50376
1995
157

Jan 2010 3820

N° d'ordre : 1555

THESE

BIBLIOTHÈQUE INTERUNIVERSITAIRE SECTION DE SCIENCES LILLE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE
en
PRODUCTIQUE, AUTOMATIQUE ET INFORMATIQUE INDUSTRIELLE

par

Marc VIDRIL
Ingénieur de l'Ecole Centrale de Lille

PILOTAGE HIERARCHISE ET REPARTI DES SYSTEMES FLEXIBLES DE PRODUCTION DANS L'INDUSTRIE MANUFACTURIERE

Soutenue le 7 juillet 1995 devant la Commission d'Examen composée de :

MM.	J.M. HENRIOUD	Rapporteur
	J.J. LESAGE	Rapporteur
	J.C. GENTINA	Examinateur, Directeur de thèse
	J.P. BOUREY	Examinateur
	J.P. CASSAR	Examinateur
	E. CRAYE	Examinateur
	X. DATIN	Invité

A mes parents, mes frères, ma soeur,
à Marie-Claude,
à ma famille,
et à mes amis

avec toute mon affection.

Table des matières

Avant propos XIII

Introduction 1

PARTIE 1 : CONTEXTE GENERAL DU TRAVAIL 7

Chapitre 1 : Les systèmes flexibles de production manufacturière 9

Introduction 9
1.1 Systèmes à événements discrets 9
1.1.1 Définition 9
1.1.2 Classification des systèmes de production selon leur degré de discrétisation 10
1.2 Assemblage vs usinage 11
1.2.1 Définitions 11
1.2.2 Principales différences 11
1.3 Typologie des systèmes d'assemblage 12
1.3.1 Classement par mode opératoire et type de transfert 13
1.3.2 Regroupement structurel 15
1.3.3 Classement par type d'organisation 15
1.4 Les flexibilités physiques 17
1.4.1 Flexibilités liées aux systèmes de transfert 17
1.4.2 Flexibilités liées aux moyens de production 18
1.4.3 Flexibilités liées aux produits 18

Chapitre 2 : Principes de commande et de modélisation des SFPM 19

Introduction....19
2.1 L'automatisation des SFPM....19
2.1.1 Structure des systèmes automatiques....19
2.1.2 Problématique de la commande des SFPM....20
2.2 Hiérarchisation des décisions....21
2.2.1 Principe....22
2.2.2 La structuration verticale du CIM....22
2.2.3 Notion de centres de décision....25
2.2.4 Notion d'autonomie décisionnelle....27
2.2.5 Application de la hiérarchisation dans ce travail....28
2.3 La modélisation des SFPM....28
2.3.1 Objectif de la modélisation....28
2.3.2 Principe de modélisation d'un SED....29
2.3.3 Les entités caractéristiques d'un SFPM....29
2.3.4 Les outils de modélisation....30
2.3.5 Avantages et inconvénients des RdP....31

Chapitre 3 : La supervision dans SCOPES 33

Introduction....33
3.1 La supervision : pourquoi et comment ?....33
3.1.1 La supervision, réponse à un besoin....33
3.1.2 Vers des progiciels de supervision spécialisés....34
3.2 La supervision dans le projet SCOPES....35
3.2.1 Fonctions de supervision SCOPES....35
3.2.2 Mise en oeuvre modulaire....37
3.2.3 Modes de fonctionnement supervisé et simulé....38
3.3 Positionnement et intérêt de l'étude....38
3.3.1 Approches de structuration traditionnelles....39
3.3.2 Intérêt d'un pilotage hiérarchisé au niveau cellule....42
3.3.3 Intérêt du mode simulation....44

Chapitre 4 : Un exemple d'atelier flexible :le site productique de la DRD 47

Introduction....47
4.1 Architecture physique....47
4.1.1 Magasin central....47
4.1.2 Système de convoyage....48
4.1.3 Cellule d'assemblage....49
4.2 Architecture informatique....49
4.3 Produits assemblés....50
4.3.1 Schéma des produits....51
4.3.2 Séquences d'assemblage....51
4.4 Fonctionnement général....52
4.5 Intérêt de l'exemple....53

PARTIE 2 : ANALYSE ET MODELISATIONS 55

Chapitre 5 : Analyse des éléments d'un SFPM 57

Introduction 57
5.1 Les produits 57
5.1.1 Définitions 57
5.1.2 Opération caractéristique 58
5.1.3 Etat d'un produit 59
5.1.4 Notion de gamme 59
5.1.5 Représentation des gammes 60
5.1.6 Regroupement en famille et notion de variante 62
5.2 Les lieux 63
5.2.1 Décomposition structurelle 63
5.2.2 Relations de composition 65
5.2.3 Relations d'accessibilité 67
5.3 Les moyens 67
5.3.1 Décomposition structurelle 69
5.3.2 Décomposition fonctionnelle 69
5.3.3 Comportement d'une ressource 70
5.4 Les opérations 71
5.4.1 Opérations réelles et opérations logiques 71
5.4.2 Modalité opératoire et affectation 72
5.4.3 Application au site DRD 72

Chapitre 6 : Processus de fabrication d'un produit 75

Introduction 75
6.1 Elaboration du processus de fabrication 76
6.2 La décomposition arborescente des produits 78
6.3 Organisation du processus de fabrication 80
6.3.1 Processus de fabrication et gammes 81
6.3.2 Opérations et phases 83
6.3.3 Mouvements et tâches 84

Chapitre 7 : Organisation hiérarchique du système de commande 85

Introduction 85
7.1 Hiérarchie externe et hiérarchie interne 85
7.1.1 Centres décisionnels et hiérarchie externe 85
7.1.2 Interface externe d'un centre 86
7.1.3 Structure d'un centre décisionnel et hiérarchie interne 87
7.2 Fonctionnement de la hiérarchie 90
7.2.1 Ordonnancement / gestion des stocks 90
7.2.2 Pilotage 92
7.2.3 Commande / communication 94
7.2.4 Surveillance / analyse / maintenance 95
7.2.5 Conclusion 96
7.3 Implantation physique du système de commande 97

Chapitre 8 : Modélisation en vue du pilotage 99

Introduction 99
8.1 Modèle des gammes opératoires 99
8.1.1 Inconvénients de la représentation linéaire 99
8.1.2 Modélisation retenue 100
8.1.3 Cas de synthèse des gammes relatives à des articles différents 104
8.1.4 Application à l'exemple du site TE 106
8.2 Modèle de l'installation 107
8.2.1 Les éléments de l'installation 107
8.2.2 Les relations d'accessibilité 107
8.3 Affectations 109

Chapitre 9 : Modélisation en vue de la simulation 111

Introduction 111
9.1 Les étapes de la simulation 111
9.2 La démarche de modélisation proposée 112
9.2.1 Principe 112
9.2.2 Etapes de la modélisation. 113
9.3 Etape 1 : construction des graphes 115
9.3.1 Les réseaux produits (RP) 115
9.3.2 Les réseaux transitiques (RT) 116
9.3.3 Les réseaux comportementaux (RC) 118
9.4 Etape 2 : connexion des graphes 119
9.5 Utilisation du modèle 122
9.5.1 Validation analytique 122
9.5.2 Simulation 123

PARTIE 3 : UNE SOLUTION AU PILOTAGE DES FLUX 125

Chapitre 10 : Principes généraux de pilotage 127

Introduction 127
10.1 Contraintes sur le pilotage 127
10.1.1 Contraintes temporelles 127
10.1.2 Contraintes utilisateurs 128
10.2 Principes retenus 128
10.2.1 Production par campagne 128
10.2.2 Mise en oeuvre modulaire 129
10.2.3 Possibilité de prise de décision centralisée 130
10.3 Scénario général de pilotage 130
10.4 Les deux types de coordination 131

Chapitre 11 : Coordination de la production 133

Introduction 133
11.1 Principe 133
11.2 Les différents types de "requêtes" 135

11.2.1 L'ordre de fabrication (OF) 135
11.2.2 Le bon de fabrication global 136
11.2.3 Le bon de fabrication local 137
11.3 Le processus d'instanciation 138
11.3.1 Etape n°1 : instanciation d'un ordre de fabrication 139
11.3.2 Etape n°2 : instanciation des bons de fabrication globaux 139
11.3.3 Etape n°3 : instanciation des bons de fabrication locaux 140
11.4 Le suivi de la production 142
11.4.1 Relations entre un OF et les bons globaux associés 142
11.4.2 Relations inter bons globaux 143
11.4.3 Relations entre un bon global et les bons locaux associés 144
11.5 Fin de campagne et désinstanciation 146
11.6 Signalisation et coordination de la production 147
11.6.1 Principe de la signalisation 147
11.6.2 Contrôle des changements de campagne 147
11.7 Fonctionnalités locales et globales 150

Chapitre 12 : Utilisation des graphes d'influence 151

Introduction 151
12.1 Présentation générale 151
12.2 Illustration de la méthode : exemple 1 153
12.2.1 Choix du point d'application de la consigne 153
12.2.2 Définition des zones 154
12.2.3 Construction des arcs TT dans les zones 155
12.2.4 Définition des flux de zone 156
12.2.5 Ajout des influences inter-zones 157
12.3 Illustration de la méthode : exemple 2 160
12.4 Résumé de la méthode 163

Chapitre 13 : Coordination des transferts 167

Introduction 167
13.1 Principe 167
13.1.1 Spécificité des phases de transfert 167
13.1.2 Notions de routage et de trafic 169
13.2 Les règles d'aiguillage 170
13.2.1 Présentation 170
13.2.2 Algorithme local de gestion du trafic 173
13.2.3 Emploi des règles d'aiguillage 174
13.2.4 Paramétrage statique ou dynamique 174
13.2.5 Initialisation des règles d'aiguillage 175
13.2.7 Application à l'exemple du site DRD 176
13.3 Les règles de destination 178
13.3.1 Un problème de niveau global 178
13.3.2 Proposition de mise en oeuvre 179
13.3.3 Application à l'exemple du site DRD 180
13.4 Signalisation et coordination des transferts 182
13.4.1 Réservation lors des changements de campagne 182
13.4.2 Réaction aux pannes 182

13.5 Fonctionnalités locales et globales 184

Chapitre 14 : Calcul des règles d'aiguillage **185**

Introduction 185
14.1 Principe général 185
14.1.1 Prise de décision dans un problème de choix 186
14.1.2 Simplification pour le problème du choix de trajectoire 187
14.2 Formalisation et notations 188
14.2.1 Notion de position 188
14.2.2 Indexage utilisé 189
14.2.3 Matrices IJJ et IJI 189
14.3 Les fonctions d'évaluation 190
14.3.1 Fonctions d'évaluation sur la première partie de trajectoire 190
14.3.2 Fonctions d'évaluation sur la seconde partie de la trajectoire 191
14.4 Les fonctions de stratégies 193
14.5 Mise en oeuvre par les règles d'aiguillage 194
14.5.1 Cas général 194
14.5.2 Cas pratiques 195
14.5.3 Rapports avec les règles de destination 198
14.5.4 Application à l'exemple du site DRD 199

Conclusion **201**

Bibliographie **207**

ANNEXES **215**

Annexe A : La modélisation par Réseaux de Petri **215**

A.1 RdP simples 215
A.2 Pliage et RdP de haut niveau 218
A.3 Les RdP à Objets 221

Annexe B : Exemple de calcul des fonctions d'évaluation **225**

B.1 Numérotations 225
B.2 Matrices initiales 226
B.3 Calcul de OPTA et OPTS 228

Annexe C : Exemples d'utilisation des graphes d'influence **231**

C.1 Exemple 1 : boucle de retouche avant 231
C.2 Exemple 2 : boucle de retouche arrière 233
C.3 Exemple 3 : séquence linéaire d'opérations 236
C.4 Exemple 4 : synchronisation de gammes 238
C.5 Exemple 5 : application au produit P du site 241

Annexe D : Architecture logicielle du superviseur **245**

D.1 Intégration dans MONITOR 77/2. 245
D.2 Les tâches 247
D.3 Flux d'informations entre tâches. 249

Annexe E : Tâche de simulation (XSIM) **253**

E.1 Objectif et philosophie adoptée 253
E.2 Principes de simulation 254
E.3 Utilisation de la tâche simulation 257

Annexe F : Tâche de pilotage (XFLOW) **261**

F.1 Principe de la tâche 261
F.2 Utilisation de la tâche pilotage 265

Avant propos

Les travaux présentés dans ce mémoire ont été réalisés dans le cadre d'une Convention Industrielle de Formation par la Recherche (contrat CIFRE n° 91470) accordée par l'Association Nationale de la Recherche Technique (ANRT). Cette convention associe :

- la société Schneider Electric par l'intermédiaire de sa Direction de Recherche Automatique et Productique dirigée par Monsieur M. Allain ;
- l'Ecole Centrale de Lille et plus particulièrement le Laboratoire d'Automatique et d'Informatique Industrielle de Lille (LAIL) dirigé par Monsieur J.C. Gentina.

Elle constitue le prolongement des contacts engagés de longue date entre ces deux partenaires, témoignant ainsi d'une réelle convergence de préoccupations, tant sur le plan scientifique, que sur le plan de la collaboration enseignement-industrie.

Qu'il me soit ici permis de remercier toutes les personnes qui, à titre divers, ont permis que ce travail voit le jour.

Je remercie tout particulièrement Monsieur J.C. Gentina, Directeur de l'Ecole Centrale de Lille, et Monsieur X. Datin, responsable du Laboratoire Productique de Schneider Electric, pour m'avoir guidé dans mes travaux de recherche. Leur confiance, leurs conseils et leur enthousiasme m'ont été d'une aide précieuse durant ces trois années.

Je suis très reconnaissant à Monsieur J.J. Lesage, Professeur à l'Ecole Normale Supérieure de Cachan et membre du Laboratoire Universitaire de Recherche en Production Automatisée (LURPA), et à Monsieur J.M. Henrioud, Professeur à l'Université de Franche-Comté et membre du Laboratoire

d'Automatique de Besançon (LAB), pour l'honneur qu'ils me font en acceptant d'examiner ce travail et d'en être les rapporteurs.

Je suis également flatté de la présence dans le jury de Monsieur J.P. Bourey, Professeur à l'Ecole Centrale de Lille, de Monsieur E. Craye, Maître de Conférence à l'Ecole Centrale de Lille, de Monsieur J.P. Cassar, Maître de Conférence à l'Université des Sciences et Technologies de Lille et les remercie vivement de l'intérêt qu'ils ont porté à cette thèse.

Mes remerciements vont également à toutes les personnes qui, au sein de la Direction des Recherches de Schneider Electric, m'ont encadré et conseillé au cours de ses trois années :

- Monsieur X. Datin, responsable du Laboratoire Productique (1992-1994) et actuellement Chef de Projet chez MAA, pour son dynamisme et ses précieux conseils pratiques qui se sont toujours révélés sources d'idées nouvelles ;
- Monsieur P. Charlès, responsable du Laboratoire Automatique et Productique (1987-1992), qui m'a recruté et accueilli dans son équipe ;
- Madame S. Dimster, membre du Laboratoire Automatique et Productique (1987-1992), pour m'avoir incité à m'engager dans ce travail et pour sa disponibilité à mon égard lors des premiers mois de recherche ;
- Monsieur M. Allain, responsable de la Direction de Recherche Automatique et Productique (1992-1995) pour avoir su faciliter le déroulement de mes travaux ;
- Monsieur D. Sornicle, Directeur de la Direction des Recherches de Schneider Electric, pour m'avoir accueilli dans son Centre de Recherche de Nanterre et pour les moyens qu'il a mis à ma disposition pour réaliser ce travail.

Cette recherche étant en grande partie le fruit d'un travail d'équipe, je ne saurais commencer ce mémoire sans remercier les personnes avec lesquelles j'ai été amené à collaborer, en particulier Messieurs S. Agneray, S. Piou et B. Lavergne, membres du Service Productique de Schneider Electric, ainsi que Monsieur E. Verdebout, membre de l'Institut de Microtechnique (IMT) de l'Ecole Polytechnique Fédérale de Lausanne (EPFL). Les discussions que nous avons eues sur le sujet ont toujours été passionnantes et sources d'idées nouvelles.

Enfin, je tiens à remercier vivement l'ensemble des membres de la Direction Automatique et Productique de Schneider Electric, ainsi que ceux du Laboratoire d'Automatique et d'Informatique Industrielle de Lille (LAIL) pour la sympathie qu'ils m'ont manifestée tout au long de ces trois années de recherche.

Introduction

Aujourd'hui, le contexte fortement concurrentiel et évolutif des marchés pousse les entreprises à améliorer sans cesse leur organisation afin de se doter d'une capacité de réaction toujours plus grande. En effet, dictées par le marché et non plus par la technologie, les temps de renouvellement des produits et de leurs moyens d'assemblage ont passé de 10 à 3 ans en quelques années. L'avantage compétitif réside désormais, pour une entreprise, dans sa capacité à offrir à ses clients de meilleurs produits plus vite que ses concurrents. Tout retard équivaut à une perte de parts de marché et donc à un allongement du délai de retour sur investissement risquant de dépasser la durée de vie du produit.

Afin de répondre à cet impératif de réactivité, optimiser les seuls moyens de fabrication ne suffit plus : à quoi bon un système fortement automatisé et très flexible si sa conception a demandé de longs mois voire des années d'efforts ? Il faut en fait repenser le système de conception lui-même. De cette réflexion est né le concept d'*ingénierie simultanée* (parfois appelé également *ingénierie parallèle* ou *ingénierie concourante*). L'objectif de ce concept est simple : il s'agit d'amener les ingénieurs développant des produits à travailler le plus tôt possible avec ceux développant les moyens de production et le système de contrôle/commande associé.

Les bénéfices potentiels de l'ingénierie simultanée sont nombreux : outre la réduction des délais permettant l'introduction rapide du produit sur le marché, elle permet, par une meilleure conception, l'abaissement des coûts du produit et l'amélioration de la qualité.

Amener les divers intervenants du processus de développement à collaborer plus efficacement n'est pas une mince affaire. Cela nécessite l'intégration des divers outils utilisés à chaque niveau, ainsi que la capitalisation des connaissances. Conscients des enjeux, de nombreux laboratoires, tant

universitaires qu'industriels, travaillent sur des méthodologies de conception intégrées permettant de traduire l'expérience acquise en un savoir formel afin de faciliter sa capitalisation et sa réutilisation.

Souvent spécialisé dans un domaine particulier, chacun s'est généralement focalisé sur l'une des facettes particulières du processus de développement. Ce processus peut grossièrement être décomposé en trois phases : conception des produits, choix des séquences d'assemblage et des moyens de production, enfin élaboration du système de contrôle / commande. Bien entendu, selon les concepts de l'ingénierie simultanée, ces trois phases sont imbriquées. Le processus de développement doit être conçu selon une approche modélisation / évaluation itérative. En ce sens les résultats d'une étape ne doivent jamais être considérés comme définitifs. D'où l'intérêt de méthodes systématiques permettant une concrétisation et une validation rapide des seules idées procurant une réelle valeur ajoutée. Par exemple, dans le domaine qui nous intéressera dans ce mémoire, une méthodologie de conception du système de commande doit fournir un cadre organisationnel précis permettant au concepteur de se focaliser principalement sur les stratégies d'exploitation à mettre en oeuvre.

Afin d'aboutir à un système d'ingénierie simultanée réellement complet, les diverses méthodologies, élaborées jusqu'ici séparément, doivent elles-mêmes être intégrées au sein d'un seul outil. De cet état de fait, il résulte que les solutions globales au problème de l'ingénierie simultanée ne peuvent provenir que de collaborations entre laboratoires aux compétences différentes. Le principal problème est alors d'établir une architecture cohérente permettant d'organiser les diverses phases de conception, tant des produits que des moyens de production et du système de commande, de façon non séquentielle. Les points essentiels concernent la structuration des connaissances, la définition des étapes où ces connaissances sont générées ainsi que les mécanismes de dialogue permettant leur remise en cause ou leur capitalisation. Une fois l'architecture établie, chaque équipe peut alors adapter les outils qu'elle maîtrise en vue de leur intégration au sein d'un système complet.

C'est dans le cadre d'une telle collaboration que s'intègrent les travaux présentés dans ce mémoire. Il s'agit du projet SCOPES que nous présentons ci-dessous.

Le projet SCOPES

SCOPES est l'acronyme de Systematic COncurrent design of Products, Equipments and control Systems[1]. Il s'agit d'un projet de recherche européen (programme ESPRIT[2] III). D'une durée de trois ans, il a débuté en juillet 1992 et regroupe six partenaires : deux industriels (Schneider Electric et Dassault Systèmes) coordonnant les travaux et quatre laboratoires universitaires

[1] Ingénierie simultanée et systématique des produits, des équipements et des systèmes de commande.

[2] European Strategic Project of Research on Information Technologies.

(Cranfield University[3], CRIF[4], EPFL[5] et IAT[6]) dont les domaines de compétences recouvrent l'ensemble des domaines de l'ingénierie simultanée.

L'objectif de ce projet est de définir et de valider les spécifications d'outils informatiques intégrés d'aide à la conception et à la fabrication de produits. Ces outils ont pour but explicite de permettre la mise en oeuvre des concepts de l'ingénierie simultanée. Ils se focalisent sur l'amélioration de la communication entre les divers acteurs intervenant le long du cycle de vie d'un produit.

Ces outils se répartissent en deux domaines : la conception du produit et de l'installation d'assemblage sur une station de CAO (partie "hors-ligne"), puis son contrôle (partie "en-ligne") à partir d'un superviseur coordonnant l'activité de contrôleurs locaux répartis (automates industriels). L'architecture fonctionnelle de la solution se décline sous la forme de modules (cf. figure 0.1). Le détail de cette architecture sort du cadre de cette introduction.

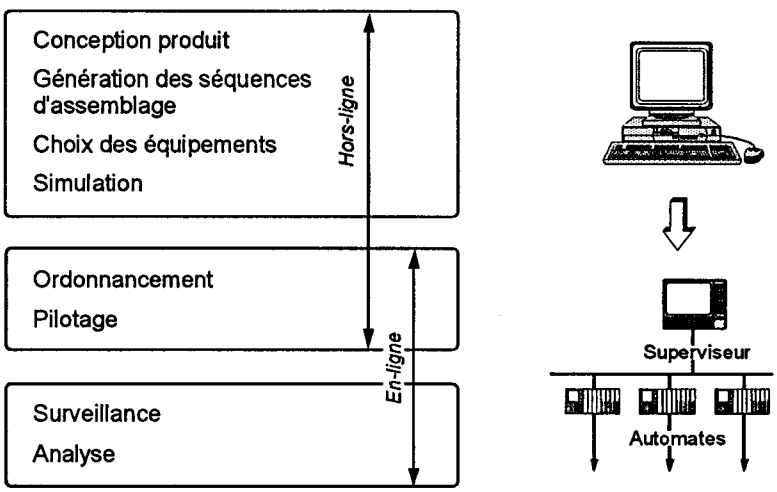


Figure 0.1 : les modules SCOPES.

La cohérence entre les deux domaines est assurée par la présence des modules communs de pilotage et d'ordonnancement qui ont un fonctionnement identique dans la station de conception et dans l'architecture de supervision : les stratégies mises au point en simulation sont immédiatement applicables sur l'installation réelle par simple transfert de données.

Le lecteur intéressé par les divers modules et leur intégration dans l'architecture multi-niveaux proposée par SCOPES pourra se référer aux divers "deliverables" (rapports remis aux experts de la Communauté Européenne)

---

[3] CIM Institute, Cranfield Institute of Technology, Grande-Bretagne.

[4] CRIF, Centre de recherche scientifique et technique de l'industrie de fabrications métalliques, Bruxelles, Belgique.

[5] IMT-EPFL, Institut de Microtechnique de l'Ecole polytechnique Fédérale de Lausanne, Suisse.

[6] IAT, Institut für Arbeitswissenschaft und Technologie-management, Stuttgart Universität, Allemagne.

rédigés dans le cadre de ce projet. La liste exhaustive de ces "deliverables" peut être trouvée dans [SCOPES Technical Annex, 1994]. Une présentation synthétique du projet peut être trouvée dans les actes des séminaires organisés à l'issue du projet sur l'apport de SCOPES à l'ingénierie simultanée [SCOPES Workshop 1 à 4, 1995].

Pilotage et ingénierie simultanée

Le travail présenté dans ce mémoire concerne le domaine de la conception du système de contrôle/commande des systèmes de production. Dans ce cadre, nous avons collaboré au projet SCOPES pour l'étude du module de pilotage et son implantation au sein du système de supervision (partie "en-ligne").

Ainsi que nous l'avons déjà mentionné, l'objectif principal d'une méthodologie de conception du système de commande dans le cadre de l'ingénierie simultanée est de fournir une organisation et des outils génériques. Le but n'est pas de contraindre le concepteur mais de lui fournir un cadre formel lui permettant d'oublier les contraintes de programmation pour se concentrer sur les points essentiels, à savoir la définition de la politique de gestion de son installation. La solution que nous proposons dans ce mémoire a été élaborée dans ce sens.

Le parti pris de ce travail n'est pas tant d'élaborer de nouveaux concepts que d'utiliser et d'intégrer des concepts existants et validés dans une perspective industrielle. Aussi nous sommes-nous inspirés d'un certain nombre de travaux existants.

A ce titre, il nous faut citer le projet CASPAIM (Conception Assistée des Systèmes de Production Automatisés en Industrie Manufacturière) développé au LAIL (Laboratoire d'Automatique et d'Informatique Industrielle de Lille). Ce projet se propose d'élaborer une organisation, une modélisation et une implantation "universelle" de la commande des systèmes de production. Il est bâti autour du modèle du cycle en V désormais classique organisant la chaîne de développement d'un système en plusieurs étapes dont les principales sont : la spécification, la conception, l'intégration et l'implantation [Lesage, 1991] [Verdin, 1991]. Divers outils, adaptés à chacune de ces phases, sont proposés. De nombreuses publications concernent ce projet. Citons, parmi celles que nous avons consultées : [Cruette, 1991], [Bourey, 1993], [Ausfelder, 1994], [Huvenoit, 1994].

De même, certains des principes de pilotage présentés dans la troisième partie de ce mémoire ont été élaborés initialement à l'Institut de Microtechnique de l'EPFL sous l'égide du Dr E. Verdebout. Ils ont été complétés, enrichis et adaptés dans le cadre de ce travail. Le lecteur intéressé par les travaux originaux pourra se référer à [Verdebout, 1992] ou [Verdebout, 1993].

Dans le cadre du projet SCOPES, deux "deliverables" concernent le pilotage : [SCOPES D4, 1993] qui expose les spécifications fonctionnelles des divers modules développés au cours du projet et surtout [SCOPES D21, 1994] qui présente les spécifications détaillées relatives au seul module de pilotage.

Organisation du mémoire

Ce mémoire est divisé en trois parties.

Partie 1

La première partie se concentre sur le cadre général du travail. Elle se décompose en quatre chapitres.

Le chapitre 1 présente les caractéristiques des systèmes de production considérés dans ce travail, à savoir les systèmes flexibles de production manufacturière (SFPM).

Le chapitre 2 expose les principaux concepts liés à l'automatisation et à la modélisation des SFPM. Les principes de structuration du système de commande (niveaux fonctionnels du CIM, notion de centres décisionnels) qui ont inspiré l'approche retenue dans ce travail sont notamment exposés.

Le chapitre 3 se focalise sur la solution de supervision proposée dans le cadre du projet SCOPES. Une architecture modulaire, constituée de six modules calqués sur les fonctions couramment rencontrées dans le secteur manufacturier (ordonnancement, gestion des stocks, pilotage, surveillance, analyse et maintenance) a été retenue. Le positionnement de notre étude, à savoir la fonction pilotage et le mode de simulation du superviseur est ensuite détaillé.

Dans le chapitre 4, nous présentons un exemple d'atelier flexible, celui du site expérimental de Telemecanique, qui nous permettra d'illustrer le propos dans la suite du mémoire.

Partie 2

La partie 2 propose une démarche d'analyse et de modélisation d'un SFPM en vue du pilotage. Elle est organisée en cinq chapitres.

Le chapitre 5 présente les divers "acteurs" du système de production, organisés suivant trois classes caractéristiques, à savoir les produits, les lieux et les moyens. L'analyse menée ne prend pas en compte la façon dont sera coordonnée la production. Elle s'attache uniquement à définir les *opérations* que fournit ou subit chaque "acteur".

La coordination fait l'objet du chapitre 6 concernant le processus de fabrication. Nous exposons les diverses notions (*synoptique*, *gammes opératoires*, *phases* et *opérations* essentiellement) permettant de spécifier la façon d'obtenir un produit dans une installation donnée.

A l'issue de ces deux chapitres d'analyse, nous serons capables, au chapitre 7, de présenter l'architecture du système de commande élaboré dans ce travail. Une hiérarchie externe bâtie sur une décomposition structurelle de l'installation permet de décomposer le système de commande en centres décisionnels. A l'intérieur de chaque centre, une hiérarchie interne bâtie sur une décomposition fonctionnelle, permet d'organiser les traitements suivant le type de décision de coordination à effectuer (ordonnancement, pilotage, commande).

Le chapitre 8 s'attache aux divers modèles nécessaires à la fonction pilotage. Il s'agit des gammes opératoires permettant de coordonner la production, du modèle de l'installation permettant de coordonner les transferts et du modèle des affectations reliant les opérations spécifiées dans les gammes aux moyens de l'installation.

Enfin, le chapitre 9 nous permettra d'expliquer la démarche adoptée en vue de valider le paramétrage des modules de supervision, notamment celui de pilotage, avant leur mise en oeuvre effective sur site. Il s'agit d'une simulation à partir d'un modèle du procédé ne prenant en compte que les contraintes physiques que s'imposent les différents "acteurs" du système, indépendamment des processus de fabrication. Le modèle que nous présentons permet de mettre en évidence l'ensemble des états utilisables du système. Il est ainsi possible de vérifier que les décisions de pilotage n'entraîneront pas le système vers un état bloquant ou interdit.

Partie 3

Cette partie, organisée en cinq chapitres, expose la solution pratique au problème du pilotage mise en oeuvre dans le cadre de nos travaux.

Le chapitre 10 présente les diverses contraintes s'exerçant sur le pilotage ainsi que les mécanismes retenus pour satisfaire ces contraintes. Le découpage de la fonction coordination en coordination de la production et coordination des transferts est également explicité. Les quatre chapitres suivants sont organisés suivant ce découpage.

Le chapitre 11 présente la solution retenue pour la coordination de la production : sont ainsi présentés successivement les divers types de requêtes qui, instanciées dynamiquement lors de chaque campagne de production, permettent de spécifier le travail à réaliser au niveau de la campagne (*ordre de fabrication*), au niveau des opérations de gamme (*bon de fabrication global*) et, enfin, au niveau des moyens de fabrication (*bon de fabrication local*).

Le chapitre 12 s'intéresse, quant à lui, à un mécanisme particulier intervenant lors de la phase de suivi de la fabrication entre deux changements de campagnes. Il s'agit du mécanisme de gestion des relations mutuelles entre bons de fabrication globaux d'une même campagne. Modélisées au moyen des *graphes d'influence*, ces relations permettent de gérer correctement l'avancement de la campagne et d'en contrôler l'achèvement.

Le chapitre 13 concerne la coordination des transferts. Les notions de *trafic* et de *routage*, ainsi que les deux mécanismes permettant de les mettre en oeuvre, à savoir les *règles d'aiguillage* et les *règles de destination*, sont présentés.

Le chapitre 14, enfin, expose une méthode algorithmique permettant de calculer les diverses règles d'aiguillage intervenant dans la gestion du trafic.

La conclusion nous permettra de récapituler les principaux avantages de la solution proposée et de préciser les perspectives d'utilisation industrielle de la maquette développée dans le cadre de cette recherche.

Partie 1

Contexte général du travail

Chapitre 1

Les systèmes flexibles de production manufacturière

Introduction

Nous présentons dans ce chapitre les caractéristiques des systèmes considérés dans le cadre de ce travail, à savoir les systèmes flexibles de production manufacturière (SFPM).

Ces derniers appartenant à la classe des systèmes à événements discrets complexes, nous en rappelons donc tout d'abord la définition.

Nous présentons ensuite les deux grandes catégories de systèmes de production manufacturière que sont les systèmes d'usinage et les systèmes d'assemblage. Les principales différences sont exposées. Une typologie des systèmes d'assemblage, domaine que nous avons particulièrement étudié, est également proposée.

Enfin la notion de flexibilité physique, caractéristique sine qua non de la flexibilité globale d'un système de production, fait l'objet d'une présentation détaillée.

1.1 Systèmes à événements discrets

1.1.1 Définition

Un *système à événement discret* (noté *SED* dans la littérature) se caractérise par le fait que l'évolution de ses *variables d'état*, c'est à dire les

paramètres caractéristiques qui en constituent l'image, s'effectue de manière discrète, par "sauts" successifs. Chaque saut est initié par l'occurrence d'un *événement*. Entre deux événements, les variables d'état, donc l'image que l'on a du système, restent constantes.

Il faut remarquer que, dans la nature, aucun système ne saurait répondre à la définition des SED. En effet, les variables d'état représentant alors des grandeurs physiques, elles évoluent toujours de manière continue. Ainsi en est-il, par exemple, de la variable caractérisant la position d'un ascenseur. Cependant, dans de nombreux cas, la connaissance à tout instant des valeurs exactes des variables d'état importe peu. Ainsi, pour la commande de l'ascenseur, il suffit d'en connaître la position par rapport aux étages desservis : rez-de-chaussée, entre rez-de-chaussée et étage 1, étage 1, etc. Le processus consistant à découper le domaine de valeur d'une variable d'état en diverses plages selon certains seuils est appelé *discrétisation*. Bien entendu, divers degrés de discrétisation sont possibles, suivant le niveau de précision souhaité. Pour l'ascenseur, par exemple, il peut être utile de définir des positions intermédiaires entre les positions définies précédemment, afin de gérer des phases d'accélération et de ralentissement dans le but d'améliorer le confort des utilisateurs et de ménager le matériel.

Tous les systèmes physiques ne se prêtent pas de la même façon à la discrétisation. De plus, un même système peut apparaître à la fois discret et continu suivant le point de vue auquel on se place. Ainsi, du point de vue de la régulation du moteur entraînant l'ascenseur, préférons-nous raisonner sur des variables non discrétisées.

1.1.2 Classification des systèmes de production selon leur degré de discrétisation

Au niveau des systèmes de production, il est courant de distinguer trois types de systèmes :

- Les *systèmes de production discrets* : ce sont ceux qui réalisent des transformations sur des composants unitaires. Ils se prêtent facilement à la discrétisation qui permet de les ramener aux hypothèses des SED. Cette catégorie peut être confondue avec la production manufacturière constituée par les ateliers d'usinage et d'assemblage.
- Les *systèmes de production continus* : ce sont ceux qui mettent en oeuvre des procédés continus modélisés par des équations d'état déduites de lois de la physique. Grâce à ces lois, il est possible de réguler les variables d'action de manière continue. Se classent dans cette catégorie les industries chimiques, le raffinage du pétrole, la production du verre et du papier, le traitement des eaux, certaines activités agro-alimentaires, etc.
- Les *systèmes mixtes ou hybrides* : ces systèmes comprennent à la fois un aspect continu et un aspect discret. C'est le cas, par exemple, dans la sidérurgie où la coulée du métal en fusion, régulée à partir d'équations d'état continues, cède la place en fin de ligne à des entités discrètes (appelées blooms ou brames avant utilisation dans les laminoirs). Il en va de même dans

de nombreuses industries agro-alimentaires et pharmaceutiques (production de yaourts, de boissons, de médicaments, etc.).

Nous ne nous intéressons dans ce travail qu'aux systèmes de production discrets.

1.2 Assemblage vs usinage

Parmi les systèmes de production appartenant à la classe des SED, il est courant d'opposer systèmes d'usinage et systèmes d'assemblage. Des différences essentielles entre assemblage et usinage peuvent en effet être mises en évidence.

1.2.1 Définitions

Dans l'usinage, le produit à réaliser est obtenu à partir d'un bloc de matière brute que l'on façonne de façon à obtenir une forme et un état de surface donnés. Quelque soit le procédé employé (action mécanique, attaque chimique, faisceau électronique, jet de plasma, ...), une opération d'usinage aboutit toujours à retirer une quantité de matière à la pièce initiale.

Dans l'assemblage, le produit à réaliser est considéré comme un ensemble d'objets simples (les composants ou constituants) intégrés en un objet unique par le moyen de contraintes géométriques ou physiques (les liaisons) qui limitent ou annulent les possibilités de mouvement relatif des composants. L'assemblage d'un produit consiste donc en l'établissement de toutes ses liaisons dans un ordre approprié. Une opération d'assemblage comporte un positionnement géométrique suivi le plus souvent d'un procédé physique de solidarisation permanente (soudage, collage, etc.) ou non (vissage, clipsage, etc.).

1.2.2 Principales différences

Flux physiques

L'usinage procède par enlèvement de matière. Celle-ci étant considérée comme rebut, le seul flux "produit" à gérer est celui de la pièce à usiner. Au contraire, l'assemblage procède par ajout de composants distincts : un système d'assemblage d'un produit P_i à partir de m_i composants présente une sortie unique, mais m_i entrées qui doivent être organisées et combinées pour obtenir le flux de sortie. La synchronisation des flux convergents au niveau d'une opération d'assemblage est primordiale afin de réaliser le rendez-vous exact des constituants impliqués.

Temps opératoires

Contrairement à l'usinage, la plupart des opérations d'assemblage ont des durées brèves (de 0,5 à 10 secondes), comparables voire inférieures aux temps de transfert inter-opérations. Une optimisation rigoureuse du choix et de l'implantation des équipements s'avère nécessaire de manière à éviter le sous-emploi des équipements productifs ainsi que l'accumulation des en-cours dus aux systèmes de transfert. Ce problème d'optimisation complexe ne sera pas traité dans ce travail où nous supposons l'atelier préexistant ou connu sur plan. De nombreux travaux ont porté sur ce problème. Dans cadre du projet SCOPES, une approche par algorithmes génétiques a été adoptée [SCOPES D15, 1994] [SCOPES D18, 1995].

Technologies mises en oeuvre

Enfin, il est à remarquer qu'il n'existe actuellement pas de technologie propre à l'assemblage. Les problèmes de manipulation de composants ne se distinguent guère de ceux rencontrés dans d'autres domaines : manutention, chargement/déchargement de machines, conditionnement, etc. Les systèmes réalisant ces opérations sont le plus souvent spécifiques et dédiés aux seuls produits traités. D'où ce constat : la polyvalence qui existe en usinage ne se retrouve pas en assemblage. Le centre automatique d'usinage n'a pas d'équivalent en assemblage.

Dans la suite de ce mémoire, nous nous intéresserons plus particulièrement aux systèmes d'assemblage. Cependant, la plupart des modélisations et solutions apportées pour le pilotage s'adaptent également aux systèmes d'usinage.

1.3 Typologie des systèmes d'assemblage

Un système d'assemblage est constitué d'un ensemble de postes de travail et de stockage (manuels ou automatisés) reliés entre eux par un ou plusieurs systèmes de transfert (également manuels ou automatisés).

Les opérations réalisées au niveau des postes ne se limitent bien souvent pas à l'assemblage des composants proprement dit. Elles incluent également d'autres opérations nécessaires à la fabrication du produit : contrôle fonctionnel, opération de marquage, de groupage, d'emballage, ...

Les composants proviennent d'un système de production amont situé ou non dans la même entreprise (système d'usinage, de moulage, autre système d'assemblage).

1.3.1 Classement par mode opératoire et type de transfert

On peut classer les systèmes d'assemblage en fonction de leur mode opératoire et du type de transfert utilisé [Datin *et al.*, 1994].

Type de transfert / *Mode opératoire*	Lié (synchrone, indexé)	Libre (asynchrone)	
		en ligne	hors ligne
manuel	A1	A2	A3
manuel assisté (semi-automatique)	B1	B2	B3
automatique (polyvalent ou dédié)	C1	C2	C3

La figure 1.1 présente des exemples des différents systèmes mentionnés dans le tableau.

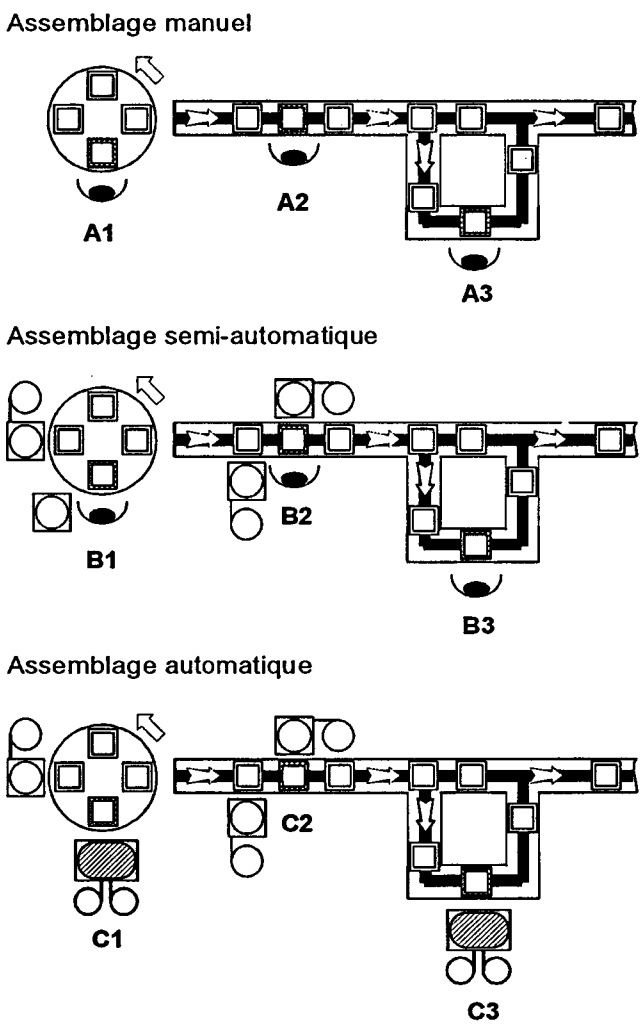


Figure 1.1 : exemples de systèmes d'assemblage.

Le *transfert lié* impose le synchronisme des postes desservis. De tels systèmes sont représentés par les machines à cames et les tables tournantes. Les cadences peuvent être élevées. Cependant, la panne de l'un des postes entraîne l'arrêt de l'ensemble des postes.

Le *transfert libre*, quant à lui, permet de découpler les postes successifs entraînant une plus grande souplesse en cas de panne. Les solutions technologiques sont nombreuses : chariots automoteurs filoguidés ou radioguidés, chariots tractés par chaînes enterrés, nacelles aériennes, convoyeurs sur table, etc.

Dans le cas des convoyeurs sur table, très répandus, les stations peuvent être disposées *en ligne* ou *hors-ligne*. Dans le transfert en ligne, les stations sont placées en série sur le transfert principal et les produits traités séquentiellement. L'installation hors ligne permet, d'une part, d'ajouter à la souplesse en introduisant des circuits alternatifs, et, d'autre part, de diviser le débit du flux parvenant aux postes. Nous citerons ici trois architectures couramment rencontrées dans les entreprises visitées (cf. figure 1.2) :

- en *guichet* : un produit est extrait du flux principal, traité puis réinséré dans le flux avant le traitement d'un autre produit.
- en *dérivation* : la station gère un aiguillage d'entrée et un aiguillage de sortie. La différence avec la station en guichet est la présence de stocks tampons en amont et en aval de la machine.
- en *épi* : à la différence de la station en dérivation, la station en épi n'a qu'un seul aiguillage qui joue le rôle d'aiguillage d'entrée et de sortie.

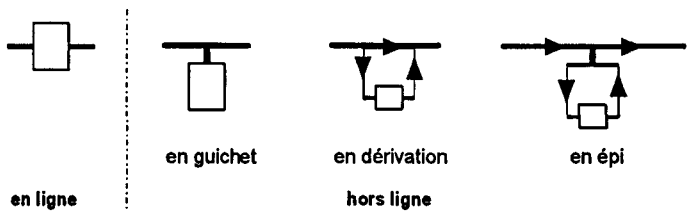


Figure 1.2 : configurations courantes des postes sur un transfert libre par convoyeur sur table

L'assemblage semi-automatique est un assemblage manuel avec assistance mécanique. Les distributeurs de composants ou autres dispositifs sont insérés dans la ligne et le temps d'assemblage par composant est ainsi réduit.

Parmi les équipements automatiques, deux grandes catégories peuvent être dégagées :

- *équipements dédiés* : ces équipements sont conçus pour réaliser une seule opération. Ils sont utilisés pour des raisons de coût plus faible et/ou lorsque les cadences sont élevées. Exemple : machines spécialisées synchrones, tables tournantes, têtes de travail et distributeurs automatiques, manipulateurs à cycle fixe, etc.
- *équipements polyvalents* : ces équipements peuvent accomplir plusieurs opérations différentes. Un temps de préparation est parfois

nécessaire entre deux opérations. Exemple : machines à transfert libre, têtes de travail programmables simples ou robots avec pinces spécialisées (capables d'accomplir plusieurs tâches d'assemblage), magasins de composants à chargement manuel.

1.3.2 Regroupement structurel

Un système d'assemblage peut souvent être considéré suivant plusieurs niveaux, chaque niveau pouvant lui-même être considéré comme un système d'assemblage à part entière.

Une décomposition courante dans l'industrie est la décomposition hiérarchique atelier / cellule / station / poste. Un atelier (ou ligne) d'assemblage est constitué de cellules (ou îlots). Chaque cellule fabrique un sous-ensemble du produit, ou participe à l'assemblage final. Les transferts entre cellules peuvent être automatiques à transfert libre (chariots filoguidés, chariots radioguidés, transstockeurs, convoyeurs aériens, etc.) ou manuels. Chaque cellule est elle-même composée de stations. Les transferts entre stations sont le plus souvent du type automatique à transfert libre (convoyeurs sur table, tapis roulant). Enfin, une station est parfois elle-même décomposée en terme de postes. Les transferts entre postes sont le plus souvent synchrones.

Il est à remarquer que cette décomposition n'est pas figée. Aucune norme ni consensus ne permet de définir clairement chacun des niveaux. En fait, chaque concepteur de lignes organise ses systèmes suivant ses propres critères : regroupement de produits, implantation des équipements, partage d'équipements, types des systèmes de transfert, etc.

Nous reviendrons sur ce problème de décomposition dans la suite de ce mémoire.

1.3.3 Classement par type d'organisation

Deux grands types d'organisation des flux physiques dans l'atelier sont couramment rencontrés dans l'industrie.

Organisation arborescente

La figure 1.3 illustre le premier type. Chaque cellule est une série strictement ordonnée (par le système de transfert) de stations. Plusieurs architectures sont possibles : en arbre, en arête, en ligne. Les produits circulent donc en respectant un ordre préétabli. Les systèmes d'assemblage à transfert lié rentrent bien sûr dans cette catégorie.

Dans cette organisation, la position du produit donne aussi son état de fabrication. La flexibilité éventuelle de la ligne provient uniquement de la fabrication successive possible de variantes de produits (cf. § 1.4). Le routage

des produits est imposé. La compensation des produits défectueux est parfois empirique.

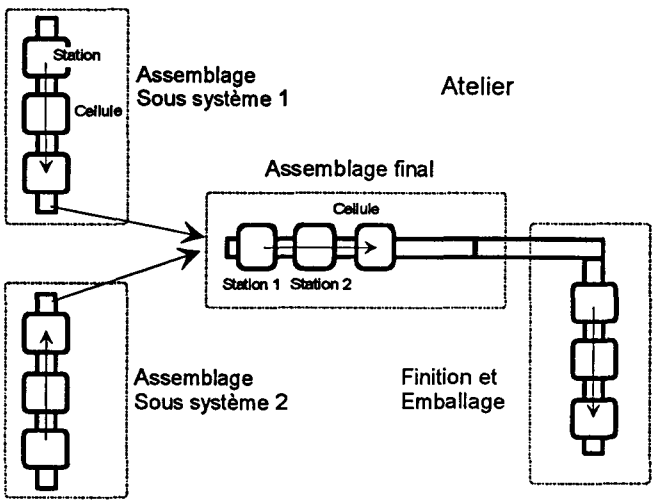


Figure 1.3 : ligne d'assemblage de type flow shop.

Dans la littérature, ce type d'organisation où les séquences opératoires sont identiques pour tous les produits est souvent qualifié de *flow shop*.

Organisation par circuit d'adressage

Dans le deuxième cas, illustré sur la figure 1.4, les produits circulent en tenant compte de la gamme d'assemblage et de la disponibilité des stations. Ils ne respectent pas un ordre préétabli des stations.

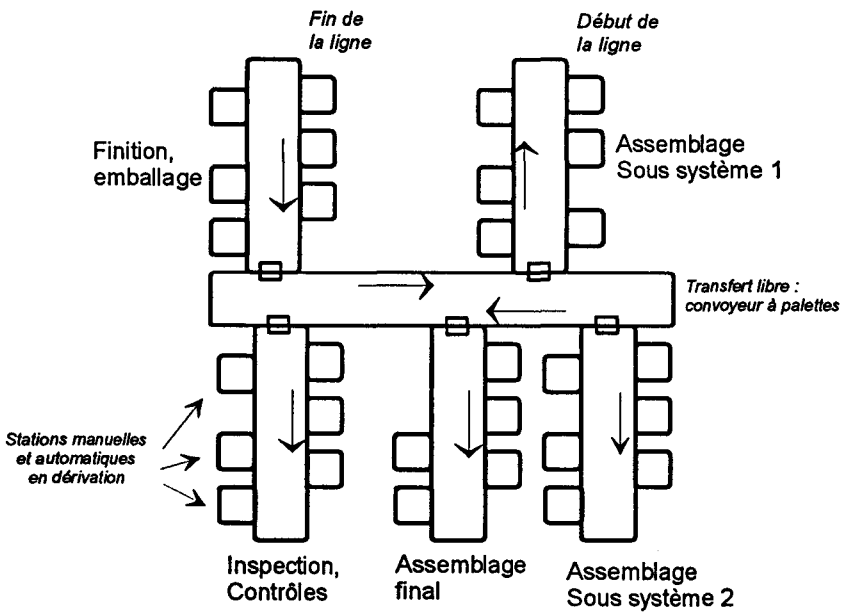


Figure 1.4 : ligne d'assemblage flexible.

Dans ce type d'organisation, les systèmes de transfert présentent de nombreuses flexibilités de routage (cf. § 1.4). Les cellules comportent souvent des boucles de retouche des produits défectueux pour assurer la fabrication du nombre de produits prévu dans chaque lot de fabrication.

Cette organisation est très prometteuse pour améliorer la productivité et réduire les délais de fabrication pour des produits avec de nombreuses variantes.

Dans la littérature, ce type d'organisation où plusieurs séquences opératoires sont simultanément possibles est souvent qualifiée de *job shop*.

La notion de système flexible, couramment employée par de nombreux auteurs, fait le plus souvent référence à ce second type d'organisation. Ceci est un léger abus de langage car les systèmes de type flow shop présentent très souvent la flexibilité de fabrication des variantes.

Les divers types de flexibilité caractérisant un système dit flexible font l'objet du paragraphe suivant.

1.4 Les flexibilités physiques

Pour qu'un système de production, tant dans l'usinage que dans l'assemblage[1], soit flexible, il est tout d'abord nécessaire qu'un certain nombre de conditions que nous qualifierons de physiques soient remplies. Les différents types de *flexibilités physiques* sont présentés ci-après. Nous verrons, au chapitre 2, que l'exploitation de ces flexibilités nécessite que soient également remplies d'autres conditions au niveau du système de commande, caractérisant la *flexibilité décisionnelle* [Combacau, 1991].

1.4.1 Flexibilités liées aux systèmes de transfert

Ce type de flexibilité apparaît lorsque les produits peuvent suivre différents trajets au sein du système. Deux cas de cas de figure se présentent :

- existence de plusieurs chemins pour desservir une même machine (flexibilité interne au système de transfert) ;
- existence de plusieurs machines candidates pour la prochaine opération à réaliser sur un produit.

[1] Dans la littérature spécialisée, il est courant d'utiliser les initiales FMS (Flexible Manufacturing System) et FAS (Flexible Assembly System) pour désigner respectivement les systèmes flexibles d'usinage et les systèmes flexibles d'assemblage.

Cette flexibilité de routage est fréquente sur les cellules d'assemblages dont les stations sont disposées hors ligne autour d'un convoyeur sur table en anneau. Une telle installation sera présentée au chapitre 4.

1.4.2 Flexibilités liées aux moyens de production

Trois types de flexibilité liés aux moyens de production peuvent être distingués :

- *polyvalence* des moyens : un même moyen de production peut réaliser indifféremment plusieurs opérations distinctes. Changer de nature d'opération peut cependant entraîner des coûts de changement d'outils, de téléchargement de programme, etc. ;
- *parallélisme* des moyens : plusieurs moyens identiques sont placés en parallèle, permettant ainsi de diviser les temps de cycle ;
- *partage de ressource* : un même moyen de production est capable d'effectuer des opérations sur plusieurs postes.

1.4.3 Flexibilités liées aux produits

Deux types de flexibilité concernent les produits :

- *ordre sur les opérations* : un même produit peut être réalisé suivant des séquences d'opérations différentes. Ce type de flexibilité apparaît essentiellement en usinage. En assemblage où les contraintes de précédence sur les opérations sont plus nombreuses, les séquences opératoires sont le plus souvent linéaires.
- *production simultanée* : plusieurs variantes de produits appartenant à des lots de fabrication différents peuvent être réalisées simultanément au sein du système de production et donc interagir du point de vue des délais de réalisation. La version flow shop de cette flexibilité, omniprésente dans les ateliers visités, consiste à être capable de produire successivement plusieurs variantes sur la même ligne ; les lots de fabrication se suivent. Parfois, des produits fictifs aisément reconnaissables (par exemple, des palettes peintes en rouge) permettent de repérer les changements de lots, et d'avertir ainsi les opérateurs.

Nous nous sommes volontairement limités dans cette présentation aux seules flexibilités utilisables dans le cadre de l'exploitation à court terme réalisée au niveau du pilotage. Il en existe d'autres, exploitables à moyen ou long terme pour adapter la production aux fluctuations du marché (introduction de nouveaux produits, modification des séquences opératoires, possibilité d'ajout de stations modulaires, etc.). Pour une étude complète, on se référera à [Browne *et al.*, 1984], [Falkner, 1986], [Tombak et De Meyer, 1988], [Sethi et Sethi, 1990].

Chapitre 2

Principes de commande et de modélisation des SFPM

Introduction

Ce chapitre présente les principes de commande et de modélisation des SFPM qui nous ont guidé dans nos travaux.

Après un bref rappel sur l'automatisation des SFPM, nous décrivons la démarche par décomposition hiérarchique couramment employée en productique. La structuration essentiellement verticale préconisée dans l'approche CIM est décrite. Nous détaillons également un certain nombre de concepts issus de l'approche systémique et relatifs à l'élaboration des décisions dans une structure hiérarchisée.

Le dernier paragraphe propose, quant à lui, une analyse du problème de la modélisation des SED et son application aux SFPM. Parmi les divers outils de modélisation graphiques bâtis sur la notion d'état, nous en avons retenu deux : les graphes d'état adaptés à la modélisation des entités individuelles et les Réseaux de Petri permettant d'exprimer les relations entre entités.

2.1 L'automatisation des SFPM

2.1.1 Structure des systèmes automatiques

Dans le cadre de son automatisation, un système de production peut être organisé, comme tout système automatique, suivant le découpage illustré sur la figure 2.1 [Combacau, 1991], [Cruette, 1991].

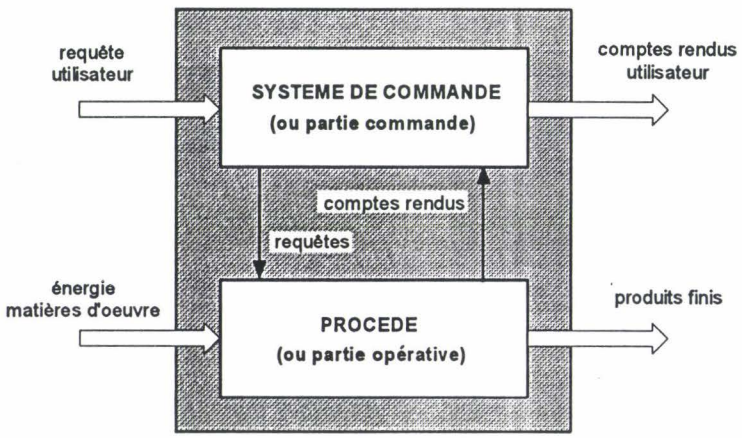


Figure 2.1 : organisation d'un système de production automatique.

Le *procédé* ou *partie opérative (PO)* désigne l'ensemble des objets matériels intervenant dans le système de production. Afin de réaliser la production demandée, le procédé met à la disposition de la commande un certain nombre d'actions commandables ou *opérations*. Ces opérations sont réalisées par l'intermédiaire des moyens de fabrication de l'atelier. Ces derniers consomment de l'énergie et des matières premières fournies par le milieu extérieur et les transforment en produits finis.

Le *système de commande* ou *partie commande (PC)* sert de relais entre l'utilisateur et le procédé. Sa fonction est de décharger l'utilisateur d'une grande partie du travail de contrôle, but fondamental de toute automatisation. Il génère pour cela des consignes ou *requêtes* à destination du procédé à partir de la consigne globale fournie par l'utilisateur et des informations ou *comptes rendus* remontées du procédé. Eventuellement, il rend compte à son tour à l'utilisateur du fonctionnement constaté, permettant ainsi un "feed-back" éventuel sur la consigne utilisateur. L'objectif du système de commande est de respecter, par un calcul adéquat des requêtes, la consigne utilisateur.

2.1.2 Problématique de la commande des SFPM

A un instant donné, un ensemble d'opérations déclenchables par le système de commande permet d'agir sur le procédé. Cet ensemble évolue dans le temps en fonction de la réalisation des opérations. En effet, la réalisation d'une opération modifie l'état du procédé, autorisant ou interdisant de nouvelles opérations. Ainsi, la réalisation de l'opération "prendre pièce" par un bras manipulateur élimine cette opération de la liste des opérations réalisables et rend possible la réalisation de l'opération "déposer pièce" par ce même bras manipulateur.

Dans certains cas, parmi les opérations réalisables à un instant donné, certaines ne peuvent être lancées simultanément. Nous parlerons alors de *concurrence* entre opérations. Cette caractéristique traduit les flexibilités que

possède le système. Elle impose un **choix** parmi l'ensemble des opérations en concurrence. C'est le cas des systèmes de production flexibles.

Les états que peut prendre le procédé à l'issue d'une opération peuvent être regroupés en trois catégories représentés sur la figure 2.2 (d'après [Combacau, 1991]) :

- les **états interdits** correspondent aux états rendus impossibles par les lois de la physique ; toute opération occasionnant le passage dans l'un de ces états entraîne systématiquement la détérioration de certains éléments du système. Par exemple, l'accès simultanée de deux éléments mobiles sur un même lieu correspond à un état interdit.
- les **états bloquants** sont ceux qui ne permettent plus d'atteindre l'état final souhaité. A partir d'un tel état aucune séquence d'opérations ne permet de retrouver la zone des états utilisables. Un exemple d'un tel état apparaît lorsque une ressource A attend pour être libérée la libération d'une ressource B elle-même en attente de la libération de A (blocage par étreinte fatale).
- les **états utilisables** correspondent à ceux utilisables par la commande ; à partir de n'importe quel état utilisable, il est possible de trouver une séquence d'opérations amenant le procédé à l'état final souhaité.

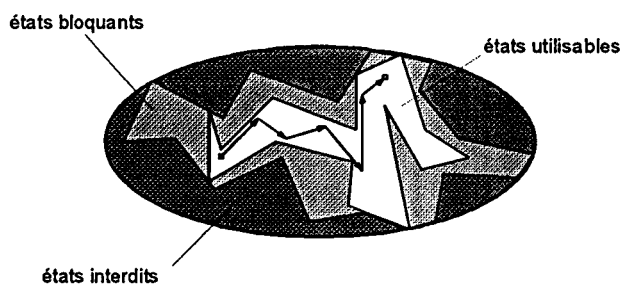


Figure 2.2 : états possibles du procédé.

Le problème consiste donc à trouver le meilleur chemin, parmi l'ensemble des états utilisables, de façon à amener le procédé de l'état courant à un état final donné correspondant à la consigne fixée par l'utilisateur. Etant donné la complexité des SFPM, le nombre de chemins disponibles peut être très grand. Une structuration du système de commande s'avère donc nécessaire.

2.2 Hiérarchisation des décisions

Face à un problème complexe, une solution intuitive consiste à le décomposer en problèmes plus simples. La hiérarchisation constitue l'une des techniques de structuration les plus employées dans le domaine de la productique.

2.2.1 Principe

D'après le dictionnaire, hiérarchiser un ensemble complexe consiste à l'organiser en groupes ou partitions successives.

Deux axes de partitionnement peuvent être dégagés : d'une part, un axe vertical consistant à découper l'ensemble en différents niveaux successifs, et, d'autre part, un axe horizontal consistant à partitionner chaque niveau en plusieurs modules. Le résultat est illustré sur la figure 2.3.

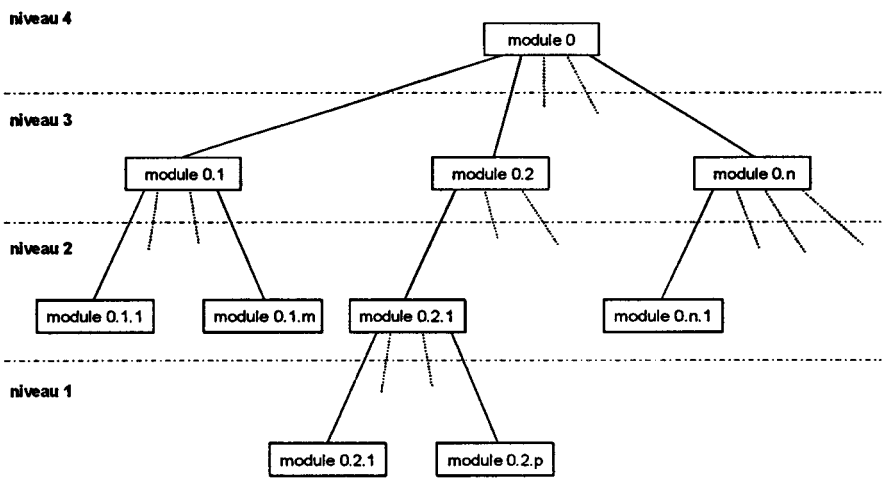


Figure 2.3 : structure hiérarchique.

Nous obtenons ainsi une structure pyramidale dans laquelle un module d'un niveau n est rattaché à un ou plusieurs modules du niveau n-1 et, généralement, à un seul module du niveau n+1.

La hiérarchisation, appliquée à la productique, s'est révélée féconde. Nous présentons, au § 2.2.2, les approches de structuration désormais classiques introduites dans le cadre du CIM. Le § 2.2.3 présente, quant à lui, une analyse selon une approche systémique du processus de décision dans une organisation hiérarchisée. L'architecture du système de commande que nous développerons au chapitre 7 s'appuie sur les concepts introduits par ces deux approches.

2.2.2 La structuration verticale du CIM

Le concept CIM ("Computer Integrated Manufacturing"), développé dans les années 80, a pour but l'intégration de l'ensemble des services de l'entreprise par la mise en place d'un réseau rapide et cohérent d'échange d'informations [Waldner, 1990]. Ce concept s'étend bien au delà des seules fonctions liées à la production.

La structuration des échanges d'informations proposée dans le cadre du CIM se focalise essentiellement sur une décomposition hiérarchique verticale, classiquement représentée sous la forme d'une pyramide. De multiples variantes existent suivant les auteurs. L'approche classique fait correspondre à chaque

niveau la fonction générale réalisée, le type d'ordinateur utilisé et la nature des réseaux informatiques mis en oeuvre (cf. figure 2.4).

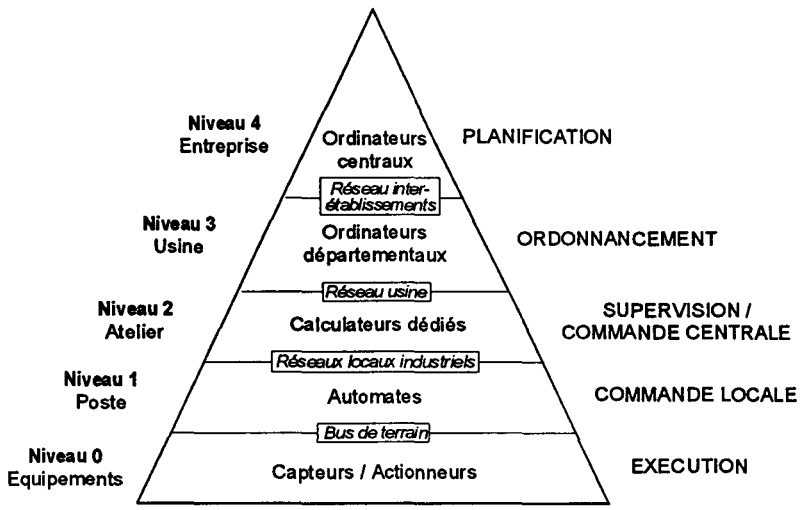
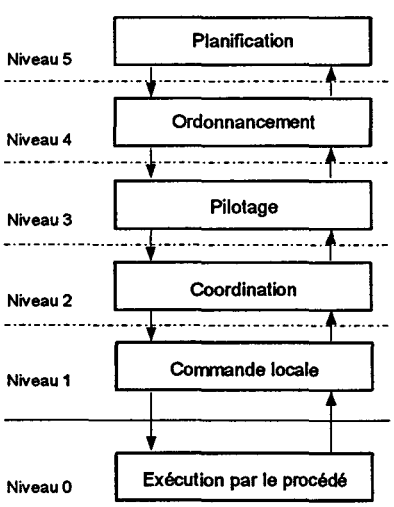
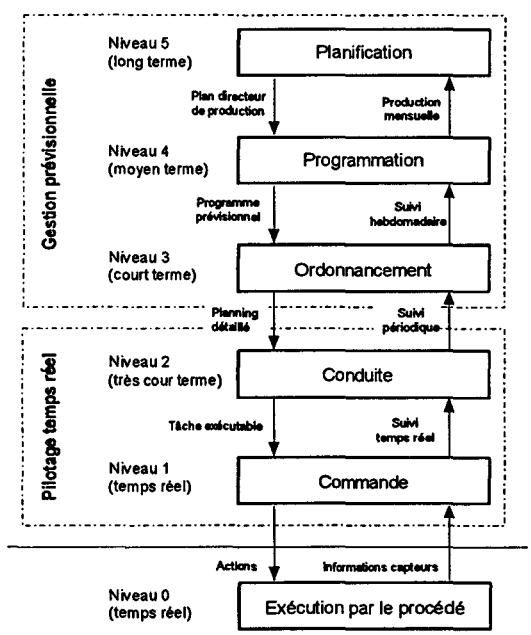


Figure 2.4 : pyramide CIM.

De nombreux travaux se concentrent sur l'aspect fonctionnel, le plus sujet à interprétations. Le nombre et le rôle exact de chaque niveau diffèrent suivant les auteurs. Citons, à titre d'exemple et sans entrer dans le détail de chaque fonction, l'approche adoptée par le groupe Système de production du LAAS à Toulouse [Cardoso, 1990] [Combacau, 1991] (cf. figure 2.5.a), celle proposée à l'Université de Valenciennes [Dindeleux, 1992], [Aleonard *et al.*, 1994] (cf. figure 2.5.b), enfin celle élaborée dans le cadre du projet CASPAIM II au LAIL [Huvenoit, 1994] (cf. figure 2.5.c).



- a -



- b -

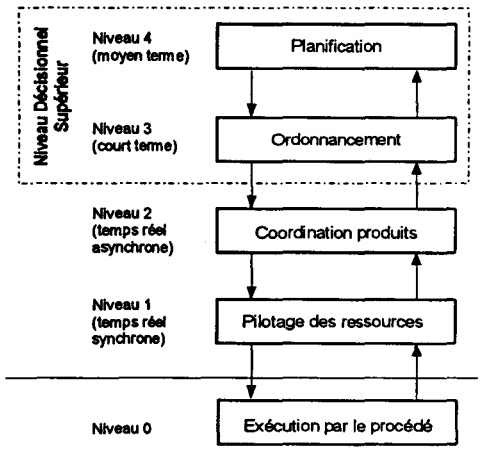


Figure 2.5 : hiérarchie de commande selon :a) [Combacau, 1991]
b) [Dindeleux, 1992] c) [Huvenoit, 1994].

Les relations entre différents niveaux doivent être caractérisées suivant deux phases distinctes [Dindeleux, 1992] : la *phase d'exploitation* établit les décisions directes de conduite, tandis que la *phase d'adaptation* fixe les objectifs et règles de stratégie utilisées lors de l'exploitation.

Phase d'exploitation

Dans cette phase, deux directions d'échange peuvent être mises en évidence : d'une part, une direction descendante suivant laquelle l'élaboration des requêtes s'effectue de manière de plus en plus précise jusqu'à ce qu'elles soient directement exécutables par le procédé ; d'autre part une direction ascendante selon laquelle les comptes rendus d'exécution sont successivement agrégés.

Les relations ainsi établies doivent former une boucle de régulation selon le principe illustré sur la figure 2.6 [Huvenoit, 1994].

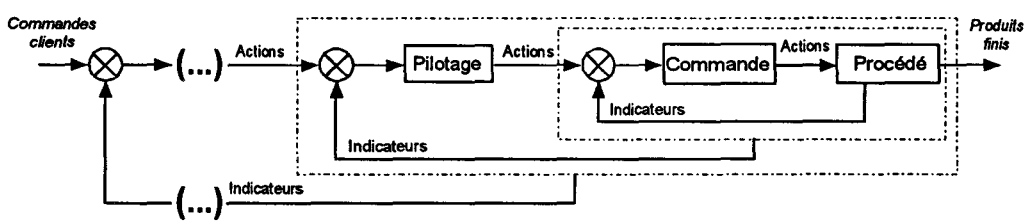


Figure 2.6 : boucles de régulation inter-niveaux.

Pour que cette structure fonctionne correctement, il est nécessaire de définir avec soin le domaine de compétence de chaque niveau. Une remise en question des consignes fournies à un niveau donné doit avoir lieu si les perturbations ne peuvent être absorbées au niveau correspondant : le niveau supérieur doit être informé (ou doit pouvoir détecter par lui-même) tout écart

par rapport au fonctionnement prévu afin d'appliquer les corrections nécessaires.

Phase d'adaptation

En plus des diverses boucles de régulation de la phase d'exploitation, il est nécessaire de mettre en place diverses boucles de contrôle permettant d'affiner les variables d'action, d'adapter les règles, voire de remettre en cause les stratégies utilisées au niveau de chaque boucle d'exploitation.

Ce contrôle, réalisé par les divers acteurs humains intervenant à chaque niveau, passe par la mise en place d'indicateurs de performance et de tableaux de bord permettant d'apprécier l'activité de chaque boucle (commande, pilotage, ordonnancement, etc.) suivant une stratégie globale d'exploitation souhaitée.

2.2.3 Notion de centres de décision

Loin de s'opposer aux principes exposés dans le cadre du CIM, l'approche par centres de décision proposée dans [Pellet, 1984] définit de manière plus fine les relations entre modules. Elle s'appuie sur une analyse générale du processus de décision dans une organisation hiérarchisée, selon une approche systémique, i.e. en replaçant le problème dans son environnement et en le considérant dans sa globalité [Le Moigne, 1984].

Chaque module de la hiérarchie est défini comme un *centre de décision* autonome en relation avec les autres centres. Deux types de relations peuvent être mis en évidence.

Relation inter-niveaux : coordination.

Ces relations interviennent entre un centre d'un niveau de décision donné et plusieurs centres du niveau inférieur. Etant donné que les centres d'un même niveau n'ont qu'une vue restreinte de l'organisation générale, ils ne sont pas à même d'assurer la cohérence de leurs décisions mutuelles. C'est au centre de décision de niveau supérieur que revient le rôle d'établir le cadre décisionnel des modules inférieurs : il s'agit d'une fonction de *coordination*. Les cadres locaux, calculés en fonction de l'objectif global assigné au centre supérieur, fixent les objectifs locaux de chaque centre. Ces objectifs locaux doivent être perçus comme des contraintes par les centres subordonnés : ils ne doivent en aucun cas être remis en cause localement sous peine de voir l'objectif global détruit.

Relation intra-niveau : coopération

Ce type de relations a lieu entre centres décisionnels d'un même niveau. Elles ont pour but, lorsque l'objectif local d'un centre ne peut plus être atteint, d'essayer d'absorber la perturbation résultante au niveau des centres voisins. Le niveau de coordination supérieur n'intervient que lorsque la perturbation ne peut être résolue par coopération. Ce principe soulève néanmoins le problème du respect de l'objectif global assigné au niveau supérieur.

En raison du principe de récursivité des systèmes introduit par Le Moigne [Le Moigne, 1984], chaque centre doit être considéré selon une vue externe identique à celle du système décisionnel auquel il prend part. Ainsi, quelque soit le niveau auquel on se place, un centre possède sa propre partie opérative (constituée des centres de décisions inférieurs auquel il impose ses décisions), et son propre environnement externe. En raison de la hiérarchisation, cet environnement peut être découpé en environnement externe et entourage, l'entourage étant constitué des centres du même niveau ou du niveau supérieur avec lesquels il dialogue (cf. figure 2.7).

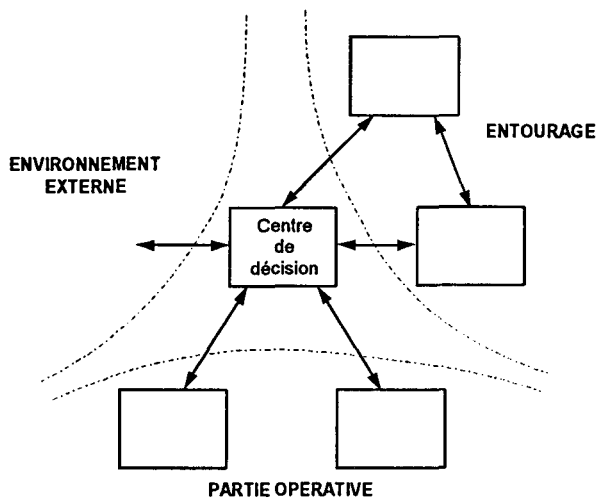


Figure 2.7 : vue externe d'un centre décisionnel.

De façon duale, la structure interne d'un centre décisionnel doit être identique quelque soit le centre considéré. Pellet distingue trois phases dans l'établissement d'une décision par un centre donné :

- une phase d'acquisition permettant d'acquérir les objectifs et le cadre décisionnel provenant du centre supérieur ; le centre local doit en particulier s'assurer que les objectifs assignés peuvent être réalisés dans le cadre décisionnel transmis.
- une phase de coopération avec les centres de même niveau permettant de limiter la remise en cause du cadre décisionnel précédemment accepté ;
- une phase de coordination où le centre coordonne à son tour les centres constituant sa partie opérative dans le but de satisfaire l'objectif qui lui a été assigné.

Appliquée à la structuration du système de commande d'un SFPM, cette analyse a donné lieu au système CODECO (COnduite DEcentralisée COordonnée d'atelier) développée au LAG de Grenoble [Pellet, 1985] [Kallel *et al.*, 1985]. La structuration en terme de centres décisionnels est ici appliquée aux fonctions de gestion proches du système physique. Les fonctions amont (planification, programmation) ne sont pas concernées. Cette approche propose

une décomposition de la conduite bas niveau en deux fonctions : d'une part, l'ordonnancement consistant à répartir dans l'espace et dans le temps la charge de travail, et, d'autre part, le lancement consistant à autoriser la réalisation des tâches sur chaque machine en respectant les conditions de réalisabilité (disponibilité des produits, des outils, des programmes, etc.). Le système proposé comprend ainsi deux niveaux décisionnels : un niveau de coordination global de l'atelier réalisant la fonction ordonnancement et un niveau de conduite locale des machines réalisant le lancement. La figure 2.8 illustre cette approche.

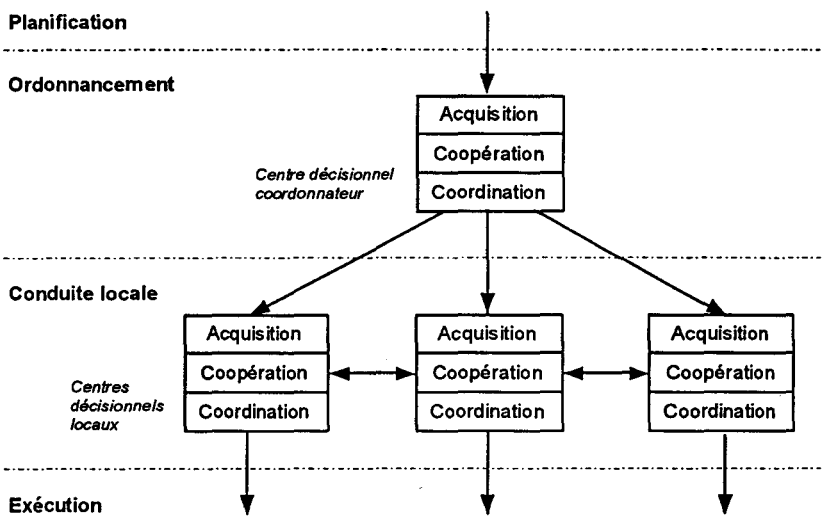


Figure 2.8 : structuration du système de commande dans CODECO (d'après [Dindeleux, 1992]).

Une généralisation du système aux niveaux de gestion supérieurs semble possible, mais n'est pas proposée par les auteurs.

2.2.4 Notion d'autonomie décisionnelle

Pour fonctionner correctement, le principe de hiérarchisation impose que chaque module possède un degré d'autonomie suffisant. Si ce n'est pas le cas, la dépendance des modules locaux vis à vis du module global génère un excès d'échanges entre niveau local et niveau global susceptible de perturber le fonctionnement du système, contredisant ainsi la propriété essentielle des systèmes de commande dits "temps réel"[1].

Une approche classique consiste à associer à chaque requête émise par un niveau n+1 vers un niveau n une marge temporelle que le niveau n peut soit conserver, soit transmettre vers le niveau n-1. Ces marges sont destinées à absorber les divers aléas survenant au niveau considéré afin d'éviter une trop grande fréquence des remises en cause des décisions du niveau n+1. Le

[1] Pour pouvoir être qualifiée de "temps réel", un système de commande doit en effet être capable de suivre l'état dynamique du procédé et d'établir les consignes à appliquer en un temps compatible avec la vitesse d'évolution du procédé.

problème consiste dès lors à trouver un compromis correct entre transmission totale de la marge (grande autonomie des niveaux locaux) et non transmission (absence d'autonomie).

Les travaux sur l'ordonnancement menés au LAAS à Toulouse sont bâtis sur ce principe de recherche d'autonomie. Le calcul et l'exploitation des degrés de liberté temporels ont abouti à plusieurs logiciels d'ordonnancement dits temps réel : ces logiciels ne fournissent plus, pour un objectif de production donné, un ordonnancement figé (absence de marge) mais un ensemble d'ordonnancements (valides sur une certaine marge) ; au pilotage temps réel de choisir parmi ces ordonnancements en fonction de l'état courant de l'atelier (logiciels MASCOT [Esquirol, 1987] et ORDO [Le Gall, 1989]).

2.2.5 Application de la hiérarchisation dans ce travail

Vis à vis de la structuration CIM, le travail présenté dans ce mémoire concerne les niveaux proches du procédé. Dans le cadre de la partie supervision du projet SCOPES, trois niveaux ont été retenus : ordonnancement, pilotage et commande. Nous expliciterons le rôle de chacune de ces fonctions dans la suite de ce mémoire.

La notion de centre de décision a été employée dans ce travail, avec, néanmoins, une interprétation sensiblement différente de l'approche CODECO. Les centres décisionnels seront en effet définis à partir d'un découpage structurel de l'installation et non plus à partir de la combinaison d'un découpage fonctionnel (ordonnancement / lancement) et d'un découpage structurel (atelier / machine).

La hiérarchie fonctionnelle ordonnancement-pilotage-commande sera associée à chacun des centre décisionnels ainsi mis en évidence : selon la position dans la hiérarchie, l'une ou l'autre des fonctions sera prédominante, mais toutes seront présentes.

La notion de coopération restant à notre connaissance très peu étudiée, nous ne retiendrons pas ce mécanisme : l'entourage des centres décisionnels se réduira aux seuls centres hiérarchiquement supérieurs. La notion de coordination sera, par contre, largement employée. Elle interviendra à plusieurs niveaux comme nous le montrerons au chapitre 7.

2.3 La modélisation des SFPM

2.3.1 Objectif de la modélisation

Afin de permettre une prise de décision, dans le cadre de la commande d'un SFPM, une connaissance sur les évolutions possibles de ce système doit être disponible. Cette connaissance est fournie par un *modèle* du système.

