



**HAL**  
open science

# Commande et supervision des procédés discontinus : une approche hybride

David Andreu

► **To cite this version:**

David Andreu. Commande et supervision des procédés discontinus : une approche hybride. Automatique / Robotique. Université Paul Sabatier - Toulouse III, 1996. Français. NNT : . tel-00139761

**HAL Id: tel-00139761**

**<https://theses.hal.science/tel-00139761v1>**

Submitted on 3 Apr 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Année 1996

---

# THÈSE

---

présentée au

Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS

en vue de l'obtention du

Doctorat de l'Université Paul Sabatier de Toulouse

Spécialité: Informatique Industrielle

par

David ANDREU

Maître ès Sciences

---

## COMMANDE ET SUPERVISION DES PROCÉDÉS DISCONTINUS: UNE APPROCHE HYBRIDE

---

Soutenue le 15 Novembre 1996 devant le jury :

Président	M.	COURVOISIER <i>Professeur UPS, Directeur de l'INSA (Toulouse)</i>
Rapporteurs	S.	MACCHIETTO <i>Professeur Imperial College (Londres)</i>
	F.	PRUNET <i>Professeur USTL (Montpellier)</i>
	Y.	QUENEC'H DU <i>Professeur SUPELEC, Chef de Service (Rennes)</i>
Examineurs	J.	BARRAULT <i>Ingénieur, Responsable des applications procédés SIEMENS (Nice)</i>
	H.	PINGAUD <i>Maître de Conférences ENSIGC (Toulouse)</i>
	R.	VALETTE <i>Directeur de recherche au LAAS-CNRS (Toulouse)</i>
Directeur de thèse	J.C.	PASCAL <i>Maître de Conférence UPS (Toulouse)</i>

Thèse préparée au Laboratoire d'Analyse et d'Architecture des Systèmes  
du CNRS, 7 avenue du Colonel Roche 31077 Toulouse Cédex 4

Rapport LAAS N° 96429

## Avant-propos

Les travaux présentés dans ce manuscrit ont été effectués au Laboratoire d'Analyse et d'Architecture des Systèmes du Centre National de la Recherche Scientifique, au sein du groupe Systèmes de Production. Je tiens à exprimer ma gratitude à Monsieur Alain Costes, Directeur du LAAS, et à Monsieur Jean-Claude Hennet, Responsable du groupe SP, pour leur accueil.

Je tiens à remercier tout particulièrement mon directeur de thèse, Monsieur Jean-Claude Pascal pour la confiance et la liberté d'action qu'il m'a accordées. Merci pour tes encouragements, ton soutien et ta complicité pendant ces années de recherche.

Je suis extrêmement reconnaissant à Messieurs François Prunet, Yves Quenec'hdu et Sandro Macchietto, pour avoir accepté la charge d'être rapporteurs des travaux présentés. J'exprime également ma reconnaissance à Messieurs Marc Courvoisier et Joel Barrault pour avoir accepté de prendre part au jury. Je leur exprime à tous mes remerciements pour l'intérêt qu'ils ont porté à ces travaux.

Par ailleurs, je ne saurais oublier l'accueil chaleureux à l'Imperial College de Monsieur Macchietto et de son équipe, durant un séjour qui fut très enrichissant.

Je désire exprimer toute ma gratitude à Messieurs Robert Valette et Hervé Pingaud pour leur collaboration inestimable, leur disponibilité et les nombreuses discussions et conseils qui m'ont aidé et orienté dans ce travail. Je tiens à leur témoigner toute mon amitié.

Je remercie également tous les membres du groupe SP pour leur soutien amical, et plus particulièrement Philippe et Michel, souvent mis à contribution durant mon travail, pour leur aide et leur gentillesse.

Durant ces années passées au LAAS, j'ai pu apprécier l'amitié de certaines personnes qui, au delà de l'ambiance très amicale qu'ils ont su créer, m'ont toujours apporté leur soutien dans les moments critiques. Pour tout ce que nous avons pu partager je vous dis Merci.

Mes remerciements enfin à tout le personnel technique et administratif du LAAS, avec une attention particulière envers Madame Dufour, Messieurs Berty et Daurat pour leur efficacité et leur disponibilité.

Je ne saurais terminer sans rendre un hommage à trois êtres extraordinaires Karine, Serge et Maxime. Je souhaite leur dédier ce mémoire.

# Table des matières

<b>Table des matières</b>	<b>I</b>
<b>Introduction</b>	<b>1</b>
<b>Première Partie : Cadre de l'étude</b>	<b>4</b>
<b>1 Les procédés discontinus</b>	<b>5</b>
1 Classification des systèmes de production . . . . .	5
1.1 Les systèmes continus . . . . .	6
1.2 Les systèmes à événements discrets . . . . .	7
1.3 Les systèmes discontinus . . . . .	9
2 Spécificités des procédés à traitement par lots . . . . .	10
2.1 Le produit . . . . .	10
2.1.1 La nature continue . . . . .	10
2.1.2 La notion de lot . . . . .	11
2.1.3 L'état du lot . . . . .	11
2.1.4 Les contraintes technologiques . . . . .	12
2.2 Le procédé . . . . .	12
2.2.1 Exemple de procédé de distillation . . . . .	13
2.2.2 Exemple de réacteur exothermique . . . . .	14
2.2.3 La configuration du procédé . . . . .	15
2.3 Le processus . . . . .	17
3 Conclusion . . . . .	19
<b>2 La commande et la supervision des procédés à traitement par lots</b>	<b>21</b>
1 La «norme» pour la conception du système de commande/supervision . . . . .	22
1.1 La hiérarchisation selon l'équipement . . . . .	22
1.2 La hiérarchisation selon les recettes . . . . .	23
1.3 La hiérarchisation fonctionnelle . . . . .	25
1.4 La hiérarchisation de l'aspect procédural de la recette . . . . .	26
2 Exemples de systèmes de commande/supervision . . . . .	27
2.1 Les systèmes de contrôle/commande basés sur des unités pilotes . . . . .	28
2.1.1 SUPERBATCH [Liu & Macchietto 95] . . . . .	28



2.1.2	FLESCHE [Puigjaner <i>et al.</i> 94] . . . . .	30
2.2	Les systèmes de contrôle/commande industriels . . . . .	32
3	La problématique de la surveillance . . . . .	33
3.1	La surveillance des systèmes à événements discrets . . . . .	34
3.2	La surveillance des systèmes continus et discontinus . . . . .	35
3.3	Limitations d'une approche purement discrète de la supervision . . . . .	36
4	Conclusion . . . . .	38
<b>3</b>	<b>La modélisation</b> . . . . .	<b>39</b>
1	Les différentes approches de l'aspect hybride . . . . .	39
1.1	L'approche intégrée . . . . .	39
1.2	L'approche séparée . . . . .	40
2	Approches basées sur les réseaux de Petri . . . . .	41
2.1	Les approches intégrées . . . . .	42
2.1.1	Les RdP Continus . . . . .	43
2.1.2	Les RdP Hybrides . . . . .	45
2.1.3	Les RdP Lots . . . . .	46
2.1.4	Les RdP Continus Causaux . . . . .	47
2.2	Les approches séparées . . . . .	49
2.2.1	Le Grafchart et des systèmes d'équations . . . . .	49
2.2.2	Les RdP Interprétés et des systèmes d'équations . . . . .	50
2.2.3	Les RdP Colorés Temporisés à Décision Interprétés et un système d'équations . . . . .	53
3	Comparaison des approches: conclusion . . . . .	56
	<b>Deuxième Partie: Une nouvelle approche pour la conduite des procédés discontinus</b> . . . . .	<b>59</b>
<b>4</b>	<b>Principes de l'approche</b> . . . . .	<b>61</b>
1	Réflexions sur l'influence de la notion de configuration dynamique . . . . .	61
1.1	La commande modulaire . . . . .	62
1.1.1	Un exemple simple . . . . .	62
1.1.2	La conduite d'un procédé de distillation batch . . . . .	64
1.1.3	La conduite d'un réacteur batch multifluide . . . . .	66
1.2	La commande répartie . . . . .	67
1.3	La commande hiérarchisée . . . . .	68
1.4	Conclusion . . . . .	69
2	L'architecture du système de commande/supervision . . . . .	69
3	Supervision des procédés hybrides . . . . .	71
3.1	Supervision en boucle ouverte . . . . .	72
3.2	Supervision en boucle fermée . . . . .	75
3.3	Nécessité d'un modèle hybride pour la supervision . . . . .	78
3.3.1	Aspect continu et commande . . . . .	78

3.3.2	Aspect continu et surveillance . . . . .	79
3.3.3	Détermination de la fenêtre temporelle . . . . .	80
4	Bilan de l'approche . . . . .	82
<b>5</b>	<b>Un modèle hybride pour la supervision</b>	<b>83</b>
1	Un modèle hybride du procédé . . . . .	83
2	Un exemple de modèle de référence hybride: une unité de stockage . . . . .	85
3	Les règles d'évolution . . . . .	88
3.1	Exécution d'une seule recette . . . . .	88
3.2	Exécution de plusieurs recettes en parallèle . . . . .	91
4	Une détection affinée . . . . .	94
5	Emulation du modèle de référence . . . . .	99
6	Le modèle de référence au sein du système de commande/supervision . . . . .	101
7	Résumé des interactions discret-continu . . . . .	102
8	Conclusion . . . . .	104
<b>6</b>	<b>Les détails de la méthode</b>	<b>105</b>
1	Le joueur de réseaux de Petri . . . . .	105
2	La partie commande . . . . .	106
2.1	Homogénéité de la modélisation . . . . .	106
2.2	La communication entre les niveaux . . . . .	107
2.3	La communication entre la partie commande et le modèle de référence . . . . .	108
2.4	La mémorisation du parcours du lot traité . . . . .	110
3	Le modèle de référence hybride . . . . .	112
3.1	Le modèle à événements discrets utilisé . . . . .	112
3.2	La mémorisation des activités exécutées sur l'unité . . . . .	112
4	Le générateur d'événements . . . . .	113
4.1	La notion d'événement . . . . .	113
4.2	L'algorithme du générateur . . . . .	115
5	Conclusion . . . . .	115
<b>Troisième Partie: Exemples d'applications</b>		<b>117</b>
<b>7</b>	<b>La conduite d'un procédé de distillation</b>	<b>119</b>
1	La modélisation au niveau supervision . . . . .	119
1.1	La recette et les ressources . . . . .	120
1.2	Les modèles de référence des unités . . . . .	121
1.3	Interactions entre la recette et les modèles de référence . . . . .	124
2	La modélisation au niveau coordination . . . . .	126
2.1	Les fonctions et les ressources . . . . .	126
2.2	Les modèles de référence des appareils . . . . .	128
2.3	Interactions entre les fonctions et les modèles de référence . . . . .	130
3	Interactions entre les niveaux . . . . .	130

4	Conclusion . . . . .	132
<b>8</b>	<b>Commande et supervision de la conduite thermique d'un réacteur</b>	<b>133</b>
1	La modélisation . . . . .	133
1.1	La recette de l'opération . . . . .	133
1.2	Les fonctions . . . . .	136
1.2.1	Les fonctions d'application et de retrait de fluide . . . . .	136
1.2.2	La fonction de purge . . . . .	138
1.2.3	La fonction de commutation . . . . .	140
1.2.4	La fonction de conduite thermique . . . . .	140
2	La simulation . . . . .	144
3	Conclusion . . . . .	144
	<b>Conclusion générale</b>	<b>147</b>
	<b>Bibliographie</b>	<b>151</b>
	<b>Annexe</b>	<b>162</b>

# Introduction

Parallèlement à une production de masse et standardisée, est apparue une production diversifiée en petites ou moyennes séries qui dépend des changements conjoncturels du marché, traduits par des évolutions de la demande en quantité et nature des produits. Dans le domaine de la chimie par exemple, la tendance est de mettre sur le marché des produits spécialisés, chimie fine très spécifique et à haute valeur ajoutée, dont les quantités demandées sont faibles. Ces faibles quantités et surtout les variantes des produits concernés nécessitent des choix technologiques pour lesquels les réactions se déroulent dans des processus travaillant sur des campagnes de lots [Vanbergen 94] : ce sont les systèmes de production à traitements par lots, appelés couramment «procédés batch».

Le système de production doit par conséquent s'adapter rapidement à des campagnes de produits nouveaux. Pour cela, l'atelier doit être flexible, voire polyvalent. En effet, ces nouvelles fabrications impliquent la plupart du temps une nouvelle configuration de l'atelier, et par ailleurs, un même appareil peut être amené à fonctionner avec des règles opératoires qui diffèrent d'un lot à l'autre.

Dans ce type de production, le premier objectif devient la constance de qualité du produit. Assurer une qualité constante se traduit généralement par un gain de productivité dans ces processus de fabrication pour lesquels les écarts de conduite ont souvent des conséquences irréversibles pour le produit [Vanbergen 94]. L'automatisation et l'informatisation apportent, entre autres, la reproductibilité nécessaire à cette constance de qualité du produit, ainsi que la traçabilité de la production (archivage des données techniques) qui est imposée pour la certification qualité ISO 9002, dont l'obtention est devenue pratiquement obligatoire.

L'intérêt porté aux procédés batch, initialement par la communauté scientifique universitaire, est apparu de manière conséquente dans les années 70 [Rippin 93]. Dès les années 80, des organisations internationales ont publié des recommandations concernant le développement de la régulation par lots, définissant ainsi ce qu'il semble possible de qualifier aujourd'hui de «norme», dont l'impact est important aussi bien sur les travaux de la communauté universitaire que sur ceux de la communauté industrielle.

En comparant les besoins de l'industrie batch en 1983 [Mehta 83] et le bilan dressé en

1994 [Liu & Macchietto 94], il est possible de constater que de nombreux progrès ont été apportés au niveau de la maîtrise des opérations unitaires, au niveau de l'ordonnancement, ainsi qu'au niveau technologique permettant une meilleure intégration des différentes fonctions du système de production. Cependant, les systèmes développés souffrent de l'hétérogénéité des outils et des modèles utilisés. Ceci est la conséquence d'une démarche consistant à traiter les problèmes séparément sans considération de l'aspect global [Puigjaner *et al.* 94]. A titre d'exemple, nous pouvons reprendre un constat évoqué dans [Macchietto 93] sur le lien «réactif» qu'il reste encore à établir, entre l'ordonnancement hors temps-réel et sa mise en œuvre temps-réel (mise en œuvre impliquant des ajustements).

Historiquement, les besoins d'automatisation des procédés batch ont principalement fait appel à des solutions sur mesure, avec des coûts de développement et de maintenance élevés. L'avis est maintenant quasi unanime sur la nécessité d'une approche globale de l'informatisation de la production en incluant, en organisant et en hiérarchisant toutes les fonctions de l'exploitation, de la planification jusqu'au contrôle des processus [Destenave 91]. L'objectif, très ambitieux, est d'avoir un système de conduite rapidement et facilement adaptable aux besoins de production.

Nos travaux s'inscrivent plus particulièrement dans le cadre des aspects temps-réel de l'automatisation des procédés batch. Cette dernière est rendue difficile par leurs spécificités de fonctionnement au sein duquel coopèrent des aspects continus et discrets, faisant donc appel à la fois aux techniques d'automatisation séquentielle<sup>1</sup> et continue<sup>2</sup>. Cette cohabitation de deux mondes différents que sont le continu et le discret, nous plonge dans la problématique hybride largement considérée dans la communauté scientifique, notamment au sein du groupe de travail français sur les systèmes dynamiques hybrides de l'AFCEET/SEE<sup>3</sup> depuis la fin des années 80.

Dans le cadre d'une approche globale des aspects temps-réel, prenant en compte les directives de la «norme», nous proposerons un système de commande/supervision hiérarchisé, au sein duquel la problématique hybride est abordée en respectant le formalisme de chaque monde.

La première partie du manuscrit a pour objectif de préciser la problématique des systèmes dynamiques hybrides dont les systèmes batch sont une composante. Nous situerons tout d'abord ces systèmes dans la classification traditionnelle des systèmes de production, en essayant de faire percevoir au lecteur non familier, une image de leurs spécificités. Deux exemples de procédés seront d'ailleurs évoqués afin d'aborder la notion importante de configuration.

Le second chapitre sera consacré à la présentation de la «norme», ainsi qu'à la présen-

---

1. séquence d'événements traduisant des états discrets successifs

2. maîtrise de trajectoires de variables d'état continues

3. Association Française des Sciences et Technologies de l'Information et des Systèmes / Société des Electriciens et des Electroniciens

tation d'exemples de systèmes de commande/supervision développés selon ce guide méthodologique. Nous ouvrirons ensuite une parenthèse sur la problématique de la surveillance qui ne saurait être ignorée lors de la conception d'un système d'automatisation.

Le troisième volet de cette partie portera sur la modélisation des systèmes hybrides, complétant ainsi le cadre de l'étude qu'il nous a semblé nécessaire de définir sous ces trois angles.

La seconde partie du manuscrit est dédiée à la proposition d'une nouvelle approche de la commande et supervision de ces procédés batch. Cette présentation de nos travaux, basés sur l'ensemble des constats établis, débutera par une réflexion sur l'influence de la notion de configuration dynamique du procédé sur la conception du système de commande. Une proposition d'architecture sera ensuite exposée. Cette dernière représente les fondations sur lesquelles va s'appuyer notre conception que nous aborderons par construction progressive. Ainsi, le passage d'une supervision en «boucle ouverte» à une supervision en «boucle fermée» nous amènera à l'introduction d'une interface appelée *Générateur d'événements*. Puis, après en avoir évoqué la nécessité, nous proposerons un modèle hybride pour la supervision.

Ce modèle hybride fera alors l'objet d'un chapitre à travers lequel nous détaillerons notre structuration du modèle défini pour la supervision, dont nous préciserons les règles d'évolution assurant la cohérence du modèle. Nous résumerons ensuite les interactions entre les divers éléments constitutifs du modèle.

Le dernier volet de cette partie, abordera les détails de la méthode, notamment par des considérations de mise en œuvre.

Enfin, la troisième et dernière partie du manuscrit présentera deux exemples d'application de nos travaux. Ces deux exemples sont complémentaires : l'un abordera la conduite d'un procédé de distillation en se plaçant à un niveau d'abstraction élevé, l'autre la conduite d'une opération de transformation sur un réacteur batch exothermique.



# Première Partie

## Cadre de l'étude





# Chapitre 1

## Les procédés discontinus

L'objectif de ce chapitre est de décrire le contexte dans lequel s'inscrit l'approche de la commande/supervision que nous allons proposer. Les procédés discontinus constituent un domaine dont les spécificités sont nombreuses mais peuvent être distinguées sous deux composantes : le procédé et les acteurs [Destenave 91]. Ces acteurs sont de cultures différentes, les chimistes, les technologues, les opérateurs «procédé», etc., et leur coopération est difficile mais aboutit à la définition complète du produit et de sa production. L'opérateur «procédé» est un acteur à part entière de la production, il est un élément nécessaire à la conduite et au suivi de la production. Il ne serait pas réaliste de vouloir concevoir un système de commande/supervision excluant totalement l'opérateur de la boucle de commande/supervision. Nos travaux n'abordent pas l'interaction entre l'opérateur et le système de conduite automatisée et ne prétendent en aucun cas substituer au rôle de l'opérateur un traitement totalement automatique.

