d'un mode d'utilisation (voulu) et d'un état (subi). Cette combinaison définit la liste des services disponibles à chaque instant ainsi que leur version (nominale, dégradée, ...) [Staroswiecki et al, 95].

Enfin, le modèle externe d'instrument intelligent que propose l'équipe ASPIC permet la construction d'instruments complexes par agrégation d'instruments de niveau inférieur. Ce modèle ramène le passage d'un niveau à l'autre à la définition d'opérateurs de construction des modules de gestion des modes d'utilisation et des modes de marche qui structurent les services du niveau supérieur [Bayart et al, 95].

L'approche de l'équipe ASPIC est très intéressante par la structuration et la rigueur qu'elle propose pour appréhender la notion de mode de marche. Il manque cependant une définition claire des contraintes de fonctionnement à intégrer afin de construire le gestionnaire des modes d'un instrument intelligent complexe. En effet, on ne peut pas, à notre avis, réduire le problème de la gestion des modes à de simples opérateurs d'agrégation d'automates.

#### IV.5.2. L'équipe PFM

##### Introduction à CASPAIM [Bourey, 93]. [Craye, 94]

Le projet CASPAIM a pour objet de définir une méthode et des outils homogènes d'aide à la conception et à la réalisation des systèmes de commande (commande directe, supervision ...) des Systèmes Flexible de Production Manufacturière. La méthode de conception proposée s'articule autour de 2 axes :

- Définir les moyens nécessaires à l'analyse, la conception et l'évaluation du système de commande à partir de la spécification des produits à fabriquer.
- Proposer une architecture matérielle de l'atelier à partir des moyens potentiels ou existants.

La démarche de conception, qui sera plus amplement présentée dans le chapitre suivant, est

modulaire, progressive et assistée. Elle s'inscrit dans le cadre du Génie Automatique [Verdin, 91], [Morel, 92] et propose des outils :

- de conception de la commande qui intègrent la notion de sûreté de fonctionnement
- d'analyse de l'architecture matérielle et d'évaluation des solutions
- d'ordonnement et de pilotage temps réel

#### Position du problème de la Gestion des Modes dans CASPAIM

La conception initiale [Bourey, 88], [Kapusta, 88], la simulation [Castelain, 87] et l'implantation [Craye, 89] de la commande ont été l'objectif de l'équipe PFM. Cependant, si les Réseaux de Petri (RdP) permettent de bien modéliser la commande du système de production en fonctionnement normal, de nombreux problèmes subsistent lorsqu'il faut tenir compte des contraintes réelles de l'environnement de production.

L'équipe PFM du LAIL a élargi son champ d'étude en s'intéressant à d'autres domaines de la conduite tels que la gestion de production, la surveillance et en particulier, les aspects liés à la gestion des modes de marche.

La gestion des modes de marche correspond au pilotage effectif des ressources de production en prenant en compte leurs différents états possibles. La modification de ces états peut tout aussi bien provenir de la commande d'un opérateur qu'être la conséquence d'une perturbation. Dans ces deux cas, la difficulté de mise en œuvre est due au caractère interdépendant des ressources de production. En effet, une ressource de production n'est pas une entité isolée ; au contraire, tout changement d'état se répercute nécessairement sur la configuration globale de la cellule de production. Notre objectif est de proposer un gestionnaire des modes d'exploitation d'un SFPM à travers une modélisation qui représente les ressources en mettant en avant leurs interactions mutuelles.

Nos travaux tendent à améliorer et à étendre les concepts du Gemma repris notamment dans la thèse de S. Bois dont nous rappelons les principales orientations.

#### **IV.5.3. La gestion des modes de marche dans CASPAIM [Bois, 91]**

L'approche de S. BOIS se situe dans le contexte de l'intégration des fonctions de la commande d'un SFPM : surveillance, commande de coordination etc... Elle est fondée sur la notion de contrainte opératoire entre les ressources de production connexes. Ces contraintes entraînent des dépendances entre les modes de marche et d'arrêt de ces ressources. S. Bois propose une analyse ascendante à partir des machines et de leurs états. Il utilise le GEMMA pour modéliser le comportement des moyens de production.

Modélisation du comportement des moyens de production.

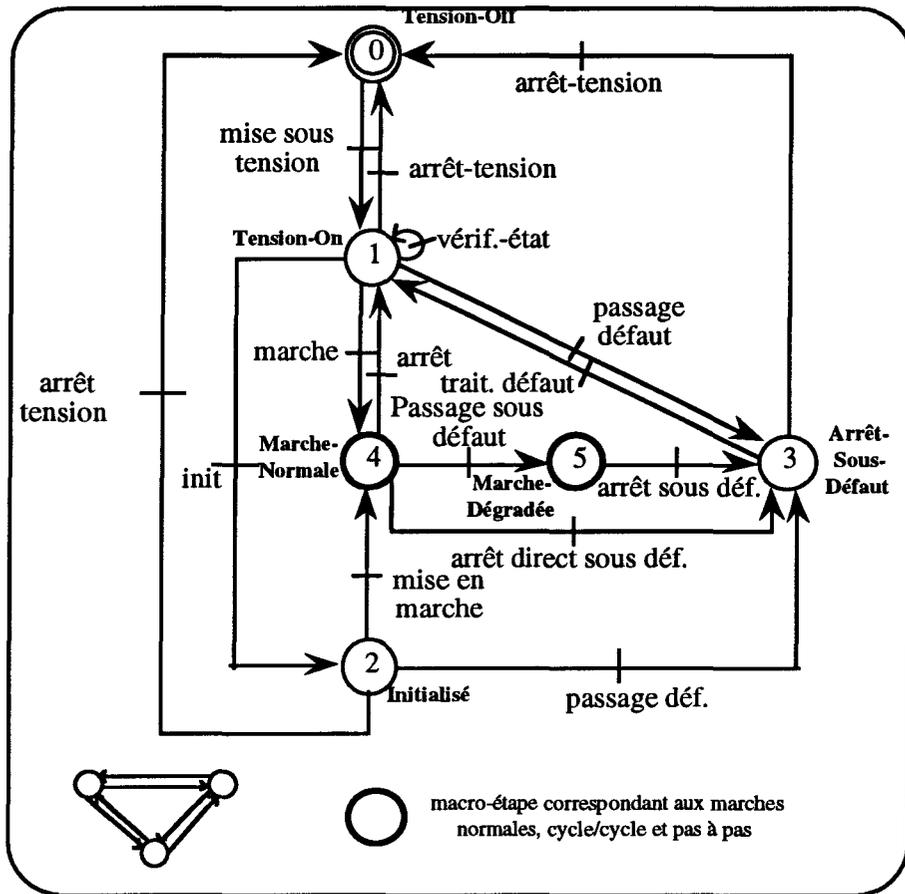
On distingue deux types de machines : les **machines effectives** et les **machines virtuelles**.

Une machine effective est une entité matérielle commandable : butée, aiguillage, convoyeur, machine outil, robot etc.... Le comportement de chaque machine effective est représenté par un graphe d'état. Le graphe comporte six états dans le cas le plus complexe :

- Etat 0 ou Tension-Off : l'entité est hors énergie. C'est l'état initial. Il est représenté par un double cercle.
- Etat 1 ou Tension-On : il correspond à une mise sous tension de l'entité en dehors de toute autre commande.
- État 2 ou Initialisé : l'entité a été configurée (vérification, marche de préparation, etc.) et est prête à passer en marche normale.
- Etat 4 ou Marche Normale : c'est le mode de fonctionnement normal de l'entité.
- Etat 5 ou Marche Dégradée : une défaillance a été détectée. On poursuit la production en mode dégradé.
- Etat 3 ou Arrêt sous-défaut : la défaillance détectée est assez importante pour que l'on arrête la machine dans une position de repli du type arrêt des avances avec gel de la commande. Notons qu'à ce stade la machine est toujours sous tension.

Les états 4 et 5 sont des macros-états ( i.e., composés de plusieurs sous-états ).

La figure IV.8 illustre les différents états ainsi que les transitions qui les relient. Ce graphe concerne le comportement d'une machine effective complexe.

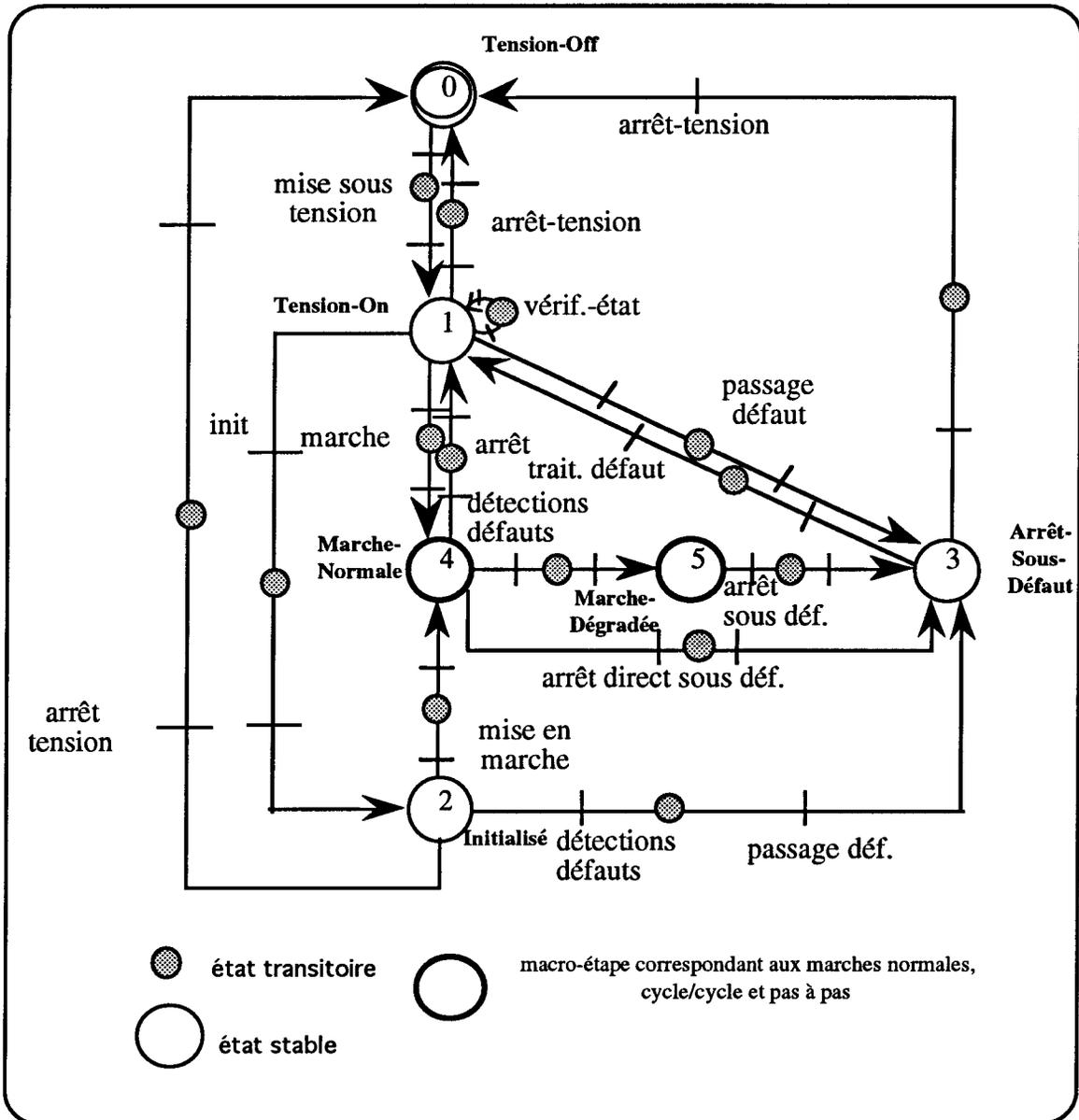


*Figure IV.8 : graphe de comportement d'une machine effective complexe*

Une machine virtuelle est un regroupement de machines virtuelles ou effectives. Une machine virtuelle est construite en regroupant des machines dont les comportements sont liés par des contraintes de fonctionnement.

Une machine virtuelle est composée de sous-machines. Elle en hérite les comportements. L'évolution de son état dépend directement de l'état de ses machines composantes. L'idée directrice du regroupement de machines effectives en machines virtuelles est d'assurer la cohérence des états des sous-machines. La structure de base du graphe de comportement d'une machine virtuelle est la même que celle des machines effectives. L'expression des contraintes permettant d'associer des machines en machines virtuelles détermine les règles de composition d'état. L'évolution de l'état d'un système peut être la conséquence de l'évolution du processus qui modifie l'état d'une ou plusieurs machines. Le nouvel état de la machine virtuelle est alors automatiquement calculé. L'évolution de l'état du système peut aussi être la conséquence d'une commande de l'opérateur qui souhaite faire évoluer l'état de la machine virtuelle. Il faut alors amener les machines composantes dans des états compatibles. Pour tenir compte des modifications pouvant survenir individuellement sur les sous-machines, il est nécessaire d'introduire dans le comportement de base, des états transitoires appelés états instables.

Une machine virtuelle se trouve dans un état dit "instable", si une, au moins, de ses sous-machines se trouve dans un état incompatible avec celui des autres sous-machines ou lorsque la conformité des états de ses sous-machines n'est pas vérifiée. Il s'agit alors d'effectuer un test de cohérence des états dépendants. Un état instable est en fait un état transitoire entre deux états stables du comportement de base. La figure IV.9 illustre ces états transitoires. Cette représentation devient inutile si on adopte le formalisme des réseaux de Petri T-temporisés.



*Figure IV.9 : graphe de comportement d'une machine virtuelle.*

**CONCLUSION**

Le modèle du système de contrôle/commande, présenté dans cette première partie de notre mémoire, montre l'importance d'une approche globale et cohérente au sens de l'intégration de la conception des SFPM. Les liens entre la supervision et le pilotage dans le cas du recouvrement d'erreurs et la problématique de la cohérence des informations et des modèles en sont deux exemples significatifs.

Les principales approches proposées par la communauté française pour gérer les modes des systèmes automatisés de production sont rappelées dans le tableau ci-après. Seule l'approche développée dans [Bois, 91] propose un gestionnaire des modes de marche et d'arrêt d'un SAP composés. Les autres approches proposent des outils de représentation des modes par composition d'automates sans véritablement donner de méthode pour spécifier ces modes. L'approche de l'équipe ASPIC du LAIL est particulière. Elle ne figure pas dans ce tableau, car elle a surtout contribué à formaliser les notions de modes de marche dans le cas des capteurs intelligents qui, à la différence des SAP, ne sont pas commandables.

Approche	Structuration d'un SAP unitaire	Structuration d'un SAP composé (définition des contraintes)
[ADEPA, 79 et 92]	oui	non
[Anakök, 87]	oui	non
[Bois, 91]	oui	oui
[Parayre, 92]	oui	non
[Bilan, 94]	oui	non

***Les principales approches françaises de la gestion des modes de marche et d'arrêt***

Nous avons développé la problématique de la supervision et de la gestion des modes en étendant les concepts du GEMMA. En effet, il ne s'agit plus de gérer la marche et l'arrêt d'un système automatisé de production unitaire mais de gérer les configurations d'un Système Flexible de Production composé et complexe. En effet, la flexibilité permet de réagir à d'éventuelle défaillance des ressources de production par une reconfiguration du système. Cette adaptabilité n'a d'intérêt que si elle reste exploitable par la supervision et le pilotage.

Nous allons développer dans la suite de ce mémoire :

- Une représentation structuro-fonctionnelle, seule capable de garantir la cohérence des informations et des modèles utilisés par la conduite d'un SFPM.
- Un gestionnaire de modes fondé sur des approches et des outils synchrones.

## **PARTIE B : MODÉLISATION DES SFPM ET GESTION DES MODES**

Chapitre V : Présentation de CASPAIM [Bourey, 93], [Amar, 94] et [Craye, 94]

Chapitre VI : La supervision dans CASPAIM

Chapitre VII : Exigence d'une approche fonctionnelle pour la gestion des modes

Chapitre VIII : Représentation fonctionnelle pour la gestion des modes d'un SFPM

## INTRODUCTION

La modélisation est le cœur du système d'information d'un SFPM. La difficulté de la modélisation réside dans son caractère multi-niveaux et multi-facettes. Chaque composante de la commande d'un SFPM a sa propre représentation de l'atelier. De plus la hiérarchisation de la commande impose plusieurs niveaux d'agrégation et d'abstraction du SFPM.

Après une présentation générale du projet CASPAIM dans lequel s'inscrit notre travail, cette deuxième partie présente les approches de modélisation des SFPM adoptées par l'équipe PFM du LAIL. Nous décrivons les modèles de la commande de coordination ainsi que les modèles de la surveillance. Nous présenterons plus longuement le modèle proposé dans [Bois, 91] pour la gestion de la marche et de l'arrêt des machines. Nous montrerons que cette approche est limitée, d'une part, parce qu'elle ne prend en compte qu'une catégorie restreinte de SFPM et d'autre part, parce qu'elle n'a pas pour objectif la résolution des problèmes de reconfiguration des SFPM. Elle se contente d'assurer la cohérence des commandes effectives de marche et d'arrêt des ressources de production.

Nous proposons dans cette partie une approche fonctionnelle (par les fonctions) de la modélisation des SFPM. Nous montrerons qu'une telle approche est plus adaptée qu'une approche purement structurelle (par la structure et les machines) pour gérer les modes de fonctionnement et les configurations des SFPM. Nous proposerons une représentation arborescente que nous appellerons arbre COD pour Chargement, Opération et Déchargement.

## **CHAPITRE V : PRÉSENTATION DE CASPAIM [BOUREY, 93], [AMAR, 94] ET [CRAYE, 94]**

D'après [Anakök, 87], une approche par des “objectifs produits” et une approche par des “objectifs systèmes” sont nécessaires pour appréhender efficacement un système automatisé de production dans sa globalité. L'approche par des “objectifs produits” permet de définir la structure fonctionnelle du système de production, en partant des caractéristiques du produit et de sa gamme de fabrication [Amar, 94]. L'approche par des “objectifs systèmes” permet de définir des modes d'exploitation et de fonctionnement normaux ou dégradés du système de production en décrivant les procédures de passage d'un mode à un autre.

C'est ainsi que cette approche système a été exploitée dans CASPAIM pour définir une organisation et une gestion des modes de marche [Bois, 91], [Kermad et al, 93a]. Elle est fondée sur une représentation structurelle du système de production. Nos travaux, prenant en compte la dimension fonctionnelle, intègrent, dès la conception, les différents modes de fonctionnement ainsi que leur gestion en précisant les liens avec le système de coordination en terme de réorganisation des graphes de commande et de coordination des ressources.

Ce chapitre va décrire les principales étapes de conception dans le projet CASPAIM (figure V.1) afin de mieux situer la place de la supervision d'une manière générale et la position de nos travaux dans la démarche générale. Le lecteur qui souhaite plus de précisions sur les travaux développés dans le cadre du projet CASPAIM peut se référer à l'annexe, jointe à ce mémoire.

### **V.1. Phases de spécification**

#### **V.1.1. Spécification de la partie logique : les objectifs de production**

Cette étape concerne la description formelle des objectifs de production du point de vue des produits. Cette description est indépendante des moyens de production et de l'architecture du système. Elle aboutit à la définition fonctionnelle des processus de fabrication des produits. Chaque processus est représenté par une Gamme Logique avec un graphe de type Réseau de Petri (RdP) [Cruette, 91].

Les gammes logiques permettent de prendre en compte le séquençement élémentaire des opérations caractéristiques (transformation élémentaire qui engendre un changement d'état du produit) appliquées sur chaque type de produit en mettant en évidence les possibilités de flexibilité dans l'ordre de ces opérations.

#### **V.1.2. Spécification de la partie physique : les moyens de production**

La spécification de la partie physique (ou spécification opérationnelle) [Amar, 94] va décrire les différentes ressources de production (fabrication, transport, manipulation,...) du procédé

indépendamment de la partie logique qui est orientée produit. Cette phase comporte trois étapes : l'élaboration de la bibliothèque fonctionnelle, l'analyse structuro-fonctionnelle et la caractérisation de l'architecture du système de production.

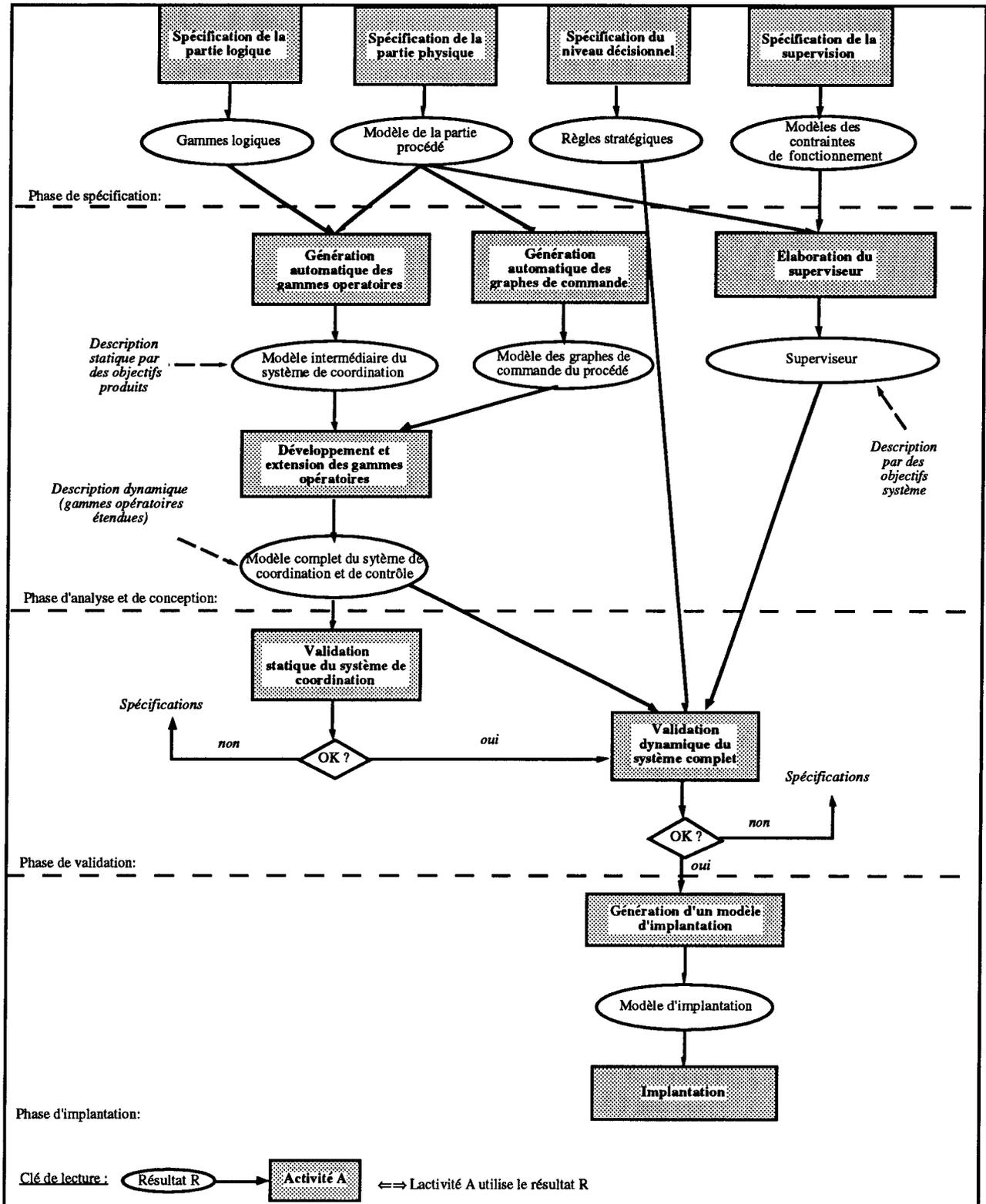


Figure V.1 : Démarche de conception dans CASPAIM

### **V.1.3. Spécification de la partie décisionnelle : les règles stratégiques**

Cette étape, traitée dans [Hammadi, 91] et [Tawegoum, 94], consiste à élaborer et à décrire des stratégies de pilotage du système pour lui conférer un comportement autonome par résolution des conflits et déterministe par résolution des problèmes de routage temps réel des produits. En outre, l'étude du niveau décisionnel vise à optimiser les flux de production en recherchant à la fois la meilleure façon d'insérer les entrées et d'opérer simultanément la résolution des conflits d'accès [Ohl et al, 94], [Camus et al, 94]. La démarche considérée s'appuie sur la notion de gammes opératoires et intervient très tôt dans la démarche globale de conception. Elle concourt ainsi à l'aspect prototypage rapide et vise une fois de plus, dès les phases amonts de spécification, à assurer un premier niveau de validation.

### **V.1.4. Spécification de la supervision : la surveillance et la gestion des modes**

La spécification de la supervision permet de définir les types de surveillance et de gestion de modes. Cette spécification commence par la description des grandeurs à surveiller et la définition de l'observabilité du SFPM. Elle permet au concepteur de préciser les objectifs globaux de production en précisant à chaque fois l'organisation du SFPM et les contraintes de fonctionnement entre les ressources de production. C'est aussi l'occasion de préciser les seuils définissant le taux de dégradation d'une ressource ainsi que les paramètres à prendre en compte pour apprécier l'état du SFPM dans une configuration donnée.

La conception de la surveillance au L.A.I.L. a pris deux orientations complémentaires que nous présenterons par la suite. Nous verrons en outre, que L'approche de la gestion des modes d'un SFPM proposée dans CASPAIM s'appuie sur le GEMMA. Nous montrerons que cette démarche ne suffit pas pour modéliser des systèmes où des ressources complexes côtoient des ressources simples. Nous montrerons ensuite, que cette approche ne permet pas l'exploitation de toute la flexibilité des SFPM.

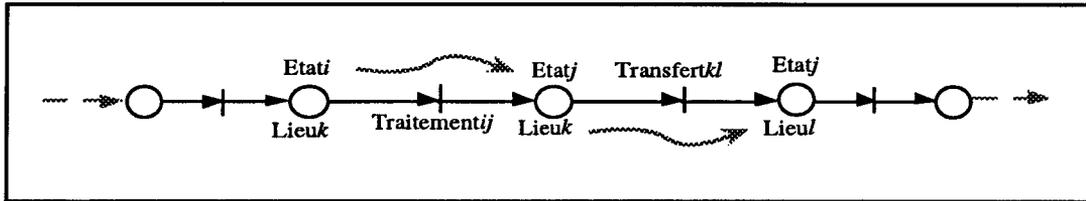
## **V.2. Phase d'analyse et de conception**

Les spécifications des parties logique et physique permettent le développement des différents modèles du système de commande. Elles permettent en particulier, la génération automatique des gammes opératoires [Amar, 94] et la conception des modèles de comportement et de commande de la partie opérative [Amar, 94] et [Huvenoit, 94].

Ces spécifications aboutissent l'élaboration d'une phase de "prototypage" qui permet d'implanter des gammes opératoires. Les validations statiques et dynamiques de ces gammes opératoires vont permettre de vérifier les propriétés du système modélisé.

### V.2.1. Génération automatique des Gammes Opératoires [Amar, 94]

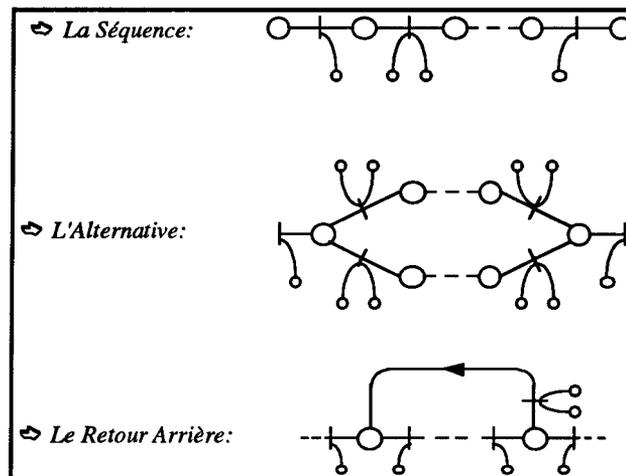
La phase de génération des gammes opératoires permet de développer les différentes gammes logiques en tenant compte notamment de l'aspect transitique de la partie physique. Sur le graphe RdP d'une gamme opératoire, chaque **place** représente, en plus de l'état fonctionnel hérité des gammes logiques, l'état positionnel d'un produit. Chaque **transition** représente soit un **traitement** (changement d'état), soit un **transfert** (changement de lieu) (figure V.2).



*Figure V.2 : Description d'une gamme opératoire*

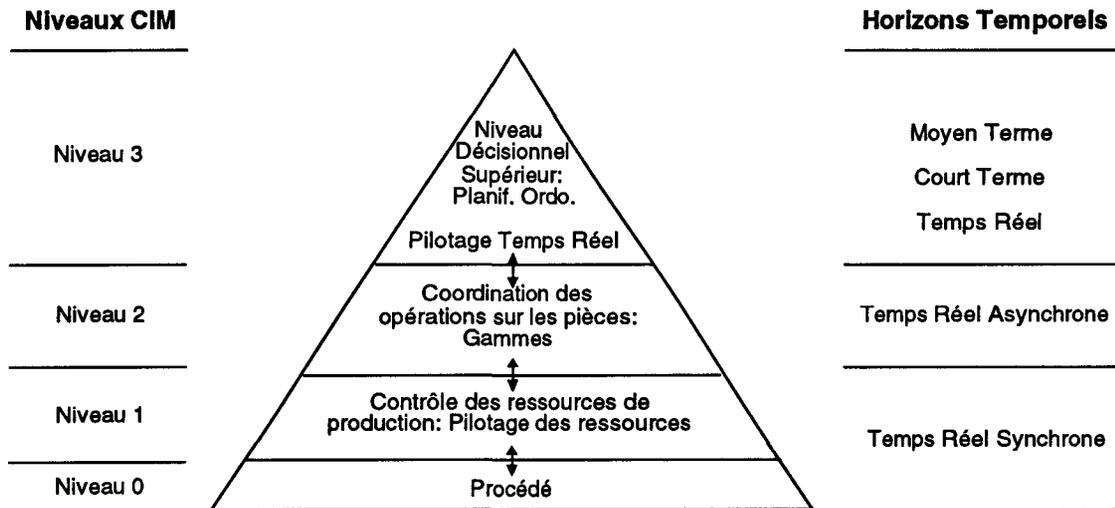
#### Développement et extension des gammes opératoires

Les gammes opératoires décrivent le parcours des produits à travers le système de production. Les gammes opératoires étendues sont composées de trois types de structures de base (figure V.3) : la séquence qui correspond à la structure minimale, l'alternative qui correspond à un indéterminisme (choix entre différentes ressources de production ou flexibilité entre différentes opérations) et le retour arrière nécessaire dans le cas d'une stratégie de contrôle particulière ou de rejets d'allocations (Nack).



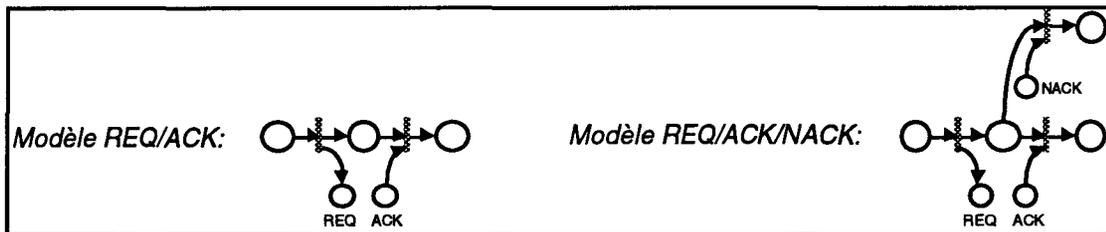
*Figure V.3 : Structure des gammes opératoires étendues*

Ces modèles caractérisent la partie coordination du contrôle/commande global d'un SFPM. Ils sont intermédiaires entre le Niveau Hiérarchique et le pilotage direct des ressources de production (figure V.4).



*Figure V.4 : Hiérarchisation de la commande*

En effet, les gammes opératoires communiquent avec le niveau hiérarchique supérieur afin de résoudre leurs indéterminismes et émettent des requêtes vers les modèles des ressources opératives. Ces requêtes vont caractériser en particulier la nature du traitement à réaliser sur un type de produit donné ; elles permettront donc de lancer le programme adéquat de pilotage. Ces échanges inter-niveaux sont réalisés par des protocoles de communication qui respecteront la structure hiérarchique de la commande et permettront une distribution du code dans un environnement réparti et hétérogène (figure V.5).



*Figure V.5 : Modèle d'échange au sein des gammes étendues*

### V.2.2. Construction des graphes de contrôle (ou de commande) des ressources [Huvenoit, 94]

La description structuro-fonctionnelle proposée dans [Amar, 94] distingue deux types de ressources : les ressources simples et les ressources complexes.

#### Les ressources simples

Une ressource simple ne subit aucun indéterminisme de fonctionnement. Sa commande sélectionne une séquence opératoire d'actions élémentaires en fonction de la nature de la ressource :

Les graphes de commande des ressources simples sont obtenus à partir d'une bibliothèque de graphes de commande partiels construits avec deux primitives de base : la séquence et l'alternative. La synchronisation éventuelle est assurée par le couple requête et accusé de réception de la figure I.5. La notion de ressource simple est à rapprocher de la notion de machine effective [Bois, 91].

### Les ressources complexes

Le graphe de commande ne suffit plus pour caractériser une ressource complexe ; il est nécessaire de disposer en plus, d'un modèle (ou filtre) comportemental qui garantit la cohérence des diverses sollicitations avec l'état global de la ressource. Le modèle comportemental décrit la ressource et sa dynamique d'évolution. Il peut inclure un résolveur d'indéterminismes capable de gérer la flexibilité interne de la ressource. Une ressource complexe est une machine virtuelle [Bois, 91] dans la mesure où elle peut être considérée comme un regroupement de ressources simples.

### V.3. Phase de validation

L'évaluation des performances du modèle permet de vérifier le respect des spécifications quantitatives. Une approche qualitative des gammes opératoires associées aux modèles de la partie opérative peut déterminer la "robustesse" ou la tolérance aux fautes du système. L'occurrence d'une perturbation (une relation d'accessibilité disparaît, une ressource de production n'est plus opérationnelle, ...) n'empêche pas le système d'assurer une certaine production en changeant de mode de marche ou de production. Ce prototypage, par un retour arrière rapide, peut aider à l'élaboration de l'architecture physique de production.

La validation dans CASPAIM repose sur deux démarches : une validation statique des gammes et une validation dynamique du système.

#### **V.3.1. Validation statique des gammes**

La validation statique des gammes est principalement :

- une démarche d'analyse des propriétés de terminaison propre et de vivacité du modèle de description des gammes afin d'assurer que toute gamme pourra être effectivement réalisée et que toutes les variantes prévues seront possibles [Cruette, 91],

- une méthode de recherche des blocages liés à une saturation du système de transport [Ausfelder, 94] ou d'une étreinte fatale liée à des utilisations croisées de ressources.

### V.3.2. Validation dynamique du système

Deux approches sont possibles pour valider dynamiquement une solution : les démarches analytiques et les démarches de simulation.

Seule l'approche par simulation a été mise en œuvre complètement [Ausfelder, 94]. Certains travaux sont en cours [Ohl et al, 94] pour intégrer une démarche analytique basée sur les Réseaux de Petri Stochastiques ou les Réseaux de Files d'Attente dans un cadre d'évaluation des performances.

L'approche analytique permet d'effectuer des validations préliminaires et intermédiaires du modèle de coordination (gammes) en cours de développement. Le développement progressif des gammes et la représentation en niveaux devraient favoriser cette approche.

### V.4. Phase d'implantation

Les premiers travaux de CASPAIM concernant l'implantation [Craye, 89] sont aujourd'hui totalement remodelés pour tenir compte des profondes modifications apportées au projet CASPAIM depuis 1991.

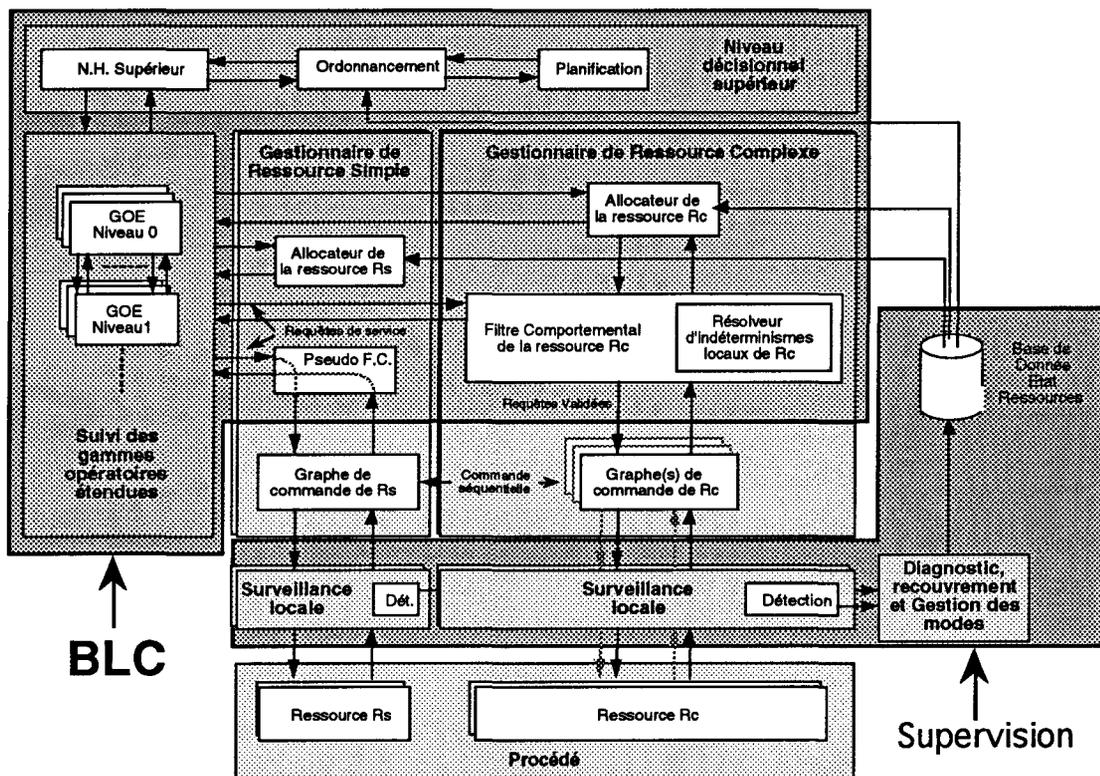


Figure V.6 : Architecture de contrôle d'un SFPM [Huvenoit, 94]

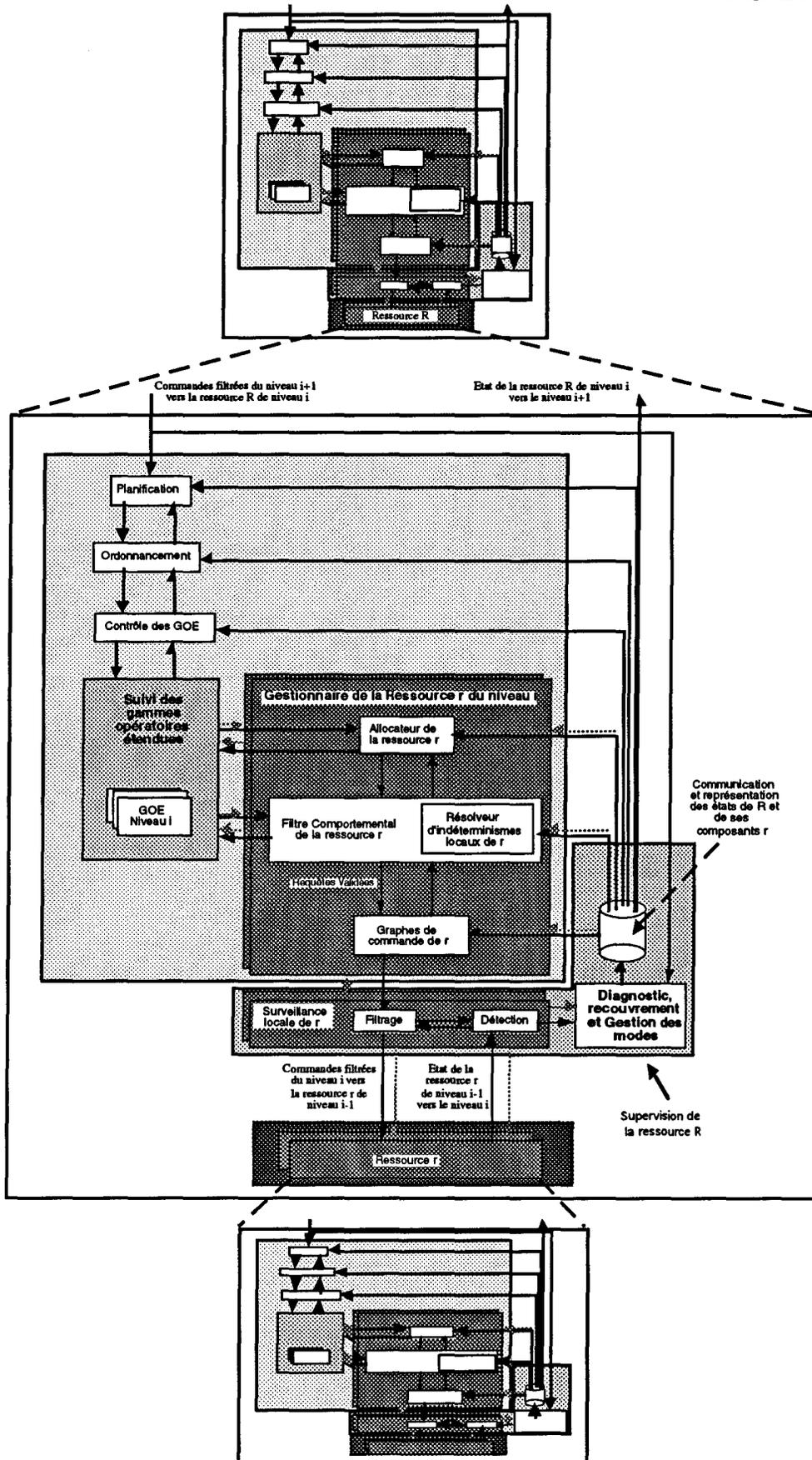


Figure V.6.b : Architecture hiérarchisée de contrôle d'un SFPM

[Huvenoit, 94] a proposé une approche d'implantation basée en grande partie sur une transposition des modèles de CASPAIM en langage ADA. Cette implantation s'appuie sur l'intégration des modèles décrits précédemment qui permet d'obtenir la structure complète de la commande d'un SFPM. Nous retrouvons dans la figure V.6 [Huvenoit, 94] le Niveau Hiérarchique (NH) ou Décisionnel qui prend en compte la planification et l'ordonnancement. Le NH contrôle les différentes gammes opératoires étendues qui vont solliciter les gestionnaires de ressources. Le gestionnaire intégrera ou non un filtre comportemental suivant le type de la ressource. Des modules de surveillance locale et directe [Elkhatabi, 93] s'insèrent entre les graphes de commande et les ressources. En cas de détection d'une défaillance, le gestionnaire des modes reconfigurera les données caractéristiques de l'état des ressources. Ces données serviront ensuite aux allocateurs de ressources qui rejetteront ou mettront en attente les demandes d'allocation.

Des travaux plus récents en cours de développement utilisent HOOD pour définir une méthode et une architecture informatique de gestion/contrôle de référence pour l'implantation [Farah, 93]. Cette architecture pourra s'appuyer sur une structure hiérarchique multiniveaux de la commande. Il est nécessaire d'étendre la proposition de [Huvenoit, 94] à une structuration de la commande sur plusieurs niveaux. En effet, la figure V.6 n'est qu'un cas particulier où la commande du procédé se fait à un seul niveau. La hiérarchie considérée est une hiérarchie fonctionnelle qui décompose le processus de commande du procédé. La figure V.6.b propose une vision réellement hiérarchique qui met en évidence à la fois la hiérarchie fonctionnelle et la hiérarchie de niveau de la commande.

La figure V.6.b illustre la commande d'une ressource R à un niveau i. La ressource R est composée de plusieurs ressources r dont il s'agit de planifier, d'ordonner et de coordonner le fonctionnement. L'importance des fonctions de planification, d'ordonnancement et de gestion des ressources r dépend de la complexité de R et de la structure du SFPM que l'on choisit : R peut aussi bien être un flot de production qu'un robot de manutention.

Ce chapitre a brièvement présenté les outils de modélisation ainsi qu'une méthodologie de conception que CASPAIM propose pour appréhender la complexité des systèmes flexibles de production. L'association de ces outils et méthodes permet d'aboutir à la définition de modèles décrivant le contrôle/commande du système de production. Ces modèles concernent les produits et la définition de leur gamme de production, les ressources de production et leurs comportements mais également les stratégies de production, la conduite temps réel,...

La supervision est une partie très importante de cette architecture et de cette démarche de conception globale. Elle intervient à travers la surveillance et la gestion des modes et des configurations du SFPM. Le chapitre suivant présentera plus amplement ces deux aspects de la supervision à travers les travaux de [Elkhatabi, 93], [Toguyeni, 92] et [Bois, 91].

## **CHAPITRE VI : LA SUPERVISION DANS CASPAIM**

La supervision dans CASPAIM prend en compte deux aspects : la surveillance et la gestion des modes. L'aspect interface Homme-Machines est traité par le LAMIH de Valenciennes [Milot, 88] avec lequel le LAIL collabore dans le cadre du Groupe de Travail régional 6 du GRAISYHM.

### **VI.1. La surveillance**

CASPAIM propose deux approches complémentaires de la surveillance : [Elkhatabi, 93] propose une surveillance réactive de bas niveau qui de filtrer la commande des actionneurs et [Toguyeni, 92] propose une surveillance en ligne capable de détecter des dysfonctionnements et d'identifier les composants défectueux. Ces deux approches s'appuient sur des modèles différents.

#### **VI.1.1. La surveillance réactive [Elkhatabi, 93]**

Les modèles utilisés dans [Elkhatabi, 93] identifient l'ensemble des actions réalisables ainsi que leurs conditions de réalisation. Ils sont obtenus par une décomposition structurelle du procédé complétée par une spécification des actions de chaque composant.

Les comportements élémentaires sont définis à partir des modes d'évolution des composants dont les "Objets Commandables Élémentaires" (OCE) et les "Objets Commandables" représentent le comportement. L'intégration des contraintes fonctionnelles et de sécurité aboutit au regroupement des composants contraints. Ce regroupement donne naissance à des Composants Fonctionnels Logiques (CFL) (figure VI.1).

La surveillance est organisée autour d'un Composant Fonctionnel Logique (CFL) (figure VI.2). Les commandes sont adressées au filtre du CFL, représentant les comportements élémentaires des composants contraints. Un ordre non compatible avec l'état réel du procédé est rejeté et conduit à un gel de la commande et à une reconfiguration. La rapidité de détection et le séquençage strict des commandes permet d'éviter la propagation des erreurs. Par contre, un ordre valide sera transmis au composant physique et activera simultanément le contrôle de son exécution au sein du Sous Module de Contrôle (SMC). Ce contrôle est basé sur la notion de "chien de garde" [André, 93], [Elkhatabi, 93A].

En effet, la détection d'une défaillance gèle le système de commande du composant en cause et évite la "contagion" des autres composants.

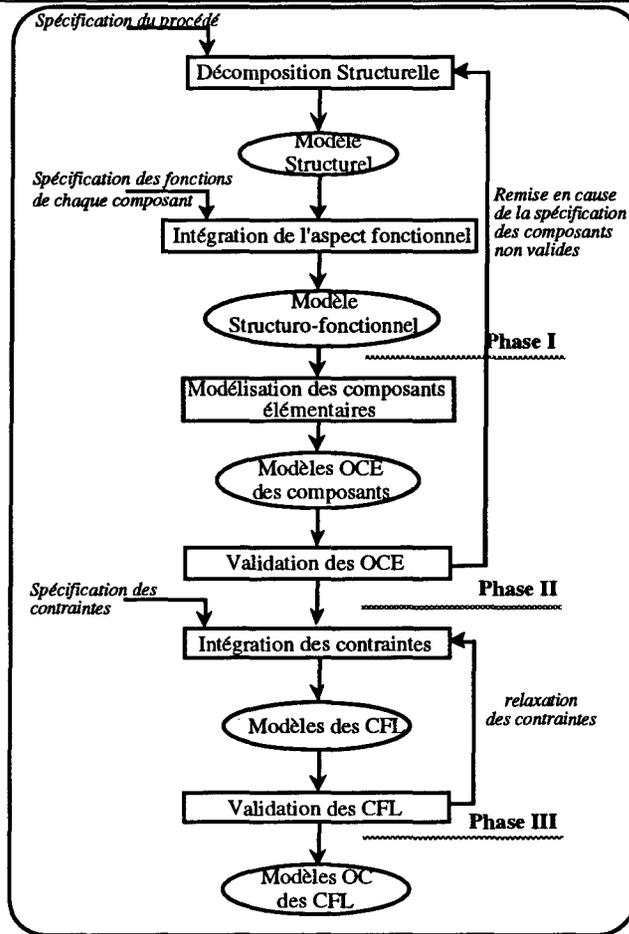


Figure VI.1 : Démarche de modélisation des CFL [Elkhattabi, 93]

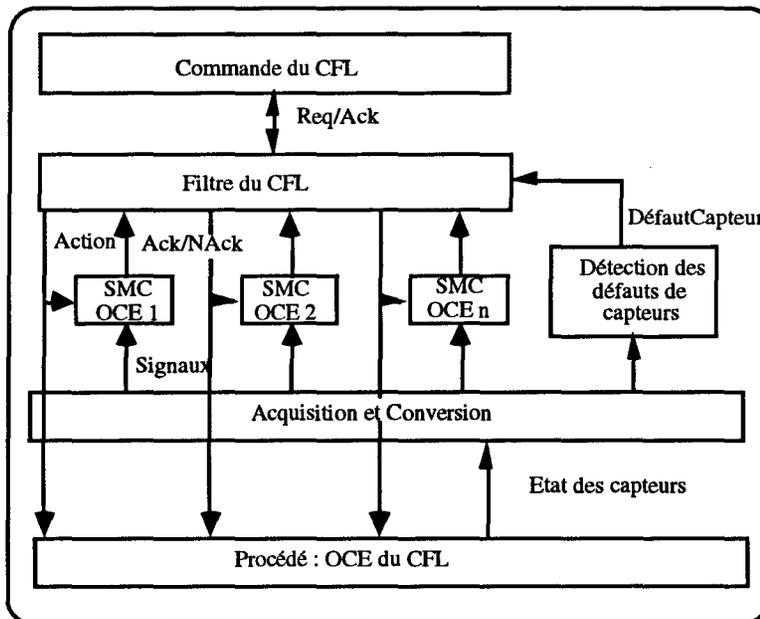


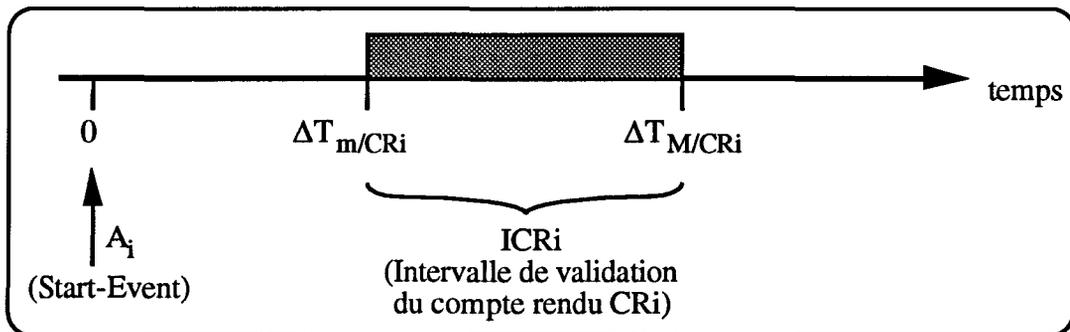
Figure VI.2 : Organisation de la surveillance de bas niveau d'un CFL

### VI.1.2. La surveillance en ligne [Toguyeni, 92]

La surveillance en ligne a pour but de surveiller automatiquement, pendant son fonctionnement, tout Système Automatisé (SA). La prise en compte de la dimension temporelle est la dominante des modèles proposés dans [Toguyeni, 92]. Cette prise en compte du temps a abouti à la proposition d'une surveillance en ligne d'un SFPM opérationnel. Nous présentons dans ce paragraphe deux étapes clés de la surveillance : la détection et le diagnostic.