Un modèle d'un système doit permettre, à partir de n'importe quel état admissible du système, de déterminer les divers états accessibles et pour chacun d'entre eux d'établir la valeur de certains critères, considérés significatifs pour les décisions à effectuer.

Les critères significatifs différant suivant les niveaux de décision, plusieurs modèles existent pour un même système de production. Ainsi le procédé peut-il être modélisé selon divers points de vue : contrôle/commande, pilotage, ordonnancement, traitement d'erreur, etc. En outre, un même point de vue, considéré à deux niveaux "structurels" différents, fait appel bien souvent à deux modèles distincts. Dans ce dernier cas, l'utilisation d'un outil de modélisation présentant des possibilités de hiérarchisation peut s'avérer profitable.

2.3.2 Principe de modélisation d'un SED

Les Systèmes flexibles de production que nous désirons modéliser appartiennent à la classe des systèmes à événements discrets (cf. § 1.1). Nous rappelons ici les hypothèses classiques relatives à cette classe de systèmes :

- le système est considéré comme un ensemble fini d'*entités* en relation mutuelle ;
- chaque entité est caractérisée par un ensemble fini de *paramètres* ; l'ensemble des valeurs prises par ces paramètres à un instant donné s'appelle *état* de l'entité à cet instant ; l'ensemble des états de chaque entité à un instant donné définit l'état du système à cet instant ;
- un ensemble fini d'*événements*, les *opérations* dans notre cas, permet de faire évoluer les entités d'un état à un autre ; l'état du système reste constant entre deux événements ;
- une opération nécessite, pour pouvoir se réaliser, la présence de certaines entités dans certains états ; sa réalisation entraîne l'évolution de ces entités ; l'ensemble des opérations affectant l'état d'une entité définit le *comportement* de cette entité.

Modéliser un système à événements discrets, et donc un SFPM, revient à isoler les différentes entités caractéristiques du système et, pour chacune d'entre elles, à définir son comportement.

2.3.3 Les entités caractéristiques d'un SFPM

La détermination des diverses entités caractéristiques du système et de leurs paramètres est un problème crucial. Ce problème s'apparente à celui du choix des objets dans les approches objets en informatique. Il n'existe pas de règle absolue en ce domaine. Si quelques méthodes d'aide à la conception des objets existent (GOOD : General Object-Oriented Software Development ; HOOD: Hierarchical Object-Oriented Design ; MOOD : Multiple-View Object-Oriented Design Methodology ; OOA : Object-Oriented Analysis ; etc.)

[Barbier et Jaulent, 1992], le choix des "bons" objets reste cependant lié en dernier ressort à la compétence du concepteur. Il en va de même de la définition des paramètres caractéristiques et du comportement des divers objets.

Dans le cas des SFP, plusieurs candidats "objets" peuvent être identifiés à partir d'une description succincte :

> L'objectif d'un SFP est de produire un certain nombre de *produits* à partir de *pièces brutes* (usinages) ou/et de *constituants* (assemblages). Ces produits et constituants à traiter ou assembler circulent dans l'atelier en occupant divers *lieux* de transit (convoyeurs, palettes, plateaux, stockeurs ...). Les transferts d'un lieu à l'autre ainsi que les traitements et assemblages sont effectués par l'intermédiaire de *moyens* ou *ressources actives* (opérateurs humains, robots, machines spéciales ...). L'ordre des opérations (transferts, traitements, assemblages ...) n'est pas quelconque a priori : les séquences d'opérations réalisables par chaque ressource sont fixées (*comportement* de la ressource), ainsi que celles réalisables sur les produits (*gammes de fabrication*).

En résumé, des produits occupant divers lieux de l'atelier subissent des opérations fournies par les moyens de production. La connaissance de l'état des lieux, des produits et des moyens suffit donc pour connaître l'état du système.

La démarche d'analyse et modélisation que nous détaillerons en partie 2 s'articulera donc autour des éléments appartenant à ces trois classes.

2.3.4 Les outils de modélisation

Une fois les entités caractéristiques isolées, la spécification de leur comportement nécessite l'emploi d'une **méthode** et d'un **outil de représentation** adaptée. L'aspect méthodologique sera traité dans la seconde partie de ce mémoire. Quant à l'outil de modélisation à adopter, il doit permettre de représenter les contraintes logiques sur les opérations, caractéristiques des SED, à savoir :

- le *séquencement* des opérations (exemple : lever butée → indexer palette → déclencher cycle opératoire → désindexer palette → etc.) ;
- le *parallélisme* des opérations (exemple : plusieurs produits peuvent être simultanément traités sur plusieurs postes) ;
- la *synchronisation* entre opérations (exemple : assemblage de deux sous-ensembles) ;
- l'*alternative* ou *conflit* entre opérations (exemple : aiguiller une palette vers une station en dérivation ou la laisser passer) ;

Parmi les outils de modélisation disponibles, nous nous sommes particulièrement intéressés aux outils graphiques, facilitant le raisonnement humain. Citons en particulier, sans prétendre à l'exhaustivité :

- les *automates d'état* (ou machines à états ou graphes d'état) [Courvoisier et Valette, 1986];

- le *GRAFCET* [Blanchard, 1979] [Bouteille *et al.*, 1992] ;
- les *Systèmes de Transitions* [Arnold, 1992] ;
- les *Statecharts* [Harel, 1987] ;
- les *Réseaux de Petri* [Petri, 1966] ;
- les *Réseaux de files d'attente* [Gelende et Pujolle, 1982].

Lorsque l'on ne s'intéresse qu'aux états successifs des entités considérées isolément, les automates d'états suffisent. Nous les retrouverons donc dans la partie 2 relative à la modélisation. Cependant, dès que l'on désire modéliser les interactions mutuelles entre entités, les graphes d'état s'avèrent très vite insuffisants, entraînant une explosion combinatoire du nombre d'états recensés. Notre choix s'est alors porté sur les Réseaux de Petri (RdP par la suite) pour les nombreux avantages que cet outil présente et que nous résumons au paragraphe suivant. Une présentation du fonctionnement des RdP et des diverses variantes disponibles peut être trouvée en annexe A.

2.3.5 Avantages et inconvénients des RdP

La modélisation à l'aide des RdP offre premièrement l'avantage d'être une méthode de modélisation graphique permettant ainsi un maniement aisée et intuitif par l'esprit. Malheureusement, elle ne fournit pas de méthode universelle pour modéliser un système donné. Une méthode courante consiste à rechercher des modules encapsulant des motifs fréquemment rencontrés afin de permettre la représentation de structures complexes. Citons à ce titre les motifs développés au LAG à partir des RdP colorés (blocs ressource, fifo, stock, etc.) [Martin, 1987] ou la méthode par graphes de processus communicants développée dans le cadre du projet CASPAIM I au LAIL [Bourey, 1988]. Ces méthodes ne sont pas les seules. Nous en fournissons une, adaptée à nos besoins, au chapitre 9.

Comme le montre la figure 2.9, les RdP fournissent une syntaxe suffisante pour modéliser les caractéristiques logiques pouvant apparaître dans les SED.

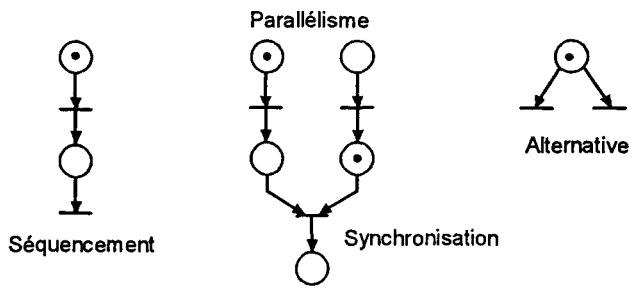


Figure 2.9 : représentation par RdP des principales contraintes logiques sur les opérations dans un SFPM.

Le formalisme initial des RdP ne prend pas en compte de manière explicite le concept de hiérarchisation. Néanmoins, certaines méthodes de construction des modèles par analyse descendante (top-down) ou ascendante (bottom-up) aboutissent implicitement à des modèles hiérarchisés. Afin de

valider cet état de fait, une formalisation du concept de hiérarchisation appliquée aux RdP a été formulée dans [Huber *et al.*, 1990]. L'intégration de ces résultats à la méthodologie de modélisation proposée au chapitre 9 constitue une piste de recherche possible pour la suite de ce travail.

Un second avantage des RdP est que les concepts de base des RdP sont mathématiquement très avancés. Les RdP sont utilisés depuis longtemps dans le domaine de l'informatique et leur emploi tend à se répandre dans le domaine des systèmes de production. Ils sont, d'autre part, proche de l'outil GRAFCET largement utilisé par les automaticiens français. Cependant, nous devons nuancer ce point car de nombreux formalismes RdP existent (cf. annexe A) entraînant une méfiance de la part des milieux industriels. Une phase de standardisation serait nécessaire pour faire des RdP un véritable outil utilisable par les automaticiens en dehors du monde de la recherche.

Notons immédiatement que, dans le cadre de l'utilisation qui en a été faîte dans le cadre de ces travaux, cet inconvénient importe peu, les RdP devant être cachés en dernier lieu à l'utilisateur puisque nous visons à une génération automatique du modèle (cf. chapitre 9).

Un troisième avantage des RdP, qui nous a particulièrement intéressé, est la possibilité de validation formelle d'un modèle spécifié à l'aide de ce formalisme. Les travaux théoriques sur la preuve des "bonnes" propriétés d'un modèle RdP sont en effet très riches et suffisamment avancés pour fournir un certain nombre de résultats significatifs : absence de blocage mortel (réseau pseudo-vivant), retour à l'état initial (réseau réinitialisable), absence de point d'accumulation (réseau borné). Les méthodes actuelles sont basées sur la recherche des invariants, la réduction ou la synthèse progressive du graphe (analyse top-down ou bottom-up), ou encore le calcul du graphe des marquages accessibles. On se reportera à [Brams, 1983a], [Esteban, 1985] ou à [Peterson, 1981] pour une étude complète de ces méthodes.

Un dernier avantage, toutefois non exploité dans ce travail, est le fait qu'un modèle de commande, modélisé sous forme de RdP, peut être très rapidement transformé en un logiciel de commande. Deux techniques sont alors possibles. D'une part, la *compilation* consistant à traduire le modèle RdP dans un langage de programmation adapté (le langage doit notamment prendre en compte la notion de parallélisme) : le langage ADA semble être le langage de prédilection en ce domaine [Estrailler, 1989] [Huvenoit, 1994]. D'autre part, l'*interprétation* consistant à "jouer" le modèle codé dans une base de données grâce à un interpréteur ou "joueur" de RdP. Celui-ci fait évoluer le marquage du réseau et, à chaque événement, la commande est calculée et l'état du système mis à jour. La conservation d'un même formalisme, de la conception à la mise en oeuvre, est un avantage certain, car permettant d'éliminer tout risque d'erreur lors d'une transcription dans un autre formalisme.

Chapitre 3

La supervision dans SCOPES

Introduction

Après avoir mis en évidence les orientations actuelles en matière de supervision, nous exposons dans ce chapitre l'approche développée dans le cadre du projet SCOPES.

Le superviseur proposé, dédié au secteur manufacturier, est conçu sous forme de modules correspondant aux grandes fonctions de gestion couramment rencontrées dans les industries de ce secteur. Ces modules permettent une adaptation de la supervision en fonction du besoin.

L'aspect original du système réside dans l'intégration des fonctions au sein d'un même système facilitant ainsi la mise en place de boucles de régulation réactives entre fonctions.

Le positionnement principal de notre étude, à savoir la fonction pilotage, est ensuite explicité.

3.1 La supervision : pourquoi et comment ?

3.1.1 La supervision, réponse à un besoin

Avec l'automatisation de systèmes de production de plus en plus complexes est apparue très vite la nécessité de mettre à disposition de l'exploitant des informations claires lui permettant de contrôler son procédé. Cette nécessité s'est particulièrement faite sentir dans le domaine des systèmes continus mettant en oeuvre des installations pouvant s'étendre sur une grande

superficie (raffinerie, cimenteries, etc.). Les premiers systèmes de collecte et d'affichage centralisés d'informations consistent en des synoptiques ou tableaux de contrôle directement matérialisés sur les pupitres de commandes au moyen de voyants lumineux, de cadrans à aiguilles ou d'afficheurs numériques. L'adjonction à ces pupitres d'organes de commande (boutons poussoirs, manettes, etc.) permet de leur adjoindre des fonctions de commande centralisées rudimentaires.

La notion de supervision apparaît réellement avec l'introduction de l'informatique. Elle apporte souplesse d'évolution et puissance de traitement que ne possédaient pas les tableaux de contrôle figés utilisés auparavant.

Le rôle de la supervision reste toujours, à l'heure actuelle, de collecter des données de manière à effectuer des opérations de visualisation, de contrôle et de commande centralisées des systèmes automatisés. Cependant, le champ de la supervision s'est considérablement élargi. Elle apparaît aujourd'hui comme un véritable outil de traitement de l'information et de communication et ne se limite plus aux seuls process continus. Elle permet d'établir une base de données cohérente qui alimente les diverses fonctions de gestion de l'entreprise.

3.1.2 Vers des progiciels de supervision spécialisés

Les demandes de plus en plus sophistiquées des divers acteurs de l'entreprise ont conduit les logiciels de supervision à s'enrichir et à intégrer de nouvelles fonctions. Aux fonctions classiques de présentation et de stockage de données brutes (affichage de synoptiques, consignation d'états, archivage, journaux de bord, gestion de recettes, gestion d'alarmes, établissement de bilans, etc.) se sont progressivement ajoutées des fonctions complexes apportant une véritable valeur ajoutée à l'information (analyse statistique du procédé, surveillance dynamique, aide à la conduite, traitement d'images, identification, etc.).

Face à cette richesse fonctionnelle, le développement "*sur mesure*" d'une application de supervision se révèle très rarement rentable. Une seconde solution de mise en oeuvre consiste à utiliser des "*boîtes à outils*" ou bibliothèques fournissant des fonctionnalités de base sous forme de "briques logicielles" pré-développées. Cependant leur mise en oeuvre nécessite une compétence en informatique et sont de ce fait réservés aux grandes sociétés ou aux intégrateurs. La solution idéale pour les petites sociétés se trouve dans les *progiciels intégrés*. Dans cette approche, la notion de programmation est complètement remplacée par la notion de paramétrage. Les gains en temps de développement sont considérables et, du fait de la large diffusion que connaissent ces logiciels, les applications sont relativement exemptes d'erreurs.

Cependant l'avènement des produits de supervision standards ne répond pas à tous les problèmes des utilisateurs. D'où la spécialisation de certaines fonctions à un métier donné. Cette tendance est confirmée par l'apparition de progiciels mettant en avant leur performance par rapport à une application précise. Citons notamment l'introduction, dans certains outil de supervision modulaires, de fonctions dédiées aux procédés continus (modules de

régulation : PID, logique floue, etc.) ou à la gestion du bâtiment (contrôle d'accès, surveillance, gestion de l'énergie, climatisation, etc.).

C'est cette dernière orientation qui a été adoptée dans le cadre du projet SCOPES : l'objectif de ce projet, pour la partie conduite d'atelier, consiste à développer un superviseur adapté aux besoins propres du **secteur manufacturier**.

3.2 La supervision dans le projet SCOPES

3.2.1 Fonctions de supervision SCOPES

Six grandes fonctions, correspondant aux différentes fonctions de gestion de production couramment rencontrées dans l'industrie manufacturière, ont été recensées [Datin *et al.*, 1994].

Ordonnancement

L'ordonnancement réalise la gestion des commandes. Les références des produits indiquées dans les commandes sont destinées à reconstituer les stocks de certaines variantes, ou correspondent directement à des commandes clients.

L'ordonnancement consiste à organiser les commandes en ordres de fabrication (OF par la suite) et à lancer les OF. Chaque OF correspond à un lot de fabrication homogène : un lot désigne une seule variante de produit ; tous les produits du lot seront fabriqués selon une même gamme d'assemblage. La taille limite d'un lot dépend de l'atelier (cadence, taille des stocks, capacité des contenants utilisés pour le stockage ou la distribution, etc.).

La correspondance entre les commandes et les ordres n'est pas directe. Un OF peut regrouper plusieurs commandes d'un même produit. A l'inverse, une commande peut être décomposée en plusieurs OF pour intercaler la fabrication d'un autre OF, dans le cas d'une commande urgente par exemple.

Le choix d'affectation des moyens de production à utiliser est traditionnellement assigné à cette fonction. Pour ce problème, deux options sont possibles : soit l'affectation est résolue de manière déterministe par l'algorithme d'optimisation (ordonnanceurs classiques), entraînant une faible marge de manoeuvre en cas d'aléas ; soit la décision est traitée au niveau du pilotage par l'application de règles préétablies. Cette dernière option, favorisant l'autonomie du niveau pilotage, a notre préférence.

Gestion des stocks

On distingue les stocks de composants sur chaque machine, les stocks de produits en cours et les stocks de produits finis. La gestion des stocks consiste à maintenir ces divers stocks à un niveau souhaité. Diverses politiques de

gestion sont applicables. La décision de réapprovisionnement s'accompagne soit d'une commande auprès d'un sous-traitant (stocks de composants ou de matières premières), soit d'un ordre de fabrication que devra traiter la fonction ordonnancement (stocks de produits finis).

Pilotage

Le pilotage est la gestion des lots de fabrication en cours, également appelés *campagnes de production*. Il résulte en un séquencement des opérations sur les différentes stations qui permettent de fabriquer les produits dans chaque lot, conformément à leurs séquences d'assemblage.

Cette fonction doit gérer en temps réel les flexibilités éventuelles de l'atelier. Elle dispose pour cela des informations fournies par la fonction ordonnancement (prochain lot à effectuer) et l'utilisateur (règles d'affectation et de routage), ainsi que de l'état courant de l'atelier (mode de marche des stations locales, état des produits, ...).

Surveillance

La surveillance est la fonction d'affichage d'informations sur le déroulement de la fabrication dans l'atelier. On distingue des informations sur les équipements (mode de marche des stations, pannes de machines, ...) et les produits (lots en cours, produits en cours d'assemblage, taux de rebuts, ...). Ces informations sont destinées aux opérateurs, à l'agent de maîtrise qui supervise l'atelier ainsi qu'aux agents de maintenance.

Analyse

L'analyse est la fonction de traitement des données sur le déroulement de la production. Ces données concernent aussi bien les produits (retards à la livraison, taux de rejet par lot de fabrication, ...) que les machines (temps d'attente, de maintenance, ...). L'analyse sert de support au suivi et à l'amélioration de la productivité et de la qualité des produits.

Maintenance

La maintenance comprend l'ensemble des actions destinées à maintenir ou rétablir un moyen dans un état de fonctionnement donné. Deux stratégies de maintenance complémentaires doivent être distinguées :

- **Maintenance curative** : une panne survient dans un équipement ; il faut avertir les personnes concernées, en fonction de la gravité de l'incident. Les stations sont pourvues de procédures de réparation, qui peuvent influencer les fonctions de pilotage (si le routage des produits doit être modifié) et d'ordonnancement (si un produit ne peut provisoirement plus être fabriqué).
- **Maintenance préventive** : des opérations de maintenance sont prévues périodiquement selon différents critères, par exemple le temps de fonctionnement ou le nombre d'opérations effectuées par l'équipement.

3.2.2 Mise en oeuvre modulaire

Le système de supervision développé dans le cadre du projet SCOPES est illustré sur la figure 3.1.

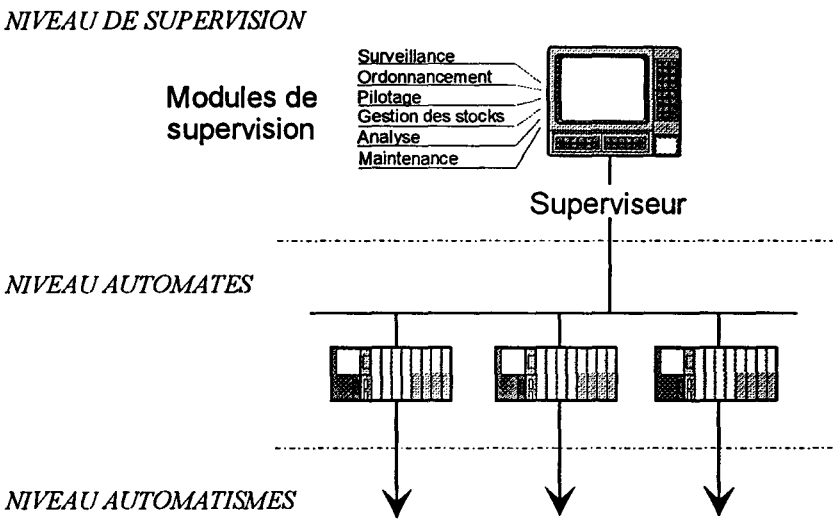


Figure 3.1 : système de supervision modulaire SCOPES.

Les diverses fonctions présentées au § 3.2.1 sont disponibles sous forme de modules, au niveau d'un superviseur, que le concepteur peut intégrer dans la gestion de sa ligne. Un certain nombre de traitements rattachés aux divers modules fonctionnels doivent également être exécutés au niveau des automates. Ces traitements, normalisés, seront également implantés de façon modulaire. Outre ces traitements, le niveau automate réalise également les fonctions de commande bas niveau des automatismes.

La figure illustre l'organisation du système de commande à deux niveaux retenue dans SCOPES : un niveau atelier ou cellule, contrôlé par le superviseur, et un niveau station, contrôlé par des automates. Nous montrerons cependant, au chapitre 7, comment cette architecture peut être généralisée à n niveaux : ce chapitre nous permettra en effet de préciser, à la lumière de l'analyse présentée dans les chapitres 5 et 6, le rôle et les liens mutuels des divers modules fonctionnels dans le cadre des principes de hiérarchisation exposés au chapitre 2.

L'approche modulaire adoptée est justifiée par le fait que les utilisateurs n'ont pas tous les mêmes besoins en matière de supervision. Trois grandes catégories d'applications doivent être distinguées d'après [Datin *et al.*, 1994] :

- Les **systèmes de suivi de fabrication** se contentent de collecter des données sur la fabrication. Ces données peuvent être exploitées en temps réel, par les opérateurs travaillant sur la ligne (dans un but de responsabilisation), ou en temps différé par les services qualité et maintenance par exemple. Ces systèmes n'intègrent alors que les modules correspondant aux fonctions surveillance et analyse.

- Les **systèmes de gestion de production** réalisent, en plus du suivi, la fonction ordonnancement : cette intégration permet une meilleure réactivité en cas d'aléas (pannes ou commandes urgentes). Ils ne permettent cependant pas d'agir sur le fonctionnement dynamique de la ligne : ce fonctionnement est entièrement réalisé de façon locale, sans intervention du niveau supervision.
- Les **systèmes de gestion de la fabrication** incluent, en plus, le module de pilotage. Ils permettent d'agir, au niveau supervision, sur le comportement de la ligne. Le paramétrage du niveau commande et/ou certaines prises de décision étant ainsi remontés à un niveau supérieur disposant d'une plus grande richesse d'informations, ces systèmes réalisent, à priori, une conduite plus performante car mieux finalisée.

Le système de supervision proposé peut fonctionner selon deux modes, présentés ci-dessous.

3.2.3 Modes de fonctionnement supervisé et simulé

Dans le *mode de supervision*, le superviseur est connecté à la partie opérative constituée, à ce niveau, des divers automates de station (au niveau des automates, la partie opérative est formée des divers automatismes, conformément au principe de récursivité des systèmes – cf. § 2.2.3). Les décisions et paramétrages calculés par les divers modules fonctionnels du superviseur, sont alors directement pris en compte par le système réel.

Dans le *mode simulation*, le superviseur n'est plus connecté au procédé réel, mais à un modèle simulant le fonctionnement des automates et des stations. Ce modèle, ainsi que l'outil chargé de le mettre en oeuvre, doivent respecter la spécificité des solutions techniques mises en oeuvre : nous verrons en effet que les solutions développées pour le pilotage ont des conséquences sur les algorithmes locaux de commande ; ces conséquences doivent être prises en compte dans la simulation. L'intérêt de ce mode sera explicité au paragraphe 3.3.3.

3.3 Positionnement et intérêt de l'étude

Les travaux menés dans le cadre de cette recherche ont porté principalement sur l'étude et la mise en oeuvre de la fonction pilotage, tant au niveau du superviseur (module de pilotage) qu'au niveau des automates (intégration à la commande de bas niveau et communication avec le module de supervision). La structuration hiérarchique et l'intégration des fonctions retenues dans le projet SCOPES constituaient autant d'opportunités pour la mise en place de solutions innovantes, certes encore peu répandues dans

l'industrie mais répondant à un réel besoin. Le besoin en pilotage et l'intérêt d'une solution hiérarchisée font l'objet du paragraphe 3.3.2.

Dans le cadre de nos travaux, un effort important a été consacré à la modélisation. Nous nous sommes très rapidement intéressés à l'étude d'un modèle de validation dont l'analyse nous aurait permis d'extraire un certain nombres de règles permettant d'assurer que les décisions de pilotage maintiennent le procédé dans la zone des états utilisables (cf. § 2.1.2). La première étape de ce projet, à savoir la définition et le développement d'une méthodologie de modélisation générale a pu être mené à bien dans le cadre de la recherche présentée dans ce mémoire. Cette méthodologie fait l'objet du chapitre 9. La seconde étape, à savoir l'étude par méthodes analytiques du modèle ainsi obtenu constitue le sujet de travaux à venir.

Faute de validation théorique, le modèle obtenu s'est néanmoins avéré intéressant dans le cadre du mode simulé du superviseur. L'étude et la réalisation d'un module de simulation à partir de ce modèle a ainsi constitué une seconde partie de nos travaux. Les avantages d'un tel mode de simulation dans le cadre d'un superviseur sont exposés au paragraphe 3.3.3.

3.3.1 Approches de structuration traditionnelles

Avant d'aborder les avantages de la solution hiérarchisée préconisée dans le cadre de ces travaux, nous rappelons ici les approches de structuration les plus rencontrées dans l'industrie. Ces deux approches sont le plus souvent employées en parallèle, à des niveaux "structurels" (poste, station, cellule, atelier, usine – cf. § 1.3.2) différents. Il est en effet rare qu'une seule des approches puissent répondre à l'ensemble des contraintes existant aux différents niveaux : la première solution présuppose une réactivité que ne possède pas toujours l'ensemble du système de production. A un certain niveau de gestion, il est ainsi courant d'anticiper sur la demande (calcul des besoins par la méthode MRP par exemple) afin de compenser l'inertie des niveaux inférieurs. Une structure hiérarchisée s'impose alors, afin de mettre en oeuvre les niveaux "prévisionnels" de la gestion (ordonnancement, planification, programmation).

Solutions "locales" ou "horizontales" (niveaux cellule et entreprise)

Au niveau cellule, plus rarement atelier, les solutions techniques de pilotage que nous avons le plus souvent observées reposent sur l'utilisation d'automates et la mise en oeuvre de stratégies locales. L'échange d'informations entre centres de charge (stations dans le cas d'une cellule, cellules dans le cas d'un atelier) s'effectue de façon directe, selon une structuration horizontale.

Deux stratégies sont communément rencontrées : le pilotage par l'aval (ou par la consommation) consiste, pour un centre fournisseur, à ne produire que sur demande d'un centre client (flux tiré). Cette stratégie trouve son application la plus célèbre avec l'approche Kanban. A l'inverse, le pilotage par l'amont consiste, pour un centre client, à ne produire que sur ordre d'un centre fournisseur (flux poussé). Notons que ces deux stratégies ne sont pas

exclusives : elles peuvent être simultanément rencontrées au sein d'un même système, deux centres de charge pouvant être gérés différemment, l'un en flux tiré, l'autre en flux poussé.

La mise en oeuvre des échanges informationnels repose le plus souvent sur des techniques "papier" classiques (type bons de fabrication), plus rarement sur des moyens informatiques (réseau locaux reliant chaque poste), ce dernier système aboutissant généralement à la mise en place d'une structure hiérarchisée.

Il est à noter que cette solution "horizontale" se retrouve systématiquement au niveau entreprise : il est clair en effet que les échanges entre entreprises différentes, s'effectuent selon un mode horizontal, sans intervention d'un niveau hiérarchique. Le mode de fonctionnement est le plus souvent en flux tiré, les livraisons des fournisseurs étant "tirées" par les commandes des clients.

La situation est moins évidente au niveau des usines d'une même entreprise : un niveau hiérarchique peut alors planifier les échanges inter-usines. Cependant, pour des raisons d'indépendance entre usines, l'approche horizontale est généralement adoptée dans la plupart des grands groupes industriels ; cette indépendance peut parfois même aboutir à la mise en concurrence des usines fournisseurs de l'entreprise avec celles d'autres entreprises.

Solutions hiérarchisées (niveaux atelier et usine)

La mise en oeuvre d'un niveau hiérarchique est nécessaire lorsqu'on désire coordonner de façon centralisée l'activité de chaque centre de charge. Par abus, nous appellerons superviseur, l'organe centralisé réalisant cette coordination, quelque soit le niveau "structurel" concerné.

Deux approches extrêmes peuvent être distinguées.

La première approche, que nous qualifierons d'*approche planifiée*, consiste à établir un ordonnancement détaillé de la production prenant en compte l'ensemble des contraintes du système. Les superviseurs fonctionnant suivant cette approche intègrent un générateur d'ordonnancement et un contrôleur d'exécution. L'ordonnancement aboutit à un séquencement de tâches à réaliser au niveau de chaque centre de charge. Le module de contrôle-lancement effectue en temps réel le lancement des tâches en tenant compte de l'état courant de l'atelier. Il effectue des tests de faisabilité avant chaque opération. Si les tests sont positifs, le lancement ordonne l'exécution de la tâche par le centre de charge correspondant. Dans le cas contraire, la tâche est retardée jusqu'à ce que l'ensemble des conditions soit vérifié.

En raison des aléas survenant en cours d'exploitation, le planning prévisionnel ne peut, bien souvent, être respecté et un réordonnancement, gourmand en temps de calcul, doit alors être effectué. Ce manque de réactivité limite cette approche aux seuls niveaux atelier et usine ; aux niveaux inférieurs les réordonnancements seraient trop nombreux.

Le découplage entre ordonnancement et lancement, bien qu'à l'origine du manque de réactivité, constitue l'une des raisons qui ont permis l'implantation de cette approche dans les entreprises. En effet, si l'ordonnancement nécessite en lui-même une grande puissance de calcul[1], le lancement peut, quant à lui, être réalisé par des moyens réduits et répartis au niveau de chaque centre de charge, rendant ainsi ces centres indépendants du niveau hiérarchique pour les traitements temps réels.

La mise en oeuvre des échanges entre superviseur et centres locaux, peut s'effectuer soit par des techniques "papier" classiques, soit par informatique, cette dernière solution tendant à se généraliser avec l'apparition de réseaux locaux industriels performants et le développement des concepts CIM.

La seconde approche, que nous qualifierons d'*approche temps-réel ou non-planifiée*, consiste à résoudre les problèmes de choix au moment où ils surviennent, les décisions prises étant immédiatement appliquées. La combinatoire intrinsèque au problème d'ordonnancement est alors considérablement réduite. Si cette approche apporte une solution très réactive au problème des perturbations (pas de réordonnancement), elle ne permet pas de prévoir les conséquences à long terme des décisions prises et le système peut travailler à trop court terme.

La mise en oeuvre de cette approche s'appuie sur des systèmes experts temps réel implantés au niveau du superviseur. Cette centralisation d'une partie des traitements temps réels constitue un obstacle à la mise en oeuvre industrielle de cette approche. Admissible au niveau d'une station ou d'un poste, la dépendance à un organe de commande centralisé doit en effet être évitée aux niveaux cellule ou atelier. Pour notre part, nous n'avons pas rencontré d'exemple industriel mettant en oeuvre cette approche dans le secteur manufacturier.

Entre ces deux approches opposées, plusieurs solutions peuvent être envisagées de façon à concilier réactivité et optimisation globale. Ces solutions consistent à laisser une plus grande autonomie aux niveaux locaux, l'ordonnancement ne résolvant plus la totalité des choix. L'approche CODECO et les techniques d'ordonnancement par groupes de tâches permutables proposées par le LAAS (cf. chapitre 2) s'intègrent dans cette démarche.

Notons pour finir que la mise en place d'un niveau hiérarchique n'est pas incompatible avec une gestion en flux tiré ou en flux poussé : la transmission des ordres s'effectue alors par l'intermédiaire du superviseur.

---

[1] Notons néanmoins que des techniques manuelles d'ordonnancement existent et sont encore largement employées dans l'industrie.

3.3.2 Intérêt d'un pilotage hiérarchisé au niveau cellule

Nous n'avons pas rencontré, jusqu'à présent, de système de pilotage industriel hiérarchisé au niveau cellule. Deux raisons permettent d'expliquer cette situation :

- importance, à ce niveau, des perturbations nécessitant une très grande réactivité du système de pilotage ; les systèmes ordonnancement-lancement classiques trouvent ici leur limite ;
- réticence des industriels à rendre la production dépendante de traitements temps réels centralisés.

La solution de pilotage développée dans le cadre de ce travail propose une réponse à ces deux problèmes.

Appliqués aux SFPM, l'utilisation d'un module de pilotage centralisé coordonnant le travail des stations locales permet, en effet, un certain nombre d'avantages significatifs par rapport à une solution purement locale. Ces avantages sont de quatre ordres.

Amélioration du suivi de la production

Un niveau hiérarchique de pilotage permet un suivi discrétisé par station des données caractérisant l'état d'avancement de la production.

Cette amélioration du suivi trouve tout son intérêt dans le cadre de l'intégration avec les autres fonctions industrielles. Les tendances d'évolution étant détectées au plus tôt, il est alors possible de mettre en place des boucles de régulation performantes et réactives entre fonctions.

Les bénéfices poursuivis par l'amélioration du suivi de la production sont :

- ♦ un meilleur respect des dates de livraisons des commandes clients par une plus grande réactivité de la fonction ordonnancement ;
- ♦ l'amélioration de la qualité du cycle de fabrication par la détection précoce des problèmes de fabrication (usure des outils, mauvais réglage machine, lot de pièces défectueux, ...) ;
- ♦ l'amélioration de l'équilibrage des charges grâce à la connaissance instantanée de la productivité de chaque machine.

Optimisation de la gestion des campagnes

L'un des problèmes essentiels de la production par lots dans un système flexible de production est la gestion des changements de campagne au niveau de chaque ressource de production. Dans le cas d'une opération réalisable sur plusieurs stations, cette détection ne peut s'effectuer localement. Une connaissance globale est nécessaire pour assurer que toutes les pièces d'un lot ont bien subi l'opération et autoriser ainsi le changement de configuration des moyens de production.

Le problème est encore compliqué par la nécessaire gestion des défauts sur les produits. Lorsqu'une pièce est rebutée ou retouchée divers traitements doivent être effectués afin d'assurer l'achèvement correct de la campagne. Il

s'agit, notamment de vérifier que l'ensemble des opérations spécifiées ont toutes été correctement réalisées sur l'ensemble des pièces du lot. Seule une gestion centralisée est à même de mener à bien ces traitements.

Les bénéfices poursuivis par la gestion centralisée des campagnes sont :

- l'amélioration de la productivité en diminuant les temps morts entre campagnes successives ;
- un meilleur respect des dates de livraisons des commandes clients grâce à la compensation des pièces rebutées et à la gestion des boucles de retouche ;
- l'amélioration de la réactivité par la diminution de la taille des lots et la réalisation simultanée de plusieurs lots.

Amélioration de l'exploitation des flexibilités liées au routage

Afin d'améliorer la réactivité du système de commande, nous avons vu qu'un certain degré d'autonomie doit être accordé à chaque niveau hiérarchique de la commande. Au niveau du pilotage, l'autonomie est caractérisée par un certain nombre d'indéterminismes non résolus par l'ordonnancement. L'option retenu dans nos travaux est de laisser ouvert le choix du routage des produits (choix du trajet et/ou choix de la prochaine station). La résolution à un niveau centralisé de ces choix permet de profiter d'informations non disponibles localement, et donc de permettre une optimisation globale.

Les bénéfices poursuivis par une optimisation globale du routage sont :

- une meilleure réactivité aux pannes ;
- l'amélioration de l'équilibrage des charges sur les stations ;
- l'augmentation de la productivité.

Réalisation d'un outil de pilotage "universel"

Un dernier avantage, d'ordre pratique, d'un niveau centralisé de pilotage est la possibilité de standardiser les divers programmes réalisant les diverses fonctionnalités du pilotage. Les modules logiciels implantés dans les automates peuvent ainsi être identiques pour chaque station et configurés, par téléchargement de paramètres, à partir du module central de pilotage situé dans le superviseur. Les diverses données de configuration (séquences d'assemblage, paramètres opératoires, stratégies de choix, etc.) sont gérées de façon cohérente au niveau du superviseur et non plus distribuées sur les diverses stations de la ligne. L'utilisation de la puissance informatique du superviseur peut alors être exploitée (éditeurs graphiques, analyse de cohérence, etc.), permettant ainsi la définition aisée de nouvelles stratégies ou de nouvelles variantes sans stopper la production.

Les bénéfices poursuivis par une configuration centralisée sont :

- la simplification de la mise en oeuvre d'un système de pilotage ;
- l'amélioration de l'exploitation d'une ligne par la modification aisée des paramètres locaux ;
- l'augmentation du nombre de variantes pouvant être définies ;

- la réduction des coûts de développement d'une application grâce à la standardisation des programmes.

3.3.3 Intérêt du mode simulation

L'objectif du mode de simulation est de fournir une *aide à l'exploitation* du système de production. Cet aspect de la simulation est encore très mal rempli par les logiciels de simulation du marché actuel. Ceux-ci se concentrent essentiellement sur l'aspect *aide à la conception*, en fournissant des interfaces ergonomiques de construction et de modification des modèles, ainsi que des outils d'analyse des résultats performants. Lors de la phase conception, un outil de simulation s'avère très utile pour vérifier que l'installation respecte les contraintes de production en terme de performance et de coût : les choix sont très nombreux, tant au niveau de l'architecture de l'atelier que des solutions pour assurer l'augmentation progressive de la production.

En outre, lorsqu'ils se présentent comme outil d'aide à l'exploitation, la majorité des simulateurs se focalisent essentiellement sur l'aspect simulation proprement dit sans tenir compte des possibilités liées à l'architecture de commande réelle. Un fossé est ainsi apparu entre le monde de la simulation, capable de produire et valider des stratégies de pilotage sophistiquées, et le monde des automatismes, bien souvent incapable d'exploiter ces stratégies. L'approche adoptée dans ce travail se veut donc résolument inverse : nous nous sommes tout d'abord focalisés sur l'étude du module de pilotage et son intégration dans l'architecture de commande, avant de spécifier le module de simulation. Ce dernier prend ainsi en compte l'ensemble des fonctionnalités locales (niveau automates), notamment celles ayant trait à la spécificité des solutions techniques adoptées pour le pilotage. L'architecture logicielle de supervision retenue (cf. annexe D) est significative : la majorité des tâches logicielles développées, notamment pour l'ordonnancement et le pilotage, sont communes aux deux modes de fonctionnement.

En tant qu'outil d'aide à l'exploitation, la simulation s'intègre dans le cycle de vie d'une application de supervision à plusieurs niveaux.

Définition du cahier des charges

A ce niveau, la simulation permet au concepteur de l'application de supervision de fournir au client potentiel une maquette animée du produit final. Cette présentation attractive permet ainsi au client de se faire rapidement une idée des possibilités de la supervision ; cet aspect "commercial" ne doit pas être négligé car la supervision, bien que prometteuse, n'est pas encore entrée dans la réalité de toutes les entreprises du secteur manufacturier. Il est ainsi difficile au client éventuel de fournir un cahier des charges complet et exhaustif de ses besoins en supervision et cela d'autant plus qu'il n'existe pas de norme précise dans ce domaine.

L'utilisation de la simulation dans l'établissement du cahier des charges s'effectue selon le processus itératif suivant : modification de la maquette,

présentation du résultat au client, nouvelles modifications, etc. Grâce à la simulation, il est ainsi possible de valider auprès du client les diverses fonctionnalités du superviseur, tant au niveau des aspects statiques (constitution des écrans, synoptiques, etc.) que dynamiques (enchaînement des écrans, réaction aux aléas, alarmes, suivi de la production, etc.).

Phase de développement

Le développement d'une application de supervision se divise en deux phases. Dans une première phase, le concepteur configure le logiciel sur une version de développement du système. Une étape de validation est souhaitable avant la mise en oeuvre effective sur site. Cette validation doit être effectuée à deux niveaux :

- vérification de la configuration des modules : il s'agit de corriger d'éventuelles erreurs d'introduction des données (vérification syntaxique, logique d'enchaînement des écrans, etc.).
- vérification du comportement des stratégies de production utilisées au niveau des modules décisionnels (ordonnancement, pilotage et gestion des stocks) : par exemple, au niveau pilotage, les fonctions de stratégie laissées au libre choix de l'utilisateur doivent être réglées avec précision afin d'éviter la surcharge ou la famine de certains moyens, ou, plus grave, leur blocage.

L'utilisation de la simulation permet de faciliter cette validation, notamment sur le deuxième point nécessitant une évolution dynamique des données. Néanmoins, la nécessité d'établir, et donc également de valider, le modèle utilisé pour la simulation ne doit pas être perçue comme une charge de travail supplémentaire : la configuration du mode simulation doit au contraire s'intégrer dans le processus de configuration global.

Une fois la mise au point effectuée, il s'agit de transférer, dans une seconde phase, les divers fichiers de configuration sur une version d'exploitation du système, située dans l'atelier, en remplaçant le mode simulation par le mode supervision.

Formation du personnel

L'intérêt de l'option simulation pour l'aide à la formation des opérateurs provient de la similitude entre les outils (écrans, structures de données) utilisés en simulation et ceux utilisés en supervision. Le mode simulation peut ainsi être exploité par les opérateurs afin d'apprendre l'utilisation du superviseur sans occasionner de perturbations sur le fonctionnement réel de l'installation ou avant même que cette installation n'existe.

Phase d'exploitation

Lors de l'exploitation du système, il est parfois nécessaire de modifier les diverses stratégies d'exploitation paramétrant les modules fonctionnels de gestion dynamique de la ligne (pilotage, ordonnancement et gestion des stocks).

Cela peut être dû à l'introduction de nouvelles variantes, à l'automatisation d'un poste suite à une montée en cadence, ou au choix de nouvelles stratégies plus performantes (phase d'adaptation – cf. § 2.2.2). Là encore une validation est nécessaire avant implantation effective. Le recours à la simulation est justifié par le fait que cette validation ne peut être réalisée que très difficilement sur le système de production lui-même. Outre les risques matériels encourus, la comparaison de différentes stratégies dans divers cas de fonctionnement du système (arrêt ou panne de certaines machines) impose l'arrêt de la production normale du système. Envisageable pour un site expérimental ou lors de la phase de mise au point initiale du système, un tel arrêt du système est inadmissible en phase d'exploitation.

Une première possibilité est d'utiliser la simulation "en ligne", i. e. directement au niveau de l'atelier : le mode simulation est alors utilisé en parallèle au mode supervision sur le même superviseur. Un tel fonctionnement nécessite de dupliquer la base de données constituant l'image de l'atelier sur laquelle raisonnent les diverses fonctions de la supervision. La simulation serait alors conçue comme un outil d'aide à la décision "en-ligne" permettant à un opérateur de valider ses choix en comparant diverses alternatives. Cette application fut un temps envisagé dans le cadre de SCOPES, mais s'est révélée difficile à exploiter en pratique, tout au moins aux niveaux relativement bas retenus dans ce projet (cellule ou îlot) : à ces niveaux, les décisions doivent être effectuées dans des temps incompatibles avec une analyse préalable par simulation.

Le mode simulation retenu ne peut donc fonctionner conjointement au mode supervision. L'utilisation de la simulation en phase d'exploitation doit être effectuée "hors ligne" selon le même principe que celui décrit pour la phase de développement : mise au point au bureau d'étude puis transfert vers l'atelier. Néanmoins, un transfert de la base de données courante du superviseur de l'atelier vers le simulateur du bureau d'étude est possible afin de permettre l'étude en différé de diverses alternatives à partir d'un état réel de l'atelier.

Chapitre 4

Un exemple d'atelier flexible : le site productique de la DRD

Introduction

La mise en place en 1987 du Site productique au sein de la Direction recherche et Développement (DRD) de Telemecanique (Nanterre – France) avait pour but initial la mise en oeuvre et l'étude en vraie grandeur de l'offre Telemecanique en matière de systèmes automatisés de production. Depuis juillet 1992, cette étude se focalise sur les systèmes de supervision en raison du choix de ce site comme scénario de validation des résultats du projet SCOPES.

Ce chapitre propose une description de ce site qui sera utilisé, dans la suite de ce mémoire, comme exemple illustratif.

4.1 Architecture physique

Le site expérimental, présenté sur la figure 4.1, est constitué d'un système de magasinage central et d'une cellule d'assemblage reliés entre eux par un système de convoyage par nacelles.

4.1.1 Magasin central

Constitué d'un stockeur rotatif dont les étagères supportent des plateaux dans lesquels sont rangés les produits (2 plateaux par étagères * 10 étagères * 6

produits par plateaux = 120 produits maximum), le magasin central (noté MAG sur la figure) assure une fonction de tampon de stockage vis à vis d'une zone non détaillée (service livraison / réception ou autre atelier). Un manipulateur pneumatique à cycle fixe assure le transfert des plateaux entre ce stockeur et les nacelles du système de convoyage aérien.

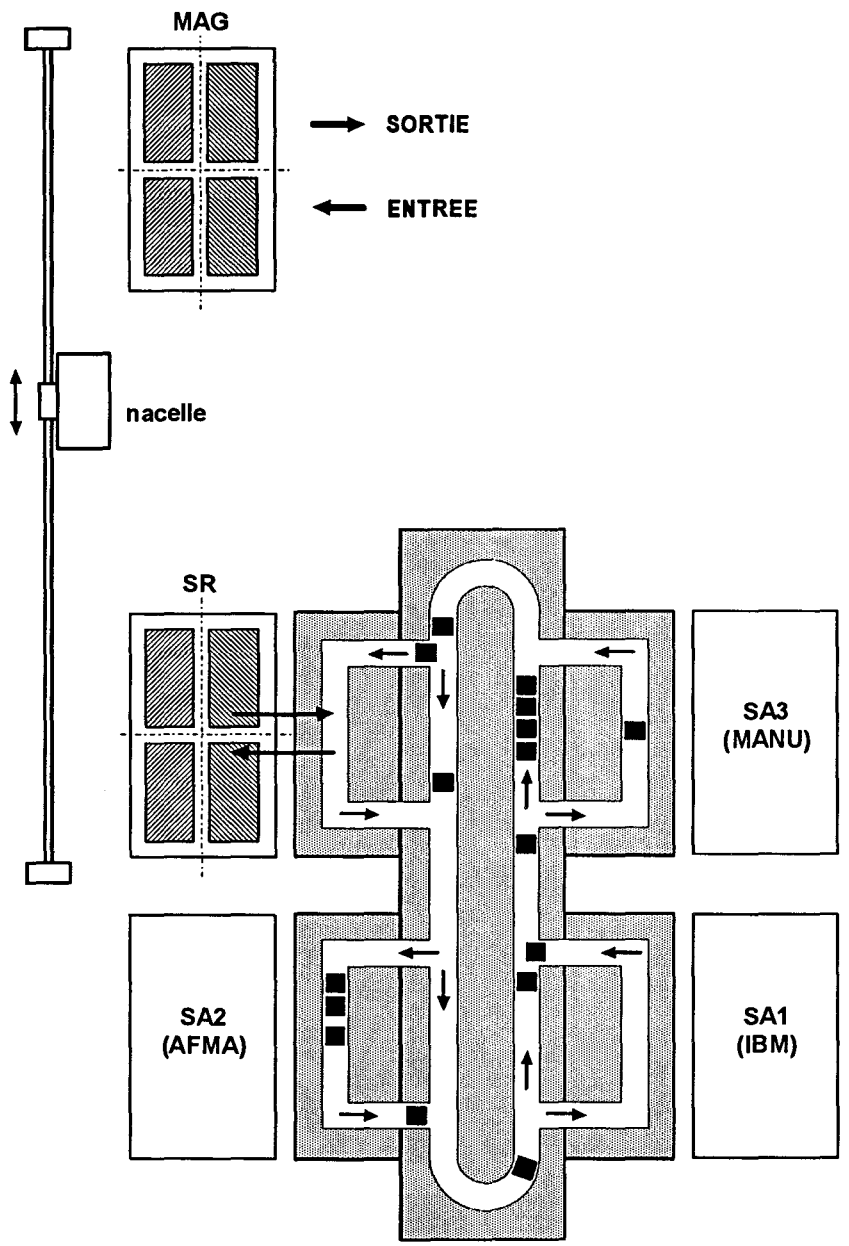


Figure 4.1 : architecture du Site.

4.1.2 Système de convoyage

C'est un convoyeur aérien constitué d'un monorail motorisé sur lequel circulent des balancelles ou nacelles (une seule pour le site). Sa fonction est de

véhiculer les plateaux entre les stockeurs rotatifs du magasin central et de la cellule d'assemblage.

4.1.3 Cellule d'assemblage

Elle est constituée d'un convoyeur central reliant quatre stations de travail montées en dérivation (SA1, SA2, SA3, SR). La fonction de ce convoyeur est d'assurer le transport des pièces par l'intermédiaire de palettes en nombre a priori variable. Des étiquettes magnétiques permettent de coder un certain nombre d'informations sur les palettes au niveau de chaque station. Ces informations sont utilisées pour permettre le routage des palettes au niveau de chaque aiguillage.

Les assemblages sont effectués par ajouts de composants extérieurs sur des embases convoyées par les palettes. Ces opérations sont assurées par des robots d'assemblage au niveau des stations SA1 et SA2, et par un opérateur humain au niveau de la station SA3.

La palettisation des embases vierges et la dépalettisation des produits finis sont assurées par la station SR. Les embases vierges proviennent de plateaux placés sur les étagères d'un stockeur rotatif analogue à celui du système de magasinage. Les produits assemblés sont redéposés dans les emplacements vides des plateaux. Ce stockeur a pour fonction de stocker momentanément les embases (avant leur lancement sur le convoyeur central) et les produits assemblés (avant leur départ vers le magasin), découplant ainsi la cellule d'assemblage du reste de l'atelier. Les transferts entre plateaux et palettes sont effectués par un manipulateur programmable à deux axes, tandis que la manipulation des plateaux provenant de la nacelle est assurée par un manipulateur à cycle fixe.

4.2 Architecture informatique

L'architecture informatique permettant la commande du site est représentée sur la figure 4.2.

Chaque station de la cellule d'assemblage est contrôlée par un automate programmable industriel de marque Telemecanique. Ces automates sont connectés entre eux par l'intermédiaire d'un réseau local industriel Mapway. Sur ce réseau est également connecté un premier PC industriel CCX 77 de marque Telemecanique faisant office de superviseur cellule. Un second CCX 77 remplit le rôle de superviseur atelier : il est connecté, d'une part, au superviseur de la cellule d'assemblage (réseau Ethernet), et, d'autre part, aux automates assurant la commande de la nacelle et du magasin central (liaison Unitelway).

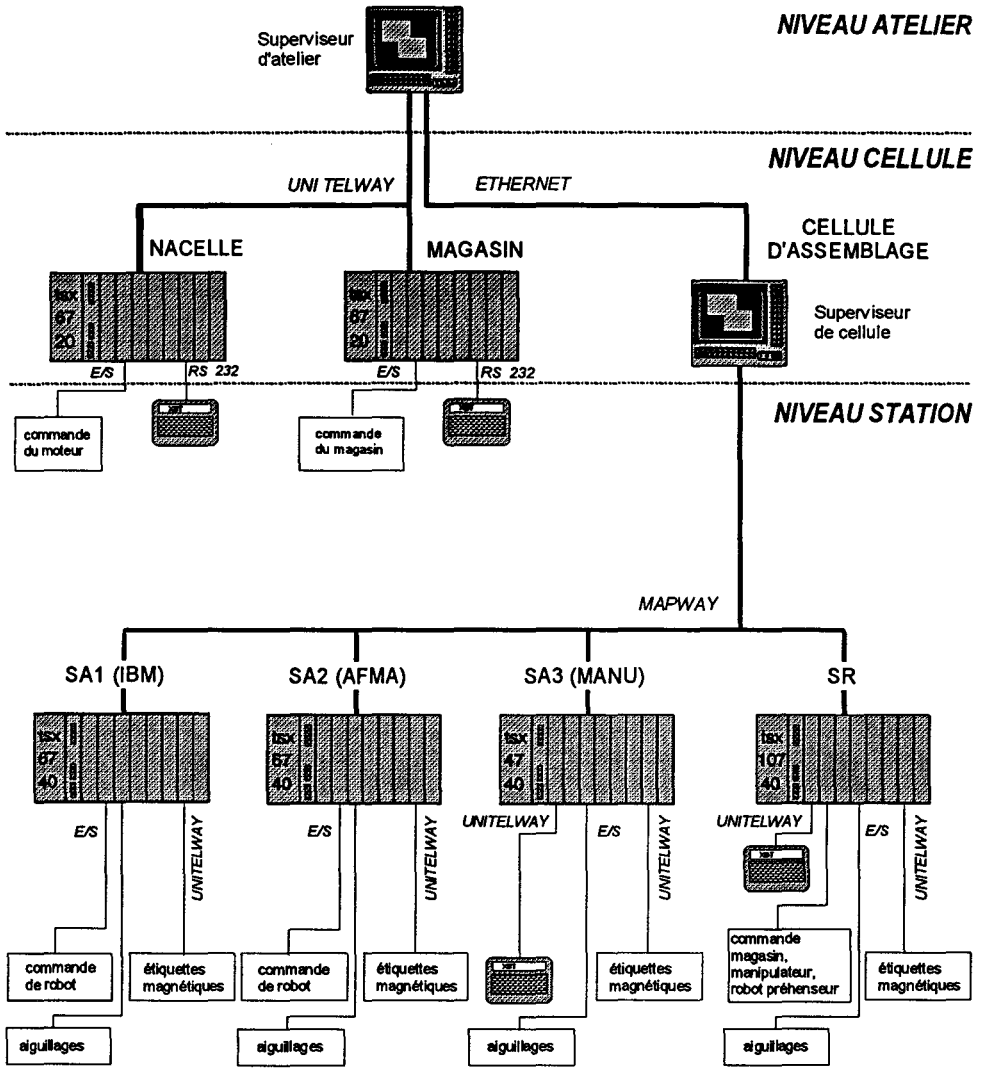


Figure 4.2 : architecture informatique du site.

4.3 Produits assemblés

Les produits assemblés ont été choisis pour leur simplicité en terme d'opération d'assemblage. L'objectif n'est pas, en effet, d'étudier la technologie de l'assemblage, mais les flux circulant dans l'atelier.

Les produits retenus sont constitués d'une embase dans laquelle sont insérées des petits cubes afin de former des lettres. Il y a trois types d'embases correspondant aux lettres S, E et P. Ces embases servent à fabriquer trois familles de produits (produit_S, produit_E et produit_P). Trois variantes sont possibles pour chaque produit. Elles sont matérialisées par la couleur des cubes insérés : noir, rouge et vert.

4.3.1 Schéma des produits

La figure 4.3 représente les trois types de produits fabriqués.

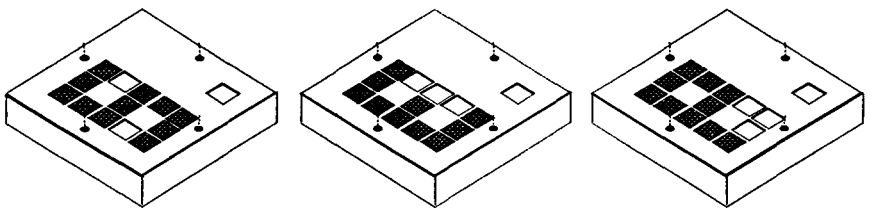


Figure 4.3 : les produits du Site.

4.3.2 Séquences d'assemblage

Les séquences d'assemblage possibles pour chaque famille de produit sont les suivantes :

Produit S :

Une seule opération est nécessaire pour obtenir produit_S :

- S_op1 réalisable sur les stations AFMA ou IBM pour les trois variantes possibles (cf. figure 4.4).

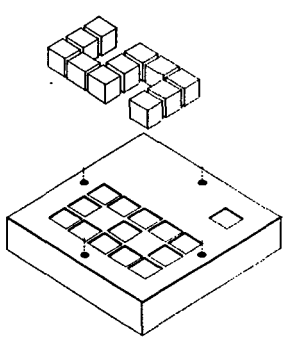
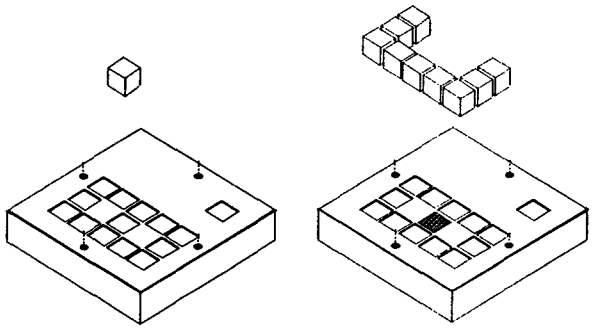


Figure 4.4 : opération S_op1.

Produit E :

Deux opérations sont nécessaires pour obtenir produit_E :

- E_op1 réalisable sur les stations AFMA ou MANU pour les trois variantes possibles (cf. figure 4.5.a).
- E_op2 réalisable sur les stations IBM pour les trois variantes possibles (cf. figure 4.5.b).



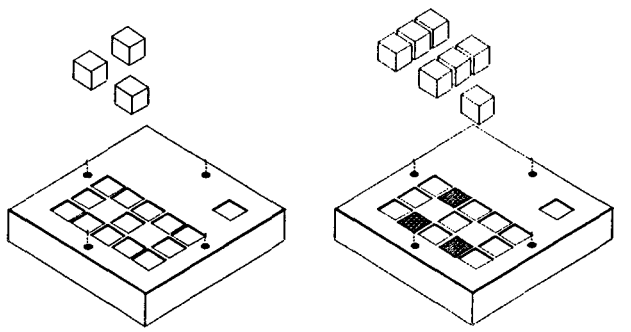
- a - - b -

Figure 4.5 : opérations a) E_op1 et b) E_op2.

Produit P :

Deux opérations sont nécessaires pour obtenir produit_P :

- P_op1 réalisable sur les stations AFMA ou MANU pour les trois variantes possibles (cf. figure 4.6.a).
- P_op2 réalisable sur les stations IBM pour les trois variantes possibles (cf. figure 4.6.b).



- a - - b -

Figure 4.6 : opérations a) P_op1 et b) P_op2.

Un contrôle de qualité est réalisé au niveau de la station IBM : lorsqu'un produit défectueux est détecté, il doit alors être orienté vers la station manuelle pour être soit retouché, soit rebuté.

4.4 Fonctionnement général

L'agent chargé de l'exploitation de la ligne reçoit des ordres de fabrication de la forme : "réaliser n lots de pièces de type t à livrer pour xx heures, yy minutes".

A cet effet, il positionne des plateaux de constituants non montés dans le magasin central, ces plateaux correspondant aux lots commandés. Il renseigne ensuite le superviseur d'atelier de la présence de ces plateaux, puis lance la production. Il n'a plus alors qu'à vérifier le bon fonctionnement des automatismes.

Le superviseur d'atelier calcule, en fonction des ordres de fabrication, un ordonnancement des tâches à réaliser au niveau atelier. Suivant l'ordre établi, les plateaux sont pilotés vers la cellule d'assemblage (pilotage atelier) où ils seront rangés dans le stockeur rotatif pendant leur traitement.

Le superviseur de cellule détermine alors l'ordre dans lequel les plateaux seront vidés de leurs pièces sur les palettes. Une fois celles-ci chargées, il s'agit de les piloter de station en station pour réaliser les gammes (pilotage cellule).

Les stations sont gérées par des automates industriels qui réalisent le séquencement des actions élémentaires correspondant au diverses opérations réalisables sur la station.

L'assemblage achevé, les pièces sont rapatriées vers les plateaux. Lorsqu'un plateau est complet, il est piloté vers le magasin central (pilotage atelier) où il peut être déchargé manuellement. La production s'interrompt lorsque l'ensemble des ordres de fabrication lancés a été traité.

4.5 Intérêt de l'exemple

Conçu pour permettre l'étude des architectures de commande dans le domaine des systèmes flexibles d'assemblage, le site présente l'ensemble des particularités essentielles caractérisant les SFPM envisagés dans le cadre du projet SCOPES (cf. § 1.4) :

- flexibilité des moyens de production : un même moyen peut réaliser plusieurs opérations et une même opération peut être réalisée par plusieurs moyens différents ;
- flexibilité de routage : la trajectoire des produits dans l'atelier n'est pas fixée a priori ;
- ressources partagées : dans le cas d'équipements flexibles, plusieurs opérations réalisables par une même ressource sont simultanément déclenchables ;
- permutation de l'ordre des opérations sur un produit : les séquences d'assemblage ne sont pas complètement ordonnées ;
- flexibilité de production : plusieurs variantes d'une même famille et/ou plusieurs produits de familles différentes peuvent être simultanément réalisés.

Des ateliers présentant une, voire plusieurs, de ces caractéristiques commencent à apparaître dans l'industrie. A ce titre, l'exemple du site est significatif. Cependant, la flexibilité obtenue n'est bien souvent employée que pour répondre à un problème spécifique et local. Bien que présente, elle est limitée dans les faits par le système de commande incapable de réaliser une prise de décision optimale par rapport aux capacités de l'atelier. Les architectures de commande classiques trouvent ainsi leur limite dans le cas des ateliers présentant les caractéristiques précédentes.

Les travaux menés au sein du Laboratoire de Productique de Schneider Electric, notamment dans le cadre du projet SCOPES, ont pour but de répondre à ce problème en proposant une architecture de système de commande permettant d'exploiter les flexibilités apparaissant sur les lignes de production actuelles. Le site productique, présentant la totalité des caractéristiques des systèmes flexibles, constitue un "banc de test" idéal pour valider les concepts proposés.

Partie 2

Analyse et modélisations

Chapitre 5

Analyse des éléments d'un SFPM

Introduction

Ce chapitre a pour but de définir un certain nombre de concepts et définitions relatifs aux divers "acteurs" d'un système de production. Ces derniers sont regroupés en trois classes : les produits, les lieux et les moyens.

Nous nous intéressons ici à chacune des classes prise isolément en essayant de proposer pour chaque élément une démarche d'analyse systématique, applicable à n'importe quel cas de figure.

Concernant les produits, nous présentons la notion de gamme logique et mentionnons seulement celle de gamme opératoire. Cette dernière notion, en reliant produits, lieux et moyens n'est pas à notre sens caractéristique de "l'acteur" produit. Elle s'apparente à un "scénario" permettant de spécifier le déroulement de la production. Ce point fera l'objet du chapitre suivant.

La notion d'opération apparaît dans chaque étude. Nous avons donc reporté les concepts associés dans un paragraphe particulier.

5.1 Les produits

5.1.1 Définitions

Dans le vocabulaire de la gestion de production, le terme *produit* désigne tout objet de transaction commerciale entre une entreprise et ses clients. Ainsi, outre les biens matériels fabriqués par l'entreprise, ce terme peut recouvrir également des prestations de service ou des biens immatériels tels que des

logiciels. Nous considérons ici les produits en tant que biens matériels. Fabriquer à moindre coût des produits conformes à leurs spécifications constitue la raison d'être du système de production.

Conformément à cette définition, un objet physique non commercialisé n'entre pas dans la catégorie des produits. Une autre notion existe en gestion de production : l'*article*. Ce terme, qu'il convient de bien distinguer de celui de produit, s'applique à tout élément physique distinct **géré** par l'entreprise, que cet élément soit fabriqué, sous-traité ou acheté : produit fini, ensemble, sous-ensemble, pièce, composant ou matière première. Chaque article possède un identifiant qui en permet la gestion.

Un produit est donc un article qu'une entreprise décide de commercialiser [AFNOR, 1991].

Par abus de langage, le terme produit désignera indifféremment les notions de produit et d'article telles que nous venons de les définir.

Un produit est obtenu par la réalisation d'un certain nombre d'*opérations caractéristiques*.

5.1.2 Opération caractéristique

Une opération sera dite caractéristique lorsqu'elle permet d'augmenter le coût du produit. Conformément à l'approche présentée dans [Verdebout, 1993], la notion de coût, quantifiable, sera préférée à celle de valeur ajoutée (utilisée dans [Telemecanique, 1993] pour définir la notion d'*opération directe*) non quantifiable.

En remarquant que les seules opérations permettant d'ajouter un coût sont celles réalisant une *transformation fonctionnelle*, nous retrouvons la définition adoptée dans le projet CASPAIM [Amar *et al.*, 1992] [Ausfelder, 1994] : une opération est dite caractéristique si elle réalise une transformation fonctionnelle sur le produit, c'est-à-dire si elle n'a pas comme unique objectif un transfert ou un positionnement.

Les opérations caractéristiques peuvent être classées selon le nombre de constituants intervenant dans l'opération : opérations unaires ou traitements (contrôle[1], usinage, ...) et les opérations n-aires ou assemblages (insertion, collage, agrégation, ...).

Les opérations de transfert n'entraînent que des dépenses sans apport de coût. Réduire ce type d'opérations est, pour cette raison, l'un des critères que cherche à optimiser le bureau des méthodes pour concevoir l'organisation du système de fabrication.

A l'issue d'une opération, le *stade de fabrication* et éventuellement l'*état de conformité* du produit évoluent.

---

[1] Les opérations de contrôle augmentent le coût du produit en améliorant la qualité de ce dernier.

5.1.3 Etat d'un produit

Pour un produit (article) en cours de fabrication, nous appellerons *stade de fabrication* ou *état d'avancement* du produit (article) la liste des opérations que ce produit (article) a déjà subies. Une définition duale est obtenue en considérant la liste des opérations restant à effectuer. La notion d'*état d'un produit* couramment rencontrée dans la littérature [Henrioud, 1989] [Cruette, 1991] [Ausfelder, 1994] correspond à cette définition.

L'*état de conformité* d'un produit qualifiera, dans ce travail, le degré de conformité du produit par rapport aux critères de qualité attachés au stade de fabrication courant. Ces critères font partie du dossier de définition du produit. Les états extrêmes d'un produit à un stade donné sont l'état bon ou conforme et l'état mauvais ou non conforme.

L'ordre dans lequel doivent s'effectuer les opérations sur les produits n'est pas quelconque. Il doit notamment respecter les contraintes d'assemblage des divers constituants. A ces contraintes disponibles dès l'issue de la phase de conception, s'ajoutent d'autres contraintes liées aux procédés de fabrication adoptés par le bureau des méthodes.

Du point de vue mathématique, l'ensemble des opérations caractéristiques à effectuer sur un produit est un ensemble partiellement ordonné, i.e. l'ordre de certaines opérations est indifférent.

Le problème se pose maintenant de représenter les diverses possibilités d'enchaînement des opérations.

5.1.4 Notion de gamme

La notion de *gamme*, conformément aux définitions couramment admises [AFNOR, 1991] [Telemecanique, 1993], ne permet de représenter qu'une suite **linéaire** figée d'opérations. Une gamme représentant une séquence d'opérations caractéristiques sera appelée *gamme logique*. Cette gamme sera déclinée en *gammes opératoires* lorsque les opérations correspondront à des opérations réelles réalisées par les moyens du système de production (cf. chapitre 6).

Plusieurs gammes logiques peuvent être technologiquement possibles pour un même produit. La détermination de l'ensemble des gammes admissibles pour un produit donné n'entre pas dans le cadre de notre travail. Dans les cas simples, tels les produits du site, ce travail s'effectue naturellement. Mais une approche méthodologique rigoureuse peut être nécessaire dans le cas d'usinage ou d'assemblage complexes [Bourjault, 1984] [Henrioud, 1989] [Henrioud et Bourjault, 1990].

Les gammes opératoires sont établies lors de l'industrialisation par réduction de la flexibilité sur l'ordre des opérations. Afin de faciliter cette phase ou dans le but de conserver cette flexibilité, une représentation concise de l'**ensemble** des gammes admissibles a été recherchée par certains laboratoires. De nombreux formalismes ont ainsi été développés : séquences fonctionnelles d'assemblage [Bourjault, 1984], arbres fonctionnels d'assemblage

[Henrioud, 1989], graphes de NOAH [Sacerdoti, 1977], Réseaux de Petri [Bourjault *et al.*, 1987]. C'est ce dernier formalisme qui a été choisi dans le projet CASPAIM à l'origine de nos travaux [Cruette, 1991] [Ausfelder, 1994] [Huvenoit, 1994].

5.1.5 Représentation des gammes

Une gamme, et plus généralement un ensemble de gammes, peuvent être représentés par un graphe. Un graphe est construit à partir d'un ensemble de noeuds et d'arcs.

Lorsque l'on n'utilise qu'une classe de noeuds, deux points de vue complémentaires sont possibles [Henrioud, 1989] : associer les noeuds aux opérations ou aux stades (cf. figure 5.1).

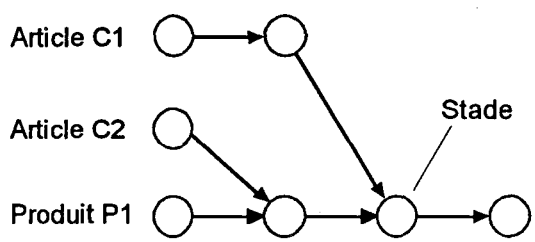


Figure 5.1 : modèle par stades des gammes.

Dans le cas de la définition par les opérations, il est courant de représenter chaque opération par un symbole graphique correspondant au type de l'opération réalisé (assemblage, transfert, contrôle ...). Cependant, aucune norme internationale ne définit une telle représentation. De nombreux classements et formalismes ont été élaborés, notamment en Allemagne [VDI 2860, 1982], au Japon [Yokoyama, 1988], et chez certains industriels [Telemecanique, 1989] ou laboratoires universitaires [Verdebout, 1993]. Les symboles utilisés sur la figure 5.2 sont tirés de ce dernier travail.

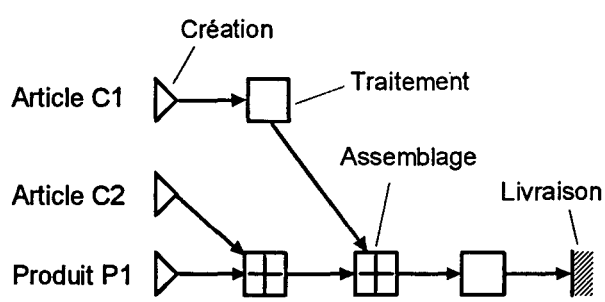


Figure 5.2 : modèle par opérations des gammes.

L'utilisation d'un graphe biparti [Berge, 1987] offre l'avantage de pouvoir marier les deux points de vue. En effet, deux classes de noeuds étant disponibles, il devient possible de représenter à la fois les opérations et les

stades par un type différent. C'est notamment le cas lorsqu'on opte pour les RdP (cf. annexe A).

Les gammes sont décrites à l'aide des RdP de la manière suivante : les opérations sont représentées par les transitions et les différents stades des produits par les places. Les jetons représentent alors logiquement les produits (cf. figure 5.3).

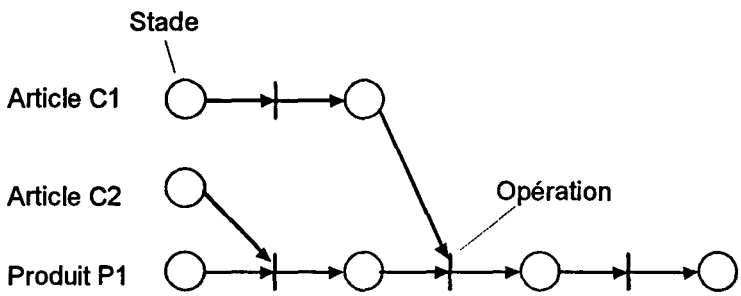


Figure 5.3 : Modèle RdP des gammes.

Bien que très rarement utilisée dans la pratique – la plupart des systèmes informatiques de gestion de la production étant bâti sur la notion **linéaire** de gammes – la flexibilité sur l'ordre des opérations peut être modélisée sur les graphes de précédence. La figure 5.4 illustre un ordre total entre op1 et {op2, op3} et un ordre partiel entre op2 et op3.

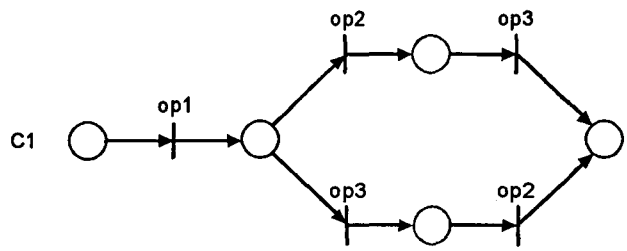


Figure 5.4 : Ordre total et partiel entre opérations.

Application aux produits du site TE

La figure 5.5 représente les gammes logiques des produits du site en utilisant la modélisation par RdP. Les gammes opératoires seront, quant à elles, illustrées dans le chapitre 8, à l'aide de la modélisation par opération utilisée dans le cadre du pilotage.

Remarque : lorsqu'une opération d'assemblage met en oeuvre des composants élémentaires ne subissant aucun traitement préparatoire (tel les cubes dans notre exemple), il est possible de ne pas représenter les stades correspondant aux composants, les contraintes introduites n'étant pas significatives (cf. produit_S).

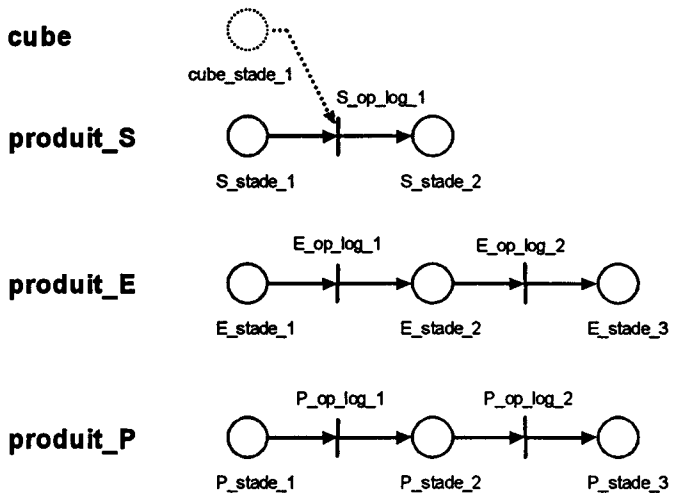


Figure 5.5 : gammes logiques des produits du site DRD.

5.1.6 Regroupement en famille et notion de variante.

Une *famille* de produits (ou d'articles) est définie comme étant un ensemble homogène de produits au sens de leur conception et des moyens mis en oeuvre pour les fabriquer. A chaque famille correspond un ensemble de gammes et un ensemble de paramètres appelés *variabilités* à instancier avant tout lancement en production. Les valeurs prises par les variabilités sont utilisées lors de la fabrication pour paramétrer certaines opérations, par exemple le choix de la couleur au niveau d'une opération de peinture.

Une **instanciation donnée des variabilités** définit une *instance* de la famille en question. Lorsque chaque variabilité prend ses valeurs dans un ensemble fini, il est possible d'énumérer les instanciations possibles : on obtient la notion classique de *variante*. Cette énumération, fortement combinatoire pour les produits définis par plusieurs variabilités, conduit le plus souvent à limiter le nombre de variantes disponibles en interdisant certaines combinaisons de valeurs des variabilités.

Lorsqu'il n'est pas possible d'énumérer l'ensemble des combinaisons utiles, la notion de variante apparaît insuffisante. Nous avons donc été amenés à introduire la notion de *jeu de variantes*, définissant un ensemble de variantes à partir d'un **ensemble de conditions sur les variabilités**. De cette manière, un produit peut correspondre à plusieurs jeux de variantes, si les valeurs de ses variabilités satisfont plusieurs jeux de conditions.

Sur l'exemple du site DRD, il a été arbitrairement défini quatre familles de produits (correspondant à S, E, P et SE). Chaque famille possède une seule variabilité : la couleur des cubes à insérer. Cette variabilité ne pouvant prendre que trois valeurs (rouge, vert et noir), l'ensemble des variantes est dénombrable. Variantes et jeu de variante sont donc ici confondus. La définition de chaque jeu de variante est la suivante :

- variante_rouge : (couleur = rouge) ;
- variante_vert : (couleur = vert) ;

♦ variante_noir : (couleur = noir) ;

Les conditions sont ici très simples. Nous donnons au paragraphe 8.1.2 un exemple plus complexe.

5.2 Les lieux

L'analyse et l'identification des divers lieux physiques d'un système de production s'effectuent en trois étapes. La première étape consiste à décrire de façon hiérarchique l'organisation des divers lieux de transit de l'atelier (cf. § 5.2.1). La seconde étape permet de spécifier le type et le nombre d'éléments mobiles pouvant occuper à un instant donné chacun des lieux (cf. § 5.2.2). Enfin, la troisième étape décrit les possibilités de transfert entre lieux (cf. § 5.2.3).

5.2.1 Décomposition structurelle

Cette décomposition s'effectue par niveaux successifs, chaque niveau consistant à détailler les éléments structurels identifiés au niveau précédent. Le niveau 1 est constitué de l'ensemble de l'atelier.

Description d'un niveau

Un élément d'un niveau n de la description se compose ainsi d'éléments de niveau n+1 (niveau de détail croissant) reliés entre eux par un ou plusieurs *systèmes de transfert* et d'une ou plusieurs zones de stockage ou *buffers* servant d'interface avec le niveau précédent. Deux ou trois niveaux suffisent généralement pour décrire la structure d'un système de production. La figure 5.6, inspirée de [Aubert-Barbier, 1990], illustre le principe de cette hiérarchie.

Nous donnons ci-après quelques exemples de désignation des niveaux (en caractères gras, celles adoptées dans le projet SCOPES) :

- **atelier**
- (îlot: EPFL)
- **cellule**, installation, ligne
- (îlot: ECLille)
- **station**, poste, machine

↓ niveau de détails croissant

Ces termes, fréquemment rencontrés dans la littérature, recouvrent de multiples définitions, parfois contradictoires. Cela est dû à l'extrême variétés des architectures possibles en ce domaine (cf. chapitre 1). C'est pourquoi nous préférons, tout en fixant un cadre descriptif précis, laisser à chacun le choix de l'appellation conforme à son expérience.

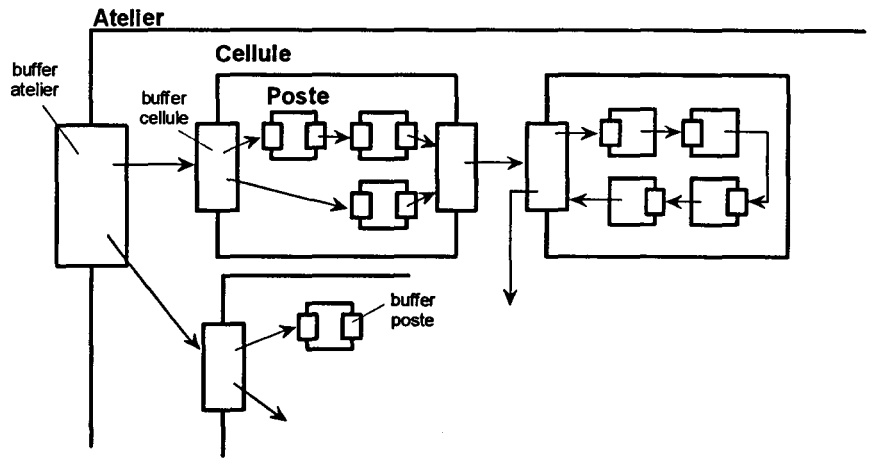


Figure 5.6 : décomposition hiérarchique d'un système de production.

Les buffers de stockage inter-niveaux ont un rôle essentiel : ils permettent de découpler les divers flux existant dans les systèmes et de régler les rendez-vous au niveau des postes d'assemblage.

Description d'un système de transfert

Un système de transfert peut être décomposé en *plots* reliés par des *zones de transfert* [Ausfelder, 1994].

Par définition, seuls les éléments mobiles situés sur les plots sont observables (capteurs) et donc commandables (blocage, orientation, etc.). Les plots interviennent notamment au niveau des aiguillages chargés de l'orientation des pièces et au niveau des indexeurs chargés d'immobiliser les pièces pendant les opérations effectués sur chaque poste (cf. figure 5.8). Les éléments situés sur les zones de transfert ne peuvent être commandés. Ils évoluent de manière autonome jusqu'au plot suivant où ils seront réorientés.

Application au site DRD

L'application de la démarche à l'exemple du site DRD est donnée sur la figure 5.7.