Nous avons donc choisi de nous concentrer sur le procédé et ses caractéristiques intrinsèques. A travers ce chapitre nous allons tout d'abord situer cette catégorie de procédé dans la classification traditionnelle, puis décrire plus en détail certaines de leurs spécificités en comparaison avec les procédés manufacturiers. Cette comparaison repose sur l'idée qu'il semble possible de capitaliser un savoir issu des recherches menées sur les systèmes dits à événements discrets (SED).

### 1 Classification des systèmes de production

Un système de production a pour objectif de transformer sous certaines conditions un élément initial en un élément final, afin que sa valeur sous certains critères soit augmentée. Cette définition donnée dans [de Bonneval 93], basée sur les notions de transformations et de valeur ajoutée donne une image générale d'un système de production. Tout système de production se base sur l'exploitation d'un procédé par une commande dans le but de réaliser des produits. Cependant, ils appartiennent à différentes catégories dont la classification peut être effectuée selon divers critères dépendant du point de vue selon lequel le système est considéré. Sans vouloir rentrer dans des considérations d'élaboration d'une classification, ou

de définition des critères, cette partie n'a pour but que de situer les systèmes de production discontinus dans la traditionnelle partition entre les systèmes continus et les systèmes à événements discrets.

## 1.1 Les systèmes continus

La connotation de système continu est typiquement issue du traitement de grandeurs comme la température, la pression, les flux, etc., c'est à dire de variables continues au sens où elles prennent leurs valeurs sur l'ensemble des réels lorsque le temps, lui-même continu, évolue [Cassandras 93]. La représentation de l'évolution de ces grandeurs, traduisant le comportement dynamique du procédé, est alors une représentation mathématique à l'aide de modèles continus, différents selon la finalité (statique ou dynamique continu pour l'optimisation, dynamique échantillonné pour la commande autour d'un point de fonctionnement), tels que les équations différentielles, les équations aux différences, les fonctions de transfert, ou encore les équations d'état. Le fonctionnement d'un procédé chimique, par exemple, est représenté, autour d'un régime permanent, sous la forme d'un système d'équations algèbro-différentielles:  $f(x', x, u, t) = 0$  (avec  $x(t)$  le vecteur d'état,  $x'(t)$  la dérivée temporelle de  $x(t)$ ,  $u(t)$  le vecteur des variables de commande, et  $t$  le temps). La complexité de ce type de système de production implique l'emploi de modèles de représentation basés sur la description détaillée des mécanismes physico-chimiques dont le procédé est le siège. Dans la conception de ces modèles la connaissance des phénomènes est souvent une étape limitante.

Les opérations de transformation agissent sur un flot de produit, ou flux de matière, toujours en déplacement : les transferts et transformations sont *ininterrompus dans le temps*. Dans un procédé continu, la fonction de chaque appareil est unique, l'ensemble que forme leur combinaison est dédié à une seule production pendant une période donnée : l'installation est composée de parties *non séparables dans l'espace*. La configuration est alors unique pour un fonctionnement considéré, même si plusieurs produits peuvent être fabriqués parallèlement sur l'installation ; ces différents produits correspondent à des prélèvements à différents stades de la production, c'est-à-dire en plusieurs points de la ligne.

Nous retiendrons comme définition celle d'un système de production à **état continu** au sein duquel le flot de produit continu chemine d'appareil en appareil avec un débit **strictement positif quel que soit le temps** [Pingaud *et al.* 92]. Des systèmes de production de cette nature se retrouvent par exemple dans l'industrie pétrolière et dans la production d'énergie électrique.

Seuls les régimes transitoires (non stationnaires), tels que les phases de démarrage et d'arrêt de l'installation, peuvent induire des changements de configuration, et laisser apparaître la notion d'événement comme par exemple les événements de début et de fin de la phase de démarrage.

Le système continu peut être défini comme un système multivariable dont l'état ins-

tantané est représenté par un vecteur d'état réel de dimension finie [Rakotoson *et al.* 92]. L'état du procédé est habituellement décrit par la valeur de ce vecteur d'état pour les flux d'entrée et de sortie des différents appareils, et des grandeurs intrinsèques (pression, température). La conduite de la production repose alors essentiellement sur des systèmes de régulation qui ont pour rôle de maintenir l'ensemble du procédé dans un état permanent correspondant au point nominal de fonctionnement. Les effets des perturbations sur le système sont compensés par des actions correctrices menées soit sur les consignes des régulateurs soit directement sur les organes du procédé tels que les vannes.

L'exemple de la colonne à distiller continue est une illustration typique. Le liquide à distiller (alimentation) est introduit en permanence dans la colonne. Lors de la distillation (séparation de corps par chauffage), le produit le moins volatil est récupéré au pied de la colonne, et le plus volatil en tête de la colonne. Il y a donc un débit gazeux vers le haut et un débit liquide vers le bas, provoqué en tête de colonne par un reflux<sup>1</sup> qui a pour but de déplacer des équilibres thermodynamiques liquide-vapeur, afin d'enrichir la composition du plus volatil (concentration du produit de tête). La colonne est caractérisée par un ensemble de grandeurs continues telles que les débits d'alimentation, de sortie (produit de tête condensé), de reflux, la puissance de chauffage, etc. Le modèle (ensemble de lois) utilisé pour sa description, dépend de l'objectif : un modèle statique pour calculer le dimensionnement de la colonne, ou un modèle dynamique dans le cadre de la régulation. Pour atteindre le point de fonctionnement fixé et la qualité du produit de tête requise, la régulation va agir simultanément sur l'alimentation, la puissance de chauffe, et le débit de la vanne de reflux afin de compenser l'influence des perturbations.

## 1.2 Les systèmes à événements discrets

Dans la commande des systèmes, il est des situations dans lesquelles la connaissance exacte des variations de ces grandeurs continues (les variables d'état) n'est pas utile. Seules certaines valeurs prises par ces variables présentent un intérêt : ce sont les valeurs significatives comme les seuils, qui permettent de déduire l'état du procédé de manière suffisante pour pouvoir le commander. Prenons l'exemple de l'ascenseur dont la position est définie par une variable continue : l'ascenseur doit parcourir une trajectoire continue dans l'espace pour passer d'un étage à un autre. Une représentation du système par équation d'état permet de connaître à chaque instant, pour une commande donnée, la position occupée (résolution de ces équations). La connaissance des positions correspondant aux étages, donc un ensemble de seuils particuliers sur la trajectoire, peut être suffisante pour commander l'ascenseur d'étage en étage. Cette discrétisation de l'espace transforme le système à état continu en un système à états discrets, et si la transition d'état n'est plus fonction du temps mais dépend de l'occurrence asynchrone d'événements (franchissements de seuils), c'est un système à événements discrets. Il s'agit en fait d'un système représenté à l'aide d'un modèle à événements discrets, qualifié par abus de langage de système à événements

---

1. réinjection dans la colonne du produit de tête condensé.

discrets.

L'état n'est plus caractérisé sur l'ensemble des réels mais sur un ensemble fini de valeurs (l'ensemble des entiers positifs par exemple). L'évolution de l'état est alors considérée à des instants de temps particuliers auxquels correspondent des changements d'états significatifs, appelés **événements**. Le comportement dynamique est plus facile à représenter car le mécanisme de transition d'état est basé sur la logique, cependant sa résolution est plus complexe [Cassandras 93]. Les modèles classiques de représentation de ces systèmes sont les Automates Finis, les StateCharts, le Grafcet, et les réseaux de Petri. D'une façon générale, l'état du système est décrit par un ensemble de variables discrètes traduisant l'état du produit, des ressources, et des activités. Contrairement aux systèmes continus, il est possible d'utiliser un même formalisme à différents stades du développement du système automatisé [Quénech'hdu & Buisson 92] : spécification, conception et réalisation de la commande par exemple.

Cette vision des systèmes est très fréquente dans l'industrie manufacturière. L'installation est constituée de machines de production reliées par des systèmes de transport : l'installation est composée de parties *séparables dans l'espace*. Le produit chemine de machine en machine, et subit des transformations réalisées dans un temps fini : la transformation est *interrompue dans le temps* lorsque le produit est stocké entre deux opérations sur deux machines distinctes. L'évolution de l'état du produit est totalement dépendante des opérations qu'il subit, et cet état reste *invariant* lors du stockage et du transport. Il est par conséquent possible de localiser le produit dans l'espace et dans le temps. Cette localisation s'avère cependant plus difficile dans les productions en grande série où le flot de pièces (produits) est quasiment de nature continue.

La séquence d'opérations effectuée sur un produit est définie par une *gamme*. Cette dernière précise l'ordre dans lequel les opérations doivent être menées en respectant les contraintes de précedence qui garantissent l'obtention du produit final. La *gamme logique* fait abstraction des moyens de production et ne définit que l'ordre des opérations. La prise en compte des ressources de production de l'installation pour la réalisation de la gamme logique donne lieu à la *gamme opératoire*. Une installation flexible, c'est à dire offrant plusieurs cheminements possibles du produit dans l'atelier (flexibilité physique), est facteur de complexité dans la gestion de la production car plusieurs gammes opératoires sont possibles pour la réalisation d'un même produit (flexibilité décisionnelle), et facteur de complexité dans la conduite de la production (flexibilité opérationnelle).

Cette représentation par un modèle à événements discrets d'un système qui peut comprendre des activités de nature continue à durée finie, comme c'est le cas du perçage d'une pièce par exemple, repose sur la possibilité d'appréhender cette continuité par une vision discrète : c'est le principe de l'abstraction. Cependant il est des systèmes de production dont l'aspect continu ne se limite pas à la nature de l'activité mais se retrouve également dans la nature du produit, et qui, de par les problèmes spécifiques posés, ne peuvent être

considérés dans cette catégorie. Il s'agit des systèmes de production discontinus, encore appelés systèmes de production hybrides.

### 1.3 Les systèmes discontinus

Ces systèmes se situent par définition à mi-chemin entre les systèmes continus et les systèmes à événements discrets. Soit les deux types de systèmes sont couplés, la production est continue et il y a un conditionnement discret du produit (fabrication de bouteilles, fabrication de pièces de fonderie, fabrication de fromages, etc.), soit les deux aspects continu et discret cohabitent dans un même système de production où les traitements sont continus mais effectués par lots (les fermentations d'alcool, les transformations en chimie fine), soit le système de production est une combinaison des précédents (fabrication des pots de yaourt). Dans le cadre de nos travaux, nous nous intéresserons plus particulièrement à la notion de traitement par lots.

Cette coexistence des aspects continu et discret au sein d'un même système, conduit à une représentation faisant appel à la fois à des modèles continus et à des modèles à événements discrets. Nous aborderons les différentes approches de la modélisation de l'aspect hybride dans un chapitre ultérieur.

Dans ces systèmes de production, le flot des produits chemine d'appareil en appareil par quantités finies, d'où la notion de traitement par lots. La production discontinue est constituée d'une série d'activités **isolables dans le temps**, les activités sont de durées finies, et **isolables dans l'espace**, les activités peuvent être localisées sur une partie du procédé. L'aspect continu du traitement apparaît au sein de l'activité comme dans le cas des activités de transfert, de régulation ou de transformation. C'est cette notion de processus isolable dans le temps, auquel il est donc possible d'associer un début et une fin, qui confère au système un caractère de système à événements discrets.

La séquence d'opérations à effectuer sur un lot de produit est définie par une **recette**, comparable à la notion de gamme utilisée dans les systèmes manufacturiers. La recette est essentielle, surtout pour les installations multiproduit, car elle définit pour chaque produit non seulement la séquence complète des opérations, mais également l'ensemble des données telles que les valeurs des paramètres à utiliser comme consignes.

De manière comparable aux assemblages dans le manufacturier, la fabrication de produits se fait dans certains cas par le traitement en parallèle de plusieurs matières premières qui viennent se mélanger dans un même réacteur. Le fonctionnement global repose alors sur la synchronisation et la coordination de différentes activités continues, qui constituent un des points sensibles du système de commande [Soto-Fernandez 89] [Bellorin 94].

A partir de cette description du système basée sur les activités, la distinction entre l'aspect continu et l'aspect discret, entre la représentation continue et discrète, est souvent

perçue comme une distinction seulement relative au point de vue selon lequel le système est considéré [Pourcel 91]. Comme nous l'avons évoqué précédemment, la distinction est alors posée en terme de niveau d'abstraction : l'aspect discret étant considéré comme une vision abstraite de l'aspect continu. Nous aurons l'occasion de revenir sur cette considération de l'aspect hybride du système de production discontinu, mais précisons dès à présent que nous considérons plutôt qu'il y a coexistence de deux états dépendants, l'état continu et l'état discret, ces deux visions étant complémentaires et non pas seulement l'abstraction de l'une par l'autre.

Le contexte général étant défini, abordons une étude plus précise des spécificités des procédés à traitements par lots.

## 2 Spécificités des procédés à traitement par lots

La présentation succincte des systèmes de production à traitement par lots, appelés aussi procédés «batch», que nous venons d'effectuer laisse apparaître la possibilité d'utiliser des méthodologies et des modèles développés dans le cadre des systèmes manufacturiers. Cependant une transposition directe n'est pas réalisable et des considérations supplémentaires sont à prendre en compte face aux spécificités de ces procédés [Rakotoson *et al.* 92]. Nous reviendrons sur cette transposition au prochain chapitre, toutefois à travers les caractéristiques du produit, du lot, et du procédé, ainsi qu'à travers des considérations technologiques, illustrons des facteurs de différences entre ces systèmes de production.

### 2.1 Le produit

La nature du produit et la variété de produits à fabriquer influent aussi bien sur la conception du système de production que sur sa gestion ultérieure. La nature continue des produits traités dans les procédés batch contribue largement à leur spécificité [Bellorin 94].

#### 2.1.1 La nature continue

Le produit est *non saisissable*, non identifiable au sens d'une entité atomique élémentaire tel un produit manufacturier : la matière manipulée est sous forme fluide (liquide, gaz). Une conséquence directe de cette nature continue est le couplage d'appareils (appareil source et appareil cible) induit par le transfert de produit. Les appareils restent liés (lien physique établi par les tuyauteries dans lesquelles la matière circule), et synchronisés (synchronisation de la vanne de sortie de l'appareil source avec la vanne d'entrée de l'appareil cible) durant le transfert. L'appareil source et l'appareil cible sont alors totalement dépendants dans cette configuration : *les ressources sont étroitement couplées*. Ce couplage est exprimé dans [Rakotoson *et al.* 92] par la notion de "Macro-Ressource Temporairement Isolable". Ce couplage n'est pas présent dans les procédés manufacturiers où les machines sont indépendantes pendant le transport : dès que la pièce a été chargée sur l'organe de

transport la machine source peut être considérée comme disponible. Nous pouvons déjà noter l'importance de la configuration physique du procédé.

### 2.1.2 La notion de lot

Un lot correspond rarement à une collection d'entités élémentaires (objets, pièces, etc.) mais essentiellement à des quantités finies ou des volumes de matière. Un lot est par exemple défini par un niveau dans une cuve de stockage ou dans un réacteur. Lors d'une transformation ce volume est amené à varier car des produits sont ajoutés ou séparés : la taille du lot est donc variable au cours de la fabrication. Par conséquent, contrairement aux procédés manufacturiers, un lot **n'a pas une image exclusivement discrète** même à un haut niveau d'abstraction, et ne peut pas être décomposé en  $n$  éléments ( $n$  fractions de lot) à un niveau plus bas.

### 2.1.3 L'état du lot

L'état d'un lot est caractérisé par des variables discrètes telles que sa localisation, l'opération en cours, etc., et des variables continues comme la concentration, le pH, la viscosité, par exemple. Lors d'un traitement tous les éléments de matière subissent collectivement et *indistinctement* le même changement d'état, c'est à dire la même évolution. L'état du produit peut être instable au sens où la matière est vivante, évolutive, car elle est soumise à des évolutions internes dues à la réaction entre ses différents constituants. Il peut donc y avoir évolution de l'état du produit au cours du temps indépendamment des opérations de transformations spécifiées par la recette. Cette notion d'instabilité du produit soulève un problème de commandabilité. En effet, contrairement aux produits traités en manufacturier dont l'état est stable dans l'espace et dans le temps à l'image des solides [Rakotoson *et al.* 92], et dont l'état est directement une conséquence de la commande appliquée<sup>2</sup>, l'état du produit continu ne dépend pas seulement de la commande.

Cette évolution «secondaire» du produit est un phénomène courant dans les processus chimiques, les fermentations [Daubas 94] et les réactions exothermiques [Louleh 96] en sont des exemples. Cette caractéristique se retrouve également dans les fonderies de fontes lourdes ; la fonte liquide est une matière difficile à manipuler car ses propriétés changent lors des phases de transport ou d'attente [Pierreval 94].

Cette instabilité du produit implique dans certains cas une gestion différente des ressources. L'attente entre deux opérations peut être une cause de dégradation de la qualité du produit. Différentes stratégies de transfert du produit d'appareil en appareil sont à prendre en considération : stockage intermédiaire illimité, stockage intermédiaire limité, pas de stockage intermédiaire (avec attente nulle ou non), etc.

---

<sup>2</sup> état invariant après l'opération de perçage, fraisage, etc.



### 2.1.4 Les contraintes technologiques

Les contraintes technologiques rencontrées dans ces procédés se situent principalement au niveau de l'observation du procédé.

Les possibilités de mesure et d'analyse en ligne sont limitées, surtout sur les procédés chimiques et biotechnologiques. L'observabilité est donc limitée par la difficulté d'obtenir des informations sur l'état du produit (concentration, viscosité, etc) [Rippin 89]. Soit par manque de capteurs (capteurs non-existants), soit par défaut de robustesse de l'instrumentation, soit à cause de mesures ayant un temps de réponse trop long pour effectuer une adaptation des paramètres de traitement du lot courant : les informations sur le produit sont fréquemment obtenues par prélèvement d'échantillons qui sont ensuite analysés en laboratoire [Destenave 91].

Cette difficulté est parfois compensée par l'utilisation d'estimateurs permettant de «reconstituer» en temps-réel l'évolution des variables d'état du procédé à partir des mesures disponibles, ce sont les capteurs logiciels [Chérut & Flaus 94].

D'autre part, le coût élevé de l'automatisation de certaines opérations, la maîtrise parfois incomplète de certains processus, et la différence entre la technologie du laboratoire et celle de la production (échelles différentes<sup>3</sup>) qui impose une adaptation des paramètres de fonctionnement, laissent une place importante aux interventions manuelles de l'opérateur [Vanbergen 94]. Ces interventions portent souvent sur une modification des paramètres opératoires comme les consignes, et elles sont accomplies dans la plupart des cas d'une manière non procédurale (basée sur la connaissance et l'expérience de l'opérateur). D'où la difficulté supplémentaire de considérer simultanément deux modes de marche, mode manuel et mode automatique.