#### La détection

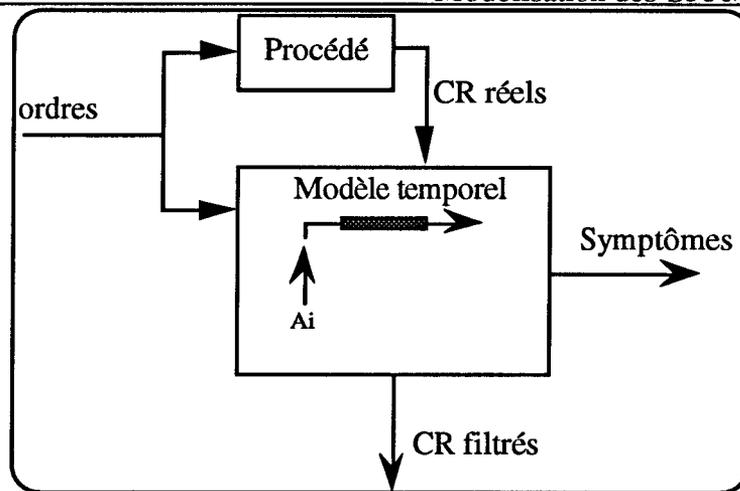
Les modèles proposés dans [Toguyeni, 90] et [Toguyeni, 92] s'inspirent de la notion de filtre introduite par le laboratoire du CRAN/LACN, [Alanche, 86] et [Lhoste, 91]. Ces filtres ont été étendus pour représenter, outre la partie opérative, toute opération (action ou macro-action), quel que soit le niveau de commande. Cette extension a été obtenue grâce à l'introduction d'une approche temporelle dans ces filtres. La figure VI.3 illustre une fenêtre temporelle  $[\Delta T_m, \Delta T_M]$  associée à une opération  $A_i$ . Le compte rendu  $Cr_i$ , associé à  $A_i$ , n'est validé qu'à l'intérieur de cette fenêtre.  $\Delta T_m$  et  $\Delta T_M$  sont les dates extrêmes relatives à l'instant de déclenchement de  $A_i$ . L'événement  $A_i$  considéré comme un événement initiateur du processus de détection sera appelé "Start\_Event".



*Figure VI.3 : Modèle associé à une opération [Toguyeni, 90]*

La détection se manifeste par l'émission d'un symptôme qui correspond à une différence entre les comportements prévu et observé [De Kleer, 87] (figure VI.4). Cette différence est signalée par l'absence d'un compte-rendu dans son intervalle de validation. [Toguyeni, 92] définit deux types de symptôme :

- les symptômes de type I correspondant à des compte-rendus attendus et non reçus dans leurs intervalles de validation.
- le symptôme de type II correspondant à des compte-rendus intempestifs qui interviennent en dehors de leurs intervalles de validation.



*Figure VI.4 : Filtrage à partir de la modélisation temporelle [Toguyeni, 92]*

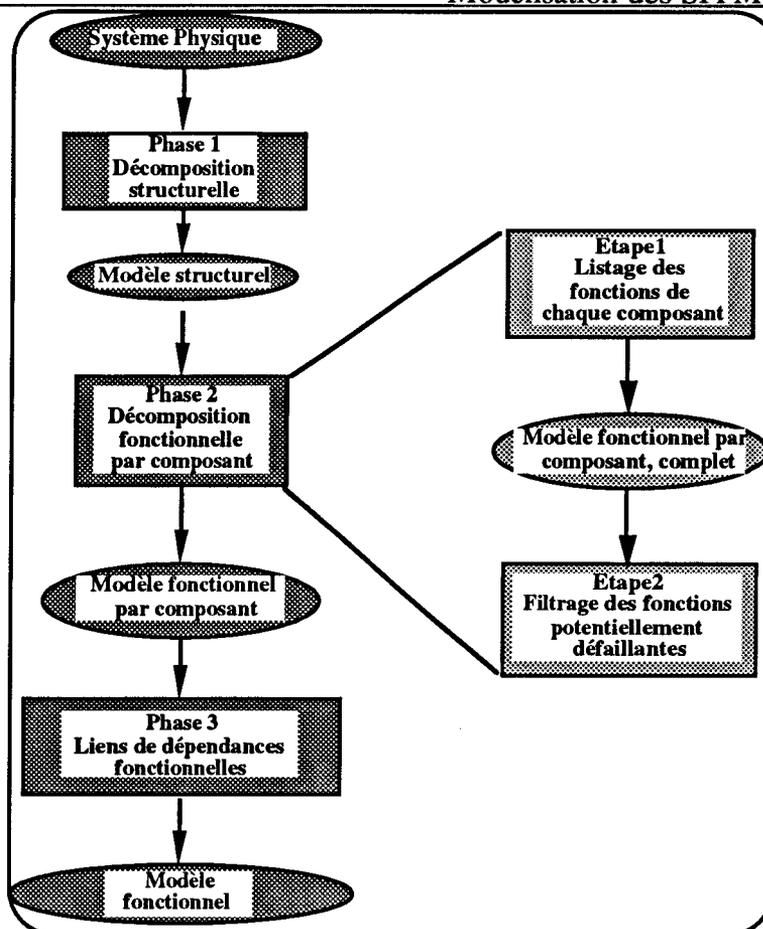
### Le diagnostic en ligne

Les symptômes déclenchent l'opération de diagnostic qui va déterminer leur origine. De nombreuses méthodes ont été proposées pour réaliser cette fonction [Villemeur, 88], [Chang et al, 89] et [Toguyeni, 90]. Ces méthodes proposent une modélisation du système peu compatible avec le diagnostic en ligne contrairement à [Toguyeni, 92] qui propose un modèle fonctionnel enrichi d'un modèle comportemental et d'une dimension temporelle. La figure VI.5 résume la démarche de construction du modèle fonctionnel.

La **phase 1** décompose le système en composants de premier niveau qui sont eux-mêmes décomposés jusqu'à l'obtention des composants de base. Les composants de base sont, soit des composants observables par des capteurs, soit des composants que la maintenance ne décompose pas. Cette phase de décomposition s'achève par l'intégration des liens de dépendance structurelle pour obtenir le modèle structurel de base.

La **phase 2** établit, pour chaque composant, la liste des fonctions susceptibles d'être défailtantes pendant la durée de vie moyenne du composant [RAC, 91]. On obtient alors un modèle fonctionnel réduit dédié au diagnostic.

La **Phase 3** produit un graphe orienté (figure VI.6) dont les nœuds représentent les fonctions, et les arcs les liens de dépendance. L'orientation des arcs indique le sens de dépendance des fonctions.



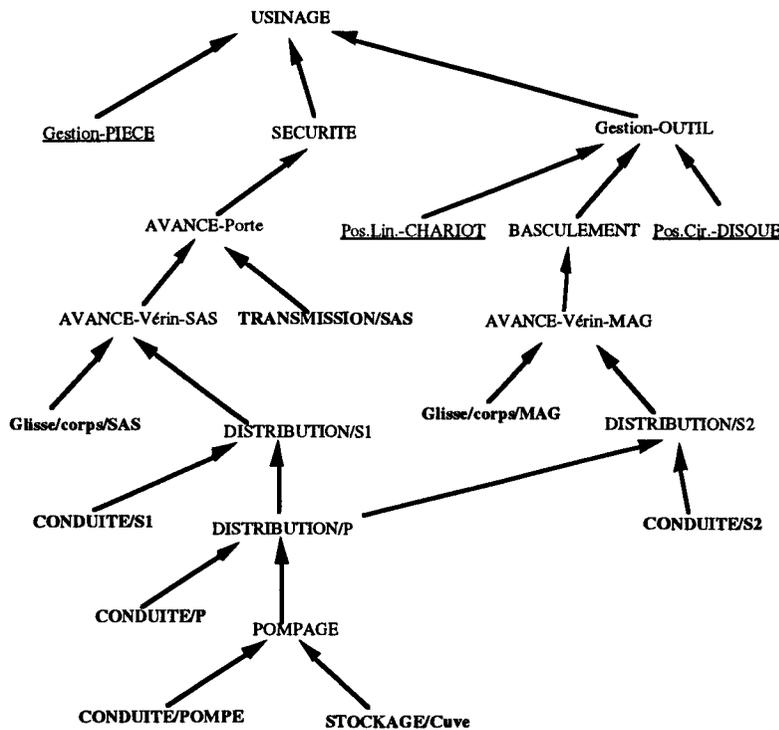
*Figure VI.5 : Conception du Modèle Fonctionnel pour le diagnostic [Toguyeni, 92]*

Le diagnostic comporte deux phases : la **localisation** et l'**identification**

La localisation (ou diagnostic local ) circonscrit l'origine de la défaillance à une zone du système. Elle est réalisée par les **noeuds initiateurs** correspondants aux fonctions du système ayant une observabilité directe. Le mécanisme de localisation est mis en œuvre par l'interprétation de STC (Signature Temporelle Causale) [Toguyeni et al, 90], [Toguyeni et al, 91], un type de règles modélisant le comportement défaillant d'une portion du système.

L'identification (ou diagnostic global) détermine les **fonctions initiatrices** ou **noeuds terminaux** (similaires aux feuilles d'une arborescence au sens de la théorie des graphes [Gondran et al, 79]) qui sont à l'origine d'une défaillance. Pour cela, le GF est identifié à un graphe d'hypothèses dans lequel quatre types de noeuds sont définis : les **noeuds initiateurs**, les **noeuds "validation-directe"**, les **noeuds "validation-indirecte"**, les **noeuds simples**. Chaque type de noeuds correspond à des fonctions qui ont un apport différent dans le cadre de l'identification du fait de leurs propriétés statique (position dans le graphe) ou dynamique (type et nature de l'observabilité). Ainsi, les hypothèses sont générées par les noeuds initiateurs et sont validées par les noeuds "validation-directe" ou "validation-indirecte". Les noeuds "validation-directe" ouvrent le système à

d'autres formes de diagnostic, notamment celles liées aux techniques de maintenance prédictive. Le GF et le modèle structurel sont conjointement utilisés pour réaliser l'analyse des conséquences (ou pronostic) afin d'anticiper l'appel par la PC de fonctions dont la défaillance est latente. Nos travaux reprennent cet objectif de pronostic à travers la mise à jour réactive de l'état des ressources et des fonctions du SFPM à partir du modèle fonctionnelle de la Partie II et de la formalisation des flux d'information dans le SFPM. La proposition de [Toguyeni, 92] ne tient pas compte de la nécessité d'une approche réactive de la gestion de l'information puisque c'est une démarche essentiellement hors ligne.



*Figure VI.6 : Graphe Fonctionnel (GF) d'un centre d'usinage [Toguyeni, 92]*

Les deux approches de la surveillance que nous venons de présenter utilisent des modèles différents mais elles sont complémentaires. En effet, L'approche de [Toguyeni, 92] garantit l'observabilité du procédé en réalisant la détection et le diagnostics des défauts des ressources en supposant la commande est saine. Or, l'approche de [Elkhatabi, 93] garantit une commande saine en filtrant les ordres émis vers les actionneurs supposés observables. Des travaux en cours tentent de rélisier l'interfaçage de ses deux approches, notamment à travers la liaison entre le filtrage et la détection.

Les travaux de AKA Toguyeni et de S. Elkhatabi nous intéressent car ils permettent de faire l'hypothèse d'une partie opérative complètement observable et commandable. Les fonctions de filtrage et de détection du premier niveau (le plus proche du procédé) de l'architecture de commande de la figure V.6.b peuvent être assurées par les fonctions proposées dans [Toguyeni, 92] et [Elkhatabi, 93].

## VI.2. La gestion des modes de marche et d'arrêt [Bois, 91]

L'approche de la gestion des modes d'un SFPM proposée dans [Bois, 91] s'appuie sur le GEMMA. Elle ne prend en compte que l'aspect ressource du système de production. Nous montrerons au chapitre suivant que cette démarche ne permet pas de modéliser des systèmes où des ressources multi-fonctions coexistent avec des ressources mono-fonctions. Nous montrerons, ensuite, que la modélisation structurelle proposée ne permet ni la gestion des modes de fonctionnement ni l'exploitation de la flexibilité des SFPM.

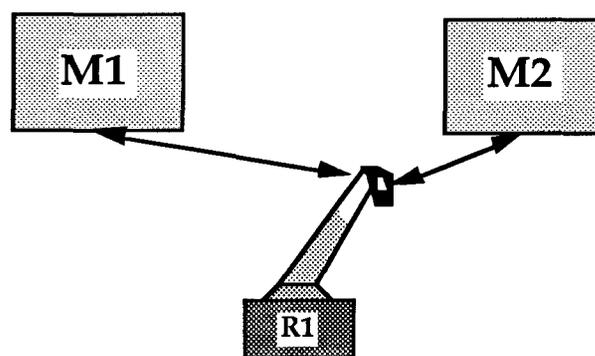
[Bois, 91] propose une représentation arborescente des systèmes de production. Cette représentation modélise les contraintes entre les machines d'un SFPM. Elle a pour objet la gestion des modes de marche des machines (i.e. leur commande effective) en intégrant la modification de leurs états. La démarche de modélisation suit 3 étapes :

- **la spécification du système de production** par l'inventaire de toutes les machines ;
- **la spécification des contraintes entre machines** ;
- **le regroupement des machines** dans des sous-ensembles suivant une analyse ascendante.

### **VI.2.1. Les contraintes entre machines**

[Bois, 91] distingue deux types de contraintes : les contraintes sur états (Contrainte de Coopération, Contrainte de Coopération Partagée, Contrainte d'Exclusion et Contrainte Structurelle) et les contraintes sur changements d'état (Contrainte PROCédurale, Contrainte d'Observabilité). Toutes ces contraintes sont représentées par une variante d'arbre ET/OU. En fait, la relation ET/OU est typée par l'intégration d'une contrainte qui paramètre le type de la relation. Les Opérateurs "ET/OU" utilisés interviennent dans le calcul de l'état de la machine virtuelle et la mise en conformité de l'état des machines composantes.

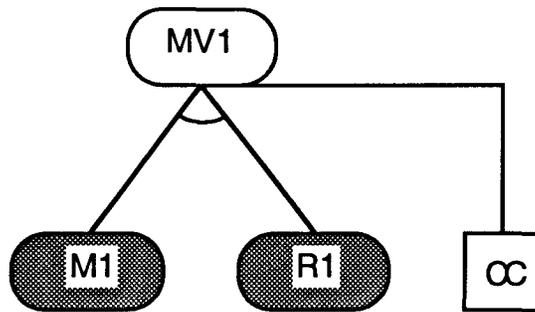
Ce paragraphe présente les contraintes définies dans [Bois, 91] en les illustrant par l'exemple de la figure VI.7 :



*Figure VI.7 : Exemple de cellule flexible*

La Contrainte de coopération CC

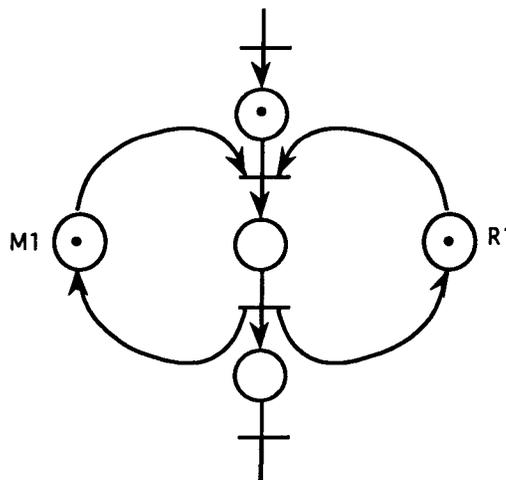
Deux ou plusieurs machines sont liées par une contrainte de coopération si elles sont fortement interdépendantes.



*Figure VI.8 : Contrainte de Coopération*

Dans l'exemple de la figure VI.7, la machine M1 a besoin du robot R1 pour produire. De même, si nous faisons abstraction de la machine M2, il est clair que le robot R1 n'a pas de raison d'être en marche si la machine M1 ne l'est pas. Ces deux machines effectives peuvent donc être réunies en une machine virtuelle MV1 par un opérateur de type "ET" pour tenir compte de leur contrainte de coopération (figure VI.8).

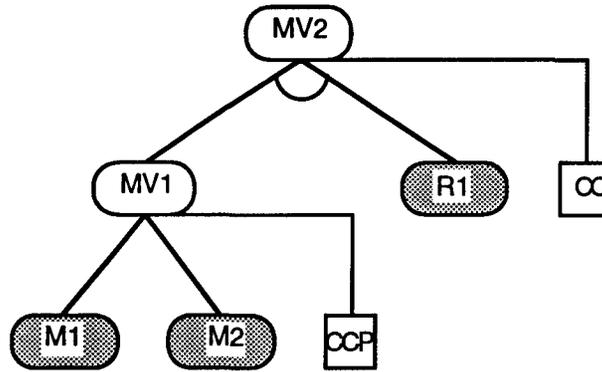
La figure VI.8.b. illustre avec le formalisme réseau de Petri (RdP) l'évolution de la commande des deux ressources M1 et R1.



*Figure VI.8.b : Description RdP de la contrainte de coopération*

La Contrainte de coopération partagée CCP

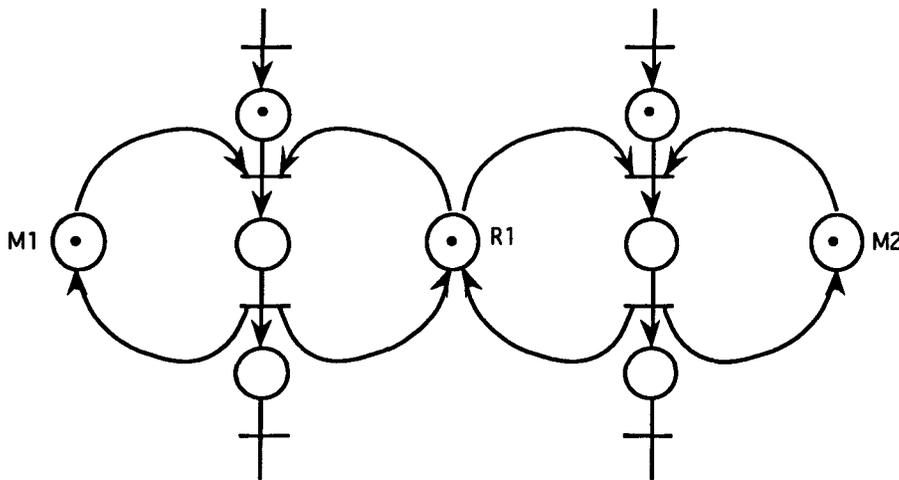
Deux ou plusieurs machines sont liées par une contrainte de coopération partagée si elles peuvent fonctionner de manière indépendante sans avoir d'autre lien que des demandes d'utilisation, à certains moments, d'une troisième ressource.



*Figure VI.9 : Contrainte de Coopération Partagée*

Le raisonnement du paragraphe précédent pour M1 et R1 vaut aussi pour la deuxième machine M2 et le même robot R1. Cela signifie que M1 et M2 partagent la même ressource R1. Ces deux machines peuvent donc être regroupées en une machine virtuelle par un opérateur de type "OU" pour modéliser la contrainte de coopération partagée qui les relie (figure II.9).

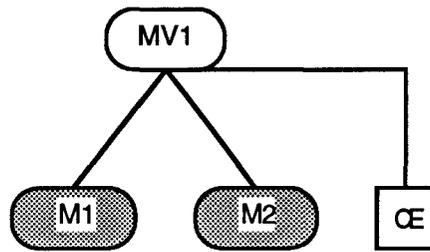
La figure VI.9.b. illustre la commande des ressources M1, M2 et R1.



*Figure VI.9b : Description RdP de la contrainte de coopération partagée*

### La Contrainte d'exclusion CE

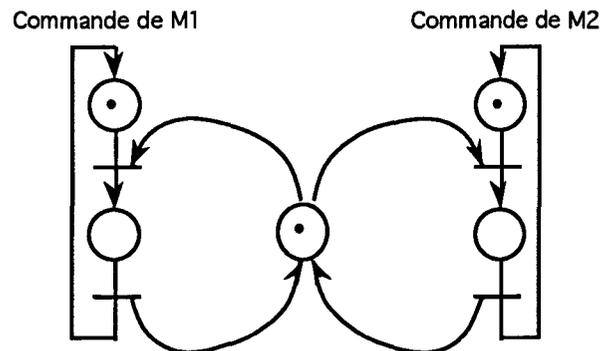
Deux ou plusieurs machines sont liées par une contrainte d'exclusion si leurs états doivent être complètement exclusifs. Par exemple, ce sont des machines qui ne peuvent pas être en marche en même temps.



*Figure VI.10 : Contrainte d'exclusion*

M1 et M2 sont, par exemple, 2 machines qui travaillent sur une zone opératoire commune. Elles ne peuvent donc pas être actives en même temps.

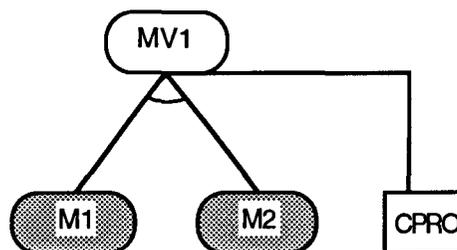
La figure VI.10.b. illustre l'évolution de la commande des deux ressources en exclusion mutuelles M1 et M2.



*Figure VI.10.b : Description RdP de la contrainte d'exclusion*

### La Contrainte de Procédure (CPRO)

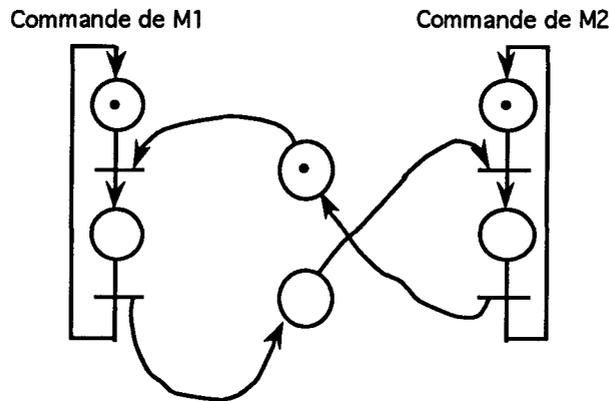
Cette contrainte prend en compte une procédure de fonctionnement liant deux ou plusieurs machines.



*Figure VI.11 : Contrainte de Procédure*

Deux pièces sont d'abord lavées par M1 puis assemblées par M2. Ces deux machines doivent toujours travailler dans cet ordre.

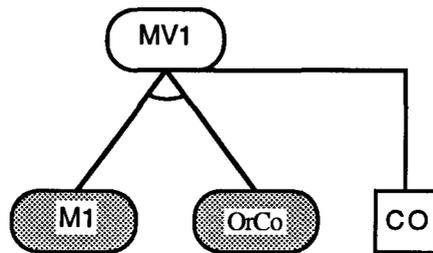
La figure VI.11.b. illustre la commande des deux ressources M1 et M2 dans le cas d'un fonctionnement procédural cyclique.



*Figure VI.11.b : Description RdP de la contrainte de procédure*

### La Contrainte d'observabilité CO

C'est une contrainte de coopération particulière. En effet elle représente le lien entre une machine opérative et un organe de commande ou de suivi. Elle modélise la possibilité d'accéder aux informations nécessaires à la commande et au contrôle de la machine.



*Figure VI.12 : Contrainte d'Observabilité*

OrCo est un organe de commande et d'observation (automates, PC, réseau, ...). Il permet de connaître l'état de M1 et de la commander. L'ensemble M1-OrCo constitue la machine virtuelle MV1.

### Les Contraintes structurelles

Ce sont des contraintes liées à la structure du système. Deux configurations peuvent se présenter :

- *Fonctionnement en série* : la sortie d'une machine correspond à l'entrée d'une autre.
- *Fonctionnement en parallèle* : deux machines effectuent, en parallèle, le même travail et sont complètement indépendantes.

### VI.2.2. Regroupement des machines

Le problème posé au concepteur est d'effectuer des regroupements de machines permettant d'expliquer le comportement d'un système de production à partir du comportement de ses composants et des différentes contraintes qui lient le comportement des machines. [Bois, 91] propose d'utiliser une table des contraintes ainsi que des règles de regroupement.

#### La table des contraintes

Les machines sont listées en entrée d'une table dont le rôle est de présenter la spécification des contraintes et les liens entre les ressources du système de production (figure VI.13).

	M1	M2	M3	M4	M5		Mi
M1		C12	C13	C14	C15		C1i
M2	C21						C2i
M3	C31						C3i
M4	C41						C4i
M5	C51						C5i
Mi	Ci1	Ci2	Ci3	Ci4	Ci5		

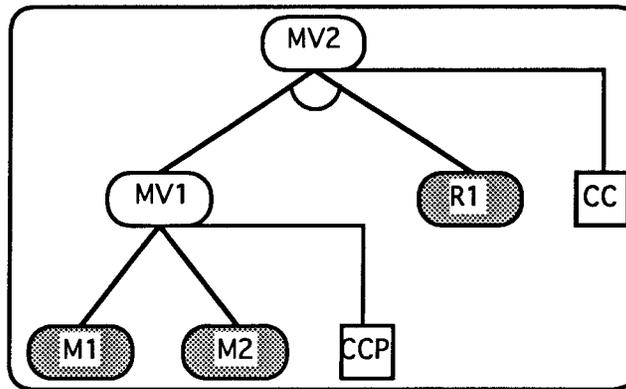
*Figure VI.13 : Exemple de Table*

$C_{ij}$  signifie que la machine  $i$  et la machine  $j$  sont mutuellement contraintes. Mais certaines  $C_{ij}$  ne sont pas commutatives : une machine exerce unilatéralement une contrainte sur une autre (la machine  $i$  exerce la contrainte  $C$  sur la machine  $j$ ). C'est notamment le cas de la contrainte d'observation qui est unilatérale.

#### Les règles de regroupement

Considérons à nouveau l'exemple des machines M1-M2 de la figure 2. Il y a une Contrainte de Coopération (CC) entre chacune des machines M1 et M2 et le robot R1. Une Contrainte de Coopération Partagée lie M1 et M2. Trois regroupements sont a priori possibles : regrouper d'abord M1 (respectivement M2) avec le robot R1 pour constituer une machine virtuelle MV1. Puis, regrouper MV1 avec M2 (respectivement M1) de manière à reconstituer le système de production M1-M2. Le problème est alors de déterminer la contrainte qui va lier MV1 et M2 (respectivement M1) sachant que les deux machines se trouvant dans le regroupement partagent avec M2 (respectivement M1) des contraintes différentes. Cet exemple illustre la nécessité d'avoir des règles

de regroupement de manière à rendre la modélisation systématique. S. Bois en propose trois qui permettent d'obtenir la troisième configuration présentée figure VI.14 :



*Figure II.14 : Application des règles de regroupement à l'exemple*

**Règle 1** : regrouper des **machines connexes** qui partagent les mêmes contraintes, en machines virtuelles par une phase d'analyse ascendante à partir des informations regroupées dans la table des contraintes. Ces machines virtuelles sont elles-mêmes regroupées jusqu'à la constitution d'une seule machine virtuelle.

**Règle 2** : définir une priorité entre les contraintes pour rendre déterministe les regroupements effectués. Cette priorité est liée à l'aspect critique d'une relation. Ainsi si une contrainte d'observabilité n'est plus respectée, la machine concernée par cette relation n'est ni observable, ni commandable. L'analyse du degré de criticité de chaque relation permet de proposer la relation d'ordre suivante qui va de la contrainte la plus forte à la contrainte la plus faible :

- Contrainte d'Observabilité.
- Contrainte de Coopération.
- Contrainte d'Exclusion.
- Contrainte de Coopération Partagée.
- Contrainte de Procédure.
- Contrainte Structurale Série.
- Contrainte Structurale Parallèle.

**Règle 3** : à contraintes égales, le regroupement impliquant le moins de machines est prioritaire.

### VI.2.3. Structuration et fonctionnement du gestionnaire de modes de marches [Bois, 91]

Le gestionnaire proposé dans [Bois, 91] sert à la gestion des modes de marche et à la coordination des différentes fonctions du contrôle/commande (appelé aussi Niveau Hiérarchique). Ce gestionnaire fonctionne selon trois phases :

- Une phase de génération d'ordres, également appelés buts. Cette phase est initialisée, soit

par l'opérateur qui peut décider l'arrêt d'une ou de plusieurs machines, soit par une modification dans la base de données dynamique du procédé.

- Une phase de résolution des buts de type génération de plan. Cette phase spécifie les actions à réaliser et classe les buts par ordre de priorité.
- Une phase de correction qui émet des actions mettant en jeu un recouvrement au niveau du procédé (arrêt effectif d'une machine) et un recouvrement au niveau de la commande (nouveau routage des pièces).

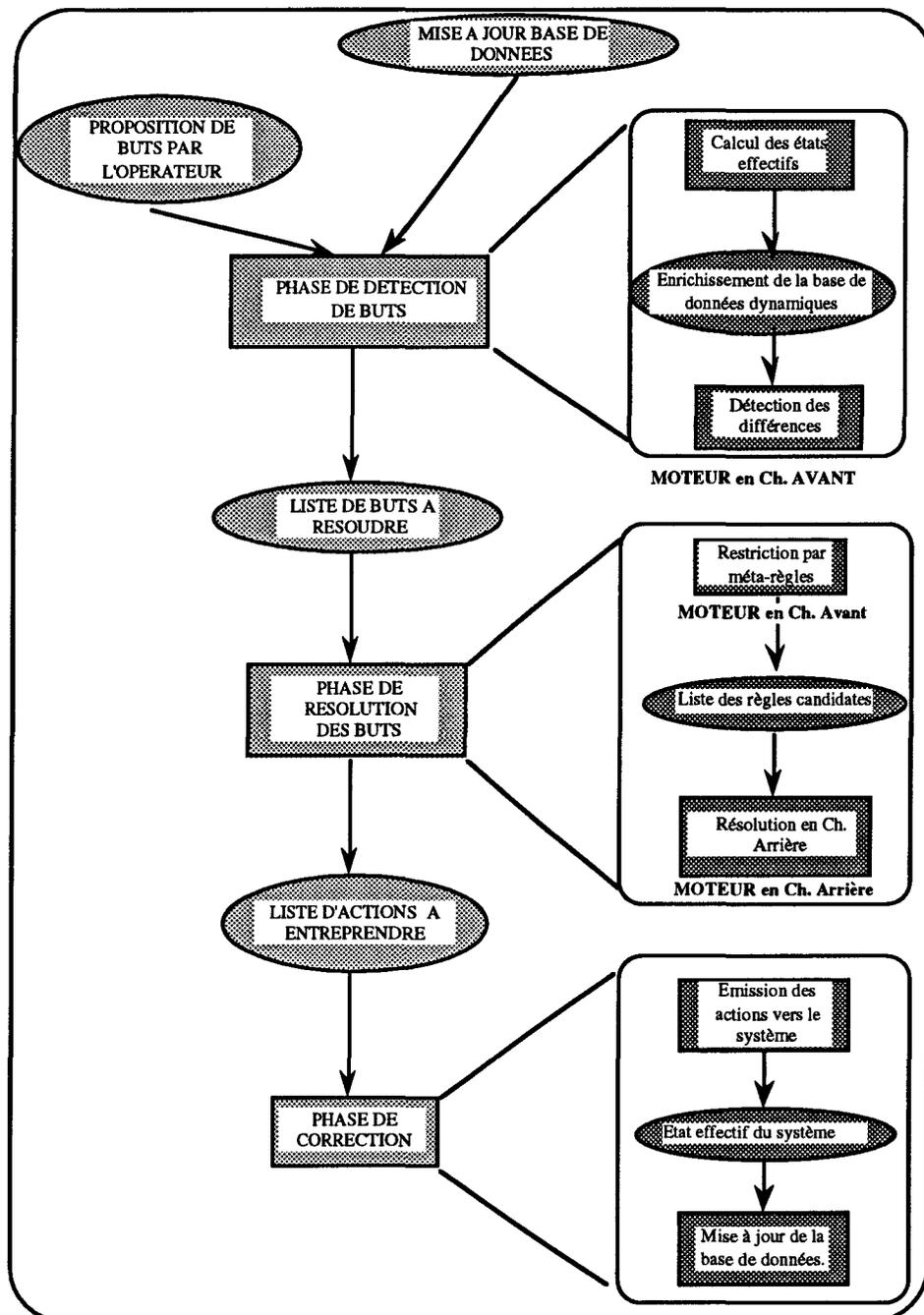
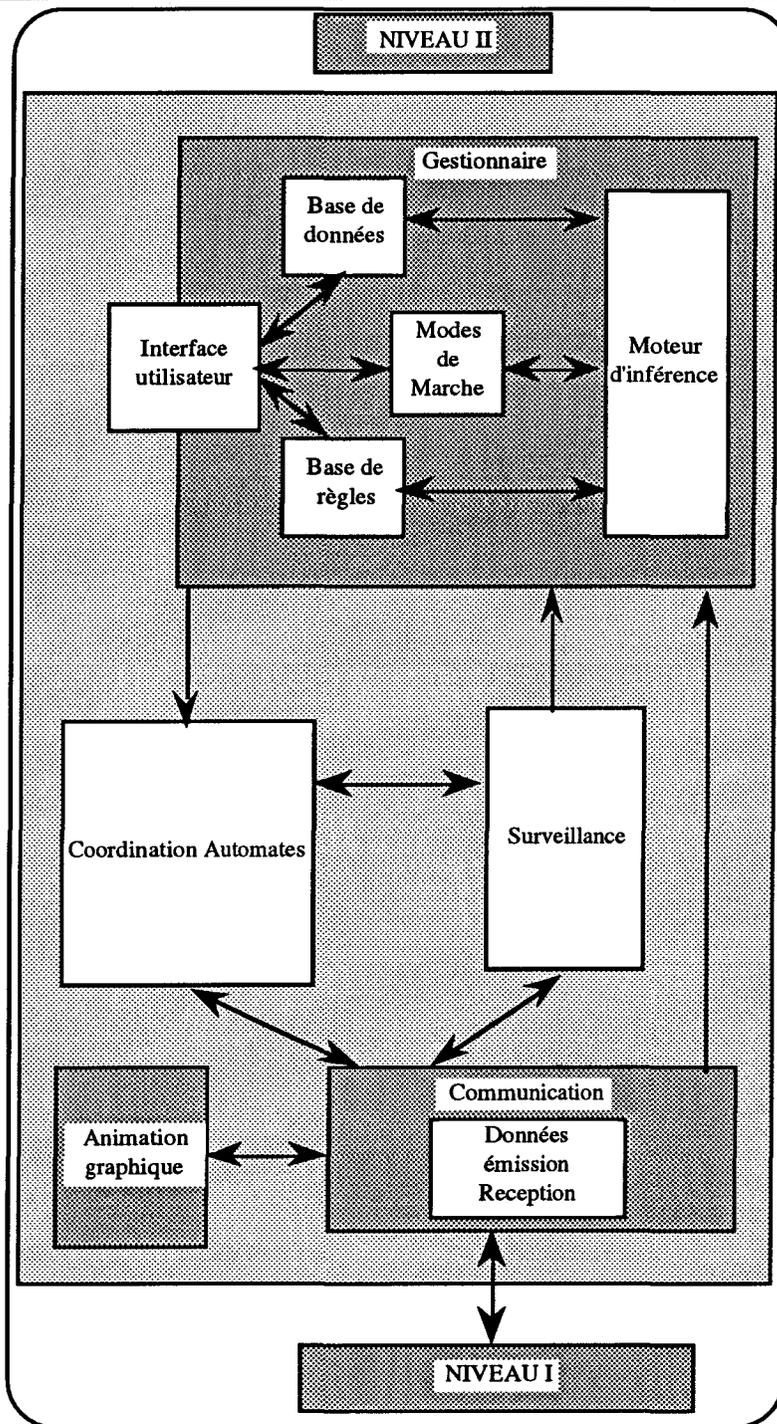


Figure VI.15 : Fonctionnement du gestionnaire d'après [Bois, 91]



*Figure VI.16 : Structuration du Niveau Hiérarchique et du gestionnaire des modes de marche  
d'après [Bois, 91]*

Le gestionnaire des modes est lié aux trois fonctions principales du Niveau Hiérarchique (figure VI.16) :

- La surveillance [Toguyeni, 92] qui permet d'identifier le dysfonctionnement d'une ressource. Le gestionnaire doit alors arrêter cette ressource et assurer la sécurité de son environnement selon le processus décrit par la figure VI.15. Le but à atteindre ne sera pas

proposé par l'opérateur mais par la surveillance.

- La coordination des automates qui résout les indéterminismes du graphe de coordination [Craye, 89],
- l'interface utilisateur qui permet à l'opérateur de paramétrer le fonctionnement de l'atelier ou de le gérer manuellement.

### Conclusion

La démarche que nous venons de présenter est un exemple de mise en œuvre du GEMMA. Toutefois, elle reste limitée dans la mesure où les graphes de comportement des machines n'intègrent pas tous les états spécifiés dans le GEMMA. En effet, il est difficile d'interpréter les contraintes opératoires exprimées dans la composition des états liés à la maintenance ou aux tests, tout en garantissant l'unicité de mode du système. Cette approche présente également plusieurs autres limites liées à son développement dans le contexte de la première version de CASPAIM. En effet, elle ne peut pas traduire la hiérarchisation de la commande introduite dans CASPAIM par [Cruette, 91], [Ausfelder, 94] et [Amar, 94]. En outre, elle ne tient pas compte des contraintes temporelles fortes induites par le procédé sur les couches basses de la commande. La modélisation du procédé présentée ici, a permis de construire un gestionnaire des modes de marches et d'arrêts à partir d'un générateur de plans. Ce type d'outil prend le temps d'inférer pour trouver les actions à mener pour ramener le système dans un état stable (du point de vue du concepteur). Il n'est donc pas à même de répondre à des objectifs de réactivité du système et d'assurer des traitements d'urgence ou de préemption. [Elkhatabi, 91] propose une solution partielle à ce problème à travers la surveillance directe et réactive des actionneurs du SFPM. Mais le problème reste posé en ce qui concerne les ressources opératives, car on doit intégrer beaucoup plus de contraintes liées à leur flexibilité et leur complexité.

Le dernier problème de l'approche S. Bois est lié à ses objectifs. Cette approche ne vise que la commande effective de la marche et de l'arrêt des machines. Elle n'apporte pas de solution au problème de la gestion des modes de fonctionnement et d'exploitation des SFPM par la reconfiguration dynamique et l'exploitation de la flexibilité.

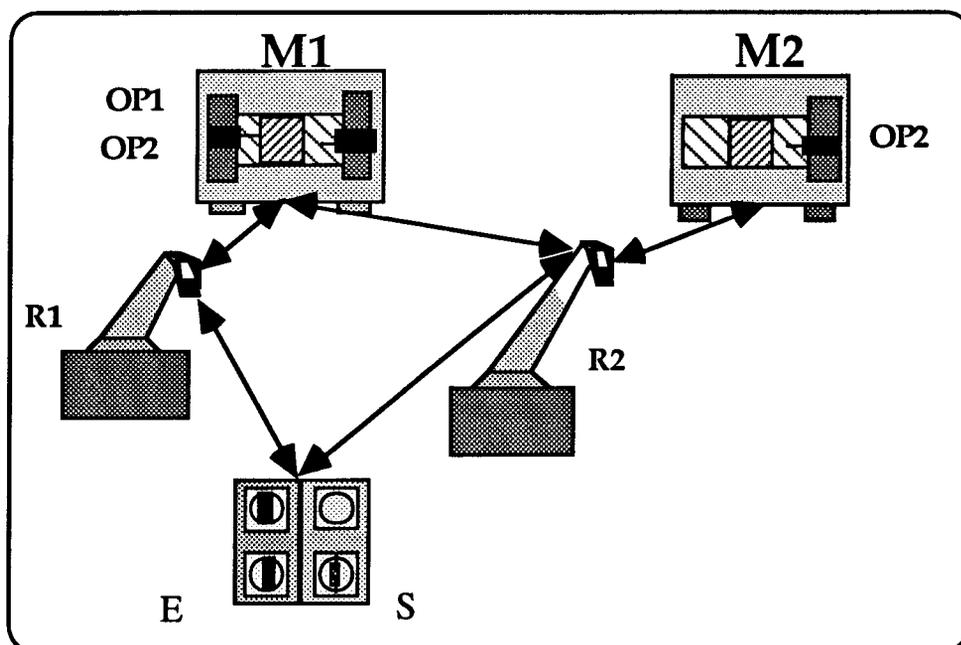
## CHAPITRE VII : EXIGENCE D'UNE APPROCHE FONCTIONNELLE POUR LA GESTION DES MODES

### VII.1. Limites de l'approche structurelle

L'approche proposée par S. Bois ne peut pas représenter les SFPM qui ont plus d'opérations que de machines et, inversement, ceux qui ont plus de machines que d'opérations différentes. La modélisation proposée dans [Bois, 91] ne permet pas de représenter l'utilisation privilégiée d'une machine pour réaliser une opération et le remplacement de cette machine par une autre en cas de défaillance. L'exemple de la figure VII.1 illustre notre propos. Cet exemple servira tout au long de ce mémoire.

La figure VII.1 représente une cellule flexible constituée de

- deux machines, M1 et M2,
- deux robots, R1 et R2,
- d'un tampon d'entrée E et d'un tampon de sortie S.



*Figure VII.1 : Exemple de cellule flexible*

La machine M1 (un centre d'usinage, par exemple) exécute les opérations Op1 (fraisage, par exemple) et Op2 (tournage, par exemple).

La machine M2 exécute la même opération OP2.

Le robot R1 assure les transferts entre M1 et les tampons d'entrée et de sortie

Le robot R2 assure les transferts entre M2 et les tampons d'entrée/sortie d'une part, et M1 d'autre

part. Ce robot peut assurer les transferts entre M1 et les tampons d'entrée/sortie ; mais il n'est autorisé à le faire qu'en cas de défaillance et de mise en retrait de R1.

Appliquons la démarche proposée dans [Bois, 91] à cet exemple. Nous aboutirons à une représentation arborescente dont nous mettrons en évidence les limites, tant pour la représentation des contraintes de fonctionnement de la cellule flexible que pour son exploitation.

La méthode de [Bois, 91] commence par la construction du tableau initial des contraintes entre les ressources (figure VII.2). Toutes les contraintes du tableau de la figure VII.2 sont des contraintes de coopération (CC).

	M1	R1	M2	R2
M1		CC1		CC2
R1	CC1			
M2				CC3
R2	CC2		CC3	

*Figure VII.2 : Tableau initial des contraintes*

La règle trois de [Bois, 91] privilégie le regroupement impliquant le moins de machines. Or, le robot R1 ne partage qu'une seule contrainte avec M1. Ces deux machines sont regroupées pour former la machine virtuelle MV1.

	M1	R1	M2	R2
M1		CC1		CC2
R1	CC1			
M2				CC3
R2	CC2		CC3	

*Figure VII.3 : Premier niveau d'agrégation*

La figure VII.4 montre que MV1 et M2 sont toutes deux en coopération avec le robot R2. Elles sont donc liées par une contrainte de coopération partagée, CCP (figure VII.5). La machine virtuelle résultant de leur regroupement est en coopération avec le robot R2 (figure VII.6).

	MV1	M2	R2
MV1			CC2
M2			CC3
R2	CC2	CC3	

*Figure VII.4 : Deuxième tableau des contraintes*

	MV1	M2	R2
MV1		CCP	CC2
M2	CCP		CC3
R2	CC2	CC3	

*Figure VII.5 : Deuxième niveau d'agrégation*

	MV2	R2
MV2		CC
R2	CC	

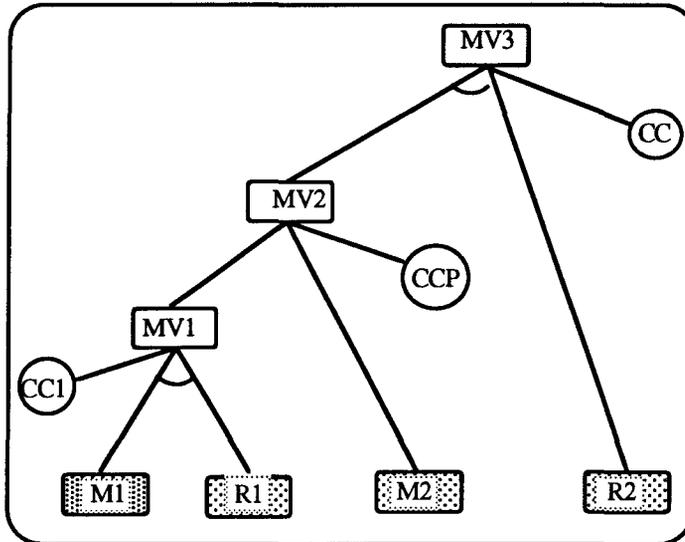
*Figure VII.6 : Troisième niveau d'agrégation*

La construction de la représentation arborescente de la figure VII.7 constitue la phase finale démarche proposée dans [Bois, 91].

Cette représentation n'est pas unique. En effet, l'application des règles de [Bois, 91] ne suffit pas à obtenir le bon modèle. Le concepteur doit avoir une "idée" de ce qu'il souhaite représenter pour y arriver. Pour obtenir MV1, il est possible de regrouper la machine M2 et le robot R2, car M2 n'est directement liée qu'à R2. MV2 serait alors le regroupement de MV1 et de R1. Le résultat est l'arbre de la figure VII.7.a. La prise en compte prioritaire des ressources de transfert a été privilégiée implicitement dans la première représentation.

La représentation de la figure VII.7a n'est pas significative du potentiel de fonctionnement et de la flexibilité de notre cellule. En effet, d'après [Bois, 91], si le robot R2 est en panne, c'est toute la cellule qui devient indisponible. Or, l'ensemble M1-R1 est complètement utilisable et permet de réaliser toutes les opérations de transformation de la cellule. La fonctionnalité de la cellule n'est pas affectée par la panne de R2. C'est sa performance qui est dégradée. De même, si le robot R1 est en panne, la machine M1 est mise hors service. Or, R2 peut se substituer à R1. Une fois encore, seules

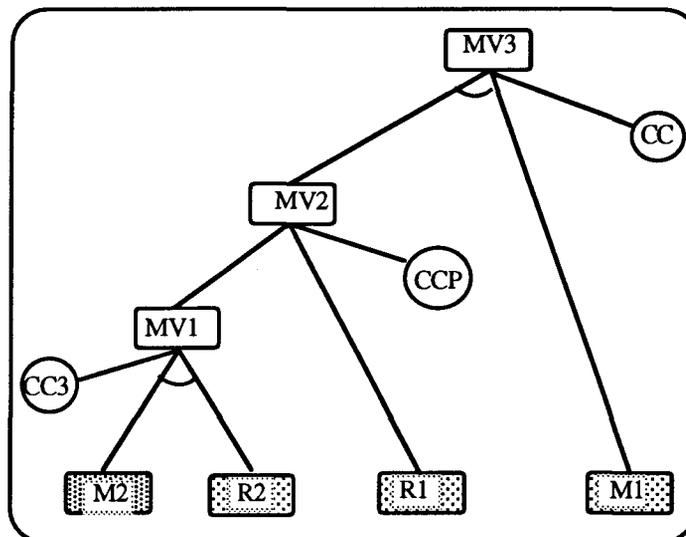
les performances du système sont affectées. Ses fonctionnalités restent intactes.



*Figure VII.7.a : Représentation arborescente*

La représentation de la figure VII.7.b est encore moins représentative du fonctionnement de la cellule. En effet, une panne de M1 paralyserait l'ensemble de la cellule, alors qu'en réalité, le groupe M2-R2 peut très bien fonctionner sans M1.

Nous verrons, plus loin, que la représentation proposée par S. Bois est, de la même manière, insuffisante pour tenir compte des évolutions récentes de la cellule flexible de l'EC Lille qui nous servira de deuxième exemple d'illustration de la démarche que nous proposons.



*Figure VII.7.b : Deuxième représentation arborescente*

L'exemple que nous venons de présenter illustre les limites d'une approche structurale des SFPM. Cette approche restreint la visibilité du système. Nous voulons voir les opérations et

l'évolution de leurs états. Les machines ne nous intéressent ici que par leurs liens avec les opérations. Nous allons montrer dans la suite de ce chapitre l'apport et la nécessité d'une approche fonctionnelle pour la représentation d'un SFPM en vue de gérer ses modes en exploitant sa flexibilité.

## **VII.2. La modélisation fonctionnelle**

La modélisation fonctionnelle est une forme de modélisation qualitative. Elle fournit des modèles de connaissance. Un modèle fonctionnel est le résultat de l'analyse de la structure du système et de la recherche de la logique des mécanismes internes.

La modélisation fonctionnelle se rapproche du concept de modèles profonds [Chandrasekaran et al, 89]. En effet, ces modèles sont caractérisés par des connaissances causales à des niveaux croissants de détail et des informations explicites sur des liens causaux entre la structure et le comportement.

La modélisation fonctionnelle, outre son aspect fonctionnel, tient compte d'une composante structurelle et d'une composante comportementale [Amar, 94], [Chandrasekaran et al, 89].

Dans l'approche de [Amar, 94], les fonctions sont vues uniquement sous l'angle de la commandabilité ou des capacités de stockage. Seules les fonctions de base ont un comportement propre. Le comportement des fonctions de niveaux supérieurs est hérité de celui des fonctions constituantes.

Dans la modélisation de [Chandrasekaran et al, 89], les fonctions sont indépendantes de l'environnement. Chaque entité physique remplit au moins une fonction. Ces fonctions sont vues comme des fonctions génériques indivisibles. Leurs combinaisons déterminent les comportements des entités composées du système.

## **VII.3. Modèle fonctionnel et modèle structurel**

Un modèle fonctionnel spécifie les différentes fonctions d'un système à partir de la composition de fonctions élémentaires (du point de vue de la commande ou de la surveillance et de la maintenance). Cette composition est réalisée en définissant des liens entre fonctions.

Un modèle structurel spécifie les composants d'un système ainsi que leurs liens structurels.