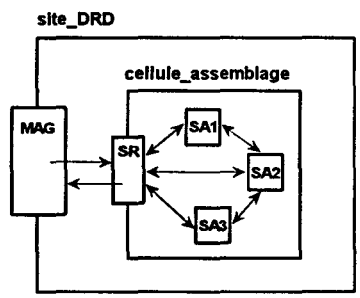


Figure 5.7 : description structurelle du site DRD.

L'atelier `site_DRD` (niveau 1) se compose d'un magasin central `MAG` et d'une cellule `cellule_assemblage` reliés par le réseau de transfert `réseau_nacelles`. La cellule (niveau 2) se décompose elle même en trois

stations (SA1, SA2 et SA3) et un magasin local (SR) reliés par le système de convoyage réseaux_palettes. Le découpage en plots et zones de transfert pour ce dernier système est donné sur la figure 5.8 pour une dérivation.

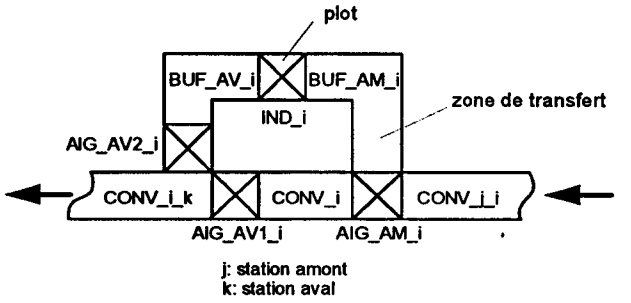


Figure 5.8 : décomposition du convoyeur à palettes du site DRD.

Une autre représentation, sous forme arborescente permettant de mettre en évidence les divers niveaux, est donnée sur la figure 5.9.

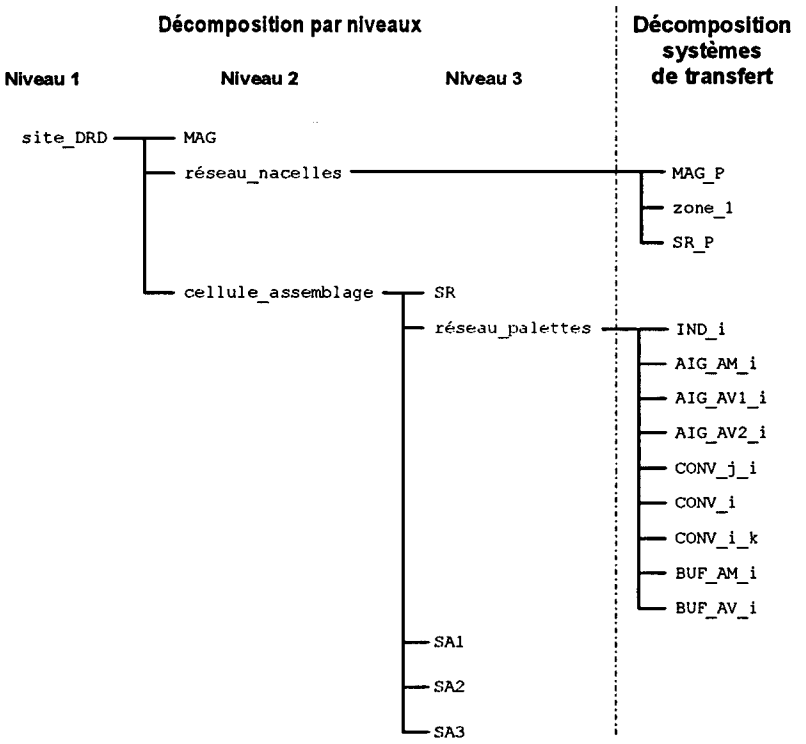


Figure 5.9 : description arborescente du site DRD.

5.2.2 Relations de composition

Les divers lieux du système, à savoir les éléments de type "buffer", "plots" et "zones de transfert" identifiés lors de l'analyse structurelle (ce sont en fait les feuilles de l'arbre issu de la description structurelle) sont composés d'un certain

nombre d'emplacements pouvant accepter les éléments mobiles circulant dans l'atelier (chariots, palettes, produits, ...). Un emplacement de stockage (ES par la suite) ne peut se trouver que dans deux états : libre ou alloué à un élément mobile de l'atelier. L'ensemble des états des ES d'un lieu constitue l'état de ce lieu.

Plusieurs types d'ES peuvent généralement être isolés dans l'atelier. Un type d'ES est défini par l'énumération des types d'objets mobiles qu'il peut recevoir. Pour l'exemple du site DRD :

```
ES_produit = {produit_S, produit_E, produit_P}
ES_plateau = {plateau}
ES_palette = {palette}
ES_nacelle = {nacelle}
```

Certaines ressources mobiles peuvent elles-mêmes être support d'autres ressources mobiles. C'est le cas notamment si les produits se déplacent dans l'atelier sur des palettes ou des chariots. Ces ressources doivent alors également être décrites en terme d'ES.

Dans le cas du site DRD, trois types de supports mobiles apparaissent : les *nacelles* circulant sur le convoyeur aérien et pouvant acheminer chacune un plateau ; les *palettes* circulant sur le convoyeur sur table et offrant chacune un emplacement pour un produit quelconque; enfin les *plateaux* pouvant recevoir 6 produits identiques.

La description en terme d'ES des éléments du site est donnée sur les figures 5.10 et 5.11.

MAG	20 ES_plateau
MAG_P	1 ES_nacelle
SR_P	1 ES_nacelle
canton_1	1 ES_nacelle
SR	20 ES_plateau
SCDF_P	1 ES_palette
IND_1	1 ES_palette
AIG_AM_1	1 ES_palette
AIG_AV1_1	1 ES_palette
AIG_AV2_1	1 ES_palette
BUF_AM_1	3 ES_palette
BUF_AV_1	3 ES_palette
CONV_4_1	2 ES_palette
CONV_1	3 ES_palette
CONV_1_2	5 ES_palette
etc.	etc.

Figure 5.11 : composition des éléments fixes du site DRD.

nacelle	1	ES_plateau
plateau	6	ES_produit
palette	1	ES_produit

Figure 5.10 : composition des supports mobiles du site DRD.

5.2.3 Relations d'accessibilité

Les relations d'accessibilité [Cruette, 1991] traduisent les possibilités de transfert entre les différents emplacements de stockage (ES) identifiés dans les relations de composition. Pour l'exemple du site il s'agit, par exemple, d'exprimer le fait que l'indexeur IND_1 peut échanger des palettes avec la zone de transfert BUF_AV_1.

Les relations d'accessibilité pour le site DRD sont représentées sur la figure 5.12 (cf. page suivante). On obtient quatre graphes distincts mettant ainsi en évidence quatre type de flux matériels dans l'atelier :

- flux plateaux
- flux nacelles
- flux palettes
- flux produits

Les flèches pointées reliant deux lieux expriment le fait que tous les ES du premier lieu peuvent échanger des objets avec tous les ES du second lieu.

Un seul flux produit apparaît dans notre cas : cela provient du fait qu'il existe dans l'atelier des ES pouvant recevoir plusieurs types de produits (palettes et plateaux "universels").

5.3 Les moyens

L'identification des moyens de production, faute de définition précise, s'avère délicate. Le terme "moyen de production" recouvre en effet des notions extrêmement diversifiées. Toute installation, machine, poste, outil, logiciel ou appareil de mesure destinés à la production et/ou au contrôle peut être qualifié de moyen.

Face à cette imprécision, divers travaux ont été menés, aboutissant à des analyses récurrentes de la notion de moyen. Citons, entre autres, le découpage fonctionnel proposé par la norme NF X 60-200, l'analyse structurelle arborescente de D. Cruette [Cruette, 1991] ou encore l'analyse structuro-fonctionnelle développée par S. Amar [Amar *et al.*, 1992].

Nous proposons, pour notre part, une décomposition en deux niveaux.

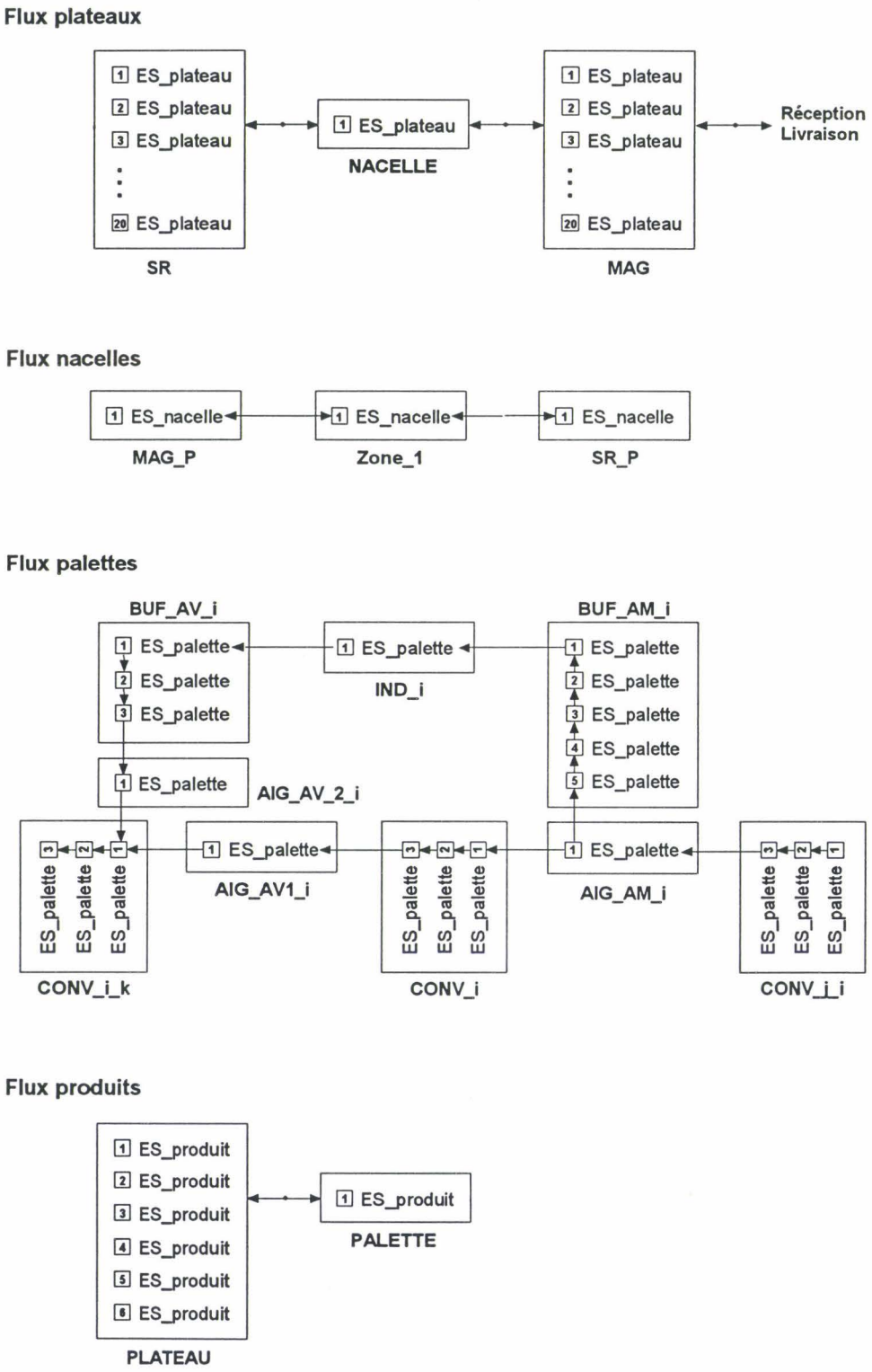


Figure 5.12 : relations d'accessibilité pour le site DRD.

5.3.1 Décomposition structurelle

La décomposition structurelle s'appuie sur la décomposition des lieux présentée au paragraphe 5.2 : nous qualifierons de *moyens structurels* les éléments identifiés lors de la description d'un niveau, à savoir les niveaux constitutifs "supérieurs", les buffers d'entrée/sortie vers le niveau "inférieur" et les systèmes de transfert entre ces éléments.

Nous obtenons ainsi une description récursive, un moyen d'un niveau de description donné pouvant lui-même être décomposé en moyens appartenant à un niveau de description supérieur. Ainsi, le moyen "atelier" correspondant au site TE est-il décomposable en trois moyens : le magasin de niveau atelier (MAG), le système de transfert par nacelles et la cellule d'assemblage. Cette dernière est à son tour décomposable en cinq moyens : les trois stations (SA1, SA2 et SA3), le stockeur rotatif (SR) et le système de convoyage par palettes.

Trois classes de moyens, élémentaires par rapport au critère de structure peuvent être distinguées : les niveaux terminaux (i.e. non décomposables), les systèmes de transport et les buffers de chaque niveau non terminal.

5.3.2 Décomposition fonctionnelle

La décomposition fonctionnelle permet de décomposer chacun des moyens structurels élémentaires en *systèmes opérants* réalisant chacun une, voire plusieurs, fonctions opératives particulières.

Deux niveaux de décomposition fonctionnelle peuvent être distingués : le premier niveau permet de mettre en évidence des systèmes physiquement indépendants (exemple : tour + robot manipulateur), décomposés dans le second niveau en sous-systèmes liés entre eux (exemple : mandrin, traînard, contre-pointe, ... d'un tour). Ce dernier niveau est récurrent, un sous-système pouvant lui-même être constitué d'autres sous-systèmes.

L'identification des systèmes fonctionnels pourra être facilitée par l'utilisation d'une bibliothèque de fonctions organisée en classes et sous-classes telle celle présentée dans [Amar *et al.*, 1992].

Le découpage fonctionnel s'applique a "tout ce qui a un moteur", ou en terme de commande "à tout ce qui est commandable". Cette règle n'est cependant pas absolue. En effet une fonction opérative peut avoir deux facettes par rapport à la commande. Elle peut être soit passive, c'est-à-dire être tributaire d'une commande sur un actionneur, soit active. Tel est le cas, notamment, de la fonction stockage dont nous reparlerons plus loin.

Un système opérant peut réaliser plusieurs fonctions. C'est le cas de systèmes flexibles tels qu'un robot ou un opérateur humain.

L'appartenance d'un système à plusieurs moyens structurels, si elle est théoriquement possible (opérateur desservant deux stations), n'est pratiquement réalisée que dans des cas particuliers ou dans les phases de réglage (régleur se partageant entre plusieurs stations). Ces problèmes d'affectation de ressources

partagées ne seront pas traités dans le cadre de ce travail. Dans certains cas, notamment lorsqu'il y a intervention humaine, il est d'ailleurs préférable de laisser ce choix ouvert.

Nous donnons ci-après quelques exemples de décomposition fonctionnelle à partir des moyens du site DRD.

- **décomposition du buffer SR de la cellule d'assemblage** : trois systèmes opérants composent le stockeur SR :
  - un stockeur rotatif ;
  - un bras manipulateur à cycle fixe en U assurant le transfert des plateaux entre le système de transfert `réseau_nacelles` et le stockeur ;
  - un bras manipulateur à cycle fixe en U assurant le transfert des produits entre le système de transfert `réseau_palettes` et le stockeur.
- **décomposition de la station IBM** : un seul système opérant compose la station IBM :
  - un robot 6 axes de marque IBM.
- **décomposition du système de convoyage par palettes** : trois types de systèmes opérants sont rencontrés lors de la décomposition d'un système de transfert :
  - des aiguillages (convergents ou divergents) réalisant une fonction d'orientation ;
  - des portions de convoyeur sur table, de rails aériens, de chaînes enterrées ... réalisant une fonction de transfert ;
  - des indexeurs réalisant des fonctions de blocage et positionnement.

5.3.3 Comportement d'une ressource

Dans la suite de ce mémoire, nous emploierons le terme générique de *ressource active* pour désigner à la fois les moyens structurels (cf. § 5.3.1) et les systèmes opérants (cf. § 5.3.2).

Chaque ressource active de l'atelier est capable d'effectuer un certain nombre d'opérations. A l'issue d'une opération l'état d'une ou plusieurs ressources évolue, interdisant ou autorisant de nouvelles opérations. Nous définirons l'état d'une ressource à un instant donné par l'ensemble des opérations que cette ressource est capable d'effectuer à cet instant.

Nous utiliserons des graphes pour représenter l'enchaînement des états accessibles par chaque ressource. Un tel graphe associé à une ressource spécifie le comportement de la ressource. Des exemples de graphes comportementaux, utilisant le formalisme des RdP, seront donnés dans le chapitre 9.

5.4 Les opérations

5.4.1 Opérations réelles et opérations logiques

Nous désignerons par *opération réelle* toute séquence d'actions réalisées par une ressource active.

Une opération réelle peut donc être réalisée par un moyen appartenant aux quatre types de moyens structurels :

- moyen de niveau non terminal
- moyen de niveau terminal
- buffer
- système de transfert

ou par un système opérant constitutif d'un moyen.

Les opérations réelles sont gérées par les divers niveaux du système de commande. Nous reviendrons en détail sur ce point au chapitre 7 traitant de l'architecture de commande.

Lorsque nous ferons référence aux opérations indépendamment des moyens chargés de les réaliser, nous utiliserons le terme d'*opération logique*. Cette notion permet de décrire un système de production sous son aspect fonctionnel, indépendamment de l'aspect opérationnel.

L'ensemble des opérations logiques d'un système de production se déduit de l'analyse des produits et des lieux (cf. § 5.1 et § 5.2). On dégage ainsi deux classes d'opérations logiques correspondant aux fonctions logistiques et transformationnelles du système :

- les opérations de transfert,
- les opérations caractéristiques.

Notons ici que la fonction de stockage ne saurait être associée à aucune opération. En effet, contrairement aux fonctions précédentes, cette fonction comprend un aspect statique, non compatible avec l'aspect dynamique associé à la notion d'opération. Le stockage correspond toujours à l'attente des conditions de réalisation de l'opération suivante sur le produit. La durée d'un stockage est donc indéterminée, contrairement à celle d'une opération[2]. Par contre, il est parfaitement justifié d'assimiler l'attente à une opération lorsque la durée en est donnée. Il s'agit alors d'une opération de traitement nécessaire technologiquement à l'obtention du produit (séchage après lavage par exemple).

[2] La durée d'une opération peut être connue de façon exacte (opérations réalisées par les systèmes opérants) ou approximative (opérations réalisées par les moyens structurels).

5.4.2 Modalité opératoire et affectation

Nous appellerons *modalité opératoire* d'une opération logique toute opération réelle concrétisant l'opération logique. Une même opération réelle peut concrétiser plusieurs opérations logiques différentes et une même opération logique peut avoir plusieurs modalités de réalisation différentes. Certaines opérations réelles ne réalisent aucune opération logique ; ce sont les opérations de préparation propres aux moyens (positionnement, prise d'outil, chargement de programmes, fermeture d'un sas de sécurité ...).

Nous appellerons *affectation* d'une opération réelle la liste des moyens capables de réaliser cette opération. Par définition, cette liste n'est jamais vide.

Sur la figure 5.13, l'opération réelle 1, réalisée par les moyens 1 et 2 concrétise les opérations logiques 1 et 3. L'opération réelle 3 effectuée par le moyen 2 n'est associée à aucune opération logique.

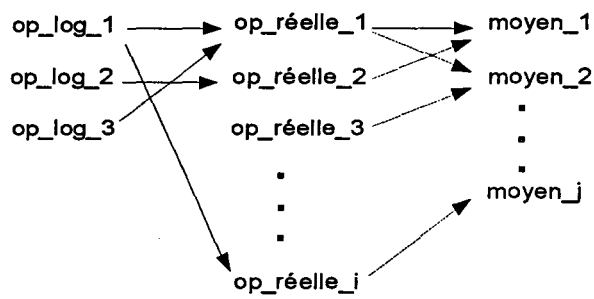


Figure 5.13 : relations entre opérations logiques, opérations réelles et moyens.

5.4.3 Application au site DRD

L'ensemble des opérations logiques de transfert modifiant l'état de chaque type d'ES se déduit simplement des relations d'accessibilité (cf. § 5.2.3).

Nous désignerons par la notation lieu_initial→lieu_final l'opération logique de transfert entre deux localisations.

Par exemple, pour le site DRD, MAG→nacelle désigne l'opération de transfert d'un plateau du magasin MAG vers la nacelle du convoyeur aérien. On désignera par []→MAG le transfert correspondant à l'approvisionnement de l'atelier en plateaux bruts, et par MAG→[] celui correspondant à la livraison des plateaux finis.

De même, l'ensemble des opérations logiques caractéristiques modifiant l'état des produits est donné par les gammes logiques (cf. § 5.1.5).

Modalités opératoires et affectations sont données sur la figure 5.14 (nous n'avons pas détaillé les modalités opératoires des opérations de transfert internes au système de transfert par nacelles).

Op. réelle	Affectations	Op. Logique
MAG_nac	MAG	MAG→nacelle
nac_MAG	MAG	nacelle→MAG
chg_MAG	MAG	[]→MAG
dechg_MAG	MAG	MAG→[]
nac_cell	cellule_assemblage	nacelle→cellule
cell_nac	cellule_assemblage	cellule→nacelle
produire_S	cellule_assemblage	S_op_log_1
produire_E	cellule_assemblage	E_op_log_1 E_op_log_2
...	...	...

a- niveau atelier.

Op. réelle	Affectations	Op. Logique
S_op_1	IBM AFMA	S_op_log_1
E_op_2	IBM	E_op_log_2
P_op_2	IBM	P_op_log_2
P_op_1	AFMA	P_op_log_1
E_op_1	MANU AFMA	E_op_log_1
SR_pal	SR	plateau→palette
pal_SR	SR	palette→plateau
P_rep_1	MANU	
P_rep_2	MANU	
...	...	...

b- niveau cellule d'assemblage.

Op. réelle	Affectations	Op. Logique
entrer_station_i	AIG_AM_i	AIG_AM_i→BUF_AM_i
indexer_i	IND_i	BUF_AM_i→IND_i
désindexer_i	IND_i	IND_i→BUF_AV_i
sortir_station_i	AIG_AV_i	AIG_AV2_i→CONV_i_k
...	...	...

c- niveau interne au convoyeur.

Figure 5.14 : modalités opératoires et affectations pour le site DRD.

Chapitre 6

Processus de fabrication d'un produit

Introduction

Le chapitre précédent a permis de mettre en évidence et de caractériser les différents éléments d'un système de production considérés isolément : des moyens effectuent un certain nombre d'opérations réelles qui permettent de modifier l'état des produits et/ou des lieux. Ces opérations doivent cependant être réalisées dans un certain ordre afin d'amener les produits au stade final souhaité.

La spécification de ce séquencement est donnée par le processus de fabrication associé au produit. Dans la pratique, l'établissement du processus de fabrication va de pair avec le choix des procédés et des moyens. Toute installation est, en effet, conçue autour d'une famille de produits dont les spécifications préexistent (gammes logiques). Les opérations réelles et les moyens sont déduits logiquement de ces spécifications.

Ce chapitre a pour objectif de préciser la démarche aboutissant au processus de fabrication d'un produit.

Les concepts présentés sont tirés de la méthodologie de développement mise en oeuvre à Telemecanique. Cette méthodologie est expliquée de façon rigoureuse et détaillée dans le SDÉ (Standard de Développement) introduit en 1988 [Telemecanique, 1993].

Bien qu'interne à Telemecanique, ce standard a été élaboré à partir de normes internationales et nationales traitant du même sujet, notamment les normes de la série ISO 9000 (9000 à 9004) concernant l'assurance qualité. La conformité à ces normes est en effet indispensable vis-à-vis des partenaires de l'entreprise susceptibles d'utiliser ou d'auditer les dossiers de développement (filiales, sous-traitants, grands prescripteurs, organismes de certification). La

démarche décrite est donc représentative des méthodes de développement existantes dans l'industrie.

6.1 Elaboration du processus de fabrication

Les diverses étapes de l'élaboration du processus de fabrication d'un produit sont illustrées sur la figure 6.1.

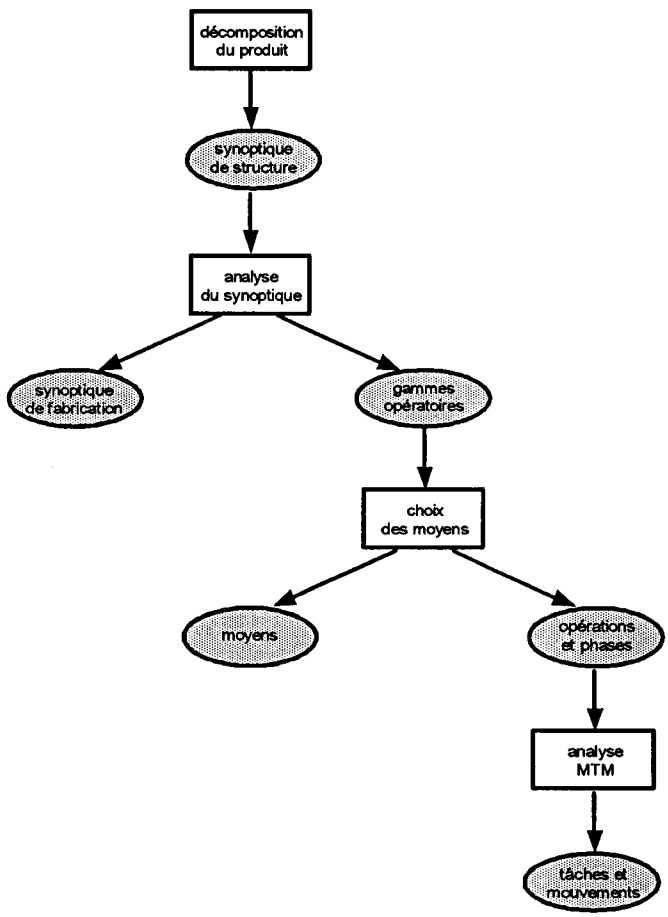


Figure 6.1 : les étapes de l'élaboration du processus de fabrication.

Quatre étapes peuvent être distinguées.

A partir des contraintes d'assemblage (déduites ou données dans le dossier de définition du produit), un choix de décomposition du produit est effectué (étape 1).

Le *synoptique de structure*, obtenu à l'issue de cette décomposition détermine de façon univoque les *gammes opératoires* et leur séquencement :

chaque sous-ensemble identifié lors de la décomposition est référencé en tant qu'article et se voit affecter une gamme (plus éventuellement une gamme de remplacement) ; par ailleurs, l'organisation arborescente du synoptique de structure impose l'ordonnancement des gammes spécifié dans le *synoptique de fabrication* (étape 2)

Le choix des moyens permet ensuite de décomposer les gammes en *phases*, puis les phases en *opérations* (étape 3).

Enfin, si besoin, une analyse fine des opérations (*analyse MTM*[1]) est pratiquée afin de mettre en évidence *tâches* et *mouvements élémentaires* (étape 4).

L'analyse descendante, à partir des données initiales, est employée lors de la phase de développement d'un produit. L'analyse ascendante, à partir des tâches ou des opérations, peut être pratiquée sur un processus existant, non conçu à partir de la méthodologie présentée.

La décomposition du produit est le fruit d'une collaboration entre les activités *conception* et *industrialisation* du processus de développement. Elle doit être soigneusement établie. En effet, elle conditionne le travail de plusieurs services de l'entreprise :

- Elle permet d'identifier les articles utilisés en gestion commerciale et gestion de production. Pour ce faire, chaque niveau de la décomposition doit coïncider à une rupture du processus de fabrication entraînant un état physique stable d'un sous-ensemble du produit. Chaque sous-ensemble dans un état stable peut ainsi faire l'objet d'une gestion et être identifié en tant qu'article. Un article peut être stocké, et/ou utilisé dans la composition d'un autre article, et/ou livré vers une autre unité de gestion ou vers l'extérieur de l'entreprise.
- Elle permet de définir les niveaux de gestion auxquels s'appliqueront ordonnancement et gestion des stocks (cf. chapitre 7).
- Elle constitue un guide lors de l'élaboration du processus de fabrication en fournissant les différentes étapes de fabrication du produit (synoptique de fabrication).

Les choix de décomposition d'un produit, ainsi que le découpage des gammes en phases, et des phases en opérations fait largement appel au "métier" du service industrialisation. Les décisions sont prises en tenant compte de différents critères tels que :

- choix entre intégration et sous-traitance totale ou partielle ;

---

[1] **Analyse MTM (Methods Time Measurement)** : effectuée par un "stémiste", décomposition qualitative et mesure de la durée des mouvements élémentaires à effectuer par un opérateur sur un poste correctement aménagé pour réaliser une opération ou une tâche donnée, suivant une table préétablie. Cette table, d'usage international, est diffusée en France par l'Association Française MTM. Le mouvement constitue le plus petit élément mesurable en analyse MTM ; l'unité de mesure est le DMH (dix millième d'heure) ou le CMH (cent millième d'heure).

- choix des moyens : spécifiques ou communs à plusieurs articles ou à plusieurs gammes ;
- équilibrage de la charge des moyens de production.

Le degré de flexibilité du système résulte de l'ensemble de ces choix. Nous discuterons au chapitre 8 des rapports entre flexibilité et processus de fabrication.

6.2 La décomposition arborescente des produits

Dans les systèmes d'assemblage, un produit (*composé*) est décomposable en sous-ensembles (niveaux 1, 2 ... n) d'articles (*constituants* ou *composants*). Cette décomposition peut être répétée pour chaque sous-ensemble constitutif. Elle s'arrête lorsqu'on arrive à un sous-ensemble simple considéré comme constituant élémentaire (le constituant élémentaire étant le constituant susceptible d'être acheté).

La liste des articles directement constitutifs d'un article est donnée par la *nomenclature* de l'article considéré. On représente classiquement les nomenclatures successives des articles constitutifs d'un produit sous forme d'un arbre (cf. figure 6.2).

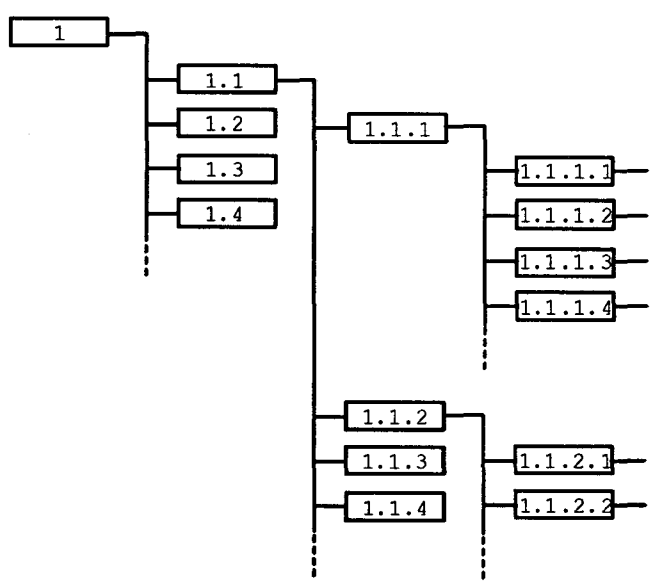


Figure 6.2 : structure arborescente d'un produit.

Cette structure permet de comprendre très rapidement la composition et l'articulation du produit "1". Chaque niveau n+2 comporte la totalité des constituants des éléments de niveau n+1, éléments devenant des constituants

des éléments de niveau n. Ainsi l'élément 1.1.2 est un constituant de l'élément 1.1 qui est lui-même constituant de l'élément 1.

La structure arborescente d'un produit comporte le nombre de niveaux nécessaires pour arriver à la décomposition la plus fine du produit. En général, trois ou quatre niveaux sont suffisants pour la compréhension d'un produit. Cette représentation permet ainsi d'identifier très facilement :

- tous les éléments constitutifs, à tous les niveaux de nomenclature, jusqu'aux constituants élémentaires ;
- les variantes possibles d'un produit ou d'un articles à partir des variantes des éléments constitutifs de niveau inférieur.

Bâtie à partir des contraintes d'assemblages (qui constituent un ensemble partiellement ordonné – cf. § 5.1.3), la décomposition d'un produit n'est unique que pour des cas particuliers : produits d'usinage, assemblages simples, assemblages fortement contraints. Le choix de la décomposition et donc des articles et nomenclatures associées constitue donc une réduction de la flexibilité liée à l'ordre des opérations.

La figure 6.3 illustre la décomposition d'un contacteur électrostatique (extrait limité aux constituants principaux).

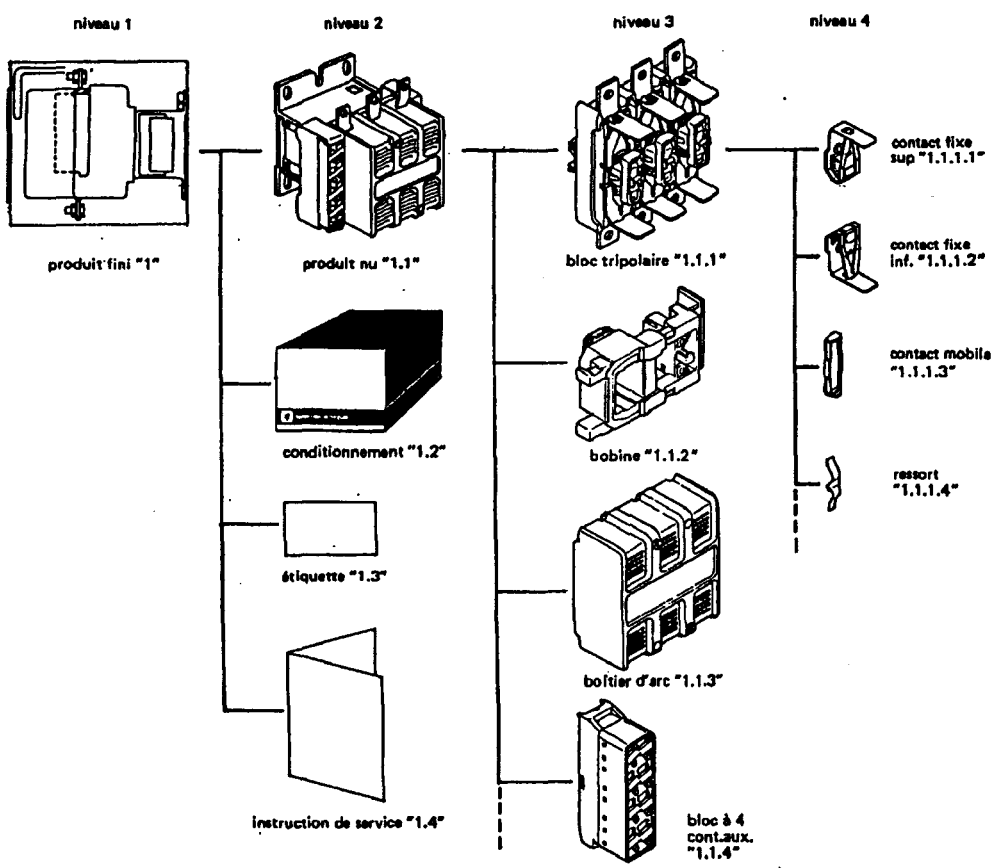


Figure 6.3 : exemple de décomposition d'un produit.

Dans la suite de ce chapitre, nous illustrerons les différentes étapes de décomposition du processus de fabrication à partir de l'exemple d'un disjoncteur moteur dont nous donnons la décomposition sur la figure 6.4 (les produits retenus dans le cadre du site productique de la DRD ne nécessitent que deux niveaux de décomposition). Cette dernière représentation utilisant les identifiants des articles est appelée *synoptique de structure* [Telemecanique, 1993].

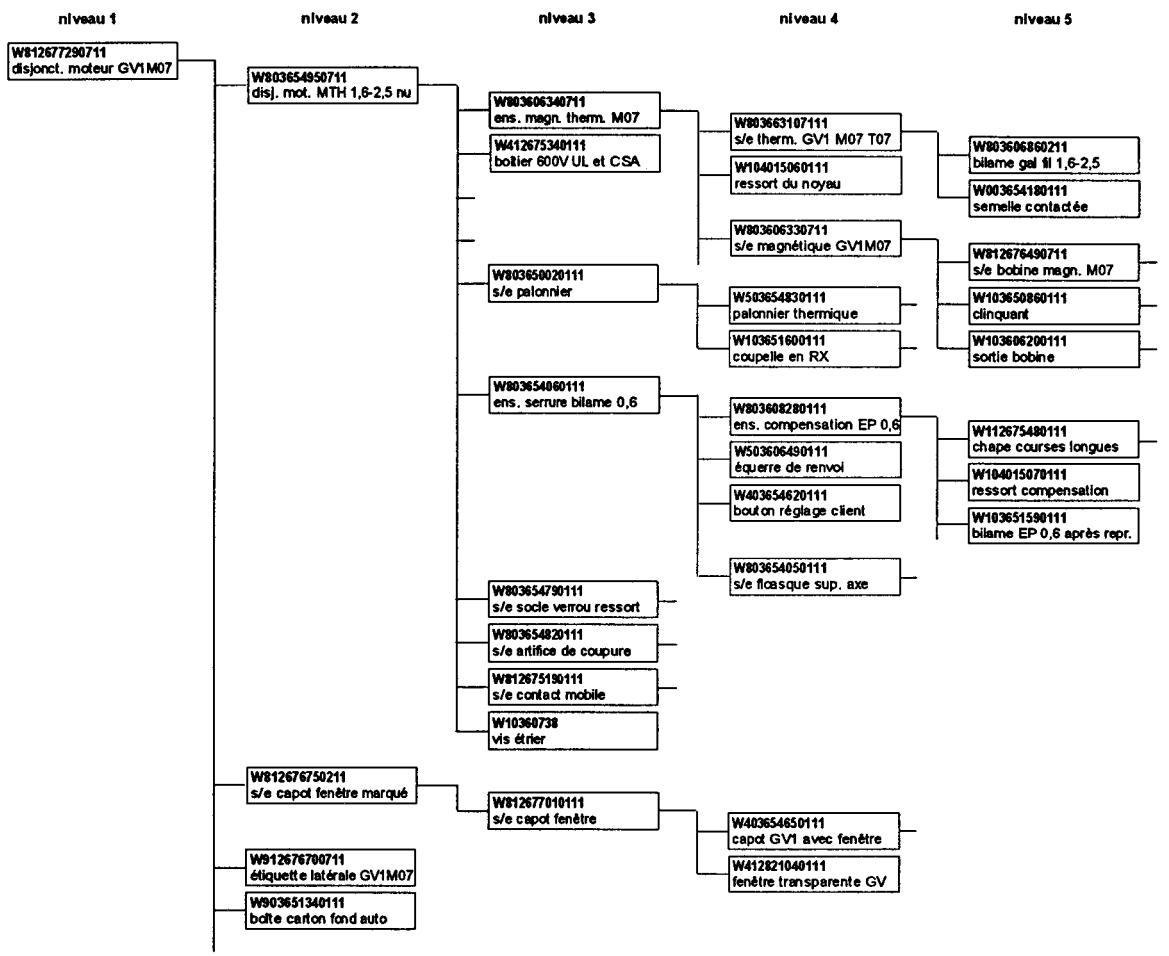


Figure 6.4 : exemple de synoptique de structure.

6.3 Organisation du processus de fabrication

La figure 6.5 met en évidence l'organisation hiérarchisée du processus de fabrication que nous détaillerons par la suite.

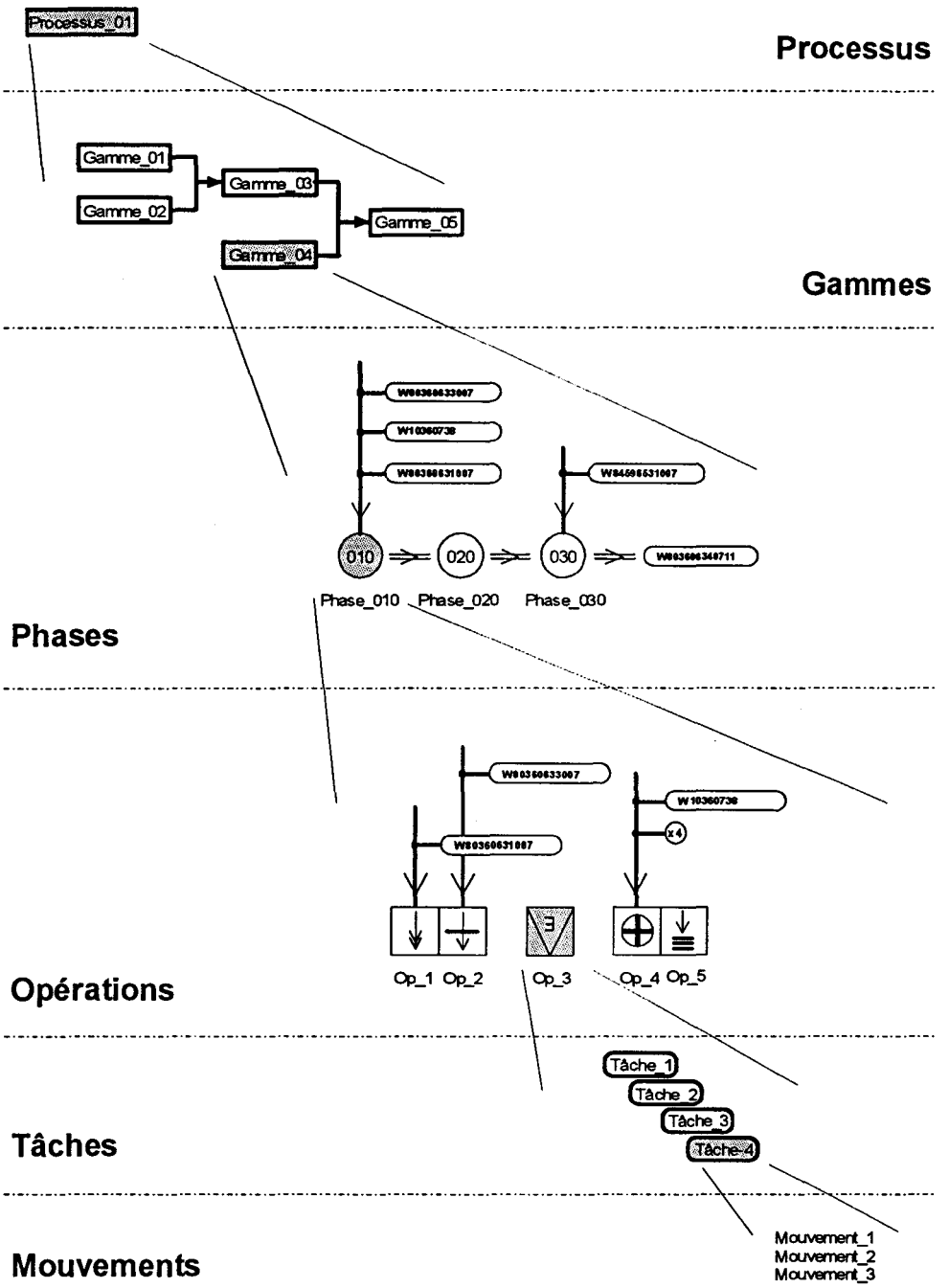


Figure 6.5 : organisation générale du processus de fabrication d'un produit

6.3.1 Processus de fabrication et gammes

Le *processus de fabrication* spécifie le séquencement des diverses gammes opératoires permettant d'obtenir un article donné à partir des constituants élémentaires. Comme pour la décomposition du produit dont il se

déduit, il peut être représenté sous forme d'un arbre. On appelle *synoptique de fabrication* une telle représentation. Le produit choisi dans l'exemple de la figure 6.6 correspond à celui du synoptique de structure de la figure 6.4. La comparaison entre les deux figures permet d'apprécier la dualité existante entre les deux représentations[2] : l'une s'attache aux constituants (synoptique de structure), l'autre aux opérations (synoptique de fabrication).

Lorsqu'il est complété par les moyens associés (non représentés sur la figure), le synoptique de fabrication porte le nom d'*analyse de processus*.

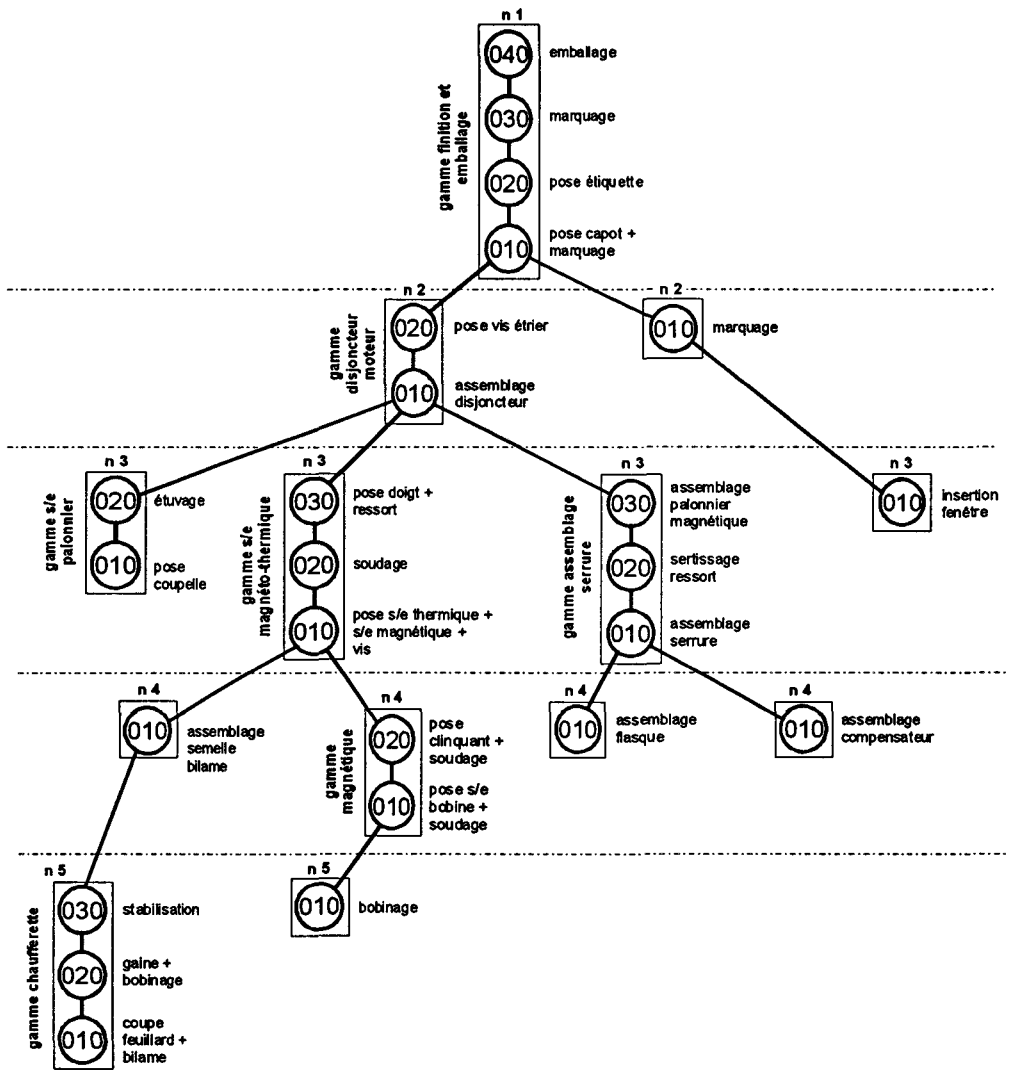


Figure 6.6 : exemple de synoptique de fabrication.

La *gamme de fabrication* d'un article représente la suite **linéaire et ordonnée** des phases permettant d'obtenir cet article. Elle peut être associée au peigne du synoptique de structure issu de l'article.

[2] Cette même dualité a déjà été rencontrée au paragraphe 5.1.5 à propos des gammes caractéristiques.

Lorsque les phases sont définies, il est possible de leur associer un temps opératoire, ainsi que la liste des composants incorporés à leur niveau. L'ensemble de ces données pour l'ensemble des phases d'une gamme représente l'*analyse de déroulement* de la gamme. La représentation graphique de l'analyse de déroulement est nommé *diagramme de déroulement*. Elle fait généralement appel à un symbolisme graphique permettant d'identifier chaque opération suivant son type (cf. § 5.1.5).

La figure 6.7 représente un exemple simplifié, sans référence à un symbolisme quelconque, de diagramme de déroulement pour la gamme "finition et emballage" de l'exemple du disjoncteur moteur (cf. figure 6.6).

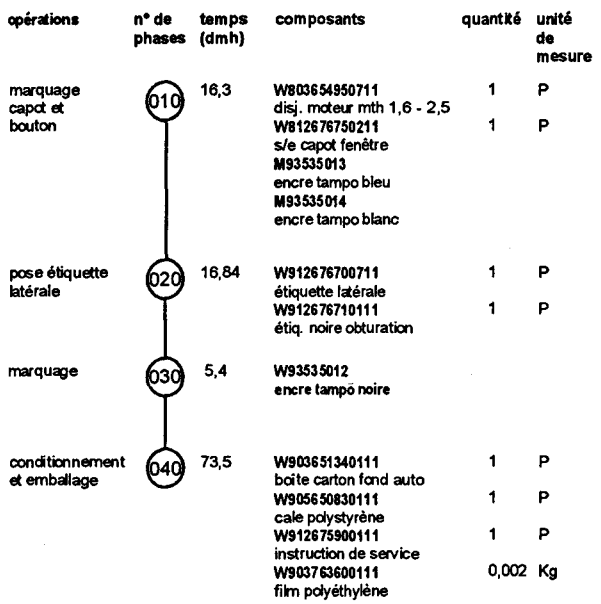


Figure 6.7 : exemple de diagramme de déroulement.

6.3.2 Opérations et phases

La suite linéaire des opérations effectuées sur un même moyen de production constitue une *phase*.

L'*opération* représente un regroupement de tâches exécutées entre l'entrée et la sortie d'un seul des dispositifs dont est doté un moyen.

La détermination des opérations se fait conjointement avec celle des procédés de fabrication, ainsi qu'avec l'étude des moyens chargés de les réaliser.

Là encore, un symbolisme graphique est généralement employé pour représenter la succession des opérations d'une phase (cf. § 5.1.5). Les symboles employés dérivent généralement de ceux employés pour les phases selon une hiérarchie classe/sous-classe. Afin de fixer les idées, nous présentons sur la figure 6.8 un exemple de représentation à partir de symboles proposés dans [Telemecanique, 1989]. Il s'agit des opérations de la phase 010 de la gamme "assemblage du sous-ensemble magnéto-thermique" du synoptique de

fabrication de la figure 6.6. Les intitulés des opérations permettent ici de comprendre les symboles employés (posage, vissage, transfert, contrôle ...). Les flèches verticales parvenant aux opérations et étiquetées par un identifiant d'article désignent les articles assemblés au niveau de chaque opération.

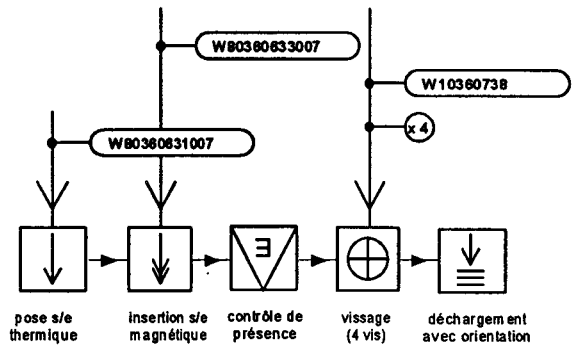


Figure 6.8 : exemple de représentation graphique des opérations d'une phase.

6.3.3 Mouvements et tâches

La *tâche* est un élément de travail, décomposable en un ou plusieurs mouvements élémentaires, et constitutif d'une opération.

Le *mouvement* est un acte humain élémentaire et défini correspondant à un déplacement du corps ou d'une partie du corps.

La décomposition des opérations en terme de tâches et de mouvements est facultative, la gestion de production ne descendant pas à ce niveau de détails. Ces niveaux sont surtout destinés à faciliter le calcul de paramètres quantitatifs de gestion associés aux opérations. Par conséquent, la spécification de l'ordre des tâches dans une opération et de l'ordre des mouvements dans une tâche n'est pas utile.

Nous avons illustré, sur la figure 6.9, les différents niveaux de décomposition de la phase 010 utilisée dans l'exemple de la figure 6.8.

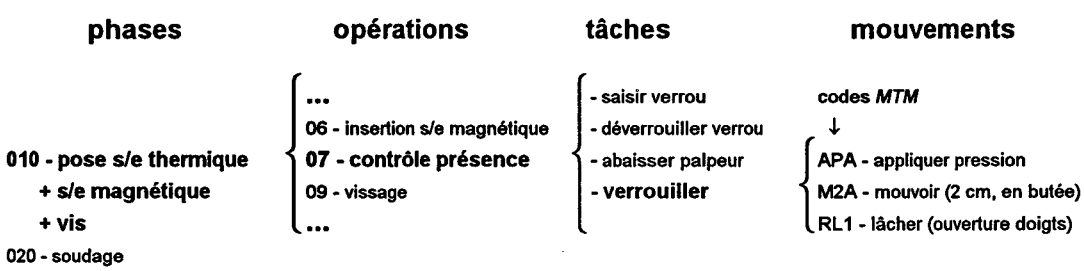


Figure 6.9 : niveaux de décomposition des phases.

Chapitre 7

Organisation hiérarchique du système de commande

Introduction

Nous précisons dans ce chapitre, à la lumière des résultats de l'analyse (chapitres 5 et 6), l'architecture du système de commande tel que nous l'envisageons dans le cadre de ce travail. Cette architecture utilise les principes de hiérarchisation introduits au chapitre 2.

Nous proposons une structuration autour des moyens de production. A partir de l'organisation hiérarchique des moyens établie au chapitre 5, nous déclinons divers niveaux de commande, structurés eux-mêmes en centres décisionnels (hiérarchie "externe"). L'organisation interne d'un centre, bâtie autour des divers types de décisions de coordination à effectuer, est ensuite explicitée (hiérarchie "interne").

De cette architecture, sera déduite l'implantation physique de la commande.

7.1 Hiérarchie externe et hiérarchie interne

7.1.1 Centres décisionnels et hiérarchie externe

Nous avons exposé au chapitre 5 une méthode de décomposition permettant d'identifier les moyens d'un système de fabrication : une hiérarchie structurelle (moyens structurels) et une hiérarchie fonctionnelle (systèmes opérants) ont ainsi été mises en évidence.

Une première structure hiérarchique du système de commande, que nous qualifierons "d'externe", est obtenue de la manière suivante à partir de la hiérarchie structurelle : à chaque niveau hiérarchique des moyens **structurels**, nous associerons un *niveau de commande* et à chaque moyen structurel appartenant à ce niveau, nous associerons un *centre de décision*.

Les requêtes à destination de chaque moyen peuvent être décomposées en sous-requêtes suivant la décomposition hiérarchique. L'ensemble de ces requêtes nécessite d'être **coordonné** afin d'atteindre l'objectif assigné au système, à savoir la production des produits.

Deux types de coordination peuvent être distingués :

- la coordination des moyens structurels constitutifs d'un même niveau non élémentaire ;
- la coordination des systèmes opérants constitutifs des moyens structurels élémentaires.

Ces deux types de coordination n'apparaissent jamais simultanément au niveau d'un même centre décisionnel, comme nous allons le voir ci-dessous.

7.1.2 Interface externe d'un centre

La figure 7.1 illustre l'organisation externe des centres décisionnels. Considérons un centre décisionnel à un niveau de commande donné (atelier, cellule ou station par exemple). Ce centre est en relation avec (cf. § 2.2.3) : l'environnement externe, son entourage, sa partie opérative.

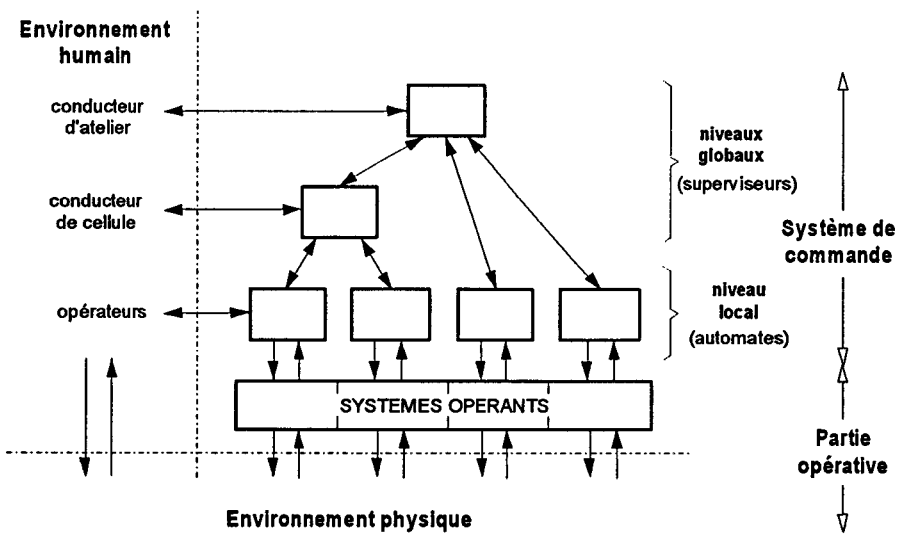


Figure 7.1 : hiérarchie externe des centres décisionnels.

Un centre ne pouvant agir physiquement sur l'environnement externe que par l'intermédiaire de sa partie opérative, les relations directes entre un centre et l'environnement se limitent aux seules interactions avec les acteurs humains intervenant à ce niveau.

L'entourage d'un centre comprend le centre hiérarchiquement supérieur ainsi que les centres de même niveau. Seul le mécanisme de coordination – reliant des centres de niveaux différents – ayant été retenu dans ce travail, les relations d'un centre avec son entourage se limitent aux seuls échanges hiérarchiques.

En raison de la décomposition hiérarchique adoptée, la partie opérative d'un centre est constituée :

- soit des centres hiérarchiquement inférieurs lorsqu'il s'agit d'un centre associé à un moyen structurel non élémentaire ;
- soit des systèmes opérants constitutifs du moyen lorsqu'il s'agit d'un centre associé à un moyen élémentaire.

Les centres associés aux moyens structurels élémentaires seront qualifiés de *locaux*, et les centres associés aux moyens non élémentaires de *globaux*. Les centres locaux étant en contact direct avec les systèmes opérants, ils ont donc en charge la coordination de ces systèmes. Les centres globaux ont, quant à eux, la charge de la coordination des centres locaux ou globaux immédiatement inférieurs. Contrairement aux centres locaux, les centres globaux peuvent intervenir à des niveaux hiérarchiques différents.

7.1.3 Structure d'un centre décisionnel et hiérarchie interne

La structure fonctionnelle d'un centre de décision est illustrée sur la figure 7.2.

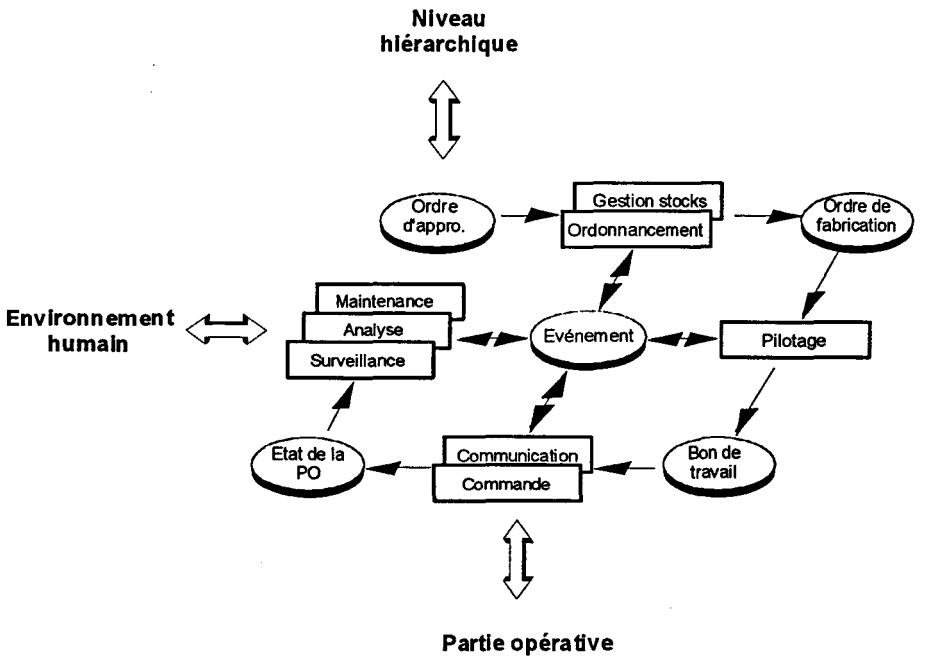


Figure 7.2 : hiérarchie fonctionnelle interne d'un centre décisionnel.

Quatre groupes principaux peuvent être distingués, constitués d'un ou plusieurs modules fonctionnels (dans un superviseur, ces modules correspondent aux modules de supervision "métier" du projet SCOPES) :

- groupe ordonnancement / gestion des stocks ;
- groupe pilotage ;
- groupe commande / communication ;
- le groupe surveillance / analyse / maintenance.

La figure met en évidence le flux décisionnel essentiel : à partir des commandes clients (ou ordre d'approvisionnement) issus du niveau hiérarchique sont générés les ordres de fabrication (groupe ordonnancement / gestion des stocks), puis les bons de fabrication (groupe pilotage), enfin les commandes élémentaires à destination de la partie opérative (groupe commande).

Bien entendu, en raison de la structure hiérarchique externe, ce flux n'est pas entièrement réalisé au niveau d'un seul centre : ainsi les commandes clients peuvent être affinées plusieurs fois – par des centres différents – avant d'être transmises sous forme d'ordres de fabrication (OF) à la fonction pilotage ; ces OF peuvent à leur tour faire l'objet d'affinements successifs avant d'être décomposés en terme de bons de fabrication. Par contre, une fois les bons parvenus à un centre local, les bons ne sont plus décomposés : une chaîne décisionnelle locale constituée des fonctions ordonnancement, pilotage et commande utilise ces bons afin de générer les commandes élémentaires à destination de la partie opérative physique. La figure 7.3 illustre l'exemple d'une chaîne décisionnelle traversant trois niveaux globaux avant de parvenir au niveau local.

Chaque centre décisionnel de la hiérarchie externe de commande se voit a priori affecter l'ensemble des niveaux de la hiérarchie interne. Néanmoins les fonctionnalités associées à chaque niveau fonctionnel diffèrent, comme nous le verrons au paragraphe suivant, selon le caractère local ou global du centre. En outre, les attributions de certains centres globaux peuvent se limiter à une seule fonction : par exemple, un centre global associé au niveau usine se limitera souvent à une fonction d'ordonnancement ; de même certains centres intermédiaires peuvent n'avoir qu'une fonction de pilotage. Notons qu'il est ainsi possible de se ramener à la structuration CIM classique (cf. § 2.2.2) dans laquelle un niveau correspond à une fonction particulière. Néanmoins, à la différence du CIM, cette correspondance n'est pas obligatoire dans notre approche. De plus, en permettant la cohabitation des fonctions dans un même centre, la structure des chaînes décisionnelles n'est pas fixe : les chaînes peuvent varier suivant la famille de produits ou, pour une même famille, plusieurs chaînes peuvent être possibles, privilégiant telle ou telle fonction (nous verrons que la structure d'une chaîne est déterminée par le choix de découpage du processus de fabrication associé au produit).

Nous détaillons dans le paragraphe suivant le rôle de chaque groupe fonctionnel ainsi que les modèles mis en oeuvre au niveau des fonctions principales (ordonnancement, pilotage, commande locale).

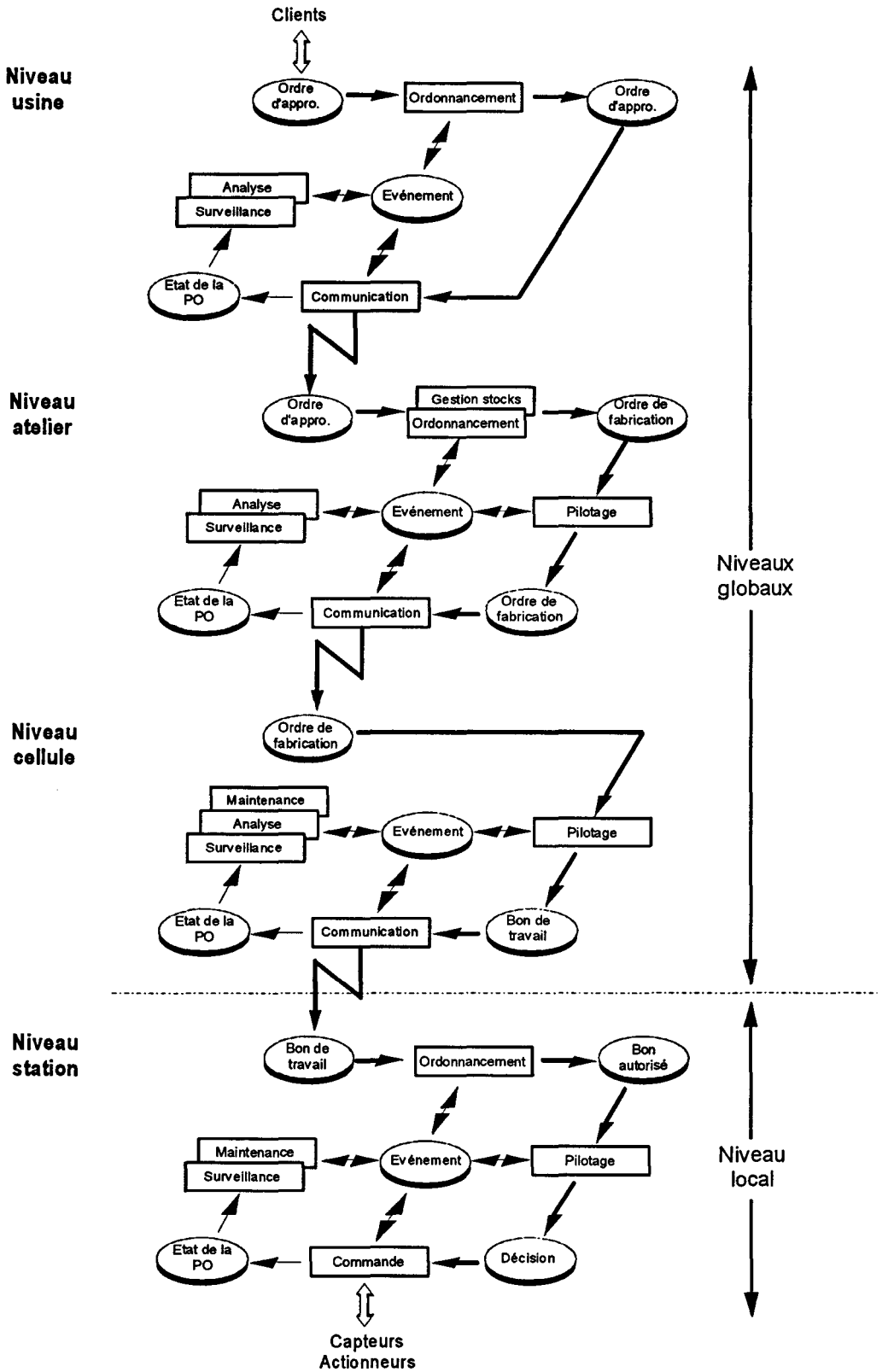


Figure 7.3 : exemple de chaîne décisionnelle.

7.2 Fonctionnement de la hiérarchie

7.2.1 Ordonnancement / gestion des stocks

Au niveau global, le groupe ordonnancement / gestion des stocks a un rôle prévisionnel. Il s'assure de la présence des matières premières et constituants dans les stocks amont, génère au besoin les ordres d'approvisionnement, puis autorise le lancement des gammes en production en transmettant les ordres de fabrication à la fonction pilotage.

A partir de la liste des commandes transmises par le niveau supérieur (sous forme d'ordre d'approvisionnement) et des nomenclatures (déduite du synoptique de fabrication - cf. ci-dessous), la fonction *ordonnancement* établit la liste des articles à produire aux niveaux des centres inférieurs, ainsi que les dates de lancement au plus tôt et/ou au plus tard. Celles-ci doivent tenir compte de façon prévisionnelle des rendez-vous d'assemblage, tout en optimisant un critère global choisi par l'utilisateur (produire au plus tôt, équilibrer les charges, ...). Les rendez-vous d'assemblage nécessitent d'être synchronisés de façon précise afin d'éviter toute attente, voire blocage, au niveau d'un moyen. Cette synchronisation constitue l'un des rôles essentiel du groupe ordonnancement / gestion des stocks[1].

L'ordonnancement doit également fournir certaines affectations. Deux cas se présentent suivant le résultat généré. Lorsqu'il s'agit d'un ordre d'approvisionnement d'un d'article pouvant être réalisé par plusieurs centres fournisseurs, le choix du centre est déterminé par l'ordonnancement. Lorsqu'il s'agit d'un ordre de fabrication à transmettre au pilotage, les affectations concernées sont celles des phases constitutives de la gamme associée (nous verrons au paragraphe suivant que le pilotage raisonne au niveau des gammes). La résolution de ces affectations n'est pas obligatoire lorsque le module de pilotage est capable de les exploiter. Tel est le cas de la solution adoptée dans ce travail : les affectations des opérations au niveau des gammes seront supposées libres à l'issue de l'ordonnancement autorisant ainsi une plus grande autonomie au niveau du pilotage.

Les dates prévisionnelles établies par l'ordonnancement sont utilisées au niveau de la *gestion des stocks* pour autoriser les lancements des articles en production, i. e. transmettre les OF à la fonction pilotage. Les politiques choisies doivent, notamment pour les stocks en amont des postes d'assemblage, coordonner de façon précise les rendez-vous [Aubert-Barbier, 1990]. Outre le respect des dates de lancement, il est également nécessaire de prendre en compte à ce niveau l'état réel du système de production, pour réagir à des événements imprévisibles susceptibles de remettre en cause le résultat de l'ordonnancement (changement de mode de marche d'un moyen en raison d'une panne par exemple).

---

[1] Il est à noter que ce rôle est négligeable (synchronisation outil - pièce à usiner) voire inexistant en usinage où la fonction essentielle de l'ordonnancement demeure l'optimisation du séquencement des lots de façon à minimiser les temps de préparation sur les machines.

Nous avons ici supposé que l'ensemble des constituants conditionnant les lancements sont fournis par les centres du niveau inférieur : cela est nécessaire en raison du mécanisme retenu au niveau des relations inter-centres, à savoir la coordination. Ainsi que nous l'avons déjà mentionné au § 3.3.1, ce type de relation verticale n'est pas réaliste aux niveaux les plus élevés de la hiérarchie (usine ou entreprise) : une structuration horizontale imposant une relation de coopération doit alors être considérée. Cette relation de coopération peut être gérée, soit au niveau de l'ordonnancement, soit à un niveau fonctionnel supérieur supplémentaire (programmation ou planification). Néanmoins, elle doit à notre sens laisser place à une relation du type coordination, a priori plus performante, aux niveaux inférieurs de la hiérarchie tant interne (pilotage, commande) qu'externe (atelier, cellule).

Au niveau local, l'ordonnancement établit le prochain bon de fabrication à considérer parmi les bons fournis par le niveau global. Deux modes sont possibles : *flow shop* et *job shop*[2]. Ces deux modes définissent, au niveau du moyen, la flexibilité relative à la simultanéité de production et par extension, conditionnent celle de l'installation au niveau auquel on se place. En mode flow shop, un moyen doit attendre la fin du lot en-cours avant de traiter le lot suivant. En mode job shop, plusieurs lots peuvent être traités simultanément. Le mode local est choisi en fonction notamment des temps de configuration nécessaires entre deux campagnes de production successives (changement d'outil, réglages, etc.). Nous verrons, au chapitre 11, que les stratégies de pilotage relatives à la gestion des changements de campagne doivent tenir compte du mode local.

Synoptique de fabrication

L'ordonnancement raisonne sur le synoptique de fabrication spécifiant les contraintes de séquencement entre gammes des articles participant à l'assemblage.

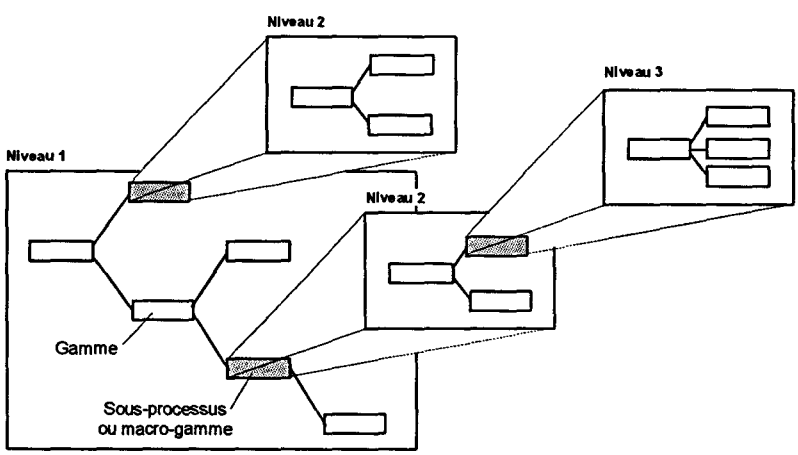


Figure 7.4 : hiérarchisation du synoptique de fabrication.

[2] Ces modes sont également nommés mode préemptif et mode concurrent par Verdebout [Verdebout, 1993].

Néanmoins, le synoptique complet d'un produit, tel que présenté au chapitre 6, doit être agrégé afin de correspondre au niveau de la hiérarchie externe de commande auquel on se place. Cette agrégation est obtenue par une hiérarchisation du modèle selon le principe illustré sur la figure 7.4. Pour un centre capable de satisfaire un ordre d'approvisionnement d'un article donné, seule la partie du graphe de cet article agrégé au niveau auquel appartient le centre est nécessaire au module d'ordonnancement.

7.2.2 Pilotage

Au niveau global, le pilotage réalise la coordination des divers moyens de production d'un même niveau de commande. Pour cela, elle calcule, à partir des ordres de fabrication fournis par l'ordonnancement, les bons de travail à transmettre aux divers centres locaux et assure le suivi de l'avancement de la production. Le suivi centralisé permet notamment d'apporter une solution à un certain nombre de problèmes jusqu'à présent très mal résolus dans les systèmes industriels rencontrés. Il s'agit essentiellement de :

- la détection des fins de campagnes au niveau de chaque moyen ;
- la compensation dynamique des pièces rebutées ;
- la gestion des boucles de retouches.

La présentation détaillée des solutions techniques retenues pour chacun de ces problèmes fait l'objet de la partie 3.

Les diverses contraintes de coordination que doit respecter le pilotage sont spécifiées par les gammes opératoires des divers produits concernant le niveau de commande considéré. Nous préciserons ci-dessous cette notion par rapport aux notions de phases et d'opérations présentées dans le chapitre 6. Le modèle des gammes opératoires que nous avons retenu dans le cadre de ce travail sera présenté au chapitre 8.

Une gamme définit un séquencement d'opérations (rassemblées en phases) à réaliser pour obtenir un article donné à partir de constituants supposés **disponibles**. Les opérations éventuelles pour obtenir ces constituants lorsqu'ils ne sont pas élémentaires doivent être spécifiées par d'autres gammes. La coordination inter-gammes et la synchronisation des rendez-vous d'assemblage résultants n'entrent pas dans le cadre du pilotage, mais dans celui de l'ordonnancement (cf. § 7.2.1). Nous verrons néanmoins au § 8.1.3 que cette affirmation doit être nuancée : une synchronisation élémentaire peut en effet être réalisée au niveau du pilotage dans certains cas particuliers.

Bien que les contraintes de coordination réduisent les possibilités d'évolution du procédé en choisissant celles qui donnent un sens à la production, tous les indéterminismes ne sont pas résolus. En effet, afin de permettre une meilleure réactivité en cas d'aléas, un certain nombre de flexibilités doivent être conservées au niveau des gammes opératoires. Dans le cadre de nos travaux, ces flexibilités correspondent aux décisions de routage et d'affectation. Les modèles que nous développerons pour le pilotage permettront de mettre en évidence ces diverses flexibilités de routage (cf. chapitre 8). Certaines de ces flexibilités, notamment les décisions d'affectation, peuvent être

résolues de façon prévisionnelle par l'ordonnancement. Néanmoins, aux niveaux inférieurs de la hiérarchie externe (cellule, îlot) les conséquences des aléas sont telles qu'elles remettent très rapidement en cause tout planning prévisionnel trop strict. Il se révèle donc plus intéressant de résoudre les flexibilités le plus tard possible, en fonction de l'état courant du système. C'est l'option que nous avons adopté dans ce travail.

Comme nous le verrons dans la partie 3, deux mécanismes mis en oeuvre au niveau du pilotage global permettront de traiter ces indéterminismes. Ils se distinguent par leur degré "temps réel". Le premier mécanisme consiste à calculer, préalablement à chaque campagne, les règles de routage les plus adaptées par rapport l'état courant du système au moment du lancement ; ces règles sont ensuite appliquées au niveau du pilotage local (paramétrage dynamique du niveau local). Le second mécanisme consiste, quant à lui, à prendre les décisions en temps réel, en réponse à des requêtes émises par le niveau local : il nécessite donc a priori une disponibilité permanente du niveau global entrant en contradiction avec la volonté des industriels de décentraliser au maximum la commande. Nous verrons dans la partie 3 que la solution que nous préconisons réalise un compromis acceptable entre approche décentralisée et approche centralisée.

Au niveau local, le pilotage assure, à partir des bons de travail transmis par le niveau global, le lancement des actions élémentaires. La grande majorité des décisions de circulation sera également résolue à ce niveau à partir de règles simples calculées par la fonction pilotage de niveau global (paramétrage dynamique) ou paramétrées à demeure par l'exploitant.

Gammes, phases et macro-phases

De même que pour le synoptique de fabrication, le modèle des gammes opératoires destiné à un centre donné doit être agrégé au niveau de commande auquel appartient ce centre. Nous étendrons aux gammes le principe déjà utilisé pour les phases. Ainsi, de même que les opérations d'une phase sont réalisées par un **même** moyen structurel élémentaire, les phases d'une gamme seront toutes réalisées par les divers moyens constitutifs d'un **même** moyen structurel non élémentaire (atelier, îlot, cellule suivant le niveau).

En pratique, il est toujours possible de se ramener à ce principe, au besoin en utilisant la notion de macro-phase afin de regrouper les séquences de phases affectées à un même moyen structurel (cf. figure 7.5).

Gammes et macro-phases spécifient ainsi le séquencement des **phases** à réaliser à un niveau de commande donné pour obtenir un article à partir de ses composants. Les phases, quant à elles, spécifient le séquencement des **opérations**. Cependant, ainsi que nous le verrons au chapitre 8, nous ramènerons, dans certains cas, la spécification du séquencement des opérations au niveau des gammes, court-circuitant ainsi la notion de phase. Ceci se justifie par le fait qu'une même opération peut apparaître dans plusieurs phases affectées à un même moyen. Cela est particulièrement vrai dans les systèmes flexibles utilisant des robots ou des opérateurs humains (station robotisée d'usinage, station manuelle d'assemblage). Bâtir les gammes, et donc le pilotage,

sur la notion d'opération nous permet de nous affranchir d'un niveau de description (la phase) qui, à notre sens, correspond à une énumération combinatoire des diverses séquences d'opérations réalisables sur un même moyen.

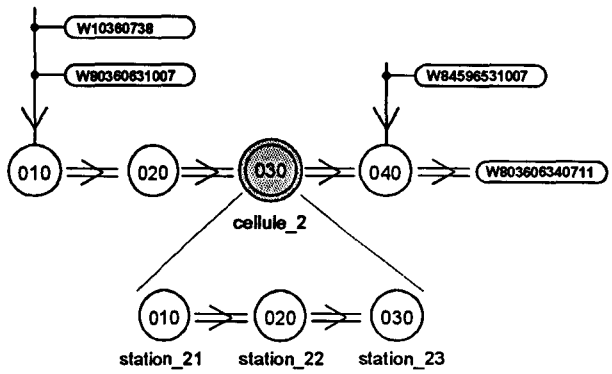


Figure 7.5 : notion de macro-phase.

Néanmoins, la notion d'opération définie dans le chapitre 6 doit, selon nous, être complétée afin de prendre en compte les concepts de modularité et de réutilisabilité développés dans le cadre des méthodes de conception par objets. Une opération devrait ainsi, idéalement, être définie de façon à pouvoir être réutilisée dans la définition de nouvelles gammes (pour le même article ou un article différent).

7.2.3 Commande / communication

Au niveau global, cette fonction réalise la communication avec les centres du niveau inférieur. Elle assure la transmission des ordres (ordres d'approvisionnement, ordres de fabrication ou bons de fabrication), ainsi que la réception des événements et comptes-rendus en provenance de ces centres. Ces informations permettent la mise à jour de la base de données sur l'état des centres locaux utilisée par les diverses fonctions de la hiérarchie interne.

Au niveau local, elle réalise le séquencement des commandes à destination des équipements de la partie opérative, ainsi que la coordination des divers équipements entre eux. Pour cela, elle dispose du modèle des chaînes fonctionnelles sur lequel nous reviendrons ci-dessous. Elle assure également le rafraîchissement des données locales sur l'état du système physique commandé. En cas d'indéterminisme résultant d'une flexibilité éventuelle du système, cette fonction génère un événement à destination de la fonction de pilotage locale qui détermine en temps réel la prochaine opération à réaliser.