## 2.2 Le procédé

Considérons le procédé comme étant le dispositif de traitement de la matière. Ce dispositif est composé de deux grandes catégories d'appareillages [Jallut & Gilles 90] :

- les systèmes de traitement physique pour la préparation des matières premières ou la finition du produit, comme les appareils de mélange et de séparation par exemple.
- les systèmes de traitement chimique tels que les réacteurs, au sein desquels ont lieu les transformations. Les réacteurs tubulaires transforment le lot de matière au fur et à mesure de son avancement dans le réacteur. Dans les réacteurs parfaitement agités, le lot entier de réactifs subit la transformation en même temps. Ces derniers peuvent fonctionner en mode fermé (chargement, réaction, puis vidange), ou en mode semi-fermé (chargement du pied de cuve, puis réaction avec une coulée de réactif en parallèle, puis vidange).

---

<sup>3</sup> dans le cas d'un réacteur par exemple, le volume, l'épaisseur de la paroi et la surface d'échange thermique différent.

Les deux exemples que nous allons présenter appartiennent aux deux catégories de systèmes de traitement que nous venons de citer. Ces exemples serviront d'illustration à plusieurs reprises dans le manuscrit.

### 2.2.1 Exemple de procédé de distillation

Le procédé de distillation [Macchietto *et al.* 89], représenté sur la figure 1.1, est constitué de :

- un réservoir contenant le produit à distiller (Feed Tank  $F_1$ ),
- une colonne à distiller (Batch Still  $S_2$ ) comprenant un ensemble de plateaux de séparation et un système de chauffage du contenu,
- un condenseur,
- un réservoir de coupe intermédiaire (Offcut Tank  $T_3$ ),
- un réservoir «intégrateur» de stockage du distillat intermédiaire (Distillate Hold Tank  $T_4$ ) : ce réservoir joue un rôle d'«intégrateur» au sens où il permet de stocker temporairement une certaine quantité de matière dont la concentration finale répond à la spécification. Les variations de concentration instantanées sont ainsi absorbées par ce stockage intermédiaire qui, sur une période donnée, permet d'obtenir une concentration moyenne.
- un réservoir de stockage du distillat final (Final Distillate Hold Tank  $T_5$ ),
- un réservoir de stockage de la solution finale en pied de colonne (Final Bottoms Product Tank  $T_6$ ).

Un lot de produit à distiller est chargé en pied de colonne. La solution considérée dans cet exemple est un mélange binaire de  $A$  et  $B$ . La distillation s'effectue en chauffant la solution qui traverse alors l'ensemble des plateaux sur lesquels s'effectue une rétention de matière. Le constituant le moins volatil,  $B$ , est récupéré en pied de cuve (bas de la colonne), et le plus volatil,  $A$ , en tête (haut de la colonne). Le débit gazeux est liquéfié à travers le condenseur. Le reflux de la solution condensée en tête est fonction de la richesse du courant en  $A$  et constitue un élément de la stratégie de conduite. La conduite est fixée par le suivi de la température en tête, qui est plus facilement mesurable que la concentration, et l'image de la concentration est obtenue par une corrélation directe entre la température et la concentration.

La conduite optimale de la distillation d'un lot (séparation des constituants  $A$  et  $B$ ) implique de changer de stratégie à des instants de temps particuliers qui sont déterminés en fonction de la concentration du produit considéré.

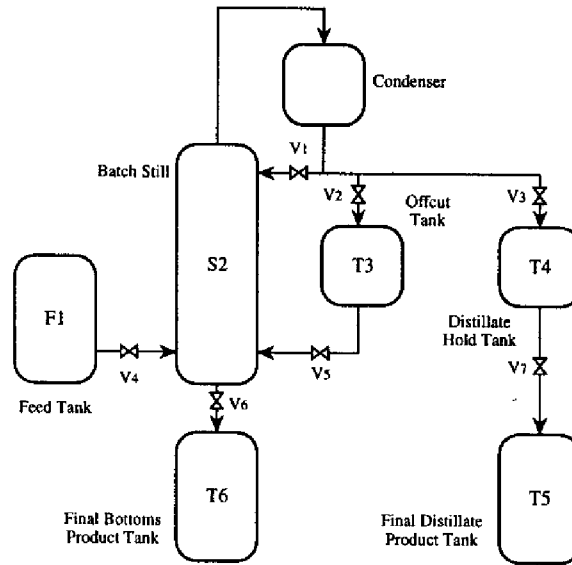


FIG. 1.1 - Exemple de colonne à distiller batch [Macchietto et al. 89]

### 2.2.2 Exemple de réacteur exothermique

La conduite des réacteurs exothermiques discontinus, fréquents dans le domaine de la chimie fine ou pharmaceutique, se ramène dans la majorité des cas à un problème de conduite thermique car la mesure de température du milieu réactionnel est quasiment la seule variable facilement mesurable dans ces transformations complexes [Louleh 96]. Le suivi rigoureux du profil de température donné (parfois couplé à un profil de coulée) est important pour la maîtrise de la transformation dont dépend la qualité du produit.

Dans l'exemple considéré, la température du milieu réactionnel est contrôlée par la circulation d'un fluide plus ou moins chaud ou froid au sein d'une double-enveloppe (double-paroi). Il existe différents systèmes de conduite thermique, dits systèmes de chauffage/-refroidissement, dont : le système monofluide consistant à faire circuler un fluide caloporteur unique à débit constant avec action sur la température (chauffage de l'eau circulant dans la double enveloppe du réacteur par mélange de vapeur, ou par échangeur de chaleur), et le système multifluide consistant à alterner des fluides, à différents niveaux énergétiques, à température fixée avec action sur le débit.

Le système de conduite présenté<sup>4</sup>, décrit sur la figure 1.2, est une combinaison des deux précédents : il est basé sur l'utilisation de plusieurs fluides à température fixée (eau glycolée, vapeur sèche) et d'un fluide, appelé fluide intermédiaire, dont la température est réglable (vapeur injectée dans l'eau du réseau).

4. unité pilote de l'Institut du Génie des Procédés à l'ENSIGC de Toulouse.

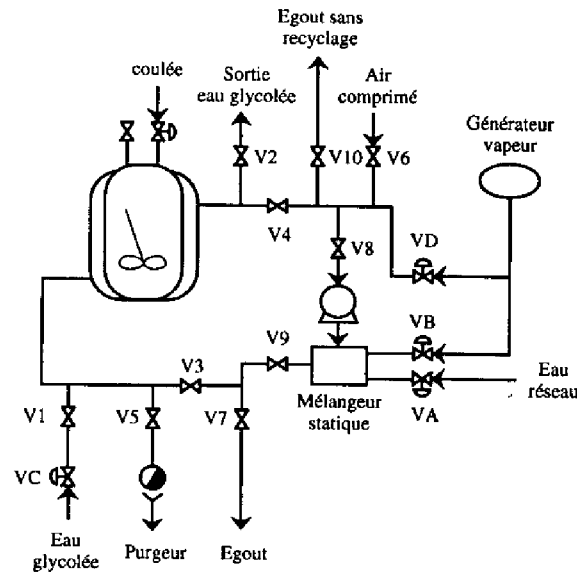


FIG. 1.2 – Le réacteur batch multifluide considéré

Un des facteurs de complexité de ce type de système de conduite thermique repose sur la gestion dynamique des différentes configurations issues du changement de fluide, la détermination d'une commande dépendant de la configuration, et l'étroite interaction entre le système de chauffage/refroidissement et celui de la coulée (les deux opérations se déroulant en parallèle). L'ensemble constitué des vannes deux états (les vannes discrètes  $V_1$ - $V_{10}$ ) permet d'établir la configuration requise, et l'ensemble des vannes à ouverture réglable (les vannes continues  $V_A$ - $V_D$ ) permet d'agir sur le flux thermique appliqué.

### 2.2.3 La configuration du procédé

Les appareils continus<sup>5</sup> sont principalement utilisés pour des productions en grande quantité (fonctionnement en régime permanent) ou celles dont les transitoires de démarrage et d'arrêt sont longs. Les appareils discontinus<sup>6</sup> sont plutôt dédiés aux petites productions à haute valeur ajoutée et peuvent offrir une flexibilité de traitement en supportant différentes transformations. Ces appareils sont disposés dans différents types de configuration, c'est à dire dans différents agencements, établissant ainsi le diagramme des flux de l'installation (flowsheet). La flexibilité de l'installation est étroitement liée à la nature des connexions entre les appareils, ces connexions pouvant être figées ou configurables. Aux différents types de configurations correspondent différentes classes d'ateliers : disposition en série, en parallèle, ou mixte (plusieurs étages en parallèle), un atelier étant également caractérisé

5. fonctionnement tel que les flux d'entrée/sortie sont  $> 0$ .

6. fonctionnement tel que les flux d'entrée/sortie sont  $\geq 0$ .

par la politique de stockage. Les grandes classes sont les suivantes<sup>7</sup> :

- Installation monoproduit / monoflux  
Un seul type de produit est traité sur une ligne de fabrication figée, et un seul lot est fabriqué à un instant donné. La ligne peut être gérée comme une seule ressource.
- Installation multiproduit / monoflux  
La ligne de fabrication comporte plusieurs unités (appareils) en série, et permet la fabrication de plusieurs lots simultanément. Les nuances de fabrication sont obtenues par des paramètres différents. Le cheminement du produit reste cependant le même.
- Installation multiproduit / multiflux (installation flexible)  
La ligne de production, composée d'unités en série et/ou en parallèle, est configurée de manière dynamique, il y a indépendance entre le produit et la ligne de fabrication. Des lots différents peuvent être traités en parallèle, cependant ils appartiennent à une même famille. La fabrication simultanée ou alternée de produits différents introduit des contraintes de nettoyage des appareils.
- Installation polyvalente  
La différence avec l'installation précédente porte sur la diversité de produits réalisables. Cette installation est composée d'appareils permettant d'effectuer un ensemble de transformations élémentaires de génie chimique. La flexibilité sur les appareils et sur leurs combinaisons crée cette polyvalence.

Un aspect important du procédé repose donc sur la notion de **configuration dynamique**. La flexibilité du procédé, reconfigurable par un système de tuyauteries et de vannes de commutation (vannes deux états dédiées), induit des flowsheet différents avec un couplage dynamique des appareils. Si l'utilisation de l'installation est rendue souple, en contrepartie la gestion des différentes configurations rend le système de commande plus complexe.

Deux autres caractéristiques intrinsèques de ces procédés sont l'importance du stockage intermédiaire et les différentes classes de ressources.

- Les ressources  
Le procédé est composé de différents types de ressources. Les ressources «discrètes» correspondent à la plupart des appareils : il est possible d'établir une distinction entre des ressources «élémentaires» pouvant effectuer une seule et unique activité à un instant donné (un réacteur en mode fermé), et des ressources «complexes» pouvant supporter plusieurs activités en parallèle (un réacteur en mode semi-fermé subissant plusieurs chargements de réactifs en parallèle). Les ressources «continues» à capacité

---

7. synthèse effectuée à partir de [Soto-Fernandez 89] et [Jost 92].

limitée correspondent aux sources énergétiques comme la vapeur, ou les échangeurs de chaleur, par exemple. Elles sont difficiles à gérer car la capacité utilisée par une activité peut varier selon les caractéristiques de l'activité (sa durée) et celles du lot (son volume) [Kondili *et al.* 93].

- Le stockage intermédiaire

Entre ces deux types de ressources se trouvent les stocks intermédiaires.

Le stockage intermédiaire est un élément important dans ce type de production car la rétention de matière joue un effet tampon. Elle permet de découpler deux traitements continus, il s'agit d'un découplage temporel partiel entre les activités antérieures et postérieures. Il est également possible de redimensionner les lots de produit entre deux traitements «discrets», ainsi que d'absorber certaines perturbations dont celles induites par les variations de temps dans ces traitements [Yi & Reklaitis 94].

Les stocks intermédiaires doivent être considérés à la fois selon leur état discret, "*le stock est-il disponible (libre) pour une opération de transfert ?*", et selon leur état continu, "*l'évolution du volume dans le stock permet-elle d'accueillir le lot qui doit être transféré ?*". Il s'agit d'ailleurs d'un point sur lequel nous reviendrons lors de la conception de notre système de supervision/commande.

Nous avons évoqué les contraintes de nettoyage des appareils induites par la fabrication simultanée ou alternée de produits différents, effectuées pour des raisons d'hygiène, de qualité du produit, ou de sécurité. Cet aspect amène à considérer différemment l'état des ressources : une ressource n'est plus seulement libre ou occupée, mais également «sale» ou «propre» par exemple. L'affectation de ressources aux opérations est alors plus complexe : "*si la ressource est libre, est-elle «sale» ?*", "*si oui, y-a-t-il incompatibilité entre le produit précédent et le produit courant ?*". Cette autre caractéristique de l'état discret d'une ressource est à considérer sur les appareils ainsi que sur les connexions, et à gérer aussi bien lors de l'ordonnancement que lors de la conduite temps-réel [Barbosa Póvoa & Macchietto 94].

### 2.3 Le processus

Le processus peut être défini comme un ensemble de phénomènes organisés dans le temps et dans l'espace. Nous étudierons les directives de structuration dont il est l'objet au chapitre suivant, mais certaines particularités, que nous allons énumérer, sont à préciser.

- La préemption et l'irréversibilité

La préemption, c'est à dire l'interruptibilité de la fabrication du produit, est rare, tout au moins dans l'industrie chimique. Le processus de fabrication est d'ailleurs fréquemment irréversible. Le lot de produit «non conforme» peut dans certains cas être déclassé dans une catégorie inférieure, ou subir un recyclage permettant d'éviter le rejet complet du lot.

- Le recyclage

Comme nous venons de l'évoquer, il est parfois possible d'effectuer des opérations de recyclage pour la récupération de lots dont la qualité n'est pas conforme à celle spécifiée. Cependant ce recyclage, dont le principe est la récupération de certains constituants du produit, va modifier les caractéristiques des ingrédients initiaux. En effet, la séparation permettant le recyclage est rarement complète (rarement parfaite), et le flux de matière recyclé étant dirigé vers les stocks de matières premières, leurs compositions sont altérées.

- Les perturbations

Le processus est sensible aux perturbations de l'environnement comme la température ambiante, le taux hygrométrique, la variation de température de l'eau du réseau en hiver et en été, etc. Il est également sensible aux perturbations induites par la fluctuation des caractéristiques des ingrédients (qualité des matières premières). Ces perturbations compliquent la conduite du processus car elles impliquent de fréquentes actions correctrices comme l'ajustement des paramètres d'une opération, voire de ceux de la recette, ces ajustements pouvant se traduire par exemple par la modification des durées opératoires [Puigjaner *et al.* 94].

- Les opérations de nettoyage

Le nettoyage de parties de l'installation implique l'existence d'appareils dédiés (par exemple un réservoir contenant le détergent, avec vannes, pompe, et système de chauffe), et connectés à l'ensemble de tuyauteries de l'installation. Les opérations de nettoyage peuvent être définies en fonction de la fréquence d'utilisation des appareils, et elles utilisent des ressources partagées dont les sources énergétiques, un sous ensemble de connexions (tuyauteries), et bien sûr l'appareil(s) à nettoyer. Par conséquent elles peuvent «bloquer» le procédé dans une configuration déterminée pendant une certaine durée.

La prise en compte de ces opérations au niveau de l'ordonnancement s'avère donc indispensable [Pantelides 93], et selon la stratégie d'exécution en temps-réel il est possible de réduire considérablement leur durée [Liu & Macchietto 94].

- L'aspect discret au sein de traitements continus

L'aspect discret est seulement apparu jusque là à travers les activités et les configurations du procédé. Toutefois, le contrôle de trajectoires de variables continues fait également apparaître des étapes discrètes [Jallut & Gilles 90]. Considérons par exemple le suivi d'une trajectoire de température imposé par une opération de transformation dans un réacteur discontinu en mode semi-fermé. Cette opération est composée de plusieurs étapes discrètes : l'étape initiale d'asservissement pour le préchauffage de la charge, l'étape de régulation pendant la réaction et la coulée, et l'étape de refroidissement du milieu réactionnel après la transformation.

Si le système de conduite thermique du réacteur discontinu est un système multifluide, comme l'exemple de réacteur présenté précédemment, les différentes étapes dans la conduite s'effectuent par définition à l'aide de plusieurs fluides caloporteurs. Par conséquent, il est nécessaire de gérer les changements de configuration du système de conduite thermique induits par les commutations de fluides. Nous illustrerons plus en détail cet aspect dans la prochaine partie.

- Le couplage dynamique de boucles de régulation

Le couplage de boucles de régulation est différent selon la stratégie de conduite du procédé. L'optimisation d'une opération «joue» sur ce couplage d'un ensemble de variables continues. Le cas de la distillation est typique. Les différentes stratégies de conduite de l'unité de distillation batch présentée, exposées dans [Macchietto *et al.* 89], reposent sur différents couplages des variables continues. La conduite optimale de la distillation est alors basée sur le changement de stratégie à des instants particuliers pendant la production.

### 3 Conclusion

Contrairement aux procédés continus, les procédés discontinus présentent des changements fréquents de modes de fonctionnement, et de configurations des unités qui induisent une grande interactivité entre les automatismes analogiques et logiques. Par rapport aux procédés manufacturiers, la nature continue du produit amène à poser différemment les problèmes de gestion et de conduite. Toutefois, le caractère de système à événements discrets conféré au système de conduite par la possibilité d'une décomposition en processus élémentaires isolables dans le temps et dans l'espace, devrait permettre de s'appuyer sur les travaux menés dans le cadre des procédés manufacturiers.

Nous allons maintenant aborder la commande et supervision de ces procédés, à travers les propositions de «normes» issues d'organisations professionnelles internationales, et différents travaux menés sur la conduite d'unités pilotes, afin d'en dégager les idées directrices.





## Chapitre 2

# La commande et la supervision des procédés à traitement par lots

Des organisations professionnelles telles que NAMUR<sup>1</sup>, ISA<sup>2</sup>, EBF<sup>3</sup>, impliquant utilisateurs et constructeurs, développent des guides méthodologiques sur la conception d'applications d'automatisation des procédés batchs (terminologie, modèles, structuration du système) [ISA/SP88 94] [Comitee 90].

Ces diverses références en cours de développement au niveau international ont pour objectif de définir ce qu'il est possible d'appeler une «norme». Il y a en effet une convergence entre les travaux de ces différents groupes, pour essayer d'établir un même ensemble de normes internationales sur le point de devenir l'IEC 1512-1<sup>4</sup>.