La démarche que nous proposons pour la gestion des modes d'un SFPM sera fondée sur la modélisation fonctionnelle. De nombreux auteurs ([Kleer et al, 80], [Keuneke, 91] [Bradshaw, 91], [Young, 91], ...) démontrent la primauté du modèle fonctionnel sur le modèle structurel.

L'exemple de la cellule flexible (figure VII.1) illustre l'importance et l'apport d'une approche

Nous allons proposer deux politiques de gestion de ce système. Une politique P1 qui correspond à une approche structurelle où la structuration de la commande est une transposition de celle du procédé. Une politique P2 qui correspond à une approche fonctionnelle où les opérations sont prises en compte indépendamment des ressources.

### **Politique P1 :**

Les requêtes de transfert entre le tampon d'entrée et la machine M1 sont exécutées par la commande directe de R1 ou R2. De même, l'opération OP2 est réalisée par la machine sollicitée par l'opérateur.

### **Politique P2 :**

Les requêtes de transferts entre le tampon d'entrée et la machine M1 sont réalisées par R1 ou R2 si R1 est hors service. L'opération OP2 est réalisée par M2 ou M1 si M2 n'est pas disponible (occupée ou hors service).

Le choix de l'une ou l'autre de ces deux politiques va influencer sur la facilité et la qualité de la gestion des modes dégradés. Dans l'hypothèse de la politique P1, le passage en mode dégradé est à l'initiative de l'opérateur pour chaque pièce à fabriquer. Si sa requête pour une pièce est rejetée, il doit décider de changer de graphe de commande et d'appeler le programme de remplacement. Cette démarche n'est pas optimale puisqu'elle perd du temps dans l'appel et la mise en œuvre des procédures de remplacement.

Dans l'hypothèse de la politique P2, le passage en mode dégradé est transparent. En cas de panne de R1 (respectivement d'indisponibilité de M2) R2 (respectivement M1) assurera le service à chaque appel. La seule conséquence est un temps d'attente plus long en moyenne pour les pièces, du fait de la dégradation des ressources.

En conclusion, nous dirons que la modélisation d'un système flexible selon une approche fonctionnelle est mieux adaptée à la prise en compte du recouvrement et donc à la gestion automatique des modes.

## **VII.4. Conclusion**

Nous avons vu les limites de la représentation uniquement matérielle de S. BOIS. Dans ce chapitre, nous proposons de considérer le système de production comme un ensemble de tâches et d'opérations liées entre elles par des contraintes.

Du point de vue de la supervision d'un SFPM, la connaissance de l'état des fonctions disponibles

est au moins aussi importante que celle des machines. Dans un système simple où à une machine correspond une et une seule fonction, ces deux approches sont équivalentes. Mais dans un système complexe l'approche "structuro-fonctionnelle" est plus puissante. Cette approche permet de tenir compte de la multiplicité d'une ressource matérielle en la considérant comme une seule ressource opérationnelle. Les machines sont prises en compte par leurs fonctions dans le système.

Notre objectif étant la gestion des modes de fonctionnement et d'exploitation du système, sa spécification se fera par l'inventaire de toutes les fonctions avec leurs contraintes mutuelles et leurs ressources d'exécution. On ne se contentera plus d'une connaissance uniquement matérielle du système.

## **CHAPITRE VIII : REPRÉSENTATION FONCTIONNELLE POUR LA GESTION DES MODES D'UN SFPM**

Nous avons indiqué, dans les chapitres précédents, les limites d'une approche strictement matérielle et structurelle dans la gestion des modes d'un SFPM. La prise en compte de la dimension fonctionnelle des SFPM enrichit la modélisation et permet une spécification et une conception plus efficaces et mieux structurées du système de contrôle/commande. C'est notamment le cas de la commande de coordination et de la surveillance dans CASPAIM qui ont pleinement exploité l'aspect fonctionnel des systèmes de production.

Nous proposons dans ce chapitre une modélisation des SFPM en vue de la gestion de leurs modes et de leurs configurations. Cette modélisation s'appuie sur les contraintes de fonctionnement entre les différentes opérations. Elle suit 4 étapes :

- La première étape permet **la spécification du système de production** par l'inventaire de toutes ses opérations. Cette spécification précise la nature des opérations : transformation, métrologie, stockage ou transfert. Elle définit aussi les lieux d'accès de chacune d'elle. Ces lieux correspondent aux lieux de [Amar, 94]. Ils concernent les lieux opératoires et les lieux de chargement et de déchargement des opérations.
- La deuxième étape permet **la spécification des contraintes entre les opérations**. Cette spécification précise la nature des contraintes et les identifie par rapport aux définitions du paragraphe suivant. Cette étape décrit notamment la nature des liens entre les opérations de transformation, de métrologie ou de stockage et les opérations de transfert. Nous obtenons alors l'ensemble des transferts de chargement et l'ensemble des transferts de déchargement pour chaque opération.
- La troisième étape réalise **le regroupement des opérations dans des sous-ensembles** que nous appellerons opérations virtuelles. Ce regroupement, résultat d'une analyse ascendante du SFPM, est décrit plus loin dans ce chapitre. Ces regroupements seront représentés par une variante d'arbre ET/OU. La relation ET/OU est typée par l'intégration des paramètres de la contrainte.
- La dernière étape spécifie pour chaque opération **la configuration de chacune de ses machines d'exécution**. Cette étape permet d'établir un lien entre les opérations du SFPM et sa structure physique. Nous aboutissons alors à une représentation structuro-fonctionnelle.

La première partie de ce chapitre est consacrée à la définition et à la représentation des contraintes entre les opérations. Nous montrerons ensuite comment utiliser ces contraintes pour obtenir une représentation hiérarchique des opérations d'un SFPM. Le modèle obtenu sera relié à la structure et aux ressources de production.

Nous terminerons ce chapitre par l'illustration de notre démarche de modélisation sur deux exemples de SFPM en indiquant comment se servir du résultat pour la gestion des modes.

### **VIII.1. Les opérations dans un SFPM**

La description fonctionnelle que nous proposons concerne les opérations qui interviennent directement sur un produit. Nous nous intéresserons à 4 types d'opérations : les transformations (usinage, assemblage, ...), les contrôles (métrologie, test, ...), les stockages et les transferts. Les opérations de contrôle et d'observation des machines sont donc exclues de notre représentation.

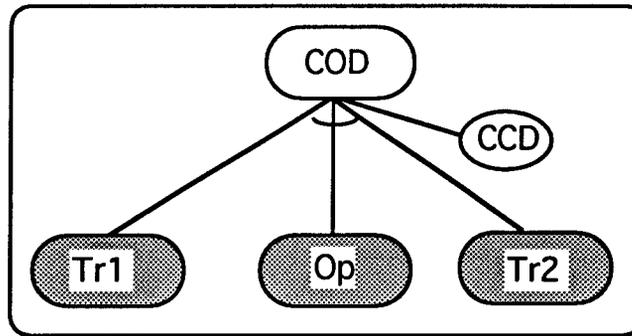
Les opérations de transfert ont un caractère particulier. Elles n'existent qu'en fonction des autres opérations. Les contraintes entre ces transferts et les autres opérations sont généralement spécifiées indépendamment des processus de fabrication dans un SFPM. Leur rôle est de charger et/ou décharger les autres opérations. Nous distinguerons donc deux classes d'opérations : les opérations de transfert que nous appellerons TRANSFERTS et les autres opérations que nous désignerons par OPÉRATIONS.

### **VIII.2. Définition et modélisation des contraintes entre opérations**

Nous avons défini deux classes d'opérations : les transferts et les autres opérations. La classe transfert a pour fonction le chargement et le déchargement de la classe opérations. Nous définissons donc un premier type de contraintes inter-classes : les Contraintes de Chargement et de Déchargement. Le deuxième type de contraintes concerne chacune des classes. Nous nous appuyons sur les travaux de S. Bois pour les définir. En effet, certaines contraintes entre les machines définies dans [Bois, 91] peuvent facilement être étendues aux opérations. C'est le cas notamment de la contrainte de procédure et de la contrainte d'Exclusion. Les contraintes de coopération ne sont pas prises en compte, car elles ne concernent que les contraintes entre transferts et opérations ; elles sont remplacées par les contraintes de chargement et de déchargement. La contrainte d'observabilité ne sert plus puisque les opérations d'observation des ressources sont exclues de notre champ de modélisation. Enfin, les contraintes structurelles ne sont à notre avis que des contraintes de procédure particulières.

#### **La Contrainte de Chargement et de Déchargement (CCD)**

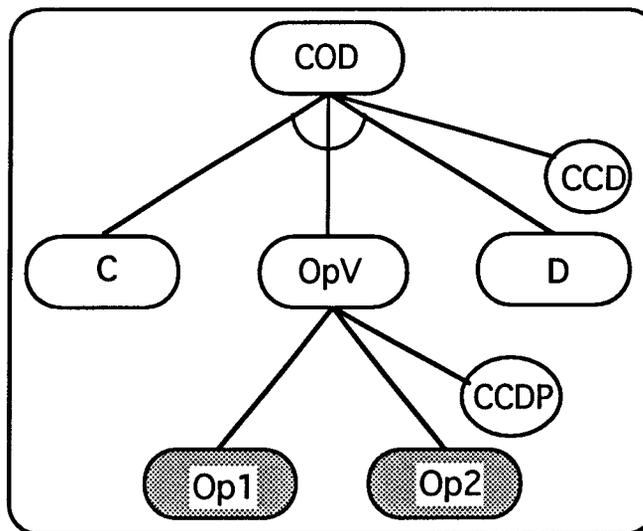
La contrainte de chargement et de déchargement concerne un transfert de chargement, un transfert de déchargement et une opération. Dans le cas de la figure IV.1, l'opération Op est chargée par le transfert Tr1 et déchargée par le transfert Tr2. Op, Tr1 et Tr2 peuvent donc être regroupées avec un opérateur de type "ET" pour constituer une opération plus abstraite COD (pour Chargement, Opération et Déchargement).



*Figure VIII.1 : Contrainte de Chargement et de Déchargement*

#### La Contrainte de Chargement et de Déchargement Partagée CCDP

Deux ou plusieurs opérations sont liées par une contrainte de Chargement et de Déchargement Partagés si elles partagent le même transfert de chargement et le même transfert de déchargement.

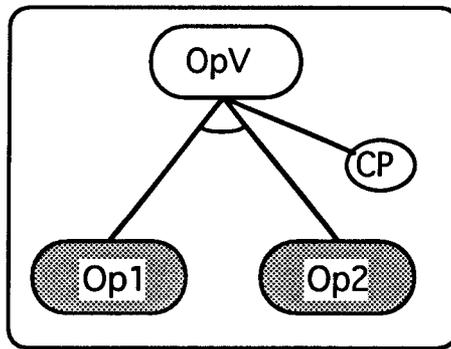


*Figure VIII.2 : Contrainte de Chargement et de Déchargement Partagée*

Op1 et Op2 sont chargées par le même transfert C et déchargées par un même transfert D. Ces deux opérations seront regroupées par un opérateur de type "OU" pour modéliser leur contrainte de chargement et de déchargement partagée. L'arbre de la figure IV.2 est l'agrégation de deux contraintes de chargement et de déchargement liant, à des moments différents, les transferts C et D à Op1 et Op2.

#### La Contrainte de Procédure (CP)

Deux ou plusieurs opérations sont liées par une Contrainte de Procédure si elles doivent suivre un protocole particulier de fonctionnement.

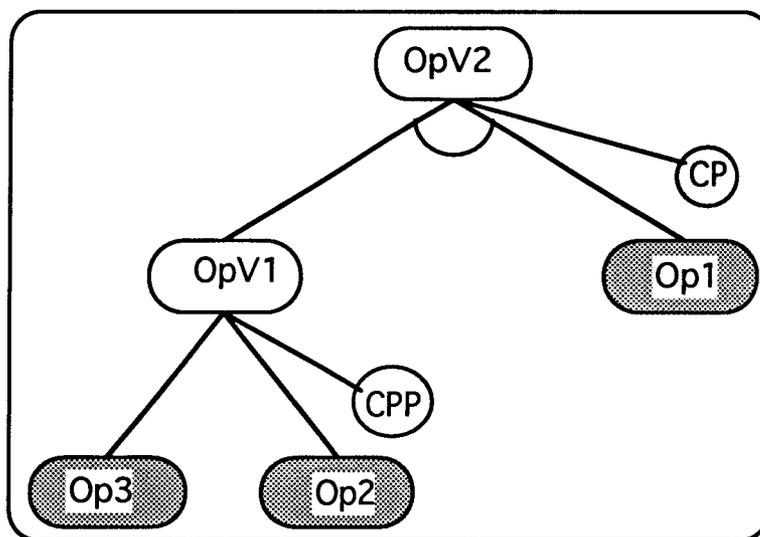


*Figure VIII.3 : Contrainte de Procédure*

L'opération Op1, une opération de lavage par exemple, doit impérativement précéder l'opération d'assemblage Op2. Ces deux opérations effectives peuvent donc être regroupées par un opérateur de type "ET" pour constituer une seule opération (figure VIII.3).

#### La Contrainte de procédure partagée CPP

Deux ou plusieurs opérations sont liées par une contrainte de procédure partagée si elles sont liées par des contraintes de procédures à une autre opération.

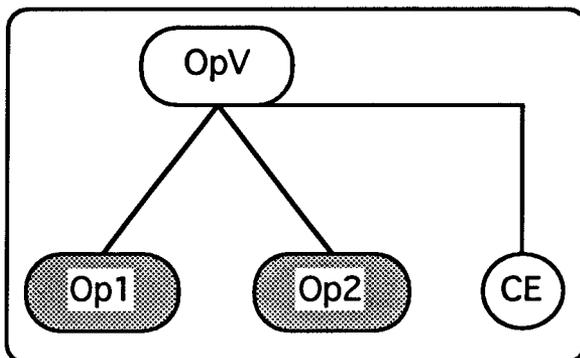


*Figure VIII.4 : Contrainte de Procédure Partagée*

Le raisonnement du paragraphe précédent pour Op1 et Op2 peut être reproduit pour un même type de contrainte entre une opération Op3 et l'opération Op1. Cela signifie que Op3 et Op2 sont liées chacune à Op1 par une contrainte de procédure. Ces deux opérations seront regroupées par un opérateur de type "OU" pour modéliser la contrainte de coopération partagée qui les relie. L'arbre de la figure IV.4 est l'agrégation de deux contraintes de procédure liant, à des moments différents, l'opération Op1 à Op3 et Op2.

La contrainte d'exclusion CE

Deux ou plusieurs opérations ou transferts sont liées par une contrainte d'exclusion si leurs états doivent être complètement exclusifs. Par exemple, ce sont des opérations qui ne peuvent pas traiter le même produit.



*Figure VIII.5 : Contrainte d'exclusion*

Op1 et Op2 sont, par exemple, 2 opérations qui travaillent sur une zone opératoire commune. Elles ne peuvent donc pas être actives en même temps.

### VIII.3. Le Regroupement des opérations

Le regroupement des opérations permet d'élaborer des vues agrégées du système de production. Le comportement global du SFPM est obtenu à partir du comportement des opérations et des transferts. Le but du regroupement étant la mise en évidence de sous systèmes indépendants, il s'agit alors de repérer des groupements d'opérations partageant les mêmes types de contraintes.

Le regroupement des opérations est effectué par une analyse ascendante qui prend en compte les opérations et les transferts effectifs pour constituer des groupes d'opérations contraintes. Lorsque ces regroupements sont réalisés, il peut apparaître, alors, de nouvelles contraintes entre les opérations virtuelles. La démarche de regroupement est à nouveau reprise au niveau supérieur pour obtenir de nouvelles opérations virtuelles. Cette démarche est répétée jusqu'à l'obtention d'une opération virtuelle unique représentant le SFPM à la racine d'une arborescence.

#### **VIII.3.1. Règles de regroupement**

Le concepteur qui souhaite réaliser le regroupement des opérations de son SFPM a besoin de critères et de règles de regroupement pour garantir l'unicité de sa modélisation. Nous proposons pour cela deux règles de regroupement :

La **première règle** établit un ordre de priorité entre les contraintes en fonction de leur criticité. Une contrainte est d'autant plus prioritaire qu'elle est critique pour le fonctionnement du SFPM.

C'est le cas de la contrainte de chargement et de déchargement dont le respect est primordial pour la réalisation d'une opération. La contrainte d'exclusion est la moins prioritaire car elle est la moins critique, puisque l'arrêt d'une opération n'empêche en rien des opérations partageant une contrainte d'exclusion. Nous proposons l'ordre de priorité décroissant suivant :

- Contrainte de Chargement et de Déchargement.
- Contrainte de Procédure.
- Contrainte de Chargement et de Déchargement Partagés.
- Contrainte de Procédure Partagée.
- Contrainte d'Exclusion.

La **deuxième règle** propose de prendre en compte le regroupement impliquant le moins d'opérations si deux contraintes de même type sont en conflit. Cette règle permet d'obtenir des groupes d'opérations dont on peut calculer plus facilement l'état en fonction de l'état des composants.

### VIII.3.2. Exemples de mise en œuvre de la modélisation fonctionnelle

La mise en œuvre de la représentation fonctionnelle requiert :

- la spécification des opérations et de leurs contraintes,
- la spécification des transferts et de leurs contraintes,
- la spécification de l'ensemble des transferts associés à chaque opération.

Nous illustrerons la mise en œuvre de la modélisation fonctionnelle que nous proposons sur deux exemples. Le premier exemple (figure VIII.6) concerne une unité de production flexible que nous avons déjà présentée au chapitre VII (figure VII.1). Le deuxième exemple concerne la cellule flexible de l'École Centrale de Lille dont les nouvelles extensions nécessitent une approche de modélisation fonctionnelle.

Premier exemple : une unité de production flexible

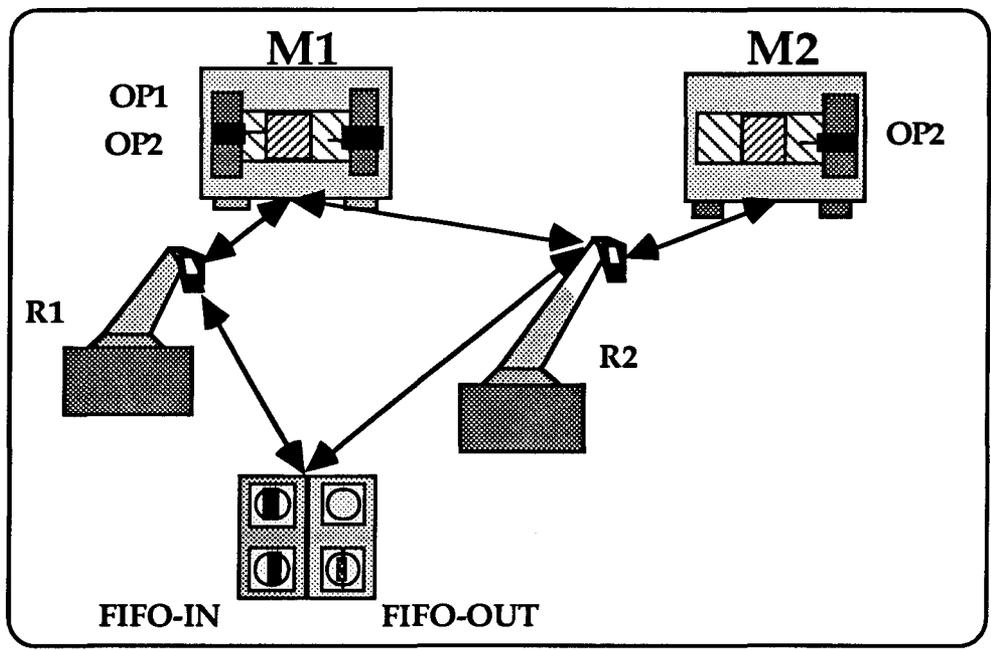


Figure VIII.6 : Exemple d'unité de production flexible

La figure VIII.7 résume la spécification fonctionnelle de notre exemple de cellule flexible.

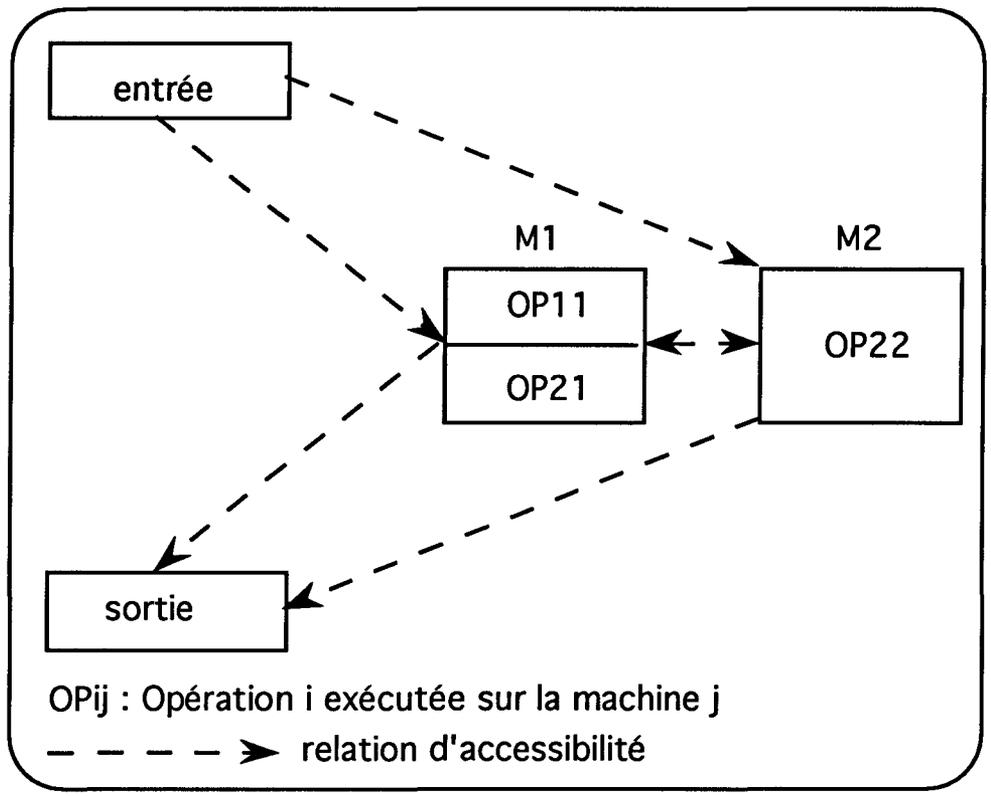


Figure VIII.7 : Les relations d'accessibilités

Cette spécification se fait en trois étapes :

La spécification des opérations

La cellule que nous présentons permet de réaliser deux types d'opérations, Op1 et Op2. L'opération Op2 peut être réalisée par M1 et M2. Ces deux versions de Op2 seront respectivement notées Op21 et Op22. Il est évident que Op1 et Op21 sont exclusives (contrainte d'exclusion) l'une de l'autre puisqu' elle partage le même lieu caractéristique.

La spécification des transferts

Les transferts possibles dans notre exemple sont : le transfert de l'entrée vers M1 (respectivement vers M2) noté E→M1 (respectivement E→M2), le transfert de M1 (respectivement M2) vers la sortie noté M1→S (respectivement M2→S) et enfin les transferts de M1 vers M2 et de M2 vers M1, notés M1→M2 et M2→M1.

Les transferts M2→M1 et M2→S sont exclusifs, car le tampon de sortie de M2 ne peut être utilisé que par un seul transfert à la fois. Il en est de même pour tous les transferts partageant la même extrémité.

La spécification des contraintes entre transferts et opérations

	Op1	Op21	Op22	E->M1	M2->M1	M2->S	M1->S	M1->M2	E->M2
Op1		CE		CC1	CC1		CD1	CD1	
Op21	CE			CC1	CC1		CD1	CD1	
Op22					CD2	CD2		CC2	CC2
E->M1	CC1	CC1			CE1			CE1	
M2->M1	CC1	CC1	CD2	CE1		CE1	CE1	CE1	CE1
M2->S			CD2		CE1		CE1	CE1	CE1
M1->S	CD1	CD1			CE1	CE1		CE1	
M1->M2	CD1	CD1	CC2	CE1	CE1	CE1	CE1		CE1
E->M2			CC2		CE1	CE1		CE1	

*Figure VIII.8 : Tableau initial des contraintes*

Les opérations Op1 et Op21 partagent une contrainte de chargement avec les transferts E→M1 et M2→M1 et une contrainte de déchargement avec les transferts M1→S et M1→M2. L'opération Op22 a une contrainte de chargement avec le transfert E→M2 et M1→M2 et une contrainte de déchargement avec M2→S et M2→M1. Nous noterons :

- $C1 = \{E \rightarrow M1, M2 \rightarrow M1\}$  l'ensemble des transferts de chargements de Op1 et Op21,

- $C2 = \{E \rightarrow M2, M1 \rightarrow M2\}$  l'ensemble des transferts de chargements de Op22,
- $D1 = \{M1 \rightarrow S, M1 \rightarrow M2\}$  l'ensemble des transferts de déchargement de Op1 et Op21,
- $D2 = \{M2 \rightarrow S, M2 \rightarrow M1\}$  l'ensemble des transferts de déchargements de Op22.

Nous remarquerons que le transfert  $M2 \rightarrow M1$  est à la fois un chargement pour Op1 et Op21 et un déchargement pour Op22. De même,  $M1 \rightarrow M2$  appartient à l'intersection de D2 et C1.

### Regroupement des opérations et des transferts

Le tableau des contraintes que propose la figure VIII.9 prend en compte les contraintes entre les opérations et chaque ensemble de chargement et de déchargement.

	OpV1	C1	D1	Op22	C2	D2
OpV1		CC1	CD1			
C1	CC1		CE1			
D1	CD1	CE1				
Op22					CC2	CD2
C2				CC2		CE1
D2				CD2	CE1	

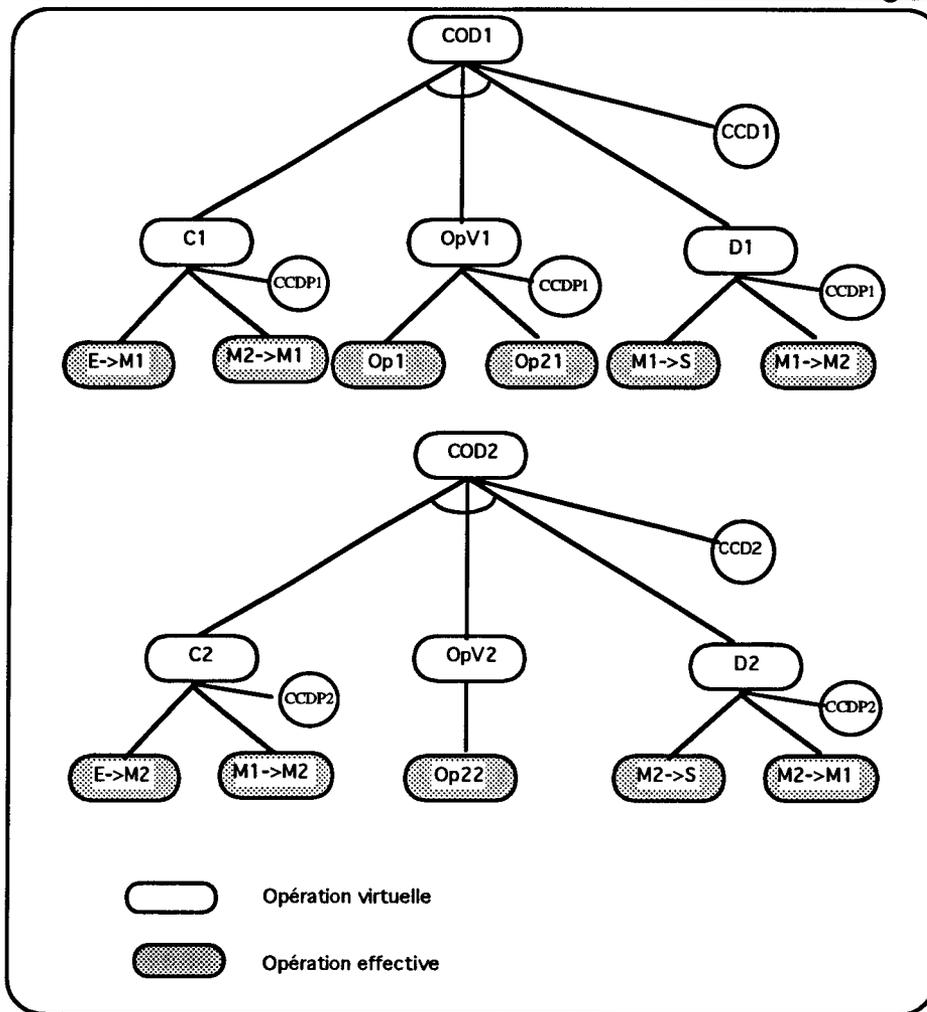
*Figure VIII.9 : Tableau des contraintes*

Les opérations Op1 et Op21 sont exclusives, car elles sont réalisées sur le même lieu caractéristique. Ces deux opérations partagent les mêmes contraintes de chargement et de déchargement. Elles sont donc liées par des contraintes de chargement et de déchargement partagés.

Les transferts  $E \rightarrow M1$  et  $M2 \rightarrow M1$  sont deux chargements possibles de Op1 et Op21. On les regroupe en un chargement virtuel. De même  $M1 \rightarrow S$  et  $M1 \rightarrow M2$  sont regroupés pour constituer un transfert de déchargement virtuel de Op1 et Op21.

Nous constatons que le transfert  $M1 \rightarrow M2$  apparaît dans deux contraintes différentes : ce transfert est à la fois un chargement pour Op1 et Op21 et un déchargement pour Op22.

Les ensembles {Chargement, Opération(s) et Déchargement} sont regroupés pour former des "COD" que nous représentons figure VIII.10.



*Figure VIII.10 : Représentation COD des opérations et des transferts*

Les arbres COD1 et COD2 correspondent à deux opérations virtuelles pour la cellule flexible. Ces deux COD partagent les mêmes transferts permettant de charger et de décharger la cellule à partir de ses tampons d'entrée/sortie. COD1 et COD2 sont alors regroupés pour former le COD\_système comportant une opération virtuelle OpV, un ensemble de transferts de chargement et un ensemble de transferts de déchargement (figure VIII.11).

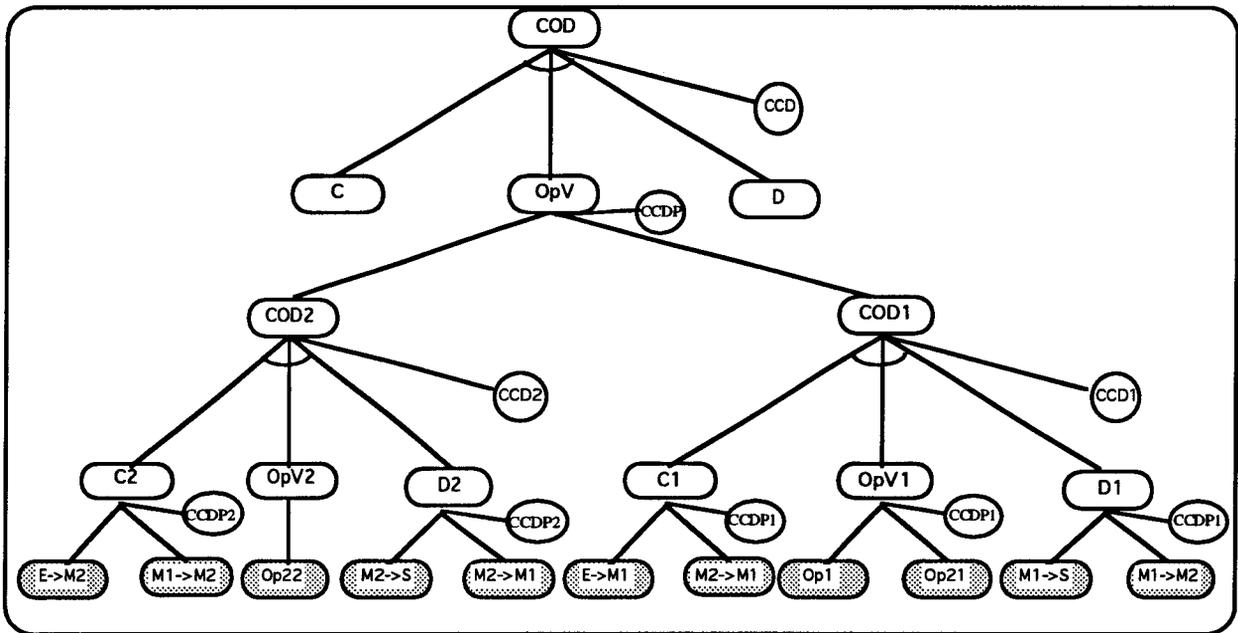


Figure VIII.11 : Arbre COD complet

La figure VIII.12 montre l'aspect itératif et générique de la représentation COD d'un SFPM. Le modèle COD du système regroupant COD1 et COD2 peut lui-même être intégré dans un SFPM plus grand.

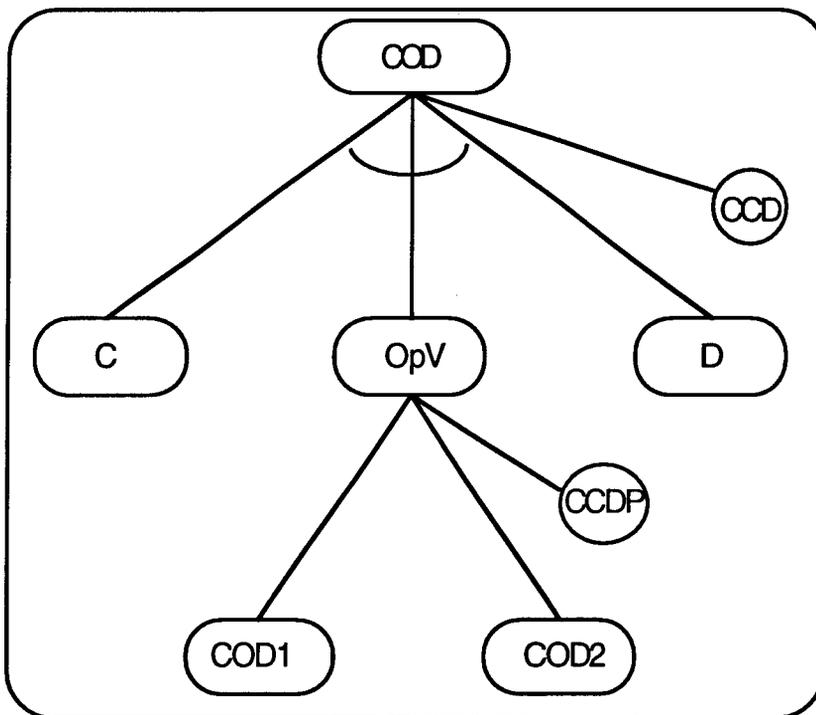
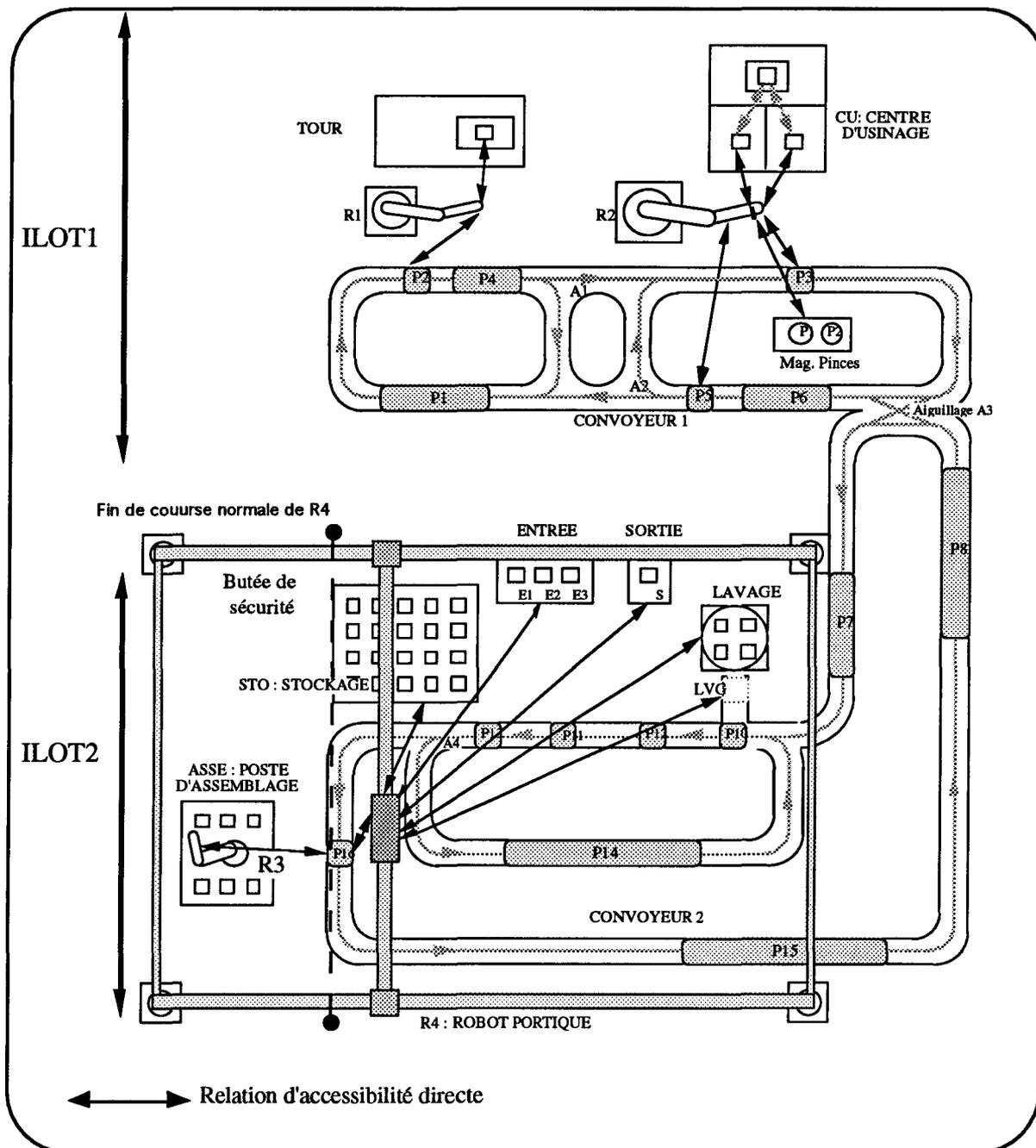


Figure VIII.12 : Aspect générique de la construction de l'arbre COD

## Deuxième exemple : la Cellule de l'Ecole Centrale de Lille



*Figure VIII.13 : La cellule flexible de l'EC Lille en 1995*

La figure VIII.13 présente l'atelier flexible expérimental de l'Ecole Centrale de Lille dans sa configuration actuelle. Cet atelier est constitué de deux systèmes flexibles connexes appelés respectivement ILOT1 et ILOT2. Chaque îlot comporte des postes de travail répartis autour d'un convoyeur de palettes central. Un aiguillage croisé A3 permet de connecter les convoyeurs des deux îlots. Les pièces brutes sont introduites par le robot R4 sur l'ILOT2 à partir d'un tampon d'entrée. Ces pièces peuvent être acheminées vers l'îlot 1 pour être usinées sur le TOUR et/ou le CENTRE D'USINAGE (CU). Ensuite, elles sont routées vers l'îlot 2 pour être, soit directement

sorties sur le tampon de SORTIE, soit lavées, assemblées puis stockées sur le stockeur (STO).

Les relations d'accessibilité entre les différents lieux opératoires de cet atelier sont représentées par des doubles flèches sur la figure IV.13. R1 assure le chargement et le déchargement du tour. R2 est lié au centre d'usinage. Le robot R3 est un robot d'assemblage qui assure en outre le chargement et le déchargement du poste d'assemblage. Le robot portique R4 assure le chargement et le déchargement de tout le système. Il assure aussi le chargement et le déchargement de la station de lavage et du stockeur.

En cas de panne de R3, le robot portique R4 peut réaliser le chargement et le déchargement de la station d'assemblage ainsi que certaines opérations d'assemblage (limitées par le coût des préhenseurs). Ce remplacement de R3 par R4 ne peut être autorisé, pour des raisons de sécurité, qu'en cas de panne et de retrait de R3. Quand ce dernier est en marche normale, une fin de course limite l'évolution de R4.

[Bois, 91] propose une représentation de l'atelier de l'EC Lille dans sa configuration de 1991. Cette ancienne configuration limitait le rôle de R4 au strict chargement et déchargement des postes de stockage et de lavage. Cette représentation est impossible à réaliser dans la configuration actuelle. En effet, le robot portique peut assurer des tâches de natures différentes : il exécute des transferts mais il peut réaliser quelques opérations d'assemblage. De plus R4 intervient à des niveaux différents de la cellule : il charge et décharge à la fois le poste de lavage et l'atelier tout entier.

L'approche fonctionnelle que nous proposons permet de représenter la complexité introduite dans l'atelier flexible de l'EC Lille dans sa nouvelle configuration. La figure VIII.14 illustre le résultat des deux premières étapes de notre approche : elle représente l'ensemble des opérations et des transferts entre lieux caractéristiques de l'atelier. La troisième étape va permettre de dégager l'ensemble des transferts de chargement et de déchargement de chaque opération. On notera par exemple  $C_{tour} = \{Ei \rightarrow tour, ASSE \rightarrow tour, CU \rightarrow tour, LVG \rightarrow tour, STO \rightarrow tour, Pi \rightarrow tour\}$  l'ensemble des transferts de chargement du tournage et  $D_{tour} = \{tour \rightarrow Ei, tour \rightarrow ASSE, tour \rightarrow CU, tour \rightarrow LVG, tour \rightarrow STO, tour \rightarrow Pi\}$  l'ensemble des transferts de déchargements du tournage.

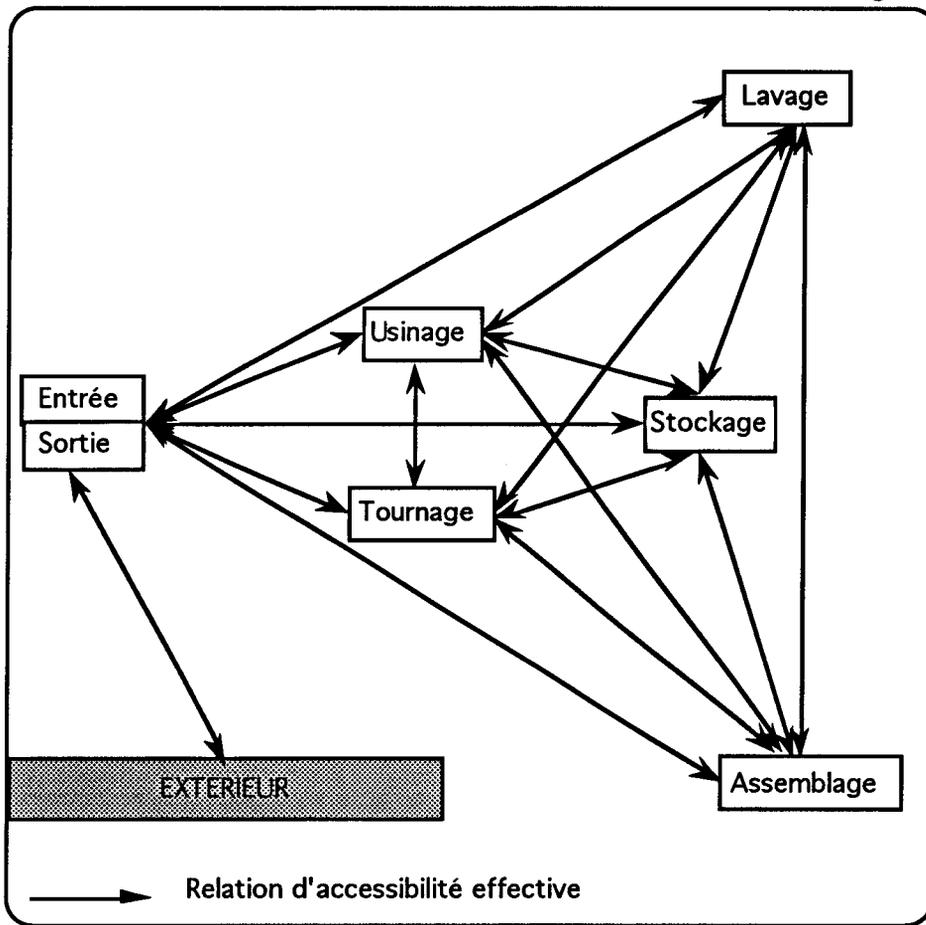


Figure VIII.14 : Spécification fonctionnelle de la cellule de l'EC Lille

Ces spécifications vont permettre de construire le COD de chaque opération. La figure VIII.15 illustre cette construction pour l'opération de tournage. Les COD ainsi obtenus sont liés par le partage d'une contrainte de chargement et de déchargement. Ce résultat est illustré par la figure VIII.16.

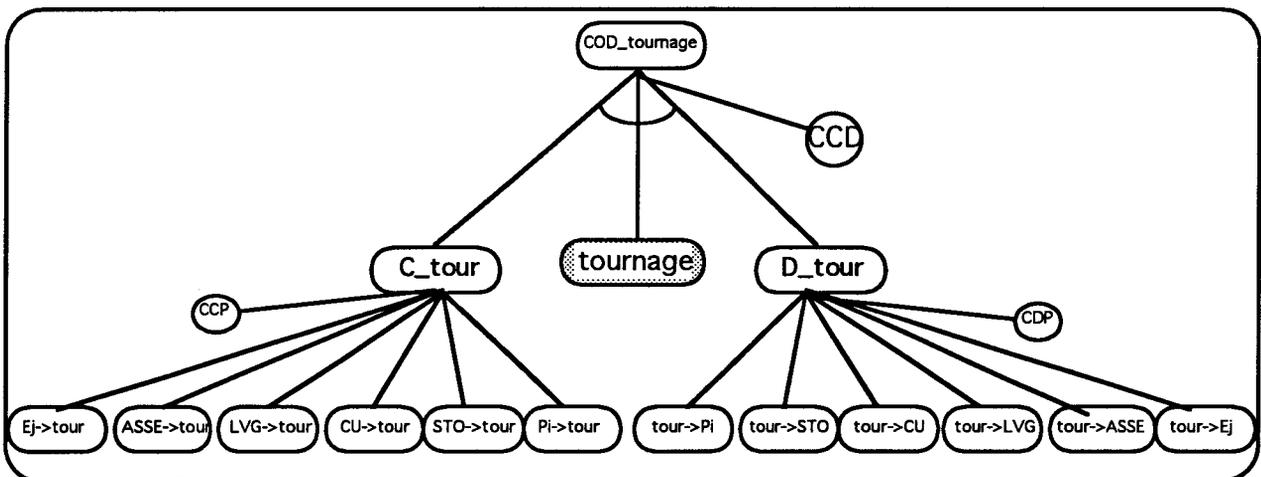
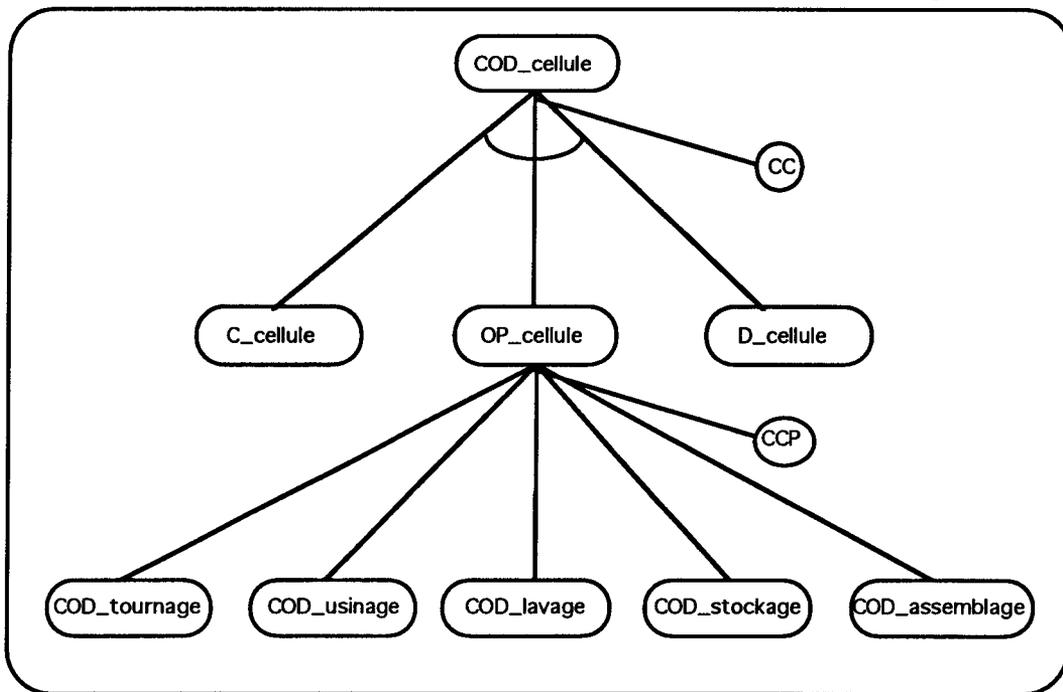


Figure VIII.15 : Représentation COD du tournage



*Figure VIII.16 : L'arbre COD de la cellule flexible de l'EC Lille*

### VIII.3.3. Intégration de la dimension matérielle du SFPM : les ressources physiques

La représentation fonctionnelle permet de modéliser la relation entre les différentes fonctions d'un SFPM. Cette représentation ne permet cependant pas de lier les opérations et les transferts aux ressources d'exécution. Pour remédier à cet inconvénient, nous proposons de compléter l'arbre COD d'un SFPM par un niveau matériel. Ce dernier niveau permet de spécifier la configuration de la (ou des) ressources capables de réaliser chaque opération effective. Les liens entre les opérations et leurs ressources d'exécution porte en plus un label. Ce label exprime une préférence d'utilisation d'une machine plutôt qu'une autre pour réaliser une opération.

Reprenons l'exemple de la cellule flexible présentée en introduction de ce chapitre pour illustrer notre propos. Les transferts  $E \rightarrow M1$  et  $M1 \rightarrow S$  du regroupement COD1 de la figure VIII.10 sont réalisés par le robot R1 en fonctionnement normal. Mais le robot R2 peut le remplacer en cas de panne. Le lien entre  $E \rightarrow M1$  (respectivement  $M1 \rightarrow S$ ) et R1 est prioritaire sur le lien entre  $E \rightarrow M1$  respectivement  $M1 \rightarrow S$ ) et le robot R2 ; d'où la présence des labels de priorité 1 et 2 sur la figure VIII.17.