Chaînes fonctionnelles

Les chaînes fonctionnelles spécifient la coordination des sous-systèmes fonctionnels réalisant les tâches élémentaires constitutives d'une opération.

Cette notion rejoint celle de graphe de synchronisation développée dans le cadre du projet CASPAIM [Ausfelder, 1994].

A la différence de la structuration en terme de gammes, phases et opérations, participant d'une "approche produit", les chaînes fonctionnelles participent d'une "approche équipement". En effet, la coordination des systèmes fonctionnels ne peut pas se ramener à un séquencement d'opérations sur les produits. On conçoit intuitivement qu'une distinction doit être opérée à un certain niveau entre la spécification des opérations à partir des produits (analyse descendante) et la spécification des tâches ou mouvements (donnant les diverses chaînes fonctionnelles) à partir des informations capteurs et commandes actionneurs (analyse ascendante). Néanmoins, la limite reste difficile à définir : une opération peut très souvent elle-même être décomposée en une séquence linéaire d'opérations plus élémentaires : où, dès lors, faut-il arrêter cette décomposition et débuter l'analyse des opérations "élémentaires" ainsi obtenues en terme de chaînes fonctionnelles ?

Le choix de la décomposition idéale reste lié à la compétence du concepteur. Ce choix s'apparente à celui de l'organisation des programmes en informatique. Les concepts de structuration introduits par l'approche objet (modularité, réutilisabilité, maintenabilité, etc.) peuvent d'ailleurs être appliqués à notre problème : il est en effet possible d'identifier les divers moyens structurels élémentaires à des objets (au sens informatique) dont les diverses chaînes fonctionnelles constitueraient les méthodes. Les gammes opératoires, et, de façon plus général, tous les modèles établis à partir de l'approche produit, sont alors assimilables à des "scénarii" ou "scripts" permettant de coordonner l'action simultanée des différents objets.

Exemple :

A titre d'exemple, deux types de chaînes fonctionnelles sont nécessaires pour commander les systèmes opérants intervenant dans un système de transfert du type convoyeur sur table tel que celui du site DRD :

- celles assurant la commande des indexeurs,
- celles assurant la commande des aiguillages.

De façon pratique, dans l'exemple du site, ces chaînes font l'objet de modules autonomes codés en GRAFCET et implantés dans chaque automate de station.

7.2.4 Surveillance / analyse / maintenance

Que ce soit au niveau global ou au niveau local, ces fonctions réalisent l'ensemble des services de traitement et d'affichage des informations caractéristiques sur le déroulement de la fabrication contenues dans les bases de données sur l'état du système (durée des pannes, nombre de pièces au rebut, état des stocks, ...). Le rôle de ces fonctions est de faciliter, par la mise en place d'indicateurs significatifs et synthétiques, l'interaction entre l'homme et le système automatisé. Cette interaction peut être envisagée selon deux points de vue : l'homme peut être considéré soit comme un simple élément de la partie

opérative se contentant alors d'obéir au système de commande (opérateurs sur poste manuel, opérateurs de maintenance), soit en tant qu'élément du système décisionnel. C'est ce dernier point de vue qui nous intéresse ici. Il se concrétise tant en phase d'exploitation (surveillance et validation des requêtes émises à chaque niveau) qu'en phase d'adaptation (choix des paramètres et stratégies d'exploitation - cf. § 2.2.2).

Les fonctions de surveillance doivent s'exercer à tous les niveaux. Au niveau station, une information courante à fournir aux opérateurs est le mode de marche de l'équipement. Au niveau cellule, on pourra afficher des informations sur l'état d'avancement de la production ou sur la nature des pannes (maintenance curative). Au niveau atelier il peut être intéressant de centraliser certaines informations afin de planifier le travail des opérateurs de maintenance (maintenance préventive). Dans tous les cas, la définition de l'affichage doit être à l'initiative de l'utilisateur.

Au niveau local, ces fonctions ont des implications sociales importantes. Elles offrent en effet un moyen efficace d'augmenter la responsabilité des opérateurs. Ce sujet a d'ailleurs reçu un regain d'intérêt ces derniers temps dans l'industrie : plutôt que d'exploiter, comme ce fut longtemps le cas, la seule fonction opérative des opérateurs, on cherche désormais à accroître leur fonction décisionnelle. Par exemple, des indicateurs peuvent être mis à disposition des opérateurs afin que ces derniers contrôlent par eux-mêmes la qualité des produits, réalisent des opérations de maintenance de premier niveau ou analysent la productivité des installations qu'ils contrôlent.

7.2.5 Conclusion

Le tableau ci-dessous résume les diverses considérations exposées ci-dessus.

Fonction	Niveau de commande	Moyens coordonnés	Outil de spécification
ordonnancement	niveau global	moyens structurels	synoptique de fabrication
pilotage	niveau global	moyens structurels	gammes opératoires
commande	niveau local	systèmes opérants	chaînes fonctionnelles

Trois fonctions principales peuvent être isolées dans le processus décisionnel d'affinement intervenant en phase d'exploitation : ordonnancement, pilotage, commande. Chacune de ces fonctions résout un type particulier de décision de coordination :

- la fonction "ordonnancement / gestion des stocks" assure la coordination des divers moyens structurels participant à la fabrication des divers sous-ensembles intervenant dans l'assemblage d'un article ;

- la fonction "pilotage" assure la coordination des divers moyens structurels participant à la réalisation des opérations spécifiées dans les gammes, en exploitant au mieux les flexibilités non résolues par l'ordonnancement (choix d'affectation et de routage essentiellement) ;
- la fonction "commande locale" assure la coordination des divers systèmes opérants réalisant les divers mouvements élémentaires intervenant dans les opérations.

En pratique et en se référant à l'implantation physique du système de commande proposée au paragraphe suivant, nous pouvons distinguer trois modules logiciels essentiels : le module d'ordonnancement (associé au module de gestion des stocks) du superviseur, le module de pilotage du superviseur et le module de contrôle / commande bas niveau implanté dans les automates.

Néanmoins, il convient de garder constamment à l'esprit le fait qu'un certain nombre de fonctionnalités correspondant aux fonctions ordonnancement et pilotage doivent également être implantées au niveau local. Nous présenterons dans la partie 3 les diverses fonctionnalités, tant au niveau local que global, de la fonction pilotage.

7.3 Implantation physique du système de commande

L'implantation physique de la commande sera, en général, calquée sur la hiérarchie externe du système de commande. Celle-ci, dérivant de l'analyse structurelle du système de production, entraîne en effet une architecture naturelle illustrée sur la figure 7.6. Chaque moyen structurel non élémentaire (MS) est géré par un superviseur (S). Les moyens élémentaires (MSE) sont, quant à eux, commandés par un contrôleur local de type automate (A). Superviseur et contrôleurs locaux sont reliés entre eux par l'intermédiaire d'un réseau local industriel. Des regroupements peuvent bien sûr être effectués en pratique pour minimiser les investissements. Ainsi un même superviseur peut-il gérer à la fois deux, voire plusieurs, moyens appartenant à des niveaux identiques ou différents. Il en est de même pour les contrôleurs locaux.

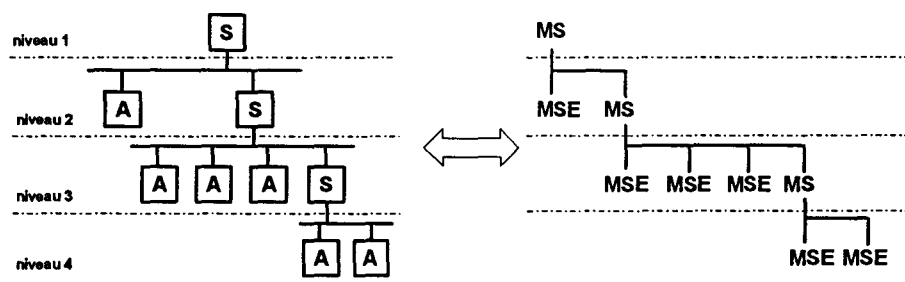


Figure 7.6 : répartition physique de la commande.

Le cas de la commande des moyens structurels élémentaires de type système de transfert est particulier. En effet, très souvent, la commande des

divers systèmes opérants associés n'est plus centralisée mais répartie. Ainsi, la commande des aiguillages d'un convoyeur à palettes est-elle la plupart du temps réalisée au niveau des contrôleurs assurant la conduite des moyens desservis. Tel est le cas du site TE, où chaque dérivation du convoyeur de la cellule d'assemblage est gérée par un automate différent.

Application au site TE

Le système de commande du site comprend trois niveaux de commande et deux systèmes de supervision (cf. figure 7.7). Comme le montre clairement la figure, ce système est calqué sur la décomposition structurelle du site. Seuls, les moyens structurels de type système de transfert ne possèdent pas de contrôleur associé : leur commande est répartie au niveau de chaque contrôleur de station.

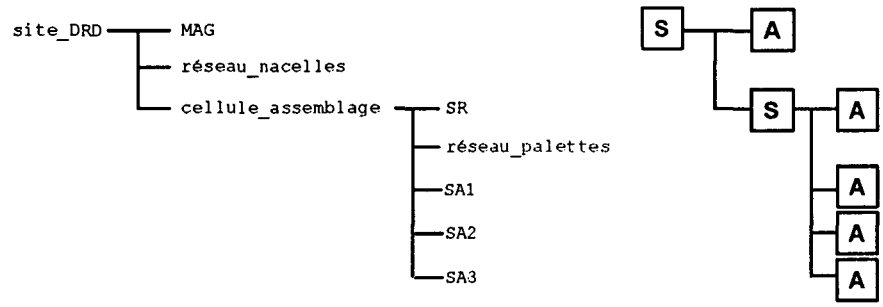


Figure 7.7 : système de commande du site

Nous ramènerons notre étude, dans la suite de ce mémoire, à l'architecture simplifiée suivante : un seul centre global coordonnant l'activité de centres locaux répartis. Le système de commande se réduit ainsi à deux niveaux, car ne comprenant qu'un seul niveau global. Néanmoins, la solution de pilotage que nous proposons en partie 3 se généralise lorsque plusieurs niveaux globaux doivent être mis en oeuvre : les fonctionnalités locales que nous établirons doivent alors également être mises en oeuvre au niveau des centres globaux, lorsque ces derniers sont eux-mêmes locaux vis à vis d'un niveau global hiérarchiquement supérieur.

Concernant l'exemple illustratif du site DRD, nous nous placerons au niveau de la cellule d'assemblage (niveau 2 de la décomposition structurelle). Le niveau global correspond au superviseur de cellule et le niveau local aux différents automates de stations (niveau 3).

Chapitre 8

Modélisation en vue du pilotage

Introduction

Nous détaillons dans ce chapitre les différents modèles établis en vue du pilotage. Il s'agit :

- des gammes opératoires, utilisées pour la gestion des opérations sur les produits ;
- du modèle de l'installation, dérivé des relations d'accessibilité et utilisé pour la conduite des transferts dans l'installation ;
- des affectations, reliant l'ensemble des opérations apparaissant au niveau des gammes à l'ensemble des moyens apparaissant sur le modèle de l'installation.

Ces modèles doivent, bien entendu, être agrégés en fonction du niveau de commande auquel on se place (cf. chapitre 7).

8.1 Modèle des gammes opératoires

8.1.1 Inconvénients de la représentation linéaire

L'utilisation d'une représentation linéaire des gammes, telle celle illustrée dans le chapitre 6, ne limite théoriquement en rien l'exploitation des flexibilités d'un système de production, à condition d'autoriser qu'un même article puisse être décrit par plusieurs gammes.

Généralement, il est défini une gamme principale (utilisée en mode automatique) et une gamme de remplacement (utilisée en mode dégradé). Un

BU LILLE

certain nombre de gammes de retouche, décrivant le travail supplémentaire à exécuter en cas de non conformité, peuvent être également associées à l'article en question. En outre, si un même article possède deux variantes de fabrication, c'est-à-dire deux nomenclatures différentes, il convient de définir plusieurs gammes de fabrication différentes, utilisant ou non certains moyens communs.

Cependant, la représentation linéaire rend difficile la compréhension des divers enchaînements possibles entre gammes participant du même article. Elle génère des séquences similaires et redondantes qu'il n'est pas facile d'identifier et de relier les unes aux autres. Elle empêche donc d'avoir une vue d'ensemble des gammes d'un article.

De plus, l'exploitation pratique de cette représentation est délicate, le changement de gamme en temps réel posant de nombreux problèmes.

Nous préférons donc utiliser une modélisation synthétique, représentant la totalité des gammes opératoires associées au produit concerné. Nous ne prétendons pas ici redéfinir la notion de gamme, mais en fournir une représentation qui permette une compréhension globale.

8.1.2 Modélisation retenue

Nous représenterons **les** gammes opératoires d'une *famille d'articles* par un graphe. Nous utiliserons la modélisation par opération (cf. § 5.1.5), consistant à associer les noeuds aux opérations.

Une opération affectée à plusieurs moyens sera représentée par un seul noeud. Les affectations seront spécifiées par ailleurs (cf. § 8.3).

Bien que l'objet du pilotage soit l'opération et non la phase (cf. § 7.2.2), les phases affectées aux systèmes de transfert ne seront pas décomposées en terme d'opérations au niveau des gammes. En effet, exprimer, à ce niveau, l'ensemble des opérations intervenant dans un transfert, résulte en une explosion combinatoire du graphe lorsque les possibilités de routage sont multiples. Tel est le cas des installations présentant des circuits d'adressage, formés d'un convoyeur central desservant des postes placés en dérivation. Le site de la DRD constitue un exemple d'une telle installation. Nous verrons, dans la partie 3, que cela aura des conséquences sur la solution de pilotage proposée.

Formalisme utilisé

Le formalisme illustré sur la figure 8.1 sera utilisé :

- les noeuds-opérations seront représentés par un cercle ;
- les arcs reliant deux opérations symboliseront les contraintes de précédences induites par les flux physiques (d'autres contraintes pourront être ajoutées grâce au *graphe d'influence* contrôlant les campagnes de production – cf. chapitre 11) ;

- ♦ les arcs seront étiquetés par variantes (cf. figure 8.1.a) et/ou par état (cf. figure 8.1.b) ;
- ♦ les arcs initiaux et terminaux (i. e. n'ayant qu'une extrémité) seront distingués des précédents ; ils matérialisent l'arrivée ou la sortie des composants (cf. figures 8.1.c et 8.1.d) ;
- ♦ les articles utilisés lors d'une opération d'assemblage seront indiqués par une étiquette sur les arcs initiaux (cf. figure 8.1.c) ;
- ♦ les articles obtenus lors d'une opération d'assemblage seront indiqués par une étiquette sur les arcs terminaux (cf. figure 8.1.d) ;

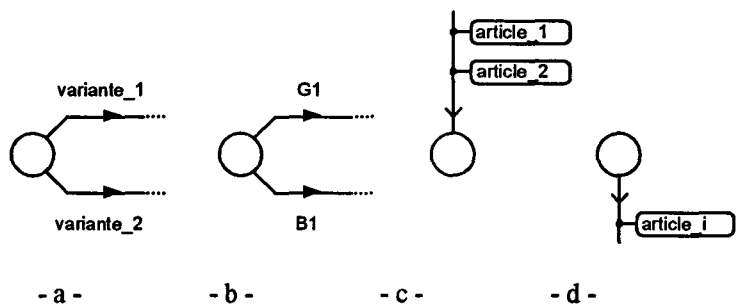


Figure 8.1 : motifs de base pour la représentation des gammes opératoires.

Examinons, à la lumière de ce formalisme, les possibilités de modélisation des diverses flexibilités apparaissant au niveau des gammes opératoires.

Variantes

L'utilisation des arcs étiquetés par variantes, ou plus précisément par jeu de variantes (cf. § 5.1.6), permet de représenter sur un seul graphe l'ensemble des gammes relatives à une même famille de produits. L'introduction de ces étiquettes a été motivée par les deux problèmes suivants :

- discrimination des moyens suivant les variantes : deux variantes d'un produit peuvent nécessiter chacune un moyen différent pour une même opération de la gamme. Par exemple, un poste de peinture peut être dédié aux couleurs claires, un autre aux couleurs foncées.
- séquences opératoires différentes suivant les variantes : la réalisation d'une variante particulière peut nécessiter un supplément d'opérations ou un séquencement différent de ces opérations.

Le premier problème pourrait être résolu en rajoutant une colonne par variabilité au tableau des affectations présenté au paragraphe 8.3. L'affectation d'un moyen à une opération pourrait, par ce biais, être reliée à une combinaison de valeurs ou de conditions sur les variabilités. Cette option a été retenue par l'Institut de Microtechnique de l'EPFL (notion de processus unifié [Verdebout, 1993]). L'utilisation des arcs paramétrés par variante fournit toutefois une solution aux deux problèmes à la fois en discriminant les séquences spécifiques à chaque variante : le parcours de la gamme effectué lors

du processus d'instanciation (cf. § 11.3) suivra les arcs correspondant à la variante demandée.

Exemple :

Soit une famille de produits définie par trois variabilités :
v1 pouvant prendre 2 valeurs : {OUI, NON}[1]
v2 pouvant prendre toute valeur entière entre 1 et 100
v3 pouvant prendre 3 valeurs : {rouge, bleu, vert}

Les processus de fabrication associés sont les suivants :
op_A puis op_B puis op_D puis op_E si v1 = OUI et v3 ∈ {bleu, vert}
op_A puis op_C puis op_D puis op_E si v1 = NON et v3 = rouge.

Cinq moyens sont disponibles : M1, M2, M3, M4 et M5

Les affectations sont les suivantes :
M1 réalise op_A
M2 réalise op_B et op_C
M3 réalise op_D
M4 réalise op_E si v2 < 45
M5 réalise op_E si v2 >= 45

La définition des jeux de variantes pour cet exemple pourrait être la suivante :
variante_1 : (v1 = OUI et v3 ∈ {bleu, vert})
variante_2 : (v1 = NON et v3 = rouge)
variante_3 : (v2 < 45)
variante_4 : (v2 >= 45)

Les jeux "variante_1" et "variante_2" sont mutuellement exclusifs. Il en est de même des jeux "variante_3" et "variante_4". Par contre, un même article peut satisfaire simultanément deux jeux non exclusifs tels les jeux "variante_1" et "variante_3".

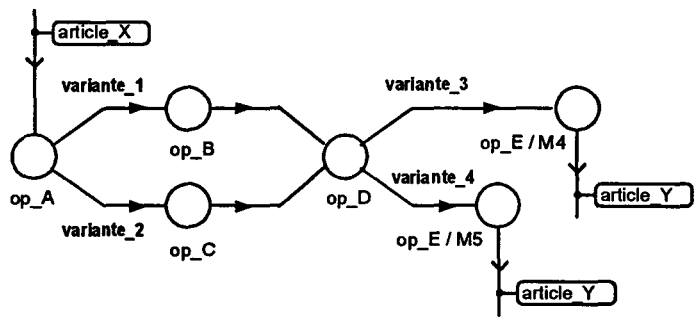


Figure 8.2 : utilisation des arcs étiquetés par variantes.

[1] La signification associée à chacune des valeurs est ici sans importance.

La représentation des gammes de la famille en question est donnée sur la figure 8.2. La duplication de l'opération op_E est nécessaire puisque nous ne tenons pas compte des variantes dans le modèle des affectations (cf. § 8.3).

Contrôles et retouches

L'utilisation des arcs étiquetés par état permet de combiner sur un seul graphe gammes normales et gammes de retouche. Il est ainsi aisé de déterminer quelles opérations réaliser en cas d'un défaut sur le produit.

Dans l'état actuel, seuls les états de conformité extrêmes (bon ou mauvais) sont pris en compte. L'extension à n degrés ne pose pas de problème. Deux types d'étiquettes permettent de discriminer les opérations à effectuer en fonction de l'état bon (étiquettes préfixées G[ood]) ou de l'état mauvais (étiquettes préfixées B[ad]).

Plusieurs schémas de mise en oeuvre de ces arcs sont possibles (cf. figure 8.3).

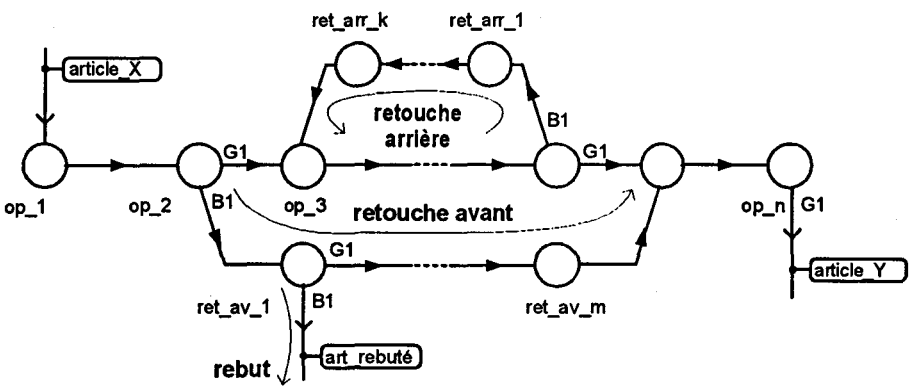


Figure 8.3 : utilisation des arcs étiquetés par état.

- **Rebut** : ce cas apparaît lorsque, à l'issue d'une opération de contrôle, un défaut irrécupérable est détecté. La pièce défectueuse est alors éjectée du système.
- **Boucle de retouche "avant"** : ce cas apparaît lorsque les opérations d'une gamme de retouche font **progresser** le stade du produit : à l'issue de la retouche le produit retrouve une branche non encore parcourue de la gamme principale.
- **Boucle de retouche arrière** : ce cas, inverse du précédent, apparaît lorsque les opérations de la gamme de retouche font **régresser** le stade du produit : le retour sur la gamme normale s'effectue donc sur une portion déjà parcourue.

  Ce dernier schéma introduit des circuits (au sens de la théorie des graphes) dans le graphe des gammes. Cette spécificité devra être prise en compte dans les algorithmes de parcours de graphe que nous utiliserons dans la solution de pilotage exposée dans la partie 3.

L'utilisation des arcs étiquetés par état, nécessaire en pratique, introduit néanmoins une difficulté qui s'avère critique lors des changements de campagne : il devient en effet impossible de déterminer par avance ou localement le nombre de fois qu'une opération donnée devra encore être effectuée pour terminer un lot en cours. Ce problème nécessite un traitement centralisé et justifie à lui seul le choix de l'approche adoptée. Nous y reviendrons en détail dans la partie 3.

Ordre sur les opérations

Du point de vue mathématique, l'ensemble des opérations à réaliser sur un produit est un ensemble partiellement ordonné. L'ordre de certaines opérations peut en effet être indifférent. Cette flexibilité sur le séquencement des opérations devrait être conservée dans un modèle des produits destiné au pilotage. Bien que complexe à gérer, elle peut en effet être exploitée pour optimiser la production et pour réagir à d'éventuels aléas.

Bien que la modélisation adoptée permette de représenter cette flexibilité, elle n'est pas exploitée dans la solution de pilotage exposée en partie 3. Elle s'avère en effet très peu répandue dans le domaine de l'assemblage auquel nous nous sommes particulièrement intéressés. Son exploitation pourrait par contre s'avérer appréciable dans le domaine de l'usinage.

Une solution envisageable, avec la solution actuelle, est de définir plusieurs variantes de fabrication par la définition d'une variabilité supplémentaire. Les valeurs prises par cette variabilité permettraient de choisir différentes séquences opératoires. Cependant l'instanciation des variabilités avant le lancement en production empêche l'exploitation en temps réel de ce principe. Il enrichit néanmoins les possibilités d'optimisation au niveau de l'ordonnancement (la variabilité introduite devenant un paramètre à optimiser).

8.1.3 Cas de synthèse des gammes relatives à des articles différents

La notion de gamme, telle que nous l'avons introduite, définit une séquence linéaire d'opérations pour obtenir un article donné à partir de ses articles constituants. Classiquement, une pièce principale (appelée récepteur), introduite lors de la première opération de la gamme, est suivie tout au long de la gamme. A chaque opération d'assemblage, un ou plusieurs constituants supplémentaires sont ajoutés (cf. figure 8.4).

L'enrichissement de la représentation linéaire, par l'introduction des étiquettes sur les arcs, que ce soit par variante ou par état, ne modifie en rien ce schéma : un article principal peut toujours être suivi le long d'un chemin du graphe enrichi. Les opérations sont effectuées les unes après les autres. Les opérations éventuelles à effectuer pour obtenir les constituants sont spécifiées sur les gammes associées à ces constituants. A l'ordonnancement d'assurer la synchronisation des diverses gammes en jouant sur les dates de lancement.

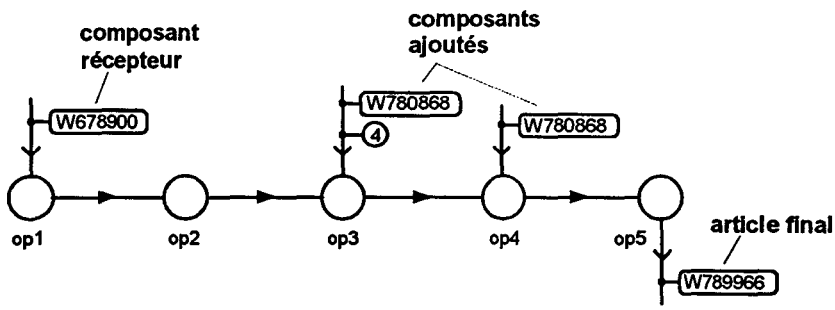


Figure 8.4 : structure linéaire classique d'une gamme.

Ce principe général s'avère cependant excessif lorsque les rendez-vous ne nécessitent pas de synchronisation précise. Tel est le cas si les durées des opérations à réaliser pour obtenir les constituants peuvent être négligées. Tel est le cas également lorsque les diverses lignes de fabrication sont balancées équitablement : il devient alors possible de lancer en fabrication chaque constituant à la même date. L'intervention de l'ordonnancement n'est alors plus justifiée pour calculer les lancements. Lorsque c'est le cas, la représentation des diverses gammes peut être synthétisée, afin d'être gérée directement par le pilotage.

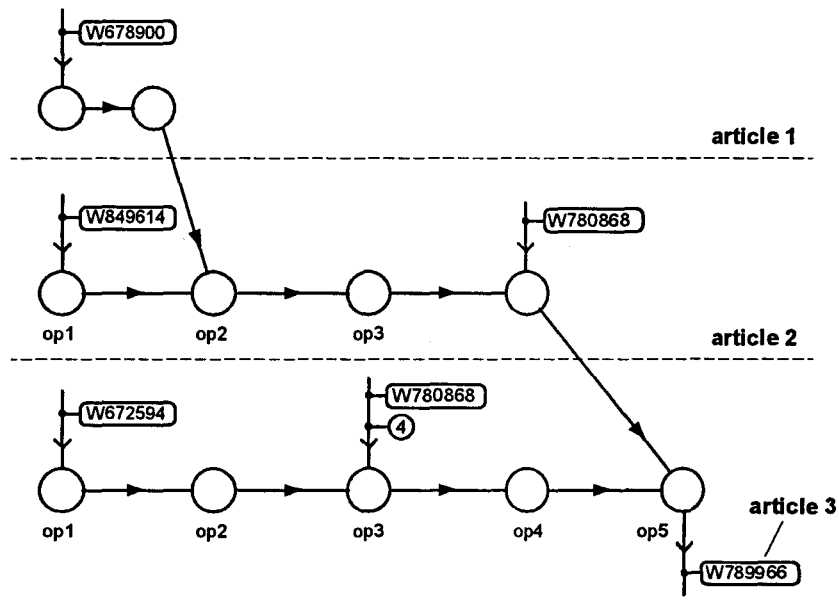


Figure 8.5 : synthèse de trois gammes.

Nous adopterons la convention proposée dans [Cruette, 1991] : les gammes des différents sous-ensembles, seront séparées par un trait pointillé horizontal. La figure 8.5 illustre le principe de la synthèse de trois gammes. Il consiste simplement à connecter les arcs initiaux et terminaux associés à un même article.

La combinaison de la représentation ainsi obtenue avec les *graphes d'influence* (cf. chapitre 12) permettra de mettre en oeuvre une synchronisation élémentaire : il sera possible de contrôler par pilotage le déclenchement d'une opération après occurrence d'une opération donnée.

8.1.4 Application à l'exemple du site DRD

Les gammes opératoires des produits réalisés sur le site sont représentées figure 8.6. Prenons l'exemple du produit S. La séquence d'opérations normale correspond à une palettisation (`SR_pal`), suivie d'une opération d'assemblage mettant en jeu 11 cubes (`S_op1`), suivie enfin d'une dépalettisation (`pal_SR`). Un contrôle a été mis en place au niveau de l'assemblage, permettant ainsi de distinguer les bons produits des mauvais. Ces derniers sont alors retouchés (`S_rep1`) avant de reprendre la séquence normale.

Le flux étiqueté par G1 issu de l'opération de dépalettisation (`pal_SR`) correspond au flux des palettes vides : la prochaine opération que doit subir ce flux correspond à l'opération de palettisation, d'où le bouclage de la gamme.

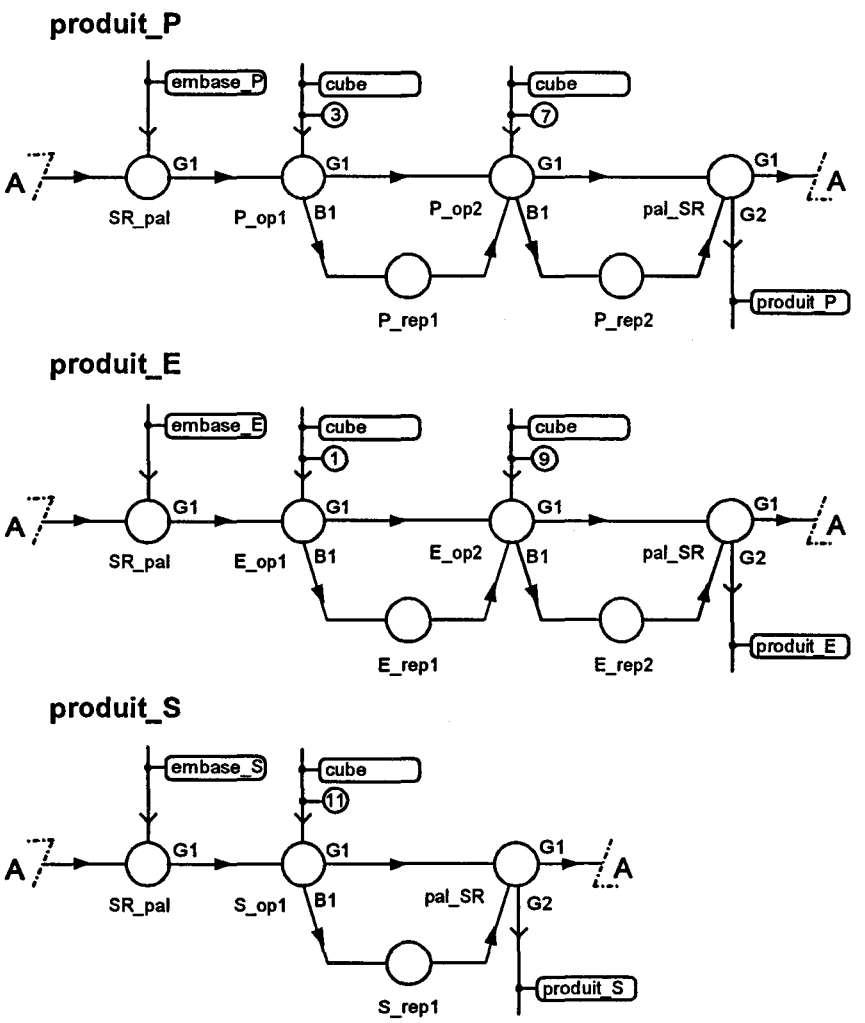


Figure 8.6 : gammes opératoires des produits du site.

8.2 Modèle de l'installation

8.2.1 Les éléments de l'installation

Conformément à la définition du rôle de la fonction pilotage, seuls les moyens structurels doivent concerner, a priori, une modélisation de l'installation en vue du pilotage. Cependant, l'exploitation des flexibilités internes aux moyens de type système de transfert nécessite, pour ces moyens, une décomposition fine en terme de systèmes opérants.

A partir de l'analyse des lieux et des moyens menés au chapitre 5, nous avons donc retenu la décomposition suivante, associée au formalisme de la figure 8.7 :

- moyens structurels autres que systèmes de transfert, décomposables en moyens élémentaires (tampon_E/S et moyen de niveau terminal) et moyens non élémentaires ;
- moyens structurels de type système de transfert. Pour ces systèmes, une décomposition en terme d'aiguillages, zones de transfert et indexeurs a été adoptée. Les indexeurs seront assimilés aux moyens qu'ils desservent.

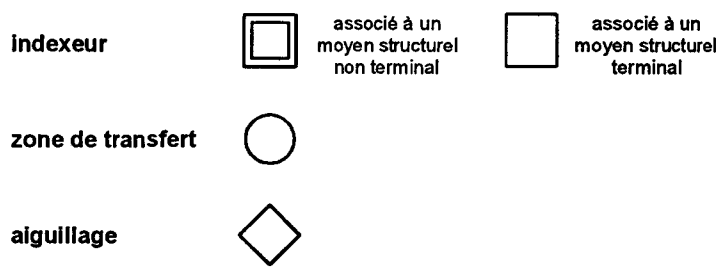


Figure 8.7 : motifs de base pour la représentation de l'installation.

8.2.2 Les relations d'accessibilité

Les diverses relations d'accessibilité exprimant les divers échanges physiques possibles entre éléments de l'installation sont représentées au moyen d'arcs orientés reliant les symboles représentant les divers éléments. Ces arcs ne correspondent à aucun élément physique de l'installation (ce ne sont pas des portions de convoyeurs ou des buffers). Les flux entrant et sortant de l'installation (réception composants, livraison produits finis, éjection des rebuts) peuvent être représentés mais, ainsi que nous le verrons dans la partie 3, ils ne seront pas utilisés par les modules opérationnels de pilotage. En revanche, ces données seraient nécessaires au niveau d'un outil d'analyse permettant de valider la cohérence des gammes avec le modèle de l'installation.

Le modèle obtenu est destiné à exploiter les flexibilités apparaissant au niveau des systèmes de transfert : il permet de déterminer, pour un mobile situé sur un système de transfert et ayant une destination donnée, l'ensemble des trajets possibles pour ce mobile. L'utilisation précise en sera explicitée au chapitre 13.

Pour l'exemple du site DRD, le modèle de l'installation nécessaire au module de pilotage de niveau "cellule" est représenté sur la figure 8.8. Conformément au principe de hiérarchisation, seul le système de transfert associé à ce niveau (le convoyeur à palettes) est représenté.

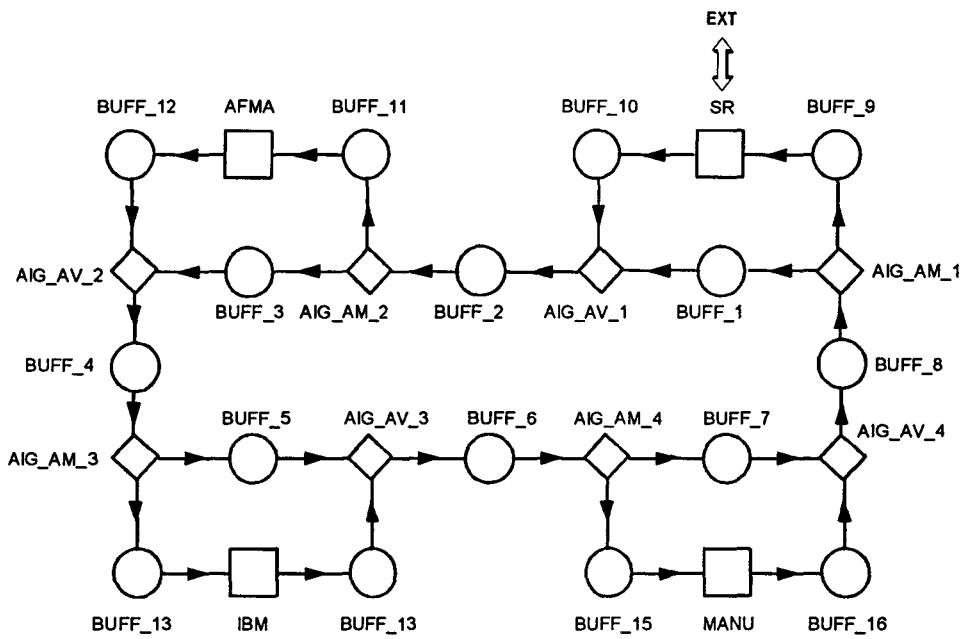


Figure 8.8 : modélisation du site TE agrégé au niveau "cellule".

Au niveau "atelier", le modèle de l'installation, toujours pour l'exemple du site DRD, serait celui de la figure 8.9. Il se réduit au système de transfert inter-cellules par nacelles aériennes.

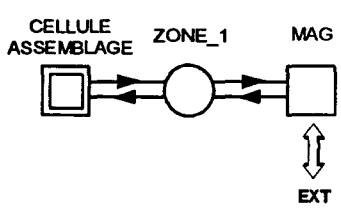


Figure 8.9 : modélisation du site TE agrégé au niveau "atelier".

8.3 Affectations

La liste des affectations de chaque opération réelle, doit être connue du pilotage. Les affectations sont fournies sous forme de tables (cf. § 5.4). Ces tables peuvent être statiques ou dynamiques suivant que l'on dispose ou non d'un ordonnancement permettant d'optimiser les affectations.

Les affectations multiples constituent, avec la flexibilité interne des systèmes de transfert (plusieurs chemins pour une même destination), l'une des flexibilités les plus intéressantes du système. Ces flexibilités permettent en effet de pallier à la défaillance de certains moyens par réorientation des produits à destination d'un moyen valide capable d'effectuer la même opération.

Dans le prototype de superviseur sur lequel nous avons travaillé, le module d'ordonnancement ne spécifie aucun choix d'affection : ils sont entièrement déterminés par le pilotage.

Pour la cellule d'assemblage du site DRD, les affectations sont données sur le tableau de la figure 8.10.

Op. réelle	Affectations
S_op1	IBM AFMA
E_op2	IBM
P_op2	IBM
P_op1	AFMA
E_op1	MANU AFMA
P_rep1	MANU
P_rep2	MANU
E_rep1	MANU
E_rep2	MANU
S_rep1	MANU
SR_pal	SR
pal_SR	SR

Figure 8.10 : affectations agrégées au niveau "cellule" pour l'exemple du site DRD.

Chapitre 9

Modélisation en vue de la simulation

Introduction

Nous proposons dans ce chapitre une méthodologie de modélisation des différentes interactions entre acteurs d'un système de production. L'outil de modélisation adopté, les Réseaux de Petri, a été justifié au chapitre 2.

Cette méthodologie se veut systématique, à l'instar de la démarche d'analyse proposée au chapitre 5 dont elle reprend les résultats. L'objectif poursuivi est de permettre son utilisation dans un outil informatique d'aide à la modélisation.

Initialement élaboré dans l'optique d'une recherche par méthodes analytiques de critères à respecter au niveau du pilotage (cf. § 9.5.1), le modèle obtenu sert de base au module logiciel de simulation intégré au superviseur et développé dans le cadre de ce travail. Afin de ne pas alourdir le corps du mémoire, nous avons reporté en annexe la description de ce module (annexe E) et son intégration dans l'architecture logicielle du superviseur (annexe D).

9.1 Les étapes de la simulation

Trois étapes sont nécessaires pour l'étude par simulation du comportement d'un système de production [Cavaillé et Proth, 1987] : la modélisation, la simulation proprement dite, et enfin l'interprétation des résultats obtenus.

La phase de simulation ne pose pas de problèmes particuliers, les principes algorithmiques sous-jacents étant parfaitement connus. Nous décrivons dans l'annexe E les solutions adoptées pour le module de simulation développé dans le cadre de ce travail.

Les difficultés liées à l'interprétation des résultats d'une simulation sortent du cadre de notre étude (rappelons qu'un module d'analyse est disponible dans le système de supervision – cf. § 3.2.1). Longtemps négligée dans les systèmes de simulation du marché – qui se contentaient de générer des listes indigestes de données – , l'analyse des données commence à être intégrée dans les simulateurs actuels, notamment grâce à l'apparition d'interfaces de présentation conviviales. Une méthodologie rigoureuse doit cependant être mise en oeuvre pour tirer tout le parti de ces logiciels [Héritier-Pingeon, 1991].

Nous nous sommes, pour notre part, focalisés sur la phase de modélisation. Il nous a en effet paru essentiel de définir clairement ce que devait être un modèle de simulation vis à vis des concepts mis en évidence lors de l'étape d'analyse.

9.2 La démarche de modélisation proposée

9.2.1 Principe

Le principe retenu est simple : il s'agit de représenter à tout instant l'ensemble des contraintes "physiques" autres que les contraintes de coordination (imposées par le pilotage) existant sur les opérations réelles déclenchables à un niveau de commande donné. Cette connaissance permettra de déterminer l'ensemble des états utilisables du système et donc de vérifier que le pilotage n'entraîne pas le système vers un état bloquant, voire interdit, du système (cf. § 2.1.2).

L'ensemble des données nécessaires à l'établissement du modèle provient donc de l'analyse détaillée présentée au chapitre 5. Les contraintes de coordination spécifiées par les gammes opératoires et objet du pilotage, ne doivent pas transparaître dans le modèle. Ce dernier devant permettre la validation du module de pilotage, il parait en effet cohérent de n'y intégrer aucune hypothèse sur le pilotage. Notamment, nous avons respecté le principe de flexibilité maximale [Cruette, 1991]. Ce principe exprime le fait que toute flexibilité entre opérations du système réel doit être conservée sur le modèle.

Nous nous plaçons à un niveau de commande donné. Le pilotage que l'on désire valider ne concerne que les opérations réalisées par les moyens du niveau considéré. La modélisation doit donc s'adapter à ce niveau de précision.

Trois types de contraintes physiques peuvent conditionner la réalisation d'une opération. Il s'agit :

- des contraintes imposées par les produits (gammes logiques) ;
- des contraintes imposées par les lieux (accessibilité) ;
- des contraintes imposées par les ressources actives (comportement des moyens et des systèmes opérants).

La démarche de modélisation que nous proposons s'organise autour de ces trois types de contraintes.

9.2.2 Etapes de la modélisation.

La figure 9.1 illustre les différentes étapes de la modélisation à partir des résultats obtenus lors de la phase d'analyse. La première étape consiste à établir les graphes représentant les contraintes imposées par chaque type d'entités caractéristiques (RC, RT et RP sur la figure – cf. ci-dessous). Dans la seconde étape, on établit les liens entre les différents graphes pour obtenir le modèle global de l'atelier constitué d'un seul graphe.

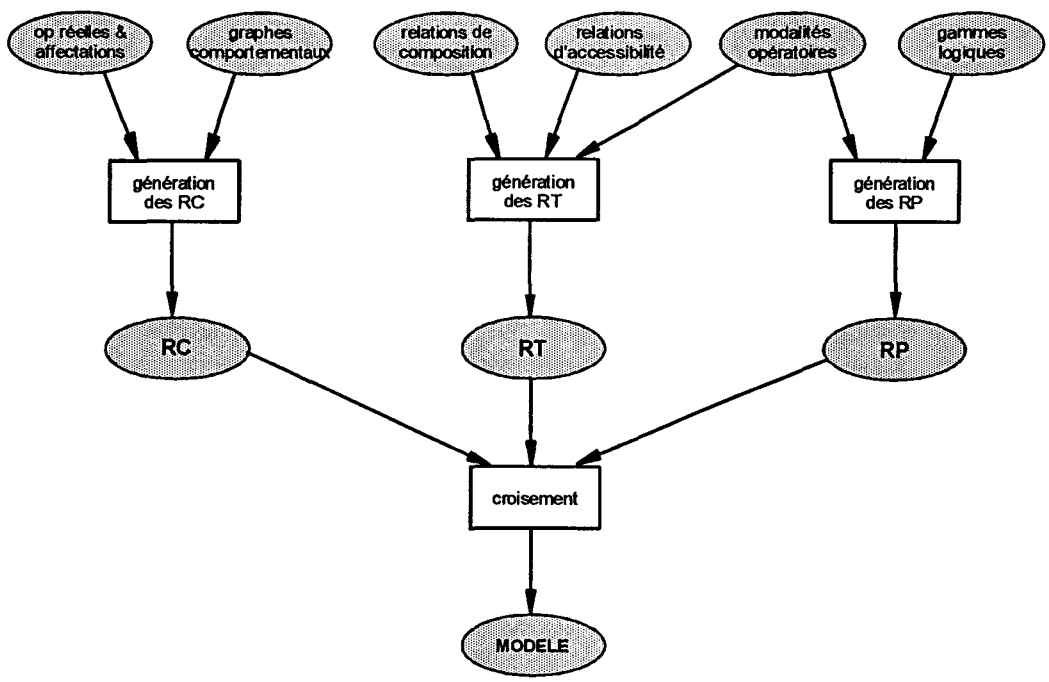


Figure 9.1 : obtention du modèle de simulation.

Résultats de l'analyse.

L'analyse du système de production a permis de mettre en évidence les éléments appartenant aux trois classes caractéristiques et de définir, pour chacun

de ces éléments, les divers états accessibles et l'ensemble des opérations permettant de passer d'un état à l'autre.

La phase d'analyse aboutit à plusieurs résultats :

- les *relations de composition* donnent la composition de chaque élément de l'atelier en terme d'ES ;
- les *relations d'accessibilité* expriment les divers flux matériels existant dans l'atelier ;
- les *affectations* spécifient les opérations réelles réalisables par chaque ressource ;
- les *graphes comportementaux* spécifient le séquencement des opérations réelles réalisables par les ressources
- les *tableaux des modalités opératoires* permettent de décrire chaque opération réelle en terme de fonctions logiques (transfert ou opération caractéristique sur les produits) ;
- les *gammes logiques* permettent de représenter les diverses possibilités de successions d'opérations caractéristiques pour réaliser un produit.

Etape 1 : construction des graphes

Cette étape s'articule autour des trois types d'éléments caractéristiques identifiés lors de la phase d'analyse, à savoir les lieux, les produits et les ressources (moyens ou systèmes opérants). Trois types de graphes permettent de représenter l'état des différents éléments appartenant à ces trois classes :

- les *réseaux produits* (RP) représentent l'état des divers produits circulant dans l'atelier : on isole autant de graphes que de familles de produits ;
- les *réseaux comportementaux* (RC) représentent l'état des ressources actives de l'atelier : il y a autant de graphes que de ressource ;.
- les *réseaux transitiques* (RT) représentent l'état des différents emplacements de stockage de l'atelier : à chaque type d'ES correspond un graphe.

Etape 2 : connexion des graphes

A l'issue de l'étape 1, trois classes distinctes de graphes ont été générées. Chacune de ces classes correspond à un type particulier de contraintes. L'étape 2 va permettre de représenter, par liaison des divers graphes, l'ensemble des contraintes conditionnant chaque opération réelle.

Les paragraphes suivants présentent en détail chacune de ces étapes. L'illustration de la méthodologie se fera à partir de l'exemple de la cellule d'assemblage du site DRD.

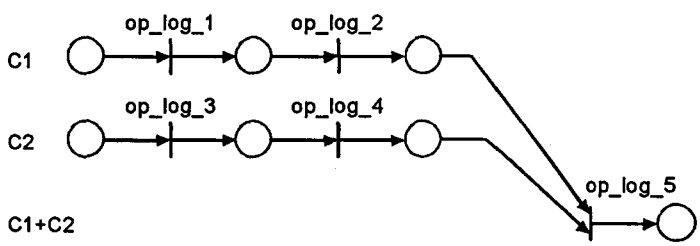
9.3 Etape 1 : construction des graphes

9.3.1 Les réseaux produits (RP)

Les RP sont établis à partir des gammes logiques, des modalités opératoires et des affectations.

Les transitions des RdP représentant les gammes logiques traduisent les opérations logiques que doivent subir les produits. Ces opérations ne correspondent pas toujours, bien que ce soit souvent le cas, aux opérations réelles réalisées par les ressources actives. Les modalités opératoires fournissent ces correspondances.

Il s'agit donc de faire correspondre les transitions du modèle des gammes logiques aux opérations réelles.



Op. réelle	Op. Logique
op_réelle_1	op_log_1 op_log_2
op_réelle_2	op_log_3
op_réelle_3	op_log_4 op_log_5

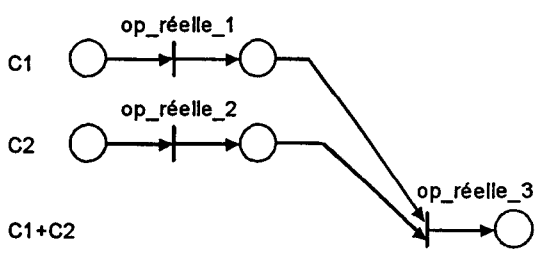


Figure 9.2 : construction des RP à partir des gammes logiques.

Le principe, illustré sur la figure 9.2, consiste à rechercher dans les réseaux les séquences de "transitions logiques" pouvant être réalisées par une opération réelle. La séquence de transitions est alors remplacée par une "transition réelle". On suppose ici que les modalités opératoires sont en accord avec les gammes logiques (i. e. pour chaque opération réelle concrétisant une

ou plusieurs opérations logiques, il est possible de trouver une séquence de transitions dans le graphe "logique").

A l'issue de cette phase, le graphe obtenu représente l'ensemble des séquencements possibles d'opérations réelles en accord avec les contraintes caractéristiques initiales imposées par les produits.

Application à l'exemple du site DRD

Les RP correspondant aux produits du site DRD sont donnés sur la figure 9.3. Les graphes obtenus sont identiques aux graphes représentant les gammes logiques (cf. figure 5.5) : les opérations réelles correspondent aux opérations logiques.

La notation pointée suivante a été employée pour désigner les opérations : `ressource1.op_réelle2` représente l'opération réelle `op_réelle2` fournie par la ressource active `ressource1`. Le point est remplacé par un point d'exclamation dans le RC associé à la ressource (cf. § 9.3.3) et par un point d'interrogation dans les RP et les RT (cf. § 9.3.2). La ressource n'est pas spécifiée lorsque l'affectation de l'opération n'est pas établie.

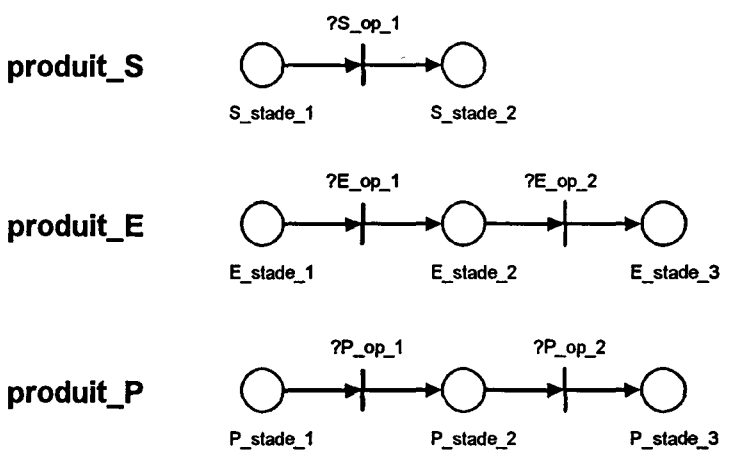


Figure 9.3 : RP correspondant aux produits assemblés sur le site DRD.

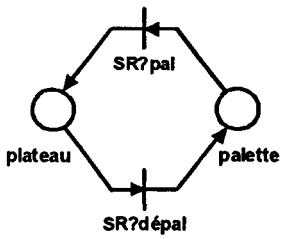
9.3.2 Les réseaux transitiques (RT)

Les RT sont obtenus à partir des relations d'accessibilité représentant les diverses possibilités de transfert entre les ES de l'installation.

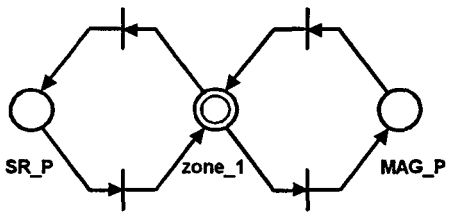
Nous n'entrerons pas ici dans le détail de la démarche, cette dernière étant illustrée dans l'annexe A à partir de l'exemple d'une portion de convoyeur. Succinctement deux phases sont nécessaires. La première phase consiste à établir pour chaque ES mis en évidence lors de l'analyse le graphe d'état représentant l'enchaînement des opérations modifiant l'état d'occupation (libre ou occupé) de cet ES. La seconde phase permet de connecter les divers graphes entre eux par fusion de transitions.

Application à l'exemple du site DRD

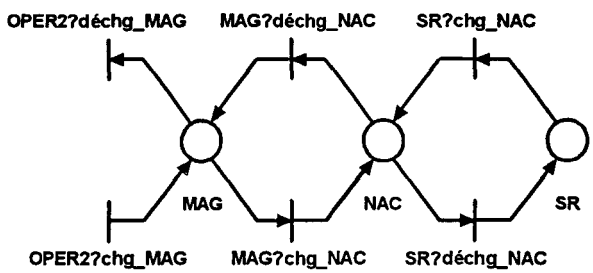
Appliqué au site, la construction des RT donne quatre réseaux indépendants correspondants aux quatre classes principales d'objets circulant dans l'atelier : nacelles, plateaux, palettes et produits (cf. figure 9.4).



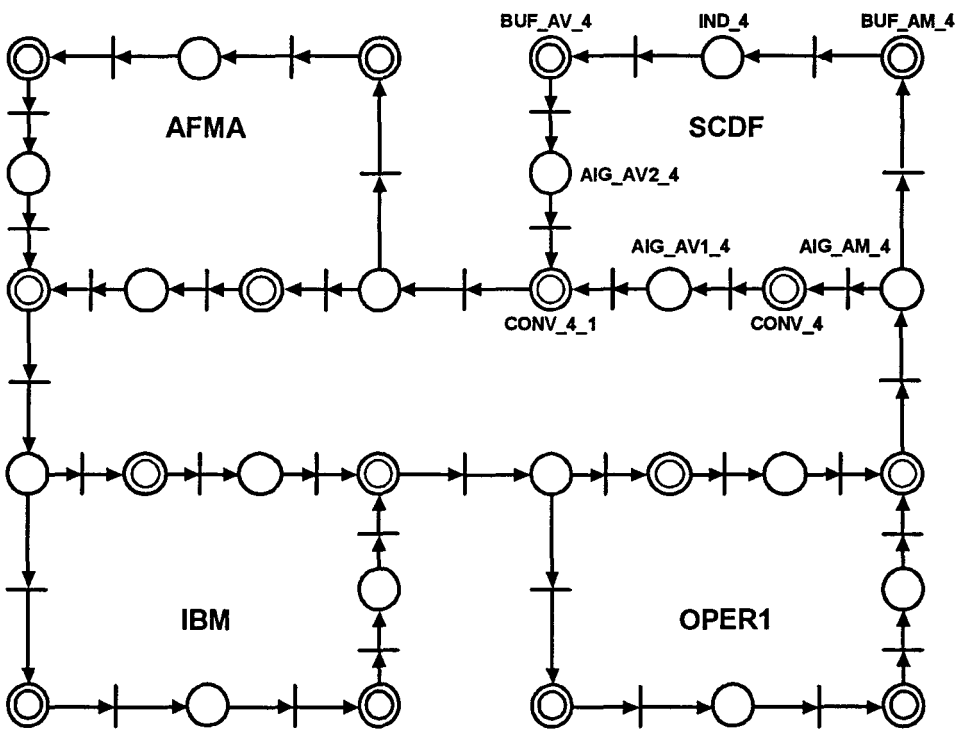
a - RT produits



b - RT nacelles



c - RT plateaux



d - RT palettes

Figure 9.4 : RT correspondant à l'exemple du site DRD.

Par souci de clarté, seules les places "occupé" ont été représentées et seuls quelques noms d'opérations ont été indiqués. Les places à contour double représentent le modèle RdP obtenu pour une portion de convoyeur (cf. annexe A).

9.3.3 Les réseaux comportementaux (RC)

Les réseaux comportementaux sont la représentation dans le formalisme RdP des graphes comportementaux définissant l'enchaînement des états que peut prendre chaque ressource (cf. § 5.3.3). Dans les applications que nous avons traitées jusqu'à présent, nous nous sommes limités aux seuls moyens structurels. Il est également possible d'associer un graphe comportemental à chaque système opérant si l'on désire valider le fonctionnement de la commande locale (chaînes fonctionnelles).

Les transitions des RC correspondent aux opérations. L'ensemble des transitions franchissables donne donc à tout instant l'ensemble des opérations que peuvent fournir les ressources.

Application à l'exemple du site DRD

Les RC correspondant aux ressources AFMA, IBM et SR du site DRD sont donnés sur la figure 9.5. Dans ces exemples, le séquencement des opérations n'est pas contraint. Ce n'est bien entendu pas toujours le cas. En fait, le nombre de contraintes est directement lié au degré de finesse accordé à la notion d'opération. Plus on se rapproche de la notion de tâches élémentaires, plus les contraintes seront nombreuses. Ainsi, dans l'exemple de SR, l'opération !pal (palettisation) peut être décomposée en une séquence contrainte d'opérations telle que : positionner_manipulateur, prendre_pièce, déplacer_pièce, poser_pièce, etc.

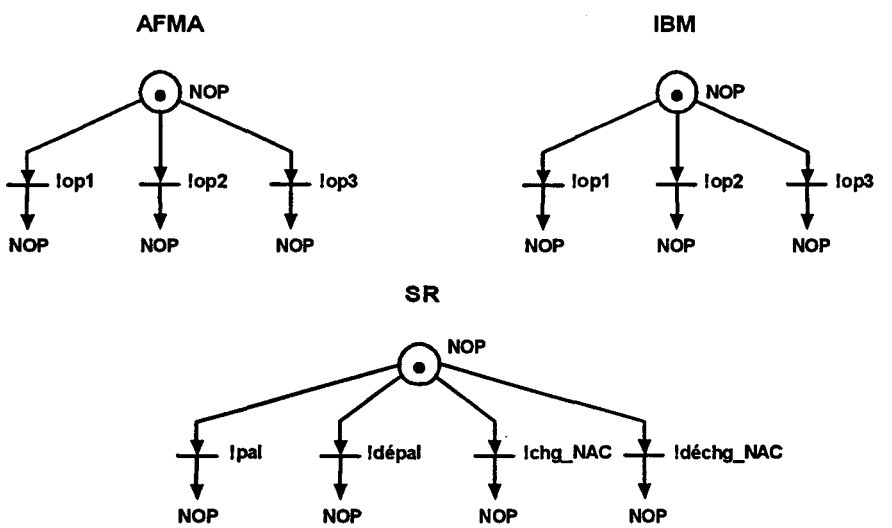


Figure 9.5 : RC correspondant aux ressources SR et AFMA du site DRD.

9.4 Etape 2 : connexion des graphes

L'étape 1 de la génération du modèle a permis d'établir trois types de réseau dont les transitions représentent :

- les opérations modifiant l'état des ES (transitions des RT) ;
- les opérations modifiant l'état des ressources (transitions des RC) ;
- les opérations modifiant l'état des produits (transitions des RP).

Cependant, une même opération a souvent des conséquences sur l'état de plusieurs entités caractéristiques à la fois et peut parfois nécessiter la présence d'autres entités dans certains états. Rappelons notamment que toute opération sur les produits ne peut être réalisée que par une ressource active. La conséquence immédiate sur nos modèles est qu'une action apparaissant dans les RP apparaît également dans les RC. Ainsi l'action de palettisation de notre exemple modifie l'état d'un ES produit sur la palette (occupation) et d'un ES produit sur le plateau (libération) ainsi que l'état de la ressource SR.

En outre, lorsqu'une opération met en jeu un produit, il convient de préciser la localisation du produit dans l'atelier autorisant la réalisation de l'opération. Il en va de même pour les supports mobiles. Ainsi, sur l'exemple du site DRD, le transfert d'un produit entre un plateau et une palette ne peut-il avoir lieu que si le plateau se trouve dans le stockeur SR et la palette sur l'indexeur IND_4.

L'étape 2 de la génération consiste à intégrer dans le modèle ces diverses relations. Trois phases sont nécessaires.

Phase 1 : connexion RP/RT

Il s'agit de préciser, pour chaque action, la localisation des produits, puis des supports mobiles supportant éventuellement ces produits. Les localisations choisies doivent bien entendu apparaître dans le RT correspondant au type de l'élément mobile considéré. Le principe est illustré sur la figure 9.6.

Le formalisme des RdP à prédicats a ici été employé. Nous invitons le lecteur à se reporter à l'annexe A pour une compréhension du mécanisme de pliage d'un RdP consistant à reporter sur des étiquettes associées aux jetons certaines informations de façon à simplifier la structure de réseau.

Dans l'exemple, la signification des variables utilisées dans les prédicats associés aux jetons et aux arcs est la suivante :

- ♦ y désigne un produit du type produit_P,
- ♦ x désigne un emplacement du type ES_produit,
- ♦ x' désigne un emplacement du type ES_palette,
- ♦ z désigne une palette.

Un jeton <x,y,z> dans la place 3 représente l'assertion suivante : l'emplacement x d'une palette z est occupée par le produit y.

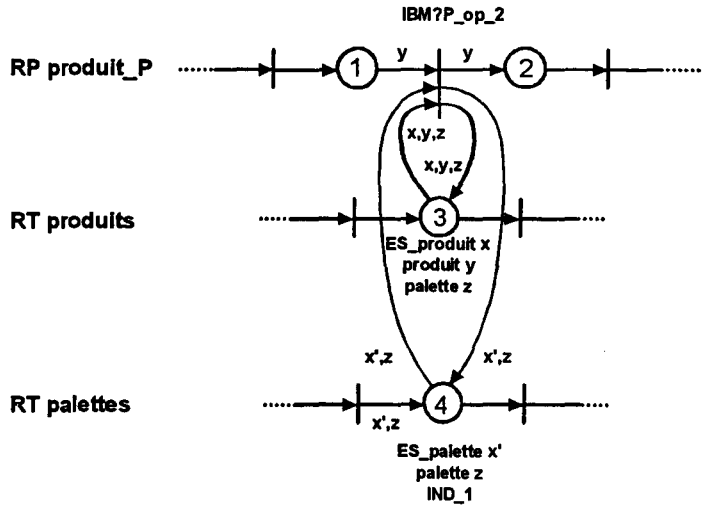


Figure 9.6 : principe de connexion RP/RT.

L'exemple s'interprète donc ainsi : un produit y du type produit_P (jeton <y> dans la place 1) ne peut subir l'opération IBM.op2 que s'il occupe un emplacement x d'une palette z (jeton <x,y,z> dans la place 3), elle-même située sur un emplacement x' de l'indexeur IND_1 (jeton <x',z> dans la place 4).

Phase 2 : connexion RT/RT

Il s'agit ici de préciser, pour chaque action de transfert à partir ou à destination d'un support **mobile**, la localisation du support. Le principe de connexion des graphes est le même que pour la phase 1 (cf. figure 9.7).

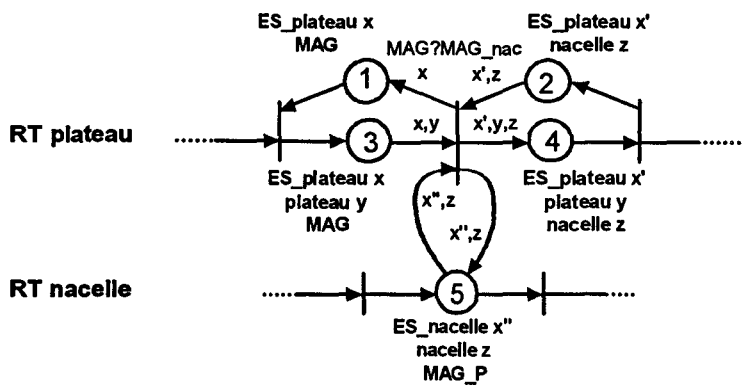


Figure 9.7 : principe de connexion RT/RT.

L'exemple correspond à l'opération de transfert d'un plateau y d'un emplacement x du stockeur MAG (jeton <x,y> dans la place 3) sur un emplacement x' d'une nacelle z (jeton <x',z> dans la place 2). Cette opération entraîne la libération de l'emplacement x (jeton <x> dans la place 1) et l'occupation de l'emplacement x' (jeton <x',y,z> dans la place 4). Bien entendu, la nacelle z doit se trouver sur l'emplacement x" situé face au stockeur (jeton <x",z> dans la place 5).

Phase 3 : connexion RT/RC et RP/RC

Cette phase permet de relier les actions demandées dans les RP et RT aux actions fournies par les RC. Ce couplage peut être réalisé de deux façons différentes : *couplage direct* simple ou *couplage par invocation.*

- Le couplage direct consiste à fusionner ensemble toute paire de transition formée d'une transition étiquetée par `?op_réelle_i` et d'une autre étiquetée par `!op_réelle_i`. Ce mode d'interconnexion nécessite la réplication de transitions demandeuses (dans les RT ou RP) si une même opération peut être effectuée par plusieurs ressources ou dans plusieurs états d'une même ressource. De même, il faut dupliquer les transitions effectrices (dans les RC) si une même opération est demandée plus d'une fois. La figure 9.8 illustre le principe.

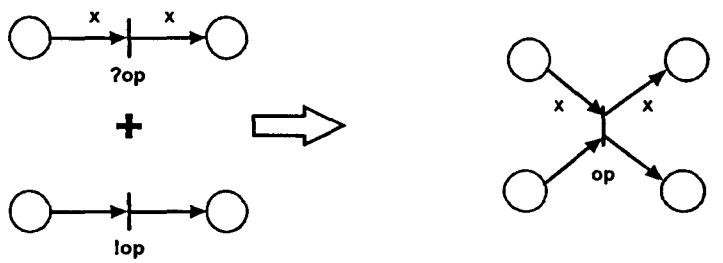


Figure 9.8 : couplage direct.

- Le couplage par invocation avec protocole de type rendez-vous ne nécessite pas de duplication de transitions. Il consiste à remplacer toute transition étiquetée par `?op_réelle_i` par le module de synchronisation de la figure 9.9.a. De façon analogue, les transitions étiquetées par `!op_réelle_i` sont remplacées par le motif de la figure 9.9.b. Le résultat est donné sur la figure 9.9.c.

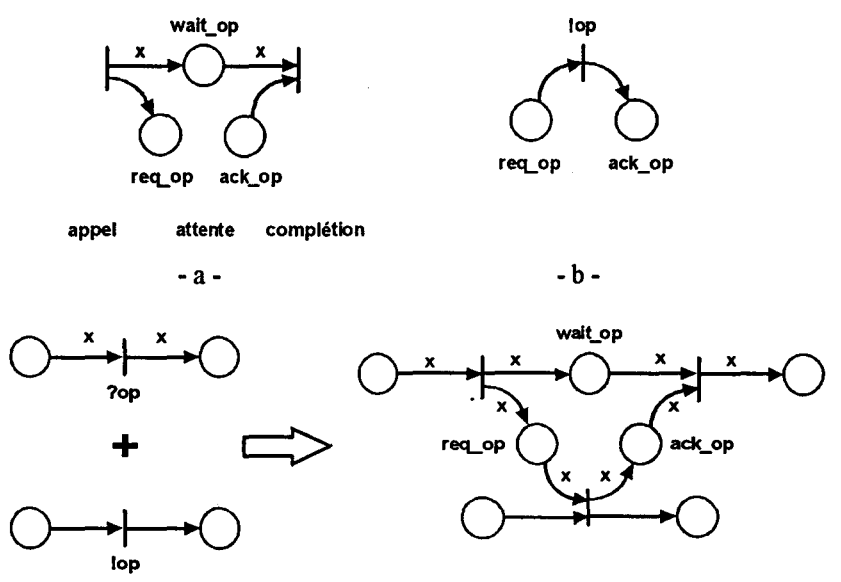


Figure 9.9 : couplage par invocation

Dans le cadre de nos travaux, le couplage direct a été préféré au couplage par invocation pour deux raisons.

Premièrement, le mécanisme de l'invocation nous semble contredire le principe de flexibilité maximale adopté dans notre démarche : le choix d'une opération a lieu avant même que l'ensemble des contraintes conditionnant sa réalisation soit vérifié. Au moment de l'invocation, les ressources affectées à l'opération peuvent en effet se trouver dans un état incompatible avec l'opération demandée. Une requête émise ne pouvant pas être annulée dans le mécanisme proposé, une éventuelle opération concurrente (flexibilité sur l'ordre des opérations par exemple) ne pourra pas être déclenchée.

Deuxièmement, l'invocation suppose une relation client-serveur n'apparaissant pas au niveau des contraintes physiques que l'on cherche à modéliser. Une telle relation apparaît, par contre, lorsque l'on intègre la commande de coordination dans le modèle [Ausfelder, 1994].

La nécessaire duplication de transitions induite par le mécanisme de couplage direct adopté ne pose pas de problème dans le cadre de l'utilisation prévue, à savoir la réalisation d'un outil informatique d'aide à la modélisation. Un algorithme de couplage permet en effet de générer automatiquement, à partir des modèles séparés des produits, des lieux et des ressources, l'ensemble des contraintes existant sur chaque opération.

9.5 Utilisation du modèle

9.5.1 Validation analytique

C'est cette utilisation en tant qu'outil de validation qui a guidé le choix de l'outil de modélisation, à savoir les RdP (cf. § 2.3). Certaines propriétés du système telles que l'absence de blocage mortel (réseau pseudo-vivant), la possibilité de retour à l'état initial (réseau réinitialisable), l'absence de point d'accumulation (réseau borné), ... peuvent en effet être prouvées à partir d'un modèle RdP.

La démarche adoptée, à savoir la modélisation des seules contraintes physiques, avait pour but de permettre la détermination de l'ensemble des situations de blocage "physique", indépendamment des contraintes de coordination spécifiées dans les gammes. Certains mécanismes peuvent être mis en oeuvre afin d'éviter l'occurrence de telles situations : il s'agit essentiellement de restreindre l'accès à certains lieux caractéristiques par ajout de contraintes supplémentaires. Le problème consiste alors à trouver le jeu de contraintes minimal de manière à ne pas pénaliser outre mesure la flexibilité et les performances inhérentes du système.

L'ajout des contraintes de coordination spécifiées par les gammes opératoires peut également générer d'autres situations de blocage. Une étape supplémentaire d'analyse, à partir d'un modèle des gammes, est donc nécessaire.

Les mécanismes de contrôle permettant d'éviter l'apparition de ces situations sont analogues à ceux précédemment évoqués.

Par rapport à notre méthodologie, nous suggérons donc une démarche d'analyse en deux étapes :

- analyse du modèle initial, i. e. sans coordination : cette étude doit permettre de dégager certaines règles permettant d'aider la conception du modèle de coordination utilisé par le pilotage ;
- analyse du modèle étendu, prenant en compte dans le formalisme RdP le modèle des gammes opératoires ; cette extension du modèle de base n'a pas été étudiée dans le cadre de ce travail. Le lecteur intéressé pourra se reporter à [Ausfelder, 1994] où une méthode complète de modélisation et de validation de la commande de coordination est exposée, avec toutefois une compréhension différente de la notion de gamme.

La validation analytique, fondamentale dans le cadre de l'établissement d'outils logiciels d'aide à la conception de la commande des SFPM, constitue l'une des extensions possibles de ce travail. Les travaux de D. Cruette [Cruette, 1991] et C. Ausfelder [Ausfelder, 1994] menés sur ce sujet dans le cadre du projet CASPAIM sont significatifs des résultats qu'il est possible d'obtenir par l'analyse. Ils valent d'être poursuivis et approfondis.

9.5.2 Simulation

A l'inverse des méthodes analytiques, la simulation permet de s'affranchir de la conceptualisation mathématique. Les simplifications ne sont plus aussi contraignantes et le champ d'application des modèles de simulation s'avère plus vaste. Les résultats fournis sont moins spécifiques. Ils nécessitent, par contre, une interprétation afin d'obtenir la réponse à un problème particulier. Néanmoins, ce caractère non spécifique des informations collectées est un atout dans le cadre de l'objectif assigné à la simulation dans ce travail, à savoir la mise au point du paramétrage des divers modules de supervision (cf. § 3.3.3). Les résultats générés peuvent ainsi être utilisés pour l'étude de l'ensemble des fonctions du système de commande.

L'utilisation du modèle RdP proposé dans ce chapitre en vue de la simulation nécessite cependant un certain nombre d'extensions. Il faut y intégrer la dynamique temporelle induite par le procédé, la prise en compte des aléas ainsi que les mécanismes de résolution des indéterminismes.

La résolution des deux derniers points est facilitée par le choix du formalisme des RdP à objets pour l'implantation. Plusieurs paramètres rattachés aux jetons-objets des ressources permettent ainsi de simuler les occurrences de pannes et d'enrayage[1] des moyens de production. De même, l'ajout d'un

[1] L'enrayage d'un équipement ne peut avoir lieu qu'en fonctionnement, à la différence de la panne pouvant survenir à tout moment.

paramètre "état de conformité" sur les jetons-objets des produits permet de tenir compte de la conformité des produits au niveau des transitions-opérations.

La résolution des indéterminismes repose sur les fonctionnalités locales du pilotage que nous établirons dans la partie 3. L'architecture de simulation choisie (cf. annexe D) impose en effet la simulation de l'ensemble des fonctionnalités du niveau local, notamment celles associées au pilotage. Ces fonctionnalités sont déclenchées sur occurrence de certains événements (arrivée d'un mobile à un aiguillage ou à un indexeur) correspondant au franchissement de certaines transitions. Les attributs portés par les jetons permettent de paramétrer les divers traitements associés.

La prise en compte du facteur temps dans le fonctionnement du modèle est obtenue, quant à elle, grâce à la notion de temporisation. Elle consiste à associer une durée à certains éléments du réseau (places, transitions ou arcs). Deux modalités de fonctionnement peuvent être distinguées. Dans les *RdP temporisés*, la durée associée correspond à une durée d'indisponibilité, i. e. les jetons utilisés dans le tir de la transition ne sont visibles dans aucune place pendant un certain délai. Dans les RdP temporels, la durée associée est une durée de sensibilisation, i. e. le tir est instantané mais doit se produire dans un intervalle de temps donné. Diverses extensions ont été proposées, suivant la combinaison (élément du réseau × modalité) adoptée : RdP à transitions temporisées [Ramchandani, 1974], RdP à places temporisées [Sifakis, 1977], RdP à arcs temporisés [Alla *et al.*, 1985], RdP à transitions temporelles [Merlin, 1974], RdP à arcs temporels [Bako, 1990], etc. Le mécanisme retenu dans le cadre de la réalisation informatique présentée en annexe E, consiste à temporiser les transitions : les jetons utilisés dans le tir de la transition n'apparaissent dans les places aval que lorsque la durée portée par la transition s'est écoulée.

Remarque :

Les gammes opératoires, utilisées par le pilotage, sont bâties à partir des gammes logiques des produits. Il est possible d'automatiser cette étape comme le prouve les travaux menés dans le cadre du projet CASPAIM (cf. conception assistée des gammes dans [Cruette, 1991] ou [Ausfelder, 1994]). Les algorithmes utilisés permettent alors de valider systématiquement la cohérence des gammes opératoires vis à vis des contraintes logiques.

Lorsqu'il est prouvé que les gammes opératoires sont cohérentes avec les gammes logiques, les RP deviennent inutiles et peuvent, pour l'utilisation en simulation, ne pas être établis : un paramètre porté par les jetons-objets associés aux produits et géré par les fonctionnalités locales simulées du pilotage suffit pour coder le stade du produit.

Partie 3

Une solution au pilotage des flux

Chapitre 10

Principes généraux de pilotage

Introduction

Ce chapitre analyse les diverses contraintes que doivent respecter les solutions techniques à mettre en oeuvre pour résoudre le pilotage. Deux types de contraintes peuvent être distingués : celles liées au caractère temps réel de l'application, et celles imposées par les utilisateurs du système de supervision.

A l'issue de cette analyse, nous serons en mesure de justifier les divers choix adoptés et de présenter le scénario général de pilotage. L'objectif visé est de concilier autonomie locale et coordination globale.

Dans un dernier paragraphe nous présenterons le découpage de la fonction de coordination globale du pilotage en coordination de la production et coordination des transferts. Les outils proposés dans la suite de ce mémoire se répartiront suivant ce découpage.

10.1 Contraintes sur le pilotage

10.1.1 Contraintes temporelles

Le niveau de pilotage étant le plus bas de la hiérarchie décisionnelle de gestion, il subit les contraintes temporelles les plus strictes. C'est au pilotage que revient en dernier lieu la prise de décision face aux choix se présentant en temps réel dans le système de production.

Les principes de hiérarchisation et de répartition que nous avons retenus (cf. chapitre 2) ont déjà permis de limiter le cadre décisionnel du pilotage. Ces

principes aboutissent au découpage de la fonction pilotage d'un système de production en divers modules associés chacun à un niveau de commande (cf. chapitre 7) : un module d'un niveau global coordonne les modules appartenant au niveau local associé.

Cependant, pour fonctionner correctement, le principe de hiérarchisation-répartition impose que chaque module local possède un degré d'autonomie suffisant. Notamment, le temps de calcul des requêtes se doit d'être inférieur à la durée d'exécution de ces requêtes par les modules locaux. C'est le cas dans les systèmes d'usinage où les temps de cycle, s'échelonnant de la minute à plusieurs heures, autorisent l'émission d'une requête à chaque cycle, i.e. pour chaque produit. Il en est de même dans le cas de l'assemblage manuel. Par contre, dans les systèmes d'assemblage automatisés, les temps de cycle très courts (typiquement de l'ordre de 0.5 à 10 secondes) interdisent l'émission par le niveau supérieur d'une requête vers le niveau inférieur à chaque cycle.

10.1.2 Contraintes utilisateurs

En plus de ces contraintes temporelles, les solutions que nous devons développer pour le pilotage doivent répondre à certains besoins de mise en oeuvre pratique. Nous retiendrons principalement que :

- la panne d'un superviseur ne doit en aucun cas entraver la production ;
- la mise en oeuvre d'un système de supervision doit pouvoir être progressive, certains utilisateurs souhaitant échelonner les investissements en fonction de la montée en cadence.

Il a été constaté que le non-respect de ces contraintes entraîne le rejet systématique du système de supervision de la part des utilisateurs potentiels actuels.

Le respect strict de ces contraintes entre d'évidence en conflit avec le principe de hiérarchisation retenu : les niveaux globaux (implantés dans les superviseurs) sont en effet indispensables pour réaliser la coordination de la production. Un compromis devra être réalisé.

Nous présentons, dans le paragraphe suivant, les principes de pilotage adoptés permettant de satisfaire les contraintes mentionnées.

10.2 Principes retenus

10.2.1 Production par campagne

L'autonomie des modules locaux, imposée par les contraintes temporelles, sera obtenue en rapportant les différentes requêtes émises par chaque niveau à la notion de *campagne de production* : une requête ne concernera plus un seul

produit, mais l'ensemble des produits appartenant à un même lot. En pratique, chaque requête émise par un niveau à destination du niveau inférieur sera affectée d'une quantité "cible" précisant le nombre de produits à traiter.

L'émission de requêtes par les niveaux successifs aboutit ainsi à créer une configuration stable du système de production au cours de laquelle la production peut s'effectuer **sans échange inter-niveaux**. Une configuration stable correspond au fonctionnement permanent entre deux changements de campagne.

La durée des phases stables, et donc l'autonomie des niveaux locaux, est fonction de trois facteurs principaux[1] :

- la taille des campagnes,
- le temps de cycle,
- le nombre de campagnes simultanément possibles.

L'un des objectifs des systèmes flexibles est bien sûr de tendre vers des campagnes simultanées, unitaires et de temps de cycle très petit de façon à accroître la réactivité du système.

Ces divers facteurs doivent être réglés de façon cohérente avec la capacité du système de pilotage en terme de temps de réponse. Les deux derniers facteurs étant fixés à l'issue de la conception, on jouera, lors de l'exploitation, sur le premier facteur afin d'augmenter ou de réduire la durée des phases stables. Plus cette durée sera faible, plus les échanges inter-niveaux relatifs aux changements de campagne (donc nécessaires) seront nombreux. Il peut même arriver que le régime permanent correspondant à la configuration stable ne puisse être atteint (campagnes trop petites et/ou trop nombreuses) : les échanges inter-niveaux sont alors permanents. Un compromis doit ainsi être trouvé entre réactivité (petites campagnes) et mise à contribution du niveau global.

10.2.2 Mise en oeuvre modulaire

La mise en oeuvre pratique de la solution proposée est la suivante : plusieurs modules locaux standardisés fonctionnant de manière autonome, paramétrés par un module de niveau global afin de fonctionner de manière coordonnée. Seuls les modules locaux sont nécessaires pour assurer la production. Ils nécessitent néanmoins un paramétrage qui peut être effectué soit manuellement, soit par l'intermédiaire du superviseur lorsque ce dernier est installé.