Ces travaux ont un impact important aussi bien sur la communauté universitaire que sur la communauté industrielle, au sens où ils servent effectivement de point de départ dans la conception des systèmes de commande/supervision. Toutefois, ils ne couvrent pas tous les champs de la production batch avec autant de précision, et restent ouverts par exemple au niveau de l'ordonnancement, de la gestion des ressources physiques et énergétiques, et de l'expression des sécurités (interblocages empêchant l'accès à des configurations interdites par exemple). Ainsi, certains industriels définissent parallèlement leurs propres méthodologies, en se basant sur celle proposée par ISA/SP88, et en essayant de combler ces «lacunes». C'est le cas de Rhône-Poulenc qui a développé la méthode ASTRID [Lebourgeois 96].

La première partie de ce chapitre est dédiée à la présentation de cet ensemble de normes, en donnant quelques exemples de systèmes de commande/supervision développés sur des unités pilotes, ou commercialisés. Nous aborderons ensuite la problématique de la surveillance qui n'est pas évoquée dans les travaux précédents, et qui pourtant ne saurait être ignorée lors de la conception d'un système de commande/supervision [Combacau 91].

- 
1. Normenarbeitsgemeinschaft für Mess Und Regeltechnik : comité allemand.
  2. Instrument Society of America, Standards Committee: comité américain.
  3. European Batch Forum : comité européen.
  4. International Electrotechnical Commission.

## 1 La «norme» pour la conception du système de commande/supervision

Nous ne présentons<sup>5</sup> que certains aspects de la «norme» ISA/SP88, ceux concernant la structuration du système de conduite de l'installation. Ainsi, plusieurs «vues hiérarchisées» sont proposées : selon l'équipement, selon les recettes, selon l'aspect fonctionnel, et selon l'aspect procédural. Cette description de l'organisation des fonctions du système de production et de la structuration de la commande, tend vers la définition d'un modèle CIPO (Computer Integrated Process Operation) qui s'apparente au modèle CIM (Computer Integrated Manufacturing).

### 1.1 La hiérarchisation selon l'équipement

Les différents niveaux normalisés, organisés sous la forme d'une hiérarchie, sont les suivants (Figure 2.1).

- *Elément*: un élément est une partie de l'équipement qui remplit une fonction de mesure ou de manipulation de variables du processus, tels que les capteurs et actionneurs.
- *Module de contrôle*: combinaison d'éléments qui a un comportement continu et/ou discret (par exemple une vanne deux états formée d'un actionneur pneumatique et d'interrupteurs de fin de course).
- *Module d'équipement*: groupe fonctionnel de modules de contrôle et d'éléments pouvant effectuer un nombre fini de fonctions spécifiques (par exemple un système de dosage). Un module d'équipement peut être associé à un appareil comme par exemple le système de conduite thermique associé à un réacteur.
- *Unité*: réunion de modules d'équipement, de composants et d'éléments afin de réaliser une opération coordonnée (un réacteur et son système de conduite thermique par exemple). Une unité de traitement peut réaliser des opérations différentes mais en ne traitant qu'un seul lot à un instant donné.
- *Ligne de production*: ensemble d'une ou plusieurs unités organisées en série ou en parallèle pour la production d'un lot.
- *Atelier*<sup>6</sup>: ensemble de plusieurs lignes de production qui peuvent être couplées par des ressources communes comme une source d'énergie ou une cuve de stockage particulière par exemple.
- Et enfin aux plus hauts niveaux, l'*Usine* et l'*Entreprise*.

---

5. synthèse effectuée à partir de [Bellorin 94] et [ISA/SP88 94].

6. appelé également installation dans le manuscrit.

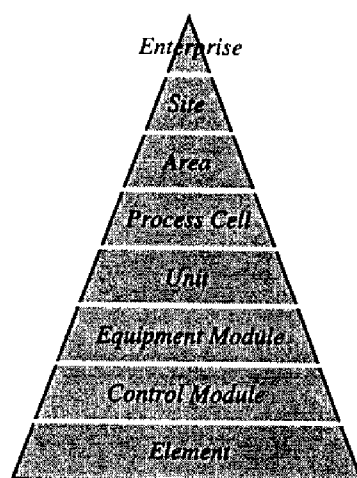


FIG. 2.1 – Les différents niveaux d'équipement (source ISA-dS88)

## 1.2 La hiérarchisation selon les recettes

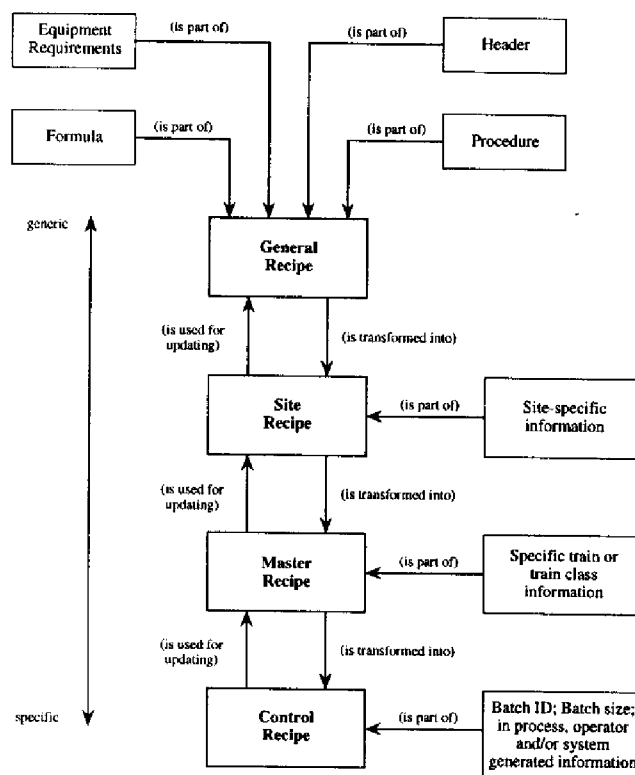
Le terme de *recette* est très explicite : “fabriquer un lot de produit chimique est similaire à préparer un plat” [Sawyer 92]. En effet, dans les deux cas une recette est utilisée pour spécifier :

- le produit à réaliser (le titre),
- le type d'équipement requis,
- la procédure de traitement (séquence des opérations à effectuer),
- et les ingrédients, quantités, paramètres et temps de fabrication, et la formule assurant la fabrication d'une quantité désirée de produit.

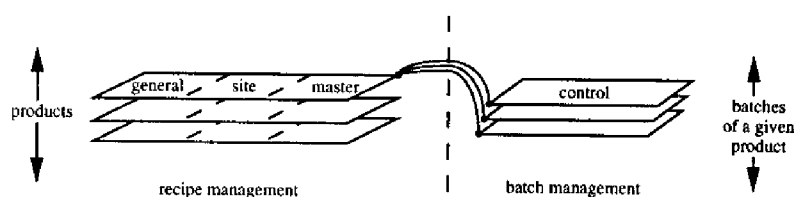
Plusieurs types de recettes sont utilisés, de la recette générique transportable, à la recette de contrôle dédiée à la fabrication d'un lot précis sur un atelier précis (Figure 2.2).

- *Recette de base* : recette générique contenant le jeu d'informations utiles pour l'automatisation de la fabrication d'un produit (les paramètres de la formule de fabrication du produit sont définis par des proportions). Elle est transportable puisqu'elle n'est liée ni à une installation particulière, ni à des équipements particuliers.
- *Recette de site* : recette adaptée au site, avec les caractéristiques précises des équipements et des ingrédients requis. Le lot étant dimensionné, elle est utilisée comme une entrée du système d'ordonnancement.
- *Recette principale* : recette de site adaptée à une campagne particulière (ordonnancement), l'équipement est décrit par classes (caractéristiques comme la capacité), et le lien avec l'installation est virtuel (aucun appareil particulier n'est désigné).

- *Recette de contrôle*: recette appliquée à un lot particulier auquel elle est associée pendant tout son processus de fabrication. Elle correspond à l'exécution temps-réel de la recette principale avec une affectation ou désignation des équipements effectifs.



2.2.1: Les niveaux de recette



2.2.2: recette/produit et recette/lot

FIG. 2.2 – Les différentes recettes (source ISA-dS88)

La notion de recette est comparable à la notion de gamme dans les procédés manufacturiers. La recette de base et la recette principale peuvent être respectivement comparées

à la gamme logique et à la gamme opératoire, la première fait abstraction des moyens de production disponibles dans l'atelier et la seconde intègre les ressources physiques nécessaires à la réalisation des opérations.

Ces différentes recettes sont utilisées à différents niveaux : la *recette de site* sert à la planification, la *recette principale* est issue de l'ordonnancement, et enfin la *recette de contrôle* sert à la conduite temps-réel de la production.

### 1.3 La hiérarchisation fonctionnelle

Comme tout système de conduite, celui des procédés batch implique un ensemble d'activités ou de fonctions. Ces fonctions sont hiérarchisées de la planification au contrôle temps-réel de la production (Figure 2.3).

Les échanges entre les différentes fonctions sont verticaux, un niveau supérieur transmet des ordres (ou requêtes) au niveau immédiatement inférieur qui lui retourne des comptes-rendus, et horizontaux pour l'échange d'informations entre des fonctions de même niveau.

- Planification/Ordonnancement ("Production Planning and Scheduling") : planification de la production et ordonnancement des lots à produire.
- Gestion des informations sur la production ("Production Information Management") : ensemble de fonctions pour la collecte d'informations sur l'état de la production, et pour la gestion de ces informations afin de constituer, entre autres, l'historique de chaque lot fabriqué (archivage).
- Gestion des recettes ("Recipe Management") : gestion des différentes recettes, recette générale et recette de site, pour parvenir à la génération de la recette principale.
- Gestion du procédé ("Process Management") : ensemble de fonctions chargées de l'exécution de la recette de contrôle générée à partir de la recette principale, de la gestion des lots en cours de fabrication, et de l'affectation temps-réel des ressources.
- Supervision ("Unit Supervision") : supervision et coordination de l'ensemble des opérations à exécuter sur le procédé pour la fabrication des lots, et actualisation des informations sur l'état des opérations.
- Commande ("Process Control") : ensemble de fonctions pour la mise en œuvre des ordres issus de la supervision pour la réalisation des opérations, et pour l'actualisation des informations sur l'état du procédé et des traitements.

Tous les niveaux ont une interaction avec celui de Gestion des informations (Figure 2.3), ces interactions n'étant d'ailleurs pas toutes clairement explicitées sur la représentation initiale donnée par la norme, nous les avons ajoutées (traits pointillés).

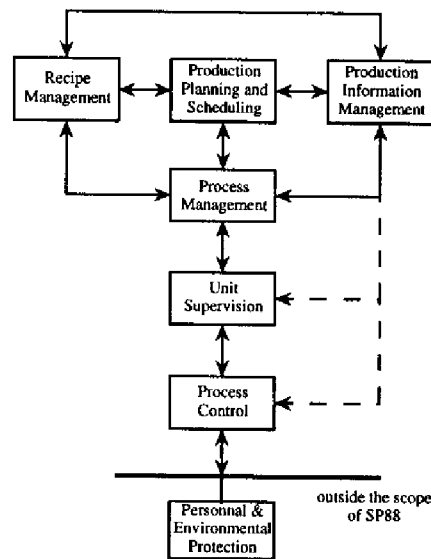


FIG. 2.3 – Les différents niveaux fonctionnels (source ISA-dS88)

#### 1.4 La hiérarchisation de l'aspect procédural de la recette

La procédure d'une recette peut être décomposée en opérations qui peuvent elles-mêmes être décomposées en phases, etc., cette décomposition successive ayant pour plus petit élément l'instruction (Figure 2.4).

- *Instruction*: commande élémentaire orientée équipement adressée à un élément ou à un composant afin qu'il effectue un traitement donné (par exemple ouvrir une vanne).
- *Pas*: séquence temporelle ou événementielle d'instructions à exécuter dans un module d'équipement en vue d'accomplir une tâche spécifique orientée processus (exemples : préchauffer à  $85^{\circ}\text{C}$ , agiter pendant 10 mn).
- *Phase*: succession de pas permettant de réaliser une fonction élémentaire comme le chargement d'un réacteur avec une matière première par exemple. Une phase est limitée par des frontières définies par les instants où le déroulement des pas peut être interrompu en toute sécurité pour l'installation et sans dommage pour le produit.
- *Opération*: ensemble de phases exécutées en séquence ou en parallèle dans une unité de production. Une opération conduit le produit d'un état à un autre, induisant une modification des propriétés physiques et/ou chimiques de celui-ci (exemples : la réaction, la séparation).
- *Procédure*: ensemble d'opérations à exécuter dans une ligne de production, pour fabriquer un produit particulier. Une procédure correspond à une stratégie générique de fabrication d'un lot.

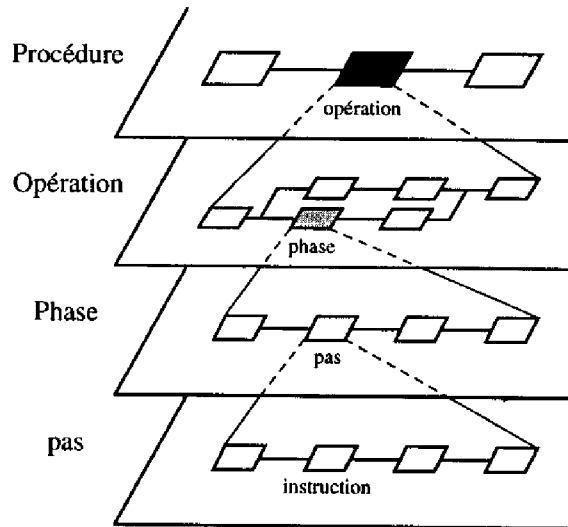


FIG. 2.4 – La décomposition de l'aspect procédural

## 2 Exemples de systèmes de commande/supervision

Conformément à la hiérarchisation recommandée par la «norme» ISA/SP88 pour faire face à la complexité du procédé à piloter, il y a un consensus sur une décomposition fonctionnelle hiérarchisée du système de production chez les universitaires [Cott & Macchietto 88], [Puigjaner *et al.* 94], et chez les industriels [CM 95].

La tendance est alors une approche généralement ascendante au niveau de l'équipement, c'est-à-dire basée sur la définition d'un ensemble d'opérations exécutables sur l'installation, et descendante au niveau des recettes jusqu'à la définition d'une recette intégrant les ressources et les contraintes de l'installation. La recette est «*la clef*» du système de conduite<sup>7</sup> [Engell & Wöllhaf 94], et le principe d'*intégration* est l'idée dominante [Macchietto *et al.* 92] [Venkatasubramanian & Stanley 93].

Ce principe d'intégration repose à la fois sur «l'assemblage» de fonctions ou modules, utilisant des outils spécifiques au problème à traiter comme la planification, l'ordonnancement et la conduite temps-réel par exemple, et sur la «combinaison» de méthodes différentes pour traiter un problème particulier comme le diagnostic par exemple. L'échange d'information nécessaire à cette intégration peut s'effectuer par communication, auquel cas les modules doivent être fortement indépendants afin d'avoir un échange réduit, ou par le partage d'une base de données commune, ce qui constitue la solution fréquemment choisie.

---

7. Recipe driven production.



## 2.1 Les systèmes de contrôle/commande basés sur des unités pilotes

Les systèmes de contrôle/commande que nous avons comparés, présentent de nombreux points communs dont notamment les caractéristiques que nous venons d'énoncer : une hiérarchisation fonctionnelle, un principe d'intégration, et une base de données partagée. L'accent est également mis sur la nécessité d'un lien «réactif» entre l'ordonnancement et la conduite temps-réel [Macchietto 93] [Puigjaner *et al.* 94]. Les raisons d'une remise en cause fréquente du plan de production établi hors-ligne (hors temps-réel) sont nombreuses et variées. Il s'agit d'une conséquence directe des spécificités abordées au chapitre précédent : une maîtrise parfois incomplète de certains processus, une nécessité d'ajuster des paramètres pour passer de la production à l'échelle du laboratoire à celle à l'échelle de l'atelier, ainsi que pour faire face aux diverses perturbations, ou encore l'adaptation de la taille des lots à la demande de production, etc. A titre d'exemple, la fluctuation des durées opératoires des processus batch est fréquemment d'environ 5% [Puigjaner *et al.* 94].

Les systèmes de contrôle/commande considérés sont des travaux issus de la communauté du Génie des Procédés :

- *SUPERBATCH* (Supervisory Batch Management System) développé à l'Imperial College [Macchietto *et al.* 89].
- *FLESCHE* (Flexible Schedule) développé à l'Universitat Polyècnica de Catalunya [Puigjaner *et al.* 94].

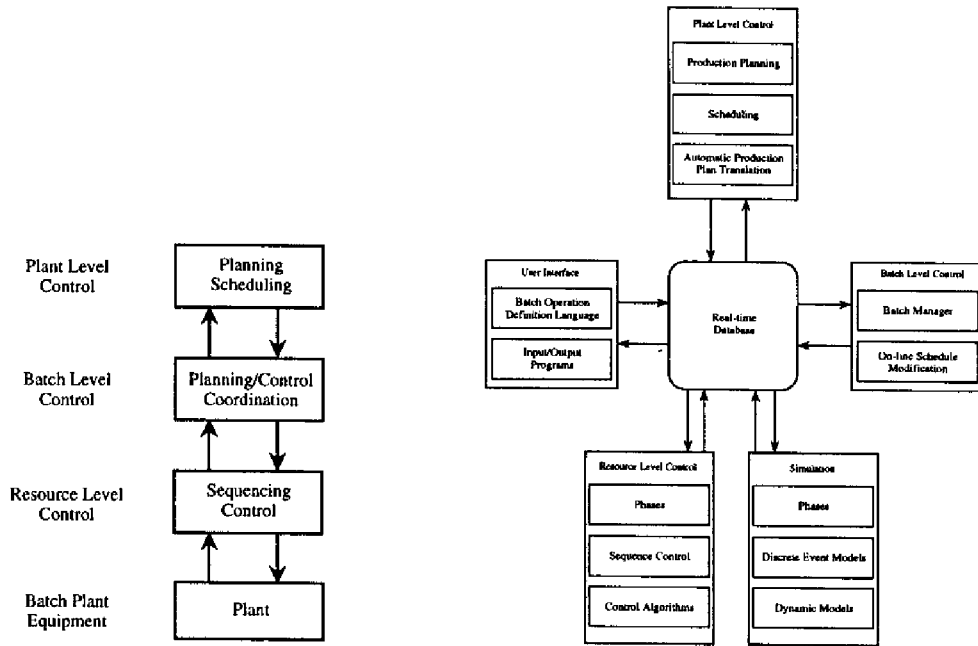
### 2.1.1 SUPERBATCH [Liu & Macchietto 95]

Il s'agit d'une structure logicielle hiérarchisée sur plusieurs niveaux (Figure 2.5.1), articulée autour d'une base de données commune (Figure 2.5.2), et basée sur une architecture répartie (Figure 2.5.3). Chaque niveau est composé de modules, dont le développement fait appel à des outils et méthodes spécifiques selon la fonction considérée : c'est le principe d'intégration que nous avons évoqué.