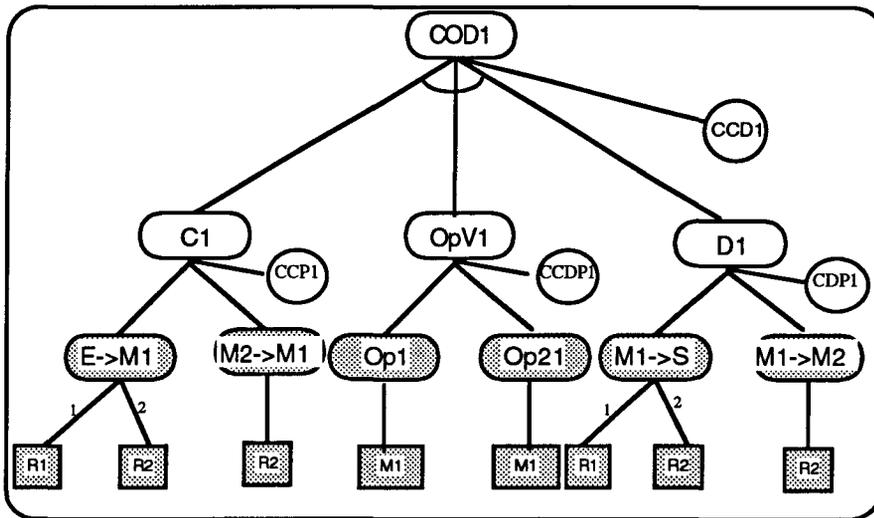


Figure VIII.17 : Intégration des ressources à la représentation COD

La même démarche peut être appliquée à la cellule flexible de l'EC Lille. La figure VIII.18 indique que le transfert entre le tour et le poste d'assemblage est composé d'un transfert entre le tour et P2 suivi d'un transfert entre P2 et P16 et d'un transfert entre P16 et le poste d'assemblage. La figure précise la ressource qui réalise chacun des transferts. Elle indique notamment que le robot R3 est utilisé en priorité par rapport à R4 pour réaliser le transfert P16→ASSE.

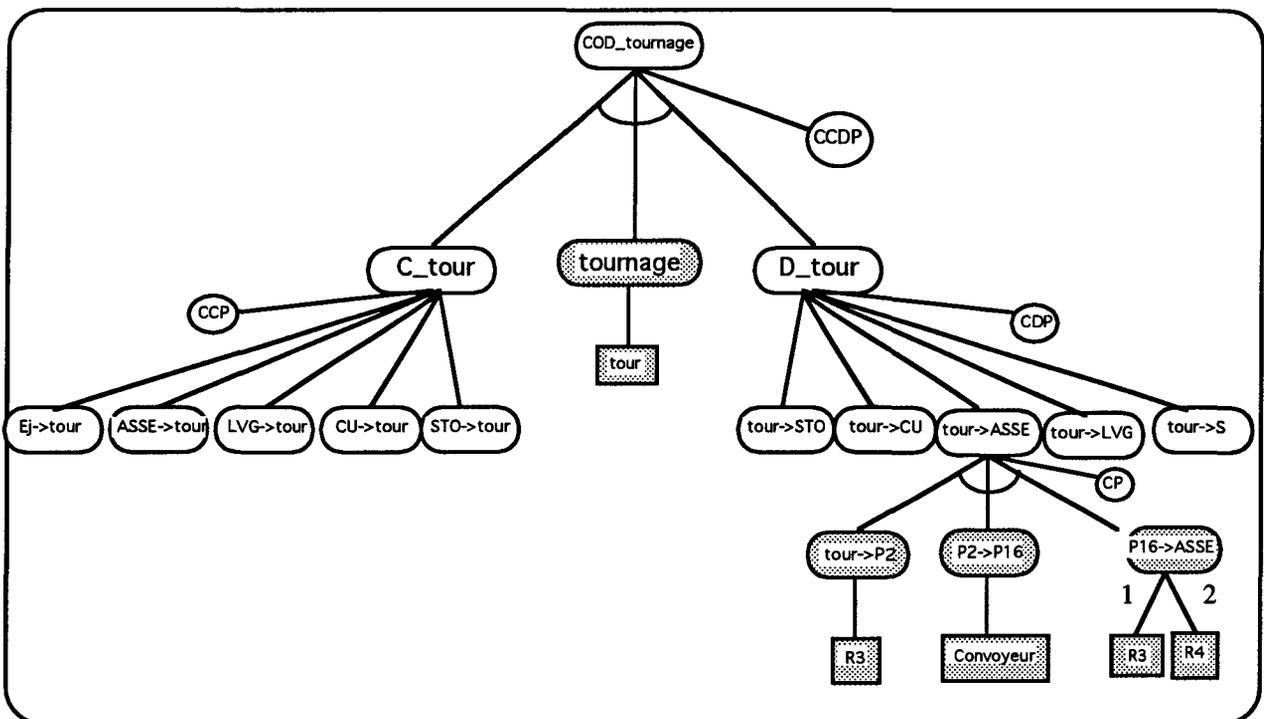


Figure VIII.18 : Décomposition d'un transfert

**CONCLUSION [KERMADE ET AL, 96]**

La flexibilité des SFPM est un des principaux facteurs de prise de décision. Cette propriété permet à ce type de systèmes de s'adapter rapidement à la production de petites et moyennes séries ainsi qu'à la prise en compte des aléas de production (modifications de production, défaillances). Il existe plusieurs types de flexibilité :

- flexibilité de gamme (ordre des opérations),
- flexibilité d'affectation (ressources pouvant réaliser une opération),
- flexibilité intra-ressource,
- flexibilité du système de transport (routage),
- flexibilité de production (gamme de produits),
- flexibilité de produit (réaliser un nouveau produit).

La flexibilité d'un SFPM conduit à la mise en œuvre :

- de gammes logiques flexibles, partiellement ordonnées et pour lesquelles les routages sont généralement sur-déterminés,
- de ressources simples (un seul lieu d'accueil par produit) ; il faut donc partager ces ressources entre plusieurs utilisateurs. Leur affectation est décidée en temps réel,
- de ressources complexes gérant simultanément plusieurs lieux d'accueil par produit (cas d'une ressource virtuelle, d'un îlot, ...) et nécessitant un modèle dynamique d'analyse et prévision du comportement.

Dans ce cadre d'hypothèses, il faut considérer deux classes de problèmes concernant la relation (fonction  $\leftrightarrow$  opération/ressource) :

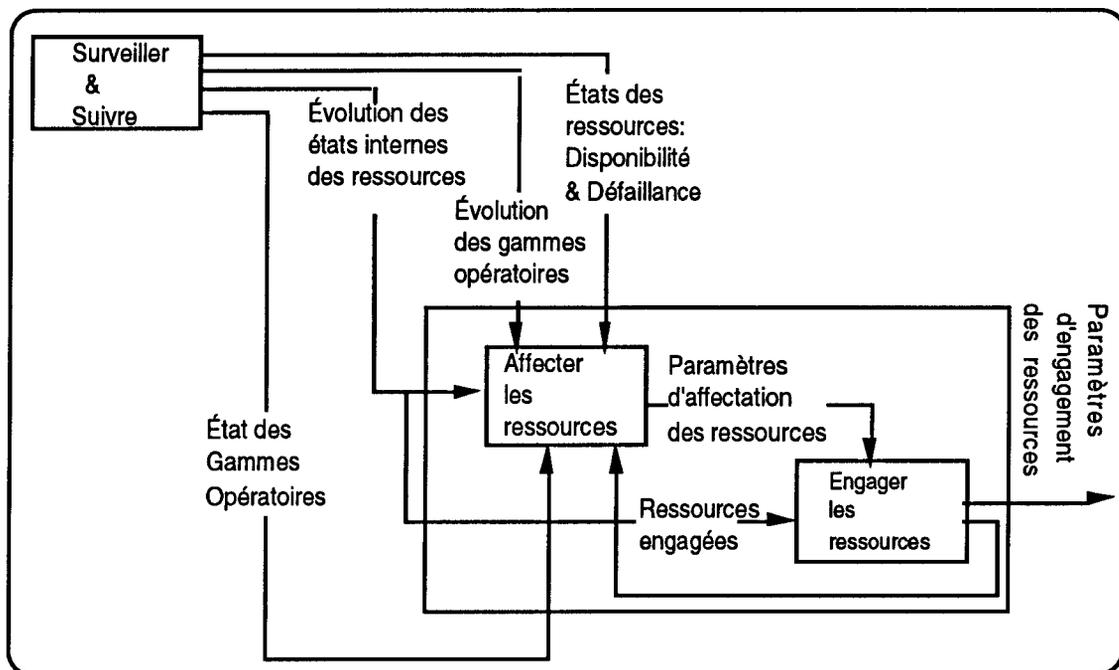
- **l'indéterminisme** : à une "fonction" invoquée par la gamme, correspondent plusieurs allocations de ressources potentielles. En fait, différentes solutions sont candidates à la réalisation de la fonction sollicitée par un produit dans un état donné de sa gamme.
- **le conflit** : plusieurs produits de gammes logiques différentes en cours d'exécution peuvent solliciter des fonctions qui ne peuvent être réalisées que sur un nombre limité de ressources

Dans l'approche CASPAIM, on a besoin de disposer à tout instant :

- de l'arborescence des fonctions associées à une configuration d'exploitation donnée,
- des gammes logiques associées à ces fonctions,
- des gammes opérationnelles qui tiennent compte des affectations de ressources préétablies et qui expriment les conflits et les indéterminismes qui sont évoqués ici.

Il semble toutefois difficile d'établir à tout instant la double correspondance (fonction  $\rightarrow$  ressources) et (ressources  $\rightarrow$  fonction) car l'affectation d'une ressource à une fonction n'est pas prédéterminée en général, en raison de la flexibilité procédurale. Cette affectation dépend de l'évolution dynamique des produits et de l'état du procédé. En effet, il faut, pour décider de l'allocation d'une ressource (figure VIII.19), disposer :

- de l'état instantané d'avancement des produits dans toutes les gammes,
- de l'état instantané de disponibilité ou de défaillance des ressources configurées,
- des affectations déjà engagées au niveau supervision/conduite.



*Figure VIII.19 : L'allocateur des ressources*

Dans le sens inverse, la défaillance d'une ressource introduit de manière préemptive l'indisponibilité ou la dégradation d'une ou plusieurs fonctionnalités du processus. Il est donc essentiel de disposer d'une correspondance synchrone entre l'état des ressources configurées au sein d'un graphe structurel des ressources et une instance du graphe des fonctions générales assurées par le système dans une configuration d'exploitation que nous pourrions appeler "graphe des fonctions

La configuration des ressources peut être dégradée du fait d'une défaillance ; le graphe des fonctions disponibles de la configuration initialement considérée doit alors être immédiatement mis à jour.

Dans ce cas la disparition ou la dégradation d'au moins une fonction intervenant dans une gamme induira l'indisponibilité ou la dégradation de la gamme (configuration réduite).

Ces mécanismes peuvent être réalisés grâce à la mise en œuvre conjointe, cohérente et synchrone des différents modèles proposés par CASPAIM, tant pour la coordination que pour la surveillance en nous appuyant sur la représentation COD que nous proposons dans cette partie.

Dans la dernière partie de notre mémoire, nous utiliserons la représentation COD (Chargement, Opération et Déchargement) pour gérer deux flux d'informations dans le SFPM : un flux ascendant lié à la surveillance et à la mise à jour de l'état du système et un flux descendant lié à la commande et à l'exploitation du SFPM à travers les modèles de la coordination de CASPAIM. Nous utiliserons des outils synchrones pour spécifier la supervision et la gestion des SFPM grâce au modèle COD.

# **PARTIE C : APPROCHE SYNCHRONES DE LA GESTION DES MODES**

Chapitre IX : Les langages synchrones graphiques

Chapitre X : Application de la représentation COD à l'exploitation et à la supervision des SFPM

Chapitre XI : Formalisation des flux d'information dans un SFPM

## INTRODUCTION

Les Systèmes Flexibles de Production Manufacturière (SFPM) sont des systèmes à événements discrets en interaction permanente avec leur environnement. Ils présentent à la fois des caractéristiques transformationnelles et réactives. Les SFPM sont transformationnels car il existe une part importante de traitement comme le suivi des produits, le respect des consignes de production, l'exploitation de la flexibilité, etc.

Les SFPM entrent aussi dans la catégorie des systèmes dits "réactifs temps-réel", car leurs réactions sont soumises à des contraintes temporelles souvent fortes. En outre, la représentation des SFPM nécessite de fortes interactions entre différents modèles. En fait, nous nous trouvons en présence d'un cas particulier de système réactif puisqu'il y a deux niveaux de réactivité : la réactivité de l'action et la réactivité de l'information. Cette constatation nous a amenés à retenir une approche synchrone [Berry et al, 91] dédiée aux systèmes réactifs pour la gestion des modes. Une telle approche se justifie en phase de spécification et de conception de la gestion des modes car elle simplifie les problèmes de communications et de diffusions. Elle permet de s'affranchir des problèmes qui ne doivent être traités qu'en phase de développement et d'implantation et qui sont plus liés aux outils asynchrones qu'à la gestion des modes. C'est le cas notamment de la gestion informatique des interruptions pour spécifier la préemption d'une fonction dans un SFPM.

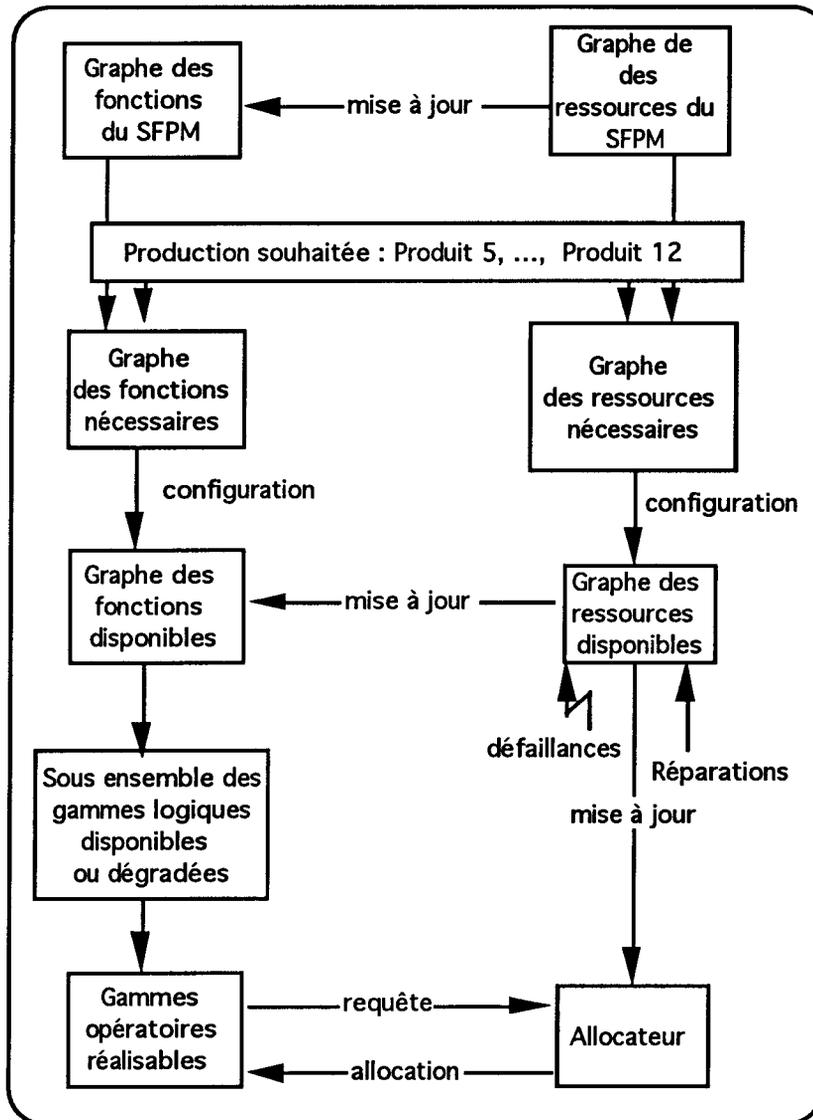
La gestion des modes d'un SFPM fait intervenir des comportements réactifs plus complexes que ceux pris en compte par le GEMMA ; il faut choisir des modèles synchrones plus riches que le GRAFCET. En effet, la gestion des modes doit pouvoir traiter en temps réel les évolutions d'un SFPM. Elle doit être capable de préempter les actions du système, c'est-à-dire interrompre un traitement avant sa fin normale en cas d'évolution dangereuse. Une décision de préemption prise au niveau des couches hautes de la structure hiérarchique, doit être transmise très rapidement aux couches basses où la contrainte temporelle est plus forte. De même, la détection de certaines défaillances au niveau des couches basses doit être transmise rapidement vers les centres de décision afin d'empêcher leur propagation dans le système de production.

Nous voulons insister ici sur le caractère bi-directionnel des flux d'information :

- du niveau ressources vers le niveau supervision : une défaillance au niveau des ressources doit conduire de façon réflexe à une mise à jour de l'état des fonctions disponibles puis des gammes contaminées. Le caractère synchrone des échanges est impératif afin d'éviter la propagation d'une défaillance et de fournir à l'exploitant du système une information précise et rigoureuse sur l'état courant de son SFPM ainsi que les anomalies détectées et la configuration qui reste disponible.

- du niveau supervision vers les ressources : lorsque l'exploitant décide, au niveau supervision des gammes, de lancer ou d'arrêter une ou plusieurs gammes afin d'adapter la production et de

recouvrer les fonctions défaillantes en reconfigurant le SFPM. On pourra déduire la structure du graphe réduit des fonctions nécessaires du graphe général des fonctions du SFPM représenté par le modèle COD. Il sera possible de déduire de ce graphe l'ensemble des ressources globalement utilisées et initialisées sans les affecter a priori. L'arrêt d'une gamme devra, par ailleurs, conduire, en phase d'exploitation, à la désactivation des fonctions correspondantes et donc à la fois à la mise à jour du graphe de fonctions nécessaires à cette nouvelle configuration ainsi qu'à l'arrêt éventuel des ressources si toutefois elles n'étaient pas requises par d'autres fonctions.



*Figure C: Configuration et allocation des ressources*

Le premier chapitre de la Partie C va présenter les outils synchrones, en particulier les formalismes StateCharts et Argos qui se distinguent par leur aspect graphique et leur relative facilité d'utilisation. Notre choix s'est porté sur le formalisme Argos [Kermad et al, 94a et 94b] car il impose une rigueur d'écriture et propose un interfaçage avec des outils de vérification et de validation formelle. En outre, l'utilisation de StateCharts pour la spécification des modes de

Partie C Approche synchrone de la gestion des modes  
fonctionnement des SFPM dans le cadre d'un projet DRED [Kermad et al, 93a] a révélé des difficultés que le formalisme Argos a permis de résoudre.

Le deuxième chapitre va lier la représentation COD des SFPM à leur exploitation en traduisant les modèles de la coordination utilisés dans CASPAIM dans une représentation arborescente. La hiérarchie des gammes opératoires qui en découle servira alors à la construction de l'architecture du contrôle/commande présentée par la figure V.6.b. Ce chapitre montrera ensuite comment gérer la représentation de la disponibilité des fonctions et des opérations du SFPM à tous les niveaux de la hiérarchie proposée.

Enfin, le dernier chapitre va mettre en œuvre le formalisme Argos pour exprimer les aspects ascendant et descendant de l'information dans un SFPM. Cette mise en œuvre sera illustrée par la formalisation de l'état de disponibilité (normale, dégradée ou hors service) et l'état d'activité (off ou actif) d'une fonction, quelque soit son niveau d'abstraction. La structure obtenue pourra être étendue pour représenter le mode de fonctionnement (automatique, pas à pas ou manuelle) et le mode d'exploitation (production ou maintenance).

## **CHAPITRE IX : LES LANGAGES SYNCHRONES GRAPHIQUES**

Les langages synchrones sont dédiés à la programmation des systèmes temps réels, et plus généralement des systèmes réactifs (i.e. un système qui réagit à des entrées issues de son environnement en produisant des sorties vers cet environnement). D'un point de vue pratique, "les systèmes réactifs doivent réagir rapidement pour ne pas perdre d'information et présenter les réactions en temps utile" [André et al, 93]. C'est dans ce contexte qu'ont été développés Esterel, Lustre, Signal, StateCharts, Argos et, plus récemment, SynChart. Ces langages ont en commun l'hypothèse synchrone qui considère que toutes les actions sont instantanées. La conséquence immédiate est que ces langages ont la capacité de traiter la préemption. On peut classer ces langages en deux catégories en fonction de leur style de programmation :

- Les langages de **type déclaratif** comme **SIGNAL** [Le Guernic et al, 91] et **LUSTRE** [Halbwachs et al, 91]. Ces langages permettent d'implémenter le comportement du système à l'aide d'équations dynamiques.
- les langages de **type impératif** comme **ESTEREL** [Boulanger, 91], **ARGOS** [Maraninchi, 90] ou **STATECHARTS** [Harel, 87]. Les relations entre entrées et sorties du système doivent être décrites de manière explicite. **ESTEREL** se caractérise par la notion de temps multiforme. Ce langage dispose par ailleurs d'un support de développement et de validation d'application plus riche que les autres langages synchrones.

Notre choix s'est porté sur les langages synchrones graphiques en raison de leur ergonomie et de la préférence industrielle pour les outils graphiques. Les langages graphiques sont une extension des automates à états finis. Les principales extensions portent sur la hiérarchisation des états et l'expression du parallélisme. Ces langages sont destinés à la description du comportement des systèmes complexes. StateCharts est le premier de ces langages. Il est fondé sur la notion d'automates à états et transitions labélisées.

### **IX.1. Présentation de StateCharts**

StateCharts est un outil de spécification conçu pour résoudre les problèmes de modélisation et de conception des systèmes temps-réel complexes. Ce paragraphe expose d'abord les entités syntaxiques fondamentales du formalisme StateCharts à l'aide de représentations graphiques et des interprétations associées. De plus amples détails sur la syntaxe et la sémantique pourront être trouvés dans [Harel, 87].

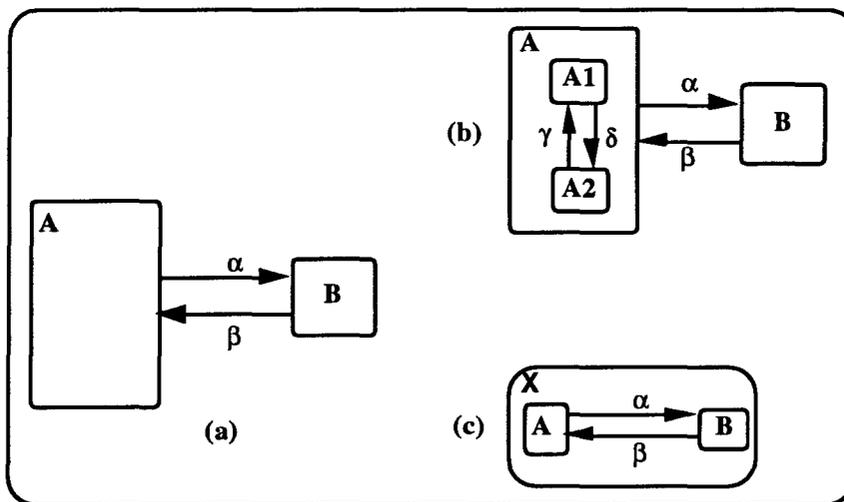
Ce paragraphe présente ensuite certains problèmes de syntaxe et de sémantique qui limitent l'usage des StateCharts. Ces problèmes ont justifié le développement de plusieurs variantes dont Argos que nous proposons d'utiliser.

Le formalisme StateCharts résulte d'une extension des machines à états finis (dont il hérite du mécanisme de séquençement pour un niveau d'abstraction) et de Higraphes (dont il reprend les concepts de hiérarchie et de parallélisme). Ce formalisme est fréquemment caractérisé par la formule suivante :

**StateCharts = diagramme d'états + hiérarchie + parallélisme + communication**

### IX.1.1. La hiérarchie

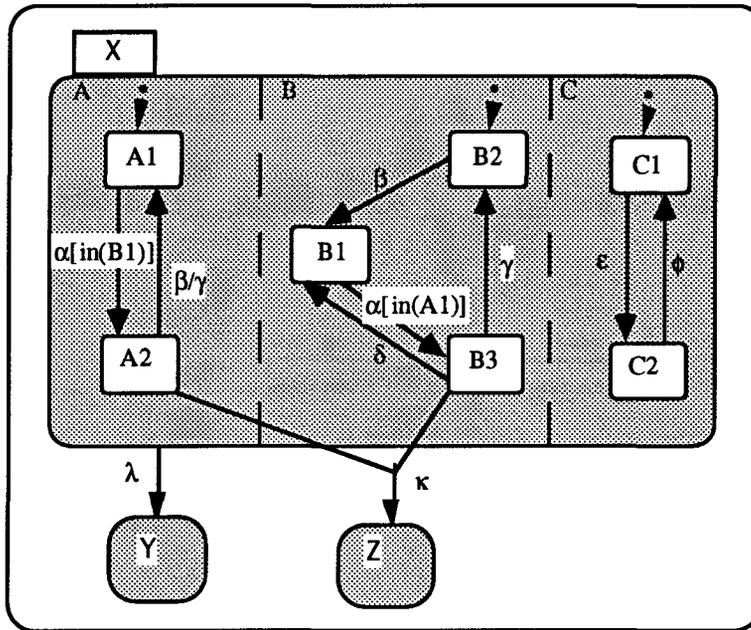
Le concept de hiérarchie introduit les notions d'abstraction et d'affinement. Ces mécanismes constituent les fondements d'une approche structurée. Ils facilitent le travail de description de systèmes de taille et de complexité importantes. Les parties (b) et (c) de la figure IX.1 illustrent les processus d'abstraction et d'affinement.



*Figure IX.1 : La hiérarchie*

### IX.1.2. Le parallélisme

La faculté à décrire le parallélisme sans subir une explosion combinatoire est un des avantages majeurs de StateCharts. L'exemple de la figure IX.2 propose un macro-état X qui englobe les sous-états A, B, et C évoluant en parallèle. Cela signifie que si X est actif, tous ses sous-états le sont. A, B et C sont activés ou désactivés simultanément avec l'entrée ou la sortie de X.



*Figure IX.2 : Le parallélisme*

Sur ce diagramme, C évolue indépendamment des états A et B. StateCharts offre la possibilité d'exprimer des synchronisations puisque les évolutions de A et de B sont liées l'une à l'autre. Pour passer de A1 à A2, et de B1 à B3, les conditions [in(A1)] et [in(B1)] (synchronisation conditionnelle) doivent être vérifiées, et l'occurrence de l'événement  $\alpha$  effective. Enfin, si l'événement  $\lambda$  survient, le système évoluera vers l'état Y quelque soit la configuration des états actifs. Par contre, l'événement  $\kappa$  ne pourra autoriser le passage de X vers Z que si A2 et B3 sont actifs.

### IX.1.3. La communication

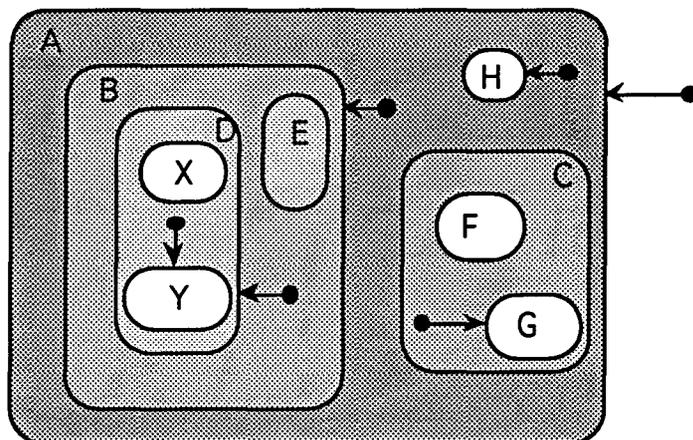
Lorsqu'un message est généré, il est instantanément diffusé. Le modèle événement/action est synchrone, c'est à dire qu'un événement et l'action qu'il produit doivent être considérés simultanés. Cette hypothèse de synchronicité est à rapprocher de celle du langage Esterel [Berry et al, 87].

La forme générale d'un label associé à une transition est "e[C]/a" où :

- "e" représente l'événement qui provoque le tir de la transition,
- "C" représente la condition (de type booléen) qui autorise le tir de la transition,
- "a" représente l'action lancée par le franchissement de la transition. Cette action peut être soit un événement généré utile au déclenchement d'une autre transition (synchronisation événementielle), soit un message destiné à la partie opérative (figure IX.2)

## IX.1.4. L'entrée dans un état

Le formalisme StateCharts offre deux possibilités d'entrer dans un état : l'entrée par défaut et l'entrée historique. L'entrée par défaut indique le sous-état vers lequel le système doit se diriger lorsqu'il entre dans un macro-état. Cette construction est l'équivalent du cercle doublé pour les machines à états ou du marquage initial en réseau de Petri. Dans la figure IX.3, quand on entre dans A, on entre particulièrement dans B ensuite D et enfin Y.



*Figure IX.3 : L'entrée par défaut*

L'entrée historique spécifique à StateCharts est symbolisée par un arc sur un cercle contenant la lettre H. Ce mécanisme permet d'entrer le dernier état actif d'un macro-état. Cette notion peut être très utile dans les mécanismes de reprise mais elle présente l'inconvénient de limiter les possibilités de validation formelle de la spécification.

## IX.1.5. Les inconvénients de StateCharts

Le formalisme StateCharts est très riche. Il permet une grande liberté dans la spécification des systèmes. Cette liberté pose le problème de la validation formelle des spécifications. [Beeck, 94] propose une étude très intéressante des problèmes de StateCharts ainsi qu'une comparaison de 20 variantes de ce formalisme. En effet, dès la publication de [Harel, 87], de nombreuses recherches ont démarré afin d'améliorer la syntaxe et la sémantique de ce nouveau formalisme. Ces recherches ont été motivées, d'une part, par le formidable pouvoir de StateCharts pour spécifier des systèmes hiérarchisés et parallèles d'une manière claire et intuitive, et d'autre part par les problèmes que sa sémantique incomplète, sa syntaxe très riche et son approche synchrone ont posés. Les principaux problèmes soulevés sont : l'hypothèse de synchronisme parfait, les transitions inter-niveaux, les problèmes de causalité, le déterminisme, la priorité d'exécution des transitions, l'auto-interruption et enfin l'interruption préemptive et non-préemptive.

Parmi les variantes étudiées par [Beeck, 94], Argos nous a paru la mieux adaptée à nos besoins de spécification et de traduction des modèles de représentation des SFPM. En effet, il corrige les

défauts de StateCharts qui ont alourdi la spécification des modes de fonctionnement des systèmes de production dans [Kermad et al, 93a]. La deuxième partie de ce chapitre présente ce langage et montre l'application que nous en proposons dans la modélisation des SFPM.

## **IX.2. Présentation de Argos**

ARGOS [Maraninchi, 90] appartient à la catégorie des langages synchrones graphiques. Il s'inspire beaucoup de StateCharts. C'est le langage que nous proposons d'utiliser en raison des possibilités de validations formelles qu'il offre.

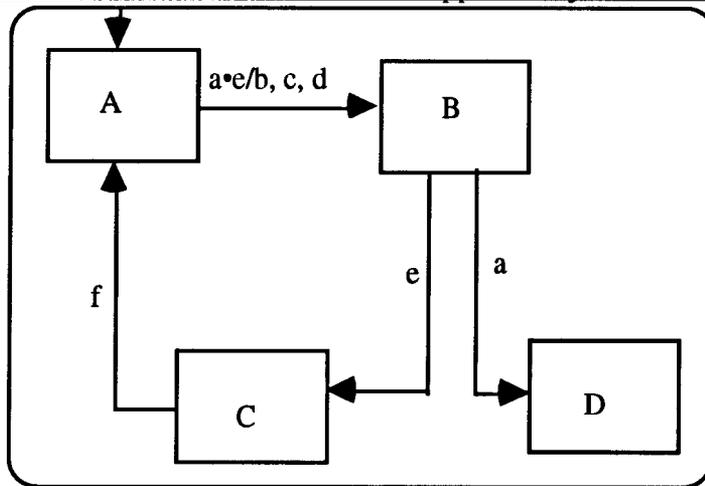
La syntaxe d'ARGOS est basée sur le formalisme des graphes de haut niveau. Sa sémantique utilise les algèbres de processus. Les automates forment la notion fondamentale du langage. Leurs décompositions parallèles ou hiérarchiques utilisent des opérateurs algébriques. La normalisation complète du langage hérite des notions des algèbres de processus classiques telles que CCS, ou des langages de programmation tels que ESTEREL ou STATECHARTS. Le formalisme ARGOS permet de définir directement les états intrinsèques d'un système complexe. Les relations entre les composants ne requièrent aucune représentation d'état supplémentaire. Le système ARGONAUTE [Maraninchi, 89], développé par le Laboratoire de Génie Informatique de Grenoble, est conçu pour décrire, spécifier et vérifier les systèmes réactifs. Il est basé sur le langage ARGOS. Il permet à l'utilisateur de préciser des propriétés au moyen de formules de logique temporelle, de produire un modèle qu'il peut évaluer et qui permet de simuler l'exécution du système, en utilisant des formes graphiques externes.

Cet outil permet d'exprimer les mécanismes essentiels dans la modélisation des SFPM. Ces mécanismes sont la hiérarchie, le parallélisme, la synchronisation de processus et de machines, la terminaison qu'elle soit interne au système de production ou externe, c'est-à-dire, préemptive. Dans les paragraphes qui suivent, nous décrivons ces mécanismes en les illustrant par des exemples simples.

### **IX.2.1. Les Automates**

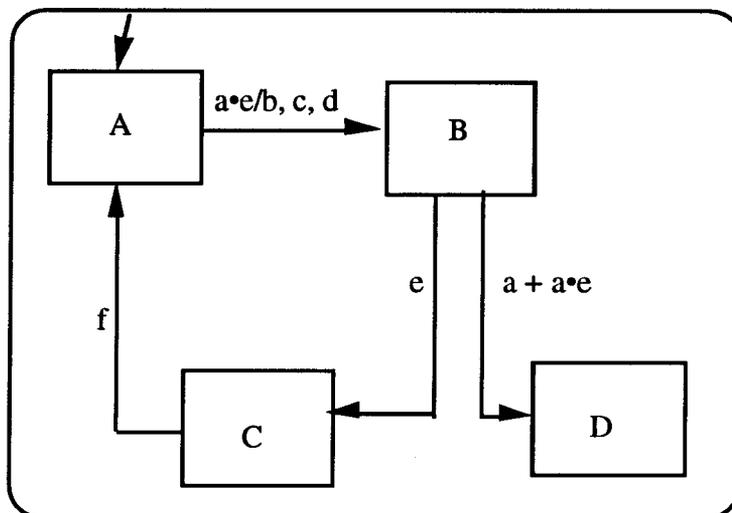
ARGOS est caractérisé par la notion d'automates. Ces automates sont décrits par des états explicites et des transitions étiquetées (figures I.4 et I.5). L'état initial est marqué par une petite flèche sans étiquette. Une comparaison avec STATECHARTS est faite dans [Maraninchi, 92]. Les étiquettes des transitions ("a", "b", ...) correspondent à des événements ou des signaux. L'absence de "a" sera représentée par sa négation notée " $\neg a$ ".

Dans les figures I.4 et I.5, (a•e/b, c, d) signifie que l'occurrence simultanée des événements "a" et "e" déclenche l'émission des signaux "b", "c" et "d" en provoquant le passage de l'état "A" à l'état "B".



*Figure IX.4 : Exemple d'Automate non-déterministe*

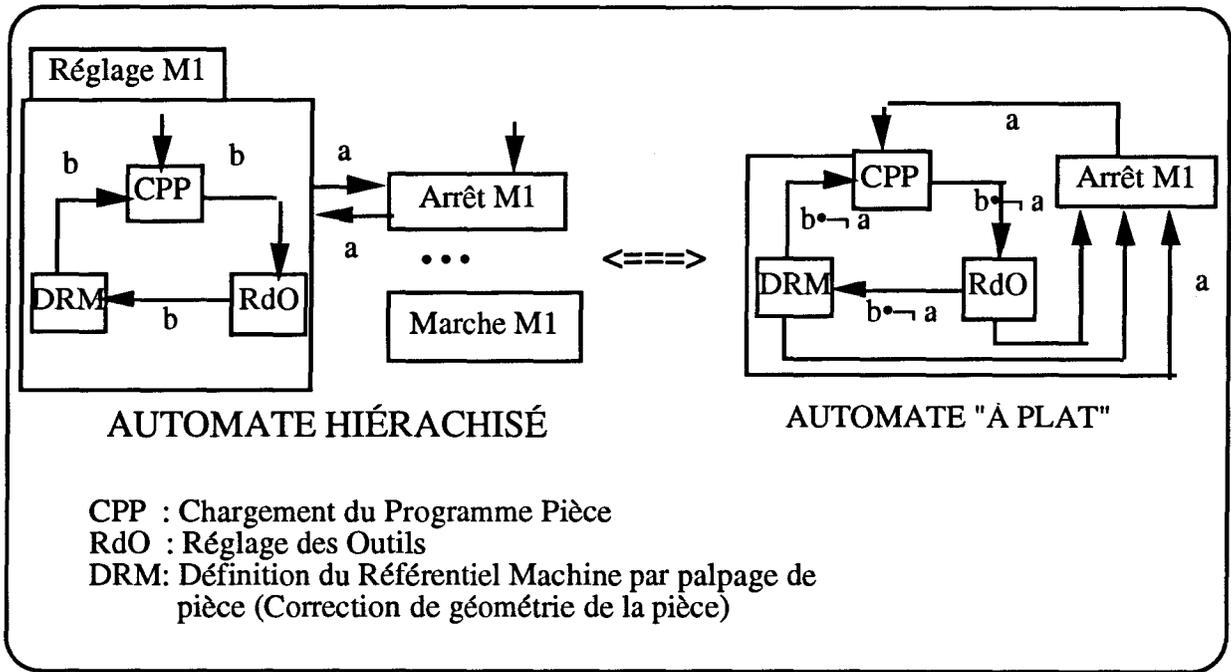
Le compilateur Argos rejettera l'automate de la figure IX.4. En effet, cet automate présente un indéterminisme. Si les événements "e" et "a" sont reçus simultanément (hypothèse prise dans Argos), l'état B étant actif, l'état actif suivant reste indéterminé puisque la transition "e•a" n'est pas prévue. Ceci est une erreur de conception. L'automate de la figure IX.5 corrige cette erreur en prévoyant la transition BD à l'occurrence de "a" ou "a•e" ( $a+a•e$ ). La transition B-C est alors implicitement interprétée comme déclenchée par l'événement " $e \rightarrow a$ "



*Figure IX.5: Exemple d'Automate déterministe*

### IX.2.2. La hiérarchie

C'est la capacité de décomposer un état d'un niveau donné, en un sous-ensemble d'états décrivant plus finement le comportement d'un système. La hiérarchie est une notion que l'on retrouve dans STATECHARTS, mais ARGOS limite son usage en interdisant les transitions inter-niveau par l'introduction des mécanismes de terminaison qui permettent un meilleur contrôle des processus

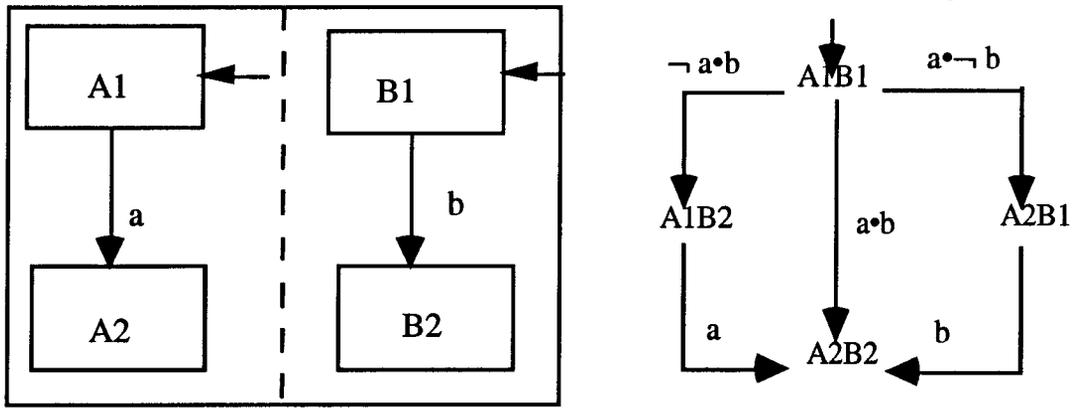


*Figure IX.6 : La hiérarchie*

Pour illustrer notre propos, prenons l'exemple d'une machine M1 que l'on peut arrêter ou régler en utilisant deux boutons de commande : a et b. Le bouton a permet de passer de l'arrêt au réglage et le bouton b permet de sélectionner le réglage souhaité. Le réglage peut être interrompu à tout moment en utilisant le bouton a. Les deux automates sont totalement équivalents. Mais notre description du fonctionnement de M1 est beaucoup plus simple à réaliser et à comprendre avec l'automate hiérarchisé.

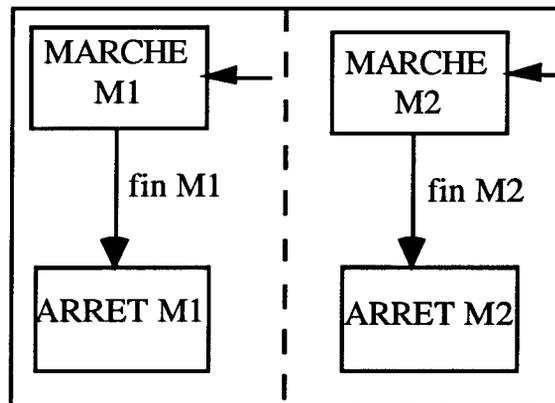
### IX.2.3. Le parallélisme

La décomposition parallèle d'un système réactif est un concept naturel puisqu'un système réactif évolue en parallèle avec, au moins, son environnement. L'opérateur de décomposition parallèle d'ARGOS permet cette méthode de conception comme l'indique la figure IX.7 où les deux automates représentés sont équivalents.



*Figure IX.7 : La décomposition parallèle*

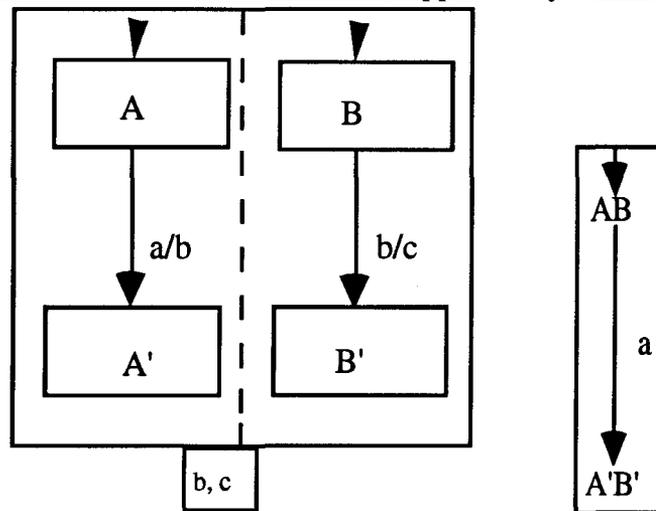
Pour illustrer la construction parallèle et son mécanisme, prenons l'exemple de la figure IX.8 qui décrit le fonctionnement parallèle et indépendant de deux machines M1 et M2. L'état initial de l'automate représenté est l'état de marche de M1 et M2. L'événement fin M1 (respectivement fin M2) déclenche l'arrêt de la machine M1 (respectivement M2) et n'a aucune conséquence sur l'état de la machine M2 (respectivement M1).



*Figure IX.8: Exemple de processus parallèles*

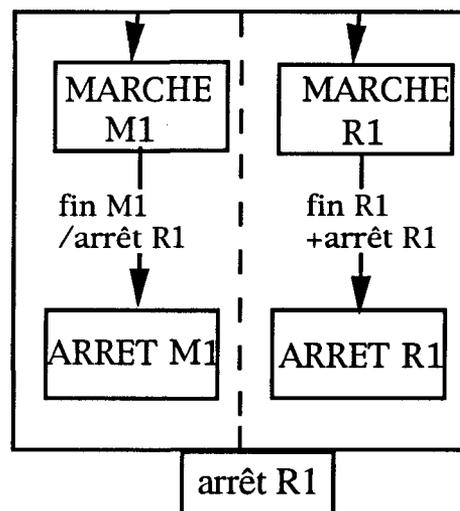
#### IX.2.4. L'événement local

C'est un opérateur original d'ARGOS qui s'inspire des mécanismes de déclaration des variables dans les langages de programmation classiques (Pascal, C, ...). L'opérateur d'événement local permet de préciser le domaine de diffusion des événements. C'est un opérateur unaire avec un paramètre qui spécifie l'ensemble des événements concernés. Dans le cas de la figure IX.9, la réception de l'événement "a" provoque la transition AA' et l'émission de b qui déclenche la transition BB'. L'émission de "c" est réalisée, mais dans l'exemple, cet événement est perdu. On notera en outre qu'un événement local ne peut pas venir de l'environnement du système. Dans la figure IX.9, la transition "AB"—( $\neg a \bullet b$ )—"AB'" est impossible.



*Figure IX.9 : L'opérateur d'événement local*

Prenons l'exemple d'une machine M1 et d'un robot R1 pour préciser l'intérêt de l'opérateur d'événement local. On considère que le robot R1 est dédié à la machine M1. L'arrêt de la machine M1 doit forcer l'arrêt du robot R1 (figure IX.10).



*Figure IX.10: Exemple d'application*

La machine M1 et le robot R1 sont initialement en marche. La fin du travail de M1 est signalée par l'événement "fin M1". Simultanément, le message "arrêt R1" est émis en provoquant l'arrêt du robot R1. Dans notre exemple, il reste possible d'arrêter le robot indépendamment de M1 si "fin R1" est émis par l'environnement. On notera que si "arrêt R1" n'est pas déclaré avec l'opérateur d'événement local, l'arrêt du robot R1 ne sera pas synchronisé avec l'arrêt de M1.

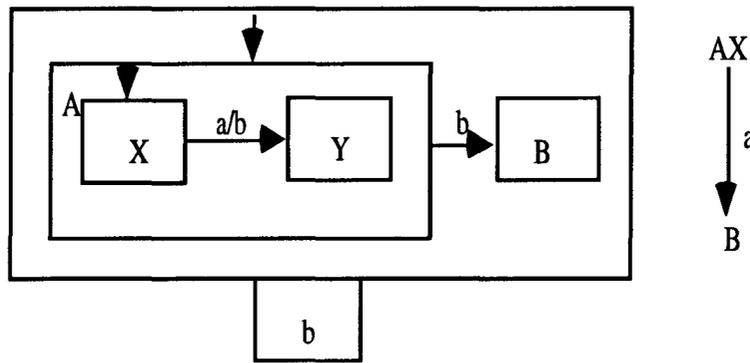
### IX.2.5. La terminaison

Trois types de terminaisons sont possibles dans un processus : normale, anormale (exception) et

ESTEREL exprime la terminaison anormale (exception) et l'interruption, respectivement avec le "TRAP" et le "WATCHING". La terminaison normale est implicite.

STATECHARTS exprime essentiellement le mécanisme d'interruption (abstraction faite des transitions inter niveau ).

Ces trois types de terminaisons correspondent à des fins de processus ou d'exécution d'une opération sur un système de fabrication ou de transport. ARGOS permet de les exprimer grâce au mécanisme d'auto-interruption de la figure IX.11. La réception de "a" provoque l'émission de b et le passage simultané de X à Y. En même temps, "b" déclenche la transition AB. On obtient alors une sortie propre de A vers B.



*Figure IX.11 : Le mécanisme d'auto-interruption*

Pour expliciter ce que nous avons appelé "Terminaison Normale", considérons le cas de M1 et M2, deux machines liées par une contrainte de coopération. La machine virtuelle obtenue est initialement dans une configuration de production. On doit pouvoir exprimer que la machine virtuelle MV obtenue par regroupement de M1 et M2 est à l'arrêt quand les deux machines M1 et M2 sont arrêtées (figure IX.12).

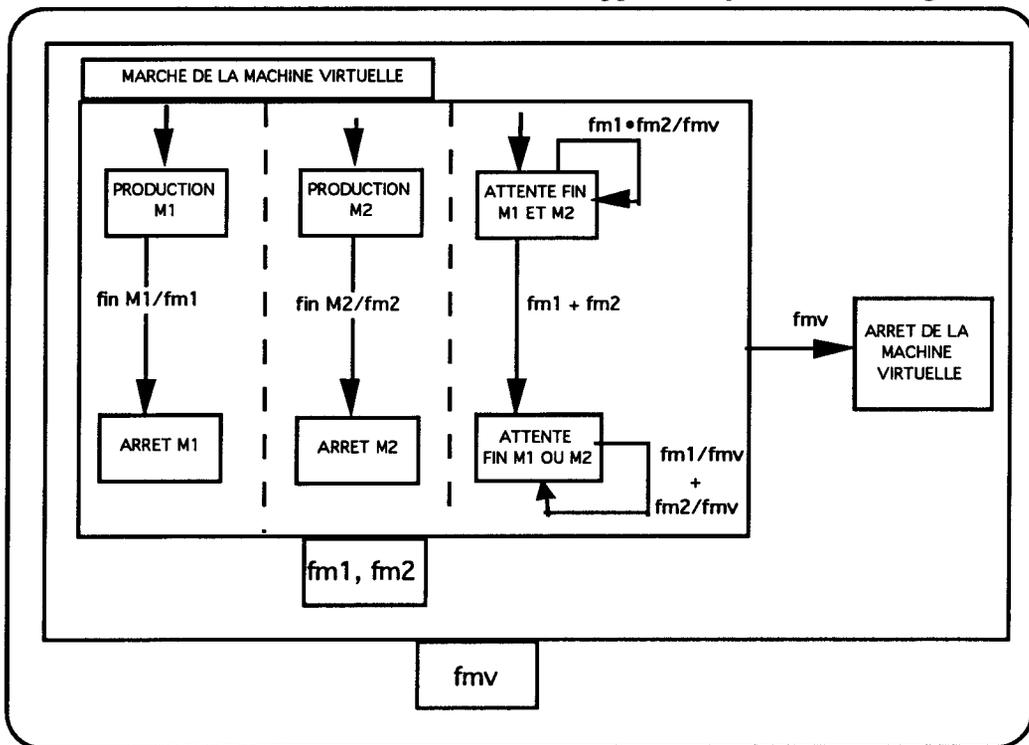


Figure IX.12 : La terminaison normale

Le mécanisme d'auto-interruption est aussi employé pour exprimer des terminaisons anormales ou des traitements d'exception.