La possibilité de configuration manuelle permet de résoudre la contrainte de mise en oeuvre progressive mentionnée ci-dessus. Elle constitue également une alternative en cas de panne prolongée du superviseur. Cependant elle limite les possibilités de changements de configuration : un opérateur doit en effet parcourir l'installation pour paramétrer chaque module réparti entraînant ainsi

[1] D'autres facteurs interviennent tel le temps de préparation des moyens entre deux campagnes successives.

perte de temps et risque d'incohérence. En conséquence, la taille des campagnes doit être augmentée. Cette solution peut suffire en cas de système non flexible dédié à une seule production : la configuration peut alors être effectuée une fois pour toute en déclarant une campagne de taille infinie.

La configuration par superviseur permet d'accélérer les changements de configurations et d'assurer la cohérence de la coordination. Lorsque les possibilités de variantes sont multiples et qu'il n'est pas possible de produire par campagnes de taille suffisante (capacité de stockage limitée), cette solution centralisée est obligatoire.

Il est important de noter que seule la phase de paramétrage nécessite une centralisation ; le principe essentiel reste la délocalisation des traitements temps réel afin de respecter la contrainte de fonctionnement sans superviseur.

10.2.3 Possibilité de prise de décision centralisée

Il est tentant, lorsque l'on dispose d'un superviseur, de tirer parti de la connaissance globale ainsi disponible afin d'affiner certaines décisions en temps réel. Nous avons pris en compte cette possibilité dans la solution proposée. Cependant, sa mise en oeuvre reste optionnelle : en aucun cas le déroulement de la production ne doit être en attente d'une décision du superviseur au cours des phases de production stables.

Le déroulement d'une prise de décision supervisée est le suivant :

- émission d'un message de type "question" par le niveau local ; déclenchement d'un mécanisme de chien de garde ;
- prise en compte et traitement de la question par le niveau global, en temps masqué par rapport à la production ;
- en cas de réponse dans les temps, prise en compte de la réponse supervisée ; sinon mise en oeuvre de la procédure locale par défaut.

Ce mécanisme est donc entièrement à l'initiative du niveau local. Il n'entre pas en conflit avec le principe de répartition retenu. En effet, il n'est pas obligatoire à la fonction principale de coordination du pilotage.

10.3 Scénario général de pilotage

A la lumière des principes présentés ci-dessus, nous pouvons désormais présenter le fonctionnement général du système de pilotage proposé.

La production que le système de production doit effectuer est transmise au pilotage sous forme d'*ordres de fabrication* successifs. Ces ordres (OF par la suite) déclenchent le lancement des campagnes de production. Ce lancement donne lieu à l'émission par le niveau global de requêtes ou *bons de fabrication* à

destination des modules locaux permettant d'une part de **coordonner** les opérations des moyens et d'autre part de paramétrer ces opérations.

Une fois les bons émis pour l'ensemble des moyens participant à la campagne, le système passe alors dans un état stationnaire du point de vue du pilotage. La production se déroule essentiellement sous le contrôle des modules locaux ; seules certaines décisions, sensibles mais non indispensables, sont remontées au niveau global suivant le mécanisme présenté au paragraphe 10.2.3.

Cet état stationnaire s'achève avec l'occurrence de deux types d'événements :

- arrivée d'un autre ordre de production ;
- achèvement de l'exécution d'une requête par un moyen local, i.e. l'ensemble des pièces constitutives de la campagne a subi l'opération spécifiée par la requête.

Lorsqu'un de ces événements survient, le module de pilotage global détermine de nouveaux bons de fabrication à lancer et les paramètres associés. Le lancement de ces bons vers le niveau local aboutira à un nouvel état stationnaire.

10.4 Les deux types de coordination

Le scénario général présenté ci-dessus est indépendant de la nature des moyens associés aux diverses requêtes. Cependant, il est apparu très tôt dans nos travaux que les systèmes de transfert occupaient un statut particulier par rapport aux autres moyens. Ceci pour deux raisons essentielles :

- la commande des systèmes opérants associés est très souvent répartie ;
- ils présentent un certain nombre d'indéterminismes locaux introduits par les diverses flexibilités de routage.

Cette particularité a déjà été observée au niveau du modèle des gammes lorsqu'il s'est avéré illusoire de décomposer chaque transfert en terme d'opérations (cf. § 8.1.2). Nous distinguerons donc ces phases des phases restantes que nous qualifierons de *transformationnelles* en référence à la fonction de transformation qu'elles réalisent sur les produits (les opérations de manipulations affectées aux moyens de type buffer seront néanmoins assimilés à cette classe).

Les phases de transfert nécessitent un paramétrage spécifique afin de permettre la commande (donc la résolution des indéterminismes) des divers systèmes opérants constitutifs des systèmes de transfert concernés. Afin de proposer un système de pilotage complet, nous avons donc été amené à élaborer une solution particulière pour coordonner les opérations intervenant

dans un transfert. Cette solution fait l'objet du chapitre 13. Elle se distingue de la solution développée pour la coordination de la production, objet du chapitre 11 (cf. figure 10.1).

Malgré ce découpage, il n'en demeure pas moins que phases de transfert et phases transformationnelles sont gérées, au niveau coordination de la production, sur le même principe. Ainsi, lorsqu'un système de transfert est installé avec son propre système centralisé de gestion du routage, notre solution pour la coordination des transferts n'a pas d'utilité. Ce système peut alors être considéré comme un moyen classique.

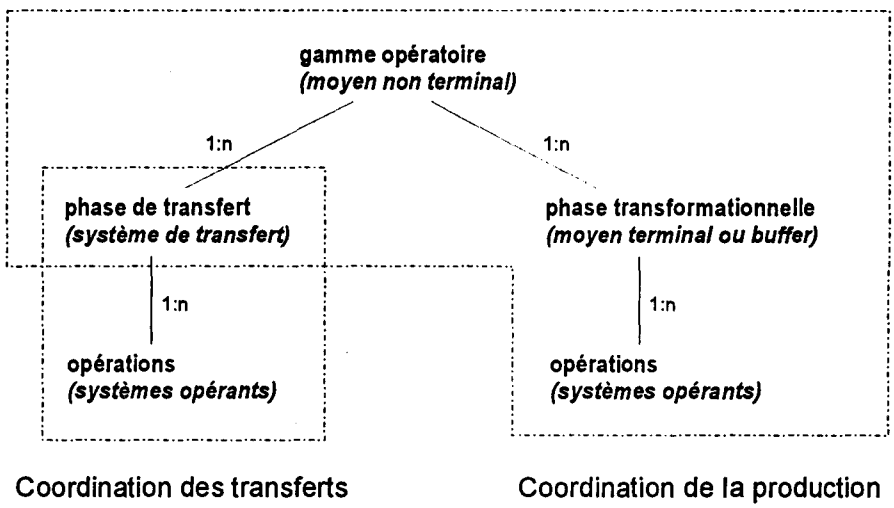


Figure 10.1 : coordination de la production et coordination des transferts.

Chapitre 11

Coordination de la production

Introduction

La coordination de la production constitue la fonction essentielle du pilotage. Elle consiste à distribuer sur chaque moyen, puis à gérer, les charges de travail correspondant à la réalisation des campagnes de production successives.

Cette fonction est réalisée suivant trois phases, répétées pour chaque campagne :

- configuration des moyens ;
- suivi de la production ;
- libération des moyens.

Nous détaillons dans ce chapitre chacune de ces phases.

Deux concepts fondamentaux, issus des méthodes de conception par objets, doivent être maîtrisés pour une bonne compréhension de la méthode. Il s'agit des notions de classes et d'instances. Désormais relativement répandues, nous n'avons pas jugé utile de les expliciter. Le lecteur non familiarisé avec ces concepts pourra se reporter avec profit à [Meyer, 1990] ou [Hill, 1993].

11.1 Principe

Afin d'obtenir une production désirée, un certain nombre d'opérations rassemblées en phases doivent être effectuées par le système de production suivant un séquencement spécifié dans les gammes opératoires. Les phases apparaissant au niveau des gammes sont effectuées par les moyens associés au

niveau de commande auquel on se place. Les relations entre gammes, opérations de phase et moyens sont illustrées sur la figure 11.1.

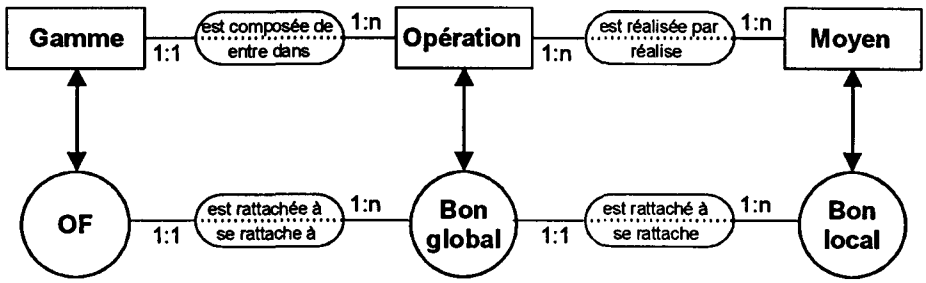


Figure 11.1 : relations entre gammes, phases et moyens et requêtes associées

A chacune des entités du modèle relationnel statique (ligne supérieure de la figure), correspond un type de "requête" spécifiant le travail à réaliser aux niveaux correspondants (ligne inférieure). Au niveau gamme correspond l'ordre de fabrication (OF), au niveau opération de phase[1] le bon de fabrication global, et au niveau moyen le bon de fabrication local.

Les relations entre entités du modèle et "requêtes" sont établies de façon dynamique lors du fonctionnement du système de production. Il s'agit du *processus d'instanciation* qui sera présenté dans le paragraphe 11.3. Ainsi, la relation entre une gamme et un ordre de fabrication n'est-elle effective que lorsque l'ordre est instancié, à savoir que chacun de ses champs de données s'est vu affecter une valeur. Il en va de même pour les relations entre opérations et bons globaux et entre moyens et bons locaux.

Lors de la phase d'instanciation, chaque moyen du système de production concerné par la production se voit affecter une ou plusieurs requêtes indiquant les opérations à effectuer. L'instanciation peut être assimilée à une configuration dynamique du système de production. Chaque campagne nécessite une configuration particulière.

A l'issue de la phase d'instanciation (ou même pendant, suivant le mode de changement de campagne adopté – cf. § 11.6.2), la production peut débuter. Le système passe alors dans une phase stationnaire du point de vue du pilotage, à savoir la phase de *suivi de la production*. Cette phase fait l'objet du paragraphe 11.4.

L'objet du suivi de la production est de détecter au niveau de chaque moyen, les fins de campagnes, matérialisées par l'achèvement des requêtes correspondantes. Un processus de désinstanciation s'engage alors, désactivant les relations dynamiques établies lors de l'instanciation.

---

[1] Il est important de noter que l'on considère bien l'opération comme étant la "brique" élémentaire de la gamme, et non la phase, conformément aux raisons indiquées au chapitre 8.

Un nouveau processus d'instanciation peut alors s'engager ou un ancien être complété en fonction des relations ainsi libérées. En effet, en raison des capacités mémoires limitées des systèmes physiques mis en oeuvre pour réaliser la commande (superviseurs et automates), le nombre d'instances pouvant être associées à chaque classe abstraite de requêtes est fixe. Il peut ainsi arriver qu'une requête ne soit pas instanciable. Dans certains cas, le processus d'instanciation peut alors être interrompu et placé en attente d'une libération d'instances.

11.2 Les différents types de "requêtes"

Nous passons ici en revue les trois classes de requêtes, objets de l'instanciation. Les principaux attributs caractérisant chacune des requêtes sont présentés. Afin de faciliter la lecture, cette présentation ne fait référence à aucun formalisme particulier.

11.2.1 L'ordre de fabrication (OF)

Les instances de cette classe constituent les points d'entrée de la fonction pilotage. Les OF spécifient le produit ou l'article à fabriquer, ainsi que la quantité à atteindre. Un OF se caractérise par :

- un **identifiant**, initialisé de manière unique lors de l'instanciation, permettant de référencer l'OF.
- un **statut** précisant l'état d'avancement de l'OF dans son "cycle de vie". Un OF passe successivement par les états "libre", "ordonnancé", "instanciation des opérations en cours", "instanciation effectuée", "production en cours", "achèvement en cours", "production terminée" avant de revenir à l'état "libre". Ce cycle, ainsi que les statuts successifs sont détaillés en annexe F.
- une **référence produit** précisant le produit (ou plus précisément une instance de la classe abstraite produit) à réaliser. Cette référence permet de retrouver la ou les gammes à "jouer" pour coordonner le système de production.
- une liste de **variabilités instanciées** (cf. § 5.1.6). Ces valeurs permettent, lors du processus d'instanciation, d'extraire du modèle synthétique des gammes utilisé par le pilotage (cf. § 8.1), les seules portions correspondant à la variante demandée.
- une **quantité cible** précisant le nombre d'articles à produire. Au niveau de l'OF, nous appellerons également *consigne* cette quantité.

- un **compteur bons**, initialisé à 0 à l'instanciation, précisant le nombre courant d'articles conformes réalisés.
- un **compteur mauvais**, initialisé à 0 à l'instanciation, précisant le nombre courant d'articles non conformes réalisés.

Les compteurs associés à l'OF permettent un suivi fin de l'état d'avancement de la production au niveau des campagnes (rappelons ici que chaque OF génère une campagne de production).

Remarque : le nombre de compteurs est directement lié au nombre de valeurs adoptées pour caractériser l'état de conformité du produit. Les études de cas menées montrent néanmoins que la solution extrême retenue est suffisante pour les scénarios envisagés.

11.2.2 Le bon de fabrication global

Les instances de cette classe caractérisent le travail à accomplir au niveau de chaque opération constitutive d'une phase de gamme. Un bon de fabrication global se caractérise par :

- un **identifiant**, initialisé de manière unique lors de l'instanciation, permettant de référencer le bon global.
- un **statut** précisant l'état d'avancement du bon dans son "cycle de vie". Un bon global passe successivement par les états "libre", "réservé", "configuration des moyens en cours", "configuration complétée", "production en cours", "production terminée", "déconfiguration en cours" avant de revenir à l'état "libre". Ce cycle, ainsi que les statuts successifs sont détaillés en annexe F.
- une référence à l'**OF associé**. Cette donnée permet de relier l'ensemble des bons globaux d'une campagne à l'OF ayant généré cette campagne. Ce lien sera utilisé, lors du suivi de production, pour mettre à jour l'état d'avancement de l'OF, en fonction de celui de certains bons significatifs (notamment les bons attachés aux opérations initiales et terminales de la gamme).
- une référence à l'**opération de phase associée**. Ce lien, établi lors de l'instanciation, permet d'exploiter les informations statiques attachées aux opérations (ou plus précisément aux instances de la classe abstraite "opération" créées lors de la configuration du système). Ainsi, par exemple, est-il possible de retrouver les variabilités paramétrant l'opération considérée et d'instancier en conséquence les champs correspondants des bons locaux.
- une liste de **bons globaux amont**, correspondant aux opérations situées en amont dans la gamme. Cette liste permet de mettre à jour l'état d'avancement du bon dans son cycle de vie lors du suivi de production.

- une liste de **bons globaux aval**. Cette liste, duale de la précédente, correspond à la même nécessité de connaître l'état d'avancement des bons globaux consécutifs (en amont ou en aval) pour mettre à jour le statut d'un bon global.
- une liste de **bons locaux associés**. Ces bons locaux sont attachés aux moyens affectés à l'opération liée au bon global. Cette liste permet de mettre à jour le statut des bons locaux en fonction de celui du bon global associé.
- une **quantité cible** précisant le nombre d'articles à produire.
- un **compteur bons**, initialisé à 0 à l'instanciation, précisant le nombre courant d'articles conformes réalisés.
- un **compteur mauvais**, initialisé à 0 à l'instanciation, précisant le nombre courant d'articles non conformes réalisés.

Les compteurs associés au bon global permettent un suivi fin de l'état d'avancement de la production au niveau des opérations.

11.2.3 Le bon de fabrication local

Les instances de cette classe spécifient le travail à effectuer au niveau des moyens affectés aux différentes opérations de phases. Un bon de fabrication local se caractérise par :

- un **identifiant**, initialisé de manière unique lors de l'instanciation, permettant de référencer le bon local.
- un **statut** précisant l'état d'avancement du bon dans son "cycle de vie". Un OF passe successivement par les états "libre", "réservé", "production en cours", "production terminée" avant de revenir à l'état "libre". Ce cycle, ainsi que les statuts successifs sont détaillés en annexe F.
- une référence au **bon global associé**. Cette donnée permet de relier le bon local au bon global ayant initié l'instanciation "locale". Ce lien sera utilisé, lors du suivi de production, pour mettre à jour l'état d'avancement du bon global, en fonction de celui des bons locaux attachés.
- un **code d'opération** correspondant à l'opération à réaliser.
- une liste de **variabilités instanciées** paramétrant les opérations.
- un **premier code de prochaine opération en cas de conformité**. Ce code spécifie la prochaine opération à réaliser sur le sous-ensemble principal lorsque l'opération courante s'est correctement déroulée.
- un **second code de prochaine opération en cas de conformité**. Ce code spécifie la prochaine opération à réaliser sur le sous-ensemble secondaire lorsque l'opération courante s'est correctement déroulée et

aboutit à deux sous-ensembles distincts (opération de type désassemblage).

- un **code de prochaine opération en cas de non conformité**. Ce code spécifie la prochaine opération à réaliser lorsque l'opération courante ne s'est pas correctement déroulée, i.e. le degré de conformité du produit ne correspond pas au degré attendu au stade où se trouve le produit.
- un **code requête** permettant d'interroger en temps réel le superviseur. Ce code permet notamment de mettre en oeuvre le mécanisme optionnel de routage par règles de destination présenté au paragraphe 13.3.
- un **signal booléen ON/OFF** contrôlant l'activité du moyen associé au bon local. L'utilisation de ce signal sera détaillée au paragraphe 11.6.
- une **quantité cible** précisant le nombre d'articles à produire.
- un **compteur bons**, initialisé à 0 à l'instanciation, précisant le nombre courant d'articles conformes réalisés.
- un **compteur mauvais**, initialisé à 0 à l'instanciation, précisant le nombre courant d'articles non conformes réalisés.

Les divers codes, ainsi que la liste des variabilités sont instanciés par recopie des informations correspondantes disponibles au niveau du bon global ou de l'OF associés pour les données dynamiques (valeurs des variabilités), ou au niveau du modèle des gammes opératoires pour les données statiques (codes de prochaines opérations).

Les compteurs associés au bon local permettent un suivi fin de l'état d'avancement de la production au niveau des moyens.

11.3 Le processus d'instanciation

L'instanciation, dans l'approche objets, consiste à affecter des valeurs aux différents attributs définissant une classe abstraite de façon à obtenir un élément ou instance de cette classe. Nous nous intéressons ici au processus d'instanciation des classes abstraites correspondant aux trois types de requêtes décrites précédemment. Ce processus s'effectue dynamiquement lors de la *phase d'exploitation* du système de production. Un autre processus d'instanciation existe, correspondant à la création des instances relatives aux classes abstraites caractérisant le système, notamment les gammes, les opérations et les moyens. Ce processus a lieu lors de la *phase d'initialisation* des modules de supervision (cf. annexe D).

Le processus d'instanciation des requêtes se décompose en trois étapes, détaillées ci-après.

11.3.1 Etape n°1 : instanciation d'un ordre de fabrication

Les OF sont instanciés par la fonction ordonnancement à partir des listes de commandes client prévisionnelles ou confirmées. Ils sont transmis à la fonction pilotage (lancement en production) dès que l'ensemble des conditions, essentiellement temporelles, fixées par les niveaux supérieurs sont atteintes.

Cette étape n'est pas de la responsabilité du pilotage. Par contre, dès qu'un OF est disponible, cette fonction prend en charge les deux phases suivantes.

11.3.2 Etape n°2 : instanciation des bons de fabrication globaux

Affectation des bons aux opérations

A partir de la référence produit fournie par l'OF, le pilotage est en mesure de retrouver la gamme opératoire à suivre[2]. Un parcours du graphe associé, à partir des opérations initiales, permet de passer en revue chaque opération de la gamme et de lui associer un bon global (cf. figure 11.2).

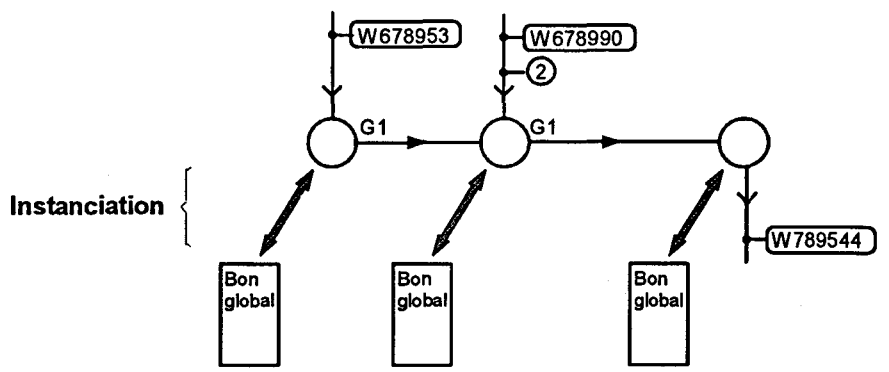


Figure 11.2 : instanciation des bons de fabrication globaux.

L'algorithme mis en oeuvre doit tenir compte des conditions sur les variabilités associées aux arcs. Les valeurs des variabilités, fournies par l'OF, permettent de discriminer certaines portions du graphe. Nous présentons ci-dessous l'algorithme que nous avons mis en oeuvre. Il utilise une méthode de parcours en profondeur.

G désigne le graphe représentant la gamme. L'association des bons globaux à chaque opération est effectuée par la procédure `visiter`. Dans la cas où tous les bons disponibles seraient déjà affectés, il est nécessaire d'interrompre le parcours dans l'attente de la libération d'un bon. La

---

[2] L'ensemble des gammes connues du pilotage est défini au cours de la phase de configuration du système de supervision (cf. annexe D).

mémorisation du contexte permettra alors de reprendre l'affectation au point interrompu.

```
Procédure affecter les bons globaux
  Tant qu'il existe un sommet de G non visité
    Choisir un sommet non visité s
    Parcourir le graphe depuis s
  Fintantque

Procédure parcourir le graphe depuis s
  Visiter s
  Pour tout arc aval (s,x) issu de s
    Si la condition sur les variabilités est vérifiée
      Si le sommet x est non visité
        Visiter x
        Parcourir le graphe depuis x
      Finsi
    Finsi
  Finpour

Procédure visiter s
  Si il existe un bon global b non affecté
    Affecter b à s
  Sinon
    Interrompre affectation et mémoriser contexte
  Finsi
```

Initialisation des quantités cibles

La procédure précédente a permis de créer les bons de fabrication globaux et de les lier les uns aux autres à partir du graphe représentant le modèle des gammes. On obtient ainsi un graphe, calqué sur le modèle du graphe des gammes. Ce graphe caractérise **une** campagne donnée de production (celle associée à l'OF).

Il s'agit maintenant d'initialiser les valeurs de l'attribut "quantité cible" des bons globaux ainsi obtenus à l'aide de la consigne spécifiée dans l'OF. En effet, ces valeurs ne sont pas initialisées lors de la procédure d'affectation : il est nécessaire de connaître l'ensemble des relations mutuelles entre bons globaux afin d'initialiser correctement la valeur cible. Ces relations sont spécifiées par l'utilisateur par l'intermédiaire des *graphes d'influence* que nous présenterons en détail au chapitre 12. De façon analogue au graphe des gammes, le graphe d'influence est reproduit lors de l'affectation des bons globaux. Un parcours du graphe ainsi obtenu, à partir du point d'application choisi pour appliquer la consigne (cf. chapitre 12), permet de diffuser cette consigne à travers l'ensemble des bons globaux attachés à la campagne.

11.3.3 Etape n°3 : instanciation des bons de fabrication locaux

Dès que l'ensemble des bons globaux associés aux opérations a été initialisé, débute l'instanciation des bons locaux. Il s'agit, pour chaque bon

global précédemment créé, de lui associer un ou plusieurs bons locaux à destination des divers moyens choisis pour réaliser l'opération correspondante (cf. figure 11.3).

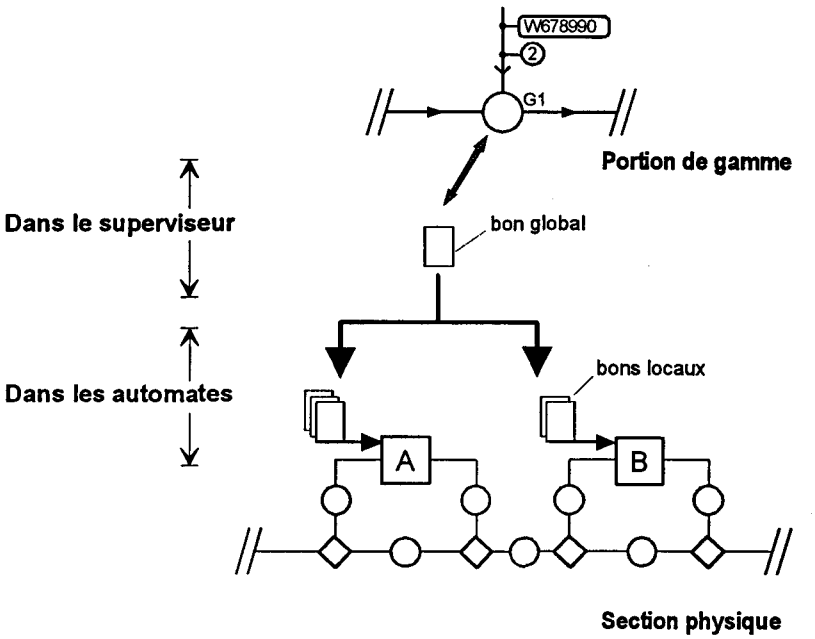


Figure 11.3 : instanciation des bons de fabrication locaux.

Cette phase est réalisée par un parcours du graphe formé par les bons globaux instanciés. L'algorithme mis en oeuvre dérive simplement de celui présenté ci-dessus :

```
Procédure instancier bons locaux
  Tant qu'il existe un sommet de G non visité
    Choisir un sommet non visité s
    Parcourir le graphe depuis s
  Fintantque

Procédure parcourir le graphe depuis s
  Visiter s
  Pour tout arc aval (s,x) issu de s
    Si le sommet x est non visité
      Visiter x
      Parcourir le graphe depuis x
    Finsi
  Finpour

Procédure visiter s
  Pour tout moyen m pouvant réaliser l'opération associée
    Si il existe un bon local b non instancié
      Instancier b à partir de s
      Lever drapeau
    Sinon
      Placer s sur la file des opérations en attente sur m
    Finsi
  Finpour
  Si drapeau bas
    Interrompre instanciation et mémoriser contexte
  Finsi
```

La procédure "visiter s" essaie d'instancier les bons locaux des moyens capables de réaliser l'opération associée au bon global examiné. Si au moins un des moyens a pu être configuré, la procédure continue, sinon elle s'interrompt.

Une fois les bons locaux chargés au niveau des automates contrôlant les moyens, ceux-ci sont donc théoriquement capables d'assurer la production demandée. Cependant, les interruptions du processus, contrairement à la phase 2, doivent ici être contrôlées avec soin. En effet, les premières opérations peuvent débuter alors même que les moyens associés aux phases suivantes ne sont pas encore configurés. Cette situation peut être inadmissible dans le cas de certains procédés.

La *signalisation*, présentée au paragraphe 11.6, résout ce problème en fournissant un moyen de contrôle de l'accès aux lieux de l'installation. Il est ainsi possible d'interdire aux objets d'une campagne de pénétrer dans une zone donnée si cette zone n'a pas été configurée pour les recevoir.

11.4 Le suivi de la production

Suivre la production consiste à mettre à jour l'évolution des différentes requêtes appartenant à une même campagne les unes par rapport aux autres. L'évolution d'une requête est caractérisée par l'évolution des valeurs de ses compteurs (compteurs "cible", "bons" et "mauvais").

Les compteurs associés aux requêtes relatives à une même campagne de production ne sont pas indépendants entre eux. Trois types de relations peuvent être distinguées.

11.4.1 Relations entre un OF et les bons globaux associés

Les compteurs associés à un OF sont mis à jour par rapport aux opérations de sortie de gamme. Deux cas sont possibles (cf. figure 11.4) :

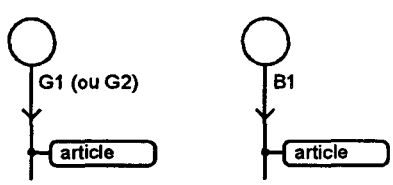


Figure 11.4 : les deux cas d'opérations de sortie.

- celles amenant l'article au stade final dans l'état de conformité demandé ; l'évolution du compteur "bon" associé à ces opérations doit être répercutée sur le compteur correspondant de l'OF.

- celles occasionnant des rebuts, le stade courant pouvant être différent du stade final ; l'évolution du compteur "mauvais" associé à ces opérations doit être répercutée sur le compteur correspondant de l'OF.

11.4.2 Relations inter bons globaux

Ce type de relations intervient lorsque les aléas apparaissant au niveau d'une opération font l'objet d'une détection, permettant ainsi de distinguer les pièces défectueuses (compteur "mauvais"). Deux cas se présentent alors : soit la pièce est retouchée, soit elle est rebutée. Dans le second cas, une *stratégie de compensation* peut avoir été prévue afin de compenser la perte et d'atteindre malgré tout la consigne fixée. Dans les deux cas, les valeurs des compteurs "cible" doivent être ajustées.

Cas d'une boucle de retouche

La figure 11.5 illustre les deux branches divergentes d'une boucle de retouche.

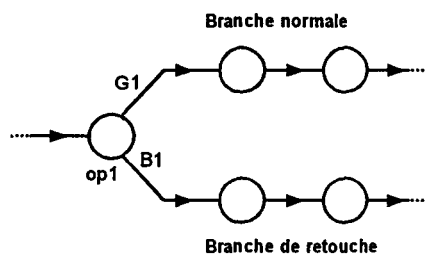


Figure 11.5 : branches divergentes d'une boucle de retouche

A l'issue de l'opération "op1", le sous-ensemble traité est dirigé soit sur la branche normale, soit sur la branche de retouche, en fonction de son état de conformité. Les opérations de la branche non choisie ne seront donc pas effectuées sur ce produit. La quantité "cible" associée à ces opérations doit donc être diminuée d'une unité.

Cas d'un rebut

La figure 11.6 illustre une opération "op1" pouvant occasionner des rebuts.

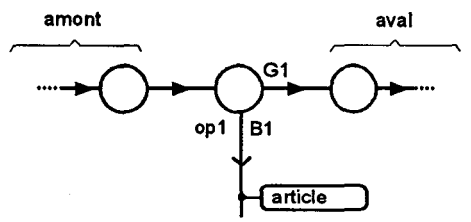


Figure 11.6 : sections amont et aval d'une opération occasionnant un rebut.

Les pièces défectueuses sont alors perdues. Sans stratégie de compensation, la quantité "cible" des opérations aval à "op1" ne sera donc pas atteinte. Elle doit donc être diminuée de la quantité de pièces rebutées. Dans le cas où une stratégie de compensation des pertes est mise en oeuvre, il s'agit par contre d'augmenter les compteurs amont, afin de traiter le nombre correspondant de pièces bonnes réintroduites.

Les relations inter bons globaux sont spécifiées par l'intermédiaire des *graphes d'influences*. Ces graphes, déjà utilisés lors de l'étape n°3 du processus d'instanciation, permettent de définir les règles d'ajustement des quantités "cible" des bons globaux en fonction des stratégies de compensation choisies, et de la structure de la gamme opératoire (nature des boucles de retouche). Nous reviendrons en détail sur les graphes d'influence au chapitre 12.

11.4.3 Relations entre un bon global et les bons locaux associés

A un bon global correspond un ou plusieurs bons locaux affectés aux moyens chargés de réaliser l'opération de gamme associée à ces bons. Lorsque l'un des moyens a réalisé une opération sur l'un des produits de la campagne en cours, les compteurs du bon local correspondant sont mis à jour. Cette évolution doit être répercutée sur le bon global associé, puis sur les autres bons locaux suivant le schéma suivant :

1 - évolution positive d'un des compteurs "bon" ou "mauvais" d'un bon local ;

2 - mise à jour du compteur correspondant du bon global ;

3 - évolution négative des compteurs "cible" des autres bons locaux associés.

Ces relations traduisent le fait qu'un produit traité au niveau d'un moyen ne doit plus être attendu au niveau des autres moyens associés à la même opération.

La mise en oeuvre pratique de ces relations doit tenir compte, à l'inverse des types précédents, de la répartition géographique des différents bons locaux. Ceux-ci se situent dans les différents automates contrôlant les moyens, alors que les bons globaux sont traités dans le superviseur.

Plusieurs paramètres doivent être pris en compte dans le choix de la solution de communication à mettre en oeuvre :

- capacité du réseau local ;
- temps de cycle du procédé contrôlé ;
- temps de traitement du superviseur ;
- nombre d'automates répartis.

Deux stratégies sont alors envisageables, selon que les contraintes sont faibles ou fortes.

Mise à jour synchrone

La *mise à jour synchrone*, par mécanisme d'interruption, des divers compteurs suppose que le temps de réponse du réseau soit nettement inférieur au temps de réponse du procédé, afin d'absorber en temps réel l'ensemble des messages générés à chaque opération. L'avantage de cette solution est de maintenir les divers compteurs associés cohérents entre eux à tout instant.

Scrutation asynchrone

La *scrutation asynchrone*, à l'initiative du superviseur, des divers compteurs locaux permet de découpler les temps de réponse du réseau de ceux du procédé. La figure 11.7 illustre le principe de la scrutation de deux bons locaux associés au même bon global.

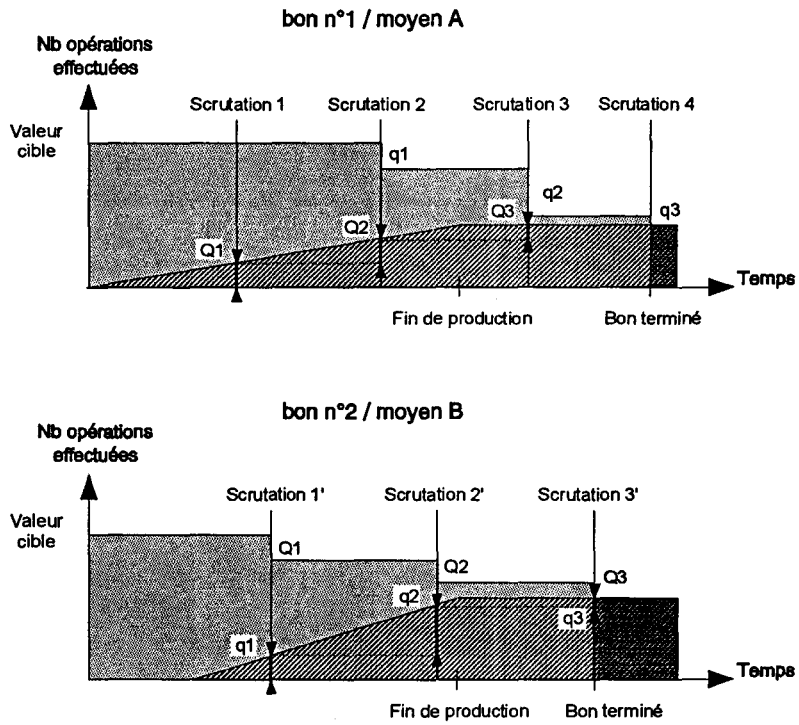


Figure 11.7 : principe de la scrutation asynchrone.

Cette solution entraîne une sous-estimation au niveau du superviseur des quantités réellement produites. Une attention particulière doit donc être accordée à l'approche des fins de campagne, afin de ne pas produire un excédent de pièces au niveau des opérations d'entrée de gamme. Plusieurs solutions sont possibles : accélérer le rythme des scrutations, passer en mode synchrone par interruption ou désengager certains moyens de façon à ce qu'un seul moyen réalise les dernières pièces de la campagne. Seules les opérations d'entrée de gamme doivent faire l'objet d'une telle attention ; ce sont elles qui provoquent l'arrivée des pièces dans l'installation (comportement actif). Les opérations suivantes se déclenchent lorsque les pièces se présentent devant les moyens chargés de les réaliser (comportement passif) : il n'est donc pas possible de les réaliser en excédent.

11.5 Fin de campagne et désinstanciation

Une requête sera dite *terminée* lorsque le nombre de produits traités correspond à la quantité cible spécifiée. Cela se traduit par l'équation suivante :

$$CIBLE = BON + MAUVAIS$$

où CIBLE, BON et MAUVAIS représentent les valeurs des champs correspondants de la requête.

Cependant, une requête terminée n'est pas automatiquement détruite. Nous verrons, en effet, que le compteur "cible" peut être modifié (de façon positive obligatoirement !) dans certains cas : modification de la consigne par l'utilisateur ou influence d'une autre partie de la gamme spécifiée dans le graphe d'influence (cf. chapitre 12) permettant ainsi de relancer l'opération associée.

Un certain nombre de conditions doivent être vérifiées pour détruire (ou désinstancier) la requête. Ces conditions dépendent de son type :

- un OF terminé ne peut être détruit que s'il est acquitté par l'utilisateur et si les bons globaux associés sont eux-mêmes détruits ; précisons ici que l'acquittement d'un OF ne peut s'effectuer que si le bon global associé à l'*opération cible principale* est lui-même terminé. La notion d'opération cible principale sera précisée au chapitre 12.
- un bon global terminé ne peut être détruit que si l'OF correspondant est acquitté et si les bons locaux qui lui sont attachés sont eux-mêmes détruits ;
- un bon local terminé ne peut être détruit que si le bon global qui l'a créé est acquitté.

Ces conditions se retrouvent sur les graphes décrivant le cycle de vie de chaque requête (cf. annexe F).

La désinstanciation d'une campagne débute par la désinstanciation des premières opérations de la gamme puis se propage le long de la gamme en direction des opérations terminales. On parle alors d'*onde de fin de campagne* par opposition à l'*onde de démarrage* accompagnant le processus d'instanciation. L'onde de démarrage accompagne généralement l'onde de fin d'une campagne précédente. Néanmoins, les modalités d'accompagnement varient en fonction de la flexibilité de l'installation liée à la simultanéité de production[3]. Ainsi, suivant les installations, les ondes peuvent-elles se "chevaucher" (cas des installations flexibles ou "job shop") ou se "suivre" (cas des installations arborescentes ou "flow shop"). Dans le premier cas, lorsque plusieurs ondes de démarrage se développent en même temps, une attention particulière doit être accordée afin d'éviter les blocages de production par "étreinte fatale" (une campagne ayant réservé un moyen A attend la libération d'un moyen B réservé par une seconde campagne, elle-même en attente de libération du moyen A).

[3] La capacité des organes de commande de chaque moyen en terme de bons locaux permet de caractériser cette flexibilité : elle donne en effet le nombre maximum de campagnes simultanément possibles sur un moyen.

Nous donnons, au paragraphe 11.6, les modes de réservation possibles des moyens pour une campagne donnée. Quant aux modes de libération, ils dépendent du *point d'application de la consigne (PAC)* dans la gamme. Nous reviendrons sur ce dernier point au chapitre 12.

11.6 Signalisation et coordination de la production

11.6.1 Principe de la signalisation

Le principe de la signalisation est simple : l'opération associée à une requête donnée ne peut s'effectuer que si le signal associé est dans l'état ON.

La signalisation permet en contrôlant l'activité des requêtes de gérer les flux physiques circulant dans l'installation. Ainsi, par exemple, si la requête associée à une opération de palettisation est désactivée, le flux palette+pièce résultant n'est plus généré.

Nous n'avons, jusqu'à présent associé l'attribut "signal" qu'aux bons de fabrication (globaux et locaux) dans le but principal de gérer correctement les changements de campagne. Rien cependant n'empêcherait d'associer également cet attribut aux OF : cela permettrait, en répercutant la valeur du signal sur les bons associés, de geler momentanément une campagne tout en conservant la configuration associée à cette campagne au niveau des automates.

Remarque :

Ce dernier résultat pouvant également être obtenu en abaissant la *consigne*, nous n'avons pas jugé utile de prendre en compte cet attribut dans la réalisation logicielle effectuée dans le cadre de ce travail (cf. annexe F).

11.6.2 Contrôle des changements de campagne

L'utilisation de la signalisation au niveau des changements de campagne a pour objectif de gérer proprement l'avancement du flux de pièces d'une campagne donnée à travers l'installation, en évitant toute interaction néfaste avec les autres campagnes en cours. Il convient, en effet, d'empêcher l'arrivée des pièces d'une campagne donnée sur un moyen tant que ce moyen n'a pas été configuré pour les recevoir.

Cela est obtenu en instanciant les bons de fabrication avec l'attribut signal placé à OFF. Les bons passent alors dans l'état "réservé" signifiant que les conditions permettant de lancer les opérations associées ne sont pas encore satisfaites. Dès que ces conditions sont remplies, le signal est placé à ON, activant ainsi la requête associée ; le moyen associé peut alors recevoir (ou

lancer si le moyen est "actif" et réalise une opération d'entrée de gamme) les pièces de la campagne.

L'activation n'est pas, à priori, simultanée pour l'ensemble des bons associés à une même campagne. Elle peut en effet être progressive. Plusieurs stratégies sont possibles ; elles dépendent de l'installation et des choix d'exploitation réalisés par l'exploitant. Nous présentons, ci-après, les deux cas extrêmes.

Transit souple à l'activation

Cette stratégie tire son origine d'une analogie ferroviaire développée par E. Verdebout à l'EPFL [Verdebout, 1992]. Dans le monde ferroviaire, la réservation des portions de voie est rendue nécessaire pour des raisons de sécurité : deux convois ne doivent en aucun cas occuper au même instant une même zone sous peine de collision.

Les voies sont donc découpées en *cantons* ne pouvant contenir normalement, à un instant donné, qu'un seul train. Chaque canton est contrôlé par un *signal d'arrêt* ou *sémaphore* empêchant l'entrée d'un convoi dans ce canton en cas d'occupation par un autre convoi.

Ce système permet de maintenir une distance de sécurité égale à la longueur d'un canton entre deux trains se suivant sur une portion de voie linéaire. En cas de présence d'aiguillages sur le canton, l'utilisation de la signalisation permet de n'autoriser l'entrée du canton qu'à partir du moment où tous les aiguillages sont correctement configurés pour l'itinéraire choisi. La figure 11.8 illustre cette situation : un convoi provenant de A et désirant se rendre vers G ne sera autorisé à pénétrer sur B que si l'aiguillage e est configuré pour cet itinéraire, interdisant par là même l'entrée sur E de tout convoi provenant de D.

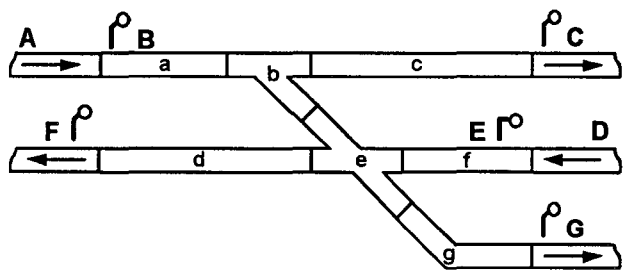


Figure 11.8 : présence d'aiguillages sur un canton de voie ferrée.

Il est à noter que l'aiguillage e peut appartenir soit au canton contrôlé par B, soit à celui contrôlé par E, suivant l'itinéraire enclenché.

Cependant, ce système n'est pas optimal : un second convoi ayant pour itinéraire D→F ne peut pénétrer sur le canton E que lorsque le convoi d'itinéraire A→G a quitté le canton B et est parvenu sur le canton G. Or il suffirait de détecter le passage du premier convoi au niveau de l'aiguillage e pour pouvoir enclencher E.

Pour résoudre ce problème, les cantons ont eux-mêmes été découpés en *zones de grill* (a, b, c ... sur l'exemple), équipées chacune d'un système de capteurs permettant de déterminer le passage des convois. Il devient ainsi possible de libérer chaque zone successivement et d'affecter ces zones à un autre itinéraire sans attendre la libération totale du canton : c'est le principe du *transit souple ferroviaire*. En libérant les zones au plus tôt, cette stratégie optimise donc le nombre de convois susceptibles d'emprunter les cantons se partageant les zones.

Si un signal était affecté à chaque zone, il serait également possible d'appliquer ce principe à la réservation : un convoi provenant de D pourrait ainsi s'avancer sur la zone f et s'immobiliser devant le signal contrôlant l'entrée de la zone e. Dans le monde ferroviaire, cette solution n'est pas possible en pratique : les distances de freinage sont trop importantes.

Dans le cas des systèmes de production, ces contraintes de freinage peuvent être négligées et le transit souple peut s'appliquer à la fois à l'activation et à la libération.

Appliquée à l'activation, le choix de cette stratégie doit cependant être laissée à l'initiative de l'exploitant. Le transit souple à l'activation équivaut en effet à activer les bons au niveau d'un moyen dès que les bons au niveau des moyens suivants ont été réservés. L'activation d'un bon peut donc avoir lieu avant que la totalité des bons associés à la campagne ait été réservée. La production peut ainsi débuter avant même que l'ensemble des moyens ne soit configuré. Cette situation peut s'avérer inadmissible pour certains procédés (ceux notamment qui nécessitent de longues phases de réglage). Lorsqu'elle peut être appliquée, cette stratégie permet cependant d'optimiser l'utilisation des moyens en diminuant les phases improductives.

Par contre, cette stratégie s'applique systématiquement à la libération dès que la consigne est atteinte au niveau du PAC (cf. chapitre 12) : puisque la consigne est atteinte il est en effet inutile de maintenir les bons locaux dans les automates. Rappelons ici que si le moyen contrôlé est configuré en mode flow shop (ordonnancement local – cf. § 7.2.1), il ne peut passer au bon de fabrication suivant que lorsque le bon courant est libéré. Maintenir inutilement des bons locaux dans les automates équivaut dans ce cas à retarder le passage du lot suivant.

Activation après réservation complète

Cette stratégie consiste à ne déclencher le processus d'activation que lorsque la totalité des bons associés à la campagne a été réservée. Elle permet donc d'assurer, à l'inverse du transit souple, que la production ne débutera pas tant que tous les moyens concernés ne sont pas correctement configurés.

D'autres stratégies, intermédiaires entre ces deux cas extrêmes, sont possibles. Néanmoins leur utilité pratique demanderait à être validée.

11.7 Fonctionnalités locales et globales

Nous énumérons ici les fonctionnalités du module de pilotage liées à la fonction de coordination de la production.

Nous avons distingué celles appartenant au niveau global de celles réalisées au niveau local. Ces deux types peuvent d'ailleurs faire l'objet de deux sous-modules distincts implantés ou non suivant le niveau. Ainsi, seules les fonctionnalités locales sont nécessaires dans les automates. Quant à un superviseur, il devra recevoir les deux types de fonctionnalités si le niveau qu'il est chargé de coordonner est lui-même local vis à vis d'un autre niveau global.

Le sous-module du niveau global a fait l'objet d'un prototype logiciel (module XFLOW) développé dans le cadre de ce travail. Une présentation succincte de ce logiciel peut être trouvée en annexe F.

Niveau global	♦ gestion des changements de campagne - instanciation des bons de fabrication - désinstanciation des bons de fabrication - gestion de la signalisation
	♦ suivi des bons de fabrication - scrutation synchrone ou asynchrone des bons locaux - mise à jour des compteurs des bons et des OF - calcul des nouvelles consignes cibles - affichage de l'état des OF et des bons - détection des fins de campagne
Niveau local	♦ mise à jour des compteurs des bons locaux
	♦ mise à jour des informations associées aux objets : - item stade courant - item état de conformité courant - item prochaine action
	♦ application de la signalisation

Fonctionnalités liées à la gestion de la production.

Chapitre 12

Utilisation des graphes d'influence

Introduction

Nous présentons dans ce chapitre le mécanisme permettant de définir les relations entre les compteurs des **bons de fabrication globaux** (compteurs de bons et mauvais produits) et les cibles de ces bons.

Nous avons vu, au chapitre précédent, que ces relations ne peuvent pas toutes être déduites du modèle des gammes. En effet, certaines sont à la charge de l'utilisateur. C'est le cas notamment de celles spécifiant la réaction du système à la perte d'un sous-ensemble suite à une mauvaise opération : diverses stratégies de compensation peuvent être adoptées en fonction de facteurs tels la nature du procédé ou la politique de production désirée.

Il a donc fallu élaborer un nouveau modèle permettant à l'utilisateur de représenter les influences mutuelles des bons globaux : il s'agit des graphes d'influence.

12.1 Présentation générale

Les influences entre bons globaux sont spécifiées par ajout d'arcs étiquetés sur le modèle des gammes utilisé pour le pilotage. Le graphe constitué de ces arcs est appelé *graphe d'influence*. Contrairement au graphe représentant les gammes, les arcs du graphe d'influence ne correspondent à aucun flux physique circulant dans l'atelier.

Les arcs utilisés pour définir le graphe d'influence portent les étiquettes suivantes :

- **TT** : (Target to Target) influence de cible à cible ;
- **GT+** : (Good to Target positive) influence positive du compteur de bons produits sur la cible
- **GT-** : (Good to Target negative) influence négative du compteur de bons produits sur la cible
- **BT+** : (Bad to Target positive) influence positive du compteur de mauvais produits sur la cible ;
- **BT-** : (Bad to Target negative) influence négative du compteur de mauvais produits sur la cible,

Toutes les influences possibles entre deux bons globaux peuvent être définies par ces cinq étiquettes.

L'existence d'un arc étiqueté entre deux noeuds-opérations doit être interprétée de la manière suivante : par exemple, pour GT+, "lorsque le compteur de bons produits de l'opération amont augmente de n produits, alors la valeur cible de l'opération aval doit être augmentée de n produits".

Le problème se pose maintenant de construire le graphe d'influence associé au modèle des gammes d'un article donné. Ce problème peut se révéler complexe dans certains cas (multiples boucles de retouches et/ou d'opérations engendrant des rebuts). Une méthode s'avère donc nécessaire.

Avant d'exposer la méthode que nous proposons, il est important de préciser le principe directeur de notre démarche.

Rappelons, tout d'abord, qu'un bon de fabrication est terminé quand la cible est égale à la somme des compteurs de bons et mauvais produits :

$$\text{CIBLE} = \text{BON} + \text{MAUVAIS}$$

Seuls les bons de fabrication terminés peuvent être détruits lors du processus de désinstanciation. Il faut donc s'assurer que le critère de terminaison puisse être vérifié pour chaque bon afin de ne pas bloquer l'onde de fin de campagne associée. Un tel blocage serait susceptible de se répercuter sur les ondes de démarrage des campagnes suivantes et donc de ralentir, voire d'arrêter, la production.

Les relations d'influence nous permettent d'agir sur les différents termes de l'équation. Il suffit donc d'établir ces relations de façon à assurer l'égalité de l'équation lorsque le dernier produit de la campagne en cours a subi l'opération spécifiée par le bon de fabrication. La méthode proposée est bâtie sur ce principe.

12.2 Illustration de la méthode : exemple 1

Nous nous proposons, dans ce paragraphe, d'illustrer la démarche de construction d'un graphe d'influence sur l'exemple de la figure 12.1.

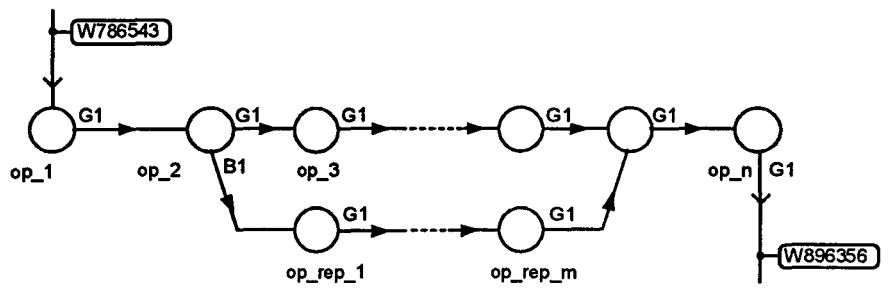


Figure 12.1 : exemple de processus opératoire présentant une gamme de retouche "avant" (exemple 1).

Dans cet exemple, la gamme principale correspond aux opérations "op_i". Une gamme de retouche "avant", formée des opérations "op_rep_i", est également spécifiée en cas de défaut au niveau de "op_2".

Nous utiliserons, dans la suite de ce paragraphe, le formalisme illustré sur la figure 12.2.

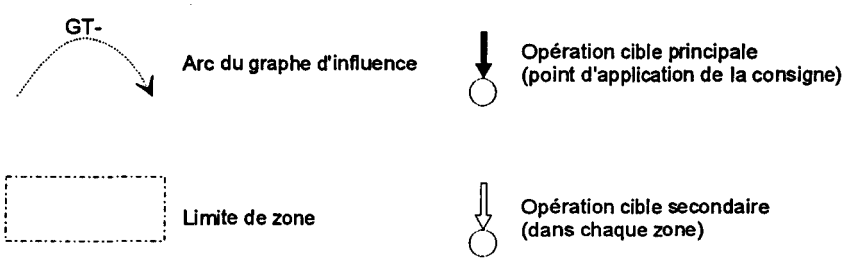


Figure 12.2 : formalisme utilisé pour la construction des graphes d'influence.

12.2.1 Choix du point d'application de la consigne

Il est nécessaire de déterminer, en premier lieu, quelle sera l'*opération cible principale*. Cette opération définit le *point d'application de la consigne (PAC)* [Verdebout, 1992] : la valeur cible choisie pour cette opération servira de référence pour la mise à jour des cibles des bons affectés aux autres opérations.

L'opération cible principale est importante ; elle est le point d'entrée de la mise à jour de toutes les valeurs cibles des bons de fabrication globaux lors du

processus d'instanciation (cf. § 11.3.2) ou d'un changement de la consigne par l'utilisateur. L'ensemble des influences sera en effet défini de façon à **satisfaire** cette consigne. Un corollaire important est que l'utilisateur doit attendre la fin de cette opération avant de demander la destruction des gammes instanciées, c'est à dire de tous les bons globaux et locaux qui ont été créés lors d'une campagne. Ceci afin d'être sûr que l'ensemble des composants introduits puisse être traité et que l'ensemble des bons instanciés puisse être terminé.

A ce point, une précision est nécessaire : il est conseillé de ne définir qu'**une seule opération cible principale**. Toutefois, la tâche XFLOW (cf. annexe E) est capable de prendre en compte plusieurs opérations cibles à partir desquelles seront diffusés les changements de valeur des cibles. Cependant, laisser plusieurs opérations cibles revient à omettre de définir des influences de type TT depuis l'une de ces opérations vers les autres. Il est donc beaucoup plus cohérent de réellement définir ces influences.

Dans notre exemple, l'opération cible choisie est l'action op_1. Définir le PAC au niveau de la première opération permet de détruire les bons de manière progressive, au fur et à mesure que la production se termine. Cependant, en cas de rebut sur l'une des opérations aval, le nombre d'articles terminés ne pourra pas être garanti. La situation est inversée si l'on place le PAC sur l'opération terminale de la gamme.

12.2.2 Définition des zones

La seconde étape consiste à définir les différentes zones dans lesquelles se trouvent les opérations. Les zones sont repérées à partir des flux physiques définis par les arcs du modèle des gammes grâce à la règle suivante :

- Deux opérations appartiennent à la même zone si et seulement si un produit subissant l'une des opérations a subi ou subira l'autre opération une et une seule fois.

Cela permet de définir différentes zones. Sur notre exemple, nous pouvons définir trois zones :

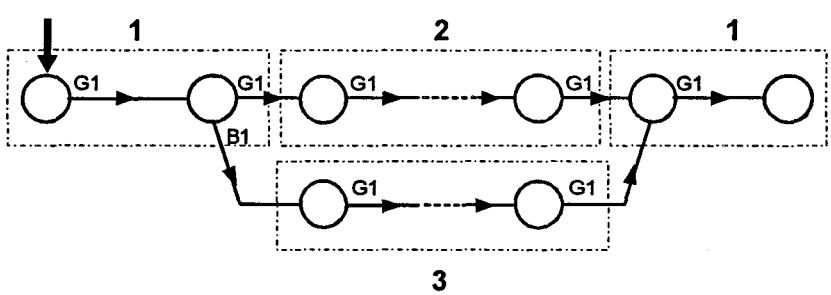


Figure 12.3 : définition des zones (exemple 1).

- **zone 1** : tous les produits effectuent une fois ces opérations,

- **zone 2** : seuls les bons produits issus de op_2 effectuent ces opérations,
- **zone 3** : seuls les mauvais produits issus de op_2 effectuent ces opérations.

A l'intérieur d'une zone, toutes les opérations doivent avoir des valeurs cible égales. Cette remarque permet de fixer dans chaque zone une *opération cible secondaire* qui doit servir de point de diffusion des changements des valeurs cibles des autres opérations de la zone à travers les arcs de type TT. De cette manière, la cohérence des valeurs cibles est assurée. L'opération cible secondaire peut être choisie au hasard parmi toutes les opérations de la zone.

La zone contenant l'opération cible principale sera appelée *zone principale*. Dans cette zone, l'opération cible secondaire doit bien sûr correspondre à l'opération cible principale.

12.2.3 Construction des arcs TT dans les zones

A l'intérieur d'une zone, on définit ensuite la diffusion des changements de valeur cible à l'aide d'arcs TT, selon la topologie suivante :

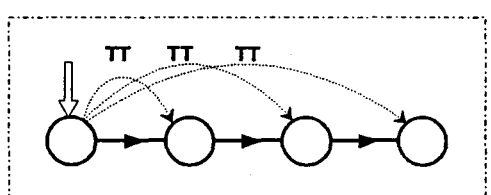


Figure 12.4 : diffusion de la valeur cible dans une zone.

D'autres topologies sont possibles, mais elles seraient strictement équivalentes à celle-ci. En effet, le but est seulement d'exprimer que toutes les valeurs cible dans une zone doivent toujours être égales.

Sur notre exemple, le graphe est maintenant dans l'état représenté figure 12.5.

Il reste désormais, pour chaque zone, à connecter l'opération cible secondaire au reste du graphe afin de compléter le graphe d'influence. Cette phase est la plus délicate. Le but de cette connexion est d'assurer que les actions vérifieront bien l'équation "CIBLE = BONS + MAUVAIS" lorsque le dernier produit sera passé.

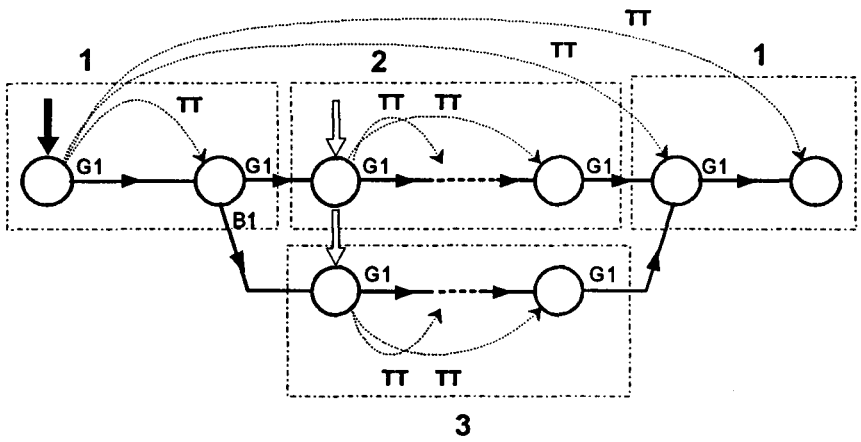


Figure 12.5 : construction des arcs TT dans les zones (exemple 1).

12.2.4 Définition des flux de zone

Cette étape consiste à établir une valeur de flux pour chaque zone. Cette valeur correspond au nombre de produits qui vont transiter dans la zone. Il faut, tout d'abord, fixer des valeurs de flux pour les opérations d'entrée du graphe physique : N1 à Nn, n étant le nombre d'opérations d'entrée dans le graphe physique. On appelle ensuite gi et bi les valeurs des compteurs de bons et mauvais produits en fin de production pour toutes les opérations capables de produire des mauvais produits.

Pour définir le nombre de produits qui vont transiter dans la zone, il suffit alors de comparer la valeur du flux entrant à celle du flux sortant. Une étude plus approfondie des structures apparaissant dans le graphe physique devrait permettre de définir un certain nombre de règles, sinon une méthode, afin de faciliter cette définition.

A l'issue de cette phase, nous obtenons donc des valeurs de flux pour chaque zone et un ensemble d'équations qui traduisent **la conservation du flux physique**. Sur notre exemple, nous avons, avec les valeurs indiquées sur la figure 12.6, l'équation de conservation suivante :

$$N1 = b1 + g1$$

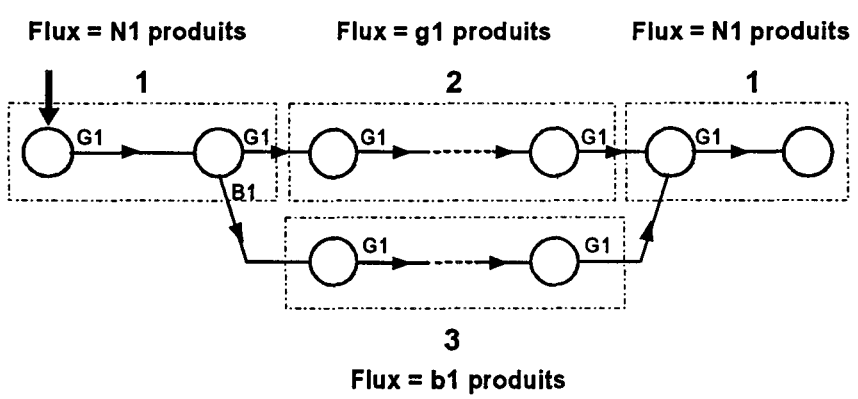


Figure 12.6 : définition des flux (exemple 1).

Les flux et les équations ainsi obtenues vont nous servir à vérifier que les influences inter zones que nous allons maintenant rajouter sont cohérentes.

12.2.5 Ajout des influences inter-zones

Il n'y a pas de méthode pour construire les influences inter-zones : la solution n'est de toute façon pas unique. Seule la pratique permet de comprendre le mécanisme. Sur des exemples simples, ces influences sont facilement appréhendables. Toutefois, il est possible de définir un certain nombre de règles de construction à respecter :

- Tous les arcs d'influences rajoutés **doivent pointer sur une opération cible secondaire,**
- Il ne doit **pas** y avoir **de contrainte d'influence sur l'opération cible principale.** Ceci est directement issu de la définition du PAC.
- Néanmoins, il peut exister des contraintes d'influence depuis une opération d'une zone vers l'opération cible secondaire de cette même zone.

Nous allons, en même temps que nous rajoutons des contraintes d'influence pour connecter les zones, construire un tableau qui nous permettra de vérifier la validité du résultat final.

Dans notre exemple, le tableau est, au départ, le suivant :

Zone	1	2	3
Flux	N1	g1	b1
Flux Max[1]	N1	N1	N1
Cible	N1	0	0

Les valeurs des flux sont les valeurs qui ont été calculées précédemment. La valeur de la cible est initialisée de la façon suivante : pour la zone principale, en fin de production, nous aurons obligatoirement la cible égale au flux. Nous diffusons ensuite cette valeur à travers le graphe d'influence. En son état actuel, les zones 2 et 3 ne sont pas atteintes car elles ne sont pas connectées aux autres zones : le résultat de la diffusion donne donc 0. Le but est de refaire cette diffusion après chaque ajout d'une contrainte de précédence. Le graphe d'influence est complet quand les valeurs de la ligne "cible" sont égales aux valeurs de la ligne "flux".

Nous allons illustrer la démarche pas à pas sur notre exemple.

---

[1] Le flux maximum pouvant passer dans la zone a été rajouté pour une raison qui sera expliquée par la suite. Attention : la valeur de ce flux maximum est dépendante du choix de l'action cible !

Ajout du premier arc : arc TT.

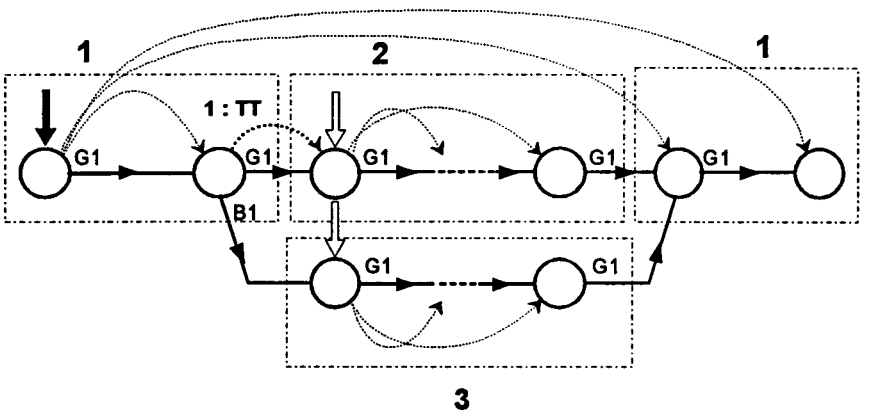


Figure 12.7 : ajout du premier arc inter-zones (exemple 1).

Le tableau devient, après diffusion suivant l'arc ajouté :

Zone	1	2	3
Flux	N1	g1	b1
Flux Max	N1	N1	N1
Cible	N1	0	0
+ 1 : TT	N1	N1	0

Ajout du deuxième arc : arc BT-.

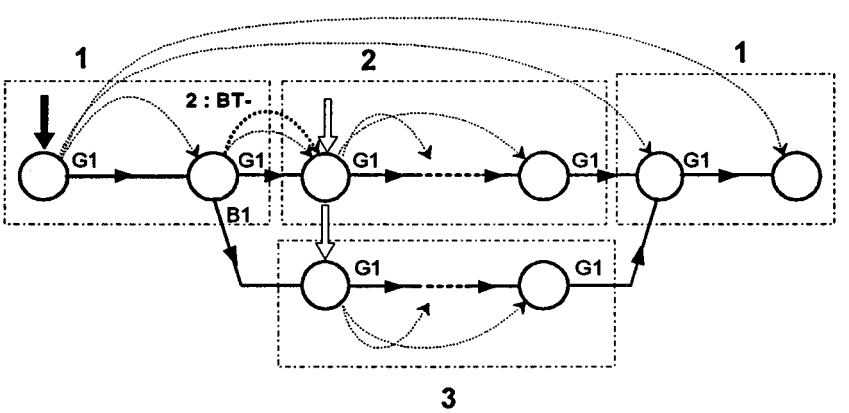


Figure 12.8 : ajout du second arc inter-zones (exemple 1).

Le tableau devient :

Zone	1	2	3
Flux	N1	g1	b1
Flux Max	N1	N1	N1
Cible	N1	0	0
+ 1 : TT	N1	N1	0
+ 2 : BT-	N1	N1 - b1 = g1	0

Le lecteur aura remarqué qu'il eut été plus direct de remplacer les deux arcs rajoutés par un seul arc de type GT+. Cependant, le résultat, tout en étant parfaitement correct, aurait été moins intéressant. En effet, lorsque la production démarre, la première solution initialise les valeurs cibles des opérations de la zone 2 avec la valeur du flux maximum susceptible de traverser la zone (ligne Flux Max). Il ne reste alors au module de pilotage global (dans le superviseur) qu'à décrémenter les valeurs des cibles en fonction de la production. La durée de la période de scrutation importe peu, tout retard dans la mise à jour n'étant pas répercutée sur la production. Dans l'autre solution, dès qu'un bon produit sort de la zone 1, le superviseur doit incrémenter les cibles de la zone 2 pour que les opérations puissent être effectuées sur le produit. Si la période de scrutation est trop longue, ce traitement peut freiner la production. Les contraintes en temps de communication sont donc nettement plus sévères avec cette dernière solution.

C'est pour faciliter la mise en place de la première solution que nous avons introduit dans notre tableau la ligne Flux Max ; cette ligne fournit la valeur d'initialisation par excès correcte des cibles de chaque zone permettant ainsi de déduire les arcs TT inter-zones.

Malheureusement, la solution préconisée n'est pas générale. En effet, il est extrêmement dangereux d'avoir une contrainte qui décrémente l'opération cible dans une zone qui contient une opération d'entrée du graphe physique. Si la décrémentation n'est pas faite instantanément, il serait possible de lancer trop de produits sur la ligne.

Nous nous efforcerons donc de respecter la règle suivante :

- Il ne doit pas y avoir d'arc de contrainte de type GT- et BT- qui pointe sur l'opération cible d'une zone qui contient une opération d'entrée du graphe physique.

Ajout des arcs 3 et 4 : arc TT et BT-.

Pour la zone 3, comme pour la zone 2, on rajoute deux arcs d'influence :

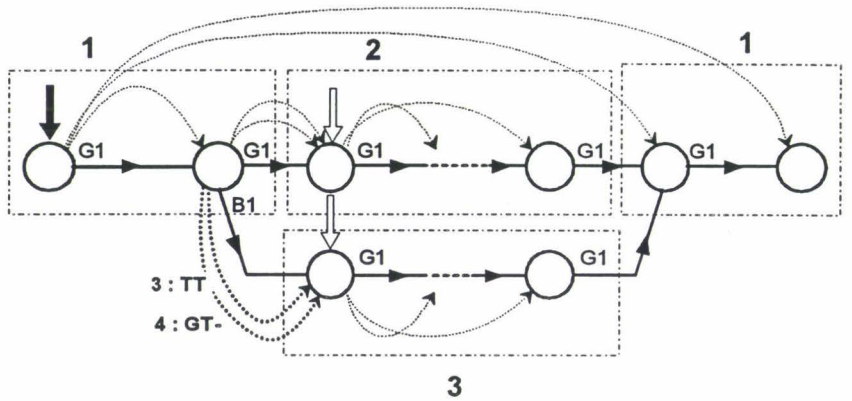


Figure 12.9 : ajout des arcs inter-zones 3 et 4 (exemple 1).

Le tableau devient :

Zone	1	2	3
Flux	**N1**	**g1**	**b1**
Flux Max	N1	N1	N1
Cible	N1	0	0
+ 1 : TT	N1	N1	0
+ 2 : BT-	N1	N1 - b1 = g1	0
+ 3 : TT	N1	g1	N1
+ 4 : GT-	**N1**	**g1**	**N1 - g1 = b1**

La valeur cible (dernière ligne) de chaque zone est maintenant égale à la valeur du flux devant circuler dans la zone (ligne 2). Le graphe d'influence est donc complet.

12.3 Illustration de la méthode : exemple 2

Nous allons maintenant appliquer la méthode sur un second exemple, illustré sur la figure 12.10, afin de compléter les définitions vues précédemment.

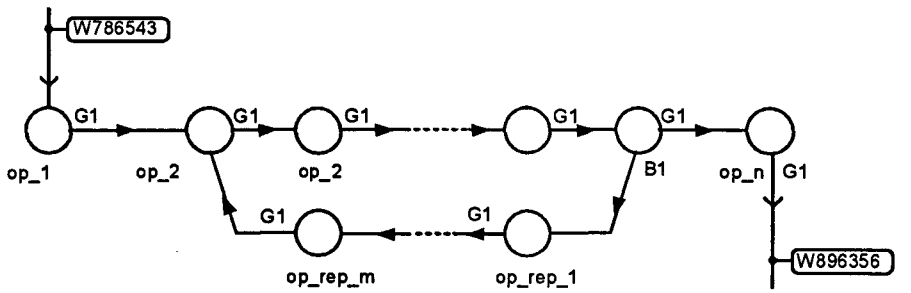


Figure 12.10 : exemple de processus opératoire présentant une gamme de retouche "arrière" (exemple 2).

Dans cet exemple, la gamme principale correspond aux opérations "op_i". Une gamme de retouche "arrière", formée des opérations "op_rep_i", est également spécifiée en cas de défaut sur l'avant-dernière opération : ces opérations permettent alors de replacer le produit au stade précédant "op_2".

La détermination des zones et des opérations cibles est illustrée sur la figure 12.11. Nous avons ici placé le PAC sur la dernière opération. Cela permet de garantir le nombre d'articles à l'arrivée. En contrepartie, la procédure de désinstanciation ne peut pas se déclencher tant que cette opération n'est pas terminée : les bons de fabrication terminés correspondant aux opérations amont restent disponibles dans les automates pour une éventuelle compensation des pertes.

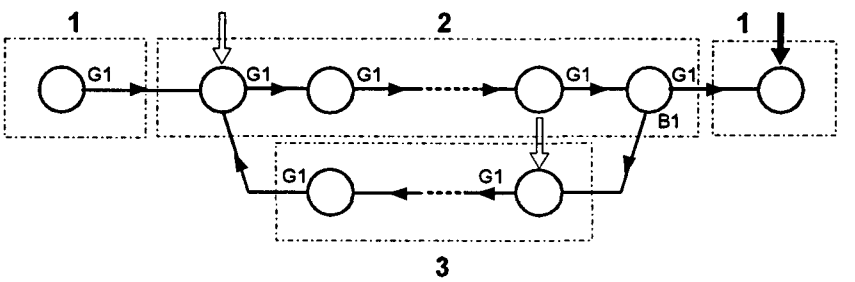


Figure 12.11 : définition des zones (exemple 2).

La construction des arcs de type TT dans les zones nous donne :

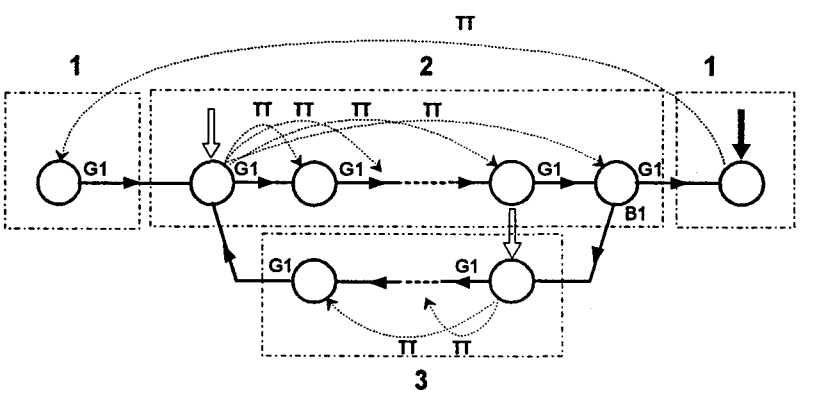


Figure 12.12 : construction des arcs TT dans les zones (exemple 2).

La détermination des flux dans les zones nous donne :

Zone	1	2	3
Flux	N1	N1 + b1	b1
Flux Max	N1	Infini	Infini
Cible	N1	0	0

g1 et b1 sont les valeurs finales des compteurs pour l'action pouvant occasionner des défauts.

La loi de conservation du flux s'écrit :

$$N1 = g1$$

Le calcul du flux maximum pouvant passer dans les zones 2 et 3 nous donne l'infini. Afin de gérer correctement ce cas, deux règles doivent être ajoutées :

- Si le graphe d'influence comporte une boucle d'arcs de type TT, les valeurs cibles des opérations sur cette boucle sont initialisées avec une valeur infinie.
- Lorsqu'un bon de fabrication global est **détruit**, les valeurs cibles des bons de fabrication associés aux opérations amont dans le graphe physique sont forcées à la somme de leurs compteurs de bons et mauvais produits. Ceci permet de détruire les bons dont la valeur cible est infinie.

Cette dernière propriété doit être utilisée avec parcimonie. En effet, il est possible de traiter le premier exemple en ne laissant que les arcs de type TT (en omettant les arcs GT- et BT-), étant donné que les bons auraient été détruits automatiquement en fin de production par application de la seconde règle. Cependant, le bon, dans ce cas, perd un peu de son intérêt. Il ne faut pas oublier que l'une des fonctions des bons de fabrication est de pouvoir suivre de façon fine l'état de la production. Il est donc beaucoup plus cohérent que la valeur cible corresponde à une réalité physique. Dans ce second exemple, la valeur infinie correspond bien à une réalité physique : le flux maximum.

L'utilisation de ces règles dans notre exemple est représentée sur la figure 12.13.

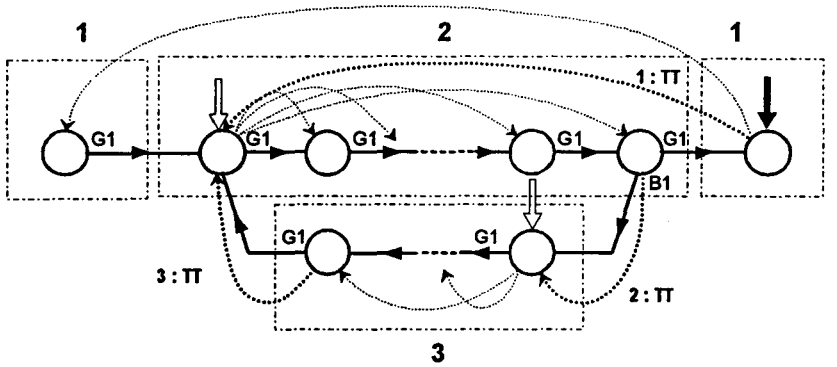


Figure 12.13 : arcs inter-zones (exemple 2).

Attention : l'arc TT numéro 1 est nécessaire ; la boucle d'arcs TT (concrétisée par les arcs 2 et 3) doit en effet être connectée au reste du graphe pour être rencontrée lors de la diffusion de la consigne à partir du PAC.

Le tableau devient :

Zone	1	2	3
Flux	N1	N1 + b1	b1
Flux Max	N1	Infini	Infini
Cible	N1	0	0
1 : TT	N1	N1	0
2 : TT	N1	N1	N1
3 : TT	**N1**	**Infini**	**Infini**

Dans cet exemple, il est inutile d'essayer de rajouter des arcs de type GT- ou BT-, puisque les valeurs cibles sont infinies ! La vérification de la cohérence du graphe d'influence est moins immédiate dans ce cas.

12.4 Résumé de la méthode

En résumé, la méthode proposée pour la construction du graphe d'influence à partir du modèle des gammes est la suivante :

- Choix de l'**opération cible principale**. Cette opération aura, en fin de production, la somme de ses compteurs de bons et mauvais produits égale à la quantité demandée dans l'ordre de production (valeur de la consigne). Dans le graphe d'influence, une telle opération ne peut être influencée par aucune autre opération. Elle doit être unique.

- Détermination des **zones** dans lesquelles se trouvent les opérations. Ces zones sont définies par la règle suivante : Deux opérations appartiennent à la même zone si et seulement si un produit subissant l'une des opérations a subi ou subira l'autre une et une seule fois. La zone dans laquelle se trouve l'opération cible principale est appelée **zone principale**.
- Choix des **opérations cibles secondaires**. Une opération cible secondaire est choisie au hasard dans chaque zone, sauf dans la zone principale ou elle doit correspondre à l'opération cible principale.
- Construction des **arcs TT dans chaque zone**, depuis l'opération cible (secondaire et/ou principale) vers les autres opérations de la zone. Ces arcs assurent qu'à tout moment, toutes les valeurs cibles des opérations de la zone sont les mêmes.
- Calcul des **flux de zones**. Ces flux se calculent en appelant N1 à Nn les flux de produits en entrée du graphe physique, et gi et bi les valeurs des compteurs pour toutes les opérations qui peuvent produire des mauvais produits. Le calcul des flux de zones fournit un certain nombre d'**équations de conservation du flux**.
- Détermination du **flux maximum** pour chaque zone. Ce flux maximum est dépendant du choix de l'opération cible. Il peut servir de valeur d'initialisation des valeurs cibles des zones si ces zones ne contiennent pas d'action d'entrée du graphe physique.
- Rajout des arcs **inter zones**. Les règles suivantes doivent être respectées :
  1. Tous les arcs d'influences rajoutés doivent pointer sur une opération cible secondaire.
  2. Il ne doit pas y avoir de contrainte d'influence sur l'opération cible principale.
  3. Il ne doit pas y avoir d'arc de type GT- et BT- qui pointe sur une zone contenant une opération d'entrée du graphe physique.
  4. Pour les zones ne contenant pas d'opération d'entrée du graphe physique, il est intéressant d'initialiser les valeurs cibles avec la valeur du flux maximum, puis de décrémenter les valeurs en fonction de la production.
- Construction **du tableau de vérification**, en parallèle du rajout des arcs inter-zones. Ce tableau a la forme suivante :

Zone	1	...	...
Flux			
Flux Max			
Cible			
+ 1er arc			

Tous les arcs ajoutés apparaissent en ligne. Pour une construction plus aisée du tableau, les zones sont connectées au graphe d'influence les unes après les autres. Ceci permet de ne modifier les valeurs que dans une colonne à la fois.

Le graphe est complet quand la valeur des cibles dans toutes les zones (dernière ligne) est égale à la valeur du flux (ligne 2).