Le plan de production (optimal) est établi par un ordonnanceur hors-ligne<sup>8</sup>. L'ensemble des informations composant le plan (les ressources affectées aux lots, la taille des lots, la séquence des lots, etc.) est transmis au système SUPERBATCH qui vérifie la faisabilité du plan sur l'installation en examinant tout l'horizon de production. Cette vérification, qui est du ressort du niveau de gestion de l'installation ("Plant Level Control"), est faite à partir de l'état courant de l'installation, en se basant sur l'ensemble des modèles (dont les procédures des recettes) et des contraintes dont dispose SUPERBATCH. Le plan de fabrication réalisable peut alors être exécuté.

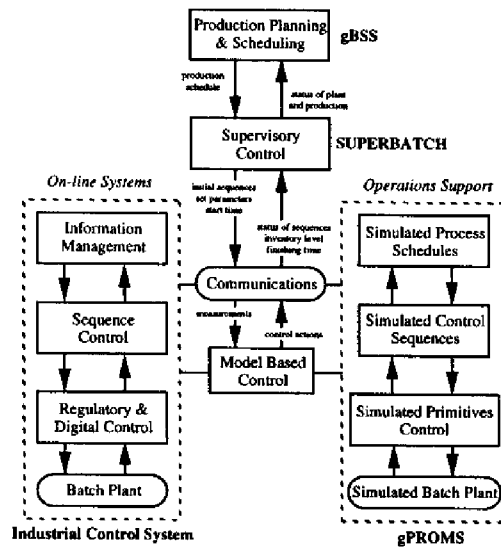
---

8. ordonnancement effectué à l'aide de gBSS (general Batch Scheduling System) développé à l'Imperial College.



2.5.1: Hiérarchie fonctionnelle

2.5.2: Structure logicielle



2.5.3: Architecture informatique

FIG. 2.5 – SUPERBATCH [Liu & Macchietto 95]

Le lancement des lots et la surveillance de leur exécution sont accomplis par le niveau de gestion des lots ("Batch Level Control"). L'exécution des lots est réalisée par l'émission de commandes vers le niveau inférieur de contrôle temps-réel ("Resource Level Control").

Les données de chaque niveau subissent une mise à jour périodique, de l'ordre des secondes pour le niveau de contrôle, de la minute pour le niveau de supervision, et de l'ordre de l'heure pour le niveau d'ordonnancement.

A partir des informations retournées périodiquement par le niveau inférieur de gestion des lots (état des ressources et des traitements), le déroulement du plan est surveillé: il s'agit essentiellement de surveillance des durées opératoires et des volumes traités. Dans le cas de «faibles» déviations, le superviseur effectue un «ré-ordonnancement»: par exemple, si la durée d'exécution d'une opération dépasse celle prévue par le plan, l'action correctrice consiste principalement à propager ce décalage sur les dates prévues pour les opérations suivantes.

Ce système de commande/supervision peut être aussi bien utilisé dans le cadre de la conduite temps-réel, par l'intermédiaire d'un système de contrôle industriel<sup>9</sup>, que dans celui de la simulation, à l'aide de l'environnement gPROMS<sup>10</sup> (Figure 2.5.3).

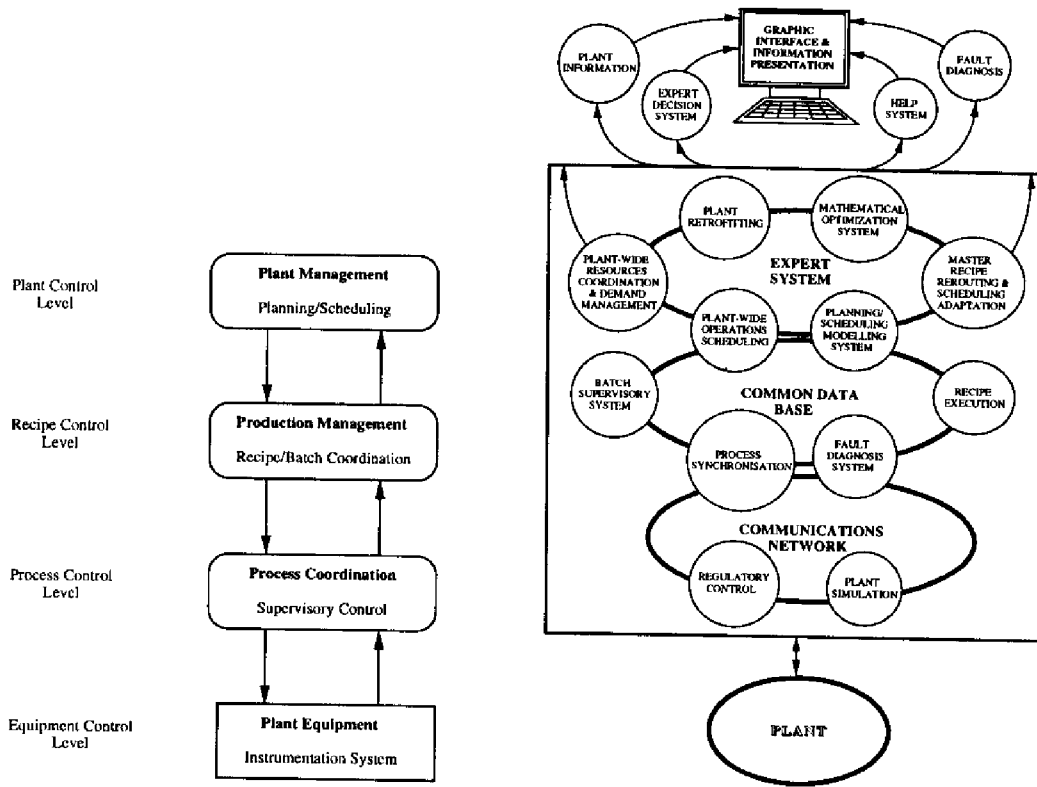
### 2.1.2 FLESCHE [Puigjaner *et al.* 94]

La hiérarchie fonctionnelle (Figure 2.6.1) et son fonctionnement sont similaires à ceux du système de commande/supervision de l'Imperial College dont les travaux sur FLESCHE se sont inspirés initialement. L'autre similitude est l'intégration: en effet, ce système repose également sur la combinaison d'un ensemble de fonctions organisées autour d'une base de données, et coordonnées par un système expert (Figure 2.6.2). Le caractère réparti de l'architecture supportant l'application est également un point commun (Figure 2.6.3).

L'interaction entre l'ordonnancement et l'exécution des recettes, pour l'absorption de déviations sur les durées opératoires, sur la qualité du produit, etc., est abordée différemment. Pour faciliter cette interaction, la notion de *recette flexible* est introduite. La recette est qualifiée de flexible en opposition aux recettes figées qui spécifient uniquement les conditions opératoires requises. Une recette flexible fournit, en plus, des règles d'adaptation de ces conditions en fonction du résultat d'exécution des opérations, de la qualité des matières premières, etc. Ainsi le lancement d'un lot s'effectue après l'initialisation de la recette correspondante en fonction de l'état courant, les paramètres de la recette étant de plus adaptés lors de l'avancement du lot. Il en résulte un ordonnancement flexible, d'où le nom de FLESCHE (Flexible Scheduling), au sein duquel les contraintes temporelles sont «souples», la priorité étant donnée à d'autres critères tels que la qualité du produit, la consommation d'énergie, etc. La modification de la recette et l'adaptation du plan de production (re-direction des lots, décalage temporel) sont tout de même limitées par les

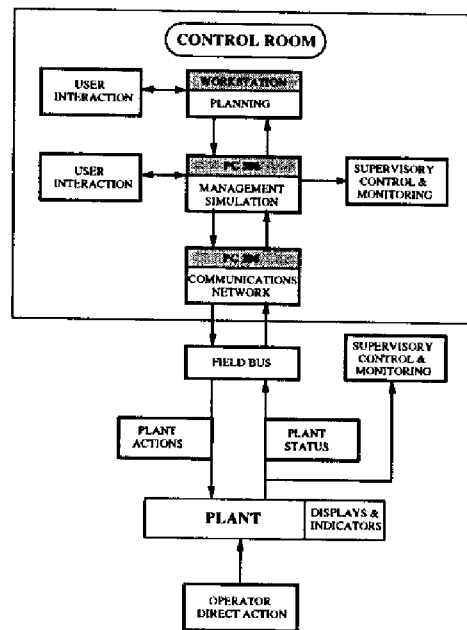
9. ACCOS: système de contrôle et de gestion des informations.

10. modélisation et simulation du procédé faite à l'aide du langage-environnement gPROMS (general PROcess Modelling System), développé à l'Imperial College.



2.6.1: Hiérarchie fonctionnelle

2.6.2: Structure logique



2.6.3: Architecture informatique

FIG. 2.6 - FLESCHE [Puigjaner et al. 94]

contraintes matérielles et les contraintes de gestion de l'ensemble des lots. Par conséquent, si la déviation est trop importante par rapport à l'objectif fixé par la recette nominale, un objectif de qualité par exemple, le niveau supérieur de planification/ordonnancement est sollicité.

Comme SUPERBATCH, ce système de commande/supervision peut être utilisé soit dans le cadre de la conduite temps-réel, soit dans celui de la simulation (Figure 2.6.2).

## 2.2 Les systèmes de contrôle/commande industriels

Les informations issues des publications [CEP 92] [CEP 93] [CM 95] sont restreintes cependant il est possible de constater que les constructeurs se sont inspirés des directives proposées, qu'ils ont parfois adaptées selon leur niveau d'expertise (contrôle avancé). La tendance est à l'intégration de techniques de contrôle avancées faisant appel à l'intelligence artificielle, aux réseaux neuronaux, aux algorithmes de contrôle perfectionnés (adaptatifs), etc. Un autre élément important est la modification du rôle de l'opérateur qui évolue plus vers le suivi de la production et l'adaptation des paramètres, le traitement des lots tendant à être automatisé (lancement et enchaînement des opérations).

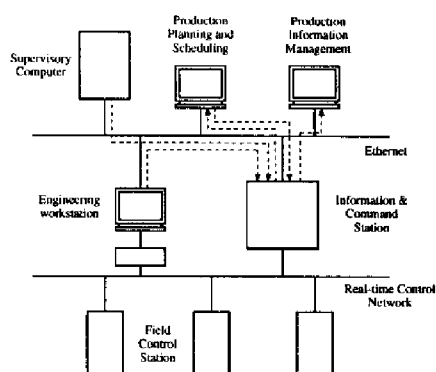


FIG. 2.7 – Architecture d'un système industriel (source Yokogawa)

L'évolution du système de commande/supervision repose plus sur la notion de configuration que sur la programmation, mais le principe utilisé pour y parvenir n'est pas évoqué.

Les structures logicielles proposées sont généralement développées selon une approche modulaire et hiérarchisée, et basées sur une architecture répartie autour d'un bus local (Figure 2.7). Parmi les systèmes existants, respectant les principes érigés par la norme, il est possible de citer [CM 95] : *Batch Supervisor* de *Honeywell*, *ChemFlex* de *Elsag Bailey*, *Batch X* de *Siemens*, et *RS3* de *Fisher Rosemount*.

### 3 La problématique de la surveillance

Le rôle de la supervision est clairement défini dans [Dubois & Gentil 90], il s'agit de "contrôler et surveiller l'exécution d'une opération ou d'un travail effectué par d'autres, sans rentrer dans les détails".

La supervision est donc un concept qui recouvre l'exécution des tâches de commande et de *surveillance* d'un procédé complexe, en se plaçant à un certain niveau d'abstraction.

En ce qui concerne la commande, la supervision doit exécuter les recettes de contrôle, c'est-à-dire piloter l'exécution de la séquence d'opérations décrite par chaque recette, et assurer la gestion temps-réel des ressources nécessaires à cette exécution. Elle est donc chargée de la configuration dynamique du procédé dont elle doit optimiser le fonctionnement selon des critères dépendant de son niveau d'abstraction (qualité, rendement, etc..).

Piloter la production signifie provoquer l'adaptation de la commande lorsqu'une déviation est constatée, cette adaptation allant de la modification du plan de fabrication par interaction avec l'ordonnancement, à la modification de conditions opératoires par interaction avec les niveaux de commande inférieurs. Adapter la commande suppose d'abord de constater les déviations (*la détection*), d'en trouver la cause (*le diagnostic*), et de déterminer la réaction adéquate (*la reprise*); c'est la fonction de surveillance de la production, fonction complémentaire à la commande [Combacau & Courvoisier 90].

Nous allons donc aborder la problématique de la surveillance dans le cadre de la conduite des procédés discontinus, problématique peu abordée dans les travaux que nous avons présentés. Nous nous pencherons uniquement sur la phase de détection des déviations ou défaillances.

Surveiller l'évolution d'un phénomène, détecter une déviation ou une défaillance, suppose avoir une référence par rapport à laquelle peut être établie une comparaison. La surveillance est donc basée sur un modèle du procédé qui permet d'effectuer une comparaison entre le comportement attendu du procédé, obtenu à l'aide du modèle, et le comportement réel observé lors de l'exécution des opérations. Il est alors utilisé dans certaines techniques de diagnostic (diagnostic fondé sur des modèles) [Künzle *et al.* 94], ainsi que pour tester des stratégies de reprise et de compensation [Valette & Künzle 94].

La nature hybride du procédé, nous amène à surveiller des phénomènes de nature différente : des phénomènes discrets, comme la vérification du respect des durées opératoires prévues, et des phénomènes continus, comme le bilan matière, ou la valeur et la vitesse d'évolution d'une variable continue. Par conséquent, nous allons nous intéresser à la surveillance, basée sur un modèle, dans le cadre des systèmes à événements discrets et dans celui des systèmes continus.

### 3.1 La surveillance des systèmes à événements discrets

Les principales approches de la surveillance basées sur un modèle du procédé (modèle des comportements normaux du procédé indépendamment de la stratégie de commande), sont les suivantes<sup>11</sup> :

- Utilisation du modèle en *filtre* [Cruette 91] [Lhoste 91].

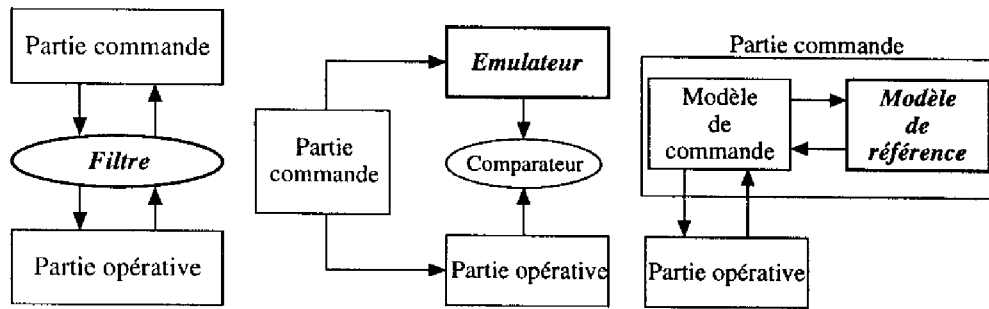
Le modèle est placé entre la partie commande<sup>12</sup> et la partie opérative<sup>13</sup> (Figure 2.8.1). Il permet d'une part d'assurer la cohérence des commandes émises en filtrant celles non prévues dans un état donné, et d'autre part de valider l'exécution correcte d'une commande en comparant le signal émis par le procédé et le signal attendu.

- Utilisation du modèle en *émulation* [Holloway & Krogh 90].

Le modèle est placé en parallèle de la partie opérative (Figure 2.8.2), et il est émulé pour imiter le comportement du procédé. Il permet de comparer les signaux émis par le procédé et ceux issus de l'émulateur.

- Utilisation du modèle en *référence* [Combacau 91].

Le modèle du procédé est placé en parallèle d'une partie commande hiérarchisée et modulaire (Figure 2.8.3). Il est consulté par la commande avant chaque émission d'une requête vers la partie opérative, afin d'en assurer la cohérence. Son évolution dépend de la commande d'une part, et de la réception des signaux émis par le procédé d'autre part.



2.8.1: Modèle en filtre

2.8.2: Modèle en émulation

2.8.3: Modèle en référence

FIG. 2.8 – Différentes positions du modèle du procédé [Chaillet-Subias 95]

11. synthèse effectuée à partir de [Zamai 94] [Chaillet-Subias 95].

12. modèle de la commande représentant la stratégie de conduite du procédé.

13. le procédé.

Une étude comparative<sup>14</sup> de ces différentes approches reposant sur des critères tels que l'efficacité de la détection, la possibilité d'intégration de l'ensemble des fonctions de la surveillance, et la position du modèle, suggère que l'approche à modèle de référence est la plus générale. Cette dernière établit une distinction entre une «défaillance commande» et une «défaillance procédé», elle permet d'intégrer le diagnostic et la reprise pour laquelle il est par exemple nécessaire de pouvoir appliquer des commandes d'exception, et surtout s'inscrit dans une structure de commande hiérarchisée et modulaire nécessaire face à la complexité du système.

### 3.2 La surveillance des systèmes continus et discontinus

Le principe de la démarche utilisée dans le cadre des systèmes continus est analogue à celui des systèmes à événements discrets (SED) puisque, dans les deux cas, la détection est basée sur la comparaison du comportement (ou de l'état) réel, avec celui d'un modèle.

La modélisation causale et/ou qualitative, est une voie très explorée dans le cadre de la surveillance des systèmes continus [Leyval *et al.* 94], pour lesquels la manipulation d'informations qualitatives permet plus facilement l'interprétation et l'explication des phénomènes observés que la manipulation d'un grand nombre de grandeurs précises. La tendance est à l'introduction du flou pour la représentation de spécifications non strictes [Montmain 94].

Ces modèles sont alors utilisés comme référence de comparaison du comportement du procédé. L'utilisation d'un modèle du procédé est d'autant plus nécessaire dans le cadre de la surveillance des systèmes continus que certaines variables donnant un sens physique et compréhensible au phénomène observé ne peuvent pas être directement accessibles par la mesure [Féray-Beaumont & Gentil 91].

Si dans certaines approches de la surveillance des SED, la détection est basée sur la vérification que le procédé évolue dans l'ensemble des états accessibles décrits par son modèle, indépendamment de la commande appliquée, il ne peut pas en être de même pour la surveillance des systèmes continus. En effet, le phénomène continu à surveiller est étroitement lié à la phase dans laquelle se trouve la commande.

Prenons par exemple le cas de la vitesse de variation d'une grandeur continue lors du régime transitoire et du régime permanent : une pente importante sur la trajectoire de la variable continue peut être considérée comme normale en régime transitoire (démarrage, changement de fonctionnement), alors qu'elle est le symptôme d'une déviation nécessitant une action correctrice, voire d'un arrêt d'urgence, lors du régime permanent (emballement d'une réaction exothermique par exemple). Considérer la vitesse d'évolution d'une grandeur continue indépendamment de la phase active de la commande amènerait donc soit à signaler un dysfonctionnement inexistant (génération d'un événement non pertinent), soit à ignorer

---

14. synthèse effectuée à partir de [Zamai 94] [Chaillet-Subias 95].



une situation anormale (perte d'un événement significatif).