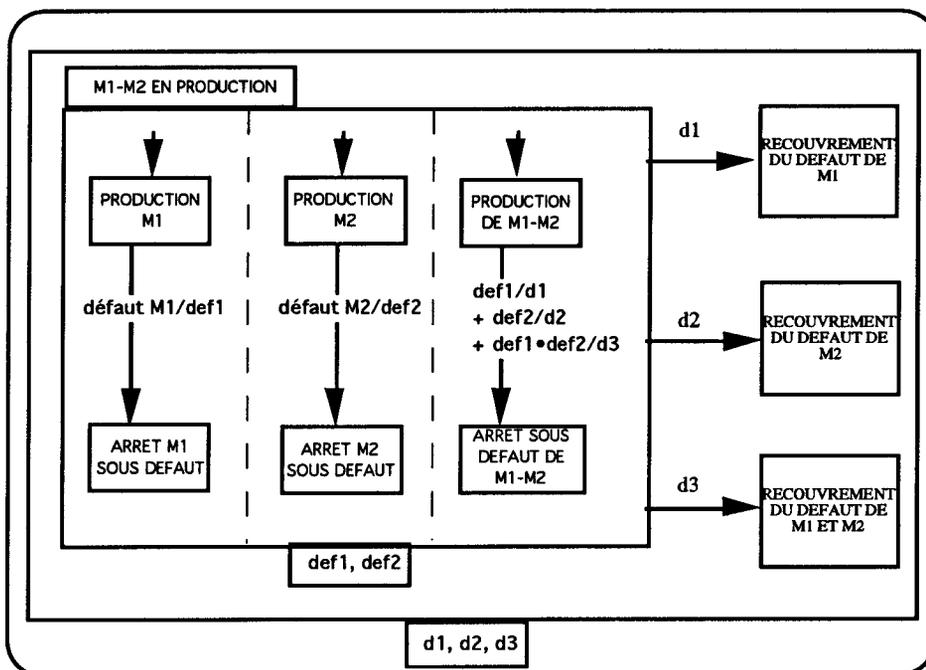
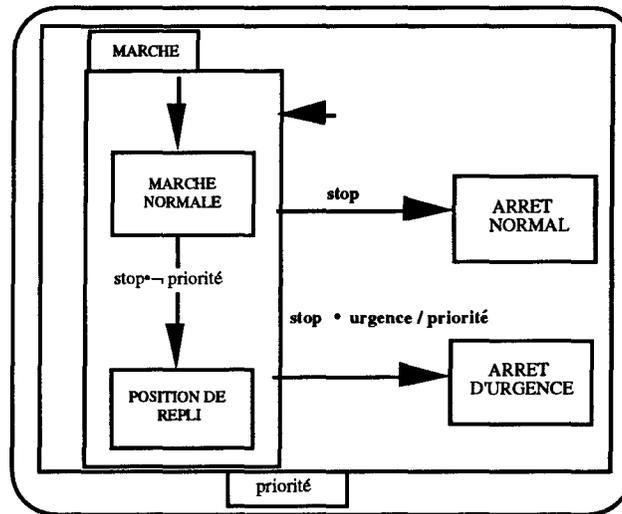


Figure IX.13: La terminaison anormale

En effet, dans l'exemple de la figure IX.13, les deux machines M1 et M2 fonctionnent en

parallèle. Leurs arrêts sous défaut doivent être traités par des procédures spécifiques de recouvrement. Ainsi, un défaut sur la machine M2 provoque l'arrêt sous défaut du groupe M1-M2 et le déclenchement d'une procédure spécifique de recouvrement. Ce mécanisme permet un contrôle en temps-réel des deux machines coopérantes. On peut, en outre, avoir une trace des événements qui ont provoqué des changements d'état du système.

La dernière application du mécanisme d'auto-interruption que nous présentons est l'interruption avec ou sans priorité.



*Figure IX.14 : L'interruption*

La figure IX.14 concerne un robot R. Ce robot, initialement en marche, peut être arrêté, soit par une requête "stop" interne à sa commande à la fin de son activité, soit par une requête "stop" externe. Quand cette requête d'arrêt externe est urgente, l'événement "urgence" est présent avec l'événement "stop", le signal local "priorité" est émis. Le robot est gelé dans sa position courante. Dans les autres cas, le robot rejoint sa position de repli avant de s'arrêter. L'événement local "priorité" permet ainsi une construction modulaire. En effet, la connaissance des mécanismes internes du macro-état "MARCHE" n'est pas nécessaire à son utilisation, il suffit de connaître ses événements d'entrée et de sortie.

### IX.2.6. L'environnement ARGONAUTE [Maraninchi, 89]

ARGONAUTE (actuellement sous X10) édite, visualise et simule les processus ARGOS [Maraninchi, 89]. Il sauvegarde les processus dans des fichiers texte qui emploient le format ARGOS. Le compilateur ARGOS peut produire un système avec des transitions étiquetées dans des formats différents qui permettent le raccordement d'Argonaute à plusieurs outils tel que :

- Aldebaran, développé par J.-C. Fernandez à l'IMAG.
- AUTO [Simone et al, 89] et Autographe, développé à Sophia-Antipolis.
- Xesar, développé à l'IMAG

## **CHAPITRE X : APPLICATION DE LA REPRÉSENTATION COD À L'EXPLOITATION ET À LA SUPERVISION DES SFPM**

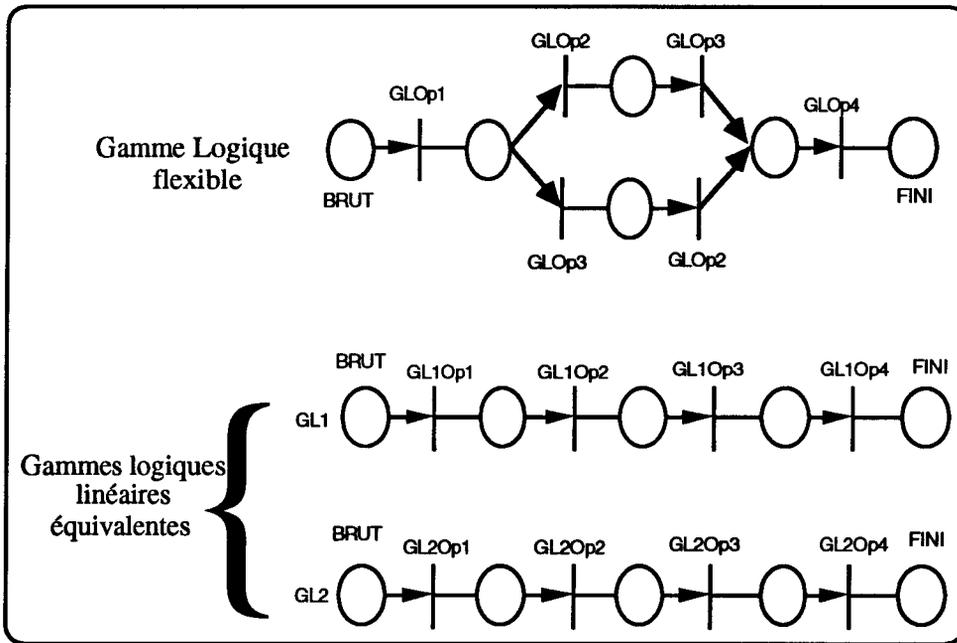
Ce chapitre aborde les SFPM du point de vue de l'exploitation et de la supervision. Nous y montrons le lien entre la représentation COD, que nous avons présenté dans la partie B, et les différents modèles de CASPAIM, tant pour la coordination que pour la surveillance. Nous expliquerons, notamment comment obtenir le graphe des fonctions nécessaires à la fabrication d'un produit à partir de sa gamme logique et de la représentation COD du SFPM. Nous montrerons ensuite comment exploiter les résultats de la surveillance pour représenter la disponibilité des fonctions du SFPM à tous les niveaux d'abstraction. La mise à jour de cette représentation se fera selon une démarche ascendante les informations issues de la surveillance des ressources physiques.. La disponibilité des fonctions du SFPM sera statique si elle est définie indépendamment des produits à fabriquer et dynamique, sinon. Elle correspond alors à l'état de disponibilité de la configuration courante du SFPM.

### **X.1. Décomposition arborescente d'une gamme logique**

Les objectifs de production sont définis par la donnée d'un flux de produits. Les états des produits sont caractérisés au sein des gammes logiques [Cruette, 91]. Chaque Gamme Logique est un modèle qualitatif qui représente les contraintes d'ordre sur les opérations du processus de fabrication d'un produit. Ce modèle est décrit par un graphe de type Réseau de Petri.

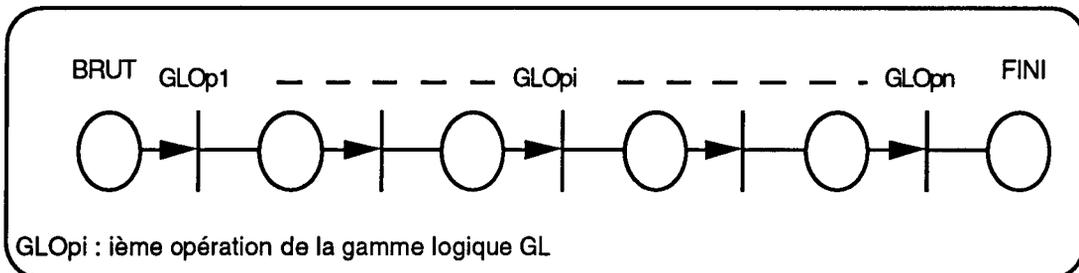
Nous allons montrer, dans ce paragraphe, le lien entre la représentation COD et les notions de Gamme Logique et de Gamme Opératoire qui sont des modèles proposés dans la démarche CASPAIM pour réaliser la commande de coordination.

Le premier réseau de Petri de la figure X.1 décrit une gamme logique flexible. En effet, les deux opérations GLOp2 et GLOp3 peuvent être réalisées dans un ordre quelconque. Cette gamme logique flexible est obtenue par la superposition de deux gammes logiques linéaires (l'ordre des opérations y est prédéfini). Nous traduisons cette superposition par une relation "OU". La gamme logique flexible correspond à la gamme logique linéaire GL1 ou GL2.



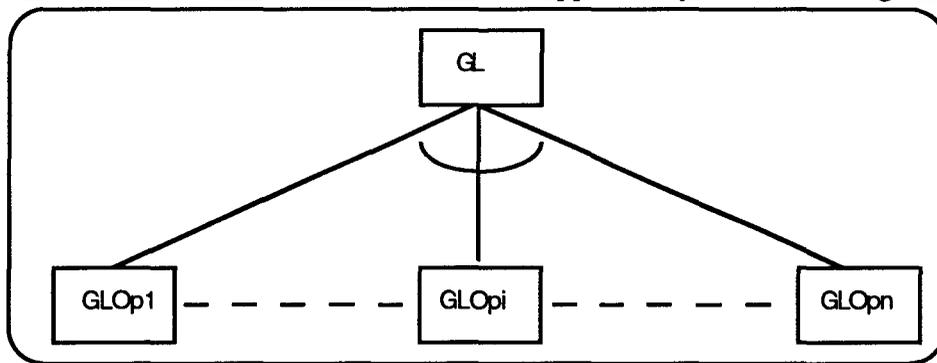
*Figure X.1 : Décomposition d'une gamme logique flexible en gammes logiques linéaires*

C'est donc le graphe d'une gamme logique linéaire (figure ci-après) que nous développerons en arborescence dans ce chapitre. L'arbre d'une gamme logique flexible sera obtenu par superposition des arbres de ses gammes logiques linéaires.



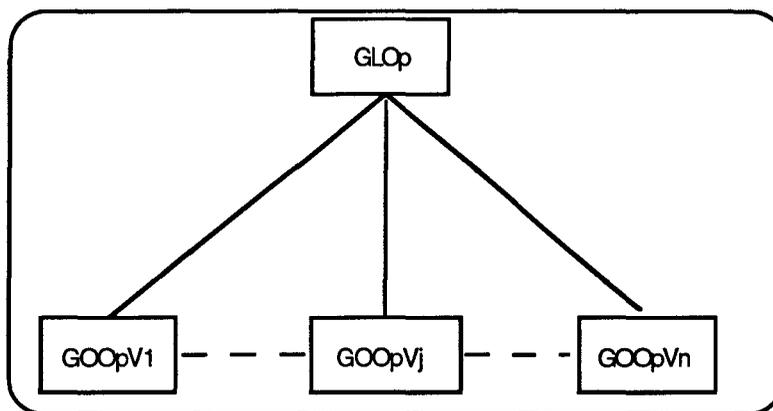
*Gamme logique linéaire*

La représentation arborescente que nous proposons pour une gamme logique est de type ET/OU. La liaison ET est une liaison de type Procédure, ses feuilles doivent être "lues" dans un ordre précis, celui des opérations (de gauche à droite dans la figure II.2).



*Figure X.2 : Décomposition d'une gamme logique*

Chaque GLOp (Opération de la Gamme Logique GL) correspond à plusieurs réalisations possibles (figure X.3). L'arbre COD va permettre d'associer à chacune de ces possibilités un COD dans lequel une seule opération est sélectionnée.



*Figure X.3 : Décomposition d'une opération de Gamme Logique*

En effet, chaque GOOpVi ( $i^{\text{ème}}$  Opération Virtuelle de la Gamme Opératoire GO) est la succession (figure II.4) :

- d'un chargement C : ensemble de transferts possibles,
- d'une opération effective : opération de gamme opératoire,
- d'un déchargement D : ensemble de transferts possibles.

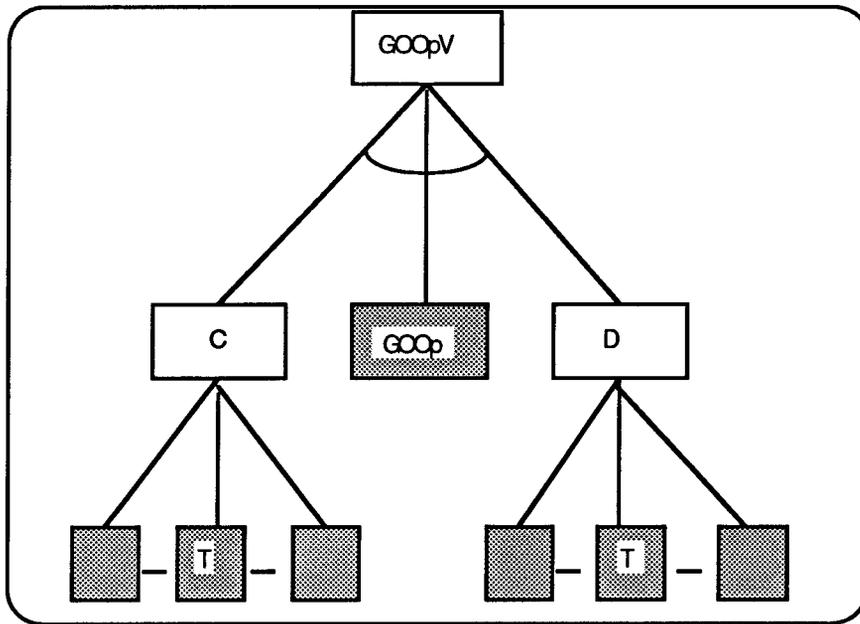


Figure X.4 : Décomposition d'une Opération Virtuelle de Gamme Opératoire

Ainsi, une opération virtuelle de gamme opératoire n'est finalement qu'un COD dans lequel l'opération de transformation a été précisée (figure X.5)

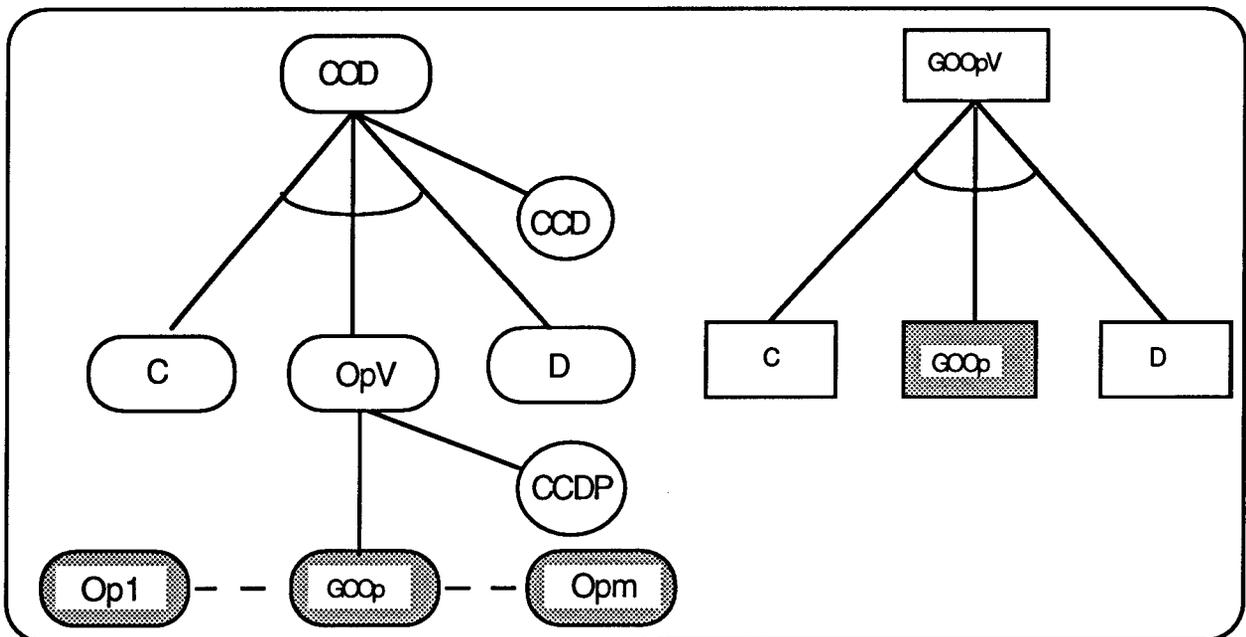


Figure X.5 : Equivalence entre COD et GOOpV

L'application de ces étapes à l'exemple de la figure X.1 aboutit à la représentation arborescente de la figure X.6. Les feuilles sont les opérations du SFPM que les objectifs de production contraignent. Le rôle de la commande et de la gestion des modes sera de gérer ces contraintes afin d'obtenir la meilleure performance.

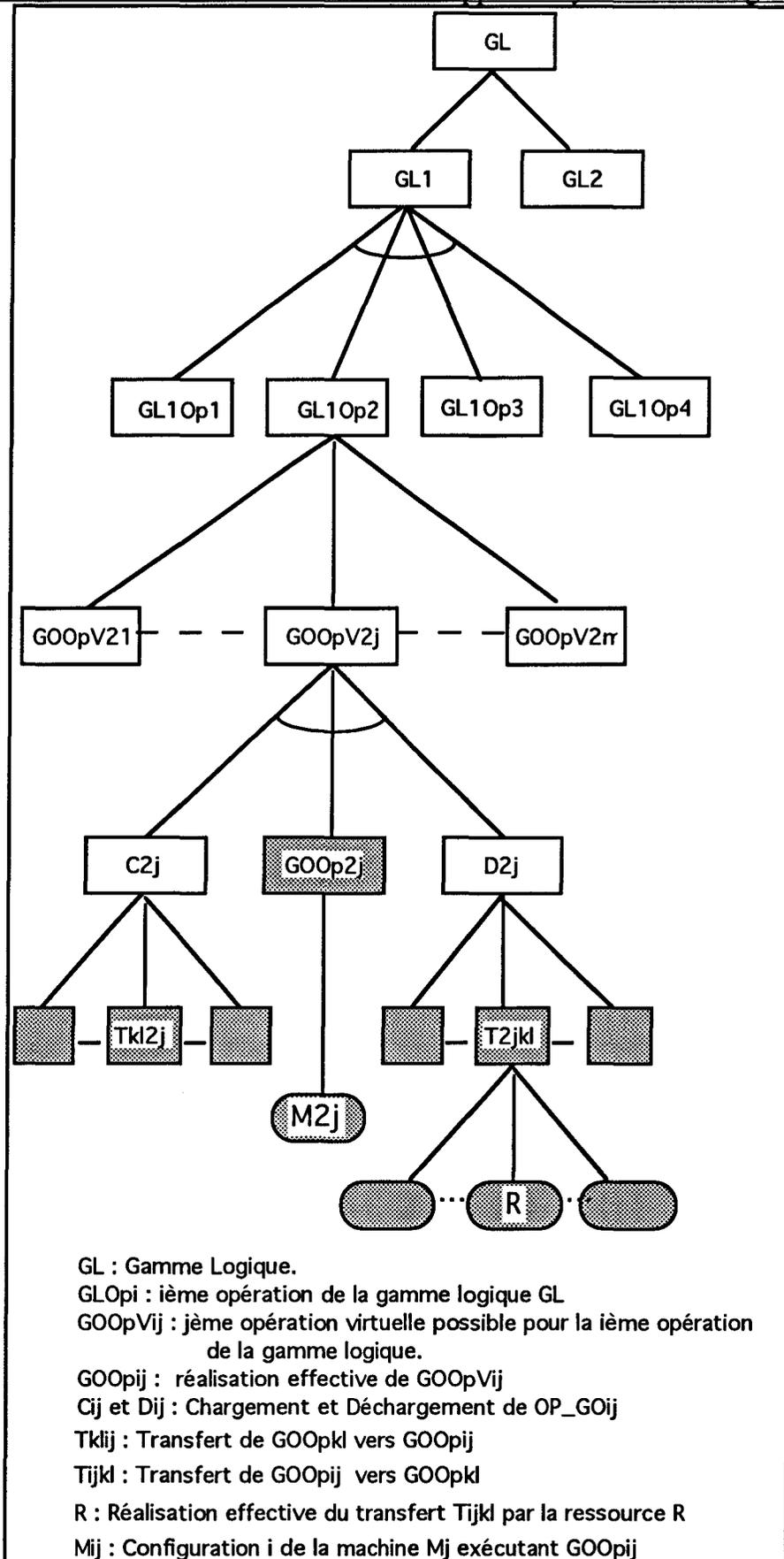


Figure X.6 : Représentation arborescente d'une Gamme Logique

**X.2. La disponibilité du SFPM et la représentation COD.**

Nous voulons déterminer l'état de disponibilité des différents sous-systèmes d'un SFPM à partir du niveau matériel d'un arbre COD. Soit  $\text{Disp}(S)$  la fonction donnant l'état de disponibilité de  $S$ . Cette fonction prend ses valeurs dans  $\{N, D, HS\}$  (Normal, Dégradé, Hors Service). La disponibilité d'un sous-système s'obtient récursivement à partir de la structure de l'arbre.

Nous avons besoin de 3 fonctions de base :

- $ID : \{N, D, HS\} \rightarrow \{N, D, HS\}$
- $ET_2 : \{N, D, HS\}^2 \rightarrow \{N, D, HS\}$
- $OU_2 : \{N, D, HS\}^2 \rightarrow \{N, D, HS\}$

Les "ET" et "OU" n-aires sont définis inductivement :

- $ET_k(E_1, \dots, E_k) = ET_2(E_1, ET_{k-1}(E_2, \dots, E_k))$
- $OU_k(E_1, \dots, E_k) = OU_2(E_1, OU_{k-1}(E_2, \dots, E_k))$

ID est l'identité,  $ET_2$  et  $OU_2$  sont définies par les tables suivantes :

Fonction  $ET_2$  :

$\begin{array}{c} \backslash \\ E1 \\ / \\ E2 \end{array}$	HS	D	N
HS	HS	HS	HS
D	HS	D	D
N	HS	D	N

Fonction  $OU_2$  :

$\begin{array}{c} \backslash \\ E1 \\ / \\ E2 \end{array}$	HS	D	N
HS	HS	D	D
D	D	D	D
N	D	D	N

La fonction "dispo" telle quelle est définie ci-dessus, permet d'évaluer la disponibilité statique du SFPM indépendamment de ses modes de production. En effet, nous avons considéré que la disponibilité d'un regroupement conjonctif (en ET) correspond à la disponibilité la plus défavorable

de ses composants. Un COD qui est la succession d'un chargement, d'une opération et d'un déchargement sera dans l'état normal si et seulement si chacun de ses trois composants est dans l'état normal. Il est dans l'état hors service si un seul de ses composants est hors service. Le COD est déclaré dégradé dans tous les autres cas. Pour un regroupement disjonctif (en OU) nous avons considéré qu'un regroupement est normal (respectivement hors service) si tous ses composants le sont. Il est dégradé dans tous les autres cas. Ce type de regroupement concerne par exemple un transfert qui peut être réalisée par plusieurs ressources.

La détermination de la disponibilité statique d'un SFPM s'appuie sur le modèle COD. Elle ne tient pas compte des priorités d'utilisation des ressources.

En cours de fonctionnement, on peut évaluer une disponibilité dynamique, qui diffère de la disponibilité statique notamment dans le cas des nœuds "OU". En effet, en fonction du mode et de la production choisis, la défaillance d'un composant peut ne pas dégrader la fonctionnalité d'un sous-système : la fonction "OU" que nous avons proposée n'est plus valable elle doit être adaptée par le concepteur au cas de son SFPM. C'est le cas de l'exemple des machines M1-M2 déjà présenté (figure VII.1) où transfert  $E \rightarrow M1$  est dans l'état de disponibilité N pour  $Disp(R1)=N$  quel que soit  $Disp(R2)$  en fonctionnement normal.

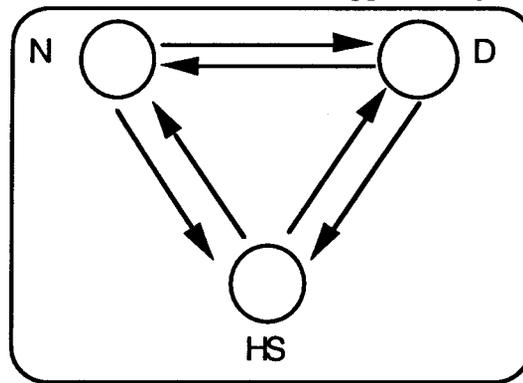
La détermination de la disponibilité dynamique d'un SFPM s'appuie sur la représentation arborescente des gammes logiques. La représentation de la disponibilité du SFPM revient, en mode dynamique, à supprimer certaines branches dans l'arbre d'une gamme logique (figure X.6).

Ce paragraphe va montrer comment construire et mettre à jour le modèle d'état de chaque élément à un niveau de la représentation à partir de ses composants du niveau immédiatement inférieur. Cette mise à jour va déterminer la possibilité d'utiliser une fonction suivant la gamme logique dans laquelle elle intervient et donc la configuration du SFPM. Ainsi, une fonction qui n'est plus utilisable, est immédiatement enlevée du graphe des fonctions disponibles. Le sous ensemble des gammes logiques disponibles ou dégradées est alors immédiatement restreint en conséquence.

Nous considérons une représentation de la disponibilité des éléments d'un SFPM par des automates à trois états : Normal (N), Dégradé (D) et Hors Service (HS). Ces automates sont définis à partir de la disponibilité des opérations et des transferts effectifs liée à celle de leurs ressources d'exécution.

### X.2.1. Disponibilité d'une opération effective

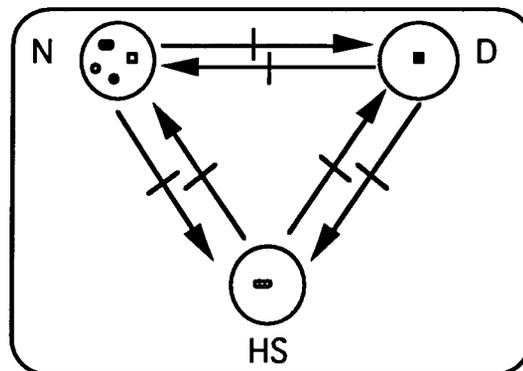
L'opération  $OpGO_{ij}$  de la figure X.7 est une opération effective. Elle est directement liée à une ressource de production. Son état dépend alors de celui de sa machine d'exécution. Nous proposons alors l'automate d'état de la figure X.8.



*Figure X.8 : Disponibilité d'une opération effective*

### X.2.2. Disponibilité d'un transfert

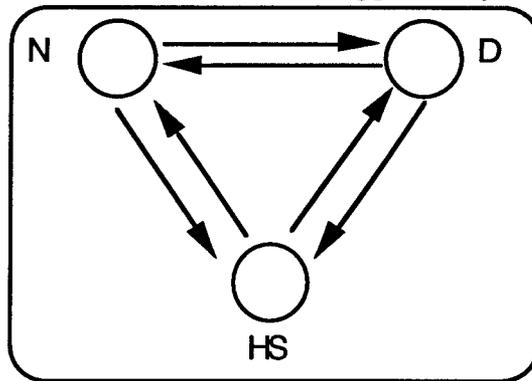
Le transfert Tijkl peut être réalisé par plusieurs ressources R. La fonction de disponibilité que nous avons proposée permet d'avoir une information agrégée de l'état du transfert. Dans le cas de la figure X.9, le transfert est considéré comme dégradé car il correspond à un regroupement disjonctif de réalisations possibles. L'automate d'état d'un tel transfert obtenu par la superposition des automates d'état des ressources qui peuvent le réaliser permet d'avoir une représentation plus fine. Nous proposons de l'illustrer ici par un Réseau de Petri coloré ; chaque marque indiquant par sa couleur la nature d'une réalisation possible.



*Figure X.9 : Disponibilité d'un Transfert*

### X.2.3. Disponibilité d'une opération virtuelle

OpVGOij est une opération virtuelle que l'on retrouve au niveau de la gamme opératoire. Son état dépend de celui de ces transferts de chargement et de déchargement ainsi que de l'opération effective qui la constitue. La représentation de son état s'obtient par un simple automate.

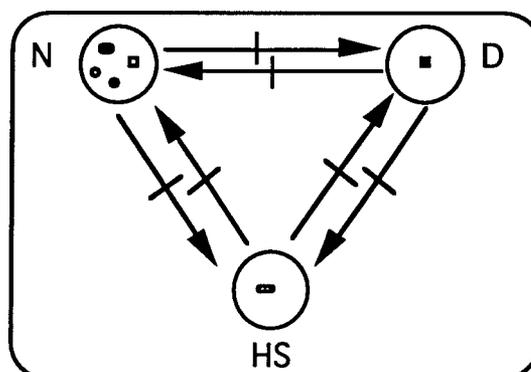


*Figure X.10 : Automate d'une opération virtuelle dans i-une gamme opératoire*

#### X.2.4. Disponibilité d'une opération dans une gamme logique

Une opération de gamme logique est une opération virtuelle qui peut être réalisée par un ensemble d'opérations virtuelles (opérations virtuelles de gamme opératoire).

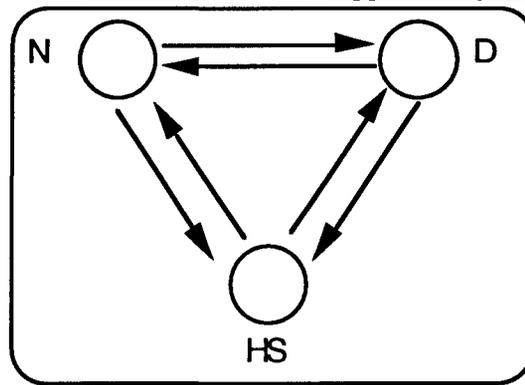
La fonction de disponibilité permet d'avoir une vision plus synthétique du système de production. L'automate d'état permet d'avoir une connaissance plus fine des capacités du SFPM. Il est obtenu par la superposition des automates d'état des opérations virtuelles qui peuvent réaliser l'opération de gamme logique. Comme pour le transfert, nous proposons de représenter cet automate par un Réseau de Petri coloré ; chaque marque représentant une opération virtuelle possible.



*Figure X.11 : Disponibilité d'une opération de Gamme Logique*

#### X.2.5. Disponibilité d'une gamme logique

Une gamme logique linéaire est une suite d'opérations de gamme logique liées par une contrainte de type ET. L'automate de la figure X.10 illustre l'état de la gamme logique : il résume l'état de toutes les opérations de la gamme logique.



*Figure X.12 : Disponibilité d'une gamme logique linéaire*

### Conclusion

Ce chapitre a permis de mettre en évidence le lien entre la représentation COD et l'exploitation des SFPM à travers la commande de coordination que l'on peut exprimer sous forme d'arborescences. La hiérarchie des gammes opératoires structure donc l'architecture du système de contrôle/commande de la figure V.6.b.

Nous avons aussi montré le rôle de la représentation COD dans la supervision. En effet, l'état des feuilles matérielles de l'arbre COD est mis à jour grâce aux fonctions de détection et de diagnostic de la surveillance des ressources du procédé. La structure de l'arbre permet alors de connaître l'état de tous les sous systèmes et de toutes les fonctions d'un SFPM. Cette connaissance doit être mise à jour dès l'occurrence d'un événement susceptible de modifier l'état général du système. Les langages synchrones répondent à cette exigence de réactivité et permettent en outre des vérifications formelles d'évolution cohérente de la spécification de la supervision du SFPM.

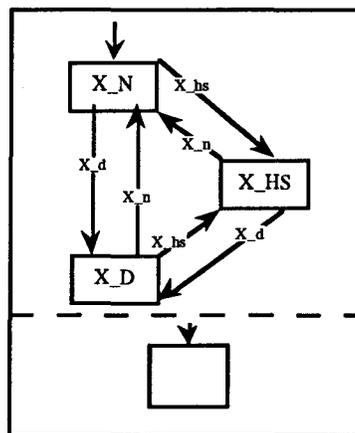
## **CHAPITRE XI : FORMALISATION DES FLUX D'INFORMATION DANS UN SFPM**

Nous proposons dans ce chapitre une représentation générique de la disponibilité d'une fonction d'un système flexible de production manufacturière. Cette représentation est d'abord utilisée pour exprimer la disponibilité statique d'un SFPM. Elle est ensuite utilisée pour exprimer la disponibilité dynamique. Nous allons mettre en évidence les mécanismes nécessaires à la mise à jour du modèle de disponibilité du SFPM à l'aide du formalisme ARGOS. Ce formalisme sera encore utilisé pour sélectionner et activer les fonctions et les ressources nécessaires à la réalisation d'une production en tenant compte de la disponibilité courante du système de production.

Ce chapitre utilise le formalisme ARGOS pour construire dans un premier temps le gestionnaire d'état du SFPM. ARGOS sera ensuite utilisé pour représenter le flux de commande du SFPM.

### **XI.1. Spécification de la disponibilité du SFPM : un modèle générique du flux ascendant**

Nous avons proposé de caractériser la disponibilité d'un objet X dans un SFPM (une ressource physique, un COD, une opération effective ou virtuelle, ...) par trois états exclusifs : un état normal (X\_N), un état dégradé (X\_D) et un état hors service (X\_HS).



*Figure XI.1 : Représentation de la disponibilité d'un élément de SFPM*

La spécification de la disponibilité d'un SFPM s'appuie sur sa représentation COD pour les aspects statiques et sur la représentation arborescente des gammes logiques pour les aspects dynamiques. Cette spécification se fera de manière récursive grâce à la construction de la figure III.1 qui illustre l'automate à trois états de X. Cet automate est contrôlé par un automate parallèle qui gère les communications avec l'environnement et les niveaux inférieurs.

L'objet X de la figure XI.2 est un élément du SFPM obtenu en regroupant des objets  $X_i$  d'un niveau inférieur liés par une contrainte de fonctionnement. L'événement "?X" provoque la mise à jour et la diffusion de l'état de X à travers les événements "X\_n", "X\_d" et "X\_hs". Dès que l'un

des composants  $X_i$  change d'état, il émet "?X" qui provoque un signal "e" forçant chaque  $X_i$  à déclarer son état courant. Des fonctions logiques "en()", "ed()" et "ehs()" permettent de calculer le nouvel état de X en testant la présence ou l'absence d'événements émis par les composants  $X_i$ . La définition des fonctions "en", "ed" et "ehs" est spécifique à chaque type de regroupement. Elles sont du ressort du concepteur du SFPM. Toutefois, nous proposons deux classes de fonctions qui permettent d'exprimer la fonction "dispo" introduite au chapitre précédent.

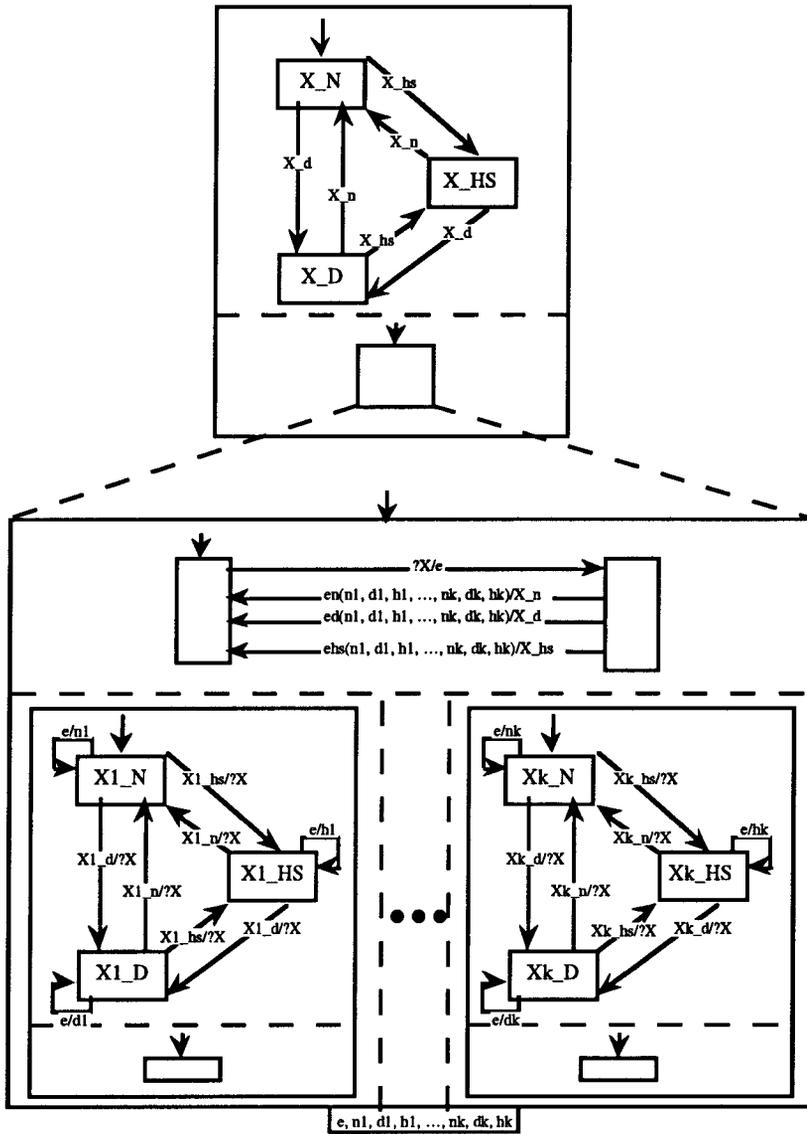


Figure XI.2 : Représentation de la disponibilité d'un élément de SFPM

Pour exprimer une contrainte conjonctive (contrainte de coopération, contrainte de chargement, ...), il est possible de prendre les fonctions suivantes :

- $en(n1, d1, h1, \dots, nk, dk, hk) = n1 \cdot \dots \cdot nk$
- $ehs(n1, d1, h1, \dots, nk, dk, hk) = h1 + \dots + hk$
- $ed(n1, d1, h1, \dots, nk, dk, hk) = (d1 + \dots + dk) \cdot \neg (h1 + \dots + hk)$

Pour une contrainte disjonctive (contrainte d'exclusion, contrainte partagée, ...), nous proposons :

- $en(n_1, d_1, h_1, \dots, n_k, d_k, h_k) = n_1 \bullet \dots \bullet n_k$
- $ehs(n_1, d_1, h_1, \dots, n_k, d_k, h_k) = h_1 \bullet \dots \bullet h_k$
- $ed(n_1, d_1, h_1, \dots, n_k, d_k, h_k) = (d_1 + \dots + d_k + h_1 \dots + h_k) \bullet \neg (h_1 \bullet \dots \bullet h_k)$

Ces deux classes de fonctions traduisent les tableaux de la fonction "dispo" présentée au chapitre X. Ces fonctions ne sont que des exemples de ce que peut choisir le concepteur d'un SFPM pour exprimer la relation qui lie des éléments du SFPM.

### **XI.1.1. Spécification de la disponibilité statique.**

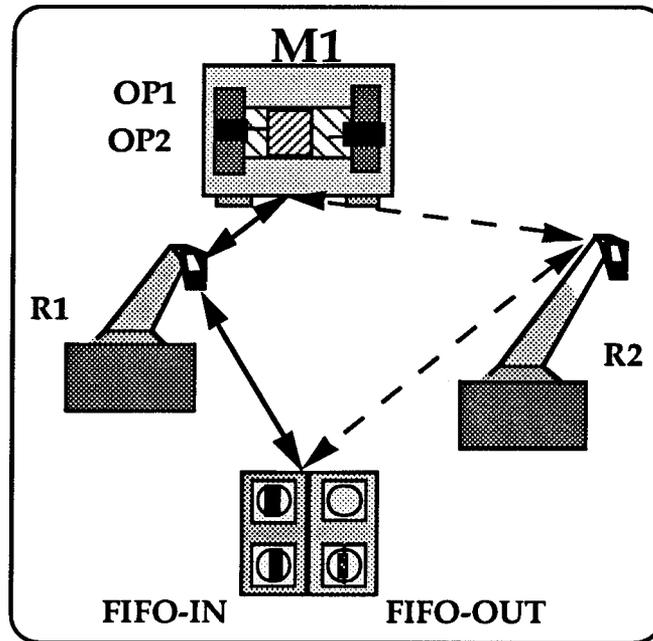
L'application du modèle de la figure XI.2 à la spécification de la disponibilité des éléments d'un arbre COD met en évidence deux difficultés : les ressources d'exécution (un robot, une machine, ...) et les transferts. En effet une ressource peut réaliser des opérations différentes ; son automate ne peut pas être inclus dans l'automate de contrôle d'une de ses opérations à l'exclusion des autres. De même, tout transfert est en même temps un chargement et un déchargement ; son automate devrait apparaître deux fois.

Le cas des ressources peut être résolu en considérant qu'une machine possède plusieurs configurations. Chaque configuration réalise une et une seule opération. L'état de chaque configuration sera décrit par un automate qui servira à la définition de l'automate de l'opération correspondante.

Le cas des transferts est plus délicat car nous devons choisir entre l'optimisation du programme de spécification qui n'écrit chaque automate qu'une seule fois et notre souhait de donner une structure hiérarchique et uniforme de la spécification de la disponibilité. Nous proposons de favoriser l'aspect hiérarchique car nous pensons que la double écriture de l'automate de chaque transfert n'est pas trop pénalisante.

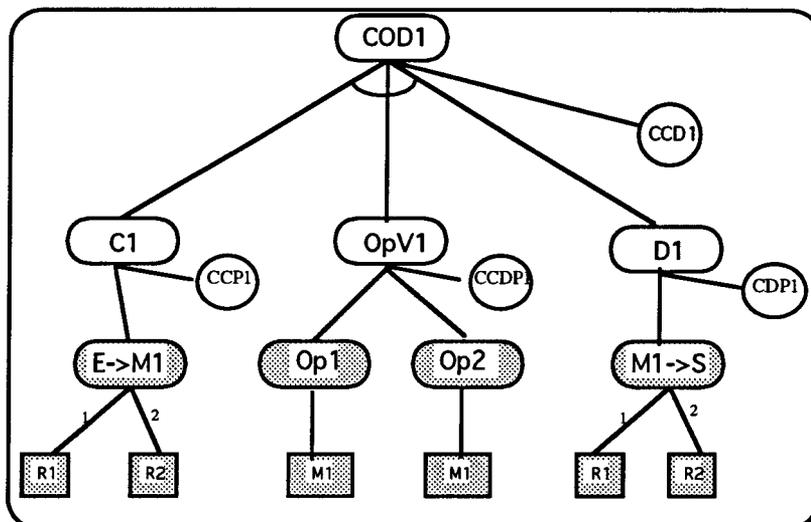
#### Exemple d'illustration

Une version simplifiée de l'exemple de la cellule flexible présenté dans la partie B va permettre d'illustrer l'utilisation du modèle générique de spécification du flux ascendant d'information. L'exemple que nous proposons comporte une machine M1 capable de réaliser deux opérations Op1 et Op2. La machine M1 est reliée aux tampons d'entrée/sortie grâce au robot R1. Un robot de secours R2 n'est activé que pour pallier une défaillance de R1 (figure XI.3).



*Figure XI.3 : Un exemple d'illustration*

La figure XI.4 montre l'arbre COD fourni par la démarche de modélisation que nous avons présentée dans la partie B. La figure XI.5 représente, au premier niveau, la disponibilité de la cellule COD1. Cette disponibilité est gérée grâce à un automate de contrôle qui combine les disponibilités de l'opération virtuelle OpV1 (groupement de Op1 qui devient Op11 et Op2 qui devient Op21), du chargement C1 et du déchargement D1. L'automate de contrôle de OpV1 est développé sur cette même figure XI.5. L'automate de contrôle de la disponibilité de C1 est développé dans la figure XI.6.



*Figure XI.4 : Représentation COD de l'exemple d'illustration*

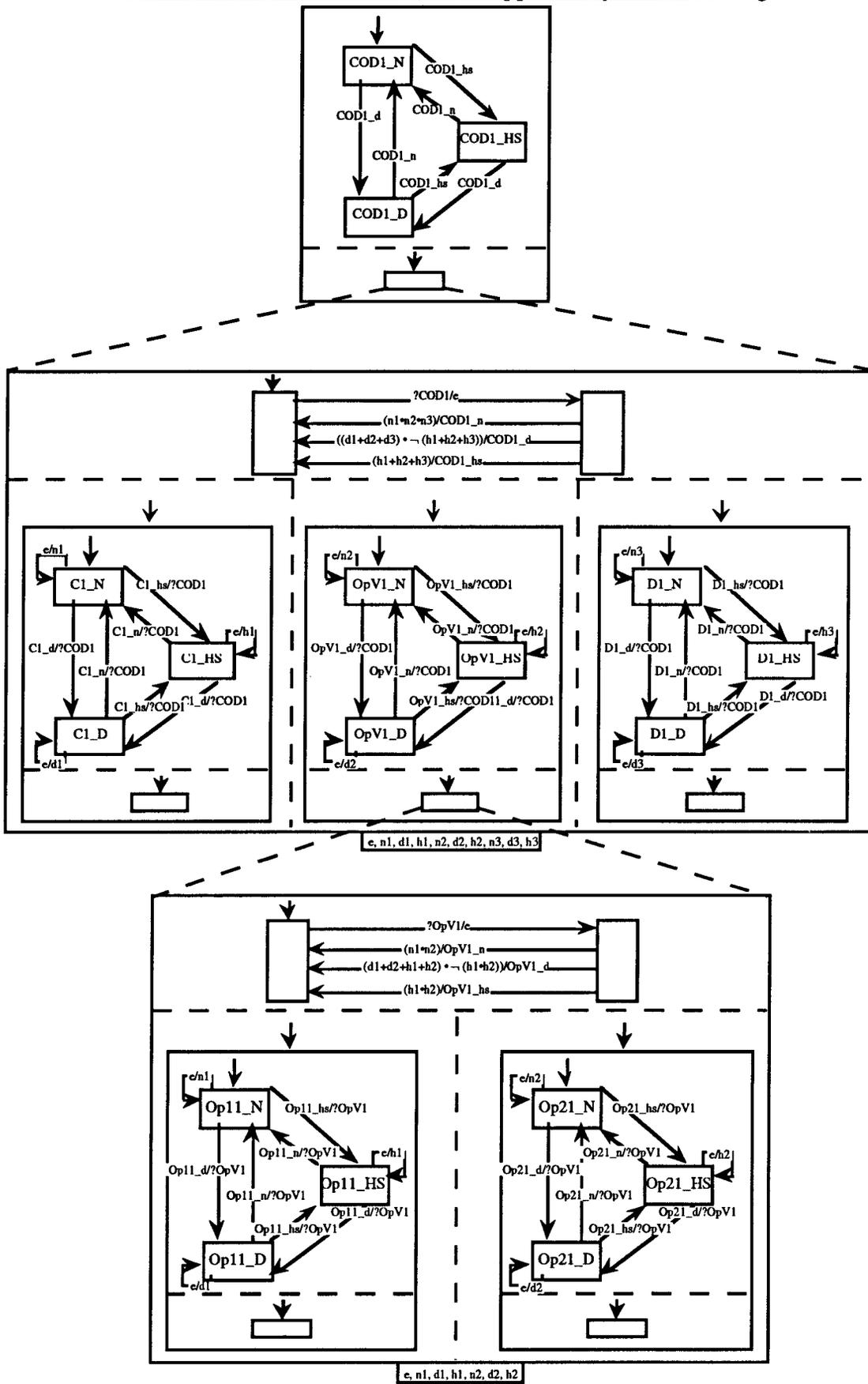


Figure XI.5 : Représentation hiérarchique de la disponibilité de l'exemple d'illustration

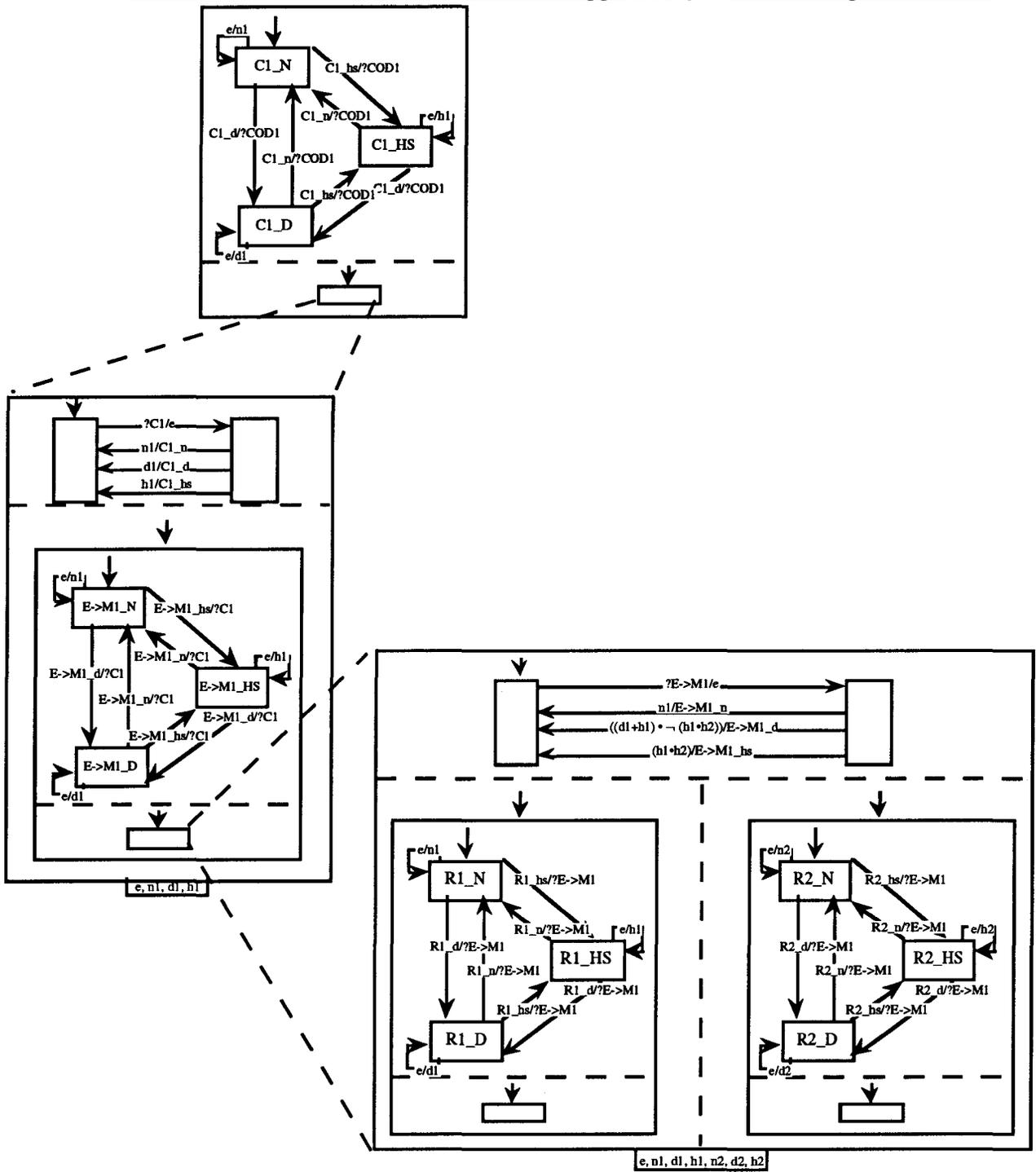


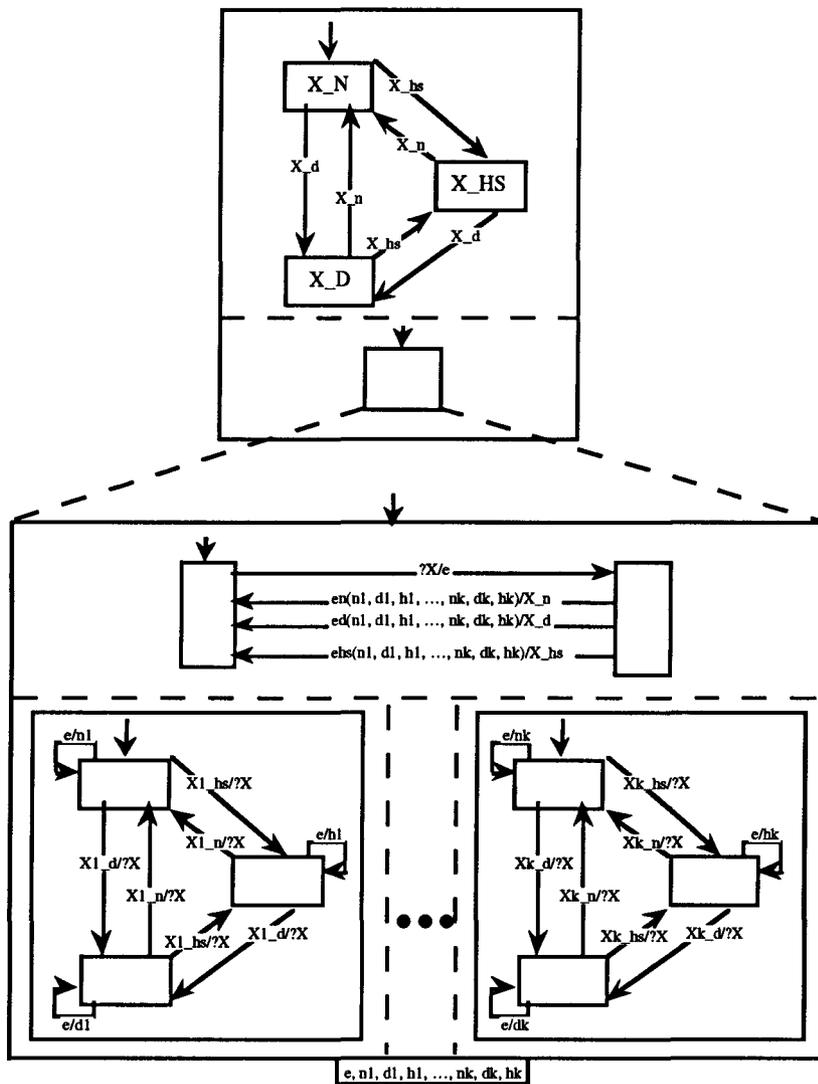
Figure XI.6 : Disponibilité du chargement C1

La figure XI.6 illustre l'influence de la disponibilité de R1 et R2 sur la disponibilité de C1. En effet, l'état du chargement C1 dépend en premier lieu de l'état du robot R1. L'état du robot R2 n'intervient que si R1 est dégradé ou hors service.