- Si besoin, gestion **des flux maximum infinis**. Pour les zones qui présentent un flux maximum infini, et qui ne contiennent **pas d'opération d'entrée du graphe physique**, il est possible de leur associer une valeur de cible infinie. Les règles qui permettent de gérer ce mécanisme sont :

    1. Lorsque le graphe contient **une boucle d'arcs de type TT**, les valeurs cibles des opérations dans la boucle sont initialisées avec une valeur infinie. Cette boucle doit être connectée au reste du graphe.
    2. Lorsqu'un bon de fabrication global est détruit, les valeurs cibles des bons associés aux opérations amonts du graphe physique sont forcées à la somme des compteurs de bons et mauvais produits.

La pratique étant le meilleur moyen pour comprendre les différentes utilisations possibles des graphes d'influence, nous donnons en annexe C quelques exemples.

Les exemples des paragraphes C.1 et C.2 correspondent aux deux exemples déjà traités. Ils illustrent les manières possibles de contrôler les deux types de boucles de retouche rencontrés : "avant" ou "arrière".

L'exemple d'une gamme linéaire, développé dans le paragraphe C.3, permet de comprendre les deux types de contre-réaction aux pertes par rapport à la position dans la gamme de l'opération cible principale. Nous avons placé à dessein le PAC au milieu de la gamme, définissant ainsi deux zones : la zone amont au PAC où toute perte doit être compensée par réintroduction de composants aux entrées (contre réaction positive), et la zone aval au PAC où les pertes ne sont pas compensées (contre-réaction négative).

Une autre utilisation intéressante des graphes d'influence est donnée par l'exemple du paragraphe C.4 correspondant à un cas de synthèse de gammes relatives à des articles différents (cf. § 8.4). La synchronisation du rendez-vous au niveau de l'opération d'assemblage est donc supposée hors du domaine de l'ordonnancement. Nous présentons deux solutions pour gérer cet exemple : pas de synchronisation si la ligne de fabrication du composant secondaire est balancée avec la ligne principale, ou synchronisation par l'intermédiaire d'un arc GT+. L'ajout de cet arc permet de conditionner le lancement en production du composant à la réussite d'une opération sur le récepteur.

Nous donnons également, au paragraphe C.5, l'application des graphes d'influence au cas du produit P assemblé sur le site DRD.

Chapitre 13

Coordination des transferts

Introduction

Lorsqu'un moyen transformationnel a achevé les opérations qu'il devait effectuer sur un produit, ce dernier doit être acheminé vers un autre moyen transformationnel. Une phase de transfert est donc systématiquement spécifiée dans la gamme entre deux phases transformationnelles.

Le simple envoi d'une requête de type bon de fabrication local, tel que défini au chapitre 11, à destination d'un système de transfert ne suffit cependant pas pour paramétrer complètement le transfert. En effet, les systèmes de transfert étant des systèmes complexes pouvant présenter un certain nombre d'indéterminismes locaux [Huvenoit, 1994], [Ausfelder, 1994], leur commande nécessite l'intervention du niveau décisionnel.

Ce niveau est parfois directement proposé avec la transitique par le fournisseur (souvent de façon centralisé). Dans le cas contraire, nous présentons dans ce chapitre, une solution respectant les principes généraux présentés au chapitre 10.

13.1 Principe

13.1.1 Spécificité des phases de transfert

Nous avons vu, au chapitre 5, qu'un système de transfert pouvait être décomposé très facilement en systèmes opérants dont les fonctions sont clairement définies : orienter pour les systèmes du type aiguillage, transporter

pour les systèmes de type convoyeur, bloquer pour les systèmes du type indexeur.

Une solution envisageable serait donc de gérer les phases de transfert de façon identique à celle des phases transformationnelles, i. e. par décomposition en terme d'opérations puis envoi de requêtes à destination de chaque système opérant. Malheureusement, l'utilisation directe de la solution pratique proposée au chapitre 11 est délicate. En effet, la représentation des opérations intervenant dans un transfert, telle que nous l'avons effectuée pour les phases transformationnelles, est illusoire en cas de flexibilité importante.

Les travaux menés au LAIL dans le cadre du projet CASPAIM sont, à cet égard, significatifs : d'une représentation de chaque possibilité de transfert par un processus dans la première version du projet, les travaux aboutissent désormais à un modèle **global** des systèmes de transfert (notion de *filtre comportemental* [Huvenoit, 1994]). Le but d'un tel modèle est de permettre la résolution des indéterminismes résultant de la flexibilité interne au système de transfert en ayant une connaissance globale de l'état du système.

C'est également vers cette approche que nous nous sommes orientés. Le modèle de l'installation que nous utiliserons a été établi au chapitre 8. Cependant, la nécessaire centralisation d'un modèle global, rapportée aux contraintes exposées au chapitre 10, ne permet pas la mise en oeuvre d'une solution utilisant en temps réel le modèle. Nous avons donc adopté le même principe que celui mis en oeuvre pour la coordination de la production : un paramétrage, calculé au niveau global à partir du modèle, des divers aiguillages permettant un fonctionnement local autonome pendant une certaine durée.

Deux types de paramètres sont mis en oeuvre lors d'un transfert : d'une part, des informations associées aux objets mobiles, et, d'autre part, des règles associées à chaque aiguillage permettant d'utiliser ces informations. Ces deux types de paramètres se rapportent respectivement aux notions de routage et de trafic détaillées dans le paragraphe suivant.

Remarque :

Puisqu'une phase de transfert doit toujours être effectuée à l'issue d'une phase transformationnelle, il est possible d'éliminer ces phases des gammes. Une gamme peut en effet être caractérisée uniquement par la donnée de ses phases transformationnelles. C'est l'approche adoptée dans [Bouchet et Lellahi, 1987].

Pour notre part, nous avons également suivi cette approche dans le cadre de la réalisation logicielle actuelle présentée en annexe F. Le module informatique XFLOW réalise l'émission et la gestion des diverses requêtes permettant la coordination des moyens concernés par la production, conformément aux mécanismes présentés au chapitre 11. Les opérations de phases gérées par ce module doivent obligatoirement être affectées à un moyen contrôlé de façon centralisé. Le contrôle des divers systèmes opérants d'un système de transfert étant bien souvent réparti, l'émission d'une requête associée à une phase de transfert n'a guère de sens avec le module actuel. Nous verrons néanmoins que le maintien des phases de transfert au niveau des gammes s'avère

utile dans certains cas (cf. § 13.2.4) ; une extension future de XFLOW devrait permettre de traiter ce cas.

13.1.2 Notions de routage et de trafic

Les notions complémentaires de routage et de trafic ne doivent pas être confondues.

Le terme de *routage* sera utilisé ici pour désigner le traitement consistant à affecter à un objet mobile sa prochaine destination dans l'installation.

Le terme de *trafic*, quant à lui, désignera la gestion des aiguillages au sein d'un système de transfert.

L'exemple suivant, tiré de [SCOPES D4, 1993], permettra d'apprécier la différence entre ces deux notions. Considérons le carrefour routier illustré sur la figure 13.1. Le choix de la destination (routage) de chaque véhicule est établi par chaque automobiliste. Ce choix est effectué indépendamment du réseau routier (supposé néanmoins relier les points d'origine et d'arrivée), et dépend de l'**intention** de chaque automobiliste. Parvenu au carrefour, chaque automobiliste devra respecter la signalisation réglant l'entrée sur la zone centrale. Cette signalisation, les feux de circulation dans l'exemple, assure la **sécurité** du trafic en éliminant les risques de collision.

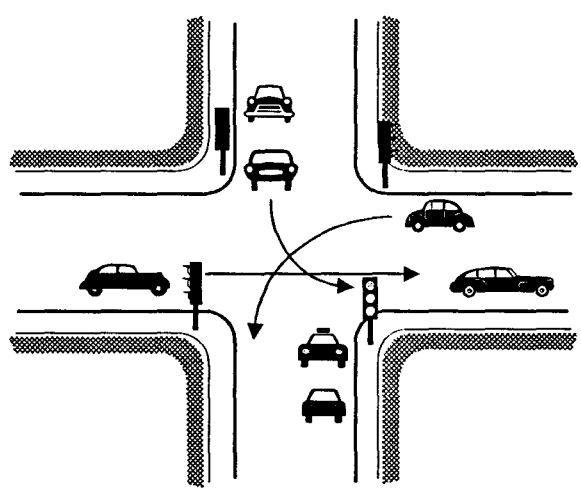


Figure 13.1 : trafic et routage au niveau d'un carrefour routier.

Le routage participe donc d'une intelligence globale. Il nécessite la connaissance d'informations sur l'ensemble du système. A l'inverse, le trafic peut être résolu à partir d'informations disponibles localement. Il participe d'une intelligence locale.

Les deux mécanismes que nous allons maintenant présenter permettent de mettre en oeuvre ces deux notions. Ce sont :

- les *règles d'aiguillage* pour le contrôle du trafic ; ces règles seront mises en oeuvre au niveau local (automate) ;

- et les *règles de destination* pour le calcul du routage ; ces règles seront mises en oeuvre au niveau global (superviseur).

Notons dés à présent que, conformément aux principes généraux présentés au chapitre 10, seul le mécanisme des règles d'aiguillage est fondamental dans notre approche ; celui des règles de destination, centralisé, n'est destiné qu'à améliorer la prise de décision par l'exploitation d'informations globales.

13.2 Les règles d'aiguillage

13.2.1 Présentation

Reprenons l'exemple du carrefour. Le fonctionnement courant d'un tel carrefour, ramené à un automobiliste, peut être décomposé en deux phases :

- Phase 1 : attente d'autorisation de pénétrer sur le carrefour. Cette autorisation est matérialisée par les feux de circulation. Elle ne prend pas en compte la destination de chaque mobile[1].
- Phase 2 : pénétration sur le carrefour et choix de la voie de sortie. Le choix tient compte ici de la destination de chaque véhicule.

Le fonctionnement d'un aiguillage d'un système de transfert sera calqué sur celui du carrefour.

Deux types de règles permettent de gérer les deux phases : les règles en entrée et les règles en sortie. Deux tableaux, correspondant à ces deux types de règles, seront associés à chaque aiguillage.

Regroupement en tables

La figure 13.2 illustre la structure de ces tableaux. L'exemple choisi correspond à la table des règles de sortie de l'aiguillage d'entrée de la station IBM du site DRD.

A chaque ligne du tableau correspond une règle (cf. partie ombrée du tableau). Le nombre de lignes dépend de la capacité en règles accordée à chaque aiguillage, fonction elle-même de la capacité mémoire de l'automate contrôlant l'aiguillage.

Les règles étant examinées successivement, l'ordre des règles revêt une grande importance : il est nécessaire de les classer de la plus précise à la plus générale. Pour un objet à orienter, plusieurs règles peuvent en effet s'appliquer. Seule la première, supposée être la plus précise, le sera effectivement.

---

[1] Il est nécessaire de relativiser cette affirmation. En effet, dans un souci d'amélioration du débit, un dédoublement des voies d'entrée est souvent effectué afin de permettre une présélection des véhicules suivant la voie de sortie. Cela équivaut à prendre en compte la destination dans la règle d'entrée du carrefour.

Signal	Campagne	Destination	Opération	Type	Paramètres
on	*	IBM	*	Priorité	1- canal vers IBM
on	*	*	E_op1	Cyclique	1- deux palettes vers canal IBM 2- trois palettes vers canal convoyeur
off	15	*	S_rep1	Priorité	1- canal vers station IBM
(...)	(...)	(...)	(...)	(...)	(...)
on	*	*	*	Priorité	1- canal vers station IBM

Figure 13.2 : table de règles d'aiguillage.

Au niveau d'une règle, deux parties principales peuvent être distinguées : la *partie discriminante*, constituée des quatre premières colonnes, fixe la condition d'application de la règle, tandis que la *partie décisionnelle*, formée des deux dernières colonnes, permet le choix du canal à utiliser (pour l'entrée ou la sortie de l'aiguillage).

Partie discriminante

Quatre données caractérisent la partie discriminante d'une règle d'aiguillage.

La première donnée est un **signal booléen ON/OFF** spécifiant si la règle est activée ou non. Parmi les règles de la table, seules celles qui sont activées sont examinées. Ce signal est analogue à celui déjà rencontré au niveau des bons locaux. L'utilisation de ce signal sera examinée en détail au paragraphe 13.4.

Les données suivantes permettent de choisir la règle en fonction des informations associées aux objets mobiles.

Trois types d'informations peuvent être utilisés comme discriminant : la "campagne d'appartenance", la "prochaine opération" et la "prochaine destination" de l'objet à aiguiller. Nécessaires au pilotage, ces informations font partie des données accompagnant les objets au cours de leur déplacement dans l'atelier (par exemple transmission d'automates à automates par réseau local ou étiquettes magnétiques associées aux palettes).

L'item "campagne" est initialisée si l'objet mobile est associé à un article en cours de fabrication (palette supportant un produit par exemple). Cette initialisation a lieu lors du lancement des articles en production (opérations de tête de gamme). L'utilisation de cette donnée comme discriminant permet de différencier les règles suivant les campagnes en cours.

L'item "prochaine opération" est initialisé systématiquement à l'issue de chaque opération au niveau d'un moyen. Ce dernier est capable de déterminer cette opération grâce aux données du dernier bon local utilisé (cf. § 11.2.3).

Quant à l'item "prochaine destination", il est établi par le mécanisme des *règles de destination* développé au paragraphe 13.3. Cette donnée, à la différence de la précédente, peut ne pas être initialisée : l'ensemble des moyens susceptibles de réaliser l'opération suivante constitue alors la prochaine

destination. De ce fait, une règle discriminée par destination doit être considérée comme étant plus précise qu'une règle discriminée par opération. Elles doivent donc être examinées préférentiellement.

La comparaison de la valeur de l'item discriminant de l'objet à celle fournie dans le champ correspondant de la règle détermine si la règle est ou non applicable. Lorsque plusieurs champs discriminants de la règle sont valués, il faut vérifier que chacun des champs coïncide. Un champ non valué (symbolisé par le caractère *) n'est pas discriminant : la condition correspondante est vérifiée quelque soit la valeur correspondante associée au mobile.

Lorsqu'aucun des champs discriminants de la règle n'est valué, la règle correspondante s'applique à tous les objets se présentant à l'aiguillage. Il est ainsi possible, par ce biais, de définir une règle par défaut, toujours applicable. Une telle règle est, bien sûr, de portée très générale et doit, par conséquent, être placée en fin de table. Par ailleurs, il ne saurait en exister plusieurs dans une même table : seule la première serait effective.

Partie décisionnelle

La partie décisionnelle d'une règle d'aiguillage spécifie la règle décisionnelle à appliquer pour choisir le canal. Une telle règle de décision sera définie par son **type** et un ensemble de **paramètres**. Nous présentons ci-après les deux types de règles que nous avons mis en oeuvre dans nos travaux. Ces règles ont été proposées initialement dans [Verdebout, 1992]. Nous en reprenons les définitions et les notations.

Elles ne sont pas exhaustives. Il est possible d'en définir d'autres suivant le même principe (type + paramètres). La seule contrainte est qu'elles ne fassent appel qu'à des informations disponibles localement. Cependant, nous montrerons au chapitre 12 que les deux types présentés sont suffisants pour la plupart des applications.

La convention suivante sera utilisée : soit AIG un aiguillage ; nous désignerons par AIG:1 à AIG:n ses canaux de sortie et par 1:AIG à m:AIG ses canaux d'entrée.

• **règle par priorité décroissante :**

La *règle par priorité décroissante* consiste à choisir le premier canal libre (resp. occupé) dans une séquence ordonnée de canaux de sortie (resp. d'entrée). Les éléments de la séquence constituent les paramètres de la règle. Seuls les canaux spécifiés seront utilisés, les autres étant supposés interdits.

Elle sera notée **RP(AIG:i_1,...,AIG:i_k).**

AIG:i_1,...,AIG:i_k représente la séquence ordonnée de canaux autorisés (ici de sortie).

Cette règle, appliquée en sortie, échoue si aucun des canaux n'est libre : la pièce située à l'aiguillage est bloquée jusqu'à libération d'un des canaux. Appliquée en entrée, elle n'échoue jamais.

• **règle par permutation cyclique pondérée :**

La *règle par permutation cyclique pondérée* consiste à desservir chaque canal spécifié en paramètre un nombre de fois donné. En cas de saturation (resp. famine) du canal de sortie (resp. d'entrée) courant, le canal suivant devient le canal courant.

D'autres variantes de cette règle existent. Pour l'application en sortie, une variante consiste à bloquer les objets en cas de saturation du canal courant si ce dernier n'a pas reçu le nombre d'objets spécifié. Une autre variante, toujours en cas de saturation du canal de sortie courant, consiste à desservir le premier canal libre, puis à revenir au canal courant[2].

Cette règle sera notée **RC((AIG:i_1,j_1),...,(AIG:i_k,j_k))**.

AIG:i_1,...,AIG:i_k représente la séquence ordonnée de canaux autorisés (ici de sortie) et j_1,...,j_k les poids respectifs de chaque canal.

Appliquée en sortie, cette règle échoue si aucun des canaux n'est libre. En entrée, elle n'échoue jamais.

13.2.2 Algorithme local de gestion du trafic

Nous donnons ci-après l'algorithme mis en oeuvre au niveau local pour gérer le trafic. Les tables de règles associées à chaque aiguillage sont supposées connues et valides. Les procédures RP et RC sont données pour le choix du canal de sortie. L'adaptation au choix du canal d'entrée ne pose pas de problème (il suffit de remplacer "libre" par "occupé" dans les algorithmes).

```
Procédure choisir un canal
  Tant qu'il existe une règle dans la table
    Si la condition discriminante est vérifiée
      Si le type de la règle est priorité
        Choisir canal selon RP(AIG:1,...,AIG:n)
      Si le type de la règle est cyclique
        Choisir canal selon RC((AIG:1,k1),...,(AIG:n,kn))
      Finsisi
    Finsi
  Fintantque

Procédure choisir canal selon RP(AIG:1,...,AIG:n)
  Pour j=1 à n
    Si le canal AIG:j est libre
      Choisir ce canal
      Interrompre la procédure (j=n+1)
    Finsi
  Finpour
```

[2] Dans cette variante RC((AIG:i_1,∞),(AIG:i_2,0),...,(AIG:i_k,0)) équivaut à RP(AIG:i_1, AIG:i_2,..., AIG:i_k) [Verdebout, 1992].

```
Procédure choisir canal selon RC((AIG:1,k1),...,(AIG:n,kn))
   Pour j=1 à n
      Si le canal AIG:m est libre
         Choisir ce canal
         K = K+1
         Si K = k_m
            K = 0
            m = (m modulo n) + 1
         Finsi
         Interrompre la procédure (j=n+1)
      Sinon
         K = 0
         m = (m modulo n) + 1
      Finsi
   Finpour

Notation :  m indice du canal courant
            K nombre d'objets déjà affecté au canal AIG:m
```

13.2.3 Emploi des règles d'aiguillage

Les règles d'aiguillage sont nécessaires lorsqu'un **choix** de canal doit être effectué entre plusieurs canaux possibles. Ainsi, la table de règles relative au choix du canal d'entrée est-elle inutile dans le cas d'un aiguillage à une seule entrée (cas des aiguillages divergents du site DRD). De même, la table de règles en sortie n'est pas utile dans le cas d'un aiguillage à une seule sortie (cas des aiguillages convergents du site DRD).

En outre, la stratégie mise en oeuvre au niveau du choix du canal d'entrée, lorsqu'un conflit se produit, se limite très souvent, dans la pratique, à fixer une priorité sur l'un des canaux, quelque soit le type et le stade du produit se présentant. Dans le cas d'un aiguillage contrôlant les sorties d'une dérivation, une politique très souvent rencontrée consiste à fixer la priorité sur le canal issu de la dérivation (priorité à la sortie). La table des règles en entrée se limite alors à une seule ligne fixant une règle du type priorité, appliquée par défaut. Rien cependant n'empêche d'essayer des politiques plus complexes.

La véritable application des règles d'aiguillage reste, bien entendu, le choix d'un canal de sortie dans le cas d'aiguillages divergents. C'est à ce niveau que les stratégies mises en oeuvre auront le plus de conséquences sur le fonctionnement de l'installation.

13.2.4 Paramétrage statique ou dynamique

La mise en oeuvre des règles d'aiguillage peut s'effectuer soit de manière statique, soit de manière dynamique.

Le *paramétrage statique* consiste à paramétrer chaque aiguillage une fois pour toute en utilisant des règles de portée générale (discriminées par opération ou par destination et non par campagne). Les règles ainsi sélectionnées sont supposées pouvoir aiguiller n'importe quel objet mobile se présentant à l'aiguillage, quel que soit la campagne en cours. Elles ont une validité

permanente et peuvent être chargées à demeure dans l'organe de commande de l'aiguillage (ou du moins tant que la configuration de marche du système reste identique).

A l'inverse, le *paramétrage dynamique* est nécessaire lorsque les stratégies de trafic diffèrent suivant les campagnes. Cela est nécessaire, par exemple, lorsque certains moyens demandent un temps de préparation important : préparés pour une campagne donnée, il ne faut pas y orienter les produits d'une autre campagne simultanément en cours. La solution passe alors par l'utilisation de règles discriminées par campagne. La validité d'une telle règle n'est plus permanente ; elle se limite à la durée de vie de la campagne. Une procédure doit bien sûr éliminer les règles non valides. Une telle procédure est facilitée si les phases de transfert sont maintenues dans la gamme et sont gérées de manières identiques aux opérations transformationnelles : il est alors possible de détecter la "terminaison" de l'opération de transfert et ainsi de détruire les règles d'aiguillage associées au bon moment.

Remarque :

Ne gérant pas les phases de transfert, le prototype réalisé dans le cadre de ce travail (cf. annexe F) ne permet pas de mettre en oeuvre le paramétrage dynamique des règles. Cette limitation ne pose de problème dans le cadre des applications très flexibles envisagées où les temps de préparation sont le plus souvent nuls.

13.2.5 Initialisation des règles d'aiguillage

Les tables de règles d'aiguillage sont initialisées au niveau global, puis descendues vers les modules locaux (manuellement ou automatiquement). L'établissement des règles à implanter à chaque aiguillage est un problème en soi : il faut en effet s'assurer de la *complétude* et de la *cohérence* de l'ensemble des règles de l'installation.

La complétude exprime le fait que tout objet se présentant à un aiguillage doit être traité ; ainsi au moins une règle doit être applicable, en entrée comme en sortie, pour cet objet.

La cohérence exprime le fait que la trajectoire d'un objet dans l'installation est en accord avec les destinations possibles de cet objet ; un ensemble de règles cohérent permet, pour tout objet en un point donné de l'installation, d'aiguiller cet objet vers une de ses destinations admissibles.

Le calcul de l'ensemble de règles ayant les meilleures performances (par rapport à un critère donné fixé par l'utilisateur) et respectant les propriétés de complétude et de cohérence s'avère délicat sans l'aide d'une méthode. Deux outils, ont été développés dans cette optique.

Le premier outil, proposé dans [Verdebout, 1992], permet d'obtenir, la *gamme de routage* d'un objet à un stade donné, à partir de sa *gamme de provenance*. La gamme de provenance d'un objet est le sous-graphe des lieux d'un système de transfert depuis lesquels un objet peut atteindre un ensemble

donné de destinations. Un tel graphe permet de déterminer l'ensemble des aiguillages devant être paramétrés pour assurer un transfert correct de l'objet au stade donné. Un algorithme simple permet de calculer ce graphe. Associé à une interface conviviale, il offre une aide non négligeable à l'utilisateur pour la détermination des aiguillages à configurer. Une fois complété par les règles choisies par l'utilisateur, le graphe ainsi obtenu porte le nom de gamme de routage. Une telle gamme, associée à une phase de transfert, permet de configurer chaque aiguillage susceptible d'intervenir dans le transfert. L'algorithme doit être appliqué pour chaque phase de transfert intervenant dans la gamme d'un article.

Cet outil ne fournit cependant aucune aide quant au choix des règles les plus favorables à une installation donnée. Le second outil, proposé dans le cadre de ce travail, offre cette possibilité. Bâti sur une méthode d'analyse combinatoire, il permet de calculer le "meilleur chemin" d'un objet suivant un critère donné et d'en déduire les règles à implanter aux aiguillages concernés. Cette méthode fera l'objet du chapitre 14.

Deux solutions sont possibles pour la mise en oeuvre de ces méthodes : "hors ligne" ou "en ligne". Dans le premier cas, le calcul des règles d'aiguillage s'effectue lors de la phase de conception ou de mise au point du système. Les paramètres ainsi précalculés sont téléchargés tels quels lors de la phase exploitation, sans prise en compte de l'état courant du système. Dans le second cas, le calcul a lieu directement au cours de la phase exploitation, juste avant le transfert des règles vers les niveaux locaux ; il est ainsi possible de tenir compte de l'état courant du système, notamment des disponibilités des moyens, dans le but d'améliorer le comportement du système.

Dans les applications que nous avons traitées jusqu'à présent, nous nous sommes limités à un paramétrage statique des règles d'aiguillage, ne nécessitant donc pas de calcul "en ligne". Cette approche, simple de mise en oeuvre, "rassure" en effet les industriels qui aiment appréhender le fonctionnement de leur système.

Néanmoins, nous pensons que, dans un futur proche, lorsque les industriels auront appris à maîtriser des systèmes de plus en plus complexes à travers les outils de la supervision (synoptiques, outils statistiques SPC et SQC, etc.), le paramétrage "dynamique" des règles calculées "en ligne" constituera une solution adaptée et réactive à la gestion du trafic.

13.2.7 Application à l'exemple du site DRD

Nous allons illustrer l'utilisation des règles d'aiguillage sur l'exemple du convoyeur à palettes du site DRD.

Huit aiguillages doivent être paramétrés (cf. figure 8.7). Les quatre aiguillages d'entrée de dérivation (AIG_AM_i) possèdent un seul canal d'entrée et deux canaux de sortie. Les quatre aiguillages de sortie (AIG_AV_i) possèdent deux canaux d'entrée et un seul canal de sortie.

Nous désignerons par **AIG:conv** et **conv:AIG** les canaux liant l'aiguillage au convoyeur central et par **AIG:station_i** et **station_i:AIG** ceux liant l'aiguillage à la station (i = 1 pour SR, 2 pour AFMA, 3 pour IBM, 4 pour MANU).

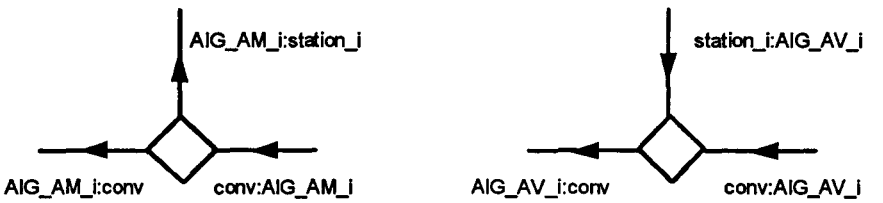


Figure 13.3 : détail d'un aiguillage a) divergent et b) convergent.

Le contrôle du canal d'entrée unique des aiguillages amont ne posent pas de problème (cf. § 13.2.3). Il en est de même pour le canal de sortie unique des aiguillages aval.

Le choix du canal d'entrée des aiguillages aval sera réalisé en donnant la priorité au canal provenant de la station. La table correspondante se verra affectée la règle suivante, appliquée par défaut :

RP(station_i:AIG_AV_i).

Le contrôle des canaux de sortie des aiguillages amont est plus délicat. C'est à leur niveau qu'est réalisée l'orientation des palettes vers les différentes stations de l'installation. De nombreuses stratégies peuvent être mises en oeuvre. Nous en illustrerons deux.

Une première stratégie est d'implanter une règle du type priorité pour chaque opération réalisable par la station et une règle par défaut pour les autres valeurs. On obtient, pour la station AFMA, le tableau ci-dessous.

De cette façon, les stations s'alimentent en produits jusqu'à saturation de leur buffer d'entrée. Une telle politique est appelée "Greedy Boy" (gamin glouton) par les anglo-saxons, car les stations ne "pensent" pas aux suivantes. En cas de stations identiques en parallèle, les stations aval se voient desservies en dernier. La charge n'est pas équilibrée.

on	*	*	P_op1	RP	AIG_AM_2:AFMA AIG_AM_2:conv
on	*	*	E_op1	RP	AIG_AM_2:AFMA AIG_AM_2:conv
on	*	*	S_op1	RP	AIG_AM_2:AFMA AIG_AM_2:conv
on	*	*	*	RP	AIG_AM_2:conv

La première règle spécifie qu'une palette transportant un produit devant subir l'opération P_op1 doit être dirigée préférentiellement sur le canal AIG_AM_2:AFMA desservant la station AFMA. En cas de saturation de ce canal, le canal AIG_AM_2:conv est choisi : la palette reste sur le convoyeur central.

Une deuxième stratégie, permettant d'équilibrer la charge, est d'introduire des règles cycliques pour chaque opération réalisable sur plusieurs stations. Ainsi l'opération "S_op1" est-elle réalisable sur les stations IBM et AFMA. En considérant les stations comme étant identiques, la solution consiste à laisser passer une palette sur deux au niveau de l'aiguillage d'entrée de la station AFMA par la règle par pondération cyclique suivante :

RC((AIG_AM_2:AFMA, 1), (AIG_AM_2:conv ,1)) ;

l'aiguillage en entrée d'IBM conserve quant à lui la règle par priorité suivante :

RP(AIG_AM_3:IBM, AIG_AM_3:conv),

orientant les palettes concernées vers IBM si le buffer d'entrée n'est pas plein, les laissant passer sinon. Si un tour de convoyeur supplémentaire n'est pas souhaitable en cas de saturation d'IBM, il est possible de bloquer la palette en attente sur l'aiguillage par la règle :

RP(AIG_AM_3:IBM).

Bien entendu, les palettes sur le convoyeur central sont également bloquées en parvenant à cet aiguillage.

Dans le cas où N stations identiques sont montées en parallèle, N-1 règles cycliques sont nécessaires sur les N-1 premières stations (de poids décroissant de N-1 à 1), la dernière recevant une règle par priorité. Si les stations ne sont pas considérées comme identiques (performance et/ou fiabilité différente), une pondération différente doit être mise en place. La méthode de calcul présentée au chapitre 14 permet d'établir une telle pondération.

13.3 Les règles de destination

13.3.1 Un problème de niveau global

Les règles de destination permettent d'établir pour un mobile parvenant à un point donné dans l'atelier, sa prochaine destination [Datin, 1994]. Rappelons que le paramètre destination associé à un objet est utilisé au niveau des aiguillages des systèmes de transfert (emploi des règles de trafic – cf. § 13.2).

Les points de décision sont laissés au libre choix de l'utilisateur. Ils doivent néanmoins correspondre aux éléments du système de transfert de type plot (aiguillage ou indexeur). Seuls ces éléments permettent en effet de connaître la position exacte des mobiles dans l'installation et donc de déclencher un processus de commande.

Comme dans tout problème de prise de décision, une stratégie mettant en oeuvre certains critères doit être appliquée (cf. § 14.1.1 sur la prise de décision). La performance de la stratégie dépend de la qualité des critères dont on dispose (précision, caractère global). En outre, plus ces derniers seront nombreux, plus les stratégies possibles seront variées et adaptées aux objectifs désirés. Pour ces raisons la mise en oeuvre des règles de destination sera centralisée. Notons immédiatement que, du fait de cette centralisation,

l'utilisation des règles par destination ne saurait être obligatoire pour le fonctionnement du système (cf. § 10.2.3). Elle en améliore les performances en mode normal sans pour autant porter à conséquence en cas de panne du superviseur.

Les critères disponibles au niveau du superviseur sont nombreux. Ce dernier dispose de toutes les informations sur le déroulement de la fabrication ; certaines données sont transmises en temps réel (mode de marche des stations par exemple), d'autres proviennent d'agrégations et sont mises à jour à intervalles réguliers par les fonctions surveillance et analyse du superviseur (nombre d'incidents des machines par heure, état d'avancement de la production, etc.). Il dispose également de l'ensemble des données fournies par les modules décisionnels amont (planification et ordonnancement) sous forme de marges temporelles (date de début et de fin au plus tôt ou au plus tard, etc.).

13.3.2 Proposition de mise en oeuvre

La technique proposée pour mettre en oeuvre les stratégies choisies par l'utilisateur est un ensemble de règles déclaratives mises en oeuvre au sein d'un système expert. Cette approche s'adapte en effet parfaitement à notre cas : le savoir-faire humain, ici la connaissance des exploitants d'un système de production, est, d'une part, insuffisamment structurée et, d'autre part, trop sujette à révisions ou compléments (suivant l'expérience accumulée) pour pouvoir faire l'objet d'une mise en oeuvre algorithmique.

Une approche prometteuse pour notre problème est constituée par les systèmes experts "flous", i.e. faisant appel à la théorie de la logique floue pour représenter connaissances et règles d'inférences. Nous avancerons deux raisons.

Premièrement, le savoir faire des ingénieurs de production relève plus d'une approche linguistique que mathématique. En d'autres termes, ils utilisent des expressions vagues et imprécises (à ne pas confondre avec incertaines !) du style "taux de production élevé" ou "fréquence d'occurrence assez faible". Ce type de connaissance ne peut que très difficilement être exploité dans les systèmes experts classiques. En effet, les techniques de représentation des connaissances usuelles imposent l'utilisation de domaines de valeurs strictement bornés : par exemple, il serait possible de définir la valeur "moyenne" du taux de production d'une machine par l'intervalle [400, 800] produits/heure ; cependant, à 399 produits/heure, ce taux serait automatiquement "faible" et, à 801 produits/heure, obligatoirement "grand". La logique floue permet, quant à elle, de conserver l'imprécision dans la représentation des connaissances.

Deuxièmement, les mécanismes de déductions mis en oeuvre par les ingénieurs relèvent bien souvent d'un **compromis** entre plusieurs règles différentes, certaines d'entre elles pouvant même agir en sens inverse (par exemple choix entre une machine chargée mais rapide et une autre plus lente mais libre). Dans une situation donnée, plusieurs règles doivent être prises en compte, chacune participant pour une part à la valeur du résultat final. Par exemple, pour un produit dont la prochaine opération peut être réalisée

indifféremment sur deux stations (station_a et station_b), les deux règles suivantes peuvent avoir été définies :

Règle 1 : Si buffer(station_a) est relativement vide
alors sélectionner ststion_a

Règle 2 : Si taux_production(station_b) est faible
alors sélectionner station_c

Le mécanisme séquentiel des moteurs d'inférence traditionnels s'avère ici inadapté, car ne prenant pas en compte le degré de vérité de chacune des assertions. La logique floue offre, quant à elle, un mécanisme d'inférence parallèle efficace : l'ensemble des règles et connaissances participent au calcul de la décision, selon leur degré de vérité respectif, permettant ainsi le calcul d'un compromis.

Une description des mécanismes experts flous et leur application au calcul des règles de destination est donnée dans [Datin, 1994].

Les règles expertes font appel en entrée à des variables accessibles sur le déroulement de la fabrication et délivrent en sortie une décision sur la variable attendue par l'automate. Le formalisme de ces règles n'a pas été fixé dans le cadre de nos travaux. Les tests que nous avons menés pour valider l'approche (notamment pour vérifier que ce mécanisme ne sature pas les échanges entre superviseur et automates) ont été réalisés à l'aide d'un interpréteur et d'un langage dérivé du Pascal. Un travail de développement à venir est d'établir un outil convivial d'édition de ces règles.

En ce qui concerne les performances de l'inférence elle-même, l'approche par logique floue offre l'avantage d'être parfaitement déterministe : le temps de réponse de l'inférence peut en effet être garantie, la notion indéterministe de bouclage (par chaînage avant ou arrière) propre aux systèmes experts classiques n'existant pas.

Remarque :

Le mécanisme de type système expert mis en oeuvre pour les règles de destination pourra être utilisé, dans une extension future du système, pour exploiter la flexibilité sur les opérations (cf. § 1.4.3). Le principe retenu est analogue : à l'issue d'une opération à un niveau local donné, une requête sera émise vers le niveau global qui retournera en réponse l'opération suivante à réaliser si un choix doit être effectué. Là encore, la centralisation de la décision, tout en permettant une conduite plus fine, ne devra pas être érigée en principe. Elle est destinée à venir en complément d'un mécanisme simple et réparti dont elle améliore les performances en marche normal, mais en aucun cas à remplacer ce mécanisme.

13.3.3 Application à l'exemple du site DRD

Nous allons illustrer l'utilisation des règles de destination sur l'exemple de la cellule d'assemblage du site.

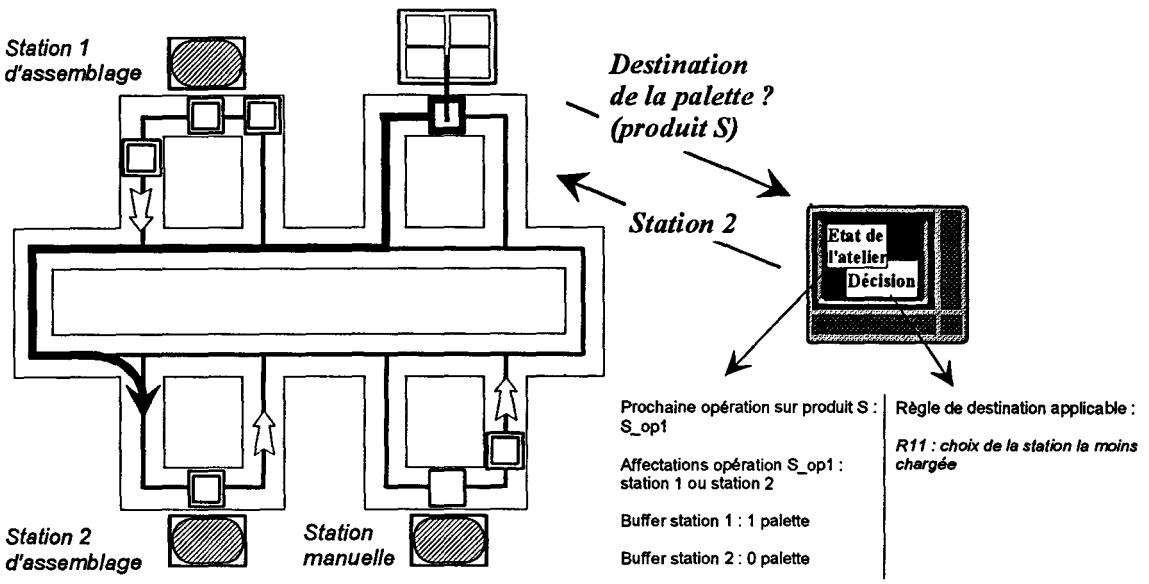


Figure 13.4 : fonctionnement des règles de destination.

Une première stratégie, illustrée sur la figure 13.4, consiste à calculer la destination de la palette au niveau de l'indexeur de la station précédente. Deux solutions sont alors possibles, suivant la nature de la station :

- solution 1 : déclenchement des opérations à effectuer sur le produit, attente du résultat des opérations, puis interrogation du superviseur ;
- solution 2 : envoie d'un message à destination du superviseur, puis déclenchement des opérations.

La première solution permet de prendre en compte dans la décision le résultat de la dernière opération. Il est ainsi possible d'orienter les produits en fonction de leur état de conformité déterminé à l'issue d'une opération de contrôle. La seconde solution permet, quant à elle, le calcul de la destination suivante en temps masqué par rapport à l'opération courante (l'état de conformité à l'issue de l'opération ne pouvant pas être déterminé ou étant supposé toujours conforme). Cette dernière solution, beaucoup moins contraignante en terme de temps de réponse, sera privilégiée.

Le principal inconvénient de cette première stratégie est lié à la nécessité de mettre en oeuvre une *stratégie de recouvrement* en cas de panne des moyens correspondant aux destinations choisies. Si la panne est importante, les produits doivent en effet être réorientés sur les moyens encore disponibles afin de ne pas saturer l'installation. Une solution au problème du recouvrement est présentée au § 13.4.2.

Une seconde stratégie consiste à prendre la décision au niveau des aiguillages. On reporte ainsi au plus tard le choix de la destination, permettant ainsi une meilleure réactivité du système en cas de changement de mode de marche des moyens de fabrication. Toutefois, cette solution impose des échanges de communication très rapides entre superviseur et automates.

Effectivement possible dans le cas du site, cette stratégie extrême n'est pas toujours applicable en milieu industriel où les réseaux sont bien souvent calculés au plus juste.

Ces deux options ne sont en aucun cas exclusives. Entre les deux comportements extrêmes consistant à prendre l'ensemble des décisions soit au plus tôt, soit au plus tard, un comportement médian doit être trouvé, fonction de l'installation et des objectifs fixés.

13.4 Signalisation et coordination des transferts

13.4.1 Réservation lors des changements de campagne

La notion de signal associée aux règles d'aiguillage rejoint dans la théorie celle associée aux bons de fabrication. En effet, nous avons vu qu'une règle d'aiguillage pouvait être considérée comme une requête à destination d'un moyen particulier (l'aiguillage). Cette analogie est particulièrement vraie dans l'approche "dynamique" où une règle n'est valide que le temps d'une campagne donnée. De ce fait, il est normalement possible d'appliquer aux règles d'aiguillage les mêmes principes d'instanciation que ceux appliqués aux bons de fabrication : notamment, lors du téléchargement, une phase de réservation est nécessaire afin de d'attendre que les cantons suivants du système de transfert soient correctement paramétrés. Cette phase de réservation est obtenue en téléchargeant les règles avec un signal placé à OFF (règle désactivée). L'activation peut ici aussi se faire suivant diverses stratégies (cf. § 11.6.2).

Nous étant limités à l'approche "statique" que nous considérons suffisante pour la plupart des applications industrielles actuelles (cf. remarque du § 13.2.4), cet emploi de la signalisation n'est plus justifié.

Néanmoins, la signalisation, appliquée aux règles d'aiguillage, trouve une autre application, fondamentale dans un contexte industriel : le recouvrement en cas de panne de l'un des moyens de production.

13.4.2 Réaction aux pannes

En cas de panne d'un moyen de production, ou, plus généralement, d'un changement de mode de marche de ce moyen, deux réactions sont possibles, suivant la gravité de l'arrêt occasionné : la mise en oeuvre d'une procédure simple de *recouvrement* ou la *reconfiguration* complète du système. Nous précisons ci-dessous la distinction que nous opérons entre ces deux notions.

Recouvrement vs reconfiguration

Les divers modes de marche d'un moyen peuvent être classés en trois grandes catégories vis-à-vis d'une opération affectée à ce moyen[3] :

- disponible,
- indisponible temporairement (exemple : enrayage),
- indisponible (exemple : panne machine, bris d'outil).

Les modes appartenant aux deux premières catégories sont équivalents lors des calculs d'ordonnancement des campagnes et, éventuellement, des règles d'aiguillages associées : la durée des arrêts de production relatifs à ces modes est supposée insignifiante par rapport à la durée prévue pour réaliser la campagne et, par conséquent, peut être négligée dans les calculs. Le passage de l'un de ces modes à l'un des modes "indisponible" (et vice versa) nécessite, par contre, une *reconfiguration* du système : recalcul des règles d'aiguillage, voire changement de campagnes si l'une des campagnes en cours ne peut plus être produite. Le but de la reconfiguration est de générer les conditions d'un nouvel état stable en adéquation avec le nouvel état du système.

Le recouvrement, quant à lui, consiste à gérer les produits des campagnes en cours dont la destination, établie par le mécanisme des règles de destination, n'est plus valide en raison d'un changement de mode de marche du moyen associé à cette destination.

Utilisation de la signalisation pour le recouvrement

Nous proposons ici une procédure, simple mais robuste, afin de maintenir les tables de règles d'aiguillage cohérentes avec le mode de marche des moyens.

Afin de rester en accord avec le principe de délocalisation utilisé jusqu'à présent, tout changement de mode de marche d'un moyen sera géré par le niveau local. Ce changement sera systématiquement répercuté vers le niveau global. Ce niveau étant en charge de la coordination, c'est à lui que revient le calcul du recouvrement à mettre en oeuvre. La procédure est la suivante :

- prise en compte au niveau des règles de destination du nouveau mode marche, afin de ne plus envoyer de produits vers le moyen défaillant. Cela peut être effectué au moyen d'un système de signalisation analogue à celui mis en place sur les règles d'aiguillage.
- mise à OFF de toutes les règles d'aiguillage dont le champ destination correspond à la destination incriminée. Dans le cas où les modes de marche sont distingués par opération, il est possible de tenir compte du champ opération de la règle afin de maintenir l'activité du moyen au niveau des autres opérations : seules les règles dont le champ opération correspond à l'opération incriminée, ainsi que les règles non valuées sur ce champ, doivent être désactivées.

La procédure inverse s'applique bien sûr lorsque le moyen repasse dans l'un des modes de marche des catégories "disponible" ou "arrêt temporaire".

---

[3] Il est à noter que la distinction des modes de marche d'un moyen suivant les opérations n'est que très rarement effectuée en pratique.

L'application de cette procédure permet d'éviter le routage des produits vers le moyen défaillant. Néanmoins, un produit à destination de ce moyen ne pourra poursuivre sa gamme qu'à la condition suivante : la prochaine opération à réaliser doit être disponible au niveau d'un autre moyen accessible à partir de la localisation courante du produit. Les règles discriminées par opérations, de portée plus générale, s'appliquent alors. Il faut, pour cela, qu'elles aient été correctement configurées au préalable. C'est le cas lorsque l'on se conforme à la seconde politique de mise en oeuvre des règles d'aiguillage présentées au § 14.5.3.

13.5 Fonctionnalités locales et globales

Nous résumons, dans les tableaux suivants, les fonctionnalités du module de pilotage liées à la fonction coordination des transferts.

Deux groupes de fonctionnalités apparaissent : celles relatives au contrôle du trafic (règles d'aiguillage), et celles traitant du routage (règles de destination). Dans ces deux groupes, nous avons distingué les fonctionnalités appartenant au niveau global de celles réalisées au niveau local. Les remarques faîtes à ce sujet au paragraphe 11.7 s'appliquent ici également.

Niveau global	♦ calcul "en ligne" des règles d'aiguillage (optionnel)
	♦ édition des règles d'aiguillage
	♦ téléchargement des tables de règles
Niveau local	♦ gestion et application de la signalisation
	♦ application des règles d'aiguillage : - choix du canal d'entrée - choix du canal de sortie

Fonctionnalités liées au contrôle du trafic.

Niveau global	♦ édition des règles de destination
	♦ application des règles de destination sur requête du niveau local
Niveau local	♦ émission des requêtes vers le niveau global
	♦ mise à jour des informations associées aux objets : - item prochaine destination

Fonctionnalités liées au routage.

Chapitre 14

Calcul des règles d'aiguillage

Introduction

Destinées à réguler le trafic, les règles d'aiguillage ont une influence fondamentale sur les performances du système. L'établissement du type de règle et des coefficients associés à implanter au niveau de chaque aiguillage doit donc être effectué avec soin. Ce problème fait l'objet du présent chapitre.

Nous exposons tout d'abord les principes généraux relatifs à la prise de décision face à un problème de choix. Fonctions d'évaluation et fonctions de stratégies seront définies.

Nous verrons que le calcul des fonctions d'évaluation se ramène à un problème de recherche combinatoire. Une méthode de résolution adaptée à notre problème est ensuite exposée. Cette méthode s'inspire de travaux menés dans le cadre du projet TE/ECLille/MRT [TE/ECLille, 1992] [Vidril, 1990].

Nous établirons également que, pour la plupart des stratégies souhaitées par les utilisateurs, les deux types de règles présentés au chapitre 13 sont suffisants.

14.1 Principe général

Rappelons tout d'abord que l'objectif fondamental du pilotage est le règlement des indéterminismes apparaissant au niveau de la partie commande lorsque plusieurs évolutions concurrentes du procédé sont possibles. Ces choix sont liés aux flexibilités que présente le système. Ils sont clairement identifiables sur le modèle de simulation (cf. chapitre 9), non déterministe : ils donnent lieu

aux divers conflits apparaissant sur le RdP (conflit transition et conflit jeton – cf. annexe A).

Il existe, par exemple, trois cas de conflit transition : l'un apparaît lorsqu'un choix de trajectoire s'offre à un objet mobile (conflit transition au niveau des réseaux transitiques), l'autre lorsque deux assemblages sur un produit peuvent être réalisés au même instant (conflit transition au niveau des gammes), et le dernier lorsqu'une ressource peut réaliser indifféremment plusieurs opérations (conflit transition au niveau des réseaux comportementaux).

L'idéal pour résoudre ces conflits serait, pour chaque choix se présentant, d'envisager toutes les conséquences possibles jusqu'à achèvement de la production et d'adopter la décision maximisant un critère établi par l'opérateur. Cependant, explorer la totalité de la combinatoire des choix en temps réel pour déterminer l'optimum absolu du critère s'avère en fait illusoire avec les moyens de calcul actuels. Une simplification est donc nécessaire pour limiter l'explosion combinatoire.

Nous nous attachons dans ce chapitre au seul problème du choix de la trajectoire des mobiles dans l'atelier. Ce problème est lié au paramétrage des règles d'aiguillage. Avant d'exposer la simplification retenue pour ce problème, nous rappelons ci-après quelques principes relatifs aux techniques de prise de décision.

14.1.1 Prise de décision dans un problème de choix.

En face d'un problème de choix où il faut prendre une décision, on utilise en général une fonction permettant d'associer une valeur à chacun des choix possibles.

La forme générale d'une telle fonction est la suivante :

$$S = \sum_i \mu_i . f_i$$

Chaque f_i est une *fonction d'évaluation* d'un critère.

Chaque μ_i est un coefficient de pondération permettant de définir l'influence du critère. Le produit $\mu_i.f_i$ doit être sans dimension dans le cas où l'on a plusieurs critères.

S peut être définie par morceaux sur son domaine de définition. μ_i peut être variable.

Suivant les cas d'application, on rencontrera les termes de *fonction de stratégie*, *fonction de décision*, *fonction d'utilité* ou *fonction de pénalité* pour désigner S.

Dans le cas où un seul critère est utilisé, le problème sera qualifié de *mono-critère*, dans le cas inverse il sera dit *multi-critères*.

Une fois la fonction de stratégie définie, la prise de décision s'effectue en adoptant le choix minimisant ou maximisant S.

14.1.2 Simplification pour le problème du choix de trajectoire

La méthode consiste à simplifier les trajectoires envisagées en ne prenant en compte de manière exacte que les conséquences du premier choix, puis d'estimer les choix suivants. Plus précisément, la trajectoire d'un mobile dans le système est divisée en deux parties :

- la première où l'on tient compte de l'état courant du système pour calculer de manière exacte les critères retenus ;
- la seconde où l'on suppose que le mobile est seul dans le système, tous les moyens lui étant dédiés, pour déterminer une estimation des critères.

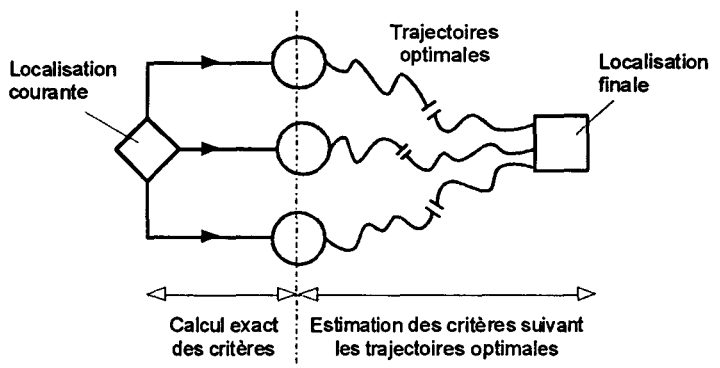


Figure 14.1 : partition des trajectoires possibles d'un mobile dans l'installation

Deux critères nous ont semblé pertinents : la durée des trajets et la durée d'occupation des moyens au cours de ces mêmes trajets. Déclinés suivant les deux parties de la trajectoire nous obtiendrons ainsi quatre fonctions d'évaluation. Ces fonctions d'évaluation seront utilisées pour évaluer en temps réel la valeur des divers critères (temps libre, temps d'achèvement, etc.) qui, combinés suivant certaines **stratégies** (achever la production au plus tôt, minimiser la charge sur les machines, etc.), permettront de choisir l'orientation des mobiles.

Il faut en permanence garder à l'esprit le fait que les trajectoires décrites ci-dessus sont virtuelles et n'incluent pas forcément la trajectoire réelle que suivra le mobile. Elles ne servent qu'à permettre la prise de décision à un instant donné en limitant le nombre des trajectoires considérées pour la recherche de l'optimum. Cela revient en fait à élaguer l'arbre combinatoire des choix avant examen. Le trajet réel dépend en fait de la position des autres mobiles, de l'occupation des machines, de la taille des files d'attentes, ... à tout instant.

14.2 Formalisation et notations

Considérons un mobile se déplaçant dans l'atelier sur un système de transfert donné. Nous voulons déterminer les règles à implanter au niveau des aiguillages de ce système de transfert.

Nous admettrons que le mobile en question peut être caractérisé par deux paramètres : sa localisation et son stade dans la gamme. La localisation ne pose pas de problème ; nous renvoyons le lecteur au modèle de l'installation établi au chapitre 8. Quant au stade, nous supposerons que le mobile peut toujours être associé à un article, donc une gamme. C'est évidemment le cas si le mobile concerné est lui-même un article. Lorsqu'il s'agit d'un support mobile (palette ou plateau), le stade caractéristique sera celui du sous-ensemble principal transporté. Dans le cas d'un support vide, une gamme particulière sera définie, permettant de spécifier la séquence d'opérations à effectuer sur ce support jusqu'à son chargement, considéré comme un assemblage. Un support vide est donc un article particulier, non consommable.

Au cours de son passage sur les divers moyens desservis par le système de transfert, la localisation et le stade du mobile évoluent. Il s'agit d'amener le mobile en un lieu du système de transfert et à un stade convenu. Ce stade n'est pas forcément le dernier stade de la gamme, le mobile pouvant quitter le système de transfert avant achèvement complet de la gamme.

14.2.1 Notion de position

Pour **une** gamme donnée, l'ensemble des lieux modélisant le système de transfert peut être classé en :

- *lieux passifs* : le mobile y passe sans subir de transformation, i.e. le stade n'évolue pas (portions de convoyeurs, aiguillages) ;
- *lieux actifs* : le mobile peut y subir une transformation (indexeur) ; il peut aussi ne faire qu'y passer (postes en ligne).

Il est important de noter que ce classement diffère à priori d'une gamme à l'autre.

La notion de *position* est définie par rapport à ce classement de la façon suivante :

- à chaque lieu passif, nous associerons une position unique, qualifiée de passive ;
- à chaque lieu actif, nous associerons autant de positions que d'opérations pouvant y être effectuées (positions dites actives), plus éventuellement une position supplémentaire si le lieu peut être passif vis à vis du produit (position dite passive).

Pour un mobile en un lieu donné du système de transfert, il est possible de lui affecter une position et une seule. La position représente en fait une double

information : d'une part le lieu physique occupé par le mobile, et, d'autre part, la dernière opération que ce mobile a subi en ce lieu.

La notion de position permet donc de localiser sur l'installation les diverses opérations à réaliser : elle synthétise le modèle de l'installation et des affectations.

14.2.2 Indexage utilisé

Les divers stades, positions et moyens concernés seront codés par des entiers afin de les manipuler plus rapidement dans la réalisation informatique des algorithmes que nous utiliserons ci-après.

Nous désignerons par :

$\mathbf{i} \in I = \{1, \ldots, I\}$,	les stades successifs associés au mobile (établis à partir de la gamme opératoire à "jouer") ;
$\mathbf{j} \in \mathcal{J} = \{1, \ldots, J\}$,	les positions pouvant être "occupées" par le mobile ;
$\mathbf{k} \in \mathcal{K} = \{1, \ldots, K\}$,	les différents moyens disponibles desservis par le système de transfert.

On nommera **mobile(i, j)**, un mobile au stade i en position j.

Nous désignerons par (i_f, j_f), le stade terminal et la position finale souhaitée du mobile.

14.2.3 Matrices IJJ et IJI

Pour un mobile associé à une gamme donnée, la description des évolutions possibles du couple (i, j) peut être ramenée à la donnée de deux matrices.

La matrice $[IJJ(i,j)]_{(i,j)\in I\times\mathcal{J}}$ donne, pour un mobile au stade i s'apprêtant à quitter la position j, l'ensemble des positions accessibles.

La matrice $[IJI(i,j)]_{(i,j)\in I\times\mathcal{J}}$ donne, pour un mobile au stade i et arrivant dans la position j, le stade résultant de l'application de l'opération associée à la position j.

Ces données étant définies pour une gamme donnée, le calcul des fonctions d'évaluation que nous allons maintenant exposer doit être répété pour **chaque gamme**.

Remarque :

Conformément à la définition que nous en avons donnée au cours de l'analyse, une gamme définit un séquencement **linéaire** d'opérations à effectuer

pour mener un article d'un stade initial à un stade final[1]. Par contre, chaque opération peut être affectée à un ou plusieurs moyens.

La méthode s'applique également dans les cas, non considérés dans notre travail, des systèmes présentant également une flexibilité sur l'ordre des opérations. La notion de gamme doit alors être redéfinie, mais les notations et matrices utilisées s'appliquent telles quelles.

14.3 Les fonctions d'évaluation

14.3.1 Fonctions d'évaluation sur la première partie de trajectoire

Considérons à un instant donné un mobile(i,j). Ce mobile peut se diriger à priori vers une position j_{acc} prise dans l'ensemble IJJ(i,j) des positions accessibles à partir de j. Certaines de ces positions lui seront bien entendu interdites : d'une part, par le stade du sous-ensemble transporté (assemblage déjà effectué ou impossible : IJI(i,j_{acc}) non défini), et d'autre part, par la présence des autres mobiles sur le système (files pleines ...).

Désignons par E = {j_{s1}, j_{s2}, ...} l'ensemble des successeurs non interdits de j pour la mobile (i,j). Le mobile prendra en position $j_s \in E$ un stade i_s tel que :

- i_s = IJI(i, j_s) = i si j_s est passive
- i_s = IJI(i, j_s) si j_s est active

Le temps mis par le mobile pour passer de (i,j) à (i_s,j_s) dépend donc à priori de i, j, j_s et de l'état d'occupation éventuelle de la zone physique associée à j_s par d'autres mobiles. C'est ce temps que l'on cherche à déterminer.

Une première hypothèse, réaliste, permet de simplifier le problème en supposant que le temps d'assemblage pour passer d'un stade i à un stade i_s est indépendant du stade i. Cette hypothèse, valide dans le cas des gammes linéaires considérées dans ce travail, demande à être nuancée dans le cas de flexibilité sur le séquencement des opérations (cf. [TE/ECLille, 1992]).

Une seconde hypothèse est de supposer le mobile seul sur le système de transfert. Nous noterons **A(j,j_s)** le temps alors trouvé pour passer de (i,j) à (i_s,j_s) et **B(j,j_s,k)** le temps consommé par le moyen k au cours de ce passage.

[1] Rappelons que le modèle synthétique établi pour le pilotage, ne représente pas une gamme, mais un ensemble de gammes (gammes normales pour chaque variante et gammes de retouche).

Si j_S est occupé par d'autres mobiles il faudra déterminer de façon exacte ou approché le temps d'attente dû à l'occupation de j_S que l'on notera $T_{occup}(j_S)$.

Le temps de passage de (i,j) à (i_S,j_S) est alors donné par :

$$A(j,j_S) + T_{occup}(j_S)$$

Le temps d'occupation du moyen k reste inchangé : $B(j,j_S,k)$.

Remarques :

Si j' n'est pas un successeur immédiat possible de j ($j' \in IJJ(i,j)$ et IJI(i,j') non défini ou IJJ(i,j) = {}) on pose $A(j,j') = +\infty$ et $B(j,j',k) = 0$;

Les temps A(j,j') et B(j,j',k) sont établis de façon expérimentale par mesure directe sur le site. Il convient, pour éviter toute ambiguïté, de bien préciser à quoi correspond physiquement, pour un mobile, le passage de la situation (i,j) à la situation (i_S,j_S). Ainsi sur l'exemple de la figure 14.2 représentant une portion de convoyeur à palettes, est-il possible de définir les éléments A(1,2), A(2,3) et A(1,3) de la façon suivante :

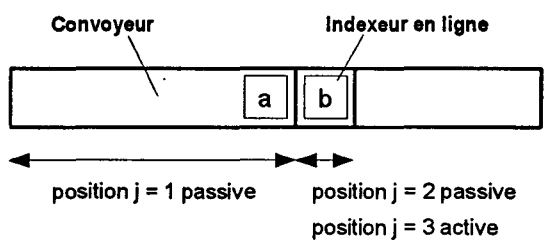


Figure 14.2 : exemple de définition des éléments de la matrice A.

- A(1,2) = temps mis par une palette pour aller de a en b sans y subir d'opération ;
- A(2,3) = temps mis par une palette pour subir, en b, l'opération associé à j = 3 ;
- A(1,3) = $+\infty$ (on considère que la position 3 n'est pas successeur de 1).

14.3.2 Fonctions d'évaluation sur la seconde partie de la trajectoire

Après avoir déterminé le temps pour passer de (i,j) à (i_S,j_S), il nous reste à calculer le temps minimum pour passer de (i_S,j_S) à (i_f,j_f).

Ce temps dépendant des positions courantes et futures de l'ensemble des mobiles, un calcul exact s'avère impossible. Aussi supposerons-nous que le mobile(i_S,j_S) est seul sur le système de transfert. Il devient alors possible de déterminer en un temps correct la trajectoire dans I×J optimale (au sens de la durée) pour achever la gamme. Les fonctions d'évaluation des critères seront établies pour cette trajectoire.

Nous noterons **OPTA(i,j)**, le temps calculé le long de cette trajectoire et **OPTB(i,j,k)** la durée d'occupation du moyen k pour la même trajectoire.

L'algorithme permettant de calculer OPTA et OPTB est dérivé de travaux effectués dans le cadre d'un précédent projet ESPRIT [ESPRIT 384, 1990]. Ces résultats étant protégés par la propriété industrielle, le détail de l'algorithme ne peut être donné dans ce mémoire. Néanmoins, nous en présentons ci-après le principe. L'annexe B présente les résultats du calcul de OPTA sur l'exemple du site DRD.

Principe de l'algorithme :

L'algorithme s'appuie sur le principe d'optimalité de Bellman : "tout sous-chemin d'un chemin optimal est optimal" utilisé en programmation dynamique.

Supposons que l'on connaisse OPTA(i,j) pour tous les stades i successeurs d'un stade i_0 donné et tout j (cf. figure 14.3) . Pour calculer OPTA(i_0,j_0), il suffit d'explorer chaque branche de l'arbre des chemins possibles à partir de (i_0,j_0) jusqu'à ce que le stade évolue vers un stade i_k ($k\neq0$). En effet, le stade du sous-ensemble porté par le mobile ne pouvant régresser, i_k appartient obligatoirement aux successeurs de i_0 ; donc la trajectoire optimale pour atteindre (i_f,j_f) à partir de i_k est connue. Or, d'après le principe de Bellman énoncé plus haut, il est inutile de considérer les trajectoires non optimales à partir de i_k. L'exploration de la branche peut donc s'arrêter.

OPTA(i_0,j_0) est obtenu en prenant l'optimum des branches explorées.

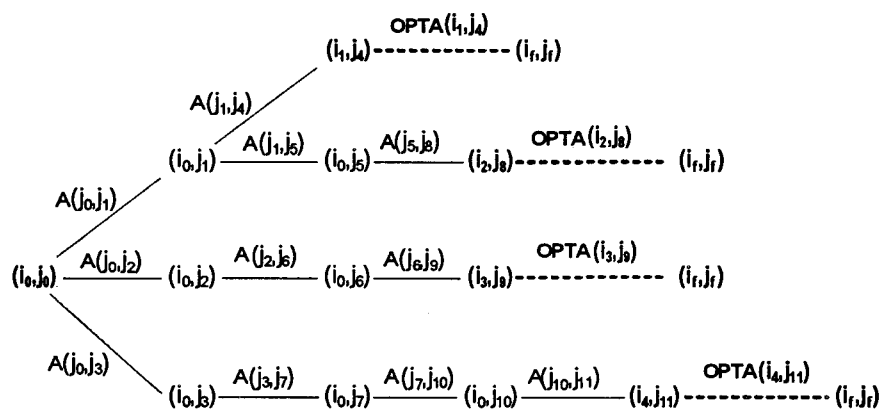


Figure 14.3 : trajectoires issues de (i_0,j_0).

14.4 Les fonctions de stratégies

Nous disposons donc pour chaque choix de successeur s, s = 1...S, des fonctions d'évaluation suivantes :

A(j,j_s) qui donne le temps de passage d'une position j à une position j_s immédiatement accessible ;

B(j,j_s,k) qui donne le temps consommé par le moyen k lors du passage d'une position j à une position j_s immédiatement accessible ;

OPTA(i_s,j_s) qui donne le temps minimal pour passer de (i_s, j_s) à (i_f, j_f);

OPTB(i_s,j_s,k) qui donne le temps consommé par le moyen k lors du passage de (i_s, j_s) à (i_f, j_f) suivant la trajectoire optimale ;

$T_{occup}(j_s)$ qui donne le temps d'occupation courant de la position j_s.

Ces fonctions permettent de donner une estimation du temps de parcours et de l'occupation des moyens sélectionnés sur la trajectoire que va suivre le mobile(i,j) orienté au premier coup vers la position j_s.

Il est à noter que seul $T_{occup}(j_s)$ nécessite un calcul en temps réel. Toutes les autres fonctions, calculées pour un mode de marche donné du système de fabrication (traduisant la disponibilité des ressources), restent valides tant que le mode de marche ne change pas.

Compte tenu des souhaits de l'utilisateur, on en déduit des stratégies multi-critères qui permettent de sélectionner j_s. Elles peuvent, par exemple, prendre en compte de façon plus ou moins prépondérante :

- la date due des sous-ensembles transportés,
- les priorités en sortie,
- l'équilibrage des moyens,
- etc.

Exemple :

A partir de ces fonctions d'évaluation nous pouvons associer à un mobile portant un article de date due DD (date limite à laquelle l'OF dont fait partie l'article doit être terminé) à l'instant t_0, des fonctions d'évaluation pour :

- le **temps pour achèvement** :

$$TA(i,j,j_s) = A(j,j_s) + T_{occup}(j_s) + OPTA(i_s,j_s)$$

Ce temps représente une approximation du temps minimum nécessaire pour amener, à partir du stade i et de la position j, le mobile au stade i_f en position j_f, en choisissant au premier coup la position j_s.

Le temps d'occupation correspondant du moyen k est alors donné par :

$$TB(i,j,j_s,k) = B(j,j_s,k) + OPTB(i_s,j_s,k)$$

- le **temps-libre** :

  $TL(i,j,j_S) = DD - t_0 - TA(i,j,j_S)$

  Ce temps représente une approximation de l'avance maximum (positif) ou du retard minimal (négatif) que possède le mobile en choisissant au premier coup la position j_S.

Si l'on veut piloter les mobiles en équilibrant temps-libre et priorité à la sortie, on peut faire le choix d'un seuil S_0 et définir la stratégie suivante :

$$S(i,j,j_S) = \begin{cases} TA(i,j,j_S) & \text{si } TL(i,j,j_S) > S_0 \\ K*TL(i,j,j_S) & \text{si } TL(i,j,j_S) \leq S_0 \text{ avec K coefficient positif.} \end{cases}$$

Pour un mobile(i,j) donné candidat à plusieurs positions, on sélectionne alors la transition s tel que $S(i,j,j_S)$ soit minimum.

Chaque fois qu'un nouveau conflit se présente, on **réévalue** les diverses fonctions d'évaluation associées aux mobiles avant d'appliquer une nouvelle fois les stratégies. Cette technique de réévaluation permanente donne à cette méthode une très grande robustesse malgré l'élagage important de l'arbre combinatoire au delà du futur immédiat. Par analogie avec l'automatique des processus continus, une telle méthode d'optimisation est dite à *horizon glissant.*

Il n'est d'ailleurs pas nécessaire de calculer $T_{occup}(j_S)$ avec une grande précision. Cette donnée peut même être négligée sans trop nuire aux qualités de la méthode.

14.5 Mise en oeuvre par les règles d'aiguillage

14.5.1 Cas général

La mise en oeuvre, à travers le mécanisme des règles d'aiguillage, des fonctions de stratégies définies par l'utilisateur s'effectue en définissant de nouvelles règles de choix d'un canal. Ces règles pourront alors être appliquées au niveau des aiguillages divergents pour le choix du canal de sortie.

Pour la stratégie énoncée dans l'exemple précédent, une telle règle pourrait être définie de la façon suivante :

S1(val1, val2, (AIG:i_1, v_1),...,(AIG:i_n, v_n))

avec :

val1 = K
val2 = S_0
$v_k = A(j,j_S^k)+OPTA(i, j_S^k)$ pour k = 1...n

en supposant l'information DD (due-date) associée aux objets mobiles. Les paramètres val1 et val2 fournissent les coefficients constants de la règle, tandis que les v_k donnent pour chaque canal AIG:i_k autorisé, la pondération à appliquer.

La durée d'occupation $T_{occup}(j_s^k)$ de chaque canal associé à la position j_s^k est supposée calculable localement. Cela est réaliste en raison du rapprochement physique des équipements associés, pouvant de ce fait être contrôlés par un même calculateur.

L'algorithme local associé à cette règle est le suivant :

Procédure choisir canal selon S1(val1,val2,(AIG:i_1,v_1),...,(AIG:i_n,v_n))

Pour k=1 à n
  Si DD-t_0-v_k-$T_{occ}(j_s^k)$ > val2
    $S_k = v_k + T_{occ}(j_s^k)$
  Sinon
    S_k = val1*[DD-t_0- v_k-$T_{occ}(j_s^k)$]
  Finsi
Finpour
Si $S_k < S^*$
  $S^* = S_k$
  $k^* = k$
Finsi
Choisir le canal k^*

De telles règles, établies pour une gamme et un état de marche donnés du système doivent, à priori, être mises en oeuvre par paramétrage dynamique (cf. § 13.2.4) calculé "en ligne" (cf. § 13.2.5), i.e. les coefficients des règles sont calculés par le superviseur au moment du lancement de l'OF (pour prendre en compte le mode de marche courant), puis téléchargés au niveau des automates. L'utilisation du discriminant "campagne", limitant la durée de vie de la règle à celle de l'OF, s'avère nécessaire.

Cette utilisation temps réel n'a toutefois pas pu être validée dans le cadre de ce travail. Cependant, d'après les performances obtenues "hors ligne", les algorithmes utilisés (pour le calcul de OPTA et OPTB, les matrices A et B étant établies une fois pour toutes) semblent suffisamment rapides pour faire l'objet d'une telle mise en oeuvre.

Toutefois, deux considérations essentielles nous incitent à penser que ce cas général n'est que très rarement utile en pratique.

14.5.2 Cas pratiques

Simplification liée à l'installation

La "configuration dynamique calculée en ligne" des règles d'aiguillage ne se justifie que dans le cadre d'ateliers très flexibles, encore très peu rencontrés

dans l'industrie. Il ne faut pas perdre de vue que les ateliers actuels dits flexibles ne présentent bien souvent qu'une flexibilité limitée :

- nombre de variantes (donc de gammes principales) limité ;
- affectations possibles souvent réduites à des postes identiques en parallèle (sauf dans les phases de montée en cadence où des postes manuels peuvent cohabiter avec des postes automatiques) ;
- pas de flexibilité sur l'ordre des opérations ;
- nombre de boucles (donc de gammes) de retouche limité ;
- généralement une seule campagne en cours (sauf naturellement lors des phases transitoires).

De fait, il apparaît, que le nombre de gammes peut facilement être énuméré. Il en est de même pour les modes de marche : seuls certains modes sont significatifs ; en outre, un mode donné peut généralement être considéré valide pendant toute la durée d'une campagne[2]. En conséquence, le nombre de configurations possibles du système (gammes×mode de marche) est limité. Les coefficients des règles peuvent donc être calculés une fois pour toutes, "hors ligne", pour chaque configuration. Si ce nombre est suffisamment faible le paramétrage du système de transfert peut également être effectué une fois pour toutes ; le mécanisme de configuration dynamique s'avère alors inutile ainsi que les fonctionnalités de niveau global correspondantes (cf. § 13.5).

Simplification liée aux stratégies

Une étude approfondie des stratégies que souhaitent pouvoir mettre en oeuvre les utilisateurs révèle que ces stratégies sont relativement simples.

Deux stratégies principales sont très souvent formulées :

- ♦ optimiser le temps de fabrication ;
- ♦ répartir les charges de certaines machines de façon équitable ou, plus rarement, suivant un taux spécifié, tout en optimisant le temps de fabrication.

La première stratégie se traduit, en utilisant les fonctions d'évaluation A et OPTA précédemment établies, de la façon suivante :

→ orienter les mobiles suivant la position j_S minimisant :

$$TA(i,j,j_S) = A(j,j_S) + T_{occup}(j_S) + OPTA(i_S,j_S)$$
$$\cong A(j,j_S) + OPTA(i_S,j_S)$$

En utilisant cette dernière approximation, TA est constante dans le temps pour chaque canal. Cette valeur peut donc être assimilée à une priorité : la mise

[2] Dans l'idéal, l'indisponibilité d'une machine ne devrait être due qu'aux opérations de maintenance, parfaitement planifiables. Un tel arrêt en cours de campagne nécessiterait, pour bien faire, un nouveau paramétrage des règles. Néanmoins, le mécanisme présenté au § 13.4.2 fournit une solution suffisante pour assurer la fin de campagne en cours (au besoin en diminuant la consigne) et changer de gamme.

en oeuvre de cette stratégie peut être effectuée au moyen des seules règles par priorité présentées au chapitre 13.

La seconde stratégie est plus délicate à mettre en oeuvre. Les fonctions d'évaluation à utiliser sont B et OPTB, disponibles pour chaque moyen dont les charges doivent être réparties. Ces fonctions permettent notamment d'estimer le temps d'occupation de chaque moyen k selon la trajectoire optimale (au sens de la durée) :

$$TB(i,j,j_s,k) = B(j,js,k) + OPTB(i_s,j_s,k) \qquad \text{avec } k = 1...K.$$

Dans le cas général, nous disposons, pour chaque canal de sortie d'un aiguillage, de K valeurs de TB ; nous disposons également d'un certain nombre d'équations spécifiant les relations entre charges (du type $C(k_i) = W.C(k_j)$ si l'on souhaite que le moyen k_j soit W fois moins chargé que k_i). La figure 14.4 illustre ce problème général.

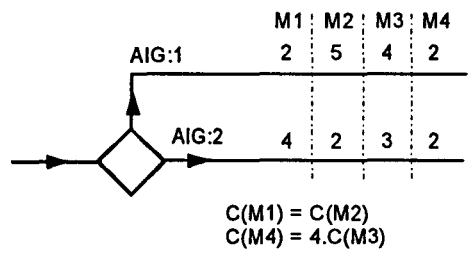


Figure 14.4 : équilibrage des charges - cas général.

Les valeurs indiquées correspondent aux durées TB d'occupation des moyens (notés Mi) suivant chaque canal. Par exemple, l'envoi d'un produit sur le canal AIG:1 entraîne une charge de travail (exprimé en unité de temps) de 2 unités sur M1, de 5 unités sur M2, de 4 unités sur M3 et de 2 unités sur M4.

La fonction de stratégie à mettre en oeuvre dans ce cas n'est pas triviale. Cependant, dans la majorité des applications pratiques, le problème se pose sous une forme simplifiée. Chaque canal de sortie d'un aiguillage peut, en effet, généralement être associé à un seul moyen de l'ensemble des moyens à équilibrer[3] ; envoyer un produit sur ce canal ne déséquilibre alors que les équations relatives à ces moyens. La figure 14.5 illustre ce cas pratique.

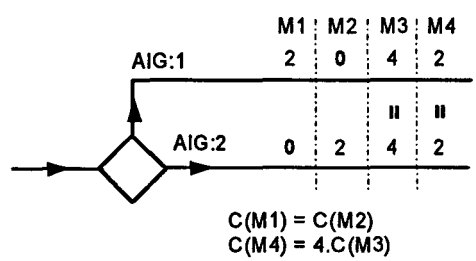


Figure 14.5 : équilibrage des charges - cas pratique.

[3] ou à un sous-ensemble de cet ensemble, constitué de moyens strictement identiques : ce sous-ensemble peut alors être assimilé à un moyen unique en divisant la valeur commune de TB par le cardinal du sous-ensemble.

Le canal AIG:1 est associé au moyen M1 tandis que le canal AIG:2 correspond au moyen M2. Choisir l'un ou l'autre de ces canaux n'influe en rien sur l'équilibrage des moyens M3 et M4 spécifié par l'équation C(M4)=4*C(M3).

Par contre, ce choix a une influence sur l'équilibrage des moyens M1 et M2 (équation C(M1)=C(M2)). La stratégie d'équilibrage des charges se résume donc au niveau de cet aiguillage à :

→ orienter les mobiles suivant la position j_s minimisant :

$$|\Sigma TB(i,j,j_s,M2) - \Sigma TB(i,j,j_s,M2)|$$

$\Sigma TB(i,j,j_s,k)$ représente une estimation de la durée d'occupation totale du moyen k.

En régime permanent, le respect de cette stratégie aboutit à une séquence cyclique du canal de sortie à choisir. Pour l'exemple, nous obtenons ainsi la séquence :

AIG:2 - AIG:1 - AIG:2 - AIG:1 - AIG:2 - AIG:1 - AIG:2

Il convient en effet d'envoyer trois produits sur le canal AIG:1 pour quatre produits envoyés sur AIG:2.

La règle décisionnelle permettant d'implanter une telle stratégie n'est pas compliquée : elle ne nécessite que la connaissance des durées estimées d'occupation des moyens associés à chaque canal. C'est une variante de la règle cyclique permettant de répartir dans le temps la charge des postes de travail. Il est d'ailleurs possible d'obtenir les mêmes ratios sur chaque canal par la règle cyclique RC((AIG:1,3), (AIG:2,4)). Cependant, on risque par cette règle d'obtenir un comportement saccadé entre les moyens M1 et M2, l'un étant chargé pendant que l'autre est libre. En fait, ce dernier problème ne se pose réellement que dans le cas où les ratios à respecter sur chaque canal sont très différents et/ou très grands. Répartir la séquence dans le temps s'avère alors nécessaire. Dans la majorité des applications pratiques, ces ratios sont souvent suffisamment faibles et une règle cyclique suffit généralement pour obtenir la répartition de charge souhaitée.

De cette discussion, il ressort que l'utilisation pratique des règles d'aiguillage peut se limiter aux seules règles par priorité et par pondération cyclique définies au chapitre précédent : l'utilisation des fonctions d'évaluations permet d'en définir les "meilleurs" paramètres selon les stratégies les plus courantes (produire au plus tôt en équilibrant ou non les charges) en assurant les propriétés de cohérence et de complétude.

14.5.3 Rapports avec les règles de destination

L'utilisation conjointe des règles d'aiguillage et des règles de destination doit être envisagée avec précaution. En effet, les deux mécanismes étant indépendant quant à leur mise en oeuvre, il est possible d'implanter des règles de destination contradictoires avec l'ensemble local de règles d'aiguillage.

Ce problème est analogue à celui de la cohérence des règles d'aiguillage : les propriétés de complétude et de cohérence doivent en fait être appliquées à l'ensemble étendu constitué des deux types de règles.

Nous avons vu que l'utilisation des fonctions d'évaluation permettait d'assurer ces bonnes propriétés : les chemins admissibles sont ceux fournissant des valeurs non infinies des critères. L'application de ces fonctions au niveau des règles de destination pourrait fournir un moyen de discriminer les destinations admissibles. Un guide méthodologique demanderait néanmoins à être établi.

Pour notre part, nous préconisons les deux politiques de mise en oeuvre suivantes, ne nécessitant pas de précautions excessives, et que nous avons testé avec succès sur le site DRD :

- **Politique 1 : utilisation du seul mécanisme des règles d'aiguillage.**
  - pour chaque configuration acceptable du système, précalcul "hors ligne" des règles d'aiguillage à l'aide des fonctions d'évaluation présentées dans ce chapitre ;
  - téléchargement des tables de règles "en une fois", i. e. sans mettre en oeuvre le mécanisme d'instanciation progressif par réservation des requêtes (cf. § 11.3) ; cette phase doit être effectuée avant le lancement effectif de la campagne associée à cette configuration ; elle est laissée à l'initiative de l'utilisateur.
- **Politique 2 : utilisation mixte.**
  - mise en oeuvre, par règles d'aiguillage discriminées par opération, de la stratégie "Greedy Boy" (cf. § 13.2.7) ; ce niveau permet d'assurer une marche dégradée en cas de panne du superviseur ;
  - implantation de règles d'aiguillage discriminées par destination pour chaque destination faisant l'objet d'une règle de destination au point suivant ;
  - mise en place de règles de destination utilisant les fonctions d'évaluation établies dans ce chapitre pour certaines décisions considérées comme étant "stratégiques".