Un exemple simple est le cas du réservoir en phase de remplissage dont le volume diminue. Aucun dysfonctionnement ne peut être signalé si toute vitesse d'évolution du volume comprise entre le débit maximal d'alimentation et celui de soutirage, et toute valeur comprise entre 0 et  $V_{max}$ , sont considérées comme normales.

Définir la détection de dysfonctionnements uniquement par rapport aux états continus accessibles est donc clairement insuffisant pour assurer une surveillance efficace. Le contexte, c'est-à-dire la configuration, dans lequel se trouvent la commande et le procédé doit nécessairement être pris en compte. Nous retrouvons en fait la problématique des systèmes discontinus. D'une façon générale, la notion d'évolution anormale d'une grandeur continue est liée aux caractéristiques de la transformation considérée. Par exemple lors d'une réaction exothermique, une forte exothermicité à un instant donné peut nécessiter dans certains cas une action correctrice immédiate, et dans d'autres cas aucune intervention car elle est caractéristique de la transformation en cours et elle est connue pour être ponctuelle (phénomène «secondaire»). Ces cas de comportements subtils rendent encore plus difficile la détection d'anormalités comportementales, et il faut parfois «laisser faire» d'où l'importance de l'appréciation de l'opérateur quant à la pertinence d'une détection [Gentil 95].

Nous avons évoqué la détection d'anomalies sur les trajectoires de variables continues, mais il faut également détecter les anomalies à un niveau encore plus bas, celui de l'acquisition des mesures. Comme nous l'avons précisé au chapitre précédent, les contraintes technologiques sont très fortes dans les procédés discontinus et continus, les capteurs présentent des problèmes de robustesse dans ces milieux souvent hostiles (bruit de mesure). Pour prévenir de mauvaises mesures, qui rendent erronée toute commande ou surveillance, différentes techniques sont utilisées, basées sur la redondance physique, ou la redondance analytique, dont la reconciliation de données par exemple [Jambois *et al.* 95].

### 3.3 Limitations d'une approche purement discrète de la supervision

A la suite des travaux de Ramadge et Wonham, une approche «commande supervisée» s'est développée en assimilant en grande partie la notion de superviseur de ces auteurs à celle qui correspond à l'interface entre la gestion de production et la commande en temps-réel dans les systèmes hiérarchisés [Peleties & DeCarlo 89] [Gollu & Varaiya 89] [Antsaklis *et al.* 93]. Ce rapprochement nous semble pourtant relativement contestable.

L'approche de Ramadge et Wonham [Ramadge & Wonham 89] repose sur un modèle abstrait du système à superviser sous la forme d'un ensemble de graphes d'états. Une notion importante est celle de commandabilité: seules certaines transitions d'état sont commandables, c'est-à-dire seules certaines transitions peuvent être «verrouillées». Les autres

peuvent se produire spontanément et seront seulement observées.

L'objectif de la politique de supervision est alors de commander les transitions d'état commandables de façon à éviter que le système n'évolue vers des états interdits (états dangereux), ou vers des situations de blocage (plus aucune évolution n'est possible).

La supervision telle que nous la concevons, comprend le pilotage du procédé pour réaliser le plan de production, et sa surveillance. Or la réalisation d'un plan de production ne peut absolument pas se représenter par un ensemble d'états interdits. La dimension temporelle par exemple est essentielle.

Il est vrai que dans les installations multiproduits multiflux, il est nécessaire d'assurer que les allocations de ressources ne mènent ni à des mélanges de produits, ni à des situations de blocage. Toutefois ces contraintes, si elles s'expriment bien sous la forme d'états interdits [Sanchez & Macchietto 93] [Rotstein *et al.* 95] [Yazdi & Jorgensen 96], doivent être prises en compte dès la génération du plan de fabrication, qui doit être faisable.

L'objectif de la gestion de production et du pilotage est de trouver une trajectoire, dans l'espace d'état du système de production, qui vérifie un grand nombre de contraintes temporelles et de contraintes d'états interdits. La simple représentation des contraintes d'états interdits découlant du partage des ressources, peut se faire sans énumération des états, en utilisant par exemple les réseaux de Petri [Valette 95] ou les graphes disjonctifs [Esquirol *et al.* 95]. Leur exploitation indépendamment des contraintes temporelles pour trouver une bonne trajectoire ne serait intéressante que si elles étaient «dominantes», or il n'est pas évident que cela soit le cas pour les procédés batch. En effet, les contraintes temporelles sont généralement prises en compte avant celles des états interdits dont la considération introduit alors un décalage temporel dans le lancement des lots ou des opérations, si cela s'avère nécessaire [Pages & Pingaud 94].

Le second rôle du superviseur est la surveillance et la réactivité face à des comportements non nominaux. Certains événements commandables peuvent cesser de l'être, et il est alors nécessaire de trouver des commandes de compensation permettant d'éviter une évolution dangereuse. Certaines approches basées sur les travaux de Ramadge et Wonham [Niel *et al.* 96] peuvent être riches dans ce cadre car les notions d'observabilité et de commandabilité sont peu explicites dans les autres approches. Toutefois, nous voyons deux limites majeures. La première est que le modèle du procédé est purement discret. La seconde est que la notion classique de détection par comparaison entre l'évolution du système et celle du modèle est absente. Le premier point nous semble particulièrement important. Même à un niveau d'abstraction élevé, c'est-à-dire très agrégé, il nous paraît souhaitable de pouvoir disposer de variables d'état continues afin de pouvoir établir par exemple des bilans matière. Se restreindre à un modèle purement discret, et ne comprenant aucune considération temporelle explicite, nous semble très réducteur.

## 4 Conclusion

Le système de commande/supervision est généralement développé selon une approche fonctionnelle qui est organisée de manière hiérarchisée, modulaire, et répartie. Un aspect important des systèmes présentés repose sur le principe d'intégration d'un ensemble de fonctions, dont l'échange d'information est basé sur l'utilisation d'une base de données partagée.

Nous avons mis en évidence que le modèle de supervision ne peut être exclusivement discret. La surveillance de phénomènes continus passe par l'utilisation d'un modèle continu même à un niveau d'abstraction élevé. La notion de configuration discrète est également essentielle.

Il est par conséquent nécessaire de manipuler des modèles de nature différente à tous les niveaux (niveaux plus ou moins agrégés), aussi bien un modèle de représentation du plan de fabrication (les recettes de contrôle), qu'un modèle du procédé et de ses contraintes.

Nous allons donc compléter le cadre de l'étude en abordant l'aspect modélisation, la problématique sera ainsi définie sous ses angles divers.

# Chapitre 3

## La modélisation

Nous avons clairement constaté que la supervision des procédés discontinus nécessite la prise en compte à la fois de l'aspect discret et de l'aspect continu. Or quand les deux aspects sont étroitement liés, et que les degrés d'intégration et de complexité sont élevés, la modélisation du système par une seule de ses composantes n'est pas possible : la représentation ne saurait être intégralement continue ou intégralement discrète [Quénec'hdu *et al.* 96], [Dubois & Gentil 90]. La modélisation de l'aspect hybride du système est alors abordée selon deux tendances : d'une part, la tendance reposant sur l'intégration des aspects continu et discret au sein d'un même formalisme, et d'autre part, celle associant un formalisme différent pour chacun des deux aspects.

### 1 Les différentes approches de l'aspect hybride

#### 1.1 L'approche intégrée

Dans la démarche d'intégration des deux aspects au sein d'un même modèle, deux approches «**intégrées**» sont possibles. Elles dépendent du modèle «dominant», c'est-à-dire du modèle à partir duquel est effectuée l'extension : **approche continue** et **approche discrète** sont les distinctions faites dans la littérature.

- L'approche «continue»

L'intégration de l'aspect discret au sein de la représentation continue s'effectue par l'introduction de variables booléennes (ou entières) au sein d'un modèle continu (système d'équations), ou par l'introduction de l'aspect événementiel au sein de graphes de flux. L'ordonnancement formulé comme un problème d'optimisation par «variables mixtes» (variables continues et entières) [Georges 96], et l'introduction des «éléments de commutation» (comportement discret ou discontinuité de structure) dans le formalisme du Bond Graph [Stromberg *et al.* 93], [Quénec'hdu *et al.* 96] en sont des exemples.

- L'approche «discrète»

L'intégration de l'aspect continu au sein d'un modèle à événements discrets comme les réseaux de Petri (RdP) est un exemple de ce type d'approche. Le champ d'application des RdP a été étendu aux systèmes continus par l'introduction de nouvelles classes de RdP au sein desquelles la notion de marquage dans l'espace des réels est apparue. La possibilité d'une évolution continue du marquage en fonction du temps qui en résulte permet la modélisation de phénomènes continus (franchissement des transitions par quantités infinitésimales). Les modèles mixtes dérivés des RdP Continus et des RdP ordinaires que sont les RdP Hybrides, permettent alors de représenter explicitement l'interaction entre la partie continue et la partie discrète par l'intermédiaire d'arcs. Les RdP Différentiels ont introduit par la suite les places et transitions différentielles [Demongodin & Koussoulas 96].

Nous reviendrons par la suite sur les principaux modèles développés selon cette approche (section 2.1).

## 1.2 L'approche séparée

La tendance opposée, celle consistant à séparer l'aspect continu et l'aspect discret, fait également apparaître deux approches : d'une part celle reposant sur l'utilisation du modèle à événements discrets comme un «**moniteur de systèmes d'équations**», et d'autre part celle fondée sur la séparation de la partie continue et de la partie discrète en introduisant une interface. Cette dernière est communément référencée **approche «superviseur»** dans la littérature.

- Le «moniteur» de systèmes d'équations

Dans un certain sens, le modèle à événements discrets peut être considéré comme un moniteur des équations formant le modèle continu. A un instant de temps donné, seul l'un des modèles est actif. Soit il y a modification de la structure des équations formant le modèle continu et le temps n'évolue pas, soit le temps évolue mais la structure des équations ne change pas.

Cette caractéristique se retrouve dans les différentes approches «**séparées**» basées réseaux de Petri, présentées à travers la section 2.2, ainsi que dans les modèles associant des équations différentielles aux transitions d'un réseau de Petri de Haut-Niveau [Wieting & Sonnenschein 95] [Brielman 95].

Parmi les approches développées selon cette tendance, nous pouvons citer celle basée sur l'utilisation combinée des StateCharts et de Signal<sup>1</sup> [Guéguen 96]. Dans cette démarche, la structure du StateChart permet la construction d'un modèle continu traduit par des expressions du langage Signal, le point sensible étant la définition

---

1. langage synchrone traitant l'aspect temporel de la combinaison de signaux [Benveniste *et al.* 93].

des mécanismes permettant de spécifier le type de relations lors des commutations en fonction du comportement attendu.

Selon la même idée, certains travaux sur les automates hybrides (extension des automates temporisés) [Nicollin *et al.* 93], [Nerode & Kohn 93], associent un automate classique et un ensemble de variables continues dont les conditions portant sur leurs valeurs sont exprimées sur les transitions de l'automate. L'évolution dynamique des variables continues est modélisée par des ensembles d'équations différentielles associés aux états discrets de l'automate.

- L'approche «superviseur»

Dans cette approche, deux blocs correspondant aux aspects continu et discret sont couplés par une interface. Chaque modèle décrit un sous-système (le procédé, son superviseur) qui peut éventuellement évoluer de façon autonome. Il ne s'agit pas de deux vues complémentaires d'un même système, mais plutôt de deux sous-systèmes représentés par deux modèles. L'interface traduit les signaux échangés entre les deux parties : les observations du procédé sont traduites en symboles (un symbole associé à chaque région de la partition de l'espace d'état des variables continues) constituant un alphabet dont chaque élément est interprété par le superviseur comme un événement (changement de région), et les commandes du superviseur sont transformées en variables d'entrée du procédé (les valeurs de la commande continue sont sélectionnées dans un ensemble fini de valeurs constantes). Les travaux basés sur les machines à états finis de [Gollu & Varaiya 89], [Antsaklis *et al.* 93], et ceux basés sur les RdP de [Peleties & Decarlo 94], [Tittus 95], pour en citer quelques-uns, s'inscrivent dans ce cadre.

Les approches que nous considérerons par la suite, qu'il s'agisse des approches intégrées ou séparées, se restreignent à celles qui sont basées sur les réseaux de Petri pour la partie discrète.

## 2 Approches basées sur les réseaux de Petri

Nous avons, dès ce chapitre, effectué un choix au sujet du modèle à événements discrets utilisé. Les réseaux de Petri constituent l'outil de modélisation retenu pour l'aspect discret des systèmes discontinus. Sans effectuer d'étude comparative entre différents modèles à événements discrets, étude que le lecteur peut trouver notamment dans [Combacau 91], précisons que ce modèle possède certaines des propriétés essentielles attendues : possibilité de hiérarchisation, modélisation directe des mécanismes de communication, expression du parallélisme, prise en compte des contraintes logiques, validation formelle de certaines propriétés du modèle, etc., sans oublier l'aspect mise en œuvre (moteur d'inférence) car outre l'aspect modélisation, l'aspect implémentation est d'importance.

De plus, les réseaux de Petri sont l'un des modèles les plus utilisés dans le cadre de la surveillance des systèmes à événements discrets [Cruette 91] [Combacau & Courvoisier 90] [Holloway & Krogh 90] [de Bonneval 93] [Valette 95].

La prise en compte d'événements extérieurs (réceptions de messages, franchissement de seuils) est généralement effectuée par l'intermédiaire des transitions. Lorsqu'une transition est sensibilisée, l'occurrence de l'événement qui lui est associé provoque son franchissement. Si un événement se produit alors qu'il n'est associé à aucune des transitions franchissables du réseau, cela signifie qu'il y a une incohérence entre l'état du modèle (son marquage) et l'évolution du procédé. Une anomalie est détectée.

Les réseaux de Petri temporels offrent en plus la possibilité d'exprimer des considérations temporelles explicites. La fenêtre temporelle de franchissement de la transition correspond à l'intervalle de temps pendant lequel l'événement associé à une transition doit être reçu. La borne supérieure de cet intervalle est donc importante, elle correspond en fait au mécanisme classique du *chien de garde* [Valette & Künzle 94].

Les réseaux de Petri ont d'ailleurs été souvent utilisés pour modéliser l'aspect discret des systèmes dynamiques hybrides<sup>2</sup> [Yamalidou & Kantor 91], [Lloyd & Salleh 91], [Hanisch 92], [Genrich *et al.* 94], [Azzopardi & Holding 95], [Genrich & H.M.Hanisch 95], [Tittus *et al.* 95], [Fleck & Hanisch 96].

Quelques notions sur les réseaux de Petri, dont certaines sur les modèles évoqués à travers le manuscrit, sont données en annexe.

## 2.1 Les approches intégrées

L'intégration de l'aspect continu au sein des RdP a modifié le concept de marquage ; le marquage n'est plus considéré dans l'ensemble des entiers naturels positifs  $\mathbb{N}^+$  mais dans l'ensemble des réels positifs  $\mathbb{R}^+$ . L'ordre chronologique d'apparition des modèles que nous allons présenter, montre l'évolution des primitives de représentation en fonction des besoins. Le point de départ est la volonté d'avoir une vue agrégée du comportement d'un réseau de Petri temporisé avec un grand nombre de jetons. Au lieu de simuler le franchissement d'une transition  $t$  tous les  $\theta_t$ , il est nettement plus simple d'écrire que le nombre de jetons déplacés par le franchissement de la transition  $t$  pendant une durée  $\Theta$  est  $\frac{\Theta}{\theta_t}$ . Cependant pour cela il a été nécessaire de considérer le marquage d'une place comme un réel et non plus comme un entier. Ce modèle a ensuite été étendu sous la forme des RdP Hybrides au sein desquels interagissent aspects discret et continu, puis des modèles à caractère plus spécifique comme les RdP Lots et les RdP Continus Causaux ont été définis. Seuls les concepts essentiels de ces nouveaux types de RdP sont présentés, les références indiquées permettront au lecteur intéressé d'accéder aux définitions formelles ainsi qu'à des exemples de modélisation.

---

2. ces travaux s'orientent essentiellement vers la simulation.

### 2.1.1 Les RdP Continus

Les RdP Continus [David & Alla 87] permettent la modélisation de systèmes dont les comportements peuvent être assimilés à des flots continus, ainsi qu'une modélisation plus condensée de systèmes discrets dont le marquage est très grand de par le nombre important d'entités mises en jeu dans le système (réduction du nombre d'états accessibles). La modification essentielle est, comme nous l'avons précédemment indiqué, l'introduction de la notion de marquage dans l'espace des réels. Le principe d'évolution du marquage se trouve par conséquent modifié :

- une transition est considérée sensibilisée dès que le marquage des places en amont est strictement non nul,
- le franchissement d'une transition n'est plus instantané puisque le jeton (marque réelle) va franchir la transition par quantité infinitésimale. Des vitesses de franchissement sont donc associées aux transitions. Une transition continue avec une place d'entrée continue et une place de sortie continue, est l'image exacte du sablier. Le jeton *s'écoule continuellement* de la place d'entrée vers la place de sortie.

Ce fonctionnement est clairement exprimé à travers l'équation fondamentale du RdP. L'équation fondamentale du RdP Discret (3.1) devient l'équation (3.2) pour le RdP Continu.

$$M_{k+1} - M_k = C.\bar{s} \quad (3.1)$$

$$\frac{\delta M}{\delta t} = C.v(t) \quad (3.2)$$

avec  $M$  le vecteur de marquage du RdP,  $C$  la matrice incidente (matrice décrivant la structure du réseau),  $\bar{s}$  le vecteur de franchissement (séquence de transitions),  $v(t)$  le vecteur des vitesses de franchissement des transitions en cours de franchissement à l'instant de temps  $t$ .

Quatre modèles de RdP Continu ont été définis, ils sont différenciés par la façon de calculer les vitesses de franchissement des transitions.

- Le RdP Continu à Vitesses Constantes [David & Alla 89]

Une vitesse de franchissement maximale est associée à chaque transition. Tant que la place d'entrée de la transition n'est pas vide, la transition est franchie selon sa vitesse maximale. Si la place d'entrée est vide mais qu'elle est alimentée en amont par une autre transition, alors la vitesse de franchissement est le minimum des vitesses maximales des deux transitions.

- Le RdP Continu à Vitesses Variables [David & Alla 89]

La vitesse maximale de franchissement d'une transition dépend du marquage des places en amont de la transition et d'une valeur constante associée à la transition.



Les marquages et les vitesses sont donc des fonctions continues du temps, alors que dans le RdP continu à vitesse constante cette propriété est vraie pour les marquages mais pas pour les vitesses qui sont simplement constantes par morceaux. Il fournit une meilleure approximation du RdP discret, en particulier lorsque le nombre de marques est petit, mais les simulations sont plus longues.