La représentation de la disponibilité du déchargement D1 peut être faite grâce à une structure identique.

### XI.1.2. Spécification de la disponibilité dynamique.

Le calcul de la disponibilité dynamique se fait grâce à la représentation arborescente des gammes logiques. Il s'agit de déterminer la disponibilité des gammes logiques des opérations de gamme logique, des opérations virtuelles de gammes opératoires et des opérations de gammes opératoires en utilisant la spécification de la disponibilité statique du SFPM. Il faut alors adapter la construction de la figure XI.2 en modifiant l'allure de l'automate de contrôle. En effet, la réécriture d'automates déjà décrits pour spécifier la disponibilité statique du SFPM n'est pas un bon choix car, il n'optimise pas le programme obtenu. Il est préférable d'utiliser des copies des automates déjà spécifiés pour les inclure dans les automates de contrôle de la représentation des états des gammes logiques. Ces copies évoluent parallèlement et en cohérence avec les automates d'origine en utilisant les mêmes signaux.



*Figure XI.7 : Représentation de la disponibilité dynamique d'un élément dans l'arborescence d'une gamme logique*

La figure XI.7 illustre cette adaptation : L'automate de contrôle de la disponibilité de l'objet X n'intègre pas directement les automates de contrôles des composants  $X_i$  qui ne lui sont pas propre et qui sont partagés avec d'autres éléments du SFPM (c'est le cas par exemple d'une opération de gamme logique qui intervient dans plusieurs gammes logiques différentes).

Exemple d'illustration :

Reprenons l'exemple d'illustration élémentaire de la figure XI.3 pour présenter la spécification de la disponibilité dynamique du SFPM. La cellule précédente doit fabriquer le produit P1 obtenu après la réalisation de l'opérations Op1 (figure XI.8).

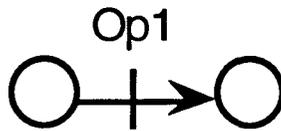
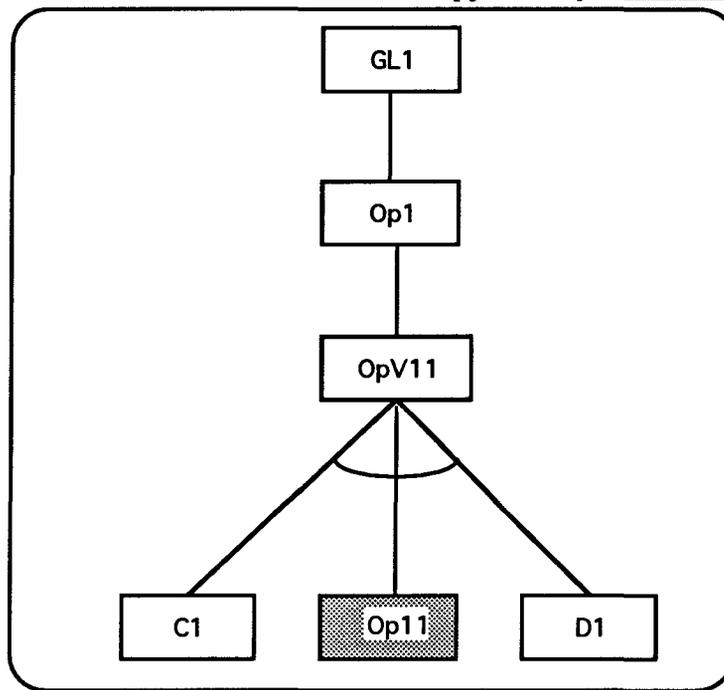


Figure XI.8 : Gamme logique GL1 du produit P1

La figure XI.9 illustre la représentation arborescente de la gamme logique GL1. Ce modèle va servir à la spécification de la disponibilité des fonctions et des ressources spécifiques à la réalisation du produit P1.

La figure XI.10 spécifie la disponibilité de l'opération virtuelle de gamme opératoire OpV11. Cette opération est le premier élément non défini dans la disponibilité dynamique du SFPM. Sa disponibilité est déterminée à partir des disponibilités de C1, Op11 et D1 définies dans le cadre de l'aspect statique du SFPM (figure XI.5 et XI.6).

La figure XI.11 intègre une instance de l'automate de disponibilité de OpV11 pour déterminer la disponibilité de l'opération de gamme logique Op1. L'automate de disponibilité de Op1 est, à son tour, instancié pour spécifier la disponibilité de la gamme logique GL1.



*Figure XI.9 : Représentation arborescente de la gamme logique GL1*

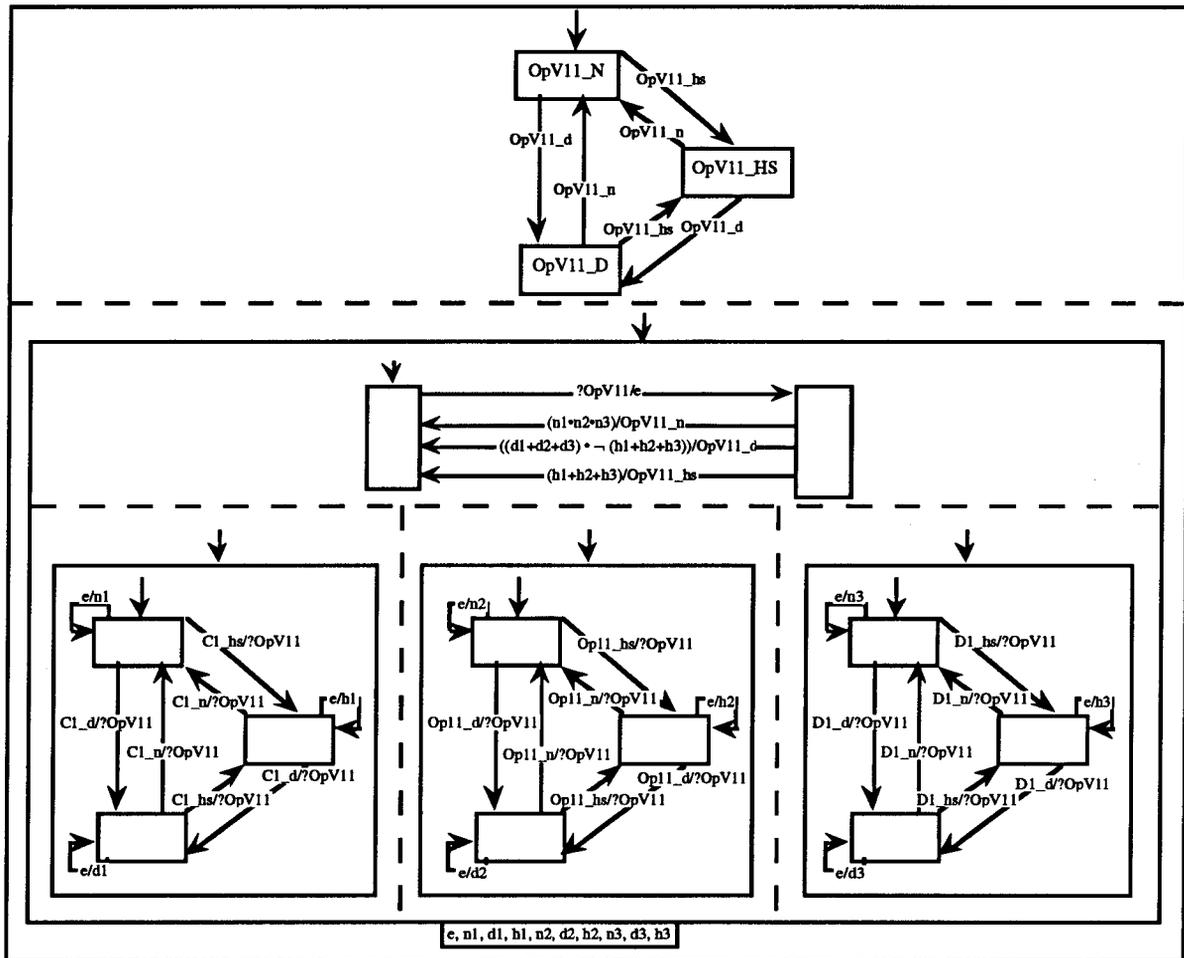


Figure XI.10 : Disponibilité de l'opération virtuelle de gamme opératoire  $OpV11$

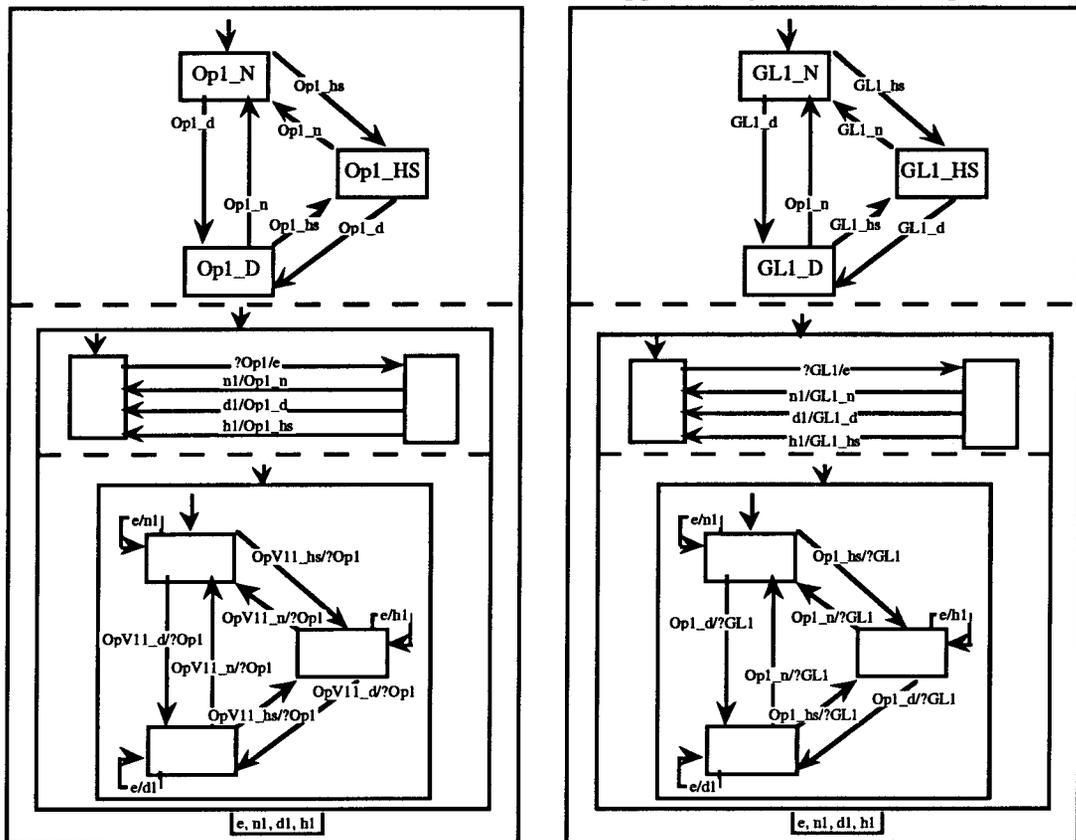


Figure XI.11 : Disponibilité de l'opération de gamme logique Op1 et de la gamme logique GL1

**XI.2. Spécification de la commande du SFPM : des modèles génériques du flux descendant**

Le flux descendant de l'information dans un SFPM est exprimé en s'appuyant sur la représentation arborescente des gammes logiques.

Tout élément X de niveau n dans une gamme logique correspond au regroupement d'éléments X1, ... et Xk de niveau n-1. Ce regroupement est soit conjonctif (type ET) soit disjonctif (type OU). Dans tous les cas l'élément X est initialement à l'arrêt dans un état X\_OFF. La mise en route de X par l'événement global on\_X fait évoluer l'état de X vers un état d'activité X\_ACTIF.

**Cas d'un regroupement conjonctif**

Dans le cas d'un regroupement conjonctif (figure XI.12), l'activation de X émet le signal ?X. Si le test de la disponibilité est positif, l'élément X est mis dans l'état de LANCEMENT. La phase de lancement s'achève quand tous ces composants sont effectivement en marche. L'élément X est alors en marche dans l'état MARCHE. Quand X est en marche, il est normalement en production. Cependant si une perturbation importante met hors service X, celui-ci est suspendu dans l'état X\_GEL ; tous ses composants sont, bien entendu, suspendus.

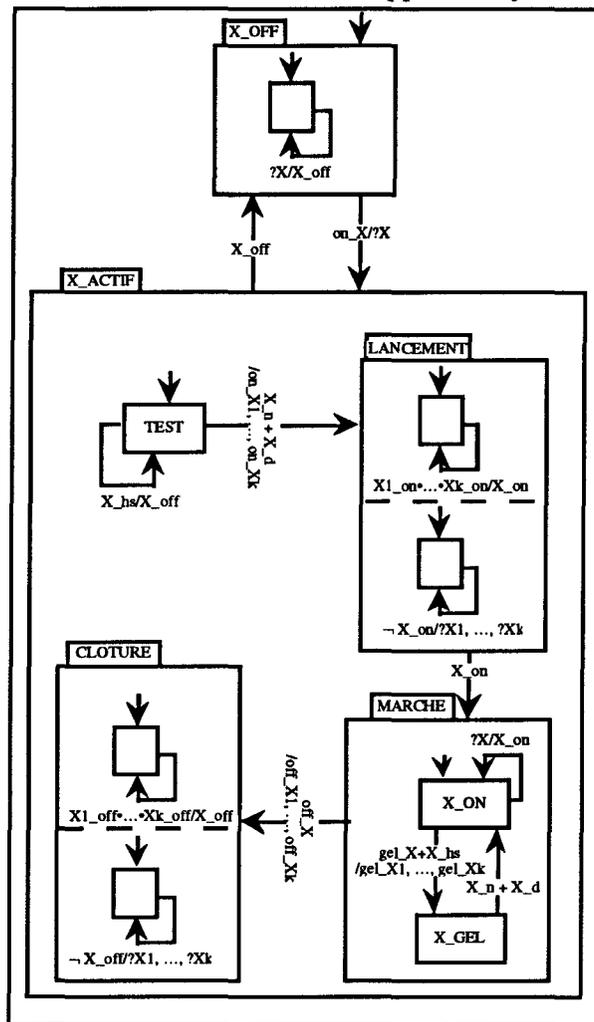


Figure XI.12 : Flux descendant dans le cas du "ET"

Le retour à la production est obtenu dès le retour à un état de disponibilité normale ou dégradée (si le concepteur du SFPM le spécifie ainsi).

L'arrêt de X est obtenu à la réception de  $off\_X$ . La procédure de clôture est alors déclenchée. Tous les composants de X sont, à leur tour, arrêtés. La sortie de l'état X\_ACTIF n'est effective qu'à l'arrêt de tous les composants  $X_i$ .

### Cas d'un regroupement disjonctif

Dans le cas d'un regroupement disjonctif (ou exclusif), la mise en route de X, lance une procédure de choix d'une configuration de production (figure XI.13). Ce choix peut aussi bien être le résultat d'un programme de gestion de la production et des ressources que l'ordre d'un opérateur humain. Quand le choix de la configuration de travail de X est réalisé, la procédure de mise en route se poursuit comme dans le cas précédent avec pour seul composant actif de X, l'élément  $X_i$  sélectionné par le choix de la configuration de production. Cependant, la clôture d'une

configuration émet l'événement local "off" qui réactive la fonction de choix d'une configuration de travail. Cette fonction peut sélectionner une nouvelle configuration ou déclencher l'arrêt de l'élément X.

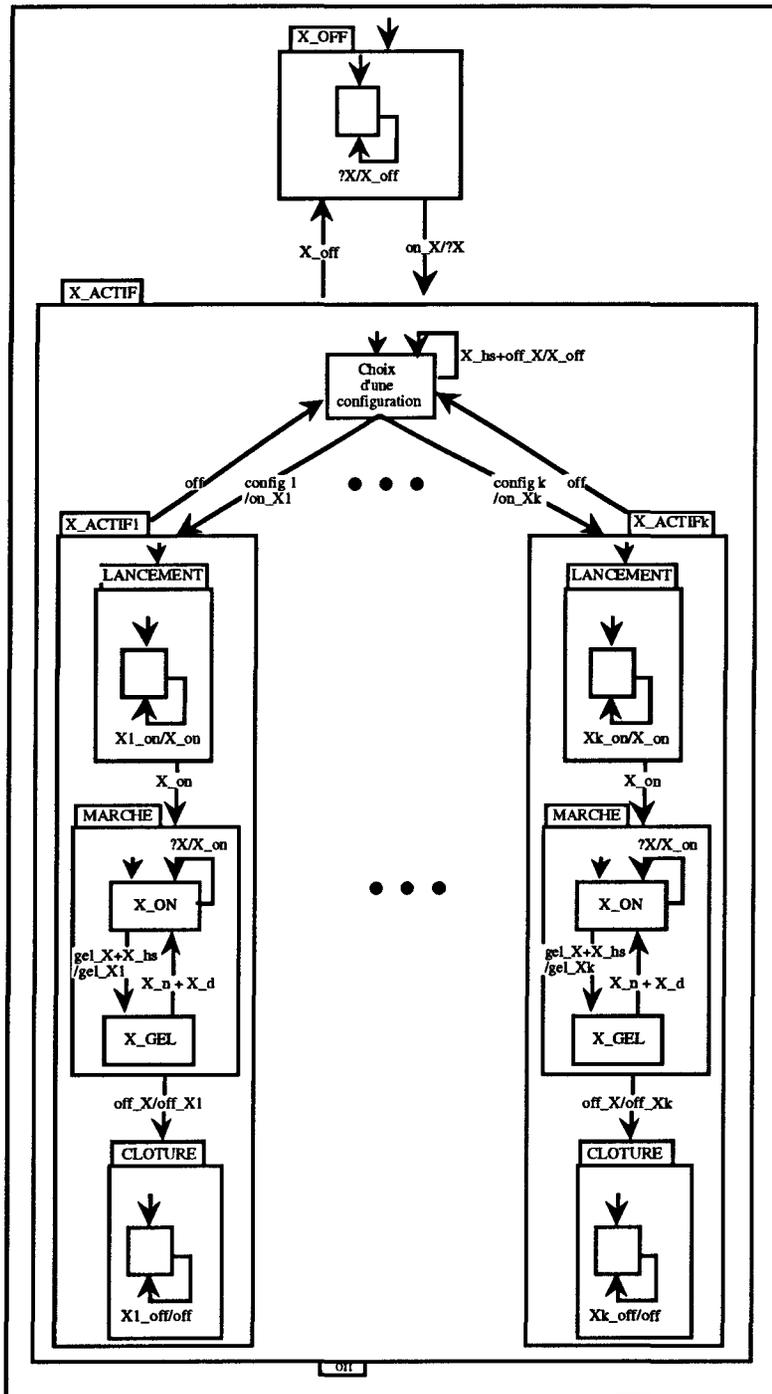


Figure XI.13 : Flux descendant dans le cas du "OU"

Exemple d'illustration :

L'activation de la gamme logique GL1 doit conduire à l'activation de l'opération de gamme

logique Op1 (figure XI.14). L'activation de Op1 conduit à son tour à l'activation de l'opération virtuelle de gamme opératoire OpV11.

La figure XI.15 illustre la transmission de l'ordre d'activation de OpV11 vers le chargement C1, l'opération de gamme opératoire Op11 et le déchargement D1. Chacune de ces fonctions transmet à ses composants l'ordre d'activation. Le chargement C1 va notamment activer le transfert E→M1. L'activation de E→M1 (figure XI.16) active la fonction de choix de la configuration de travail qui sélectionnera le robot R1 en priorité dans le cas de la cellule de la figure XI.3. Si une panne met hors service le robot R1, le gel de celui ci est déclenché. Son arrêt peut être commandé afin de pouvoir activer le robot R2 et continuer la production en mode dégradé.

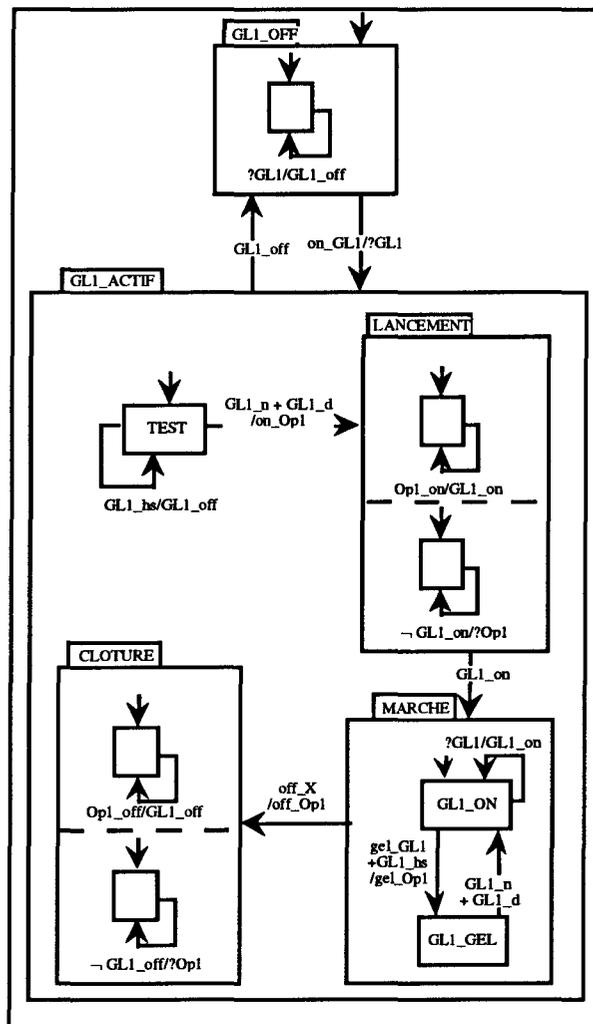


Figure XI.14 : Commande de la marche et de l'arrêt de la gamme logique GL1

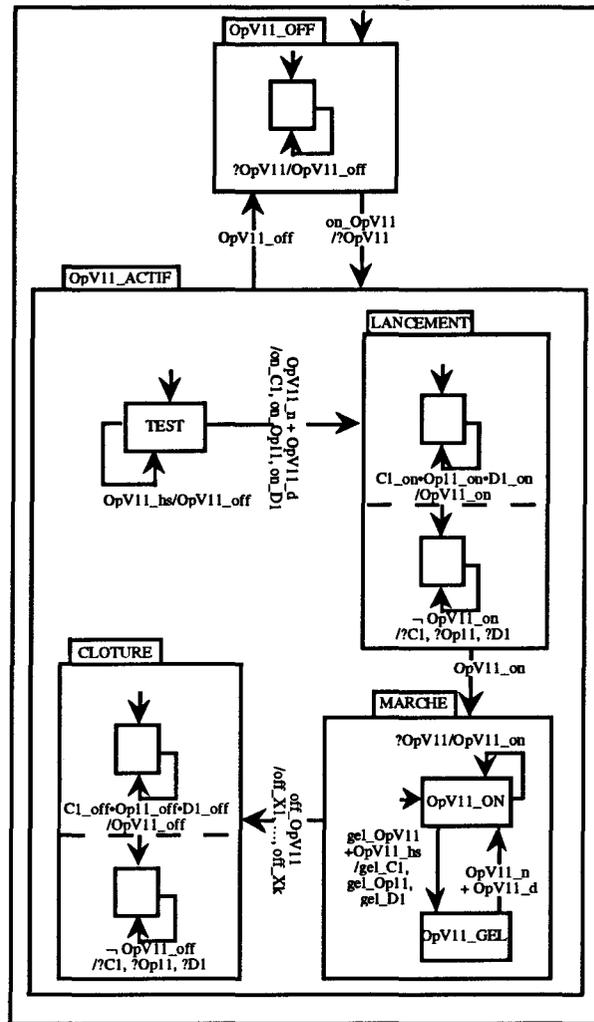


Figure XI.15 : Commande de la marche et de l'arrêt de l'opération virtuelle de gamme opératoire OpV11

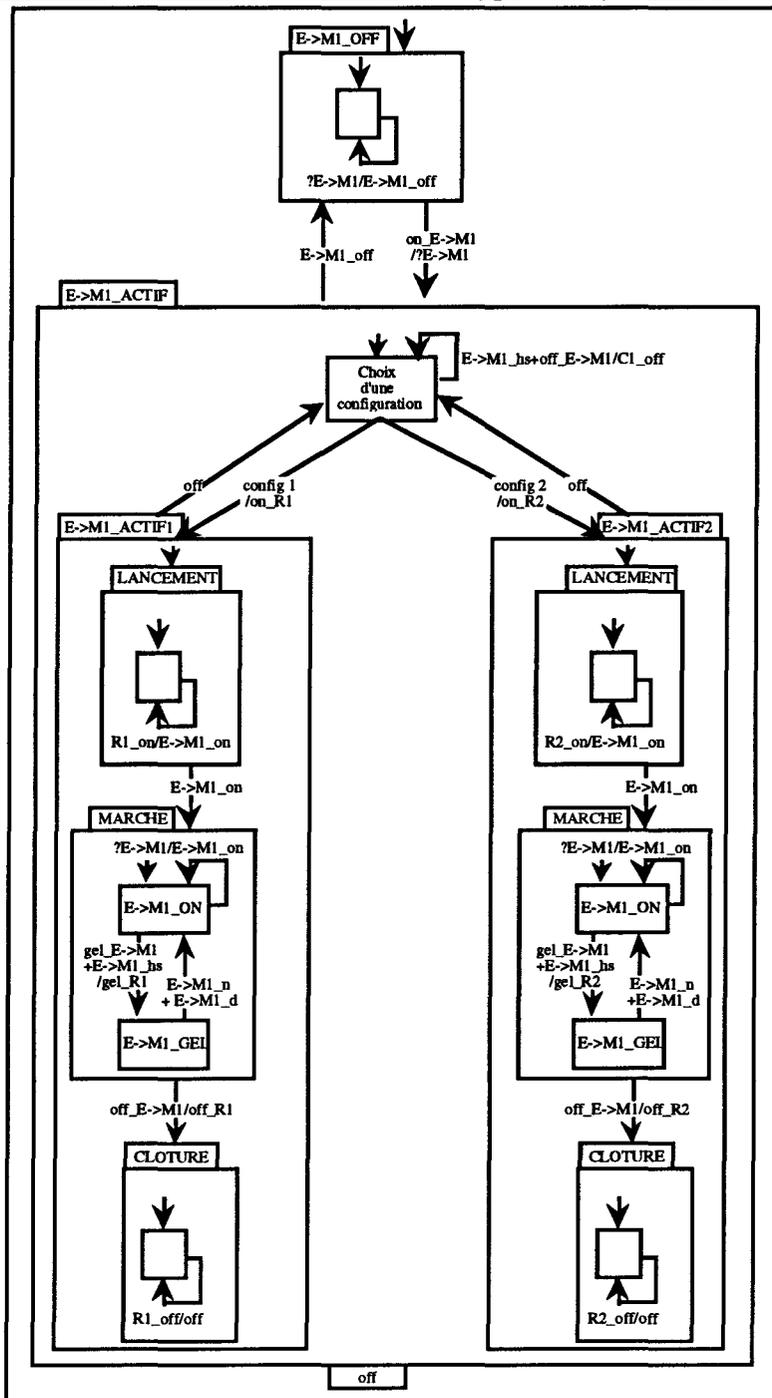
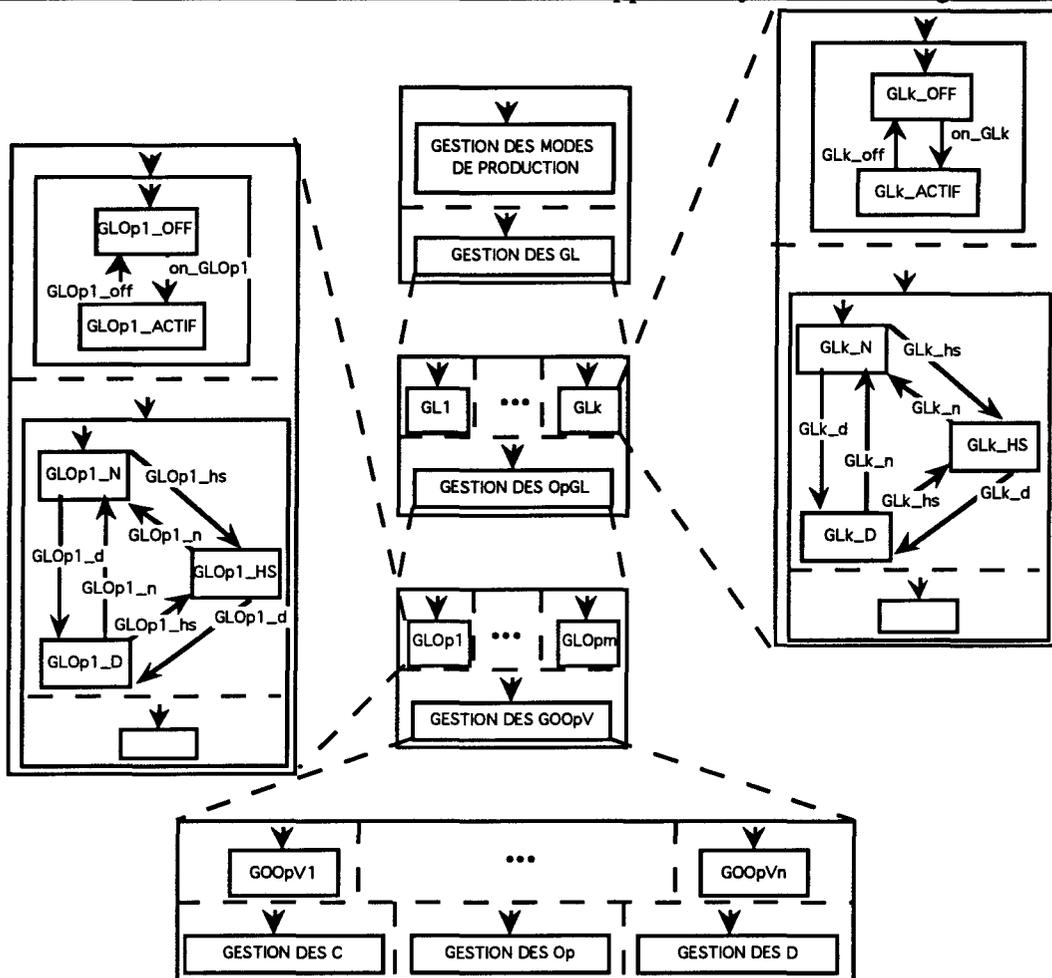


Figure XI.16 : Commande de la marche et de l'arrêt du transfert E->M1

### XI.3. Intégration des flux d'information ascendant et descendant

L'intégration des flux d'information ascendant et descendant est réalisée par l'association parallèle des automates de représentation de la disponibilité et de la spécification de la commande. La figure XI.17 indique que la gestion des modes de production permet de contrôler les gammes logiques en s'appuyant sur la disponibilité du SFPM.





*Figure XI.18 : Structure de la gestion des modes de production*

La structure de la figure XI.18 fait partie de l'architecture de contrôle/commande présentée au chapitre V (figure V.6.b). Le premier niveau commande directement les machines et les actionneurs du procédé. Le dernier niveau gère les gammes logiques et construit des modes de production.

La surveillance est assurée par les fonctions de filtrage, de détection et de diagnostic proposées par [Toguyeni, 92] et [Elkhatabi, 93] au plus bas niveau de la hiérarchie (procédé). A un niveau  $i$  plus élevé, la fonction de détection correspond à la réception d'un événement signalant un changement d'état d'une fonction ou d'une ressource du niveau  $i-1$ . La gestion de la disponibilité du SFPM détermine et diffuse le nouvel état du SFPM au niveau  $i$  et par conséquent, aux niveaux supérieurs. La fonction de filtrage est réalisée grâce à la structure des automates d'activation et de désactivation des fonction qui ne répercute un ordre vers une fonction que si celui-ci est compatible avec son état courant.

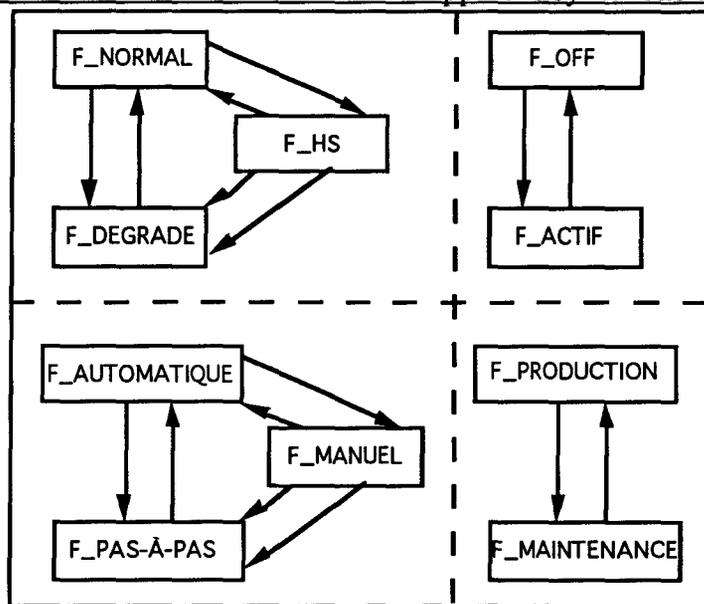
## CONCLUSION

Cette dernière partie de notre mémoire a permis de justifier la nécessité d'une approche synchrone de la gestion des modes d'un SFPM. Il est, en effet, nécessaire de disposer de la connaissance en temps réel de l'état et de la disponibilité des ressources fonctionnelles et matérielles du SFPM et de prendre en compte de manière réactive les perturbations issues de l'environnement ou les changements d'objectifs de production issus du niveau le plus élevé de la hiérarchie de commande.

Nous avons présenté dans le deuxième chapitre les langages synchrones en insistant sur les formalismes StateCharts et Argos. Ces deux formalismes permettent une spécification graphique de la gestion des modes. StateCharts a l'avantage d'être associé à un environnement de développement convivial et performant. Son inconvénient est lié à sa richesse d'expression. En effet, il autorise des spécifications et des écritures impossibles à valider formellement. Le formalisme Argos qui s'inspire largement de StateCharts restreint ce pouvoir afin de proposer des mécanismes de spécification que l'on peut valider formellement avec des outils déjà développés à cette fin.

Nous avons utilisé dans le chapitre XI le formalisme ARGOS pour proposer une spécification générique de la disponibilité des fonctions et des ressources d'un SFPM. Cette spécification s'appuie sur la représentation COD du SFPM pour la disponibilité statique et sur la représentation arborescente des gammes logiques pour la disponibilité dynamique. Nous avons ainsi pu exprimer le flux ascendant de l'information permettant de mettre à jour de manière réactive et synchrone la connaissance de la disponibilité des éléments du SFPM. Nous avons montré dans un deuxième temps comment traduire le flux descendant des informations liées à la commande et à la configuration du SFPM. Nous avons distingué deux cas : le cas des regroupements conjonctifs et le cas des regroupements disjonctifs. Enfin, nous avons indiqué comment intégrer les deux flux d'information au sein d'une structure hiérarchisée et générique afin d'assurer de façon cohérente la validité instantannée de l'état du système et de tous ses composants.

Cette modélisation pourra être étendue pour représenter toutes les familles de modes que nous avons proposé dans la Partie I. Une fonction (ou une ressource) F sera représentée par l'automates ci-après composé de quatre automates parallèles éventuellement contraints.



*Figure XI.19 : Prise en compte des familles de modes du GEMMA*

## **CONCLUSION GÉNÉRALE**

Nous avons montré dans ce mémoire qu'une approche des systèmes de production par les fonctions et les opérations permet d'aller beaucoup plus loin dans la spécification des contraintes de fonctionnement et de gestion des configurations et des modes de fonctionnement. Nous avons présenté la problématique de la supervision et de la gestion des modes en étendant les concepts du GEMMA. En effet il ne s'agit plus de gérer la marche et l'arrêt d'un système automatisé de production unitaire mais de gérer les configurations d'un Système Flexible de Production composé et complexe. En effet, la flexibilité permet de réagir à d'éventuelle défaillance des ressources de production par une reconfiguration du système. Cette adaptabilité n'a d'intérêt que si elle reste exploitable par la supervision et le pilotage, d'où la nécessité d'une approche globale et cohérente, au sens de l'intégration, de la conception des SFPM.

La contribution de nos travaux dans ce contexte ont porté sur deux points importants :

- la modélisation du système de production
- l'exploitation de la modélisation.

La démarche proposée a permis, dans un premier temps, de construire un modèle du système de production, par une analyse ascendante des contraintes et des liens de fonctionnement entre opérations. Cette technique de modélisation permet, de manière systématique, d'aboutir au modèle arborescent COD du système de production. La représentation fonctionnelle obtenue met en évidence les différents niveaux de commande du SFPM et fait ressortir les principes d'organisation et de conduite des installations dont les liens avec les fonctions sont spécifiés sur la représentation arborescente. L'originalité de cette démarche réside dans la possibilité d'intégrer tous les composants d'une unité de production. Elle permet aussi de tenir compte d'une manière très simple de l'ajout ou de la suppression d'un composant.

Dans une seconde démarche, nous avons montré comment l'exploitation de la modélisation permet de générer une spécification de la supervision du SFPM. Cette phase de traduction est basée sur la transcription directe de l'arbre COD et de l'arborescence représentant les gammes logiques dans un langage synchrone graphique : Argos. La traduction de ces arborescences sous forme de programmes synchrones permet la mise à jour réactive de l'état et de la disponibilité des composants du système de production et la transmission des ordres de reconfiguration de façon originale.

La mise en œuvre effective de notre démarche de spécification et de conception de la gestion des modes risque de se heurter aux problèmes de cycles de causalité bien connus dans les programmes synchrones. Une implantation réactive mais pas entièrement synchrone pourrait être envisagée afin de contourner cette difficultés.

## Conclusion générale

Des perspectives d'amélioration sont envisageables, d'une part, pour étendre la description de l'état d'une ressource ou d'une fonction par la prise en compte de sa disponibilité, de son activité, de son fonctionnement et de son exploitation. développer l'outil informatique mettant en œuvre notre démarche, et d'autre part pour affiner notre modèle par la prise en compte du produit non plus en tant que gamme logique mais en tant que composant dynamique du système de production.

Les travaux sur la supervision et la gestion des modes que nous avons présenté dans ce mémoire seront poursuivis au LAIL par deux thèses qui ont démarrées en novembre 1995. La première développera et résoudra les problèmes posés par la surveillance de bas niveau et l'acquisition et le traitement des signaux émis par les capteurs. La deuxième thèse s'intéresse aux problèmes d'interfaçage et d'intégration des fonctions de la supervision et de la commande.

De manière prospective, il nous apparaît que l'emploi d'outils informatiques tels que les langages objets synchrones [André et al, 96], enrichirait les capacités du gestionnaire des modes et des configurations, en facilitant le cadre de modélisation des différents composant d'un SFPM. En effet, une approche objet facilite encore davantage la démarche de prototypage en permettant une meilleure représentation des différents points de vue d'un SFPM. Elle semble mieux adaptée à la description naturelle d'un système complexe. Les différents composants du système de production sont alors décrits par des objets possédant leurs comportements spécifiques. Les contraintes de fonctionnement peuvent alors être intégrées aux caractéristiques de comportement des objets.

Par ailleurs, l'enrichissement du modèle de base par la prise en compte du produit et de son état constitue une autre perspective de recherche intéressante. Une solution possible, dans le cas des processus manufacturiers, serait de considérer l'état du produit et de l'associer à celui de la fonction et surtout de la ressource qui le traite, au même titre que ceux de ses composants. Les modes de fonctionnement et la configuration de commande de la ressource seraient alors influencés par l'état du produit en cours de traitement. En effet, les procédures de reprise après une panne en cours de fabrication, ne sont pas toujours identique et dépendent fortement de l'état du produit en cours de fabrication. L'intégration d'une étude AMDEC sur les actions correctives en cas de défaillance pourra enrichir notre approche.

Ces points constituent des axes intéressants et prospectifs de développement, dans le cadre de l'exploitation d'un système de production.

# **ANNEXE : PRÉSENTATION DE CASPAIM**

A.1. Phases de spécification

A.2. Phase d'analyse et de conception

A.3. Phase de validation

A.4. Phase d'implantation

## **PRÉSENTATION DE CASPAIM [BOUREY, 93], [AMAR, 94] ET [CRAYE, 94]**

D'après [Anakök, 87], une approche par des “objectifs produits” et une approche par des “objectifs systèmes” sont nécessaires pour appréhender efficacement un système automatisé de production dans sa globalité. L’approche par des “objectifs produits” permet de définir la structure fonctionnelle du système de production, en partant des caractéristiques du produit et de sa gamme de fabrication [Amar, 94]. L’approche par des “objectifs systèmes” permet de définir des modes d'exploitation et de fonctionnement normaux ou dégradés du système de production en décrivant les procédures de passage d'un mode à un autre.

C'est ainsi que cette approche système a été exploitée dans CASPAIM pour définir une organisation et une gestion des modes de marche [Bois, 91], [Kermad et al, 93a]. Elle est fondée sur une représentation structurelle du système de production. Nos travaux, prenant en compte la dimension fonctionnelle, intègrent, dès la conception, les différents modes de fonctionnement ainsi que leur gestion en précisant les liens avec le système de coordination en terme de réorganisation des graphes de commande et de coordination des des ressources.

La figure A résume les principales étapes de conception dans le projet CASPAIM que nous allons présenter, afin de préciser le contexte de nos travaux.

### **A.1. Phases de spécification**

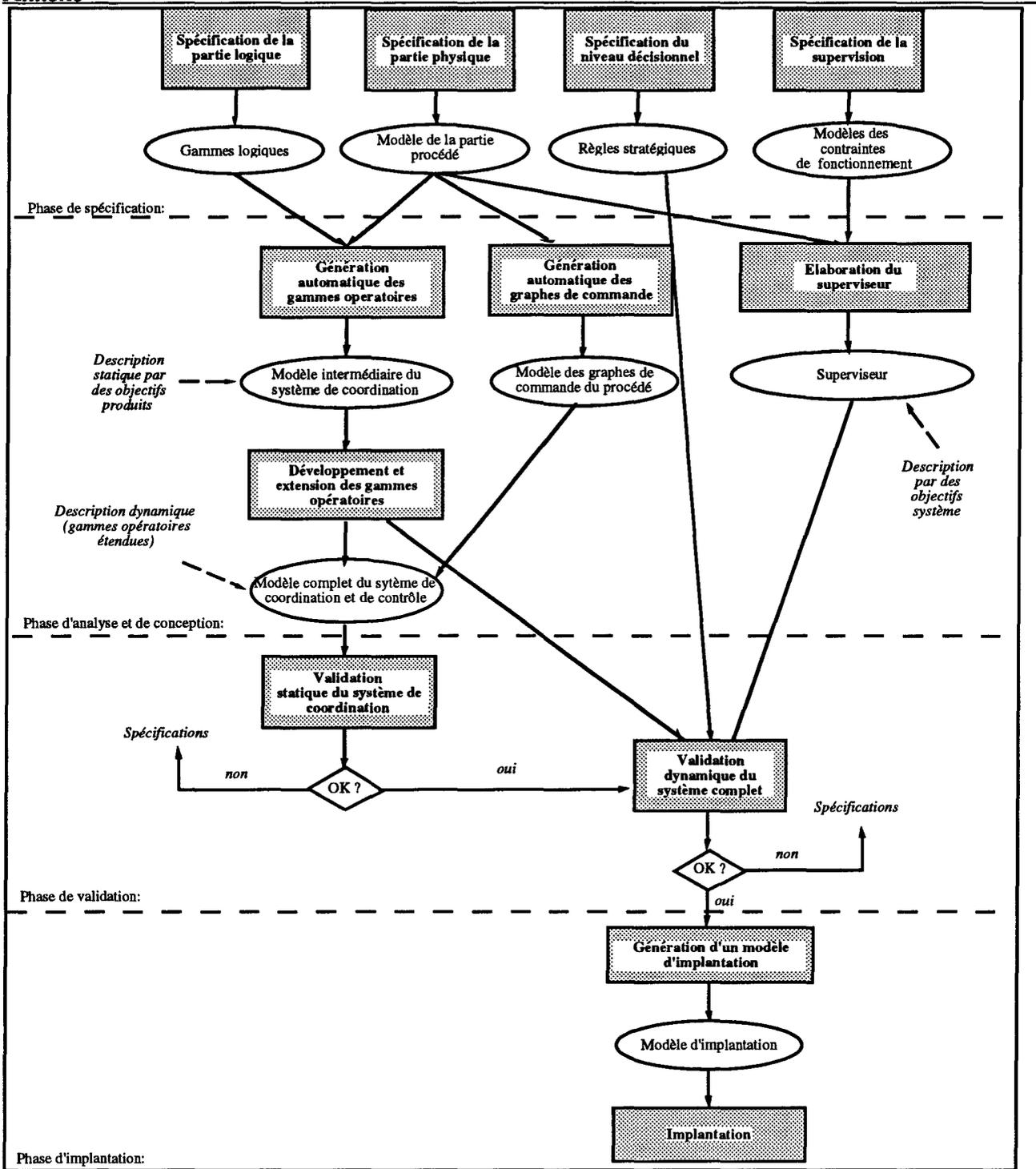
#### **A.1.1. Spécification de la partie logique : les objectifs de production**

Cette étape concerne la description formelle des objectifs de production du point de vue des produits. Cette description est indépendante des moyens de production et de l'architecture du système. Elle aboutit à la définition fonctionnelle des processus de fabrication des produits. Chaque processus est représenté par une Gamme Logique avec un graphe de type Réseau de Petri (RdP) [Cruette, 91] (figure A.1). Chaque place de ce graphe représente un état d'avancement du produit et chaque transition une opération transformant cet état.

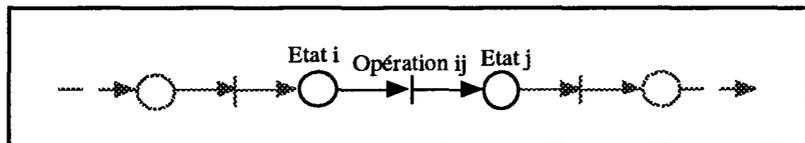
Les gammes logiques permettent de prendre en compte le séquençement élémentaire des opérations caractéristiques (transformation élémentaire qui engendre un changement d'état du produit) appliquées sur chaque type de produit en mettant en évidence les possibilités de flexibilité dans l'ordre de ces opérations.

Cette spécification met en évidence différents types de gammes non exclusives :

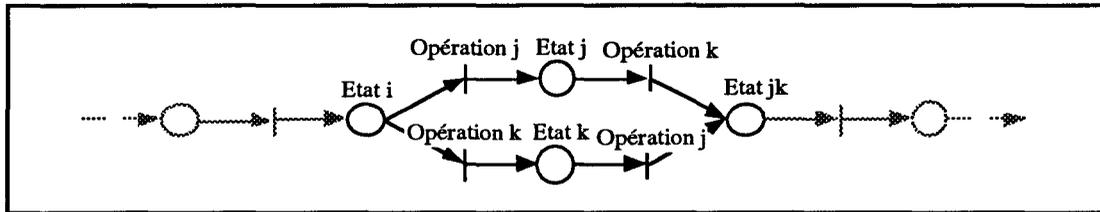
- les processus constitués de traitements unaires ne mettant en jeu qu'un seul produit à la fois. Ces processus peuvent tolérer ou non une flexibilité dans l'ordre pour toute ou partie des opérations (figures A1 et A2) ;



*Figure A : démarche de conception dans CASPAIM*

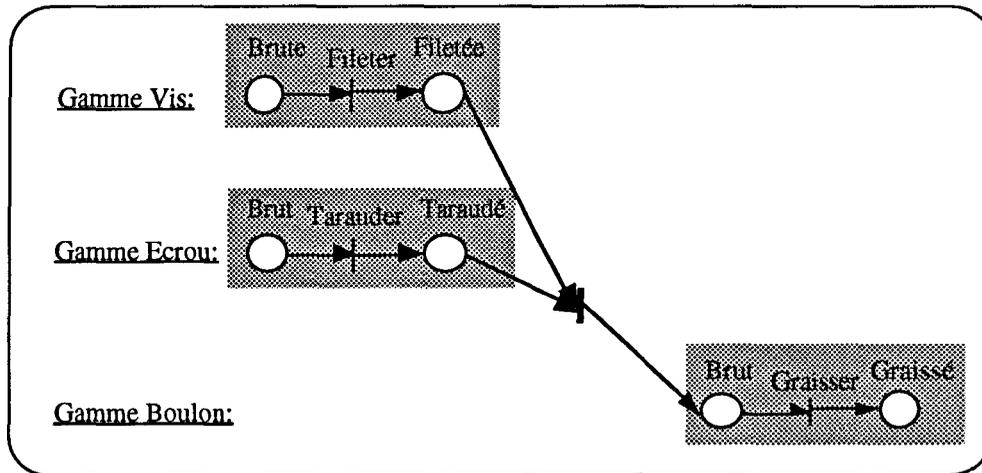


*Figure A1 : description d'une gamme logique élémentaire*



*Figure A2 : description d'un processus avec alternative*

- les processus concernant des opérations n-aires faisant intervenir un minimum de deux produits ((dés)assemblage) (figure A3).