D'autres politiques de répartition sont possibles. Elles demandent à être étudiées de façon à assurer le bon fonctionnement du pilotage dans toutes les situations. Quoiqu'il en soit, il revient en dernier lieu à l'utilisateur de vérifier que le comportement obtenu est correct (pas de blocage ou de bouclage infini) et satisfaisant (performances souhaitées du système). Cette vérification constitue l'une des raisons d'être de l'outil de simulation développé dans le cadre de ce travail (cf. chapitre 9 et annexe E).

14.5.4 Application à l'exemple du site DRD

Nous allons illustrer la démarche de calcul des règles d'aiguillage à partir des fonctions d'évaluation sur l'exemple du site DRD.

BU LILLE

La politique de mise en oeuvre choisie est la politique 1 : pour chaque configuration, les règles sont établies une fois pour toutes. La configuration choisie pour l'exemple est la suivante :

"produit_E / tous les moyens en état de marche"

La stratégie de production choisie est de produire au plus tôt. L'annexe B fournit, pour la configuration précédente, les valeurs des fonctions d'évaluation A et OPTA nécessaires.

Considérons l'aiguillage AIG_AM_2 contrôlant l'entrée de la station AFMA. Les positions concernées sont :

- la position 4 associée à l'aiguillage lui-même ;
- la position 5 associée au buffer BUFF_3 et donc au canal de sortie AIG_AM_2:conv ;
- la position 20 associée au buffer BUFF_11 et donc au canal de sortie AIG_AM_2:AFMA.

Pour une palette transportant un produit E au stade 1 (cf. annexe B pour la numérotation des stades) se présentant à cet aiguillage, l'estimation du temps avant achèvement selon chaque direction est donnée par (temps exprimé en 1/10 de seconde) :

- $TA(1,4,5) \cong A(4,5) + OPTA(1,5) = 30 + 1595 = 1625$
- $TA(1,4,20) \cong A(4,20) + OPTA(1,20) = 35 + 1275 = 1310$

Nous avons $TA(1,4,5) > TA(1,4,20)$ donc le canal AIG_AM_2:AFMA doit être choisi préférentiellement. Cela est obtenu par la règle :

RP(AIG_AM_2:AFMA, AIG_AM_2:conv)

En répétant l'opération pour chaque stade, on détermine ainsi l'ensemble des règles nécessaires au fonctionnement de l'aiguillage. Le tableau de règles à télécharger préalablement à toute campagne YY dans cette configuration est donc :

on	YY	∅	E_op1	RP	AIG_AM_2:AFMA AIG_AM_2:conv
on	∅	∅	∅	RP	AIG_AM_2:conv

Les tables à implanter au niveau des aiguillages AIG_AM_1, AIG_AM_3 et AIG_AM_4 se déduisent de la même façon.

Les résultats apparaissent évidents pour ce problème : les produits sont dirigés sur la première station capable de réaliser l'opération suivante de la gamme. Cependant, il n'en va pas toujours de même : le recours à une méthode algorithmique est alors appréciable.

Conclusion

Les travaux présentés dans ce mémoire concernent la phase de conception du système de contrôle / commande des systèmes de production flexibles dans l'industrie manufacturière. Afin de s'intégrer efficacement dans un système d'ingénierie simultanée, une méthodologie et des outils génériques doivent être mis en oeuvre à ce niveau. Seules la généricité et la formalisation permettent en effet la capitalisation et le partage des connaissances entre les diverses phases du processus de développement.

L'ensemble des outils intervenant dans la chaîne de contrôle / commande d'un SFPM ne saurait être traité dans le cadre d'un seul mémoire. Nous nous sommes, pour notre part, attachés à la définition de la fonction pilotage. Cette fonction est essentielle : elle constitue le lien entre les fonctions traditionnelles de la gestion de production (planification, ordonnancement, gestion des stocks) spécifiées suivant une logique produit et la commande spécifiée suivant une logique capteurs-actionneurs.

Résumé des travaux

Les travaux effectués se décomposent suivant trois facettes : analyse et modélisation, architecture du système de commande et enfin étude de la fonction pilotage.

Analyse et modélisation

La phase d'analyse constitue une phase essentielle dans toute démarche scientifique. Nous avons proposé une étude des éléments d'un SFPM organisée

suivant trois classes caractéristiques : les produits, les lieux et les ressources de production.

A partir des résultats obtenus, deux modélisations ont ensuite été établies. La première, destinée à la configuration des outils de pilotage, se focalise essentiellement sur l'aspect représentation. La seconde, bâtie sur l'outil RdP, est utilisée pour la simulation et l'analyse.

Les modèles établis en vue du pilotage sont déduits directement de l'analyse. Il s'agit du modèle des gammes opératoires permettant de gérer la production, du modèle de l'installation permettant de coordonner les transferts et enfin du modèle des affectations reliant les opérations apparaissant dans les gammes aux moyens de l'installation. Le modèle des gammes a particulièrement été soigné en vue de faciliter la prise en compte des variantes, des opérations de retouche et des rebuts apparaissant dans les systèmes flexibles considérés.

Le modèle RdP que nous proposons n'est pas, quant à lui, destiné à être manipulé directement par l'utilisateur. Il doit être utilisé dans des outils de simulation ou d'analyse. Il peut d'ailleurs être généré à partir des modèles de configuration utilisés par le pilotage. L'étape de modélisation a été menée à bien dans le cadre de cette recherche. Elle se veut systématique afin de s'intégrer aisément dans un outil informatique. Elle consiste, dans une première phase, à spécifier indépendamment les caractéristiques des produits, des lieux et des moyens, puis, dans une seconde phase, à connecter les divers modèles obtenus. Un outil de simulation utilisant le modèle ainsi établi a été spécifié puis réalisé. Cet outil permet de valider "hors-ligne" le fonctionnement dynamique des divers modules de supervision, dont celui de pilotage, proposés dans le cadre du projet SCOPES. Une autre utilisation possible du modèle, très prometteuse dans le cadre de l'ingénierie simultanée, est la réalisation d'outils d'aide à la conception fournissant des preuves formelles de certaines propriétés du modèle. Cette piste demeure néanmoins encore largement du domaine de la recherche en raison de la complexité des problèmes réels.

Architecture du système de commande

L'architecture du système de commande considérée dans ce travail se décompose suivant deux axes. Un premier axe, organisé suivant une décomposition structurelle du système de production (ateliers, cellules, stations par exemple), permet de partitionner le système de commande en divers centres décisionnels. A l'intérieur de chaque centre, un second axe, organisé suivant une décomposition fonctionnelle, permet de partager le processus d'élaboration des décisions entre plusieurs fonctions. Trois niveaux fonctionnels ont ainsi été mis en évidence selon le type de coordination réalisé.

Le premier niveau correspond aux fonctions ordonnancement et gestion des stocks. Il a en charge la coordination de la fabrication des divers articles entrant dans la composition d'un produit. L'un des problèmes essentiels résolus à ce niveau consiste à réaliser la synchronisation des divers rendez-vous d'assemblage.

Le second niveau, correspondant à la fonction pilotage, réalise la coordination des opérations permettant d'obtenir un article à partir de ses

articles constitutifs. Les contraintes de coordination à respecter à ce niveau sont données par les gammes opératoires.

Le troisième et dernier niveau correspond à la fonction de commande "bas niveau" des automatismes. Il réalise la coordination des divers systèmes opérants collaborant à la réalisation des diverses opérations autorisées par le pilotage.

Etude de la fonction pilotage

L'étude de la fonction pilotage a été décomposée suivant deux axes : un axe gestion de production et un axe gestion des transferts. Divers outils, répondant aux diverses fonctionnalités isolées suivant chaque axe, ont été établis. Ces outils, loin de contraindre l'utilisateur, lui permettent au contraire de mettre en oeuvre rapidement les méthodes et stratégies adaptées à son installation. En tenant compte en particulier des limitations matérielles et logicielles actuelles, ils réduisent le cadre d'étude aux seules stratégies utilisables en pratique, évitant ainsi toute "surprise" lors de la mise en oeuvre.

La philosophie suivie pour le développement de ces outils repose sur le concept de coordination : un centre global (superviseur) est chargé de distribuer le travail à des centres locaux subordonnés (automates). Ces centres locaux ne connaissent que l'ensemble des opérations qu'elles peuvent effectuer et non la façon de les organiser dans le but de satisfaire l'objectif global.

Deux types de structures de données, calculées globalement mais utilisées localement, assurent l'autonomie des contrôleurs, réduisant ainsi de manière importante le besoin en réseaux de communication performants (et chers !). Il s'agit des *bons de travail* permettant de gérer le travail en cours dans les ressources et des *règles d'aiguillage* assurant la gestion du transfert des pièces dans l'installation. Ces structures de données fournissent un support à la prise de décision décentralisée, garant de l'autonomie. Néanmoins, un mécanisme de décision centralisé, optionnel, a également été spécifié afin de permettre l'exploitation des informations globales. Ce mécanisme est appliqué au niveau des *règles de routage* réalisant le choix de la destination suivante des produits dans l'installation.

Le principe d'*instanciation dynamique* des données locales permet, en particulier lorsqu'il est possible d'appliquer la politique de *transit souple*, d'optimiser les changements de campagne de production. En réduisant les tâches répétitives effectuées par les opérateurs lors de ces phases, il facilite et accélère les changements de production, permettant ainsi la réduction de la taille de lot économique.

La phase de suivi de la production a été améliorée grâce au mécanisme de *scrutation asynchrone* du travail en cours. La connaissance de l'avancement de la production permet notamment de compenser le rejet de pièces dû à des défauts de qualité. Au lieu de compter sur une compensation prédictive à partir d'une estimation du taux de défaut, la compensation des pièces rejetées peut être ajustée précisément en temps réel. Les stratégies de compensation sont spécifiées directement sur le modèle des gammes opératoires, grâce au modèle des *graphes d'influence*. Le choix d'une stratégie de compensation adaptée permet de réduire les stocks et donc les coûts d'exploitation de la ligne.

Le traitement des produits à variantes multiples a également été facilité. L'introduction de variabilités dans la définition des gammes opératoires permet de limiter le nombre de gammes à configurer pour l'ensemble des produits de la ligne. Dans le cas où le nombre de variantes est très élevé, il devient indispensable de travailler à la commande afin de limiter les stocks. La réduction de la taille des lots et l'application de stratégies de compensation exactes, rendus possibles par les outils fournis, facilitent ce mode de fonctionnement.

Enfin, la centralisation de la configuration des données locales permet une simplification de la mise en oeuvre et de la maintenance du système. Elle permet notamment de faire évoluer très facilement le modèle des gammes, pour introduire de nouvelles variantes ou de nouvelles stations, sans avoir à modifier les programmes ou les paramètres de configuration des contrôleurs locaux. En outre, les traitements associés à la gestion de la production et à la conduite des transferts, qui font aujourd'hui l'objet de développements spécifiques à chaque nouvelle installation, sont standardisés. La durée et le coût des développements sont donc réduits.

En définitive, les outils de pilotage réalisés contribuent parfaitement à l'objectif principal de l'ingénierie simultanée, à savoir la réduction des délais de conception.

Perspectives

Le travail présenté dans ce rapport, bien que s'intégrant dans un projet de recherche, a pour finalité une utilisation industrielle. A cet effet, le prototype réalisé au cours de la recherche, doit être amélioré. Nous présentons ici les axes d'étude prioritaires.

A court terme, les actions doivent porter essentiellement sur l'achèvement du prototype afin de permettre une mise en oeuvre complète sur un exemple industriel. Il s'agit d'une part de compléter les modules fonctionnels existant, notamment ceux relatifs à l'ordonnancement et à la gestion des stocks, et d'autre part de formaliser les mécanismes d'échanges inter-modules.

Au niveau du module de pilotage, bien que la solution proposée soit fonctionnellement complète, plusieurs points nécessitent une étude complémentaire. Il s'agit de :

- la phase de reprise après une panne du superviseur;
- la prise en compte des modes de marche dégradés ;
- l'extension du principe de lancement aux actions de transfert (téléchargement dynamique des règles de trafic).

Les deux premiers points sont essentiels dans le cadre d'une mise en oeuvre industrielle.

A moyen terme, une extension du prototype serait l'amélioration des outils de configuration. Le mécanisme de configuration actuel est organisé selon les modules fonctionnels : le développeur de l'application fournit les données successivement pour chaque module fonctionnel. Ce mécanisme entraîne la répétition de certaines informations (il faut par exemple déclarer les

équipements à la fois pour le pilotage et l'analyse). Un mécanisme organisé suivant les objets (moyens de production, produits, etc.) s'avère plus adapté pour certaines données, tout en étant intuitivement plus compréhensible. La déclaration des équipements pourrait se faire à l'aide d'un outil graphique qui permettrait une construction interactive de l'installation. Une bibliothèque d'équipements prédéfinis pourrait en outre être établie, facilitant ainsi la capitalisation des connaissances. De façon analogue, la définition des gammes d'assemblage pourrait être réalisée graphiquement à l'aide d'une bibliothèque de symboles des différentes opérations d'assemblage.

Evaluation industrielle

Parallèlement à ces travaux, la confrontation du prototype à la réalité industrielle s'avère indispensable afin d'évaluer l'adéquation des solutions adoptées aux besoins. Deux types d'évaluation, ont été menées dans le cadre du projet SCOPES.

D'une part, la mise en oeuvre du prototype sur le site DRD a permis de mener une première évaluation concernant les choix technologiques adoptés. Il a ainsi été démontré l'adéquation de l'architecture de contrôle proposée en terme notamment de temps de réponse et de fiabilité.

D'autre part, une évaluation fonctionnelle a été menée par des utilisateurs potentiels externes au projet. Un groupe d'évaluation, regroupant des industriels intéressés par les outils d'ingénierie simultanée proposés a été constitué à cet effet. Citons notamment, parmi ceux particulièrement concernés par le système de supervision, PSA Peugeot Citroën (plus précisément le Centre de Recherche et d'Etudes de Produits Industriels) et MAA (Machines d'Assemblage Automatique). Les exemples de scénarios apportés ont permis de définir les besoins dans la phase initiale du projet. En outre, des évaluations des prototypes sont réalisées à intervalles réguliers par les membres du groupe. Les recommandations proposées permettent d'orienter les développements ultérieurs.

Une collaboration particulièrement étroite a été initiée avec MAA. Cette société est utilisateur potentiel du système de supervision en tant qu'intégrateur et fabricant de lignes d'assemblage. A ce titre, elle est particulièrement intéressée par les résultats du projet susceptibles d'élargir son offre. Un scénario de commercialisation de ces résultats est d'ailleurs en cours de définition.

Néanmoins, malgré ce partenariat actif et diversifié, seule la mise en oeuvre du système de supervision sur des cas industriels concrets permettra d'apporter une réponse définitive quant à l'intérêt du système de supervision proposé.

Les points suivants, concernant l'utilisation en exploitation du superviseur, peuvent d'ores et déjà faire l'objet d'une évaluation sur un exemple industriel :

- adéquation de l'approche métier retenue à la réalité du secteur manufacturier ;

- souplesse d'utilisation : possibilités de personnalisation, modification des règles de stratégie, convivialité de l'interface, utilisation du mode simulation ;
- qualité du pilotage réalisé : optimisation du changement des campagnes, respect des quantités demandées, respect des délais, accroissement de la productivité, réactivité en cas d'aléas.

D'autres points devront être analysés au fur et à mesure du développement des modules fonctionnels (qualité de l'ordonnancement et de la gestion des stocks) et des outils de configuration (souplesse d'utilisation).

L'évaluation sur un exemple industriel concret est actuellement en cours chez MAA. Les conclusions de cette étude devront permettre de valider les solutions retenues, notamment au niveau du pilotage, et d'engager la phase d'industrialisation du prototype permettant d'aboutir à terme à un réel outil industriel de supervision.

Bibliographie

[AFNOR, 1991] AFNOR. *Organisation de la production industrielle – Concepts fondamentaux de la gestion de production – Vocabulaire*. Norme française NF X 50-310, AFNOR, France, 1991.

[Agneray, 1995] S. Agneray. *Spécification de réalisation du module d'ordonnancement et de gestion des stocks*. Rapport technique Telemecanique, 1995.

[Agneray et Veaux-Logeat, 1994] S. Agneray, J.F. Veaux-Logeat. "A Functional Architecture for Concurrent Engineering : the SCOPES Project Innovation". In *Sharing CIM Solutions – Proceedings of the Tenth CIM-Europe Annual Conference*, 5-7 October 1994, Copenhagen, Denmark. J.K.H. Knudsen et al. Editeurs. IOS Press, pages 220-228, 1994.

[Agneray et Vidril, 1995] S. Agneray, M. Vidril. *Spécification de réalisation du module de pilotage*. Rapport technique Telemecanique, 1995.

[Aleonard *et al.*, 1994] C. Aleonard, E. Dindeleux et R. Soenen. "Conduite des systèmes de production". In *Les Techniques de l'Ingénieur*, ref. A8260-R7598, 1994.

[Amar *et al.*, 1992] S. Amar, E. Craye, J.C. Gentina. "Une méthode hiérarchique de spécification et de prototypage des systèmes de production flexibles". In *R.A.I.R.O APII*, AFCET Gauthier-Villars, vol. 26, n°5-6, 1992.

[Apave et Telemecanique, 1992] Travail collectif. *La sûreté des machines et installations automatisées. Apave et Telemecanique*. Sadave Editeur (référence D 313) et éditions CITEF/Telemecanique (référence MD1 SUR 1F), Paris, France, 1992.

[APICS/AFGI, 1991] AFGI (Association française de gestion industrielle), APICS (American Production and Inventory Control Society). *Dictionnaire des termes*

de gestion industrielle. Traduction française du dictionnaire APICS, Paris, mars 1991.

[Arnold, 1992] A. Arnold. *Systèmes de transitions finis et sémantique des processus communicants*. Editions Masson, 1992.

[Atabakhche Fallahi, 1987] H. Atabakhche Fallahi. *Utilisation conjointe de l'intelligence artificielle et des réseaux de Petri : Application au contrôle d'exécution d'un plan de fabrication.* Thèse de doctorat, Université Paul Sabatier, Toulouse, France, décembre 1987.

[Aubert-Barbier, 1990] M. Aubert-Barbier. *Modélisation et conduite des systèmes de production d'assemblage.* Thèse de doctorat, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace (ENSAE), Toulouse, France, juin 1990.

[Ausfelder, 1992] C. Ausfelder. *Modélisation de la Partie Commande et de la Partie Opérative d'un Système de Production Flexible*. Rapport technique du Laboratoire d'Automatique et d'Informatique de Lille (LAIL), Lille, France, mars 1992.

[Ausfelder *et al.*, 1992] C. Ausfelder, E. Castelain, J.C. Gentina. *Spécifications du Simulateur du Site Télémécanique*. Rapport technique du Laboratoire d'Automatique et d'Informatique de Lille (LAIL) en complément de [TE/ECLille/MRT, 1992], Lille, France, juillet 1992.

[Ausfelder, 1994] 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, Université des Sciences et Techniques de Lille Flandres Artois, Lille, France, mars 1994.

[Barbier et Jaulent, 1992] F. Barbier, P. Jaulent. *Techniques orientées objet et CIM*. Eyrolles, 1992.

[Berge, 1987] C. Berge. *Graphes*. Gauthier-Villars, 3e édition, 1987.

[Blanchard, 1979] M. Blanchard. *Comprendre, maîtriser et apliquer le GRAFCET*. Cepadues, 1979.

[Bouchet et Lellahi, 1987] P. Bouchet, K. Lellahi. *Pilotage d'atelier flexible par les techniques de base de données relationnelle et d'intelligence artificielle.* Rapport de recherche n°055. Laboratoire d'Informatique des Systèmes expérimentaux et leur Modélisation (I.S.E.M.), Université de Paris-Sud, France, mars 1987.

[Bourey, 1988] J.P. Bourey. *Structuration de la partie procédurale du système de commande de cellules de production flexibles dans l'industrie manufacturière.* Thèse de doctorat de l'Université des Sciences et Techniques de Lille Flandres Artois, Lille, France, mars 1988.

[Bourey, 1993] J.P. Bourey. *Méthode de conception de la commande de systèmes flexibles de production manufacturiers.* Mémoire d'Habilitation à diriger des recherches, Université des Sciences et Techniques de Lille Flandres Artois, Lille, France, février 1993.

[Bourjault, 1984] A. Bourjault. *Contribution à une approche méthodologique de l'Assemblage Automatisé : Elaboration Automatique des Séquences*

Opératoires. Thèse d'Etat, Université de Franche-Comté, Besançon, France, 1984.

[Bourjault *et al.*, 1987] A. Bourjault, D. Chappe, J.M. Henrioud. "Elaboration automatique des gammes d'assemblage à l'aide de Réseaux de Petri". In *R.A.I.R.O APII*, AFCET Gauthier-Villars, vol. 21, n°4, 1987.

[Bouteille *et al.*, 1992] N. Bouteille, P. Brard, G. Colombari, N. Cotaina, D. Richet. *Le GRAFCET*. Cepadues, 1992.

[Brams, 1983a] G.W. Brams. *Réseaux de Petri: théorie et pratique - Tome 1 : Théorie et analyse*. Masson, 1983.

[Brams, 1983b] G.W. Brams. *Réseaux de Petri: théorie et pratique - Tome 2 : Modélisation et applications*. Masson, 1983.

[Browne *et al.*, 1984] J. Browne, D. Dubois, K. Rathmill, S.P. Sethi, K.E. Stecke. "Classification of Flexible Manufacturing Systems". In *FMS Magazine*, avril 1984, pages 114-117.

[Cardoso, 1990] J. Cardoso. *Sur les réseaux de Petri avec marquages flous*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France, octobre 1990.

[Castelain, 1987] E. Castelain. *Modélisation et simulation interactive de cellules de production flexibles dans l'industrie manifacturière*. Thèse de doctorat, Université des Sciences et Techniques de Lille Flandres Artois, Lille, France, 1987.

[Cavaillé et Proth, 1987] J.B. Cavaillé, J.M. Proth. *SIPRODIS - Pratique de la simulation en production discontinue*. Collection Novotique, EC2, Nanterre, France, 1988, 333 pages.

[Combacau, 1991] M. Combacau. *Commande et surveillance des systèmes à événements discrets complexes : application aux ateliers flexibles*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France, décembre 1991.

[Courvoisier et Valette, 1986] M. Courvoisier, R. Valette. *Commande des procédés discontinus – Logique séquentielle*. Dunod, 1986.

[Cruette, 1991] D. Cruette. *Méthodologie de conception des systèmes complexes à événements discrets : Application à la conception et la validation hiérarchisée de la commande de cellules flexibles de production dans l'industrie manufacturière*. Thèse de doctorat, Université des Sciences et Techniques de Lille Flandres Artois, Lille, France, février 1991.

[Datin *et al.*, 1993], X. Datin, S. Agneray, B. Lavergne, S. Piou, M. Vidril. *Spécifications fonctionnelles d'un système de supervision pour ligne d'assemblage flexible*. Rapport technique interne, Schneider Electric, mars 1994.

[Datin, 1994] X. Datin. "Rule-based Distributed Flow Control of Flexible Assembly lines". In *Proceedings of the third International Conference on Automation Robotics and Computer Vision (ICARCV'94)*, Shangri-La, Singapour, 9-11 novembre 1994, vol. 1, pages 534-538.

[Descotes-Genon et Ladet, 1987] B. Descotes-Genon, P. Ladet. "Utilisation des Réseaux de Petri Colorés dans la simulation des systèmes de production". In

SIPRODIS - Pratique de la simulation en production discontinue. Actes du séminaire, 2 et 3 juin 1987, Paris, France. Collection Novotique, EC2, Nanterre, France, 1988.

[Dindeleux, 1992] E. Dindeleux. *Proposition d'un modèle et d'un système interactif d'aide à la décision pour la conduite d'atelier.* Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambraisis, Valenciennes, 1992.

[Donath et Graves, 1988] M. Donath et R.J. Graves. "Flexible assembly systems : an approach for near rel-time scheduling and routing of multiple products". In *International Journal of Production Research*, vol. 26, n°12, pages 1903-1919, 1988.

[Donath et Graves, 1989] M. Donath et R.J. Graves. "Flexible assembly systems : test results for an approach for near real-time scheduling and routing of multiple products". In *International Journal of Production Research*, vol. 27, n°2, pages 215-227, 1989.

[ESPRIT 384, 1990] Travail collectif. *ESPRIT Project 384 - Integrated Information Processing for Design, Planning and Control of Assembly - Final Task Reports.* Projet ESPRIT I n°384, 1990.

[Esquirol, 1987] P. Esquirol. *Règles et processus d'inférence pour l'aide à l'ordonnancement de tâches en présence de contraintes.* Thèse de doctorat, Université Paul Sabatier, Toulouse, 1987.

[Esteban, 1985] P. Esteban. *Sur la recherche d'algorithmes simplifiés d'analyse des réseaux de Petri.* Thèse de doctorat. Université Paul Sabatier, Toulouse, 1985.

[Estrailler, 1989] P. Estrailler. "The TAGADA project : ADA code generation from Petri net specification". In *Proceedings of ICNTSSD'89*, Caracas, décembre 1989.

[Falkner, 1986] C.H. Falkner. "Flexibility in Manufacturing Plants". *Proceedings of th Second ORSA/TIMS Special Conference on FMS.* Edités par K.E. Stecke et R. Suri, août 1986, pages 95-106.

[Gelende et Pujolle, 1982] E. Gelende, G. Pujolle. *Introduction aux réseaux de files d'attente.* Editions Eyrolles, Paris, 1982.

[GEMMA, 1981] Travail collectif. *Guide d'étude des modes de marches et d'arrêts.* ADEPA, 13/17 rue Périer, 92120 Montrouge, 1981.

[Harel, 1987] D. Harel. "Statecharts : a visual formalism for complex systems". In *Science of Computer Programming*, n° 8, 1987.

[Henrioud, 1989] J.M. Henrioud. *Contribution à la conceptualisation de l'Assemblage Automatisé : Nouvelle approche en vue de la détermination des Processus d'Assemblage* . Thèse de doctorat, Université de Franche-Comté, Besançon, France, décembre 1989.

[Henrioud et Bourjault, 1990] J.M. Henrioud, A. Bourjault. "Détermination des arbres d'assemblage". In *R.A.I.R.O APII*, AFCET Gauthier-Villars, vol. 24, n°6, pages 547-564, 1990.

[Héritier-Pingeon, 1991] C. Héritier-Pingeon. *Une aide à la conception de systèmes de production basée sur la simulation et l'analyse de données.* Thèse

de doctorat, Institut National des Sciences Appliquées de Lyon, Lyon, France, juin 1991.

[Hill, 1993] R.C. Hill. *Analyse orientée objets & modélisation par simulation.* Addison-Wesley, juillet 1993.

[Huber *et al.*, 1990] P. Huber, K. Jensen, R.M. Shapiro. "Hierarchies in coloured Petri nets". In *High-level Petri nets – Theory and application.* K. Jensen et G. Rozenberg éditeurs. Springer-Verlag, 1991.

[Huvenoit, 1994] B. Huvenoit. *De la conception à l'implantation de la commande modulaire et hiérarchisée de systèmes flexibles de production manufacturière.* Thèse de doctorat, Université des Sciences et Techniques de Lille Flandres Artois, Lille, France, octobre 1994.

[Kallel *et al.*, 1985], G. Kallel, X. Pellet, Z. Binder. "Conduite décentralisée coordonnée d'atelier". In *R.A.I.R.O APII*, AFCET Gauthier-Villars, vol. 19, n°4, pages 371-387, 1985.

[Knuth, 1973] D.E. Knuth. *The art of computer programming - Tome 4 : Combinatorial algorithms.* Addison Wesley, 1973.

[Le Gall, 1989] A. Le Gall. *Un système interactif d'aide à la décision pour l'ordonnancement et le pilotage en temps réel d'atelier.* Thèse de doctorat, Université Paul Sabatier, Toulouse, 1989.

[Le Moigne, 1984] J.L. Le Moigne. *La théorie du système général –Théorie de la modélisation.* 2nd édition. PUF, Paris, 1984.

[Leroux, 1994] F. Leroux. *Projet SCOPES. Etude du module de gestion des stocks.* Rapport de projet de fin d'étude, Ecole d'Ingénieurs de Tours, juin 1994.

[Lesage, 1991] J.J. Lesage. *Les outils de modélisation pour le cycle de vie des systèmes automatisés de production.* Génie automatique et production industrielle. Journées de synthèse, ISMCM, Saint-Ouen, France, 12 et 13 mars 1991.

[Martin, 1987] F. Martin. *Méthodologie de modélisation et simulation de systèmes complexes décrits par Réseaux de Petri Colorés.* Thèse de doctorat, Institut National Polytechnique de Grenoble (I.N.P.G.), Grenoble, France, avril 1987.

[Meyer, 1990] B. Meyer. *Conception et programmation par objets - Pour du logiciel de qualité.* InterEditions, 1990.

[Moalla, 1985] M. Moalla. "Réseaux de Petri interprétés et Grafcet". In *Technique et Science Informatiques (T.S.I.)*, vol. 4, n°1, pages 17-30, 1985.

[Pellet, 1984] X. Pellet. *Sur la hiérarchisation des décisions – Application à la conduite d'atelier.* Thèse de doctorat, Institut National Polytechnique de Grenoble (INPG), Grenoble, 1984.

[Peterson, 1981] J. Peterson. *Petri Net Theory and the modelling of systems.* Prentice Hall, 1981.

[Petri, 1966] C.A. Petri. *Communication with Automata.* Technical report N° RADC-TR-65-377, 1966.

[Sacerdotti, 1977] D. Sacerdotti (Stanford Research Institute). *A Structure for Plans and Behaviour*. Editions Elsevier, 1977

[SCOPES D3, 1993] Travail collectif. *SCOPES project - Deliverable D3 : Users Requirements*. Projet ESPRIT III n°6562, deliverable n° 3, juin 1993.

[SCOPES D4, 1993] Travail collectif. *SCOPES project - Deliverable D4 : Functional Specification of the System Modules*. Projet ESPRIT III n°6562, deliverable n°4, juin 1993.

[SCOPES D15, 1994] CRIF/WTCM - Industrial Automation Department. *SCOPES project - Deliverable D15 : First Specifications of Assembly Planning and Resource Planning Packages*. Projet ESPRIT III n°6562, deliverable n°15, mai 1994.

[SCOPES D18, 1994] CRIF/WTCM - Industrial Automation Department. *SCOPES project - Deliverable D18 : Second Specifications of Assembly Planning and Resource Planning Packages*. Projet ESPRIT III n°6562, deliverable n°18, à paraître en juin/juillet 1995.

[SCOPES D21, 1994] E. Verdebout (EPFL - Institut de Microtechnique). *SCOPES project - Deliverable D21 : Specification of Flow Control Techniques*. Projet ESPRIT III n°6562, décembre 1994.

[SCOPES D30, 1993] Travail collectif. *SCOPES project - Deliverable D30 : Specifications of the System Architecture*. Projet ESPRIT III n°6562, deliverable n°30, juin 1993.

[SCOPES D41, 1994] G. Wallace, I.P. Fan (Cranfield Institute of Technology). *SCOPES project - Deliverable D41 : Concurrent Engineering methodology for applying SCOPES technology*. Projet ESPRIT III n°6562, juin 1994.

[SCOPES Technical Annex, 1994] Travail collectif. *SCOPES project - Technical Annex*. Projet ESPRIT III n°6562, mai 1994.

[SCOPES Workshop 1, 1995] Travail collectif. *Simultaneous Engineering in der industriellen Montage*. Actes du séminaire sur SCOPES et l'ingénierie simultanée, IAT Universität Stuttgart, Stuttgart, Allemagne, 7 mars 1995.

[SCOPES Workshop 2, 1995] Travail collectif. *SCOPES workshop – Presentation notes*. Actes du séminaire sur SCOPES et l'ingénierie simultanée, CIM Institute, Cranfield University, Cranfield, Angleterre, 22 mars 1995.

[SCOPES Workshop 3, 1995] Travail collectif. *L'ingénierie concourante assistée par ordinateur : la solution SCOPES*. Actes du séminaire sur SCOPES et l'ingénierie simultanée, EPFL, Lausanne, Suisse, 4 avril 1995.

[SCOPES Workshop 4, 1995] Travail collectif. *Computer-Aided Concurrent Engineering : The SCOPES Solution*. Actes du séminaire sur SCOPES et l'ingénierie simultanée, CRIF/WTCM, Bruxelles, Belgique, 24 mai 1995.

[Sethi et Sethi, 1990] A.K. Sethi, S.P. Sethi. "Flexibility in Manufacturing : A Survey". In *The International Journal of Flexible Manufacturing Systems*. Vol. 2, 1990, pages 289-328.

[Shin, 1989] O.K. Shin. *Contribution au pilotage d'îlots flexibles d'assemblage.* Thèse de doctorat, Université de Franche-Comté, Besançon, France, novembre 1989.

[Sibertin-Blanc, 1985] C. Sibertin-Blanc. "High-level Petri nets with data structures". In *Workshop on Applications and Theory of Petri nets*, Finlande, juin 1985.

[Sibertin-Blanc, 1988] C. Sibertin-Blanc. "Le prototypage des applications interactives à l'aide des réseaux de Petri". In *International Workshop Software Engineering and its Applications*, pages 837-856, Toulouse, France, décembre 1988.

[TE/ECLille/MRT, 1992] Travail collectif. *Approche de prototypage de systèmes flexibles de production discrète - Développement d'un noyau de simulation - Expérimentation sur site industriel.* Rapport de fin d'étude d'une recherche financée par le Ministère de la Recherche et de la Technologie (décision d'aide n° 89P0264), Laboratoire d'Automatique et d'Informatique de Lille (LAIL), Telemecanique DRD, France, 1992.

[Telemecanique, 1989] Telemecanique. *Symboles graphiques pour schémas de principe de systèmes d'assemblage.* Document interne, Telemecanique DRD, France, novembre 1989.

[Telemecanique, 1992] Telemecanique. *MONITOR 77/2 Supervisor System – User's Manual.* Documentation Telemecanique, réf. TMX L2 FND BT V41E.

[Telemecanique, 1993] Telemecanique. *Standard De Développement (SDÉ)– Normes internes FI-40 701 à FI-40 799.* Document interne, Telemecanique, France, 1993.

[Tombak et De Meyer, 1988] M. Tombak, A. De Meyer. "Flexibility and FMS : An Empirical Analysis". In *IEEE Transactions on Engineering Management.* Vol. 35, n°2, 1988, pages 101-107.

[Tsai et Fu, 1992] C.J. Tsai, L.C. Fu. "Modular Approach for PN-Modeling od FMS Adaptable to Various Task-Flow Requirement". In *Proceedings of 1992 IEEE. ICRA.* Nice, mai 1992.

[VDI 2860, 1982] *Handhabungsfunktionen, Handhabungseinrichtungen, Begriffe, Definitionen, Symbole.* Verein Deutscher Ingenieure (VDI), Allemagne, octobre 1982.

[Verdebout, 1992] Dr. E. Verdebout. *Pilotage algorithmique du changement de campagne de production dans une installation flexible d'assemblage automatique.* Thèse de doctorat, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Suisse, 1992.

[Verdebout, 1993] Dr. E. Verdebout. *Vers des automatismes de sécurité dans la commande et la simulation des systèmes d'assemblage.* Support de cours, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Suisse, mars 1993.

[Verdin, 1991] L. Verdin. *De la spécification à l'exploitation : le cycle de vie des S.A.P.* Génie automatique et production industrielle. Journées de synthèse, ISMCM, Saint-Ouen, France, 12 et 13 mars 1991.

[Vidril, 1990] M. Vidril. *Une méthode décentralisée pour le pilotage des cellules flexibles d'assemblage*. Rapport de DEA, Université des Sciences et Techniques de Lille Flandres Artois, Lille, France, septembre 1990.

[Vidril, 1995] M. Vidril. *Spécification de réalisation du module de simulation.* Rapport technique Telemecanique, 1995.

[Waldner, 1990] J.B. Waldner. *CIM, les nouvelles perspectives de la production.* Dunod, Paris, 1990.

[Yokoyama, 1988] Dr. Y. Yokoyama. "Symbols for Assembly System". In *Techno Japan*, vol. 21, n°8, pages 8-29, août 1988.

Annexe A

La modélisation par Réseaux de Petri

A.1 RdP simples

Les RdP, tels qu'ils ont été introduits par C.A. Petri [Petri, 1966], se sont avérés être un outil puissant de modélisation des systèmes dits à *événements discrets*. Ces systèmes se caractérisent par une évolution discontinue de leur état. Les changements d'état sont commandés par l'occurrence d'événements qui se produisent à des instants particuliers séparés par des intervalles de temps pouvant être irréguliers. Les variables d'état de tels systèmes évoluent de façon brutale, par pas discret. Ils s'opposent en cela aux systèmes dits *continus* dont l'évolution se fait continûment dans le temps.

Le modèle RdP permet de représenter la *causalité* et le *séquencement* des événements se produisant dans les systèmes discrets. Il met en évidence la *décision* (le choix entre deux événements déclenchables simultanément et en conflit) et l'*indépendance* (parallélisme entre événements non corrélés).

Nous donnons ci-après une description informelle des RdP simples, suffisante pour notre propos. Une définition formelle pourra être trouvée dans [Brams, 1983a].

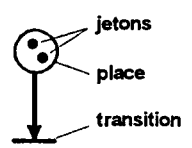


Figure A.1 : éléments d'un RdP.

Un RdP est défini par un ensemble fini de *places* et de *transitions* reliées entre elles par des *arcs orientés* (cf. figure A.1). Le *marquage* des places par

des *jetons* permet de représenter l'état du système modélisé. L'évolution du marquage découle du franchissement des transitions par les jetons (*tir* des transitions). Ce franchissement soustrait des marques dans les places amont et les dépose dans les places aval. Un nouvel état du système est donc obtenu.

La sémantique associée aux différents éléments d'un RdP est la suivante: les places correspondent aux différentes conditions d'évolution du système et les transitions aux événements affectant le système. Les arcs permettent de traduire les relations existant entre conditions et événements. La présence d'un jeton dans une place rend la condition associée vraie. Lorsque toutes les conditions liées aux places amont à une transition sont vérifiées, l'événement correspondant à la transition peut avoir lieu.

Pendant le fonctionnement du réseau il arrive parfois qu'un choix doive être fait entre plusieurs jetons ou entre plusieurs transitions. Ces *conflits* traduisent les degrés de liberté liés à la flexibilité du système. La modélisation par RdP, en conservant cette flexibilité, introduit donc des *indéterminismes* dans l'évolution des jetons. Deux types de conflits peuvent apparaître :

- le **conflit transition** ou **conflit de routage** se présente lorsque plusieurs transitions sont simultanément franchissables alors qu'elles engagent un jeton unique. Un mécanisme extérieur doit alors déterminer la transition à franchir.

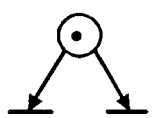


Figure A.2 : conflit transition.

- le **conflit jeton** se produit lorsque plusieurs jetons distincts valident simultanément une transition. Dans le cas des RdP simples, où les jetons ne sont pas individualisés, ce type de conflit disparaît de lui-même : il suffit de prendre le premier jeton "se présentant". Par contre, dans le cas des RdP de haut niveau (cf. § A.2), un mécanisme extérieur doit déterminer le jeton à tirer.

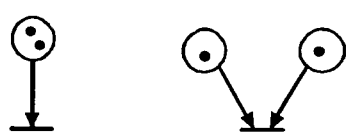


Figure A.3 : conflit jeton.

Nous illustrons ci-après le principe de la modélisation par RdP sur un exemple : la modélisation d'une portion de convoyeur à palettes.

Une portion de convoyeur, comme tout support d'éléments mobiles, est caractérisée par l'état des ES qu'il possède (cf. § 5.2.2). A chacun de ces ES peut être associé un RdP modélisant son comportement, i.e. la succession des divers états que l'ES peut occuper. En ne considérant que deux états possibles pour un ES, libre ou occupé, on obtient le graphe de la figure A.4. L'ES

correspondant à ce graphe est dans l'état occupé, comme l'indique la présence d'un jeton dans la place "occupé".

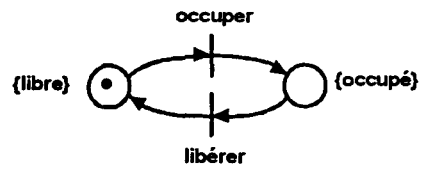
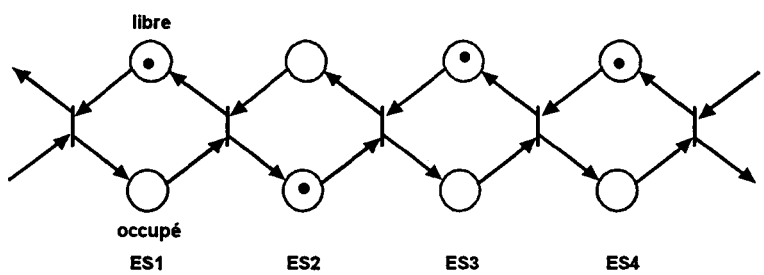
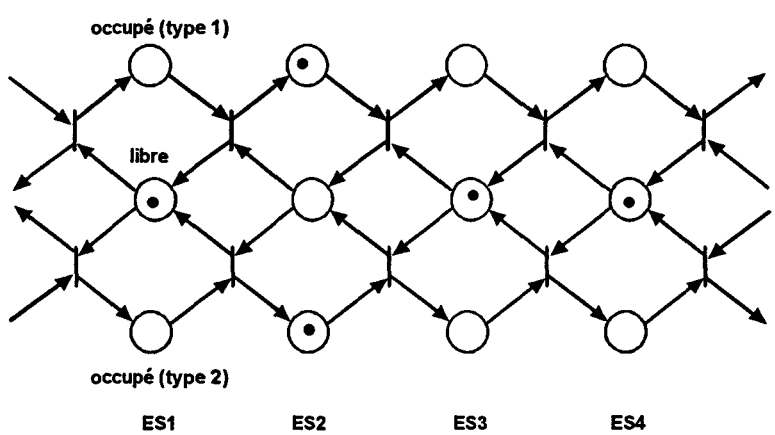


Figure A.4 : état d'un ES.

Le modèle correspondant à une portion de convoyeur est obtenu en combinant les graphes de ses ES. Les graphes de deux ES pouvant échanger des palettes sont reliés en fusionnant les transitions "libérer" et "occuper" correspondantes. La figure A.5.a montre le modèle RdP d'une portion de convoyeur à quatre places. Les emplacements ES1, ES3 et ES4 sont libres.



a- 4 emplacements, 1 type de palettes.



b- 4 emplacements, 2 types de palette.

Figure A.5 : modélisation d'un convoyeur par RdP simples.

Un tel modèle ne permet cependant pas de connaître le type de l'objet occupant l'emplacement ES2 (sauf de manière implicite dans le cas où le convoyeur ne peut être occupé que par un seul type de palette). Pour coder cette information dans les RdP classiques, il est nécessaire de multiplier le nombre de places. Ainsi le RdP de la figure A.5.b modélise un convoyeur à quatre emplacements pouvant être occupés par deux types de palettes.

L'emplacement ES2 est occupé par une palette de type 1. Il est aisé de vérifier que l'ordre des palettes introduites est respecté en sortie du convoyeur.

La méthode peut être généralisée à n'importe quel convoyeur. L'exemple choisi est cependant volontairement simpliste. Dans un contexte réel où le nombre de types d'éléments mobiles et la capacité des convoyeurs sont beaucoup plus importants, les RdP classiques conduisent à des modèles dont la taille (nombre de places et de transitions) interdit toute utilisation pratique. Il faut donc trouver un moyen pour résumer une partie de l'information codée dans la structure du réseau.

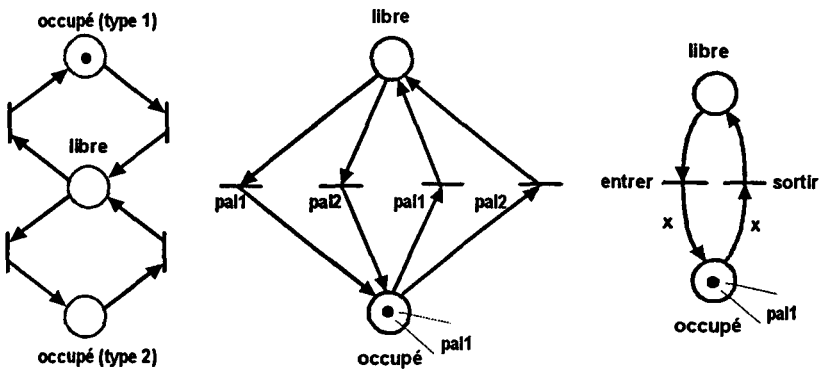
A.2 Pliage et RdP de haut niveau

L'idée du pliage tire son origine des diverses symétries existant dans la structure du réseau simple. Il s'agit de confondre entre elles les places ou transitions en symétrie. La taille du réseau est ainsi diminuée. Il y a bien entendu perte d'information qu'il s'agit de reporter à un autre endroit : les jetons. Les diverses modalités d'application de ce principe ont donné les diverses classes de RdP de haut niveau : RdP à prédicats (RdPP), RdP colorés (RdPC), RdP à objets (RdPO).

Nous n'entrerons pas ici dans les détails de chaque formalisme. Nous préférons illustrer le principe du pliage de façon intuitive à l'aide de l'exemple du convoyeur présenté au paragraphe précédent. Nous utiliserons, dans ce paragraphe, le formalisme des RdPP [Brams, 1983a]. Les RdPO, que nous avons utilisés pour la mise en oeuvre du modèle de validation (cf. chapitre 9), feront l'objet d'une présentation au paragraphe A.3.

Les symétries du modèle par RdP simples (cf. figure A.5) sont évidentes. Ainsi les places "occupé" correspondant aux deux types de palettes ont des rôles symétriques. En associant l'information type de palette aux jetons situés dans les places "occupé", il est possible de confondre ces places. Il est alors nécessaire de préciser au niveau de chaque transition la valeur du paramètre des jetons impliqués dans son franchissement. Cette information est portée par des étiquettes sur les arcs dans les RdPP. Ce premier pliage, appliqué au réseau de la figure A.6.a associé à un ES pouvant recevoir deux types de palette, aboutit au réseau de la figure A.6.b.

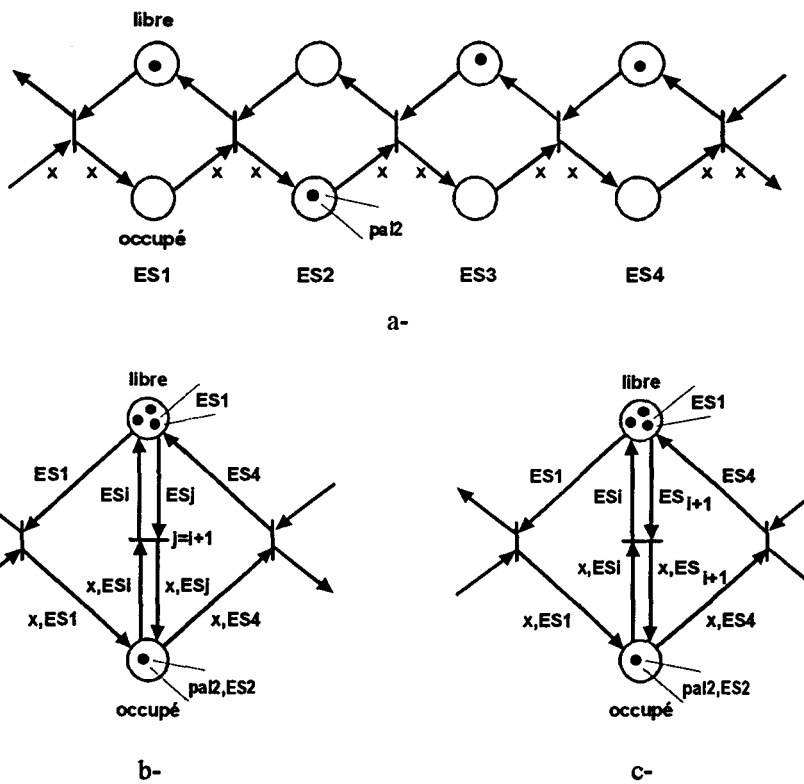
Cette première symétrie a permis de diminuer le nombre de places mais le nombre de transitions reste inchangé. Remarquons cependant qu'il existe une symétrie au niveau des transitions "libérer" et "sortir" associées à chaque type de palette. Il est possible de remplacer ces transitions par une seule transition dont les arcs seront étiquetés non plus par des constantes, mais par des variables à valeur dans {pal1, pal2}. La figure A.6.c illustre le résultat obtenu à l'issue de ce second pliage.



a- b- c-

Figure A.6 : pliage de places et de transitions.

La modélisation de notre convoyeur après ces deux premiers pliages est représentée sur la figure A.7.a. Il est à noter que la même démarche appliquée à un nombre différent de types de palettes aboutit au même réseau. Seul le domaine de valeurs des étiquettes est modifié.



b- c-

Figure A.7 : application du pliage sur l'exemple du convoyeur.

La démarche peut être poursuivie en remarquant que les ES constituant le convoyeur ont des rôles symétriques. Ici, cependant, il est nécessaire d'associer à la transition résultante un prédicat indiquant les relations entre paramètres des jetons permettant de franchir la transition. Le résultat pour notre exemple est donné sur la figure A.7.b. Les jetons portent une information supplémentaire, à

savoir l'identifiant de l'ES auquel ils sont associés. Par souci de clarté, nous remplacerons l'étiquette ESj par l'étiquette ES(i+1) de façon à expliciter le prédicat sur les arcs (cf. figure A.7.c). Plus généralement, chaque fois que le prédicat sera du type y = f(x), nous utiliserons f(x) à la place de l'étiquette y et du prédicat y = f(x).

Le principe du pliage peut donc être appliqué autant de fois que nécessaire. Il suffit, lorsqu'il y a perte d'information, d'ajouter un paramètre au niveau des jetons pour obtenir un réseau équivalent au réseau initial. Lorsque l'approche et le formalisme adopté sont compris, il n'est plus nécessaire de construire au préalable le RdP ordinaire du système à modéliser. Quelques règles simples permettent d'aboutir directement au modèle par RdPP.

Le problème consiste désormais à utiliser le pliage à bon escient. Il s'agit, en effet, de déterminer jusqu'où plier afin d'obtenir un modèle alliant lisibilité et concision. Poussée à l'extrême la démarche conduit en effet à modéliser l'ensemble des ES de l'atelier par le réseau de la figure A.8.

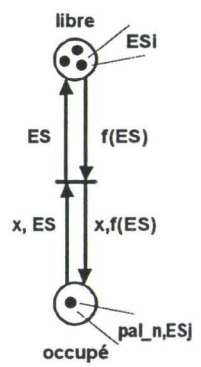


Figure A.8 : pliage complet.

La fonction f donne pour chaque ES les identifiants des ES immédiatement accessibles. La topologie entière de l'atelier est donc codée dans la fonction f. Des conflits se produisent si f n'est pas une application. Plusieurs successeurs existent alors pour certains ES (cas des ES appartenant aux aiguillages divergents).

Entre ces deux approches extrêmes (toute l'information codée dans la structure réseau ou toute l'information codée dans les fonctions), une vaste panoplie de possibilités existent. Citons, par exemple, l'approche de modélisation développée à Grenoble en utilisant les RdPC [Descotes-Genon et Ladet, 1987] [Martin, 1987]. Le modèle obtenu consiste en un réseau "universel" et un ensemble de fonctions agissant sur les couleurs des jetons (analogues à la fonction f de notre exemple) et particulières à chaque atelier. L'ensemble des informations variables est donc reporté au niveau des fonctions (ou prédicats dans le cas des RdPP).

A titre indicatif, nous présentons ci-après les choix de pliage que nous avons adoptés dans la démarche de modélisation présentée au chapitre 9.

Sont repliés :

- les graphes associés aux ES d'un même type appartenant à un même lieu de stockage : le comportement de ces ES sera donc codé par un seul graphe (exemple tiré du site DRD : les ES d'un **même** plateau).
- les graphes associés aux lieux de stockage mobiles de type identique : le comportement de ces lieux de stockage sera codé par un seul graphe (exemple tiré du site DRD : les ES de deux plateaux **différents** apparaissent sur un même graphe)
- les graphes associés à chaque produit d'une même famille : le comportement des produits d'une même famille (gammes) sera codé par un seul graphe.

Ne sont pas repliés :

- les graphes associés aux lieux de stockage fixes de même type : le comportement de ces lieux fixes sera donc codé par plusieurs graphes identiques (exemple tiré du site DRD : on associe un graphe par indexeur).
- les graphes associés aux ressources actives : le comportement des ressources sera codé par plusieurs graphes (exemple tiré du site DRD : les comportements des stations AFMA et IBM font l'objet de deux graphes différents).

A.3 Les RdP à Objets

Les RdPO [Sibertin-Blanc, 1985] se prêtent bien à une mise en oeuvre logicielle structurée par les objets. Pour cette raison, ce sont eux que nous avons choisis pour la mise en oeuvre du modèle de validation dans le cadre du simulateur présenté en annexe D.

A.3.1 Eléments du réseau

Dans les RdPO, les jetons sont formés à partir d'*objets*. Comme dans l'approche objets développée en informatique, un objet est défini comme étant une *instance* d'une *classe* d'objets. La classe d'un objet définit sa structure, à savoir l'ensemble de ses *attributs* (ou *propriétés*) et l'ensemble de ses *opérations* (ou *méthodes*) permettant de modifier les valeurs de ces attributs. Le *type* d'un objet est donné par sa classe.

Un jeton d'un RdPO est un n-uplet d'objets. Ainsi les objets se trouvant dans les places du réseau ne sont pas isolés et peuvent être liés dynamiquement les uns aux autres sur un même jeton. A la différence des jetons élémentaires circulant dans les RdP classiques, les jetons-objets sont identifiables, i.e. on peut les distinguer les uns des autres. Chaque jeton a un type défini par la donnée de son arité (longueur du n-uplet) et par la classe de chacune de ses composantes.

Un jeton d'arité 0 ne porte pas d'objet (jeton banalisé) et est équivalent à un jeton classique.

Les places possèdent un type qui détermine leur domaine de valeur. Une place ne pourra être occupée que par des jetons de même type qu'elle.

Les transitions sont constituées d'un *filtre* (ou pré-condition) et d'une *action*. Les conditions sont des expressions booléennes portant sur les attributs des objets et permettent de sélectionner les jetons situés dans les places amont. Les actions déclenchent les méthodes des objets qui franchissent la transition, faisant ainsi évoluer leurs attributs. Elles peuvent également commander la création de nouveaux objets.

Les arcs portent un certain nombre de n-uplets de variables typés. Une variable sera toujours instanciée par un objet du même type qu'elle. La donnée du type de chacune des variables et de l'arité du n-uplet définit le type du n-uplet. Dans un RdPO correct, ce type doit correspondre à celui de la place associée à l'arc. Une variable est définie par rapport à la transition liée à l'arc et intervient dans l'expression des conditions et actions de cette transition. Un même nom peut ainsi désigner deux variables différentes si elles sont associées à deux transitions distinctes.

A.3.2 Fonctionnement

C'est le franchissement ou tir des transitions qui permet de faire évoluer le marquage du réseau, et donc de modifier son état.

Une transition est dite *validée* si toutes ses places amont sont marquées. Une transition validée est dite *franchissable* si il existe une substitution associant un objet à chaque variable amont telle que la condition soit satisfaite.

Le franchissement d'une transition franchissable s'effectue en trois étapes :

- retirer des places d'entrée les objets capturés par les variables amont (variables associées aux arcs d'entrée de la transition),
- exécuter l'action associée à la transition,
- déposer dans les places de sortie les objets capturés par les variables aval.

On voit d'après ce mécanisme que le franchissement d'une transition peut consommer des objets s'il y a des variables d'entrée que l'on ne retrouve pas en sortie, et tout aussi bien en produire de nouveaux, s'il y a des variables de sortie qui ne figurent pas en entrée de la transition.

A.3.4 Exemple

Soit le réseau de Petri à Objets de la figure A.9. Pour cet exemple :

- l'ensemble des classes d'objets est {PIECE, MACHINE}. Les attributs de la classe MACHINE sont donnés sur le schéma ;

- l'ensemble des variables sur la transition t1 est {p, m} avec p de la classe PIECE et m de la classe MACHINE ;
- le type de la place P2 en aval de t1 est <PIECE, MACHINE>, de même que les n-uplets de variables attachés aux arcs ayant P2 pour extrémité ;
- le marquage initial est constitué d'un jeton-pièce (<pc1>) dans la place P1 et de deux jetons-machine (<M1> et <M2>) dans la place P4 ;
- la transition t1 est validée; elle est franchissable si l'opération à effectuer sur pc1 peut être réalisée par l'une des deux machines (condition portée par la transition) ;
- le franchissement de t1 par <pc1> et <m2> produira le marquage suivant: <M1> dans P4 et <pc1, M2> dans P2 ;
- t2 est franchissable pour le nouveau marquage puisque aucune condition ne lui est associée ;
- le franchissement de t2 par <pc1, M2> déclenchera la méthode gamme associée à l'objet pc1 permettant de faire évoluer l'attribut op de l'objet (prochaine opération à réaliser).

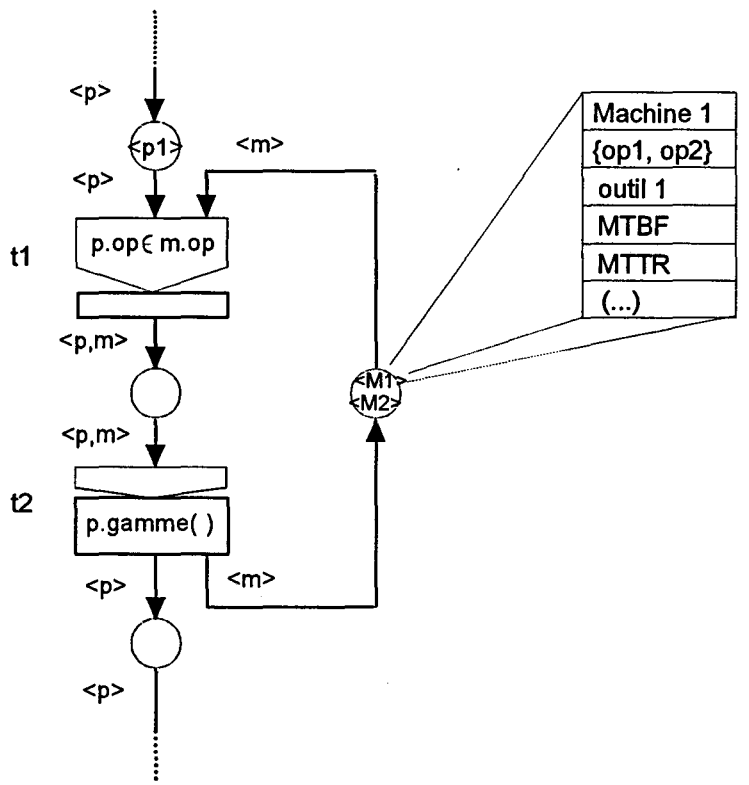


Figure A.9: réseau de Petri à Objets.

Annexe B

Exemple de calcul des fonctions d'évaluation

Le calcul des fonctions d'évaluation s'effectue, pour un produit et un état de fonctionnement des moyens donnés, en trois étapes :

- numérotation des stades et des positions ;
- établissement des matrices initiales IJJ, IJI et A ;
- exécution de l'algorithme de génération des matrices OPTA et OPTS.

La démarche est illustrée dans cette annexe sur l'exemple de la gamme principale du produit E du site DRD, l'ensemble des moyens étant en état de marche.

B.1 Numérotations

B.1.1 Numérotation des stades

Rappel : dans le formalisme utilisé, les stades correspondent aux arcs.

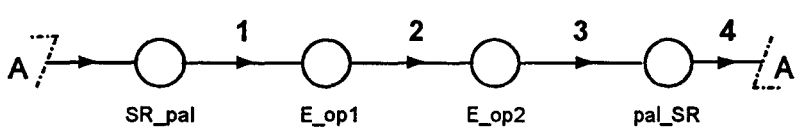


Figure B.1 : numérotation des stades pour la gamme principale du produit E.

B.1.2 Numérotation des positions

La numérotation des positions dépend, elle aussi, à priori du produit : les positions actives varient en fonction des opérations à réaliser sur le produit et une position passive pour l'un des produits peut être active pour un autre produit.

Néanmoins, dans la plupart des cas, les positions passives sont les mêmes pour chaque produit. Il est alors possible de procéder de la manière suivante pour la numérotation des positions : on numérote de 1 à n les positions passives, puis de n+1 n+N les positions actives, avec n le nombre de positions passives et N le nombre total de positions actives différentes pour **tous** les produits. La numérotation des positions est alors commune à l'ensemble des produits.

La numérotation adoptée dans notre exemple est indiquée sur la figure B.2 (voir ci-contre) :

- 1 à 28 : positions passives ;
- 29 à 33 : positions actives.

B.2 Matrices initiales

B.2.1 Matrice IJJ

i/j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	{2}	{3}	{4}	{5,20}	{6}	{7}	{8}	{9,23}	{10}	{11}	{12}	{17,26}	{14}	{15}	{16}	{17,1}
2	{2}	{3}	{4}	{5,20}	{6}	{7}	{8}	{9,23}	{10}	{11}	{12}	{17,26}	{14}	{15}	{16}	{17,1}
3	{2}	{3}	{4}	{5,20}	{6}	{7}	{8}	{9,23}	{10}	{11}	{12}	{17,26}	{14}	{15}	{16}	{17,1}
4	{2}	{3}	{4}	{5,20}	{6}	{7}	{8}	{9,23}	{10}	{11}	{12}	{17,26}	{14}	{15}	{16}	{17,1}

B.2.2 Matrice IJI

i/j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4

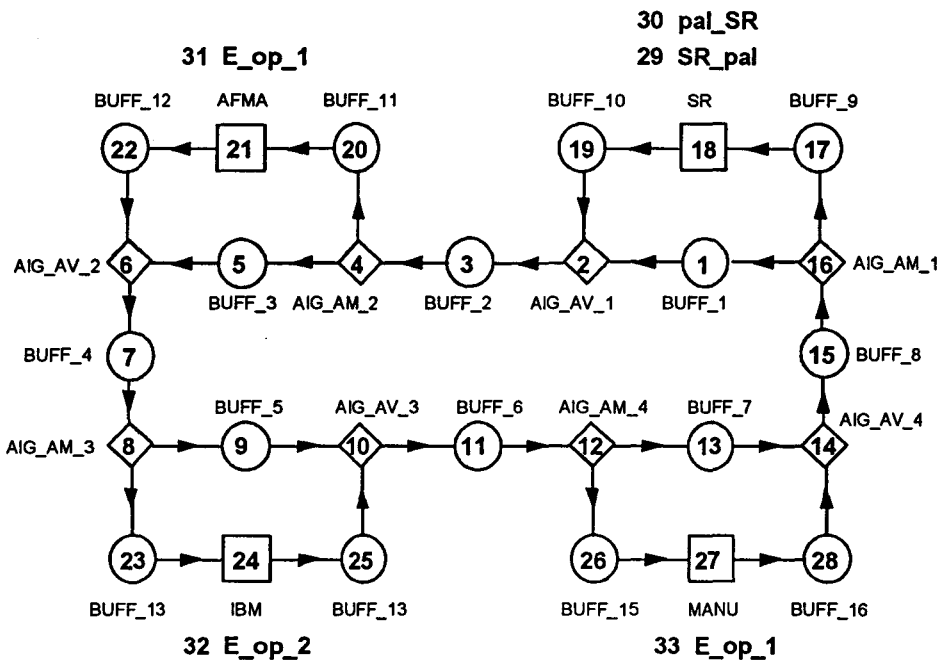


Figure B.2 : numérotation des positions pour la gamme principale du produit E.

17	18	19	20	21	22	23	24	25	26	27	28	28	30	31	32	33
{18}	{19}	{2}	{21}	{22,31}	{6}	{24}	{25}	{10}	{27}	{28,33}	{14}	{}	{}	{21}	{}	{27}
{18}	{19}	{2}	{21}	{22}	{6}	{24}	{25,32}	{10}	{27}	{28}	{14}	{}	{}	{}	{24}	{}
{18}	{19,30}	{2}	{21}	{22}	{6}	{24}	{25}	{10}	{27}	{28}	{14}	{}	{18}	{}	{}	{}
{18}	{19,29}	{2}	{21}	{22}	{6}	{24}	{25}	{10}	{27}	{28}	{14}	{18}	{}	{}	{}	{}

17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
1	1	1	1	1	1	1	1	1	1	1	1	-	-	2	-	2
2	2	2	2	2	2	2	2	2	2	2	2	-	-	-	3	-
3	3	3	3	3	3	3	3	3	3	3	3	-	4	-	-	-
4	4	4	4	4	4	4	4	4	4	4	4	1	-	-	-	-

B.3.3 Matrice A (temps en 1/10 de seconde)

i / j'	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	5	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	∞	0	20	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
3	∞	∞	0	5	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
4	∞	∞	∞	0	30	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
5	∞	∞	∞	∞	0	5	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
6	∞	∞	∞	∞	∞	0	75	∞	∞	∞	∞	∞	∞	∞	∞	∞
7	∞	∞	∞	∞	∞	∞	0	5	∞	∞	∞	∞	∞	∞	∞	∞
8	∞	∞	∞	∞	∞	∞	∞	0	30	∞	∞	∞	∞	∞	∞	∞
9	∞	∞	∞	∞	∞	∞	∞	∞	0	5	∞	∞	∞	∞	∞	∞
10	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	20	∞	∞	∞	∞	∞
11	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	5	∞	∞	∞	∞
12	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	30	∞	∞	∞
13	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	5	∞	∞
14	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	75	∞
15	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	5
16	30	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0
17	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
18	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
19	∞	5	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
20	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
21	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
22	∞	∞	∞	∞	∞	5	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
23	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
24	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
25	∞	∞	∞	∞	∞	∞	∞	∞	∞	5	∞	∞	∞	∞	∞	∞
26	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
27	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
28	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	5	∞	∞
29	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
30	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
31	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
32	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
33	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞

B.3 Calcul de OPTA et OPTS

B.3.1 Matrice OPTA

i / j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1305	1300	1280	1275	1565	1560	1485	1480	1450	1445	1425	1420	1420	1415	1340	1335
2	1170	1165	1145	1140	1110	1105	1030	1025	1345	1340	1320	1315	1285	1280	1205	1200
3	525	520	500	495	465	460	385	380	350	345	325	320	290	285	210	205
4	365	360	340	335	305	300	225	220	190	185	165	160	130	125	50	45

B.3.2 Matrice OPTS

i / j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	2	3	4	20	6	7	8	9	10	11	12	26	14	15	16	1
2	2	3	4	5	6	7	8	23	10	11	12	13	14	15	16	1
3	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
4	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	35	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	35	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	35	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
35	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
0	10	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	0	30	∞	∞	∞	∞	∞	∞	∞	∞	∞	160	160	∞	∞	∞
∞	∞	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	0	10	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	0	30	∞	∞	∞	∞	∞	∞	∞	∞	90	∞	∞
∞	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	0	10	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	0	30	∞	∞	∞	∞	∞	∞	600	∞
∞	∞	∞	∞	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	0	10	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	30	∞	∞	∞	∞	60
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	∞	∞	∞	∞	∞
∞	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	∞	∞	∞	∞
∞	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	∞	∞	∞
∞	∞	∞	∞	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	∞	∞
∞	∞	∞	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	∞	∞	0	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	0

17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
1345	1335	1305	1240	1230	1565	1490	1480	1450	1385	1375	1420	1335	∞	∞	∞	∞
1210	1200	1170	1150	1140	1110	990	980	1345	1325	1315	1285	∞	∞	1140	∞	1315
170	160	525	505	495	465	390	380	350	330	320	290	∞	∞	∞	380	∞
10	0	365	345	335	305	230	220	190	170	160	130	∞	0	∞	∞	∞

17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
18	19	2	21	31	6	24	25	10	27	33	14	18	-	-	-	-
18	19	2	21	22	6	24	32	10	27	28	14	-	-	21	-	27
18	30	2	21	22	6	24	25	10	27	28	14	-	-	-	24	-
18	18	2	21	22	6	24	25	10	27	28	14	-	18	-	-	-

Nous donnons ci-après le détail du calcul pour OPTA(1,5) et OPTA(1,20) utilisés dans l'exemple du chapitre 14 (cf. § 14.5.4) :

OPTA(1,5) = A(5, 6) + A(6, 7) + A(7, 8) + A(8, 9) + A(9, 10) + A(10, 11)
+ A(11, 12) + A(12, 26) + A(26, 27) + A(27, 33) + A(33, 27)
+ A(27, 28) + A(28, 14) + A(14, 15) + A(15, 16) + A(16, 1)
+ A(1, 2) + A(2, 3) + A(3, 4) + A(4, 5) + A(5, 6) + A(6, 7)
+ A(7, 8) + A(8, 23) + A(23, 24) + A(24, 32) + A(32, 24)
+ A(24, 25) + A(25, 10) + A(10, 11) + A(11, 12) + A(12, 13)
+ A(13, 14) + A(14, 15) + A(15, 16) + A(16, 17) + A(17, 18)
+ A(18, 30) + A(30, 18) = **1565**

OPTA(1,20) = A(20, 21) + A(21, 31) + A(31, 21) + A(21, 22) + A(22, 6)
+ A(6, 7) + A(7, 8) + A(8, 23) + A(23, 24) + A(24, 32)
+ A(32, 24) + A(24, 25) + A(25, 10) + A(10, 11) + A(11, 12)
+ A(12, 13)+ A(13, 14) + A(14, 15) + A(15, 16) + A(16, 17)
+ A(17, 18) + A(18, 30) + A(30, 18) = **1240**

Annexe C

Exemples d'utilisation des graphes d'influence

C.1 Exemple 1 : boucle de retouche avant

C.1.1 Gammes

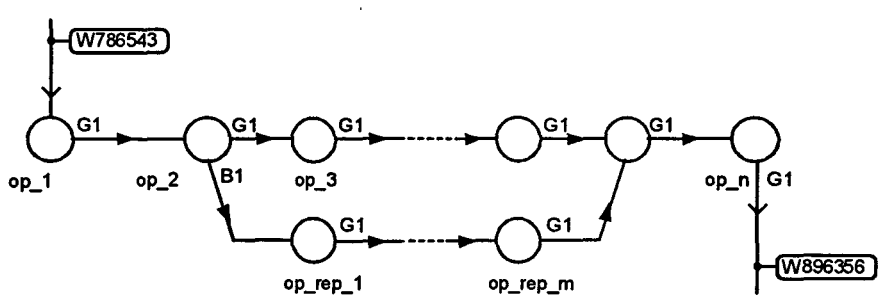


Figure C.1 : exemple de processus opératoire présentant une gamme de retouche "avant" (exemple 1).

Dans cet exemple, la gamme principale correspond aux opérations "op_i". Une gamme de retouche "avant", formée des opérations "op_rep_i", est également spécifiée en cas de défaut au niveau de "op_2".

C.1.2 Choix de l'opération cible et définition des zones

Dans cet exemple, l'opération cible choisie est l'action op_1. Définir le PAC au niveau de la première opération permet de détruire les bons de manière progressive, au fur et à mesure que la production se termine. Cependant, en cas

de rebut sur l'une des opérations aval, le nombre d'articles terminés ne pourra pas être garanti. La situation est inversée si l'on place le PAC sur l'opération terminale de la gamme (cf. exemple 2).

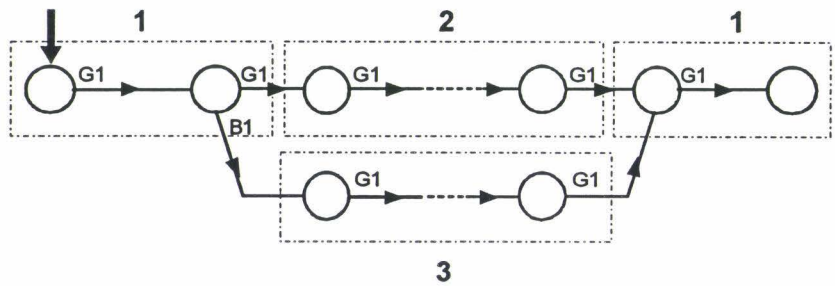


Figure C.2 : définition des zones (exemple 1).

C.1.3 Construction des arcs TT dans les zones

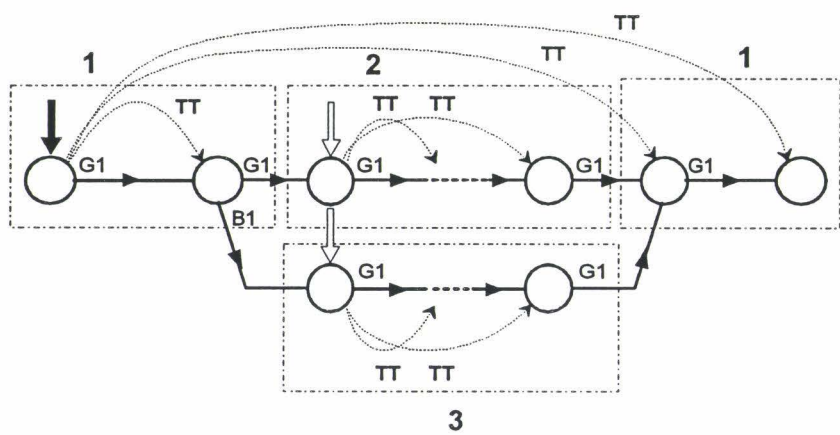


Figure C.3 : construction des arcs TT dans les zones (exemple 1).

C.1.4 Définition des flux de zone

Zone	1	2	3
Flux	N1	g1	b1
Flux Max	N1	N1	N1
Cible	N1	0	0

L'équation de conservation des flux s'écrit :

$$N1 = b1 + g1$$

C.1.5 Ajout des influences inter-zones

Quatre arcs d'influence inter-zones sont nécessaires afin d'équilibrer les flux circulant dans chaque zone.

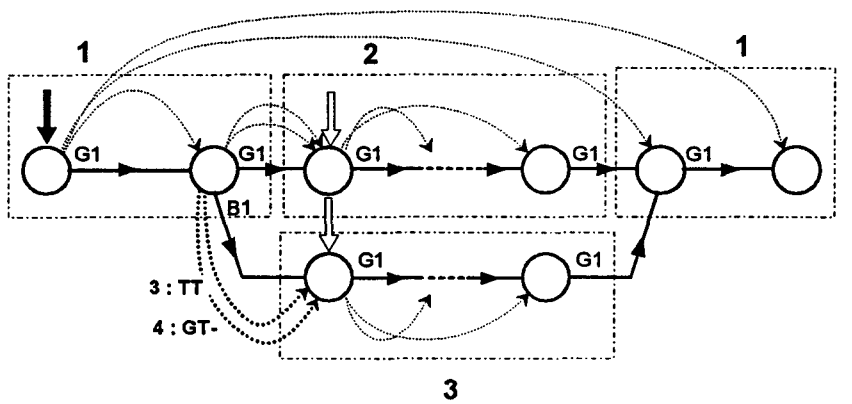


Figure C.4 : ajout des arcs inter-zones (exemple 1).

Zone	1	2	3
Flux	**N1**	**g1**	**b1**
Flux Max	N1	N1	N1
Cible	N1	0	0
+ 1 : TT	N1	N1	0
+ 2 : BT-	N1	N1 - b1 = g1	0
+ 3 : TT	N1	g1	N1
+ 4 : GT-	**N1**	**g1**	**N1 - g1 = b1**

La valeur cible (dernière ligne) de chaque zone étant égale à la valeur du flux devant circuler dans la zone (ligne 2) le graphe d'influence est complet.

C.2 Exemple 2 : boucle de retouche arrière

C.2.1 Gammes

Dans cet exemple, la gamme principale correspond aux opérations "op_i". Une gamme de retouche "arrière", formée des opérations "op_rep_i", est également spécifiée en cas de défaut sur l'avant-dernière opération : ces opérations permettent alors de replacer le produit au stade précédant op_2.

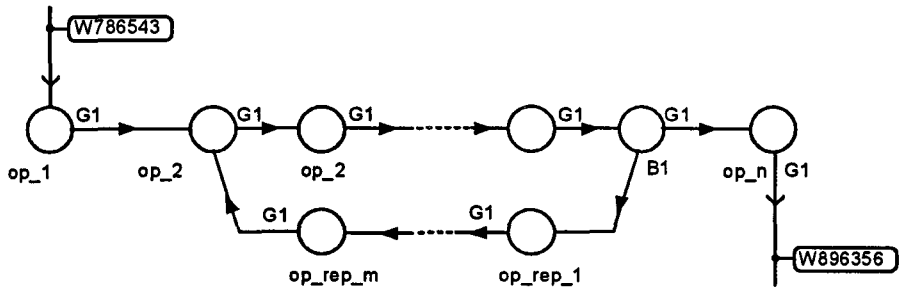


Figure C.5 : exemple de processus opératoire présentant une gamme de retouche "arrière" (exemple 2).

C.2.2 Choix de l'opération cible et définition des zones

Dans cet exemple, nous avons placé le PAC sur la dernière opération. Cela permet de garantir le nombre d'articles à l'arrivée. En contrepartie, la procédure de désinstanciation ne peut pas se déclencher tant que cette opération n'est pas terminée : les bons de fabrication terminés correspondant aux opérations amont restent disponibles dans les automates pour une éventuelle compensation des pertes.

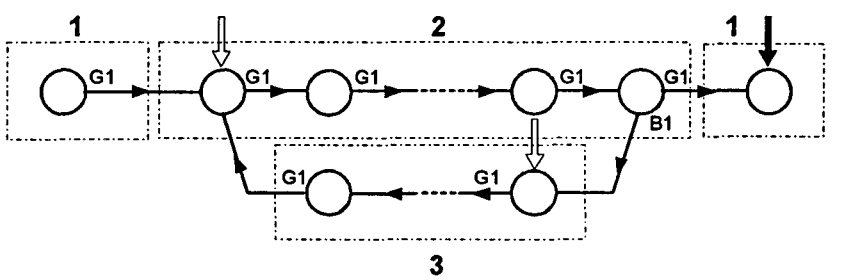


Figure C.6 : définition des zones (exemple 2).

C.2.3 Construction des arcs TT dans les zones

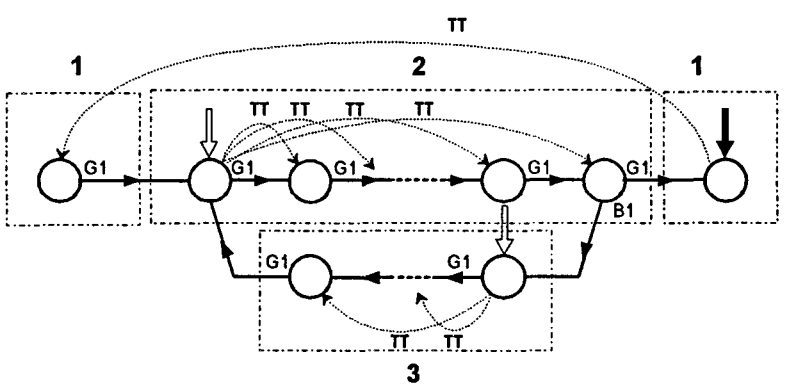


Figure C.7 : construction des arcs TT dans les zones (exemple 2).

C.2.4 Définition des flux de zone

Zone	1	2	3
Flux	N1	N1 + b1	b1
Flux Max	N1	Infini	Infini
Cible	N1	0	0

L'équation de conservation des flux s'écrit :

$N1 = g1$

C.2.5 Ajout des influences inter-zones

Trois arcs d'influence inter-zones sont nécessaires afin de relier les flux circulant dans chaque zone.

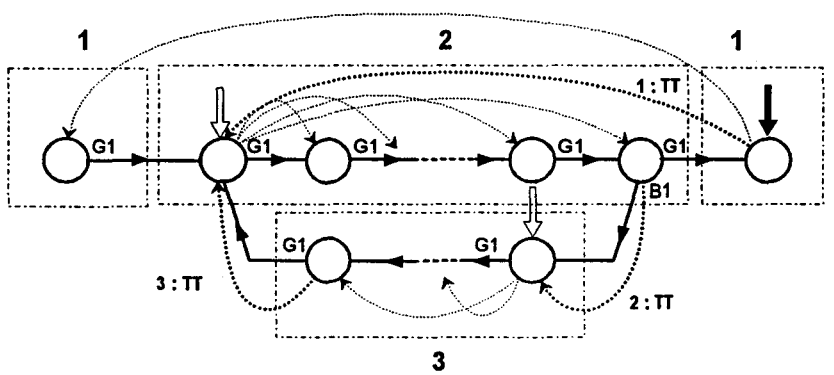


Figure C.8 : arcs inter-zones (exemple 2).