- Le RdP Continu Asymptotique [Bail *et al.* 93]

Il est possible de distinguer plusieurs phases d'évolution du marquage. Chaque phase est caractérisée par une période durant laquelle le vecteur des vitesses de franchissement est constant. La vitesse de franchissement d'une transition est alors constante par morceaux, et l'évolution du marquage est donc composée de plusieurs phases de fonctionnement. Il s'agit de l'approximation du RdP Continu à Vitesses Variables, mais la simulation est facilitée. Jusque là ces trois modèles souffrent d'une même lacune, l'environnement des systèmes étudiés ne peut pas être pris en compte sans compliquer la modélisation.

- Le RdP Continu à Vitesses Fonction du Temps [Dubois *et al.* 94]

Cette extension des RdP continus temporisés permet de prendre en compte l'environnement du système modélisé comme par exemple la fluctuation de l'approvisionnement d'un atelier. La vitesse maximale de franchissement d'une transition est soit une fonction continue du temps<sup>3</sup>, soit une fonction constante par morceaux. La vitesse de franchissement à un instant donné est déterminée par une relation identique à celle du RdP continu temporisé, avec comme seule différence une vitesse maximale fonction du temps. Dans le cas de vitesses constantes par paliers, le RdP obtenu peut être vu comme une suite de RdP continus à vitesses constantes ; un sous-modèle de RdP continus à vitesses fonctions du temps est défini : le RdP à vitesses constantes par paliers.

Le modèle RdP continu peut donc soit modéliser un système continu, soit constituer une approximation d'un modèle discret, plus ou moins finement selon la classe utilisée. Dans ce dernier cas, il permet de réduire les durées de simulation qui deviennent importantes lorsqu'un modèle RdP discret contient un grand nombre de jetons. Ces modèles ne traitent pas l'évolution du système événement par événement mais modélisent des variations moyennes. Ils s'avèrent plutôt appropriés pour la recherche des flux caractérisant le régime stationnaire et le comportement dynamique d'un système. Une application pour l'analyse des lignes de fabrication est traitée dans [Zerhouni & Alla 92]. L'utilisation de cette approche pour l'étude du bilan matière dynamique d'un procédé continu (réacteur continu en série avec deux colonnes à distiller) est par ailleurs abordée dans [Daubas 94].

---

3. la fonction continue du temps pouvant être quelconque, ce modèle englobe les précédents.

### 2.1.2 Les RdP Hybrides

Les réseaux de Petri continus sont adaptés pour la modélisation d'un système fonctionnant en régime permanent mais ne permettent pas de représenter l'influence de conditions discrètes simples sur le flux continu. Considérons le cas d'un transfert de matière à travers une vanne deux états, dont le flux continu est à débit constant en régime permanent. Ce débit est traduit par une vitesse maximale de franchissement d'une transition  $V_{max}$ . Si la vanne est fermée, la vitesse maximale correspondante devient alors nulle au lieu de  $V_{max}$ . Ce comportement est équivalent à avoir brusquement un autre RdP continu, ou bien à faire varier la vitesse de franchissement d'une transition dans un réseau de Petri Continu à Vitesse Fonction du Temps.

Les RdP Hybrides [Bail *et al.* 91] contenant des places discrètes et des places continues permettent de représenter au sein d'un même formalisme ce couplage entre un fonctionnement continu et un fonctionnement discret (représentation graphique unifiée des variables continues et des variables discrètes).

L'exemple présenté sur la figure 3.1 illustre cette interaction entre les parties discrète et continue. Les places et transitions discrètes, et les places et transitions continues ont un graphisme différent ; elles sont respectivement représentées en trait simple et en trait double.

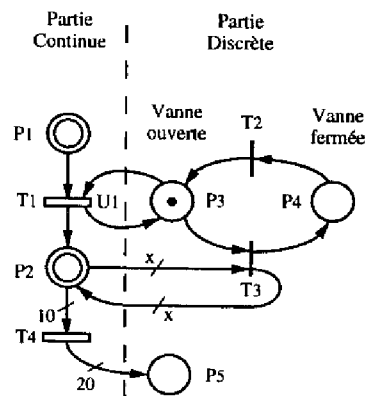


FIG. 3.1 - Exemple de RdP Hybride

La vitesse maximale de la transition  $T_1$  est  $U_1$  quand la vanne est ouverte (si le marquage de  $P_1$  le permet). La partie discrète conditionne l'évolution de la partie continue. Dès que le marquage de la place  $P_2$  atteint la valeur seuil réelle positive  $x$  indiquée sur l'arc, la transition  $T_3$  est franchie, sans modification du marquage de  $P_2$ . La partie continue conditionne l'évolution de la partie discrète. Le tir de la transition  $T_3$  provoque la fermeture de la vanne et l'écoulement est arrêté ( $P_3$  démarquée). De par leurs marquages respectifs, les parties discrète et continue ont une influence mutuelle l'une sur l'autre.

Dans le RdP continu, le phénomène d'accumulation, traduit par le marquage d'une place, résulte des conditions aux frontières : les vitesses de franchissement de transitions amont et aval de la place considérée. Le RdP Hybride augmente le potentiel de modélisation par la prise en compte de l'influence de l'environnement, et permet de représenter les transformations de lots (lot transformé en pièces et vice-versa). Cette transformation de lots est illustrée sur la figure 3.1 : un lot de 10 litres donne 20 bouteilles de 50 cl. Dès que le marquage de la place  $P_2$  atteint la valeur seuil 10 indiquée sur l'arc entrant de la transition  $T_4$ , cette dernière est franchie, et 20 jetons (marques discrètes) sont déposés dans sa place de sortie  $P_5$  (place discrète).

Utilisé pour modéliser des systèmes hybrides comme par exemple un réseau d'eau potable constitué de réservoirs interconnectés par des vannes deux états [Alla 94], ou encore un système de production d'énergie électrique d'origine hydraulique [David 91], le RdP Hybride a été essentiellement utilisé dans le cadre de l'évaluation de performances.

### 2.1.3 Les RdP Lots

Extension des RdP Hybrides, les RdP Lots ont été définis pour modéliser le comportement de systèmes de conditionnement hybrides, comportant des flux continus et des flux discrets, avec retard et accumulation [Demongodin *et al.* 93]. Qu'il s'agisse du réseau de Petri Continu ou du réseau de Petri Hybride, le phénomène d'accumulation ou de transformation est limité à une seule variable : une place donnée ne peut mémoriser qu'une seule variable continue, son marquage. Or dans certains cas, il est nécessaire de considérer tout un ensemble de caractéristiques qui évoluent chacune selon des règles spécifiques. Par exemple, un lot de bouteilles sur un convoyeur est caractérisé par sa longueur, sa densité, son nombre d'éléments, etc. Le réseau de Petri Lots étend le potentiel descriptif des réseaux hybrides en intégrant des lois de transformation des lots<sup>4</sup>. Cette intégration s'effectue par l'attribution à la sémantique d'une place (définie place «lot») d'un jeu de paramètres et d'une loi continue d'évolution temporelle du système pendant la durée du marquage. Les vitesses de franchissement des transitions (définie transition «lot») amont et aval sont alors liées au contenu informationnel de la place. La caractéristique de vitesse introduisant implicitement le temps, le modèle est par définition un RdP mixte temporisé : des temporisations peuvent être associées aux transitions discrètes et des flux de franchissement aux transitions continues et lots.

Définis pour modéliser le comportement dynamique d'un système de transport composé de convoyeurs à accumulation (accumulation des éléments convoyés), ils intègrent une formalisation de l'évolution des lots de pièces sur ce type de convoyeur. Un convoyeur est considéré comme une unité continue caractérisée par ses débits d'entrée et de sortie qui peuvent être variables selon les contraintes imposées par les éléments aux frontières comme les sites de réceptions du flux sortant par exemple. Outre ses flux d'entrée et de

4. notons que la notion de lot est différente, un lot étant défini ici comme un ensemble de pièces discrètes, même si le flux de pièces peut-être considéré continu.

sortie, un convoyeur est également caractérisé par un ensemble de paramètres tels que sa longueur d'occupation, sa vitesse et sa capacité maximale. Afin de modéliser le flux de pièces circulant sur le convoyeur, ce flux est décomposé en «lots cohérents internes» (ensembles de pièces ayant les mêmes caractéristiques de répartition sur le convoyeur pendant un intervalle de temps) décrits par quatre paramètres qui sont sa longueur, sa densité, sa position, et la date à laquelle il occupe cette position. La vitesse de déplacement d'un lot est déterminée par la fonction de calcul de l'évolution du lot au cours du temps, associée à la place le contenant.

Une extension des RdP Lots, appelée RdP Lots à Caractéristique Variable, a permis d'intégrer une deuxième formalisation représentant l'évolution de lots de pièces sur des convoyeurs sans accumulation (distance constante entre les éléments convoyés) [Demongodin *et al.* 94].

Le RdP Lots permet de représenter et de simuler le comportement temporel d'une chaîne de conditionnement en présence de perturbations courtes et locales, avec une bonne précision et un temps de calcul réduit [Audry & Prunet 95]. Un exemple d'application de ce modèle pour l'optimisation du fonctionnement de lignes de production à haute cadence est traité dans [Audry *et al.* 94]. Utilisé pour modéliser avec précision la partie physique du système, il permet de déterminer, par simulation, la loi de commande la mieux adaptée (interaction effectuée par l'intermédiaire des places discrètes).

#### 2.1.4 Les RdP Continus Causaux

Ce modèle de représentation d'un système dynamique continu est inspiré d'une part d'une méthode de modélisation causale des systèmes continus, et d'autre part des RdP Continus temporisés, plus précisément les RdP Continus à vitesse fonction du temps [Gomaa & Gentil 95].

La modélisation causale des systèmes continus évoquée, est basée sur une représentation du procédé continu par un graphe de causalité [Gomaa & Gentil 96a] comprenant : des noeuds auxquels sont associées les variables pertinentes du système, et des arcs représentant les liens causaux entre ces variables (Figure 3.2.1). L'évolution d'une variable du graphe est approximée par une fonction affine par morceaux constituant une suite de segments contigus appelés épisodes. Un épisode est défini par la date de début  $t_i$ , la valeur de début  $Val_i$ , la pente  $Pen_i$ , et la date de fin  $t_{i+1}$ . Les arcs du graphe causal portent les informations qui permettent la propagation des épisodes dans le temps, d'une variable à l'autre : le gain  $K$ , le retard pur  $r$ , et le temps de réponse  $t_r$ . Ces paramètres sont regroupés dans une Fonction de Transfert Qualitative (FTQ), et permettent de définir la réponse du noeud aval, la Réponse Qualitative (RQ).

La traduction d'un tel modèle sous la forme d'un RdP Continu Causal (RdPC<sup>2</sup>) est relativement directe (Figure 3.2).

Considérons l'exemple décrit sur la figure 3.2.2. La place  $P_1$  représente la variable

$A$ , dont l'évolution de la valeur est traduite par l'évolution du marquage de la place. La transition amont de la place  $P_1$ , transition  $T_A$ , a une vitesse de franchissement  $v_A$  déterminée par l'épisode  $E_A$  considéré (fonction affine par morceaux) :  $v_A(t) = \frac{dE_A(t)}{dt}$ . Le marquage de la place  $P_1$ , donc la valeur de la variable  $A$ , est alors exprimé par  $m_1(t) = E_A(t)$ . La place  $P_2$  représente la variable  $B$  : l'évolution du marquage correspond à la réponse qualitative de la variable  $RQ_B(t)$ . La vitesse de franchissement de la transition amont de la place  $P_2$ , transition  $T_{AB}$ , est définie comme une fonction du temps constante par morceaux. Cette vitesse dépend des paramètres de la fonction de transfert qualitative considérée  $FTQ(K, r, t_r)$ , et de la valeur de la variable  $A$  (épisodes du marquage de  $P_1$ ).

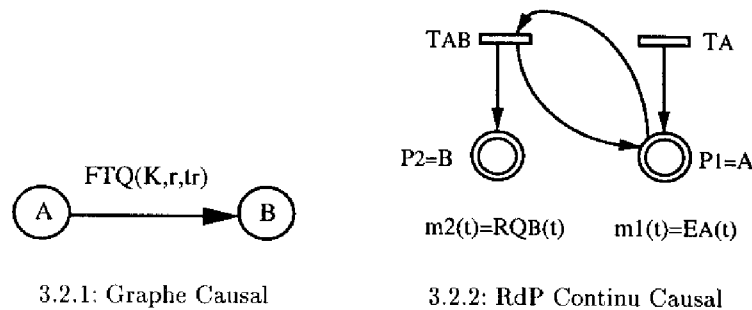


FIG. 3.2 - Exemple simple de RdP Continu Causal [Gomaa & Gentil 96a]

Une extension des RdP Continus Causaux, les RdP Continus Causaux Hybrides, a été définie pour prendre en compte l'influence de la partie discrète d'un système dynamique hybride (SDH) [Gomaa & Gentil 96b]. Le modèle résulte de l'intégration de trois RdP : le RdP Continu Causal qui modélise la partie continue du SDH, et deux RdP classiques pour modéliser la partie discrète et l'interface entre les deux parties.

Développé dans l'objectif d'établir un système d'aide à la décision basé sur un modèle qualitatif du système industriel, le modèle RdPC<sup>2</sup> est très spécifique. Il est utilisé notamment pour la prédiction du comportement (parcours «avant» sur le modèle) et le diagnostic de défauts (parcours «arrière» sur le modèle) ; les influences respectives des variables causes sur une variable effet permettent d'expliquer l'historique de la variable effet à travers les évolutions relatives de chacune de ses causes.

Des exemples d'application du modèle sur un système de régulation de température par échangeur de chaleur et sur deux colonnes pulsées couplées sont décrits respectivement dans [Gomaa & Gentil 96a] et [Gomaa & Gentil 95].

## 2.2 Les approches séparées

### 2.2.1 Le Grafchart et des systèmes d'équations

Dans un objectif de simulation, de supervision (déroulement de la recette), et d'aide à l'opérateur dans la prise de décision, le tout dans un environnement de type système expert, Arzen a abordé la modélisation d'un système hybride en couplant un modèle événementiel, le grafchart, avec des systèmes de règles et/ou d'équations.

Basé sur le Grafcet, le Grafchart a été défini pour faire face aux limitations du Grafcet (marquage binaire, jeton banalisé, etc.), tout en conservant un graphisme identique, donc connu de l'opérateur [Johnsson & Arzen 94]. En effet, le Grafcet étant une norme industrielle pour le contrôle séquentiel de bas niveau (programmation des automates), il est un des modèles vers lequel la communauté du génie des procédés s'est orientée pour représenter l'aspect discret des procédés hybrides [Rosenof & Ghosh 87] [Sawyer 93], c'est-à-dire les phases de démarrage et d'arrêt des procédés continus [Yazdi *et al.* 95], et les recettes [ISA/SP88 94].

Les modifications apportées font du Grafchart un modèle qui s'apparente aux RdP, ce qui justifie la présentation de ce modèle dans les applications basées RdP :

- les règles d'évolution du Grafchart en cas de choix (ou conflit) sont identiques à celles du RdP. Lorsque plusieurs transitions sont simultanément franchissables, elles sont simultanément franchies pour le Grafcet alors qu'une seule d'entre elles est franchie pour le Grafchart.
- le marquage n'est pas binaire, et les jetons sont individualisés par l'introduction de la notion de couleur. L'individualisation du jeton a donné lieu au Grafchart Haut-Niveau, mais nous parlerons du modèle sous son nom générique le Grafchart.

Le Grafchart décrit l'enchaînement logique des séquences d'activités, les phénomènes de synchronisation, et les phénomènes d'exception à l'aide des éléments de base du modèle : l'étape, la macro-étape (étape agrégée traduisant une séquence d'étapes plus élémentaires), la transition, et la transition d'exception. Une transition d'exception ne peut être associée qu'aux macro-étapes. Cette transition permet de désactiver le traitement effectué dans la macro-étape et de la quitter pour évoluer dans la séquence associée à la transition d'exception (Figure 3.3). Les transitions d'exception sont essentiellement utilisées pour exprimer la détection d'erreur et l'engagement conséquent de la commande vers une séquence de mise en sécurité ou de reprise.

L'évolution du marquage est effectuée selon les mêmes règles que les RdP. Les étapes auxquelles sont associées des actions contiennent un ensemble de règles traduisant les actions à exécuter lorsque l'étape est active. Par conséquent, à certaines étapes peut être associé selon le même principe, un ensemble d'équations qui sont résolues lorsque l'étape est active. Le résultat de ces calculs peut être impliqué dans la réceptivité associée à la transition de sortie de l'étape.

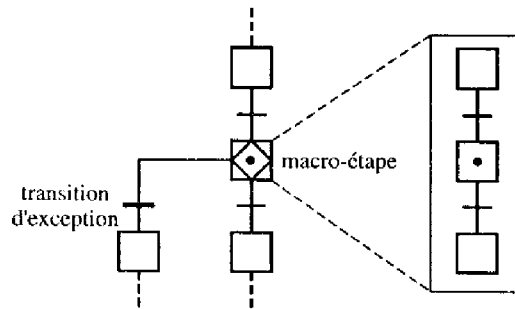


FIG. 3.3 -- Exemple de Grafchart

Quelques applications de supervision/commande basées sur le Grafchart ont été réalisées [Arzen 95] comme par exemple un système d'aide temps-réel de gestion de l'hydrogène dans une raffinerie de pétrole ou encore un simulateur pour l'apprentissage à la conduite d'un procédé de cristallisation du sucre. Elles sont implémentées dans l'environnement de développement de systèmes experts G2.

### 2.2.2 Les RdP Interprétés et des systèmes d'équations

Dans l'approche développée par Daubas [Daubas 94], la dichotomie entre les relations physiques que traduisent les équations d'une part, et les relations logiques qui décrivent les enchaînements entre les opérations d'autre part, est à la base de la séparation du modèle en deux parties : le modèle continu et le modèle discret. Partant du constat qu'à un haut niveau d'abstraction dans la description des mécanismes d'évolution du procédé, la représentation ne s'appuie que sur l'état et l'événement, les RdP ont été utilisés pour modéliser cet aspect discret du système. Les places du RdP représentent les états des ressources et les activités. Le comportement fin de ces dernières est simulé à l'aide d'un modèle continu ; à chaque activité est associé un système d'équations. Le système d'équations peut être de forme et de taille variables. Le marquage d'une place d'activité peut donc correspondre à une temporisation dont la durée est calculée par la résolution de son système d'équations. Les événements de temps (durée écoulée) ou d'état (franchissement d'un seuil par une certaine variable) sont associés aux transitions du RdP. L'occurrence de cet événement provoque le franchissement de la transition, l'évolution du marquage, et donc l'évolution du modèle continu. Lors du franchissement des transitions, le modèle à événements discrets modifie la structure du modèle continu. Les équations associées aux places dans lesquelles on enlève un jeton sont effacées et les équations associées aux places qui reçoivent un jeton sont ajoutées.