*Figure A3 : description d'un processus n-aire*

### A.1.2. Spécification de la partie physique : les moyens de production

La spécification de la partie physique ou spécification opérationnelle [Amar, 94] va décrire les différentes ressources de production (fabrication, transport, manipulation,...) du procédé indépendamment de la partie logique qui est orientée produit. Cette phase comporte trois étapes : l'élaboration de la bibliothèque fonctionnelle, l'analyse structuro-fonctionnelle et la caractérisation de l'architecture du système de production.

#### Elaboration de la bibliothèque fonctionnelle

La bibliothèque fonctionnelle générale caractérise les fonctions opératives de base de tout élément du procédé. Ces fonctions sont classées en trois catégories (figure A4) :

- les fonctions de type stockage définissent les lieux physiques élémentaires pouvant recevoir un produit (pince de robot, section de convoyeur,...) ;

- les fonctions de type mouvement définissent les organes dotés d'un certain degré de mobilité tels les dispositifs mécaniques animés d'un mouvement spatial (bras d'un robot, coulisse longitudinale,...) ;

## Annexe

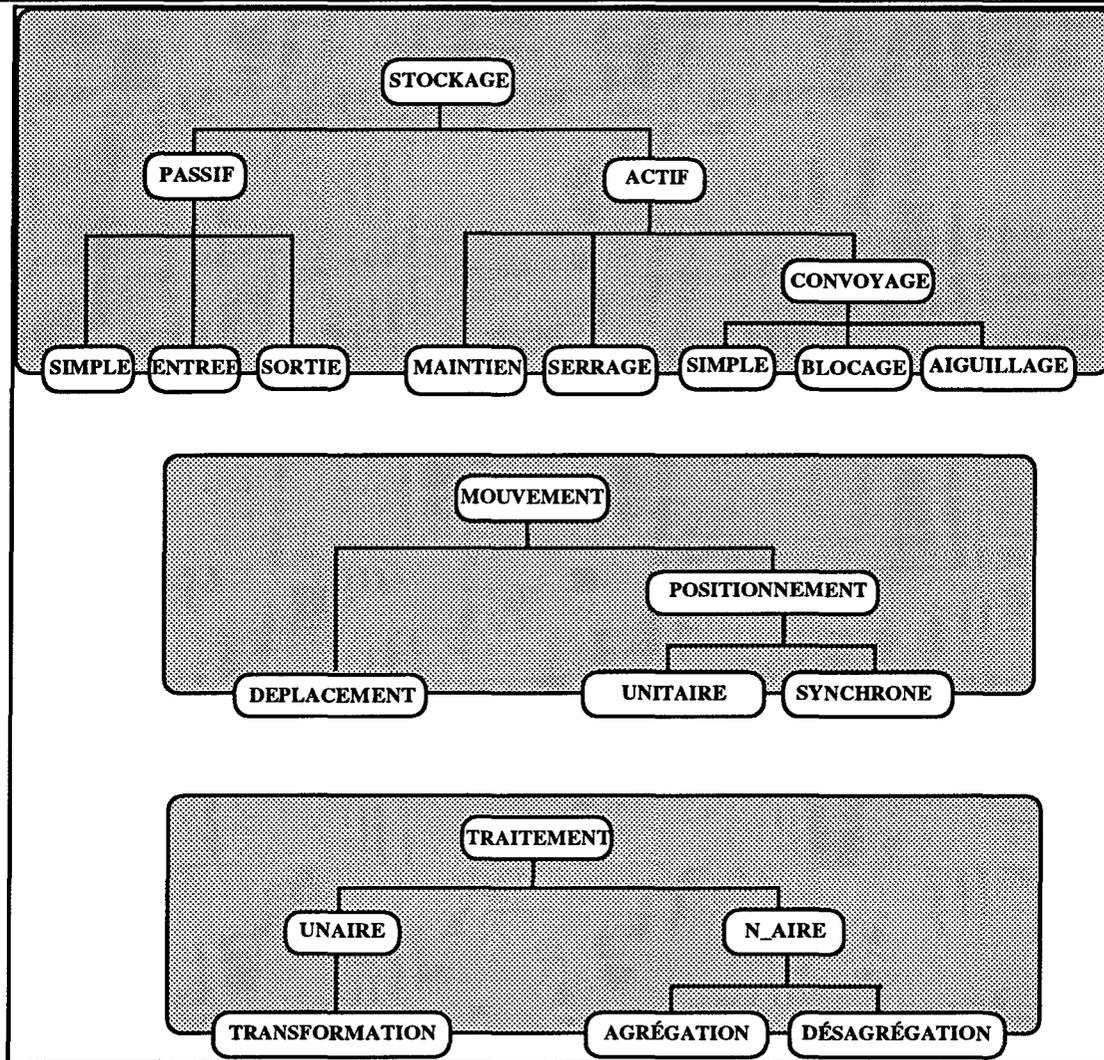
- les fonctions de type traitement représentent l'ensemble des opérations effectuées sur les produits. Ces fonctions ne sont pas élémentaires et représentent en partie une combinaison des deux autres catégories.

### La description structuro-fonctionnelle

L'analyse structuro-fonctionnelle des ressources met en évidence les organes opératifs et leurs fonctions.

L'aspect structurel correspond à la représentation de la structure d'une ressource sous forme d'une arborescence de composition. Dans cette dernière, chaque noeud désigne soit la ressource pour le noeud initial, soit un composant intervenant dans la structure en tant que sous-organe intermédiaire.

L'aspect fonctionnel correspond à la spécification fonctionnelle et à la classification de chacun des organes opératifs [Amar, 90]. Le typage des fonctions va permettre ensuite d'assister la sélection des lieux caractéristiques. En effet, les organes définis comme emplacement de stockage seront les lieux caractéristiques participant à la génération des gammes opératoires.

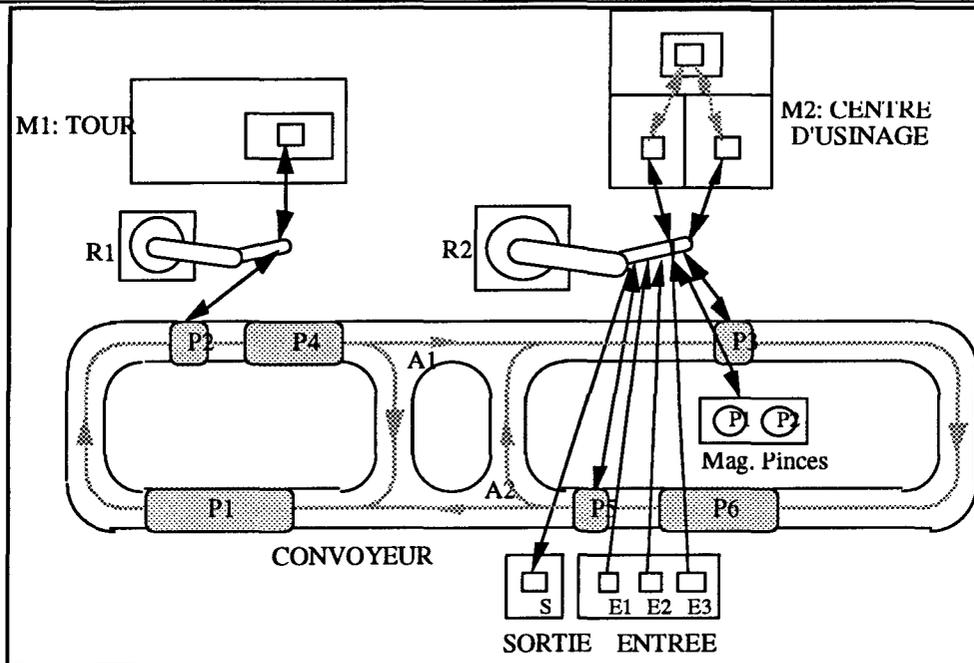


*Figure A4 : Bibliothèque fonctionnelle [Amar, 94]*

### Description de l'architecture physique

La description topologique du système de production caractérise les relations d'accessibilité et les possibilités d'échange de pièces entre les différentes ressources de production (machine, robot,...). Ces relations traduisent les premières contraintes de l'architecture physique du système de production. Les relations d'accessibilité sont associées au concept de lieu caractéristique (lieu physique qui constitue un lieu de travail ou qui est en relation d'accessibilité avec un lieu physique d'un autre moyen de production). Ce concept permet de définir deux classes de ressources : les ressources élémentaires qui ne comporte qu'un seul lieu caractéristique et les ressources complexes qui en comporte plusieurs ou qui contiennent au moins un lieu non caractéristique.

La figure A6 illustre ces concepts sur une description arborescente à deux niveaux d'une partie de la cellule de l'Ecole Centrale de Lille (figure A5). Seuls les lieux caractéristiques sont pris en compte pour chaque ressource. Ils sont associés à leurs fonctions et aux supports avec lesquels ils sont en relation externe.



*Figure A5 : Exemple d'atelier*

Les spécifications des parties logique et physique vont permettre le développement des différents modèles du système de commande. Elle permettront, en particulier, la génération automatique des gammes opératoires et la conception des modèles de comportement et de commande de la partie opérative.

Ces spécifications vont permettre l'élaboration d'une phase de "prototypage" qui permet d'implanter des gammes opératoires. Les validations statiques et dynamiques de ces gammes opératoires vont permettre de vérifier les propriétés du système modélisé.

L'évaluation des performances du modèle permettra de vérifier le respect des spécifications quantitatives. Une approche qualitative des gammes opératoires associées aux modèles de la partie opérative peut déterminer la "robustesse" du système. L'occurrence d'une perturbation (une relation d'accessibilité disparaît, une ressource de production n'est plus opérationnelle, ...) n'empêche pas le système d'assurer une certaine production en changeant de mode de marche ou de production. Ce prototypage, par un retour arrière rapide, peut aider à l'élaboration de l'architecture physique de production.

### **A.1.3. Spécification de la partie décisionnelle : les règles stratégiques**

Cette étape consiste à élaborer et à décrire des stratégies de pilotage du système pour lui conférer un comportement autonome, par résolution des conflits, et déterministe, par résolution des problèmes de routage temps réel des produits Cette étape est traitée dans [Hammadi, 91] et [Tawegoum, 94].

D'autre part, l'étude du niveau décisionnel vise à optimiser les flux de production en recherchant à la fois la meilleure façon d'insérer les entrées et d'opérer simultanément la résolution des conflits

d'accès [Ohl et al, 94], [Camus et al, 94]. La démarche considérée s'appuie sur la notion de gammes opératoires et intervient très tôt dans la démarche globale de conception. Elle concourt ainsi à l'aspect prototypage rapide et vise une fois de plus, dès les phases amonts de spécification, à assurer un premier niveau de validation.

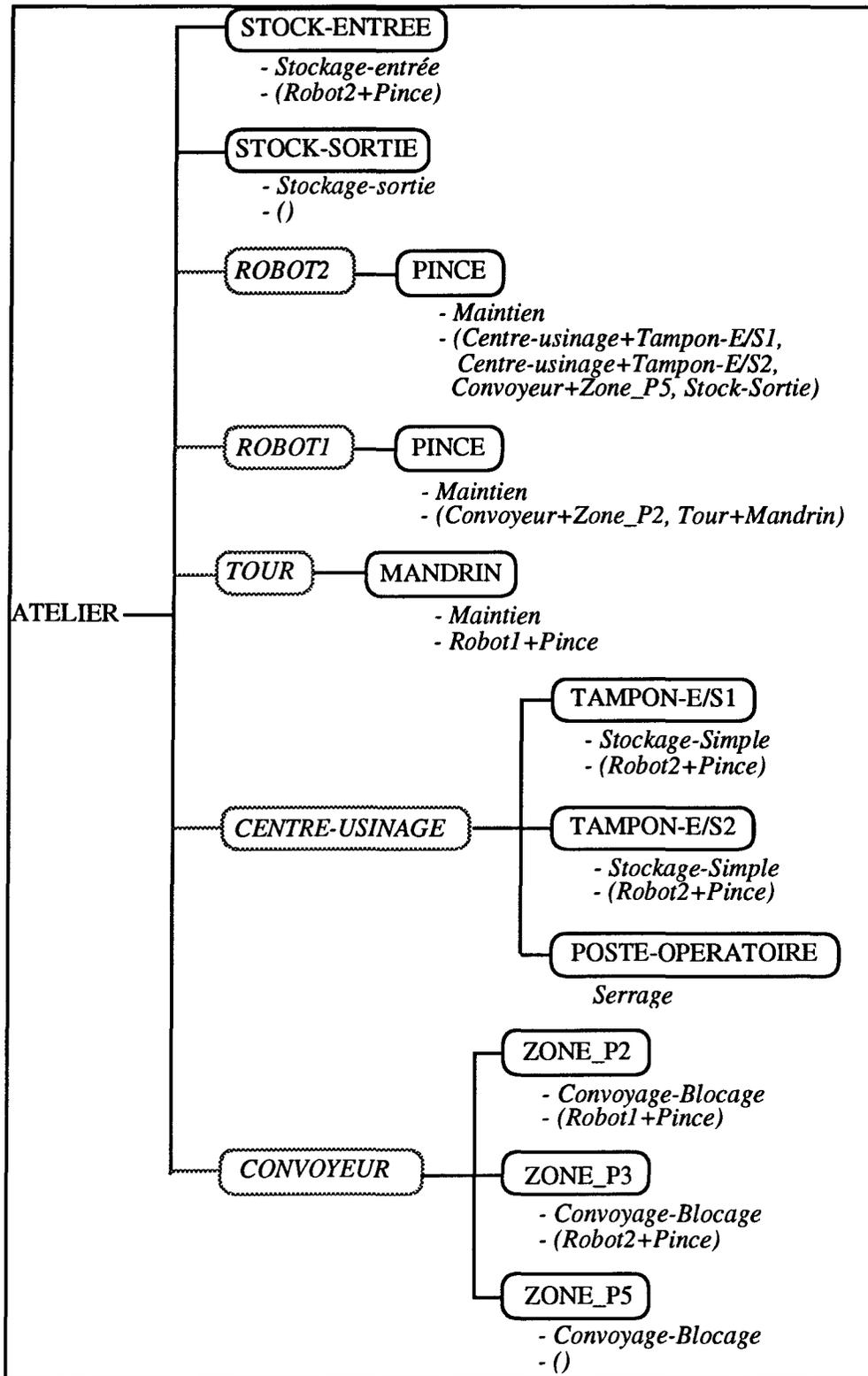


Figure A6 : Décomposition structuro-fonctionnelle, lieux caractéristiques et relations d'accessibilité

### A.1.4. Spécification de la supervision : la surveillance et la gestion des modes

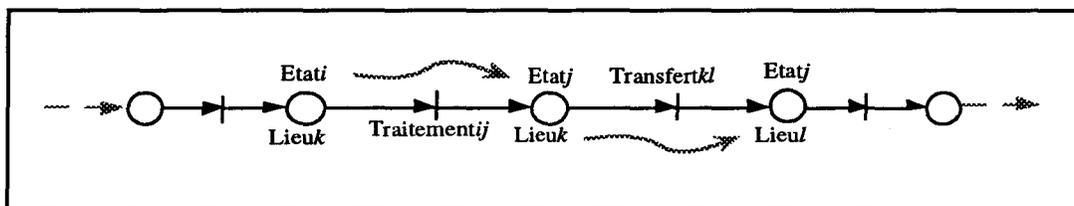
La spécification de la supervision permet de définir les types de surveillance et de gestion de modes. Cette spécification commence par la description des grandeurs à surveiller et la définition de l'observabilité du SFPM. Elle permet au concepteur de préciser les objectifs globaux de production en précisant à chaque fois l'organisation du SFPM et les contraintes de fonctionnement entre les ressources de production. C'est aussi l'occasion de préciser les seuils définissant le taux de dégradation d'une ressource ainsi que les paramètres à prendre en compte pour apprécier l'état du SFPM dans une configuration donnée.

La conception de la surveillance au L.A.I.L. a pris deux orientations complémentaires que nous présenterons par la suite. Nous verrons, en outre, que L'approche de la gestion des modes d'un SFPM proposée dans CASPAIM s'appuie sur le GEMMA. Nous montrerons que cette démarche ne suffit pas pour modéliser des systèmes où des ressources complexes cotoient des ressources simples. Nous montrerons, ensuite, que cette approche ne permet pas l'exploitation de toute la flexibilité des SFPM.

## A.2. Phase d'analyse et de conception

### A.2.1. Génération automatique des Gammes Opératoires [Amar, 94]

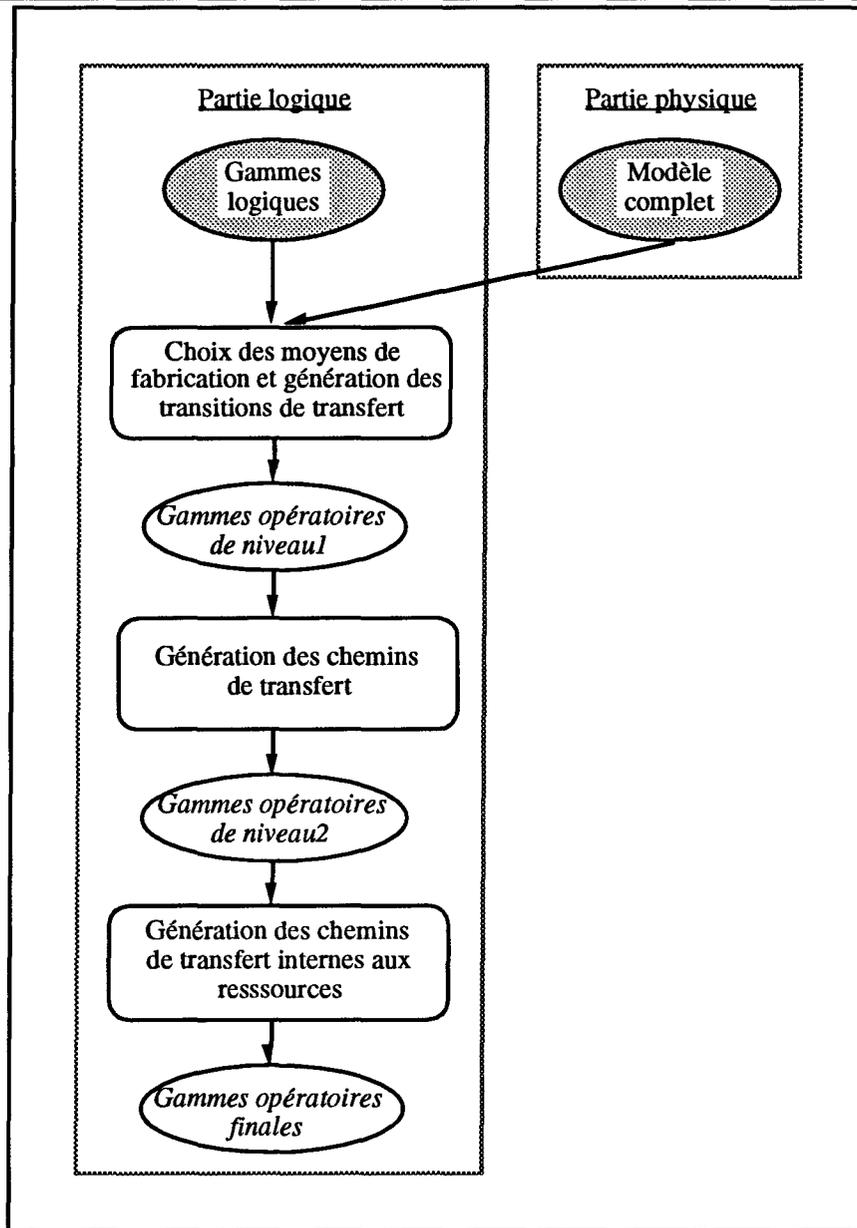
La phase de génération des gammes opératoires permet de développer les différentes gammes logiques en tenant compte notamment de l'aspect transitique de la partie physique. Sur le graphe RdP d'une gamme opératoire, chaque **place** représente, en plus de l'état fonctionnel hérité des gammes logiques, l'état positionnel d'un produit. Chaque **transition** représente soit un **traitement** (changement d'état), soit un **transfert** (changement de lieu) (figure A7).



*Figure A7 : description d'une gamme opératoire*

### Démarche de génération des gammes opératoires

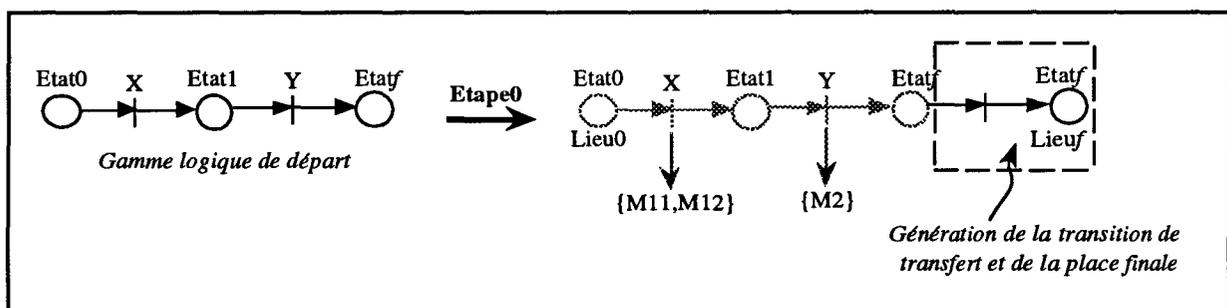
La démarche de génération se décompose en trois phases (figure A8) permettant d'obtenir l'ensemble des gammes opératoires :



*Figure A8: démarche de génération*

- Phase 1 : intégration de l'aspect fonctionnel

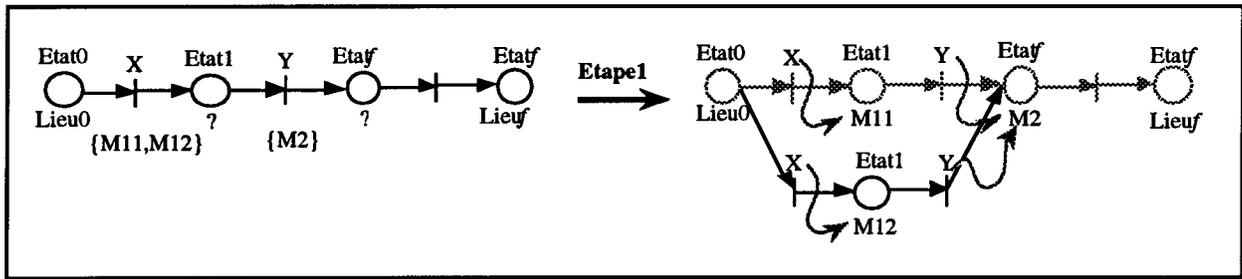
Cette phase initialise la démarche (étape 0) par spécification des lieux de départ et d'arrivée pour chaque gamme logique et des postes opératoires candidats (machines) pour chacun des traitements.



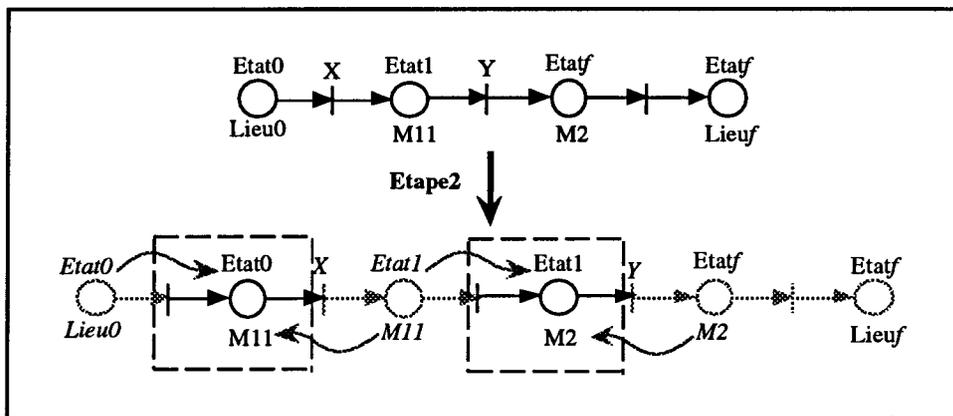
*Figure A9 : choix des lieux de fabrication*

## Annexe

Ensuite, ces postes opératoires sont effectivement intégrés au niveau des gammes (étape 1) et des transitions de transfert sont générées pour indiquer le passage d'un produit d'un poste vers un autre (étape 2).



*Figure A10: flexibilité de choix des machines*



*Figure A11: dissociation des transferts et traitements*

### - Phase 2 : intégration de l'aspect transitique

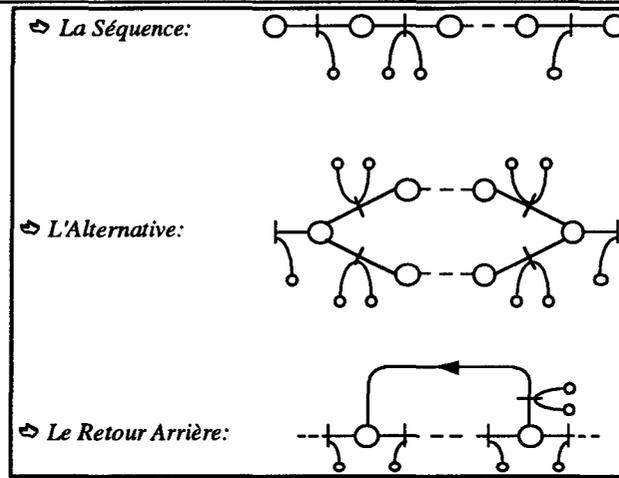
A ce niveau, les différents chemins possibles, pour transférer un produit entre deux machines consécutives au sein d'une gamme donnée, sont générés grâce aux relations d'accessibilité.

### - Phase 3 : intégration des modules de transfert

Cette dernière phase génère les gammes finales par intégration des modules de transfert spécifiques à chacune des ressources.

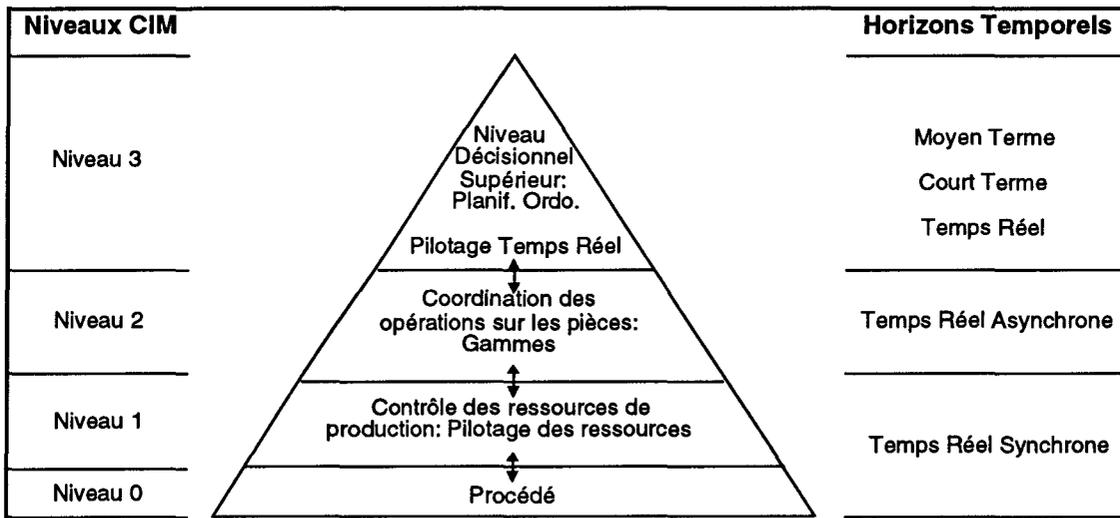
### Développement et extension des gammes opératoires

Les gammes opératoires décrivent le parcours des produits au travers des ressources de production. Les gammes opératoires étendues sont composées de trois types de structures de base (figure A12) : la séquence qui correspond à la structure minimale, l'alternative qui correspond à un indéterminisme (choix entre différentes ressources de production ou flexibilité entre différentes opérations) et le retour arrière nécessaire dans le cas d'une stratégie de contrôle particulière ou de rejets d'allocations (Nack).



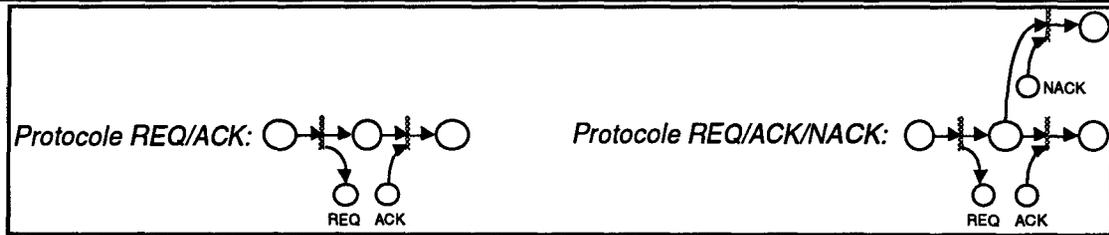
*Figure A12: Structure des gammes opératoires étendues*

Ces modèles caractérisent la partie coordination du contrôle/commande global d'un SFPM. Ils sont intermédiaires entre le Niveau Hiérarchique et le pilotage direct des ressources de production (figure A13).



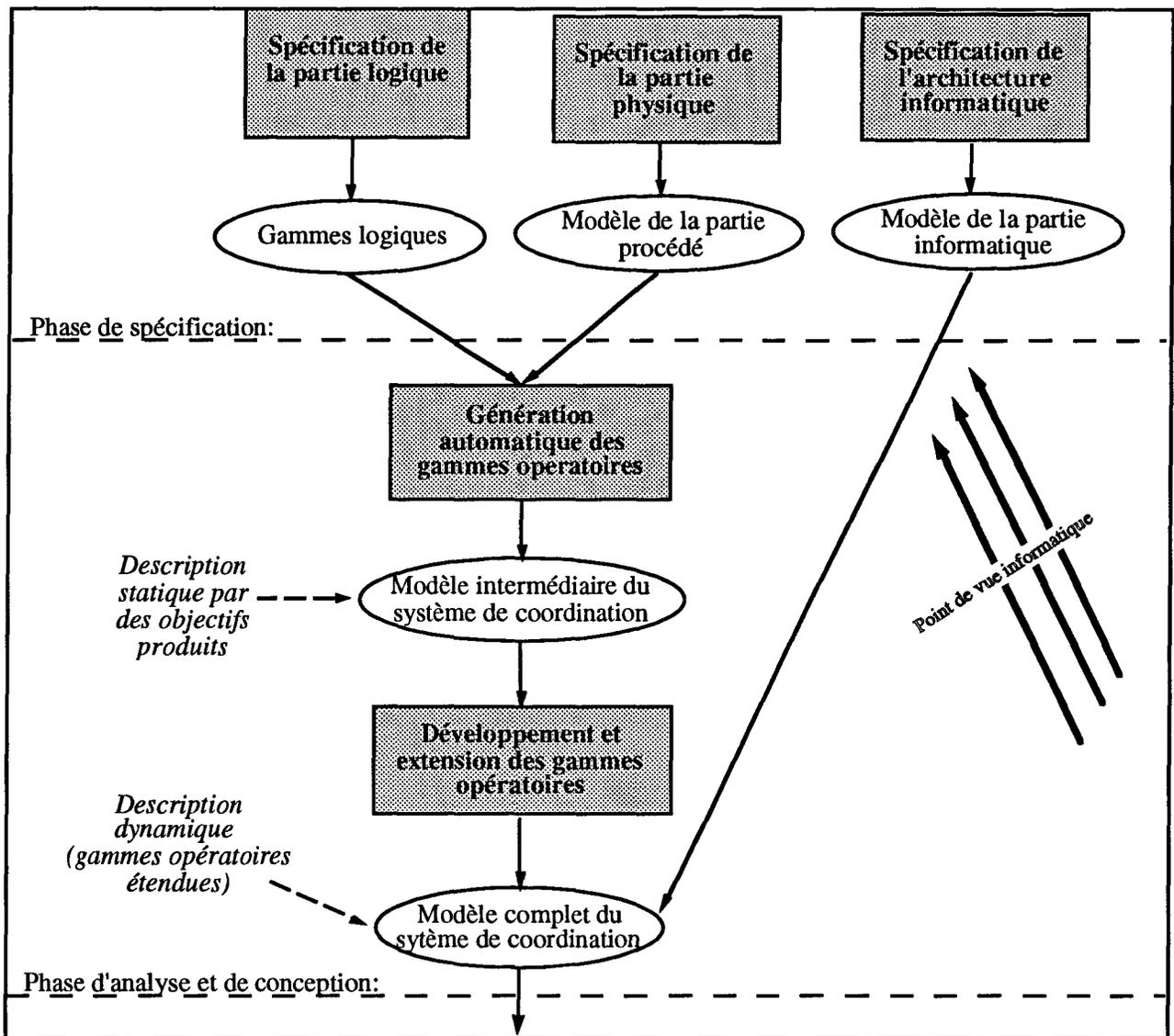
*Figure A13 : Hiérarchisation de la commande*

En effet, les gammes opératoires communiquent avec le niveau hiérarchique supérieur afin de résoudre leurs indéterminismes et émettent des requêtes vers les modèles des ressources opératives. Ces requêtes vont caractériser en particulier la nature du traitement à réaliser sur un type de produit donné ; elles permettront donc de lancer le programme adéquat de pilotage. Ces échanges inter-niveaux sont réalisés par des protocoles de communication qui respecteront la structure hiérarchique de la commande et permettront une distribution du code dans un environnement réparti et hétérogène (figure A14).



*Figure A14: Protocoles de communication au sein des gammes étendues*

Le développement des gammes ne vise pas simplement une meilleure lisibilité des modèles, il intègre la répartition des modèles sur leurs supports d'implantation. La démarche de conception présentée figure A peut ainsi être modifiée en partie selon la figure A15. Cette dernière montre que les calculateurs, commandes numériques, A.P.I., réseaux, ... sont autant de contraintes qui, si elles ne sont pas prises en compte à temps, vont remettre en cause le résultat de la conception qui peut être en quelque sorte considérée comme une étape de spécification pour un développement logiciel en vue de l'implantation réelle [Boulkroune, 93], [Desneulin, 94].

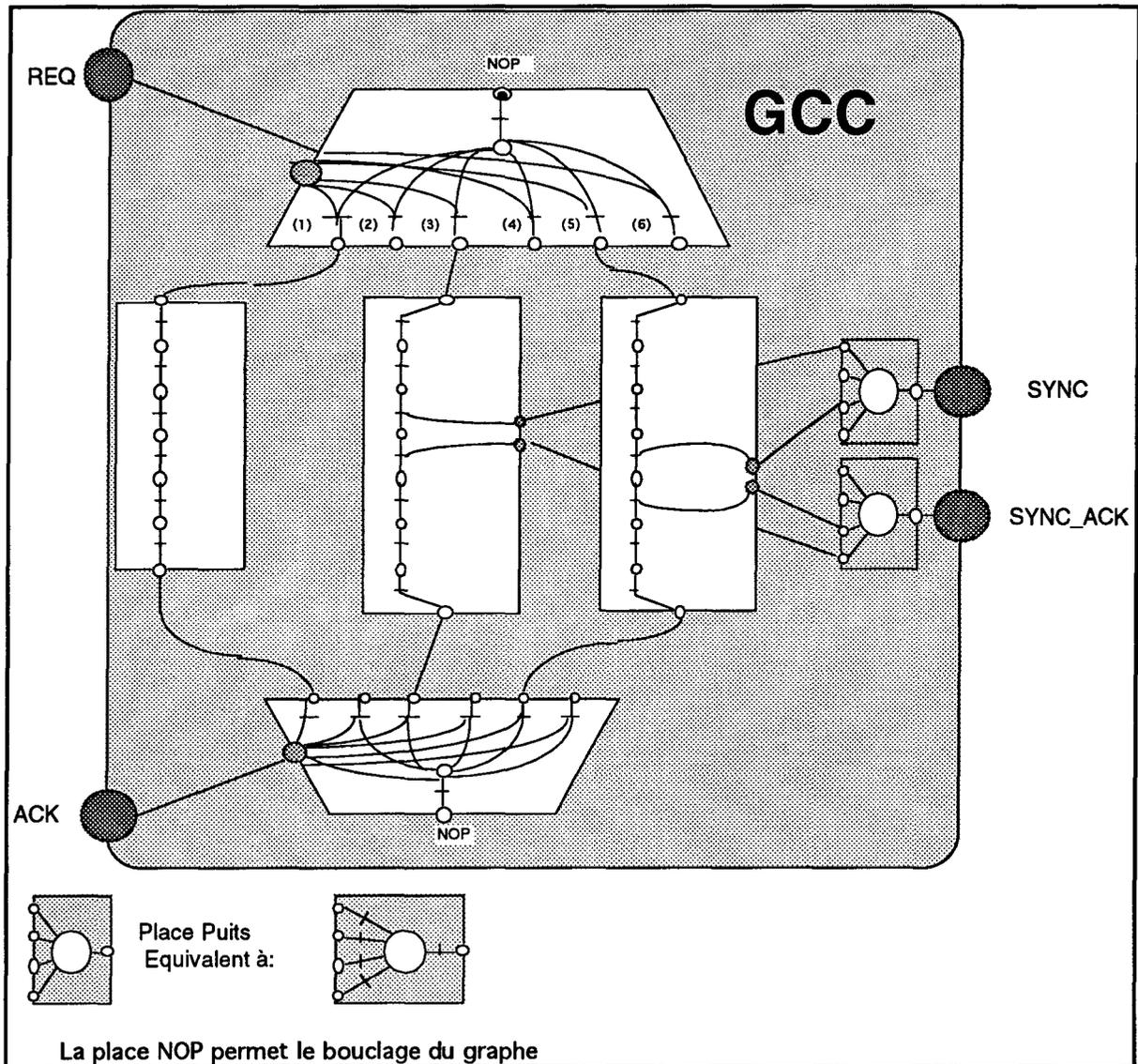


*Figure A15: Extension de la démarche de conception [Craye, 94]*

La description structuro-fonctionnelle a permis de distinguer deux types de ressources : les ressources simples et les ressources complexes.

### Les ressources simples

Une ressource simple ne subit aucun indéterminisme de fonctionnement. Sa commande sélectionne une séquence opératoire d'actions élémentaires en fonction de la nature de la ressource : choix du programme d'usinage pour les Machines Outils à Commande Numérique (M.O.C.N.), choix du type de transfert pour les robots de manutention,...



*Figure A16: Modèle des ressources simples*

Les graphes de commande des ressources simples sont obtenus à partir d'une bibliothèque de graphes de commande partiels construits avec deux primitives de base : la séquence et l'alternative. La synchronisation éventuelle est assurée par le couple requête et accusé de réception.

La figure A16 illustre différents cas de figure. Le choix numéro 1 correspond à un simple processus séquentiel sans synchronisation ; le choix numéro 3 modélise un processus qui émet une requête de synchronisation ; enfin le choix numéro 5 caractérise un processus qui attend une requête de synchronisation et répond par un accusé de réception avant de s'achever.

Les ressources complexes

le graphe de commande ne suffit plus pour caractériser une ressource complexe ; il est nécessaire de disposer en plus, d'un modèle (ou filtre) comportemental qui garantit la cohérence des divers sollicitations avec l'état global de la ressource. Le modèle comportemental décrit la ressource et sa dynamique d'évolution. Il peut inclure un résolveur d'indéterminismes capable de gérer la flexibilité interne de la ressource.

Les figures A17 et A18 illustrent la modélisation d'une ressource complexe : une partie du convoyeur de l'Ecole Centrale de Lille déjà présentée (figure A5). La figure A5 décrit structurellement le convoyeur tel qu'il est physiquement implanté. La figure A17 correspond à la modélisation de graphes de commande élémentaires et génériques correspondant au contrôle d'une butée et d'un aiguillage de ce convoyeur. Il sera nécessaire d'instancier ces graphes de commande génériques pour obtenir le graphe de commande de chaque butée et de chaque aiguillage. Enfin il est nécessaire d'associer à ces commandes un modèle comportemental afin d'assurer la cohérence entre les commandes élémentaires, d'une part, et l'état de la ressource complexe, d'autre part (figure A18). C'est l'état du modèle comportemental qui permettra de décider si l'envoi d'une palette dans un secteur par le positionnement d'un aiguillage est autorisé ou non. La commande de positionnement est filtrée et, éventuellement, différée.

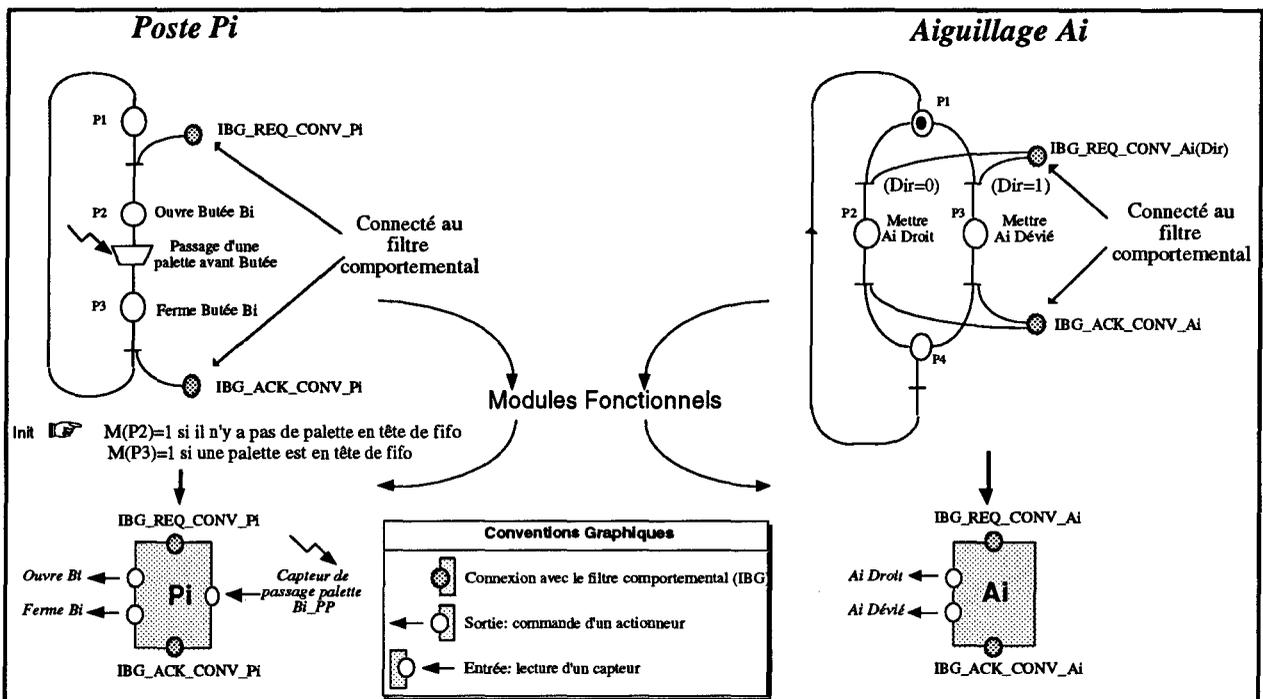
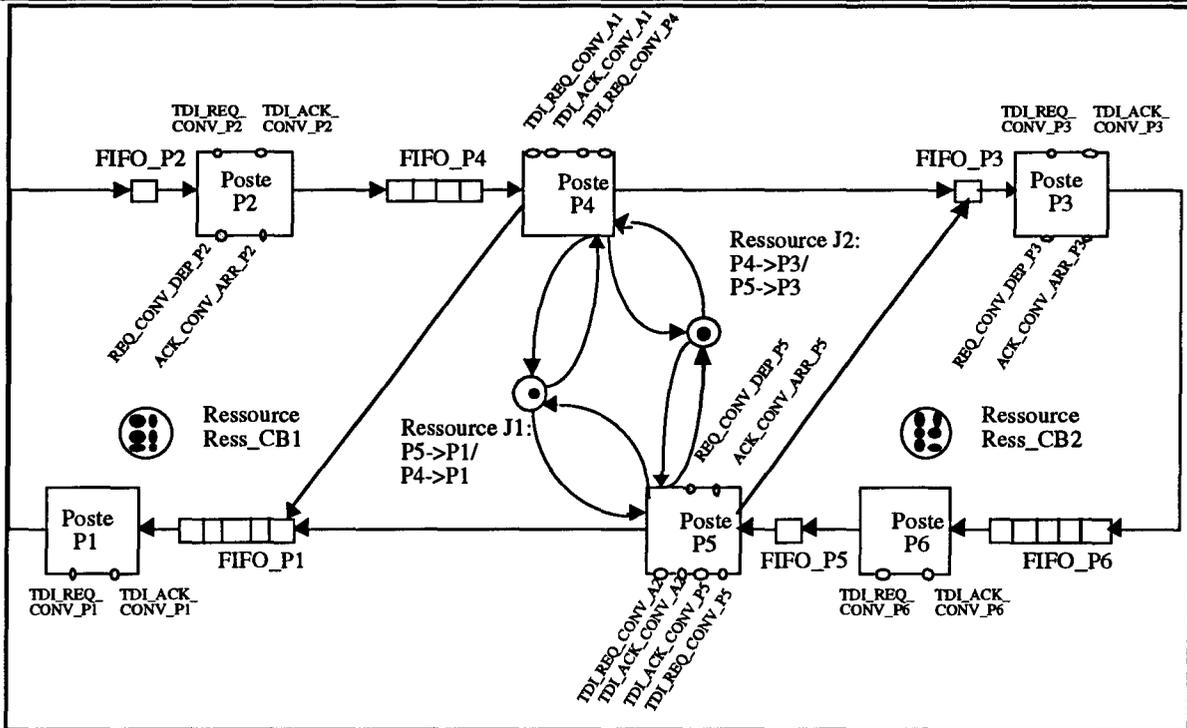


Figure A17 : Graphes de commandes génériques du convoyeur [Huvénait, 94]



*Figure A18 : Modèle comportemental du convoyeur [Huvénait, 94]*

### **A.3. Phase de validation**

La validation est explicite à ce niveau car il s'agit d'une validation globale du système de contrôle/commande. Chaque phase de la conception doit être validée localement. Cette validation locale est implicite. La validation que nous proposons de décrire ici repose sur deux démarches : une validation statique des gammes et une validation dynamique du système.

#### **A.3.1. Validation statique des gammes**

La validation statique des gammes est principalement :

- une démarche d'analyse des propriétés de terminaison propre et de vivacité du modèle de description des gammes afin d'assurer que toute gamme pourra être effectivement réalisée et que toutes les variantes prévues seront possibles [Cruette, 91],
- une méthode de recherche des blocages liés à une saturation du système de transport [Ausfelder, 94] ou d'une étreinte fatale liée à des utilisations croisées de ressources.

#### **A.3.2. Validation dynamique du système**

Deux approches sont possibles pour valider dynamiquement une solution : les démarches analytiques et les démarches de simulation.

Seule l'approche par simulation a été mise en œuvre complètement [Ausfelder, 94]. Certains travaux sont en cours [Ohl et al, 94] pour intégrer une démarche analytique basée sur les Réseaux

L'approche analytique permet d'effectuer des validations préliminaires et intermédiaires du modèle de coordination (gammes) en cours de développement. Le développement progressif des gammes et la représentation en niveaux devraient favoriser cette approche.

**A.4. Phase d'implantation**

L'intégration des modèles décrits précédemment permet d'obtenir la structure complète de la commande d'un SFPM (figure A19). Nous retrouvons au niveau le plus élevé de cette architecture, le Niveau Hiérarchique (NH) ou Décisionnel qui prend en compte la planification et l'ordonnancement. Le NH contrôle les différentes gammes opératoires étendues qui vont solliciter les gestionnaires de ressources. Le gestionnaire intégrera ou non un filtre comportemental suivant le type de la ressource commandée.

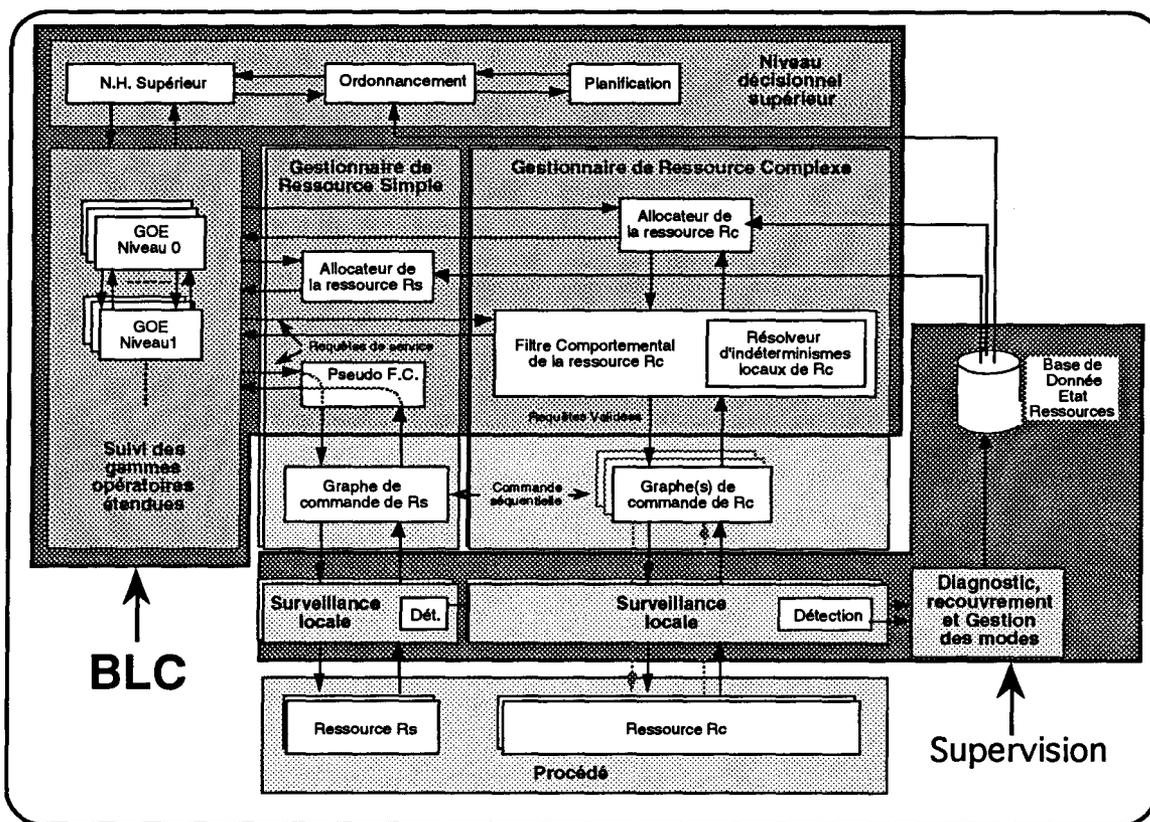


Figure A19 : Architecture de contrôle d'un SFPM.