Zone	1	2	3
Flux	N1	N1 + b1	b1
Flux Max	N1	Infini	Infini
Cible	N1	0	0
1 : TT	N1	N1	0
2 : TT	N1	N1	N1
3 : TT	**N1**	**Infini**	**Infini**

Dans cet exemple, il est inutile d'essayer de rajouter des arcs de type GT- ou BT-, puisque les valeurs cibles sont infinies ! La vérification de la cohérence du graphe d'influence est moins immédiate dans ce cas.

C.3 Exemple 3 : séquence linéaire d'opérations

C.3.1 Gammes

Dans cet exemple, une seule gamme (correspondant à une séquence linéaire d'opérations) est spécifiée. Les quatre premières opérations sont susceptibles de produire des rebuts. Les pièces rebutées, n'étant pas identifiées en tant qu'article, ne font l'objet d'aucune gestion particulière.

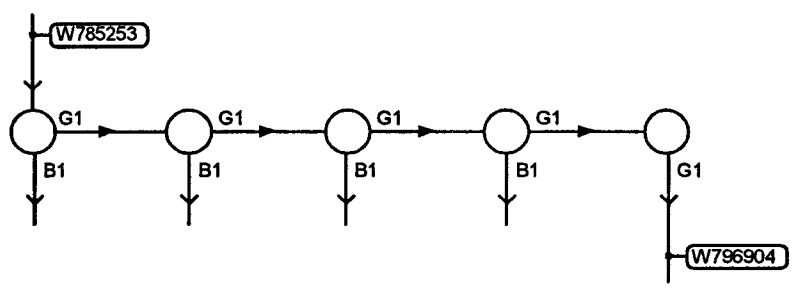


Figure C.9 : exemple de processus opératoire linéaire (exemple 3).

C.3.2 Choix de l'opération cible et définition des zones

Cet exemple permet de comprendre les deux types de contre-réaction aux pertes par rapport à la position dans la gamme de l'opération cible principale. Nous avons placé à dessein le PAC au milieu de la gamme, définissant ainsi deux zones : la zone amont au PAC où toute perte doit être compensée par réintroduction de composants aux entrées (contre réaction positive), et la zone aval au PAC où les pertes ne sont pas compensées (contre-réaction négative).

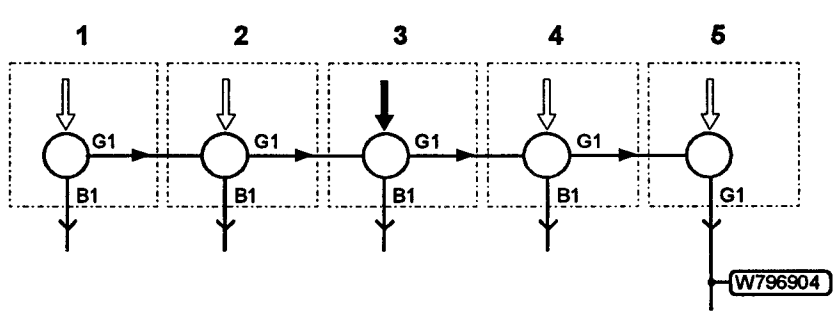


Figure C.10 : définition des zones (exemple 3).

C.3.3 Construction des arcs TT dans les zones

Dans cet exemple, toutes les opérations sont dans des zones différentes : elles correspondent donc toutes à des opérations cibles secondaires.

Les contraintes de précédence du graphe d'influence se réduisent aux seules contraintes inter zones.

C.3.4 Définition des flux de zone

Zone	1	2	3	4	5
Flux	N1	g1	g2	g3	g4
Flux Max	Infini	Infini	g2	g3	g4
Cible	0	0	g2	0	0

Les équations de conservation des flux s'écrivent :

$N1 = g1 + b1$

$g1 = g2 + b2$

$g2 = g3 + b3$

$g3 = g4 + b4$

g2 correspond à la quantité cible spécifiée dans les ordres de fabrication de l'article en question.

C.3.5 Ajout des influences inter-zones

Le flux maximum dans les zones 1 et 2 est infini. Cependant, il est impossible d'appliquer la méthode des cibles infinies dans cet exemple (cf. § 12.3), car l'opération correspondant à la zone 1 est une opération d'entrée de gamme. Le système lancerait obligatoirement trop de produits sur la ligne.

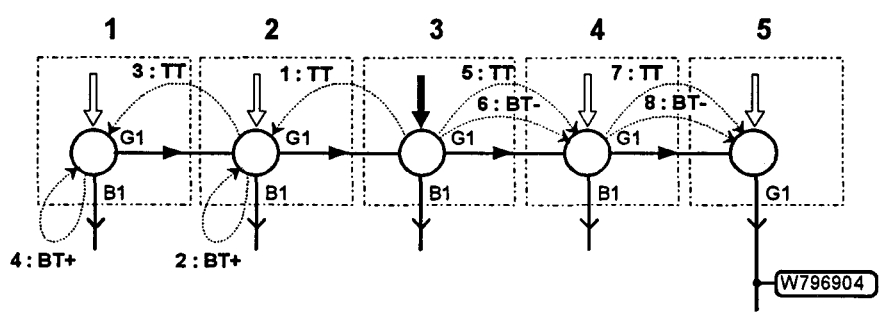


Figure C.11 : arcs inter-zones (exemple 3).

Zone	1	2	3	4	5
Flux	N1	g1	g2	g3	g4
Flux Max	Infini	Infini	g2	g3	g4
Cible	0	0	g2	0	0
1 : TT	0	g2	g2	0	0
2 : BT+	0	g1=g2+b2	g2	0	0
3 : TT	g1	g1	g2	0	0
4 : BT+	N1=g1+b1	g1	g2	0	0
5 : TT	N1	g1	g2	g2	0
6 : BT-	N1	g1	g2	g3=g2-b3	0
7 : TT	N1	g1	g2	g3	g3
8 : BT-	N1	g1	g2	g3	g4=g3-b4

La valeur cible (dernière ligne) de chaque zone étant égale à la valeur du flux devant circuler dans la zone (ligne 2) le graphe d'influence est complet.

C.4 Exemple 4 : synchronisation de gammes

C.4.1 Gammes

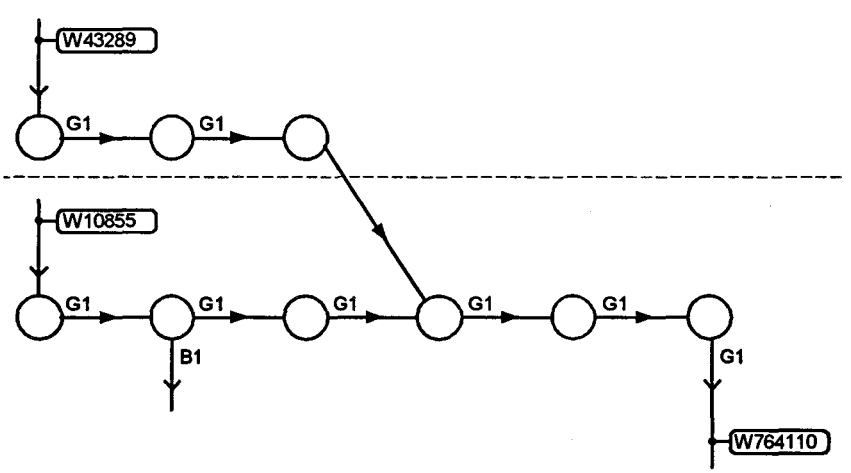


Figure C.12 : exemple de gammes nécessitant une synchronisation (exemple 4).

Cet exemple correspond à un cas de synthèse de gammes relatives à des articles différents (cf. § 8.4). La synchronisation du rendez-vous au niveau de

l'opération d'assemblage est donc supposée hors du domaine de l'ordonnancement.

C.4.2 Choix de l'opération cible et définition des zones

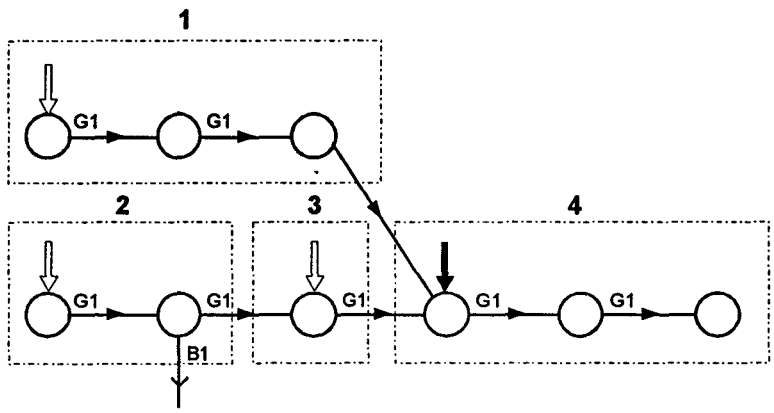


Figure C.13 : définition des zones (exemple 4).

C.4.3 Construction des arcs TT dans les zones

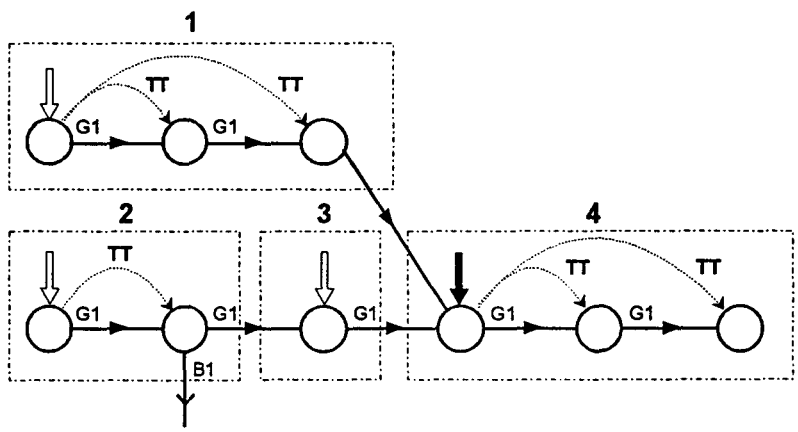


Figure C.14 : construction des arcs TT dans les zones (exemple 4).

C.4.4 Définition des flux de zone

Zone	1	2	3	4
Flux	N1	N2	g1	N1
Flux Max	N1	Infini	N1	N1
Cible	0	0	0	N1

Les équations de conservation des flux s'écrivent :

N1 = g1

N2 = g1 + b1

C.4.5 Ajout des influences inter-zones

Nous présentons deux solutions pour gérer cet exemple : pas de synchronisation si la ligne de fabrication du composant secondaire est balancée avec la ligne principale, ou synchronisation par l'intermédiaire d'un arc GT+. L'ajout de cet arc permet de conditionner le lancement en production du composant à la réussite d'une opération sur le récepteur.

Solution 1 : pas de synchronisation

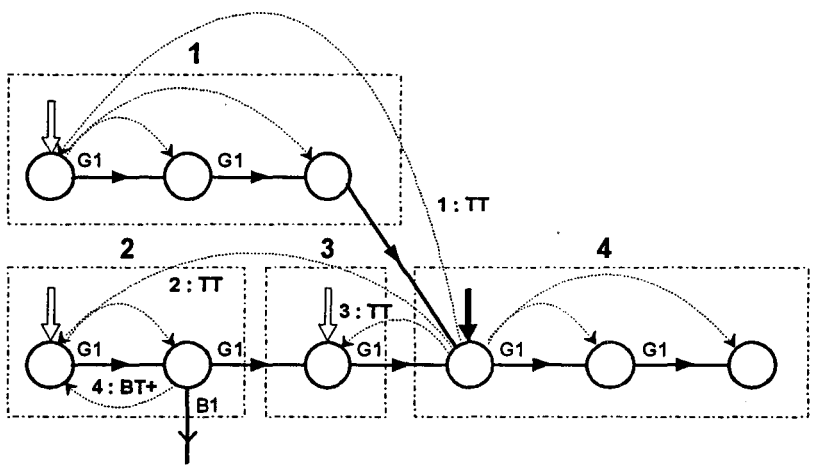


Figure C.15 : arcs inter-zones (exemple 4 – solution 1).

Zone	1	2	3	4
Flux	**N1**	**N2**	**g1**	**N1**
Flux Max	N1	Infini	N1	N1
Cible	0	0	0	N1
1 :TT	N1	0	0	N1
2 : TT	N1	N1	0	N1
3 : TT	N1	N1	N1=g1	N1
4 : BT+	**N1**	**N2=N1+b1**	**g1**	**N1**

La valeur cible (dernière ligne) de chaque zone étant égale à la valeur du flux devant circuler dans la zone (ligne 2) le graphe d'influence est complet.

Solution 2 : synchronisation par arc GT+

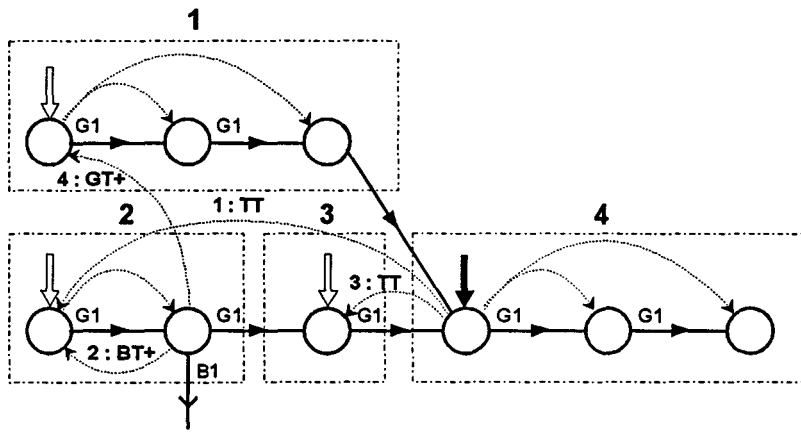


Figure C.16 : arcs inter-zones (exemple 4 – solution 2).

La fabrication du sous-ensemble correspondant à la branche supérieure n'est déclenchée que lorsqu'une pièce a subi avec réussite la seconde opération de la branche inférieure.

Zone	1	2	3	4
Flux	**N1**	**N2**	**g1**	**N1**
Flux Max	N1	Infini	N1	N1
Cible	0	0	0	N1
1 :TT	0	N1	0	N1
2 : BT+	0	N2	0	N1
3 : TT	0	N2	N1=g1	N1
4 : BT+	**N1**	**N2**	**g1**	**N1**

La valeur cible (dernière ligne) de chaque zone étant égale à la valeur du flux devant circuler dans la zone (ligne 2) le graphe d'influence est complet.

C.5 Exemple 5 : application au produit P du site

Nous donnons ici, l'application des graphes d'influence au cas du produit P assemblé sur le site DRD.

C.5.1 Gammes

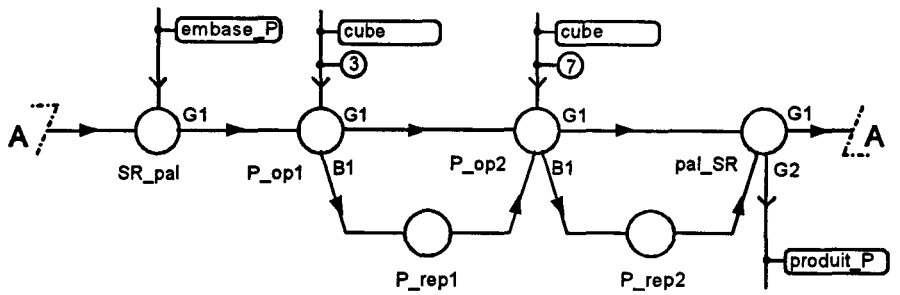


Figure C.17 : gammes associées au produit P (exemple 5).

C.5.2 Choix de l'opération cible et définition des zones

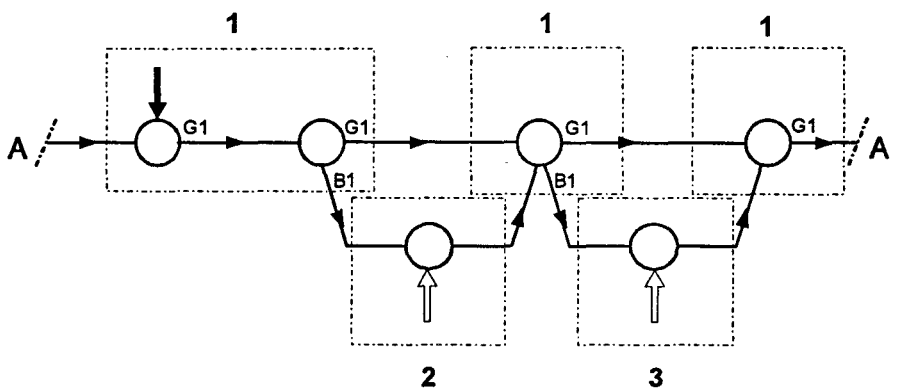


Figure C.18 : définition des zones (exemple 5).

C.5.3 Construction des arcs TT dans les zones

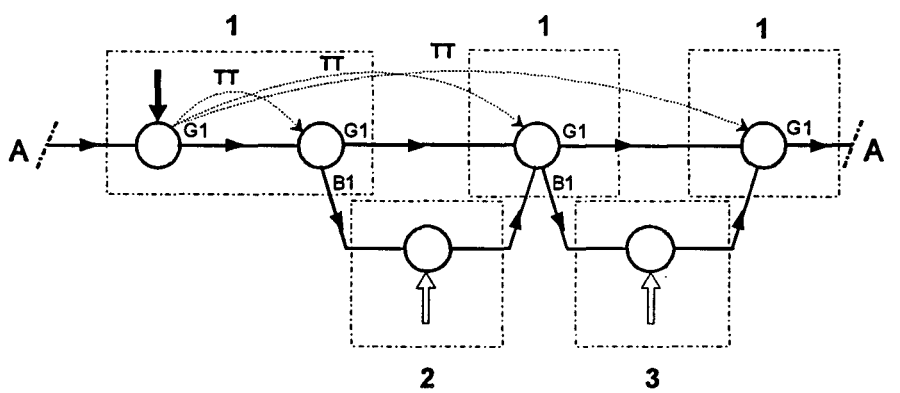


Figure C.19 : construction des arcs TT dans les zones (exemple 5).

C.5.4 Définition des flux de zone

Zone	1	2	3
Flux	N1	b1	b2
Flux Max	N1	N1	N1
Cible	N1	0	0

Les équations de conservation des flux s'écrivent :

$N1 = g1 + b1$

$N1 = g2 + b2$

C.5.5 Ajout des influences inter-zones

Quatre arcs d'influence inter-zones sont nécessaires afin d'équilibrer les flux circulant dans chaque zone.

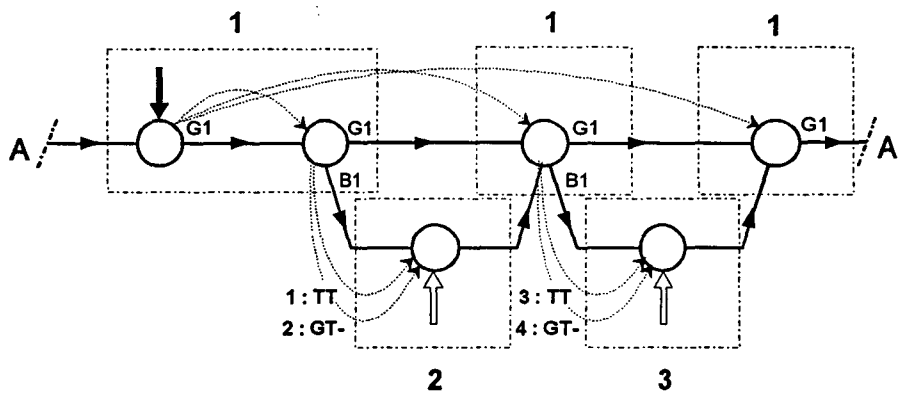


Figure C.20 : arcs inter-zones (exemple 5).

Zone	1	2	3
Flux	N1	b1	b2
Flux Max	N1	N1	N1
Cible	N1	0	0
1 : TT	N1	N1	0
2 : GT-	N1	N1 - g1	
3 : TT	N1	N1 - g1	N1
3 : GT-	N1	b1 = N1 - g1	b2 = N2 - g2

La valeur cible (dernière ligne) de chaque zone étant égale à la valeur du flux devant circuler dans la zone (ligne 2) le graphe d'influence est complet.

Annexe D

Architecture logicielle du superviseur

Nous présentons dans cette annexe l'architecture logicielle interne du superviseur développé dans le cadre du projet SCOPES.

Le premier paragraphe expose les principes d'intégration des divers modules fonctionnels dans le logiciel MONITOR 77/2 de Telemecanique. La notion de tâche logicielle est exposée. Les principales tâches utilisées sont présentées dans un second paragraphe. Les divers flux de données entre tâches dans les deux modes proposés (supervision et simulation) sont détaillés dans le troisième paragraphe.

Les tâches de pilotage (XFLOW) et de simulation (XSIM), développées dans le cadre de ce travail, font l'objet d'une présentation particulière en annexes E et F.

D.1 Intégration dans MONITOR 77/2.

MONITOR 77/2 est un système de supervision multitâche largement répandu dans l'industrie. La version utilisée dans SCOPES fonctionne sous environnement OS/2 [Telemecanique, 1992].

L'intégration dans MONITOR des modules fonctionnels présentés dans le chapitre 3 a été réalisée par l'utilisation de tâches MONITOR existantes :

- tâche graphique ;
- tâche analyse SPC/SQC ;
- tâche alarme ;
- tâches d'interfaçage avec un SGBD (Oracle, Dbase IV, etc.) ;

et par le développement de trois tâches supplémentaires principales :

- tâche ordonnancement;
- tâche pilotage (XFLOW);
- tâche gestion des stocks.

La correspondance entre modules fonctionnels et tâches est résumée dans le tableau suivant :

Module fonctionnel	Tâches associées
Ordonnancement	tâche ordonnancement tâche graphique tâche SGBD
Gestion des stocks	tâche gestion des stocks tâche graphique tâche SGBD
Pilotage	tâche pilotage tâche graphique
Surveillance	tâche graphique tâche alarme
Analyse	tâche SPC/SQC tâche graphique
Gestion des stocks	tâche gestion des stocks tâche graphique tâche SGBD

Les deux modes d'utilisation possibles du superviseur sont obtenus par l'utilisation des tâches suivantes :

- tâches de communication avec les automates (fournies dans le logiciel de base) pour le mode supervisé ;
- tâche simulation (XSIM – développé spécifiquement) pour le mode simulé.

La mise en oeuvre des tâches dans une application de supervision s'effectue en deux phases :

• Première phase : **configuration**

Lors de cette phase, un **développeur** configure chaque tâche selon l'application envisagée. Par exemple, il indique les données sur l'atelier utilisées par la tâche simulation à l'aide des outils de configuration de MONITOR : nature des stations, des systèmes de transfert, temps opératoires, etc.

• Deuxième phase : **exploitation**

Lors de cette phase, un **utilisateur** exploite les fonctions du système. Par exemple, un opérateur aura accès au synoptique de l'atelier, et pourra obtenir des informations sur une station particulière.

Pour faciliter la mise en oeuvre, une **configuration de référence** pour une application standard est proposée au développeur. Cette application comprend notamment une interface complète, prévoyant un enchaînement d'écrans pour les différents services et l'affichage d'informations standards. Certains écrans peuvent être protégés. L'accès à ces écrans, en phase exploitation, se fait par l'intermédiaire d'un mot de passe. Les écrans permettant de régler des paramètres peuvent être ainsi protégés des erreurs d'utilisation par le personnel non habilité.

La configuration repose sur des bibliothèques de référence dans lesquelles il suffit de choisir les éléments nécessaires et de les paramétrer pour l'application. Des valeurs par défaut des paramètres facilitent cette phase. Aucune connaissance en informatique n'est requise.

D.2 Les tâches

Nous passons ici en revue les principales tâches utilisées dans le cadre d'une application de supervision respectant la philosophie du projet SCOPES. Les tâches non décrites correspondent à des tâches MONITOR existantes réalisant des fonctions classiques (gestion d'alarmes, calculs interprétés et compilés, courbes de tendance, transfert de programmes, gestion de recettes, génération de rapports, etc.). Une description des ces tâches peut être trouvée dans [Telemecanique, 1992].

D.2.1 Tâche ordonnancement [Agneray, 1995].

La fonction de la tâche ordonnancement est d'établir en temps réel les ordres de fabrication à partir d'un portefeuille de commandes transmis par un système de GPAO amont (sous forme d'un fichier) ou entré directement par l'utilisateur (par l'intermédiaire de la tâche graphique).

Elle calcule une séquence d'OF et décide du lancement d'un ordre en le transmettant à la tâche pilotage. Ce choix doit bien sûr être cohérent avec l'état courant de l'atelier (ne pas lancer en fabrication un produit nécessitant une ressource actuellement en panne). Il s'agit donc d'un problème d'optimisation sous contraintes suivant un critère laissé au libre choix de l'utilisateur. Les algorithmes permettant de traiter de tels problèmes sont l'objet d'une littérature

abondante. L'un des objectifs des travaux à venir est de déterminer le ou les algorithmes les plus adaptés à chaque type d'atelier.

D.2.2 Tâche gestion des stocks [Leroux, 1994] [Agneray, 1995]

En phase configuration, le développeur décide, pour chaque stock de la politique de gestion à appliquer. En phase exploitation, cette tâche a pour but de faire respecter les politiques choisies. Elle surveille les points critiques pour lesquels le niveau des stocks passe sous une limite donnée, applique les algorithmes des méthodes d'approvisionnements configurés et communique les commandes éventuelles à la tâche ordonnancement.

D.2.3 Tâche pilotage (XFLOW) [Agneray et Vidril, 1995]

Cette tâche fait l'objet de l'annexe F.

D.2.4 Tâche simulation (XSIM) [Vidril, 1995]

Cette tâche fait l'objet de l'annexe E.

D.2.5 Tâches de communication [Telemecanique, 1992]

Cet ensemble de tâches assure la communication entre automates de station (niveau local) et le superviseur (niveau global). Ces tâches sont fournies avec le logiciel MONITOR.

D.2.6 Tâche graphique [Telemecanique, 1992]

Cette tâche réalise l'interface avec l'utilisateur pour l'ensemble des modules fonctionnels du superviseur. Par exemple des écrans doivent être définis pour :

- la visualisation et la modification des commandes et des OF manipulés par le module d'ordonnancement ;
- l'affichage de courbes ou de diagrammes générés par le module d'analyse ;
- la visualisation et l'acquittement des messages d'alarme (module surveillance) ;

- la visualisation sous forme de synoptiques de l'état global de l'atelier (module surveillance) ;
- l'affichage de l'état d'avancement des bons de fabrication gérés par le module de pilotage ;
- l'édition des règles d'aiguillage et de destination utilisées par le module de pilotage ;
- etc.

Cette tâche constitue également l'une des tâches fournies avec MONITOR.

D.2.7 Tâche analyse.

La tâche analyse a pour fonction de réaliser des traitements statistiques sur certaines données caractéristiques de l'état du procédé (durée des pannes, nombre de pièces rebutées, taux d'occupation des équipements, ...). Cette tâche est fournie en tant qu'extension logicielle à MONITOR.

D.3 Flux d'informations entre tâches.

Nous présentons ci-après le scénario d'échange entre tâches principales dans une application de supervision typique constitué d'un superviseur de cellule coordonnant un ensemble d'automates de station (scénario du projet SCOPES).

Nous supposons ici la phase configuration des tâches effectuée. Nous nous intéressons aux divers échanges intervenant en phase exploitation. Le passage dans cette phase n'est pas immédiat pour l'ensemble des tâches : une phase transitoire d'initialisation est en effet nécessaire afin d'accorder l'image sur laquelle raisonne le superviseur à l'état réel du procédé. Rappelons que le procédé, pour le superviseur, est représenté par les automates (principe de récursivité des systèmes – cf. § 2.2.3).

D.3.1 Mode supervision.

Les divers flux d'informations intervenant dans le mode supervision (i. e. connecté aux automates) sont représentés sur les figures D.1 et D.2.

La figure D.1 illustre les échanges en phase initialisation. Au cours de cette phase le système de supervision est passif vis à vis du procédé. Les informations remontant du procédé (messages automates) ne sont pas validées mais automatiquement considérées comme exactes. Deux étapes peuvent être distinguées :

- lecture des *données statiques* (i. e. n'évoluant pas en temps réel) à partir des informations de configuration (modèle des équipements et des produits) ; ces données constituent les modèles de connaissance utilisés au niveau de chaque tâche ;
- initialisation des *données dynamiques* (i. e. évoluant en temps réel) par scrutation des messages en provenance des automates ; ces données représentent l'état courant du procédé.

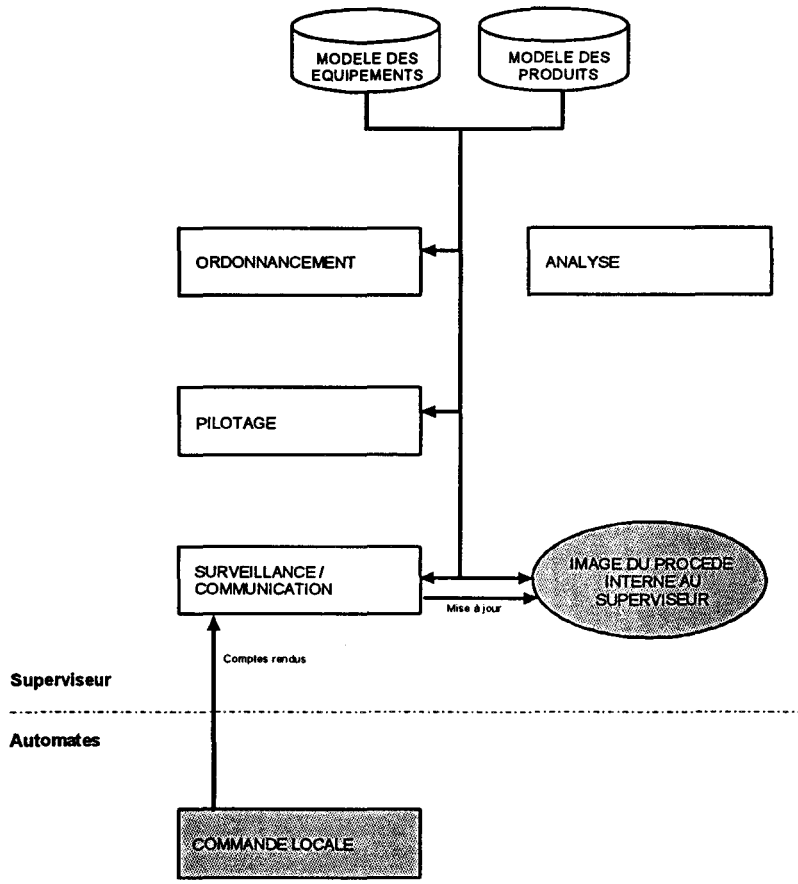


Figure D.1 : mode supervision – phase d'initialisation.

Une fois la phase initialisation complétée, le superviseur passe en phase exploitation. La figure D.2 illustre les échanges dans cette phase. Le superviseur est actif vis à vis du procédé comme l'indique le flux des ordres à destination des automates en charge de la commande locale.

A partir d'une charge de travail fournie par un poste de GPAO externe ou entrée directement par un opérateur, la tâche ordonnancement calcule une séquence ordonnée d'ordres de fabrication selon un ou plusieurs critères dépendant de la stratégie de production choisie (modèle de production) puis lance ces OF en production.

Les OF lancés sont alors pris en charge par la tâche pilotage. Cette tâche génère alors les bons de fabrication à destination de chaque station locale. Les diverses flexibilités laissées libres dans le modèle du procédé (flexibilité de

routage essentiellement) sont résolues à ce niveau conformément aux stratégies de production choisies (modèle de production).

En cas d'émission d'une requête par la commande locale (mécanisme des règles par destination), cette dernière est transmise au pilotage à travers la tâche communication. Après traitement, la réponse est transmise aux automates selon le même principe.

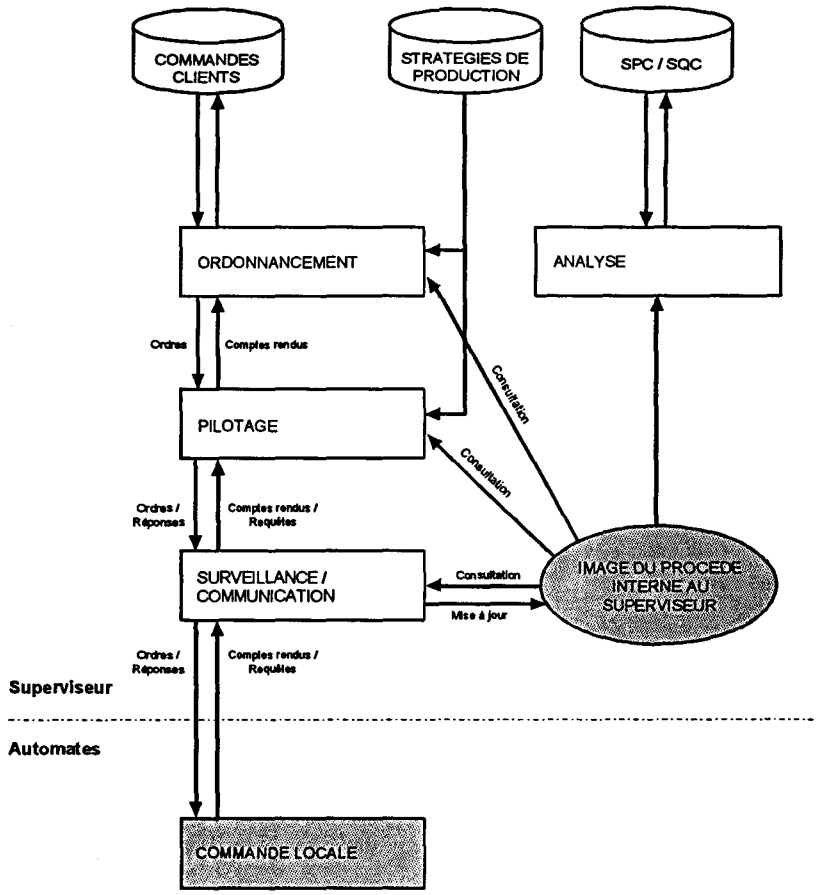


Figure D.2 : mode supervision – phase d'exploitation.

Les comptes rendus d'exécution sont traités par les tâches de surveillance qui mettent à jour les données dynamiques formant l'image du procédé interne au superviseur. Ces données servent d'état référence pour l'ensemble des tâches.

Les données formant l'image interne du procédé sont traitées par la tâche analyse pour alimenter les bases statistiques utilisées par les services maintenance et contrôle-qualité.

D.3.2 Mode simulation.

Nous rappelons que la simulation est considérée comme un outil de mise au point du système de supervision. Un minimum de différences doit donc apparaître entre les deux modes. Cela est notamment vrai au niveau des flux

d'informations entre tâches. Seuls les flux à destination et en provenance des automates disparaissent. Tous les autres restent inchangés, ce qui permet d'utiliser les tâches ordonnancement, pilotage et analyse dans les deux modes. Les figures D.3 et D.4 représentent le scénario d'échange entre tâches dans le mode simulation. On se reportera au paragraphe précédent pour une explication des divers flux.

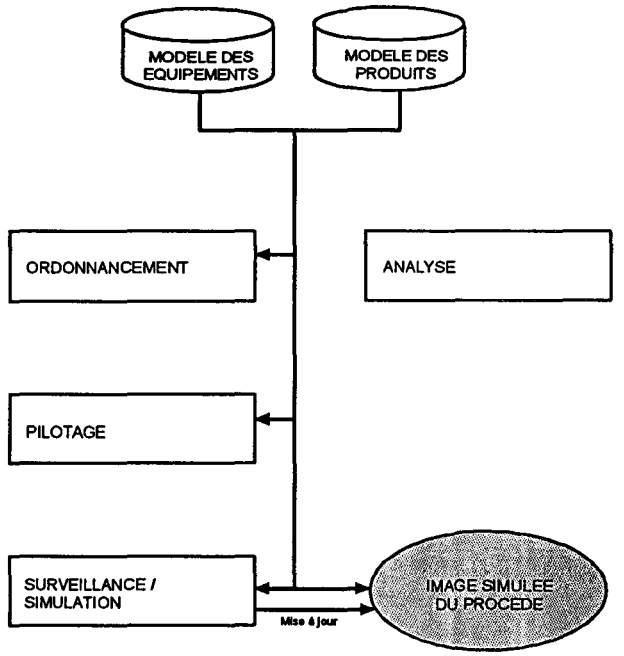


Figure D.3 : mode simulation – phase d'initialisation.

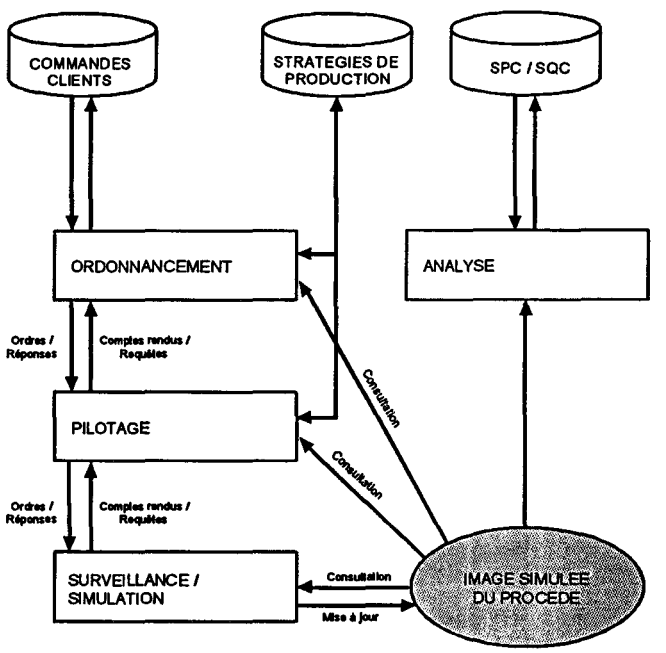


Figure D.4 : mode simulation – phase d'exploitation.

Annexe E

Tâche de simulation (XSIM)

E.1 Objectif et philosophie adoptée

La tâche de simulation permet la mise au point du système de supervision sans connexion aux équipements réels. Elle autorise notamment le réglage des stratégies de pilotage et des méthodes d'ordonnancement. Le développeur peut ainsi paramétrer le système hors ligne et analyser le comportement de l'installation avant d'installer le système en supervision sur site. Il peut également valider l'interface avec l'utilisateur (enchaînement d'écrans, définition des synoptiques, nature des affichages). Enfin, l'utilisateur peut apprendre le fonctionnement du superviseur avant son installation définitive et sa connexion à la ligne d'assemblage.

La philosophie choisie pour l'implantation est de préserver le maximum de ressemblances entre supervision et simulation. Ainsi, seules les tâches de communication disparaissent en mode simulation (cf. annexe D). Les fonctions que remplissaient ces tâches en mode supervisé sont reportées dans la tâche simulation, ainsi que celles réalisées au niveau des automates (ensemble des fonctionnalités locales de commande, pilotage et ordonnancement). Le fonctionnement du niveau local simulé est strictement identique (lorsque le modèle est correct !) à celui implanté dans les automates, rendant ainsi la simulation transparente aux autres tâches de supervision[1].

La base de données représentant l'état du système (réel ou simulé) n'étant pas dupliquée, il n'est pas possible d'utiliser simultanément les modes de simulation et de supervision. Le développeur configure et valide les diverses tâches de supervision sur un système utilisé en mode simulation, par exemple au Bureau d'Etudes, puis transfère la configuration dans un système en mode

[1] Notons néanmoins que le principe adopté ne permet pas de prendre en compte les contraintes liées au réseau de communication employé. Ceci n'est pas critique dans la mesure où les solutions de pilotage proposées permettent de s'abstraire de ces contraintes (cf. principe de scrutation asynchrone – § 11.4.3).

supervision dans l'atelier. Une fois sur site, il reste possible de passer du mode de supervision au mode de simulation mais le superviseur doit alors être déconnecté du procédé réel (plus de communication avec les automates).

E.2 Principes de simulation

Simuler, c'est exécuter le modèle de la **dynamique** du système étudié. Dans le cas de la modélisation par RdP que nous avons adoptée (cf. chapitre 9), un certain nombre de fonctionnalités doivent être rajoutées au modèle afin de prendre en compte l'ensemble des caractéristiques dynamiques du procédé modélisé. Il s'agit de la gestion du temps, de la résolution des indéterminismes et de la prise en compte des divers aléas pouvant survenir lors de la réalisation des opérations.

E.2.1 Temporisation du modèle

Le formalisme des RdP est essentiellement asynchrone : il permet de décrire la causalité (un événement est une conséquence d'un autre) mais pas les dates ni les durées. Les contraintes temporelles du type durée peuvent être prises en compte en temporisant certains éléments du réseau (places, transitions ou arcs). La temporisation revêt une importance particulière en simulation : elle permet en effet de contrôler l'horloge indiquant le temps simulé. Deux mécanismes sont possibles : soit la durée associée correspond à une durée d'indisponibilité – les jetons utilisés dans le tir de la transition ne sont visibles dans aucune place pendant un certain délai – et on parle de **RdP temporisés** ; soit la durée associée est une durée de sensibilisation – le tir est instantané, mais la transition doit être sensibilisée dans un intervalle de temps donné – et on parle de **RdP temporels**.

Le mécanisme actuellement implanté dans XSIM correspond au premier principe. Dans la solution adoptée, la temporisation n'est cependant plus attachée directement à la transition, mais aux variables associées à la transition. Ainsi deux objets franchissant une même transition peuvent rester invisibles dans les places aval pendant des durées différentes. Cette distinction est à relier à l'individualisation des jetons dans les RdPO.

E.2.2 Aspects stochastiques

La prise en compte des phénomènes aléatoires dans une simulation est indispensable afin de se rapprocher au plus près de la réalité. Bien que la mise en place d'une politique de gestion de la qualité correcte minimise ces phénomènes de façon considérable, ils ne pourront en aucun cas jamais être complètement éliminés. Il s'avère donc indispensable de les prendre en compte

lors de la conception du système de commande. La mise en place de boucle de retouches et de mécanismes de compensation en cas de rebut constituent deux possibilités offertes par les outils de pilotage proposés dans ce travail.

XSIM permet d'étudier le bon fonctionnement de ces mécanismes en offrant la possibilité de générer certains aléas suivant des lois de distribution particulières (tirage aléatoire suivant une loi uniforme, normale, Log-normale, ...) laissées au libre choix de l'utilisateur. L'implantation de cette fonctionnalité a été facilitée par le choix du formalisme des RdP à objets. Les lois de distribution aléatoires sont codées sous forme d'attributs associés aux jetons-objets circulant dans le réseau et utilisés lors du tir des transitions. Ainsi est-il possible de déclarer sur les jetons associés aux ressources certains paramètres permettant de simuler les occurrences de pannes et d'enrayage des moyens de production. Deux paramètres sont nécessaires pour simuler ces aléas : une lois de fréquence permettant de générer les instants d'occurrence et une loi de durée permettant d'en spécifier la durée. La définition d'un paramètre "état de conformité" sur les jetons associés aux produits, initialisé aléatoirement suivant une loi donnée lors du franchissement des transitions correspondant aux opérations, permet d'orienter les produits suivant leur état de conformité.

Ces exemples, certes classiques, ne sont pas exhaustifs des possibilités de XSIM. En effet, la définition des paramètres associés aux jetons-objets est laissée au libre choix de l'utilisateur, lui permettant ainsi de définir d'autres phénomènes propres à son procédé. Néanmoins, il convient de rester conscient des limites de la simulation vis-à-vis des phénomènes rares : un événement de probabilité trop faible est susceptible de ne jamais apparaître en simulation. Si l'objectif recherché est l'étude de la réaction du système de pilotage à de tels événements, il convient alors d'augmenter artificiellement leur probabilité d'occurrence.

E.2.3 Niveau décisionnel

Le niveau décisionnel ne fait pas à proprement parler de la simulation et n'est donc pas intégré à la tâche XSIM. La détection des conflits est cependant parfaitement réalisé par le logiciel qui fait appel à un module décisionnel externe pour obtenir une réponse. L'implantation du module décisionnel repose actuellement sur une tâche MONITOR standard permettant à l'utilisateur de définir ses propres routines de traitement dans un langage interprété ou compilé proche du pascal (tâche "math et logique" [Telemecanique, 1992]). Un ensemble de procédures normalisées a été développé en accord avec les spécifications des fonctionnalités du niveau local :

- gestion des aiguillages (traitement des règles d'aiguillage) ;
- gestion des indexeurs au niveau des postes opératoires (traitement des bons de travail locaux et mécanisme des règles de routage) ;
- calcul des données statistiques locales sur l'état du procédé (nombre d'opérations effectuées, durées cumulées dans chaque mode de marche, indicateurs de performance, etc.).

L'interfaçage entre les deux tâches s'effectue selon un mécanisme classique de synchronisation par requêtes avec accusés de réception.

E.2.4 Noyau de simulation

Le noyau de simulation de XSIM se divise en deux parties, correspondant à deux cycles différents dans la boucle générale de simulation représentée sur la figure E.1.

L'**interpréteur** (ou **joueur**) de RdP met en oeuvre les règles d'évolution des RdP. Il détermine qu'elles sont les transitions franchissables puis effectue leur franchissement par les jetons objets. En cas de conflit, il met en forme le problème avant d'interroger le niveau décisionnel qui lui donne en retour les actions à effectuer. L'interpréteur recherche un état stable du réseau, c'est à dire qu'il fait évoluer celui-ci jusqu'à être bloqué en attente d'un événement nouveau (cycle par **activité**).

Le mécanisme de **gestion des événements** se charge quant à lui de faire évoluer le temps. Ce mécanisme tient à jour une liste ordonnée d'événements en attente ou échéancier. Chaque fois que le réseau est dans un état stable ces événements en attente sont évalués. L'horloge de simulation prend la valeur de la date d'occurrence du premier événement, puis les événements portant la même date sont réalisés (cycle par **événement**). L'occurrence d'un événement consiste en le marquage d'une place du RdPO par un jeton.

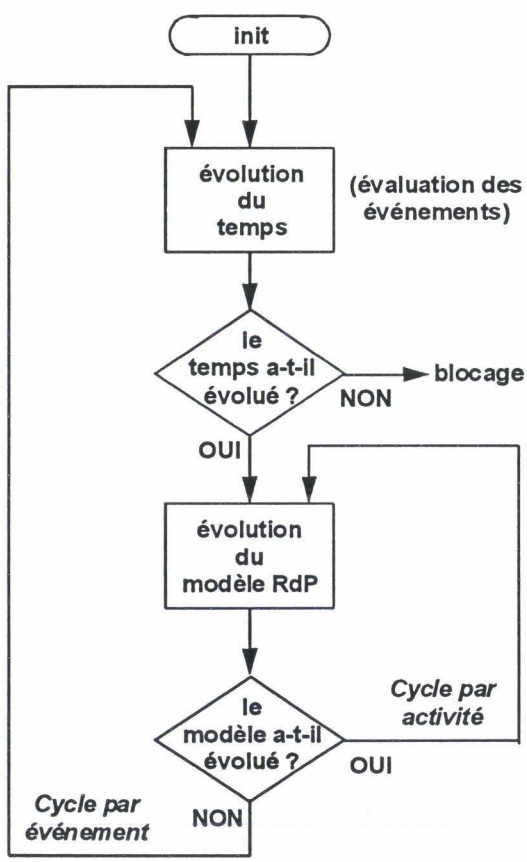


Figure E.1 : cycle de simulation.

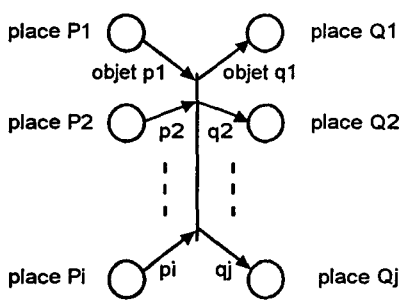
E.3 Utilisation de la tâche simulation

Comme toute tâche MONITOR, l'utilisation de la tâche XSIM peut être décomposée en deux phases (cf. annexe D) : une phase de configuration suivie d'une phase d'exploitation.

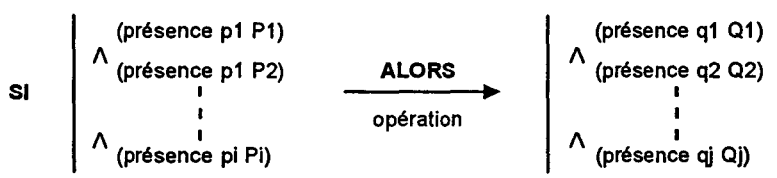
E.3.1 Phase de configuration

Destiné à être utilisé directement par des spécialistes en moyens d'assemblage, l'outil proposé ne doit pas nécessiter de connaissance particulières en techniques de simulation. Le principe de modélisation employé, notamment, doit être transparent à l'utilisateur. Un mécanisme de description du modèle faisant directement référence au formalisme des RdP n'est donc pas concevable.

Le principe de déclaration que nous avons adopté s'appuie sur la sémantique associée aux éléments du réseau : la présence d'un jeton dans une place s'interprète comme un prédicat ; les places amont à une transition peuvent alors être assimilées à des prémisses conditionnant l'opération associée à la transition et les places aval aux conséquences de cette opération. Ainsi le réseau de la figure E.2.a peut-il se traduire, en supposant que les prédicats – de la forme (présence objet lieu) – signifient la présence d'un objet sur un lieu, sous l'expression littérale de la figure E.2.b.



- a -



- b -

Figure E.2 : principe de déclaration du modèle dans XSIM.

Le formalisme adopté dans XSIM a été enrichi – en introduisant un formalisme objet dans l'expression des prédicats – afin de simplifier la déclaration et se démarquer définitivement du formalisme RdP. La configuration s'effectue en deux étapes : définition des types d'objet et des

attributs associés, puis définition des opérations. Cette seconde étape se décompose elle-même en trois phases : déclaration des objets intervenant dans l'opération, définition des prémisses et enfin définition des conséquents. L'exemple suivant illustre le formalisme employé.

Exemple : transfert d'une palette du buffer du convoyeur d'entrée de la station IBM à l'indexeur de cette même station :

SI	pal.LOC = ent_buf.BUF
ALORS	pal.LOC = ind.BUF

où : "pal", "en_buf" et "ind" désignent trois variables désignant respectivement une palette, le buffer d'entrée de la station IBM et l'indexeur de cette station ;

et : "LOC" et "BUF" représentent des attributs associés aux objets correspondants (une localisation pour l'objet palette et un buffer pur les objets buffer d'entrée et indexeur).

Outre la structure statique du modèle, des informations de configuration supplémentaires, liées aux caractéristiques dynamiques du procédé, doivent également être définies. Il s'agit :

- des durées opératoires ;
- des lois et paramètres d'enrayage (l'enrayage d'un équipement, à la différence d'une panne, ne peut avoir lieu qu'en fonctionnement). Les paramètres nécessaires sont :
  - une loi de fréquence (fréquence d'enrayage),
  - une loi de durée (durée d'enrayage) ;
- des lois et paramètres de panne (la panne d'un équipement, à la différence d'un enrayage, peut se produire à tout moment). Les paramètres nécessaires sont identiques à ceux d'une loi d'enrayage.

A titre d'illustration, nous avons représenté sur la figure E.3 les tables de définition des opérations[2]. Les données affichées correspondent à l'opération de l'exemple ci-dessus. La première fenêtre sélectionne l'opération en cours de définition (notée ici EN_BF-IND|IBM). La seconde fenêtre permet de déclarer les objets concernés par l'opération (ici une palette ainsi que le buffer d'entrée et l'indexeur de la station IBM). La troisième fenêtre décrit l'ensemble des prémisses conditionnant l'opération (il faut ici que la localisation de la palette "pal" soit le buffer "BUF" du buffer d'entrée "en_buf"). Enfin, la quatrième fenêtre permet de spécifier l'ensemble des conséquences de l'opération (la localisation de la palette "pal" devient ici le buffer "BUF" de l'indexeur "ind") ainsi que les durée opératoires (l'opération de transfert a lieu ici en 100 unités de temps)..

La déclaration du modèle ne fait donc pas référence au formalisme RdP. La construction du réseau par XSIM à partir de l'ensemble des règles déclarées

---

[2] Il existe d'autres tables pour la déclaration des objets et des attributs associés.

s'effectue ainsi de manière entièrement transparente à l'utilisateur. Néanmoins, l'entrée des données s'avère très vite fastidieuse pour un grand nombre de règles. Une étape supplémentaire vers une plus grande convivialité serait donc le développement d'une interface de configuration permettant à l'utilisateur de spécifier graphiquement la structure de son atelier à partir d'une bibliothèque d'équipements prédéfinis.

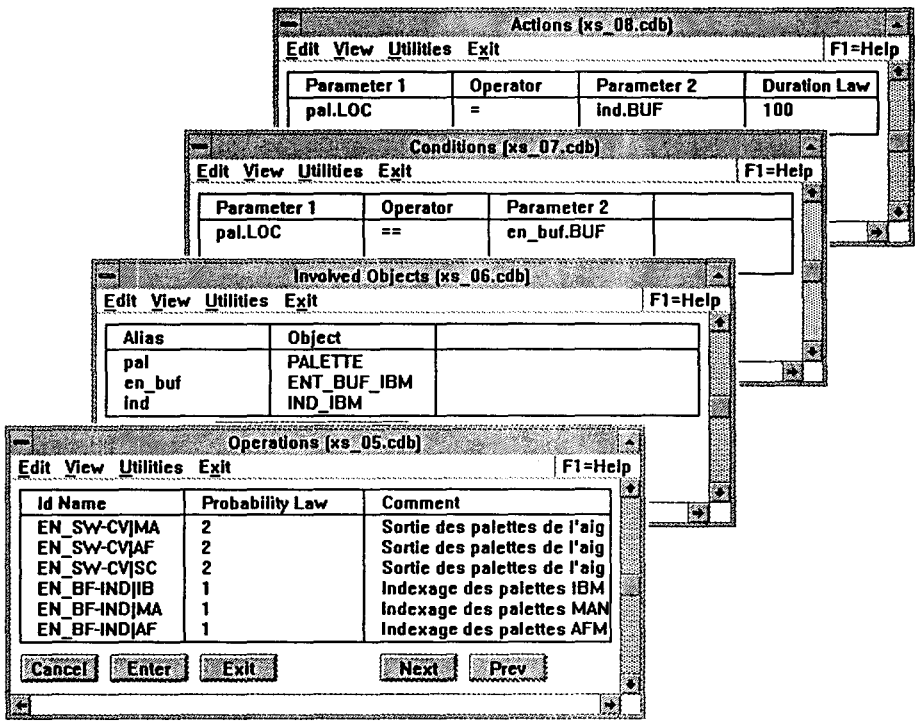


Figure E.3 : fenêtres de définition des opérations.

E.3.2 Phase d'exploitation

En phase d'exploitation, il n'est pas possible de modifier le modèle de l'installation : architecture physique, gammes, comportement des ressources (cf. chapitre 9), ainsi que les divers paramètres liés à la dynamique du procédé (durées opératoires et probabilités des aléas) restent contants. Les seules interventions possibles de l'utilisateur sur la simulation sont les suivantes :

- initialisation des horloges, en particulier de l'heure simulée qui mesure le temps, accéléré, de la simulation ;
- lancement de la simulation ;
- arrêt de la simulation (ou la demande d'une pause) : l'utilisateur peut interrompre à tout moment la simulation pour consulter l'état courant de l'installation (taux d'occupation des machines, état des équipements et des produits...) et effectuer des sauvegardes sur disque ;
- modification du pas d'évolution :
    - ♦ en mode pas à pas, la simulation s'interrompt après chaque événement significatif, ce qui permet d'étudier finement le comportement de l'installation ;

- en mode continu à pas constant (fixé par l'utilisateur), le temps simulé évolue par intervalle constant ;
- en mode continu à pas variable, on saute d'événement en événement, le temps simulé n'évoluant plus de façon uniforme.

Une fois la simulation lancée, la tâche XSIM génère et modifie dynamiquement l'état du procédé simulé d'après le modèle introduit au cours de la configuration. Elle émet des événements à destination des autres tâches de supervision (pilotage et surveillance principalement) et réagit aux événements en provenance de ces tâches. Pour gérer le trafic et le routage des pièces dans le système, le simulateur accède, via le module décisionnel, aux règles de pilotage locales des aiguillages, générées par la tâche de pilotage (cf. § 13.2). De même, les informations sur les opérations à réaliser au niveau de chaque poste sont reçues au moyen des bons de travail locaux générés également par la tâche de pilotage (cf. § 11.2.3). Les algorithmes réalisant ces traitements sont strictement identiques à ceux implantés dans les automates pour le contrôle du procédé réel. En plus de ces informations locales, le module décisionnel peut également générer des requêtes à destination du module de pilotage global afin de simuler le mécanisme des règles par destination (cf. § 13.3).

L'ensemble des modules de supervision sont disponibles lors de la simulation. Ainsi, le réglage des paramètres de pilotage et d'ordonnancement s'effectue par les écrans associés à chacune de ces tâches (cf. annexe F pour la tâche de pilotage). De même, les résultats de la simulation sont traités puis visualisés par les diverses fonctionnalités du module d'analyse. Les écrans associés au module de surveillance permettent quant à eux une visualisation en temps réel de l'évolution de la simulation. Un tel écran de surveillance, représentant un synoptique d'atelier (celui du site DRD) est représenté sur la figure E.4 .

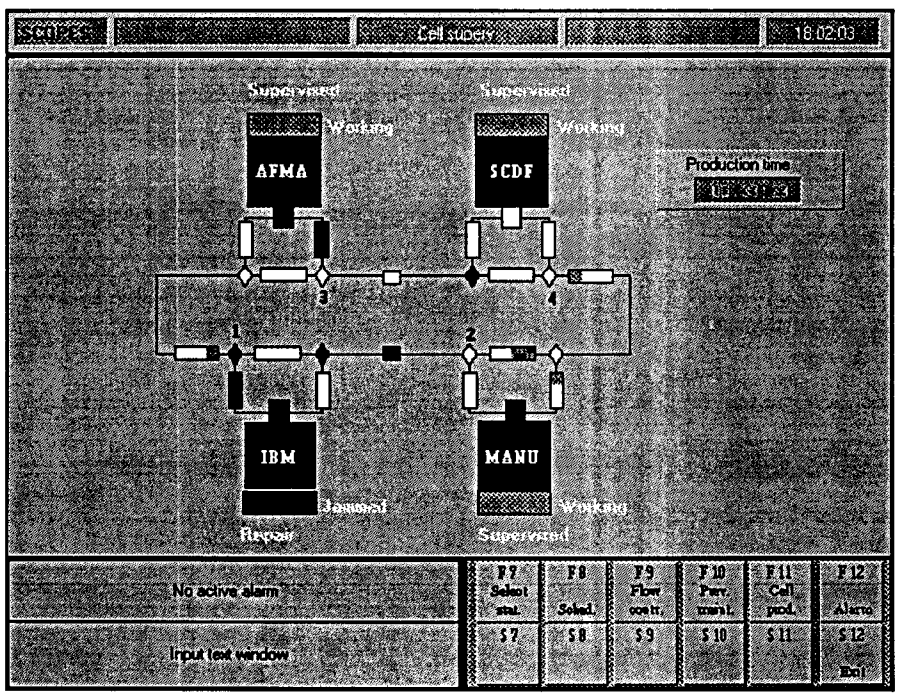


Figure E.4 : exemple d'écran de surveillance.

Annexe F

Tâche de pilotage (XFLOW)

F.1 Principe de la tâche

La tâche XFLOW réalise la gestion des diverses requêtes permettant la coordination des moyens concernés par la production, conformément aux mécanismes présentés au chapitre 11 : succinctement, elle reçoit les ordres de fabrication calculés par la tâche ordonnancement et les traduit en bons de fabrication (globaux puis locaux) qui sont transmis aux diverses tâches assurant la communication avec les automates. Elle assure légalement le suivi des bons de fabrication émis. Les fonctionnalités assurées par XFLOW sont les suivantes (il s'agit des fonctionnalités du niveau global déjà présentées au § 11.7)[1] :

- la gestion des changements de campagnes :
  - instanciation des bons de fabrication,
  - désinstanciation des bons de fabrication,
  - gestion de la signalisation (algorithme du transit souple).
- le suivi des bons de fabrication :
  - mise à jour et propagation des compteurs des bons de fabrication et des OF,
  - calcul des nouvelles consignes cibles,
  - affichage de l'état des OF et des bons de fabrication,
  - détection des fins de campagne.

Pour ce faire, la tâche XFLOW dispose des modèles des gammes et des affectations (cf. chapitre 8), ainsi que des graphes d'influence.(cf. chapitre 12).

Plutôt que de donner le détail des algorithmes, dont les principes ont déjà présentés au chapitre 11, il nous a semblé préférable de donner, dans le cadre de

[1] La communication avec le niveau local (principe de la scrutation synchrone ou asynchrone) n'est pas réalisée par XFLOW, mais par l'ensemble des tâches de communication fournies en standard avec MONITOR (cf. annexe D).

cette annexe, le cycle d'évolution de chaque requête. Ce cycle est géré par la tâche XFLOW. La position d'une requête (ou plus précisément d'une instance d'un type donné de requête) dans son cycle est codée, sous forme d'une chaîne de caractères, dans le champ "statut" associé à cette requête (cf. § 11.2).

F.1.1 Cycle de vie d'un ordre de fabrication

La valeur du "statut" associé à un ordre de production évolue suivant le graphe représenté figure F.1.

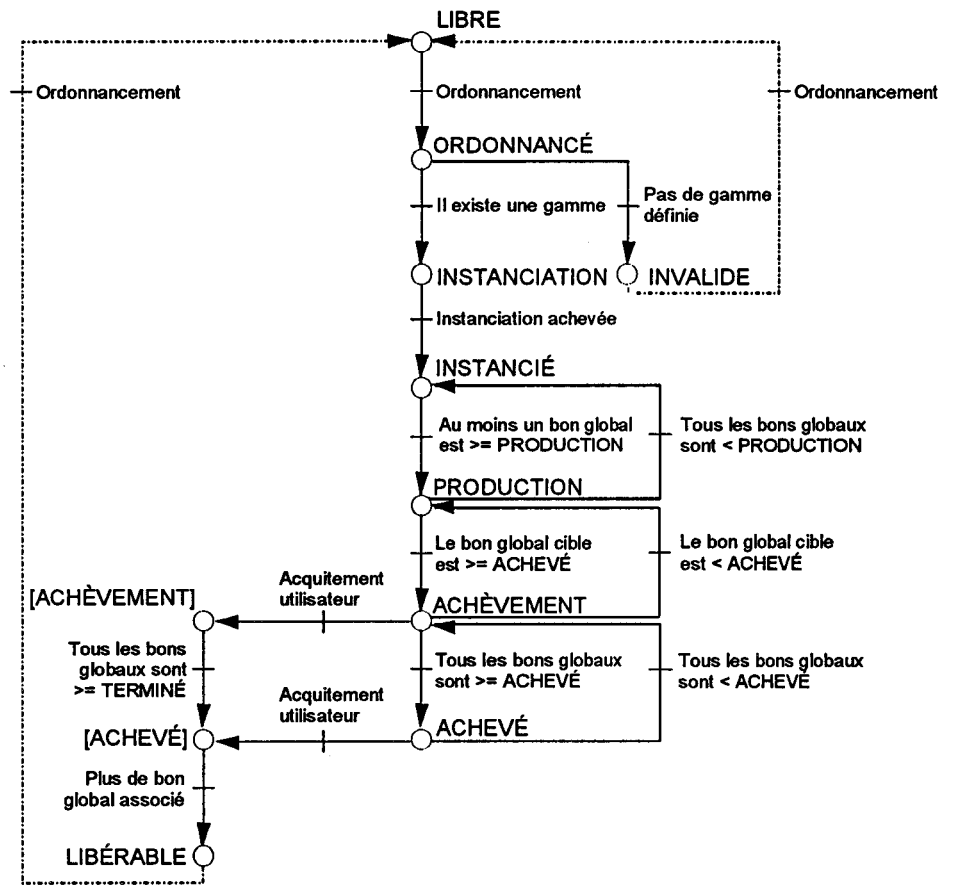


Figure F.1 : cycle de vie d'un OF.

Commentaires :

- Les évolutions correspondant aux arcs en pointillés sont gérées par la tâche ordonnancement : le passage d'un OF du statut LIBRE au statut ORDONNANCÉ constitue l'événement initiant la communication de l'ordonnancement vers le pilotage ; réciproquement le passage d'un OF du statut ACHEVÉ au statut LIBÉRABLE permet de prévenir l'ordonnancement de l'achèvement complet d'un OF.
- Les différents états ont été ordonnés afin de simplifier l'expression des conditions de changement d'état. L'évolution des OF dépend en effet, pour

certains états, de l'état des bons globaux associés (qui eux-mêmes dépendent de l'état des bons locaux associés – cf. § F.1.2). Pour les bons globaux l'ordre adopté est le suivant :

LIBRE < RÉSERVÉ < CONFIGURATION < CONFIGURÉ < [CONFIGURÉ] < PRODUCTION < [PRODUCTION] < ACHEVÉ < [ACHEVÉ] < DÉCONFIGURATION

- La libération d'un OF ne peut être initiée que lorsque l'utilisateur a acquitté l'OF. Cet acquittement n'est pris en compte que lorsque la consigne est atteinte au niveau de l'opération cible (définissant le PAC) : états ACHÈVEMENT et ACHEVÉ. La procédure de libération déclenche alors celle des bons globaux associés selon le graphe d'évolution de leurs états (cf. § F.1.2).
- Le bon global cible est le bon global associé à l'opération cible (cf. chapitre 12 pour définition de l'opération cible).
- Les arcs qui remontent des états TERMINÉ à ACHÈVEMENT, de ACHÈVEMENT à PRODUCTION et de PRODUCTION à INSTANCIÉ permettent de tenir compte des changements de valeur de la consigne par l'opérateur.

F.1.2 Cycle de vie d'un bon global

Le cycle d'évolution du "statut" associé à un bon global est représenté sur la figure F.2.

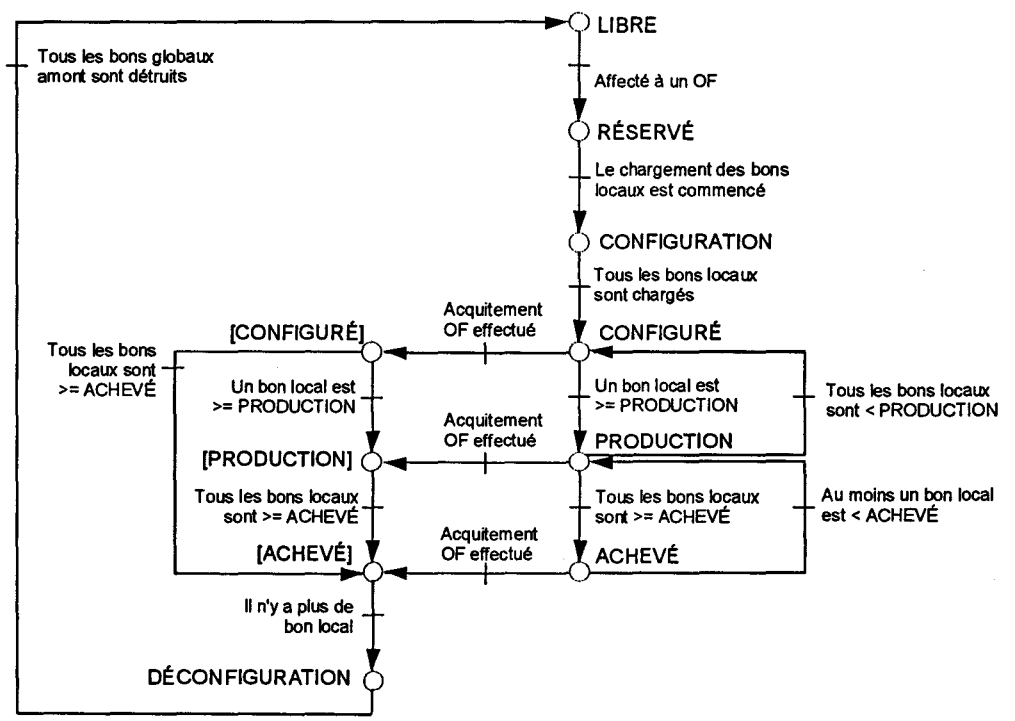


Figure F.2 : cycle de vie d'un bon global.

Commentaires :

- De façon analogue aux bons globaux, les états des bons locaux (cf. § F.1.3) ont été ordonnés afin de faciliter l'expression des conditions de changement d'état. L'ordre adopté est le suivant :

  INITIALISATION < LIBRE < RESERVÉ < [RESERVÉ] < PRODUCTION < [PRODUCTION] < ACHEVÉ < [ACHEVÉ]
- Le passage de l'état [CONFIGURÉ] à ACHEVÉ sert dans le cas où le bon local voit sa valeur cible s'annuler sans être passé par l'état PRODUCTION. Cela est possible lorsque le bon est associé à une opération située sur une boucle de retouche (par exemple une opération de réparation sur un poste manuel). On peut ainsi détruire un bon qui n'a jamais produit.
- La libération d'un bon global (initiant la procédure de déconfiguration des bons locaux associés – cf. § F.1.3) ne peut débuter que lorsque l'OF associé a lui-même été acquité.

F.1.3 Cycle de vie d'un bon local

L'évolution du champ "statut" relatif à un bon local respecte le graphe de la figure F.3.

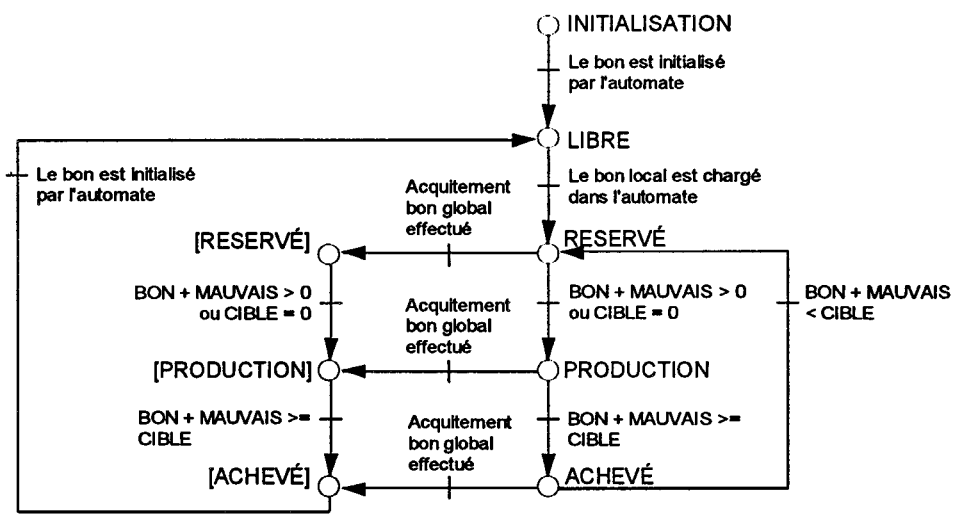


Figure F.3 : cycle de vie d'un bon local.

Commentaires :

- L'état INITIALISATION est un état transitoire permettant au superviseur de se synchroniser avec l'automate auquel est associé le bon local. La synchronisation est réalisée lorsque le champ "signal" du bon (cf. § 11.2.3) est initialisé à 0.
- La libération d'un bon local ne peut débuter que lorsque le bon global associé a lui-même été acquité.

- Les termes "BON", "MAUVAIS" et "CIBLE" dans l'expression des conditions de changements d'état désignent les valeurs des compteurs correspondants du bon local (cf. § 11.2.3).

F.2 Utilisation de la tâche pilotage

Comme toute tâche MONITOR, l'utilisation de la tâche XFLOW peut être décomposée en deux phases (cf. annexe D) : une phase de configuration suivie d'une phase d'exploitation.

F.2.1 Phase de configuration

La tâche XFLOW nécessite, préalablement à toute exécution, la configuration d'un certain nombre de données. Il s'agit d'une part de données relatives au fonctionnement du logiciel MONITOR (définition des variables de la base de données commune permettant l'échange d'informations entre tâches) et, d'autre part, de données relatives aux divers modèles utilisés par le pilotage, à savoir :

- les modèles des gammes des diverses familles de produits pouvant être assemblées sur l'installation ;
- les relations d'influence associées aux gammes précédemment définies ;
- ainsi que les diverses possibilités d'affectation pour chaque opération apparaissant sur les gammes.

Le mécanisme de configuration actuel est organisé, tout comme pour la tâche de simulation (ainsi d'ailleurs que pour la majorité des tâches MONITOR prédéfinies), sous forme de tables que le développeur vient renseigner sous forme textuel. Nous donnons sur la figure F.4 les tables permettant de définir, pour chaque famille de produit, le graphe résultant de la combinaison des gammes et des relations d'influence.

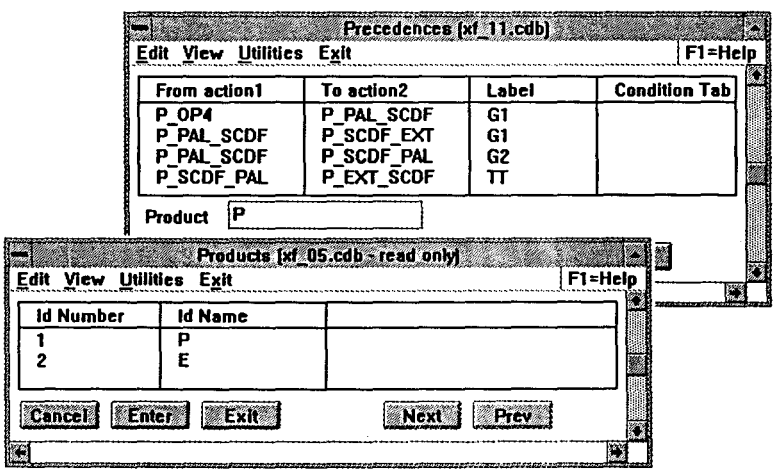


Figure F.4 : fenêtres de définition des gammes et des relations d'influence.

La première table permet de sélectionner la famille de produits, tandis que la seconde définit l'ensemble des arcs constitutifs du graphe[2]. Dans cette dernière table, la colonne "label" indique l'étiquette définissant l'état de conformité des produits pouvant emprunter cet arc (cf. § 8.1.2) ; quant à la colonne "condtion table", elle précise la table de conditions sur les variabilités s'appliquant à cet arc (cf. § 8.1.2).

F.2.2 Phase d'exploitation

Les interventions de l'utilisateur en phase d'exploitation se limitent à lancer les ordres de fabrication (en s'aidant en besoin des résultats fournis par la tâche ordonnancement) puis à surveiller l'évolution de la production. Trois types d'écran permettent l'affichage des informations concernant les diverses requêtes gérées par le pilotage :

- l'écran d'affichage des ordres de fabrication (cf. figure F.5) : il permet à l'utilisateur de visualiser les informations relatives aux divers ordres de fabrication (la fenêtre d'affichage se déplace par ligne ou par page le long de la liste des OF). Deux touches de fonction permettent de gérer les interventions de l'utilisateur : la touche F1 ("Launch") permet le lancement en production puis l'acquitement en fin de production, de l'OF couramment sélectionné, tandis que la touche F2 ("New Target") offre à l'utilisateur la possibilité de modifier la consigne de cet OF[3].

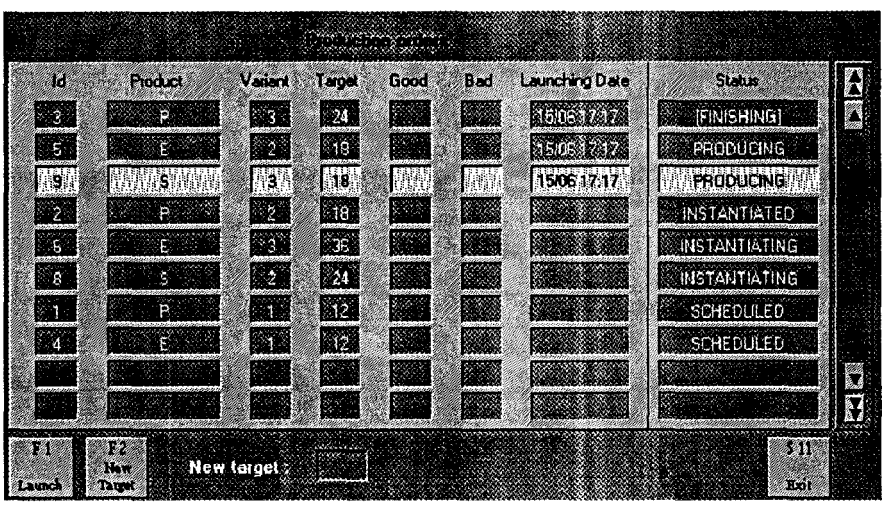


Figure F.5 : écran d'affichage des ordres de fabrication.

- l'écran d'affichage des bons de fabrication globaux (cf. figure F.6) : il permet à l'utilisateur de visualiser, sur le même principe que pour les OF, les différents bons de fabrication globaux associés à un OF donné. Aucune intervention

[2] La définition des opérations et des diverses possibilités d'affectation font l'objet d'autres tables.

[3] Une modification positive de la consigne est acceptée pour n'importe quel état de l'OF avant acquittement. Une modification négative ne peut avoir lieu, toujours avant acquittement, que si la nouvelle consigne est supérieure à la quantité déjà produite augmentée d'une valeur de sécurité.

utilisateur (autre que le parcours de la liste) n'est permise : les données des bons globaux ne peuvent être modifiées.

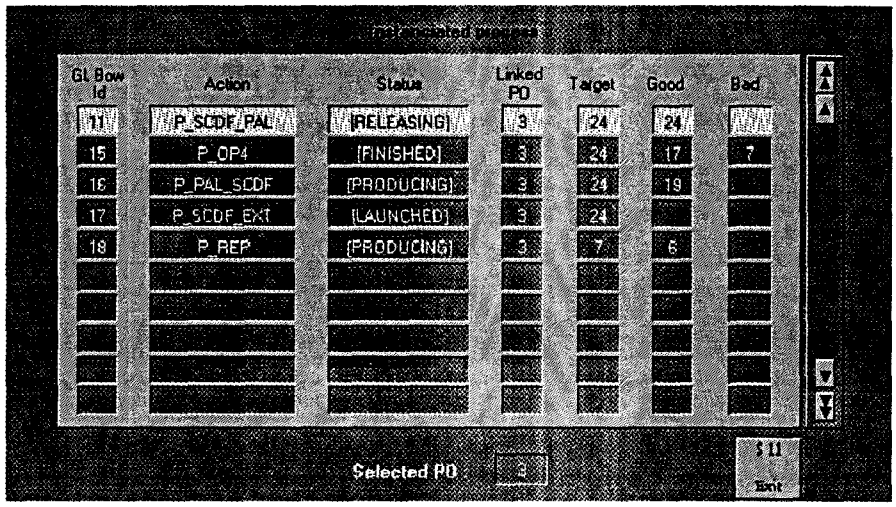


Figure F.6 : écran d'affichage des bons de fabrication globaux.

- l'écran d'affichage des bons de fabrication locaux (cf. figure F.7) : fonctionnant sur le même principe que les écrans précédents, il affiche les informations relatives aux divers bons de fabrication locaux d'une station locale donnée. Là encore aucune modification des données par l'utilisateur n'est permise.

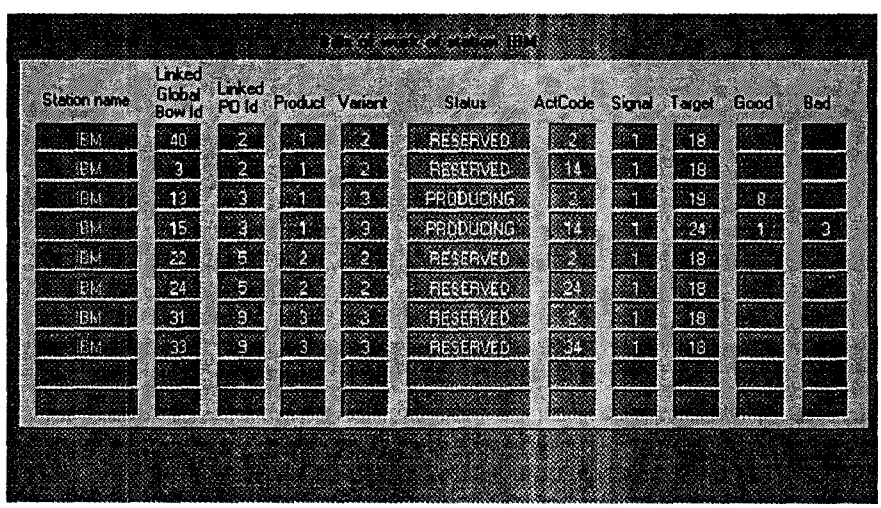


Figure F.7 : écran d'affichage des bons de fabrication locaux.

BIBLIOTHÈQUE INTERUNIVERSITAIRE
SECTION DE SCIENCES
LILLE