Des modules de surveillance locale et directe [Elkhattabi, 93] s'insèrent entre les graphes de commande et les ressources. En cas de détection d'une défaillance, le gestionnaire des modes reconfigurera les données caractéristiques de l'état des ressources. Ces données serviront ensuite aux allocateurs de ressources qui rejeteront ou mettront en attente les demandes d'allocation.

Les premiers travaux de CASPAIM concernant l'implantation [Craye, 89] sont aujourd'hui totalement remodelés pour tenir compte des profondes modifications apportées au projet

CASPAIM depuis 1991. [Huvenoit, 94] propose une nouvelle approche d'implantation basée en grande partie sur une transposition des modèles de CASPAIM en langage ADA. Des travaux plus récents en cours de développement utilisent HOOD pour définir une méthode et une architecture informatique de gestion/contrôle de référence pour l'implantation [Farah, 93].

#### **A.4.1. Implantation avec ADA [Huvenoit, 94]**

L'architecture logicielle et matérielle découle de l'architecture de contrôle proposée dans la figure A19. L'organisation modulaire de cette architecture fonctionnelle (appelée aussi **Base Logicielle Commune**) fait apparaître trois niveaux principaux : le niveau hiérarchique ou décisionnel supérieur, le suivi des gammes opératoires et le pilotage des ressources.

Les deux premiers niveaux manipulent des données abstraites. Le premier niveau assure le pilotage et l'organisation globale de la production. Le second niveau coordonne les transferts et les opérations. L'implantation de ces deux niveaux peut se faire sur un ordinateur non spécialisé de type mini (éventuellement micro) ordinateur.

L'implantation du troisième niveau dépend des ressources. Un ordinateur spécialisé de type commande numérique, automate programmable ou micro ordinateur industriel suffira pour implanter le graphe de commande d'une ressource simple quel que soit le langage cible (grafcet, LM,...). Si la ressource est complexe, il faut associer à son graphe de commande un filtre comportemental qui peut être une structure de données ou un modèle dynamique complexe de type SED.

#### **A.4.2. Implantation avec la méthode HOOD [Farah, 95]**

Le caractère hiérarchique et modulaire de CASPAIM favorise en conception logicielle l'utilisation de techniques orientées objets selon un raffinement de type "Top-Down" similaire à celui utilisé dans HOOD. La structure du contrôle/commande dans CASPAIM offre un bon exemple de structuration, d'abstraction et de décomposition de l'information. Cette structure est adaptée aux principes de la méthode HOOD.

L'approche utilisant les concepts de la méthode HOOD est aujourd'hui mise en œuvre à l'aide d'un Atelier de Génie Logiciel. Elle en est néanmoins à ses débuts et cherche encore à automatiser le passage des spécifications RdP et StateCharts (ou Argos) de CASPAIM vers des objets HOOD (figures III.5). Il reste également à étudier la distribution des objets en vue d'une implantation répartie ainsi qu'un premier niveau de validation au travers d'un exemple sur site industriel réel. L'approche dépasse la simple transposition des modèles issus de l'analyse.

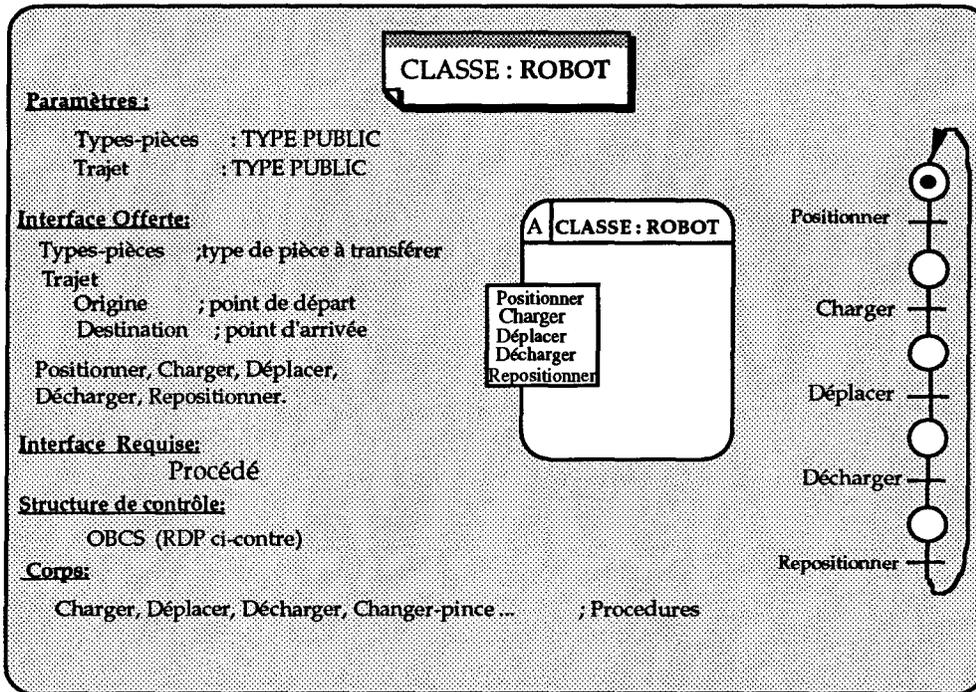


Figure A20 : Association HOOD et RdP

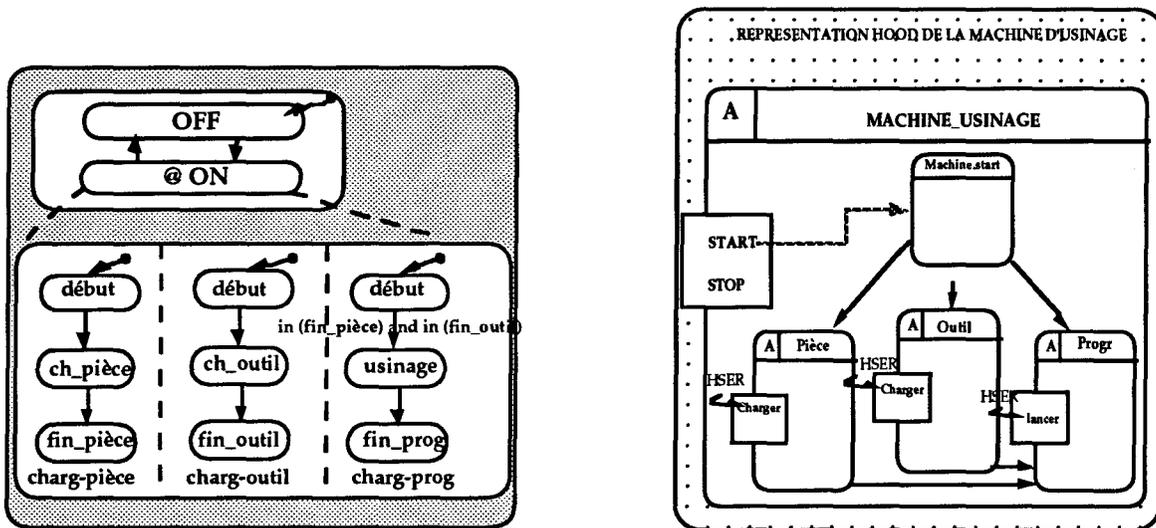


Figure A21: Association HOOD et StateCharts

**Conclusion**

Cette annexe a permis de présenter plusieurs outils de modélisation ainsi qu'une méthodologie de conception qui vise à appréhender la complexité des systèmes flexibles de production. L'association de ces outils et méthodes permet d'aboutir à la définition de modèles décrivant le contrôle/commande du système de production. Ces modèles concernent les produits et la définition de leur gamme de production, les ressources de production et leurs comportements mais également les stratégies de production, la conduite temps réel,...

## **RÉFÉRENCES BIBLIOGRAPHIQUES**

## Références bibliographiques

---

- [ADEPA, 79] : Le GEMMA : Guide des Modes de Marche et d'Arrêt. ADEPA, 1979.
- [ADEPA, 92] : N.Bouteille, P. Brard, G. Colombari, N. Cotaina et D. Richet ; "Le grafcet", Cépaduès-Editions, mars 1992.
- [Afn, 86] : AFNOR, Norme E 09-053 ; "Prévention Technique des Accidents, du Fait des risques Engendrés par les Machines". 1986.
- [Alanche et al, 86] : P. Alanche, P. Lhoste, G Morel, M. Roesch, M. Salim and P. Salvi ; "Modélisation de la Partie Opérative et Application à la Structuration de la commande des systèmes automatisés" ; journées AFCET : " Méthodes et Outils Modernes de conception et d'exploitation de la commande des procédés discontinus complexes " ; Montpellier, Mars 1986.
- [Amar, 90] : S. Amar , E. Castelain, J.C. Gentina ; "Modélisation des moyens de production par langages orientés objet en vue de la conception de la commande d'un système de production flexible" ; Congrès CIM90, Proc. pp. 323-331, Bordeaux 12-14 juin 1990.
- [Amar, 94] : S. Amar ; "Méthode de conception préliminaire du système de coordination des systèmes de production flexibles par prototypage orienté objet de la partie procédé". Thèse de Doctorat d'Université de Lille I, avril 1994.
- [Anakök, 87] : Y. Anakök ; "Une méthode d'analyse et de synthèse de la partie commande des machines et systèmes automatisés de production". Thèse de Doctorat de l'ISMCM, juillet 87.
- [André et al, 93] : C. André et M.A. Peraldi ; "Synchronous Programming : Introduction and Application to Industrial Process Control". CompEURO'93, pp 461-470, Every, Mai 1993.
- [Appelrath et al, 91] : H-J. Appelrath, and R. Bruns ; "Ein universelles Modell für Ablaufplanungsprobleme". Wirtschaftsinformatik, Vol. 33, N° 6, 1991; pp. 516 - 525.
- [Archimede, 91] : B. Archimede ; "Conception d'une Architecture réactive, distribuée, hiérarchisée pour le Pilotage des Systèmes de Production"; Thèse de Doctorat; Bordeaux, 1991.
- [Ausfelder et al, 92] : C. Ausfelder, E. Castelain and J-C. Gentina ; "An Object Oriented Simulation Tool to Validate the Dynamic Behaviour of FMS". European Simulation Multiconference ; York, United Kingdom ; 1-3 juin, 1992. pp 528..

## Références bibliographiques

- [Ausfelder, 94] : C. Ausfelder ; "Contribution à la conception d'un système de conduite pour les systèmes flexibles de production manufacturière : modélisation et validation de la commande". Thèse de Doctorat d'Université de Lille I, mars 1994.
- [Barbier et al, 92] : F. Barbier, P. Jaulent ; "Techniques orientées objet et CIM". Edition Eyrolles, 1992.
- [Bayart et al, 92] : M. Bayart, H. Gueguen ; "Les processus dis-continus". Rapport du groupe de travail AFCET/SEE, 1992.
- [Bayart et al, 95] : M. Bayart, M. Staroswiecki, A.L. Gehin ; "Mode Management in a distributed Intelligent Automated Production System". IFAC Conference on Algorithms and Architectures for Real Time Control Systems, AARTC'95. Ostende, 31 mai- 2juin 1995.
- [Bayart, 94] : M. Bayart ; "Modes de marche d'un système automatisé". Journées d'étude S3 : Stratégies de conduite en présence de défaillances, Paris, 8 décembre 1994.
- [Benassy, 87] : J. Benassy ; "La gestion de production". Edition Hermès, Paris, 1987.
- [Berry et al, 87] : G. Berry, P. Couronné and G. Gouthier ; " Programmation Synchrone des Systèmes Réactifs : le langage ESTEREL". TSI, vol.6, no. 4, (1987); pp. 305- 315.
- [Berry et al, 91] : G. Berry et A. Benveniste ; "The synchronous approach to Reactive and Real-Time Systems". Proceedings of the IEEE, Vol. 79, NO. 9, pp. 1270-1282, September 1991.
- [Biland, 94] : P. Biland ; "Modélisation des modes de marche d'un système automatisé de production"; Thèse de Doctorat, École Centrale de Nantes, février 1994.
- [Bois, 91] : S. Bois ; "Intégration de la gestion de modes de marche dans le pilotage d'un système automatisé de production". Thèse de doctorat de l'USTL, novembre 1991.
- [Bonetto, 1985] : R. Bonetto ; "Les ateliers flexibles de production". Edition Hermès, 1985.
- [Booch, 91] : G. Booch ; "Object Oriented Design, with applications". The Benjamin/Cunning Publishing Company, 1991.
- [Boulkroune, 93] : R. Boulkroune ; "Mécanismes pour la datation d'événements dans les systèmes distribués de production" ; D.E.A. de Productique de l'Université de Lille I, 28 juin 93.
- [Bourey, 88] : J.P. Bourey ; "Structuration de la partie procédurale du système de commande des cellules de production flexibles dans l'industrie manufacturière". Thèse de Doctorat de l'Université de Lille I, mars, 1988.

## Références bibliographiques

- [Bourey, 93] : J.P. Bourey ; "Méthode de conception de la commande de systèmes flexibles de production manufacturière". Habilitation à diriger des recherches, février, 1993.
- [Boussinot et al, 91] : F. Boussinot et R. De Simone ; "The ESTEREL Language", Proceedings of the IEEE, Vol. 7, NO. 9, pp. 1293-1304, September 1991.
- [Bradshaw, 91] : J. A. Bradshaw (University of Cambridge), " Device Representation. The Significance of Functional Knowledge", IEEE, APRIL 1991, pages 22-25.
- [Brams, 83] : G. W. BRAMS ; "Réseaux de Petri : Théorie et Pratique". Tome 1 : "Théorie et Analyse", MASSON 1983.
- [Brown et al, 85] : J. Brown, K.E. Stecke ; "Variation in flexible manufacturing systems according to the relevant types of automated material handling". Material Flow 2, pp. 179-185, 1985.
- [Camus et al, 94] : H. Camus, H. Ohl, E. Castelain et J.C. Gentina ; "A heuristic algorithm for the computational of cycle schedules and the necessary WIP to obtain optimal cycle time". Rensselaer's Fourth International Conference on CIM and Automation Technology, Vol. 1, pp. 339-344, Tron, NY, USA, Oct. 1994.
- [Carlier et al, 88] : J. Carlier et P. Chretienne ; "Problème d'ordonnancement : modélisation, complexité et algorithme". Masson (Collection Etude et Recherches en Informatique), Paris, 1988.
- [Castelain, 87] ; E. Castelain ; "Modélisation et simulation interactive de cellules de production flexibles dans l'industrie manufacturière". Thèse de Doctorat de l'Université de Lille I, février, 1987.
- [CEE, 91] : Communauté Economique Européenne ; "Directive Machines 89/392 modifiée 91/368. 1991.
- [Chandrasekaran et al, 89] : B. Chandrasekaran, J.W. Smith, J. Sticklen ; "Deep Models and Their Relation to Diagnosis". Artificial Intelligence in Medicine, Vol. 1, A989, pp. 29-40.
- [Chang et al, 89] : S.J. Chang, F. Diceseare, G. Godbogen ; "The generation of Diagnostic heuristics for Automated error recovery in Manufacturing Workstations". Proceedings of the IEEE International Conference on Robotics and Automation, pp. 38-43, September 1989.
- [Chassang et al, 83] : G. Chassang and H. Tron ; " Gérer la production avec l'ordinateur". Bordas, Paris (1983).

## Références bibliographiques

- [Combacau, 91] : M. Combacau ; "Commande et Surveillance des systèmes à événements discrets complexes : Application aux ateliers flexibles". Thèse de doctorat de l'Université Paul Sabatier de Toulouse, 1991.
- [Courvoisier et al, 84] : M. Courvoisier, M. Bigou, R. Valette, C. Desclaux et K. Benzakour ; "The S.C.O.I.A. Project". 6<sup>th</sup> European Conference on Electronics (EURO-CON), Brighton (UK), September, 1984.
- [Courvoisier et al, 89] : M. Courvoisier, R. Valette, AEK. Sahraoui et M. Combacau. "Spécification and implementation techniques for multilevel control of FMS". Third International IFIP Conference on Computer Application in Production & Engineering, CAPE'89, pp. 509-516, Tokyo (Japan), octobre 1989.
- [Craye, 89] : E. Craye ; "De la modélisation à l'implantation automatisées de la commande hiérarchisée de cellules de production flexibles dans l'industrie manufacturière". Thèse de Doctorat de l'Université de Lille I, janvier 1989.
- [Craye, 94] : E. Craye ; "Contribution au contrôle/commande de systèmes flexible de production manufacturière". Habilitation à Diriger de Recherches ; Université de Lille I, décembre 1994.
- [Cruette, 91] : D. Cruette ; "Méthodologie de conception des systèmes complexes à évènements discrets: application à la conception et à la validation de la commande de cellules flexibles de production dans l'industrie manufacturière". Thèse de Doctorat d'Université de Lille I, février 1991.
- [Daenzer, 87] : G. Daenzer ; "Systems Engineering". Verlag Industrielle Organisation, Zurich 1987.
- [Desneulin, 94] : P. Desneulin ; "Synchronisation d'horloges et temps physiques" ; D.E.A. de Productique de l'Université de Lille I, septembre 94
- [Elkhatabi et al, 92] : S. Elkhatabi, D. Corbeel and J.C. Gentina ; "Integration of Dependability in the Design of FMS". Proceedings of IFAC-ICOM; Toronto, Canada; May 25-28, 1992; pp. 249 - 254.
- [Elkhatabi, 93] : S. Elkhatabi ; "Intégration de la surveillance de bas niveau dans la conception des systèmes à événements discrets : application aux systèmes de production flexibles.". Thèse de Doctorat de l'université de Lille I. Septembre, 1993.

## Références bibliographiques

- [Farah et al, 95] : A. Farah, E. Craye et J.P. Bourey ; "Une méthode hiérarchique orientée objets pour l'implantation répartie du système de contrôle/commande d'un SFPM.". Actes de la 2<sup>ème</sup> Conférence Internationale sur l'Automatisation Industrielle. AIAI, Nancy, 7-9 juin 1995.
- [Farah, 91] : A. Farah ; "Détection et diagnostic de pannes dans les ateliers de production flexibles de l'industrie manufacturière" ; Rapport de DEA, LAIL/Lille1, 1991.
- [Ferreira et al, 91] : J.F. Ferreira-Ribeiro and B. Pradin ; "Selection of Manufacturing Cells for Group Technology". Proceedings of the 13th IMACS world congress, Lille, France, mai 1991; pp 675 - 681.
- [Froment et al, 84] : B. Froment, JJ. Lesage ; "PRODUCTIQUE : les techniques d'usinage flexible". Edition Dunod, Série Génie Mécanique, 1984.
- [Fu et al, 92] : Li-Chen FU and Pei-Sen LIU ; "Hierarchical Dynamical Scheduling for a Flexible Manufacturing System". 3<sup>rd</sup> International Conference on CIM, Troy, N.Y. May 20 -22, 1992; pp. 393 - 401.
- [Gherardi, 92] : G. Gherardi ; "SAHARA : un environnement de mise au point pour les programmes ESTEREL". Thèse de Doctorat de l'Université de Nice-Sophia Antipolis, Decembre 1992.
- [Gondran et al, 79] : M. Gondran, M. Minoux ; "Graphes et Algorithmes" ; Editions Eyrolles, 1979.
- [Halbwachs et al, 91] : N. Halbwachs, P. Caspi, P. Raymond et D. Pilaud ; "The Synchronous Data Flow Programming Language LUSTRE". Proceedings of the IEEE, Vol. 7, NO. 9, pp. 1305-1320, September 1991.
- [Hammadi, 91] : S. Hammadi ; "Une méthode d'ordonnancement minimisant les temps d'attente et de transit dans les systèmes de production flexible de type JOB-SHOP". Thèse de doctorat de l'Université des Sciences et Techniques de Lille, 1991.
- [Harel, 87] : D. Harel ; "STATECHARTS : a visual formalism for complex systems". Science of computer programming, n°8, 1987.
- [Hutchison, 79] : G.K. Hutchison ; "Flexible Manufacturing Systems in United States". Automaton in Manufacturing, 1979.
- [Huvenoit, 94] : B. Huvenoit ; "De la conception à l'implantation de la commande modulaire et hiérarchisée de systèmes flexibles de production manufacturière". Thèse de Doctorat d'Université de Lille I, octobre 94.

## Références bibliographiques

- [I.C.A.M.] : I.C.A.M. ; "Integrated Computer-Aided Manufacturing". Architecture ICAM project, part II, Volume IV-Function modelling manual (IDEF0), US Armed Forces, USA.
- [I.G.L. Technology:, 89] : I.G.L. Technology ; "SADT : un langage pour communiquer". EYROLLES, France (1989).
- [Ingersoll Engineers, 88] : Ingersoll Engineers ; "L'usine intégrée". Edition Hermès, Paris, 1988.
- [ISO 9000] : "Gérer et assurer la qualité". Norme AFNOR, 1992.
- [Kapusta, 88] : M. Kapusta ; "Génération assistée d'un graphe fonctionnel destiné à l'élaboration structurée du modèle de la partie commande pour les cellules de production flexibles dans l'industrie manufacturière". Thèse de Doctorat de l'Université de Lille I, décembre, 1988.
- [Kermad et al, 93a] : L. Kermad, JP. Maik, C. Ausfelder, J.C. Gentina, D. Delfieu, R. Moisand, AEK. Sahraoui ; "Integration of Operating Modes in the Control of Flexible Manufacturing Systems Combining Synchronous and Asynchronous Approaches". Proceedings of IEEE International Conference on Systems, Man and Cybernetics. VOL. 2, pp 234-239. Le Touquet, France, October 17-20, 1993.
- [Kermad et al, 93b] : L. Kermad, C. Ausfelder, J.P. Bourey et E. Castelain ; "Integrative approach for a functional specification of FMS control". Revue CIM, Vol. 6 n°4, pp. 219-227, novembre, 1993.
- [Kermad et al, 94a] : L. Kermad, J.P. Bourey, J.C. Gentina ; "Reactive Real-Time Exploitation of Flexible Manufacturing Systems". Proceedings of the 2nd IEEE Workshop on Real-Time Applications. pp 116-121. Washington, DC. July 21-22, 1994
- [Kermad et al, 94b] : L. Kermad, E. Craye, J.P. Bourey et J.C. Gentina ; "The working and exploiting mode in FMS". Proceedings of the IEEE International Conference on Systems, Man and Cybernetics. Vol. 3, pp. 1521-1526. San Antonio, Texas, Oct. 2-5 1994.
- [Kermad et al, 96] ; L. Kermad, C. André et J.C. Gentina ; "Approche synchrone des modes de marche et d'exploitation". à paraître dans les actes du congrès AFCET "Modélisation des Systèmes Réactifs", Brest, 28 et 29 mars 1996.
- [Keuneke, 91] : A. M. Keuneke (California State University), "Functional reasoning and functional modeling", IEEE, APRIL 1991.

## Références bibliographiques

- [Kleer et al] : J. de Kleer, J.S. Brown ; "A Mental Model of Physical mechanisms and their Acquisition". In Cognitive Skills and their Acquisition, J.R. Anderson, ed Lawrence Erlbaum Associates, Hillsdale, N.J., 1980, pp. 285-309.
- [Le Guernic et al, 91] : P. Le Guernic, T. Gautier, M. Le Borgne et C. Le Maire ; "Programming Real-Time Applications with SIGNAL". Proceedings of the IEEE, Vol. 7, NO. 9, pp. 1321-1336, September 1991.
- [Le Moigne, 90] : J.L. Le Moigne ; "La modélisation des systèmes complexes". Afcet Systèmes, Dunod (1990).
- [Lhoste, 91] : P. Lhoste ; "Surveillance des M.S.A.P. : les Atouts de la Modélisation de comportement". Journées d'études du Groupe GT2 du pôle SED, Paris, 7 février 1991.
- [Lissander, 90] : A. Lissander ; "Maitriser SADT". Edition COLIN, Paris, France, 1990.
- [Maik et al, 94] : J.P. Maik, D. Corbeel et J.C. Gentina ; "Objets dynamiques et évolutifs pour le système d'information des systèmes de production manufacturière". Colloque AGI'94 (Automatique, Génie informatique, Image), pp 157-160, Poitiers 2-3 juin 1994.
- [Maraninchi, 89] : F. Maraninchi ; "Argonaute, graphical description, semantics and verification of reactive systems by using a process algebra". Workshop on Automatic Verification methods for Finite State Systems, Grenoble, June 1989.
- [Maraninchi, 90] : F. Maraninchi ; "Argos: Un langage graphique pour la conception, la description et la validation des systèmes réactifs". Thesis, University of Grenoble, 1990.
- [Maraninchi, 92] : F. Maraninchi ; "Language for reactive systems: a common framework for comparing StateCharts and Argos". Spectre Report C34, LGI-IMAG, Grenoble, février 1992.
- [Millerat, 94] : P. Millerat ; "PPRD MINOTAURE : Méthode de Conception d'un Système de Pilotage d'un REP. De l'analyse du besoin à la réalisation d'études technico-économiques". HP-33/94/025. EDF-DER., 1994.
- [Milot, 88] : P. Millot ; "Supervision des procédés automatisés et ergonomie". Traité des Nouvelles Technologies, série Automatique, Edition Hermès, 1988.
- [Morel, 92] : G. Morel ; "Contribution à l'automatisation et à l'ingénierie des Systèmes Intégrés de Production". Habilitation à diriger des recherches. Université de Nancy I, 1992.
- [Murata, 89] : T. Murata ; "Petri Nets : Properties, Analysis and Applications". Petri Nets Proceedings of the IEEE, April 1989.

## Références bibliographiques

- [O'Grady et al, 92] : P. O'Grady and R. Seshadri ; "Operation of X-Cell, an intelligent cell control system". Computer Integrated Manufacturing Systems; Vol.5, Num. 1 (1992); pp 21-30.
- [Ohl et al, 94] : H. Ohl, E. Castelain et JC. Gentina ; "Synchrony Theory applied to control problems in flexible Manufacturing Systems". proceedings of IEEE/SMC'94, Vol. 2, pp 1689-1694, San Antonio, Texas, October 2-5, 1994.
- [Parayre, 92] : T. Parayre ; "Le MESAP : vers une méthodologie d'exploitation des système automatisés de production". Thèse de Doctorat de l'Université de Valenciennes et du Hainaut Cambrésis, mars 1992.
- [Pourcel, 91] : C. Pourcel, P. Pourcel ; "Conduite des procédés continus et manufacturiers : une même approche ?". 3<sup>ème</sup> Congrès International : Le Génie Industriel, Tours (France), mars 1991.
- [RAC, 91] : Reliability Analysis Center, None Electronic Parts Reliability Data, 1991.
- [Rakotoson, 93] : M.P. Rakotoson ; "Synthèses des caractéristiques et techniques de développement de la commande des systèmes dis-continus : application aux systèmes de production flexibles batchs". Thèse de Doctorat, Université de Lille I, juillet 1993.
- [Ranky, 90] : P. Ranky ; "Flexible Manufacturing Cells and Systems in CIM."; CIMware Ltd, Guildford, U.K. (1990).
- [Rodde, 89] : G. Rodde ; "Les systèmes de production : modélisation et performance". Edition Hermès, Paris, 1989.
- [Roubellat et al, 88] : F. Roubellat, and V. Thomas ; "Une méthode et un logiciel pour l'ordonnement en temps réel d'ateliers". APII, vol 22, no 5, (1988); pp. 421- 438.
- [Sahraoui, 87] : A.E.K. Sahraoui ; "Contribution à la surveillance et à la commande d'atelier". Thèse de Doctorat de l'Université de Toulouse, 1987.
- [Sahraoui, 92] : A.E.K. Sahraoui ; "An approach for monitoring discrete event systems". AFCET APII, Edition Dunod, Vol. 26, 1992, n°2, pp. 91-106.
- [Sahraoui, 93] : A.E.K. Sahraoui ; "Contribution au domaine de la spécification des systèmes réactifs complexes". Habilitation à Diriger les Recherches de l'Université Paul Sabatier, Toulouse, 1993.
- [Sahraoui, 95] : A.E.K. Sahraoui ; "A multi formalism paradigm for co-design of avionics". IEEE Conference on Rapid Prototyping and Co-Design, Atlanta, june 95.

## Références bibliographiques

---

- [Schmidt, 89] : G. Schmidt ; " CAM : Algorithmen und Decision Support für die Fertigungsplanung". Springer-Verlag, Berlin, 1989.
- [Sembougamoorthy et al, 86] : V. Sembugamoorthy et B. Chandrasekaran ; "Functional Representation of Devices and Compilation of Diagnostic Problem Solving Systems", in J. L. Kolodner and C. K. Riesbeck, Experience, Memory and Reasoning (Ed.), Lauwrence Erlbaum Associates, 1986.
- [Sethi et al, 90] : A. K. Sethi et S. P. Sethi ; "Flexibility in Manufacturing : A survey". The International Journal of Flexible Manufacturing Systems, vol. 2, Num. 4, pp 289-328, 1990.
- [Shin, 89] : O. K. Shin ; "Contribution au pilotage d'îlots flexibles d'assemblage". Thèse de Doctorat de l'Université de Franche-Comté, 1989.
- [Simone et al, 89] : R. De Simone and D. Vergamini ; "Aboard AUTO". Technical Report 111, INRIA, October 1989.
- [Slack, 88] : N. Slack ; "Manufacturing Systems Flexibility - an Assessment Procedure". Computer Integrated Manufacturing Systems; vol 1, n° 1, (1988), pp.25 - 31.
- [Staroswiecki et al, 93] : M. Staroswiecki, M. Bayart, C. Feliot ; "Un Modèle pour la structuration des Modes de Marche et la Description Externe d'un Système Automatisé". Exposé le 24/09/93 à la réunion transversale "Supervision" du Groupe de Recherche Automatique.
- [Staroswiecki et al, 95] : M. Staroswiecki, M. Bayart ; "Modes d'utilisation et modes de marche des instruments intelligents". Journées d'étude SAPID : Conception de Systèmes Automatisés de Production à Intelligence distribuée. Paris 30-31 mai 1995.
- [Tawegoum et al, 94] : R. Tawegoum, E. Castelain et J.C. Gentina ; "Dynamic operation control in flexible manufacturing systems (FMS)". Intelligent Manufacturing Systems, june 1994, (IMS'94), IFAC. Edited by Peter Kopacek.
- [Tawegoum, 95] : R. Tawegoum ; "Contrôle temps réel du déroulement des opérations dans les systèmes de production flexibles". Thèse de Doctorat de l'USTL, avril, 1995.
- [Toguyeni et al, 90] : A.K.A. Toguyeni, E. Craye, J.C. Gentina ; "A method of temporal analysis to perform on line diagnosis in the context of Flexible Manufacturing System". Proceedings of IECON'90, Vol. 1, pp. 445-450, Pacific Grove California, november 1990.

## Références bibliographiques

- [Toguyeni et al, 91] : A.K.A. Toguyeni, E. Castelain, E.Craye ; " From the treatment of failure to the management of working degraded modes", Proceedings of IMACS'91, May 91, pp 595-601, Lille-France
- [Toguyeni, 92] : A.K.A. Toguyeni ; "Surveillance et diagnostic en ligne dans les ateliers flexible de l'industrie manufacturière". Thèse de doctorat de l'U.S.T.L, 1992.
- [Verdin, 91] : L. Verdin et "l'équipe technique du CCGA" ; "De la spécification à l'exploitation : "Le cycle de vie" des SAP". Génie Automatique et Production Industrielle, Journées de synthèse, ISMCM, Saint-Ouen, 12 et 13 mars 1991.
- [Vidril, 95] : M. Vidril ; "Pilotage hiérarchisé et réparti des système flexibles de production dans l'industrie manufacturière". Thèse de Doctorat de l'Université de Lille I, juillet 1995.
- [Villemeur, 88] : A. Villemeur ; "Sûreté de fonctionnement des systèmes industriels". Ed. Eyrolles, Col. EDF/DER, Vol. 67, 1988.
- [Woodward, 65] : J. Woodward ; "Industrial Organization: Theory and Practice". University Press, London (UK), 1965.
- [Young, 91] : R. M. Young (UK Medical Research Council Applied Psychologie Unit), "Evaluating Design Using Knowledge of Purpose and Knowledge of Structure", IEEE, APRIL 1991, pages 33-40.

# **TABLE DES MATIÈRES**

<b>Avant-Propos</b> .....	<b>3</b>
<b>Sommaire</b> .....	<b>7</b>
<b>Introduction générale</b> .....	<b>9</b>
<b>Partie A : Problématique de la Supervision et de la Gestion des Modes</b> .....	<b>15</b>
Introduction .....	17
Chapitre I : Les Systèmes de production : hypothèses de travail .....	18
I.1. Définition et typologie des systèmes de production .....	18
I.2. La Flexibilité .....	19
I.2.1. Définition .....	19
I.2.2. Typologie .....	19
I.2.3. Mesure de la flexibilité .....	20
I.2.4. Flexibilité et Productivité : le dilemme. ....	21
I.3. Les Systèmes Flexibles de Production Manufacturière (SFPM).....	21
Chapitre II : L'intégration des SFPM automatisés.....	24
II.1. L'environnement du Contrôle/commande des SFPM .....	24
II.2. L'intégration .....	25
Chapitre III : Spécification fonctionnelle de la conduite d'un SFPM [Kernad et al, 93b] .....	29
III.1. Le formalisme SADT pour la spécification fonctionnelle.....	29
III.2. L'environnement de la conduite .....	30
III.2.1. La planification de la production .....	30
III.2.2. La fabrication .....	31
III.3. La conduite du système .....	31
III.3.1. Le pilotage du système.....	32
III.3.2. La supervision .....	35
Chapitre IV : La gestion des modes dans la littérature.....	41
IV.1. Le GEMMA.....	41
IV.1.1. Limites du Gemma et solutions envisagées .....	45
IV.2. Approche du Laboratoire d'Automatique Industrielle de l'I.S.M.C.M. [Anakök, 87] .....	47
IV.2.1. Limites .....	51
IV.3. L'approche du Laboratoire de Génie Industriel et Logiciel de l'Université de Valenciennes et du Hainaut-Cambrésis : le MESAP [Parayre, 92] .....	52
IV.3.1. Limites .....	53
IV.4. Approche du Laboratoire d'Automatique de Nantes [Biland, 94] .....	53
IV.5. L'approche du LAIL .....	53
IV.5.1. L'équipe ASPIC .....	53
IV.5.2. L'équipe PFM.....	54
IV.5.3. La gestion des modes de marche dans CASPAIM [Bois, 91] .....	55
Conclusion.....	59

<b>Partie B : Modélisation des SFPM et gestion des modes</b> .....	<b>61</b>
Introduction .....	63
Chapitre V : Présentation de CASPAIM [Bourey, 93], [Amar, 94] et [Craye, 94] .....	64
V.1. Phases de spécification .....	64
V.1.1. Spécification de la partie logique : les objectifs de production .....	64
V.1.2. Spécification de la partie physique : les moyens de production .....	64
V.1.3. Spécification de la partie décisionnelle : les règles stratégiques.....	66
V.1.4. Spécification de la supervision : la surveillance et la gestion des modes .....	66
V.2. Phase d'analyse et de conception .....	66
V.2.1. Génération automatique des Gammes Opératoires [Amar, 94] .....	67
V.2.2. Construction des graphes de contrôle (ou de commande) des ressources [Huvénait, 94] .....	68
V.3. Phase de validation .....	69
V.3.1. Validation statique des gammes .....	69
V.3.2. Validation dynamique du système .....	70
V.4. Phase d'implantation.....	70
Chapitre VI : La supervision dans CASPAIM .....	73
VI.1. La surveillance.....	73
VI.1.1. La surveillance réactive [Elkhatabi, 93] .....	73
VI.1.2. La surveillance en ligne [Toguyeni, 92] .....	75
VI.2. La gestion des modes de marche et d'arrêt [Bois, 91] .....	79
VI.2.1. Les contraintes entre machines .....	79
VI.2.2. Regroupement des machines .....	84
VI.2.3. Structuration et fonctionnement du gestionnaire de modes de marches [Bois, 91] .....	85
Chapitre VII : Exigence d'une approche fonctionnelle pour la gestion des modes .....	89
VII.1. Limites de l'approche structurelle.....	89
VII.2. La modélisation fonctionnelle.....	93
VII.3. Modèle fonctionnel et modèle structurel.....	93
VII.4. Conclusion.....	94
Chapitre VIII : Représentation fonctionnelle pour la gestion des modes d'un SFPM.....	96
VIII.1. Les opérations dans un SFPM .....	97
VIII.2. Définition et modélisation des contraintes entre opérations .....	97
VIII.3. Le Regroupement des opérations.....	100
VIII.3.1. Règles de regroupement.....	100
VIII.3.2. Exemples de mise en œuvre de la modélisation fonctionnelle .....	101
VIII.3.3. Intégration de la dimension matérielle du SFPM : les ressources physiques.....	110
Conclusion [Kermad et al, 96] .....	112

<b>Partie C : Approche Sychrone de la gestion des modes.....</b>	<b>115</b>
Introduction .....	117
Chapitre IX : Les langages synchrones graphiques.....	120
IX.1. Présentation de StateCharts .....	120
IX.1.1. La hiérarchie .....	121
IX.1.2. Le parallélisme.....	121
IX.1.3. La communication .....	122
IX.1.4. L'entrée dans un état .....	123
IX.1.5. Les inconvénients de StateCharts .....	123
IX.2. Présentation de Argos.....	124
IX.2.1. Les Automates .....	124
IX.2.2. La hiérarchie .....	125
IX.2.3. Le parallélisme.....	126
IX.2.4. L'événement local .....	127
IX.2.5. La terminaison .....	128
IX.2.6. L'environnement ARGONAUTE [Maraninchi, 89] .....	131
Chapitre X : Application de la représentation COD à l'exploitation et à la supervision des SFPM.....	132
X.1. Décomposition arborescente d'une gamme logique.....	132
X.2. La disponibilité du SFPM et la représentation COD. ....	137
X.2.1. Disponibilité d'une opération effective .....	138
X.2.2. Disponibilité d'un transfert.....	139
X.2.3. Disponibilité d'une opération virtuelle .....	139
X.2.4. Disponibilité d'une opération dans une gamme logique .....	140
X.2.5. Disponibilité d'une gamme logique.....	140
Chapitre XI : Formalisation des flux d'information dans un SFPM .....	142
XI.1. Spécification de la disponibilité du SFPM : un modèle générique du flux ascendant.....	142
XI.1.1. Spécification de la disponibilité statique. ....	144
XI.1.2. Spécification de la disponibilité dynamique.....	148
XI.2. Spécification de la commande du SFPM : des modèles génériques du flux descendant.....	152
XI.3. Intégration des flux d'information ascendant et descendant .....	157
Conclusion .....	160
<b>Conclusion générale .....</b>	<b>163</b>
<b>Annexe : Présentation de CASPAIM.....</b>	<b>167</b>
<b>Présentation de CASPAIM [Bourey, 93], [Amar, 94] et [Craye, 94] .....</b>	<b>169</b>
<b>Références bibliographiques .....</b>	<b>187</b>
<b>Table des matières .....</b>	<b>199</b>
<b>Table des figures .....</b>	<b>205</b>

## **TABLE DES FIGURES**

**Partie A**

Figure I.1 : Productivité , Flexibilité et Coût ..... 21

Figure II.1 : Le modèle canonique OID [Le Moigne, 1990]..... 26

Figure II.2 : Les fonctions de commande d'un SFPM..... 27

Figure III.1 : Actigrammes, définitions et décomposition ..... 29

Figure III.2 : L'environnement de la conduite..... 30

Figure III.3 : l'activité "Conduire" ..... 32

Figure III.4 : L'activité "Piloter" ..... 33

Figure III.5 : L'activité "Superviser" ..... 37

Figure III.6 : l'activité "Surveiller" ..... 38

Figure III.7 : L'activité "Gérer les Modes" ..... 39

Figure IV.1 : Perçage semi-automatique [ADEPA, 92]..... 43

Figure IV.2 : Représentation des états du système de perçage ..... 44

Figure IV.3 : Structuration de la commande du point de vue des modes de marche  
[ADEPA, 92]..... 45

Figure IV.4 : Les machines M1-M2..... 47

Figure IV.5 : Hiérarchie Structurale entre les familles de Modes ..... 49

Figure IV.6 : Une Organisation du centre d'usinage : perçage simple..... 50

Figure IV.7 : Exemple de Familles de Modes ..... 51

Figure IV.8 : graphe de comportement d'une machine effective complexe ..... 57

Figure IV.9 : graphe de comportement d'une machine virtuelle ..... 58

**Partie B**

Figure V.1 : démarche de conception dans CASPAIM ..... 65

Figure V.2 : description d'une gamme opératoire ..... 67

Figure V.3 : Structure des gammes opératoires étendues ..... 67

Figure V.4 : Hiérarchisation de la commande ..... 68

Figure V.5 : Modèle d'échange au sein des gammes étendues ..... 68

Figure V.6 : Architecture de contrôle d'un SFPM ..... 71

Figure VI.1 : Démarche de modélisation des CFL [Elkhatabi, 93] ..... 73

Figure VI.2 : Organisation de la surveillance de bas niveau d'un CFL ..... 73

Figure VI.3 : Modèle associé à une opération [Toguyeni, 90] ..... 74

Figure VI.4 : Filtrage à partir de la modélisation temporelle [Toguyeni, 92]..... 75

Figure VI.5 : Conception du Modèle Fonctionnel pour le diagnostic [Toguyeni, 92] ..... 76

Figure VI.6 : Graphe Fonctionnel (GF) d'un centre d'usinage [Toguyeni, 92] ..... 77

Figure VI.7 : Exemple de cellule flexible ..... 78

Figure VI.8 : Contrainte de Coopération ..... 79

Figure VI.8.b : Description RdP de la contrainte de coopération ..... 79

## Table des figures

Figure VI.9 : Contrainte de Coopération Partagée.....	80
Figure VI.9b : Description RdP de la contrainte de coopération partagée .....	80
Figure VI.10 : contrainte d'exclusion .....	81
Figure VI.10.b : Description RdP de la contrainte d'exclusion.....	81
Figure VI.11 : Contrainte de Procédure .....	81
Figure VI.11.b : Description RdP de la contrainte de procédure .....	82
Figure VI.12 : Contrainte d'Observabilité.....	82
Figure VI.13 : exemple de Table .....	83
Figure II.14 : Application des règles de regroupement à l'exemple.....	84
Figure VI.15 : Fonctionnement du gestionnaire d'après[Bois, 91] .....	85
Figure VI.16 : Structuration du Niveau Hiérarchique et du gestionnaire des modes de marche d'après [Bois, 91].....	86
Figure VII.1 : Exemple de cellule flexible.....	88
Figure VII.2 : Tableau initial des contraintes .....	89
Figure VII.3 : Premier niveau d'agrégation.....	89
Figure VII.4 : Deuxième tableau des contraintes.....	90
Figure VII.5 : Deuxième niveau d'agrégation .....	90
Figure VII.6 : Troisième niveau d'agrégation .....	90
Figure VII.7.a : Représentation arborescente.....	91
Figure VII.7.b : Deuxième représentation arborescente .....	91
Figure VIII.1 : Contrainte de Chargement et de Déchargement .....	97
Figure VIII.2 : Contrainte de Chargement et de Déchargement Partagée .....	97
Figure VIII.3 : Contrainte de Procédure .....	98
Figure VIII.4 : Contrainte de Procédure Partagée.....	98
Figure VIII.5 : contrainte d'exclusion .....	99
Figure VIII.6 : Exemple d'unité de production flexible .....	101
Figure VIII.7 : Les relations d'accessibilités .....	101
Figure VIII.8 : Tableau initial des contraintes .....	102
Figure VIII.9 : Tableau des contraintes .....	103
Figure VIII.10 : Représentation COD des opérations et des transferts.....	104
Figure VIII.11 : Arbre COD complet.....	105
Figure VIII.12 : Aspect générique de la construction de l'arbre COD.....	105
Figure VIII.13 : La cellule flexible de l'EC Lille en 1995 .....	106
Figure VIII.14 : Spécification fonctionnelle de la cellule de l'EC Lille .....	108
Figure VIII.15 : Représentation COD du tournage.....	108
Figure VIII.16 : L'arbre COD de la cellule flexible de l'EC Lille.....	109
Figure VIII.17 : Intégration des ressources à la représentation COD .....	110
Figure VIII.18 : Décomposition d'un transfert.....	110
Figure VIII.19 : L'allocateur des ressources .....	112

## Table des figures

### Partie C

Figure C: Configuration et allocation des ressources .....	118
Figure IX.1 : la hiérarchie .....	121
Figure IX.2 : le parallélisme .....	122
Figure IX.3 : l'entrée par défaut .....	123
Figure IX.4 : exemple d'Automate non-déterministe.....	125
Figure IX.5: exemple d'Automate déterministe .....	125
Figure IX.6 : la hiérarchie .....	126
Figure IX.7 : la décomposition parallèle.....	127
Figure IX.8: Exemple de processus parallèles .....	127
Figure IX.9 : opérateur d'événement local .....	128
Figure IX.10: Exemple d'application .....	129
Figure IX.11 : le mécanisme d'auto-interruption .....	130
Figure IX.12 : la terminaison normale .....	130
Figure IX.13: la terminaison anormale .....	131
Figure IX.14 : l'interruption .....	131
Figure X.1 : Décomposition d'une gamme logique flexible en gammes logiques linéaires .....	134
Figure X.2 : Décomposition d'une gamme logique.....	135
Figure X.3 : Décomposition d'une opération de Gamme Logique .....	135
Figure X.4 : Décomposition d'une Opération Virtuelle de Gamme Opératoire.....	136
Figure X.5 : Equivalence entre COD et GOOpV.....	136
Figure X.6 : Représentation arborescente d'une Gamme Logique.....	137
Figure X.8 : Disponibilité d'une opération effective .....	140
Figure X.9 : Disponibilité d'un Transfert .....	140
Figure X.10 : Automate d'une opération virtuelle dans i-une gamme opératoire .....	141
Figure X.11 : Disponibilité d'une opération de Gamme Logique .....	141
Figure X.12 : Disponibilité d'une gamme logique linéaire .....	142
Figure XI.1 : Représentation de la disponibilité d'un élément de SFPM .....	143
Figure XI.2 : Représentation de la disponibilité d'un élément de SFPM .....	144
Figure XI.3 : Un exemple d'illustration .....	146
Figure XI.4 : Représentation COD de l'exemple d'illustration .....	146
Figure XI.5 : Représentation hiérarchique de la disponibilité de l'exemple d'illustration .....	147
Figure XI.6 : Disponibilité du chargement C1 .....	148
Figure XI.7 : Représentation de la disponibilité dynamique d'un élément dans l'arborescence d'une gamme logique.....	149
Figure XI.8 : Gamme logique GL1 du produit P1 .....	150
Figure XI.9 : Représentation arborescente de la gamme logique GL1 .....	151
Figure XI.10 : Disponibilité de l'opération virtuelle de gamme opératoire OpV11 .....	152
Figure XI.11 : Disponibilité de l'opération de gamme logique Op1 et de la gamme logique	

## Table des figures

GL1 .....	153
Figure XI.12 : Flux descendant dans le cas du "ET" .....	154
Figure XI.13 : Flux descendant dans le cas du "OU" .....	155
Figure XI.14 : Commande de la marche et de l'arrêt de la gamme logique GL1 .....	156
Figure XI.15 : Commande de la marche et de l'arrêt de l'opération virtuelle de gamme opérateur OpV11 .....	157
Figure XI.16 : Commande de la marche et de l'arrêt du transfert E->M1 .....	158
Figure XI.17 : La gestion des modes d'un SFPM .....	159
Figure XI.18 : Structure de la gestion des modes de production .....	160

## Annexe

Figure A : démarche de conception dans CASPAIM .....	170
Figure A1 : description d'une gamme logique élémentaire .....	170
Figure A2 : description d'un processus avec alternative .....	171
Figure A3 : description d'un processus n-aire .....	171
Figure A4 : Bibliothèque fonctionnelle [Amar, 94] .....	173
Figure A5 : Exemple d'atelier .....	174
Figure A6 : Décomposition structuro-fonctionnelle, lieux caractéristiques et relations d'accessibilité .....	175
Figure A7 : description d'une gamme opératoire .....	176
Figure A8: démarche de génération .....	177
Figure A9 : choix des lieux de fabrication .....	177
Figure A10: flexibilité de choix des machines.....	178
Figure A11: dissociation des transferts et traitements .....	178
Figure A12: Structure des gammes opératoires étendues .....	179
Figure A13 : Hiérarchisation de la commande .....	179
Figure A14: Protocoles de communication au sein des gammes étendues .....	180
Figure A15: Extension de la démarche de conception [Craye, 94] .....	180
Figure A16: Modèle des ressources simples.....	181
Figure A17 : Graphes de commandes génériques du convoyeur [Huvenoit, 94] .....	182
Figure A18 : Modèle comportemental du convoyeur [Huvenoit, 94].....	183
Figure A19 : Architecture de contrôle d'un SFPM. ....	184
Figure A20 : Association HOOD et RdP .....	186
Figure A21: Association HOOD et StateCharts .....	186



## Résumé :

Nous avons montré dans ce mémoire qu'une approche des systèmes de production par les fonctions et les opérations permet d'aller beaucoup plus loin dans la spécification des contraintes de fonctionnement et de gestion des configurations et des modes de fonctionnement. Nous avons présenté la problématique de la supervision et de la gestion des modes en étendant les concepts du GEMMA. En effet il ne s'agit plus de gérer la marche et l'arrêt d'un système automatisé de production unitaire mais de gérer les configurations d'un Système Flexible de Production composé et complexe.

La démarche que nous proposons, dans un premier temps, permet de construire un modèle du système de production, par une analyse ascendante des contraintes et des liens de fonctionnement entre opérations. Cette technique de modélisation aboutit, de manière systématique, à un modèle arborescent fonctionnel du système de production. Ce modèle met en évidence les différents niveaux de commande du Système Flexible de Production Manufacturière et fait ressortir les principes d'organisation et de conduite des installations en spécifiant leurs liens avec les fonctions. L'originalité de cette démarche réside dans la possibilité d'intégrer tous les composants d'une unité de production. Elle permet aussi de tenir compte d'une manière très simple de l'ajout ou de la suppression d'un composant.

Dans une seconde démarche, nous avons montré comment l'exploitation de la modélisation permet de générer une spécification de la supervision du SFPM. Cette phase de traduction est basée sur la transcription directe des modèles du SFPM dans un langage synchrone graphique : Argos. La traduction de ces arborescences sous forme de programmes synchrones permet la mise à jour réactive de l'état et de la disponibilité des composants du système de production et la transmission des ordres de reconfiguration de façon originale.

## Mots-Clés :

SYSTÈME DE PRODUCTION FLEXIBLE, SUPERVISION, GESTION DES MODES, MODÉLISATION FONCTIONNELLE, SPÉCIFICATION, RECONFIGURATION