Lorsque le réseau de Petri est dans son état stable (toutes les transitions sensibilisées sont associées à des événements non présents pour le temps courant), le temps évolue et le système d'équations algèbro-différentielles associé aux places marquées est utilisé pour calculer l'évolution de l'état continu du procédé dans la configuration définie par le marquage courant. Il faut alors calculer la date de chaque événement associé à une

transition sensibilisée, les autres étant non significatifs dans la configuration courante.

Entre deux franchissements de transitions les événements de temps ou d'état portant sur les variables actives sont surveillés. Le premier événement détecté de cette liste d'événements possibles détermine la date de fin de la résolution du système d'équations et provoque un basculement vers la partie discrète par le franchissement de la transition correspondante. Le modèle discret gère le changement d'état en franchissant toutes les transitions franchissables au temps courant. La liaison entre les deux parties se fait donc par les événements et de manière synchrone (Figure 3.4) [Daubas *et al.* 94].

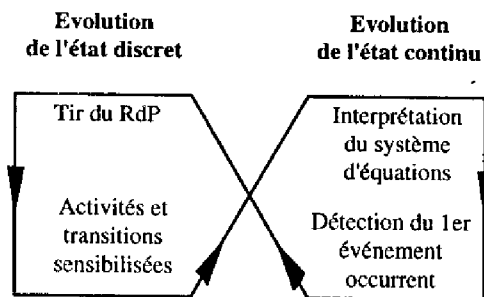


FIG. 3.4 – *Articulation des parties discrètes et continues [Daubas 94]*

Par conséquent le réseau de Petri peut être considéré comme un moniteur de systèmes d'équations, puisqu'il construit le système d'équations pour une configuration donnée et contrôle son intégration jusqu'au premier événement significatif.

Des places de contrôle sont définies dans le but de rendre explicites les décisions issues de la gestion de production. Ces places de contrôle, issues des travaux de Yamalidou [Yamalidou *et al.* 90], sont ajoutées au réseau afin de contrôler le franchissement de certaines transitions. Cette autorisation de franchissement d'une transition correspond à un choix de la date de lancement d'une activité. Les places de contrôle sont donc un moyen de prendre en compte l'influence de niveaux de gestion de production. Les aspects non déterministes, essentiellement les conflits liés à la disponibilité ou non des ressources, sont alors résolus par l'intermédiaire de ces places. Ces places de contrôle sont aussi bien utilisées pour émettre une requête de décision (l'arrivée d'un jeton dans la place traduit un appel à une procédure de prise de décision par exemple) que pour prendre en compte une décision (le jeton déposé dans la place traduit la décision qui a été prise).

Afin d'illustrer ces travaux, considérons l'exemple du procédé présenté sur la figure 3.5.1. L'installation est composée d'un stock de matière première ( $R_1$ ) et d'un réacteur ( $R_2$ ). La procédure opératoire à suivre (la recette) comprend trois étapes :

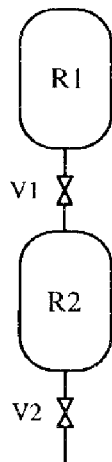
1. Transférer un lot de 100kg de matière de  $R_1$  vers  $R_2$  (le débit étant supposé être un paramètre,  $F_{12}$ ),
2. Effectuer une réaction sur le lot chargé dans  $R_2$ ,



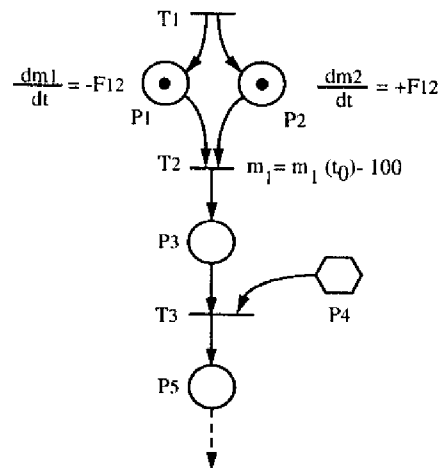
3. Evacuer le lot traité (vidange de  $R_2$ ).

Le réseau de Petri de la figure 3.5.2 correspond à la modélisation de la procédure opératoire décrite ci-dessus. Si nous décrivons les bilans matières totaux (modèle dynamique) pour les deux activités impliquées par le transfert, vidange de  $R_1$  et remplissage de  $R_2$ , deux équations sont associées aux places correspondantes  $P_1$  et  $P_2$ . L'équation associée à la place  $P_1$  traduit la diminution de la rétention massique dans  $R_1$ , notée  $m_1$ . Celle associée à la place  $P_2$  traduit l'augmentation de la rétention massique de  $R_2$ , notée  $m_2$ .

Lorsque la phase de transfert est activée, les places  $P_1$  et  $P_2$  sont marquées, et la transition  $T_2$  est sensibilisée. Le franchissement de  $T_2$  s'effectue dès que la condition de fin du transfert est vérifiée. Cette condition de fin de phase correspond à un événement d'état exprimé en fonction du soutirage de  $100\text{kg}$  du stock  $R_1$ :  $m_1(t) = m_1(t_0) - 100$ . Dès l'occurrence de cet événement la transition  $T_2$  est franchie, les jetons des places  $P_1$  et  $P_2$  sont retirés et un jeton est déposé dans la place  $P_3$ . Le marquage de  $P_3$  active la réaction sur le lot chargé dans  $R_2$ . Cette réaction correspondant à une phase purement batch, l'activité peut être considérée comme une activité d'attente de durée connue. Dans ce cas il n'y a pas d'évolution des variables d'état, seule la variable de temps évolue: l'événement correspond à un événement de temps. Un autre type d'achèvement de l'activité est possible: la décision de l'opérateur. C'est cette dernière qui est modélisée sur le RdP de la figure 3.5.2 par la place de contrôle  $P_4$ .



3.5.1: Le procédé



3.5.2: Le modèle

FIG. 3.5 – Exemple simple de modélisation

Cette approche a notamment été utilisée dans le cadre de l'ordonnancement. L'objectif n'est pas la recherche d'une solution optimale mais celui de la vérification de la faisabilité

d'un plan [Pages & Pingaud 94], par exemple élaboré par un opérateur humain.

Un cas industriel est traité dans [Pages & Pingaud 95]. Le plan de production est défini par des consignes sous la forme de campagnes de durées connues et durant lesquelles le produit est livré avec un débit constant en bout de chaîne. L'évolution du processus est alors simulée de manière rétroactive pour fournir un ordonnancement au plus tard, satisfaisant les contraintes fixées par le plan et compatible avec les équipements (tailles des réservoirs, débits, etc.).

### 2.2.3 Les RdP Colorés Temporisés à Décision Interprétés et un système d'équations

L'approche développée par Valentin [Valentin 93] repose également sur l'utilisation conjointe d'un modèle continu et d'un modèle discret. Cette approche séparée est induite par le type de fonctionnement particulier des ateliers de production considérés qui sont des systèmes de production mixtes composés de deux parties couplées, l'une effectuant une production d'un flux continu de matière et l'autre produisant par conditionnement un flux discret de pièces. Ces deux parties de l'atelier de production mixte sont synchronisées par l'intermédiaire de *transformateurs de flux*. Ces transformateurs de flux, appelés pivots, constituent l'interface entre les deux parties.

L'approche séparée a été choisie car les problèmes de conduite d'un système de production continu et d'un système de production discret sont différents : pour le premier l'optimisation du fonctionnement s'effectue sur les niveaux d'activité des unités de traitement, et pour le second par l'ordonnancement.

Le modèle discret utilisé est le Réseau de Petri Coloré Temporisé à Décision Interprété (RdPCT-DI) qui est basé sur les concepts des Réseaux de Petri Colorés Temporisés à Décision (RdPCT-D), et étendu pour prendre en compte les interactions avec un modèle continu. Chacun des modèles inclut une image comportementale de l'autre, cette image ne représentant que les interactions. La synchronisation de la partie discrète avec la partie continue est alors effectuée par l'intermédiaire de l'état de variables d'observation du modèle continu, et réciproquement par l'intermédiaire de variables d'observation de l'état du modèle discret (variables traduisant l'état des ressources, des produits, etc.). A tout moment, donc quelle que soit l'étape de production, le modèle continu est considéré dans son intégralité (le système global d'équations d'état), que l'état du système ait changé ou non.

Pour illustrer cette approche, considérons le cas traité de l'industrie papetière. La pâte à papier produite en continue est transformée en bobines ou ramettes de papier. Le transformateur de flux de type continu-discret (*machine pivot discontinue* Figure 3.6) est représenté par l'introduction d'un *stock discret fictif*, appelé *stock de synchronisation*, dans le modèle discret, et d'une *unité de traitement fictive*, dans le modèle continu. Le stock de synchronisation contient les pièces fictives à produire sur la machine pivot. Ce stock est alimenté d'une pièce à chaque fois que la machine pivot est libre et que la quantité de

matière continue nécessaire est disponible dans le bac de stockage.

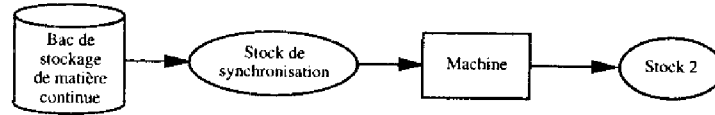


FIG. 3.6 – Schéma de la machine pivot

Le couplage entre les parties discrète et continue du modèle est représenté par :

- au sein du modèle discret : une transition source appelée *transition de synchronisation* dont la condition associée porte sur la variable d'état traduisant le niveau du bac de stockage de matière continue (Figure 3.7.1). Cette transition permet d'alimenter le stock fictif de nature discrète : une marque colorée représentant un élément de ce stock est introduite dans la place  $S_O$  sur le réseau de la figure 3.7.1. La couleur correspond au produit qui pourra être fabriqué.
- au sein du modèle continu : les équations d'état traduisant le bilan matière du bac de stockage sont modifiées pour prendre en compte l'unité de traitement fictive (Figure 3.7.2), et une variable logique traduit son état (marche ou hors-service). Cette variable logique, passant de «0» à «1» (ou inversement) suivant les phases, a pour conséquence de modifier la structure du système d'équations.

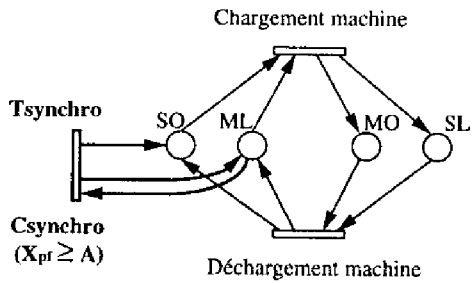
Si la machine pivot est en service :  $X_{pf}(k) = X_{pf}(k-1) - U(k-1) \quad \forall k$ ,

et si la machine pivot est hors service :  $X_{pf}(k) = X_{pf}(k-1) \quad \forall k$ ,

avec  $X_{pf}$  étant le niveau de matière dans le bac de stockage, et  $U$  le débit de soutirage.

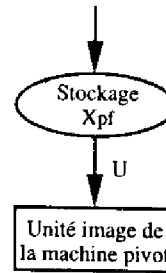
Le modèle développé est utilisé pour la conception d'un système d'aide à la décision dans la conduite d'ateliers de production mixtes (Figure 3.8).

Lorsqu'une dérive est constatée par rapport à l'objectif de production fixé, cette dérive est corrigée par un changement d'objectif ou de stratégie. L'aide à la décision consiste alors à évaluer d'autres stratégies de conduite à partir de l'état atteint en simulant l'application de celles-ci sur un horizon déterminé. Les perturbations considérées sont du genre panne d'une machine, manque de matière première, ou modification urgente du carnet de commande. Un ensemble d'indicateurs de fonctionnement permet d'évaluer la production de l'installation tels que les taux d'occupation des machines, l'état des stocks, les quantités de produits, etc.. La valeur de ces indicateurs est calculée à partir de l'observation de l'évolution dans le temps du marquage du RdPCT-DI. Les résultats temporels, issus de la simulation, sont alors présentés sous forme d'un diagramme de Gantt.



$\circ$  : emplacements occupés dans le stock  
 $\square$  : emplacements libres dans le stock  
 MO : machine occupée  
 ML : machine libre

3.7.1: Le modèle discret



3.7.2: La partie continue

FIG. 3.7 – Cas d'une machine pivot

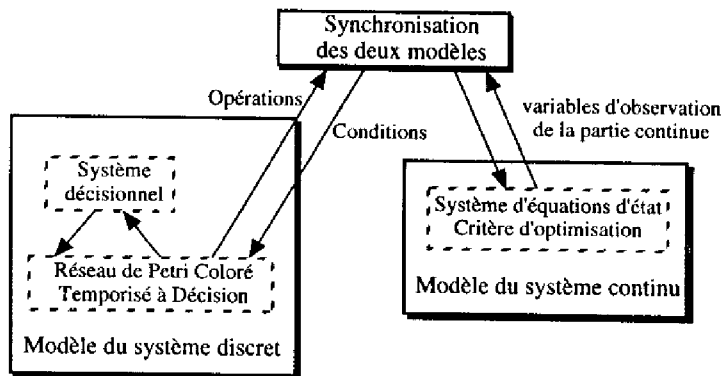


FIG. 3.8 – Structure du modèle global du système de production mixte

### 3 Comparaison des approches : conclusion

La comparaison des approches présentées doit nécessairement se situer selon deux axes : d'une part la technique de modélisation, d'autre part l'application visée du modèle, sachant que la nôtre est la supervision et la commande.

L'étude de la technique de modélisation basée sur un modèle intégré, montre une unification progressive (généralisation) des aspects continu et discret au sein d'un même formalisme au détriment de la délimitation entre le continu et le discret, ainsi qu'aux dépens du caractère général du modèle.

La délimitation entre la vision discrète et la vision continue du système est moins claire que lorsque l'on utilise deux formalismes ou deux modèles. Dans le RdP continu, l'évolution du marquage est entièrement basée sur les vitesses de franchissement des transitions : la notion d'événement se produisant lors d'une discontinuité par exemple, est représentée par la décomposition de la fonction continue exprimant la vitesse de franchissement en fonction affines par morceaux (vitesse fonction du temps ou constante), et n'est donc pas associée au franchissement d'une transition discrète. Il en est de même pour le RdP Hybride : les événements ne sont pas tous associés à des tirs de transitions discrètes, certains événements correspondent simplement à des changements de vitesse de franchissement sur les transitions continues.

Les modèles intégrant les deux aspects ont montré leur puissance mais ils sont limités par leur caractère spécifique même s'ils peuvent parfois être utilisés dans un cadre varié (procédés chimiques, mécaniques, etc.). Comme le souligne Daubas dans son étude des modèles hybrides [Daubas 94], un inconvénient du RdP Lot repose sur l'association d'une même loi à chaque place lot alors que dans un système il y a autant de lois d'évolution que d'opérations de transformation. La modélisation du convoyeur sans accumulation, dont la loi d'évolution des éléments transportés est différente, illustre cette remarque. Cette modélisation s'effectue par l'introduction d'une nouvelle place lot : la place lot sans accumulation. L'introduction d'un nouveau type de place est à chaque fois nécessaire pour la prise en compte d'une nouvelle loi d'évolution. Cette démarche est possible pour aborder un problème de modélisation spécifique mais ne peut pas être étendue à un cas général.

Certains principes d'évolution du marquage sont différents de ceux classiquement considérés sur les modèles basés RdP : considérons l'exemple du RdP Continu Causal pour lequel les vitesses de franchissement peuvent être négatives, ce qui correspond à un marquage qui «remonte» dans le réseau, ou encore l'exemple du RdP continu au sein duquel deux transitions en conflit peuvent être simultanément franchies. Dans un objectif d'utilisation du modèle en simulation, comme par exemple pour le dimensionnement de l'installation, la validation d'un plan de fabrication, ou l'aide à la décision, ces principes d'évolution sont possibles. Toutefois ils ne le sont plus dans un cadre d'utilisation en supervision et commande.

La technique de modélisation séparée, basée sur l'utilisation de deux modèles co-opérants, un modèle discret et un modèle continu, conserve un caractère plus général. L'utilisation des modèles propres à chaque aspect permet, notamment, de conserver le potentiel de modélisation dans chacun des domaines, comme le souligne d'ailleurs Valentin dans la conclusion de ses travaux [Valentin 93]. En effet, le fait qu'une marque ne corresponde pas à la valeur d'une variable continue permet de travailler sur l'ensemble des réels : une variable peut être négative alors qu'elle n'aurait pu l'être si elle correspondait à une marque du RdP. De plus, par l'association des lois aux places, tout type d'évolution des variables continues est pris en compte indépendamment de la structure du réseau : le problème du jeton qui «remonte» dans le réseau lors de la décroissance d'une variable (vitesse de franchissement négative d'une transition) ne se pose plus.

Le principe de séparation des deux aspects nous semble par conséquent être important. Les aspects continu et discret correspondent à deux mondes mathématiques différents offrant deux vues différentes du système. La vue discrète n'est pas seulement une abstraction du phénomène continu, mais les deux vues sont deux vues complémentaires d'un même environnement.

Nous avons donc opté pour la tendance séparée. Cependant, dans le cadre de la supervision des procédés discontinus, les approches séparées citées précédemment, l'approche «superviseur» et l'approche «moniteur d'équations», présentent quelques limitations, ou restrictions, quant à l'interface entre les deux aspects.

L'approche «superviseur» n'explicité l'interface entre les aspects continu et discret qu'entre le procédé et son superviseur de nature exclusivement discrète. Or, la problématique de la surveillance que nous avons traitée au chapitre précédent a clairement montré la nécessité d'un modèle continu au niveau de la supervision : par conséquent l'interface doit également être explicitée au sein du superviseur.

De plus, dans l'approche «superviseur», les signaux échangés à travers l'interface sont restreints et pré-déterminés : d'une part du superviseur vers le procédé, car la valeur de la commande continue appliquée est sélectionnée dans un ensemble fini de valeurs constantes, et d'autre part du procédé vers le superviseur, car les observations du procédé signalées sont sélectionnées dans l'ensemble des symboles appartenant à un alphabet prédéterminé. Le caractère dynamique de la configuration du procédé évoqué au premier chapitre requiert plutôt une définition dynamique des symboles à considérer au niveau de l'interface entre le procédé et son superviseur.

Les approches «moniteur d'équations» présentées ont pour inconvénient de «fusionner» dans un même état discret (une place pour les approches basées RdP, une étape pour le Grafchart) : l'état discret de la commande, c'est à dire l'étape active de la recette, l'état discret du procédé, c'est à dire la configuration active de l'installation, et l'ensemble des équations actives, c'est à dire celles traduisant les phénomènes continus ayant lieu sur le procédé (sauf dans [Valentin 93], où la notion d'équations actives n'existe pas : à tout

moment le modèle continu est considéré dans son intégralité, que l'état du système ait changé ou non).

L'orientation de nos travaux est fondée sur le respect de l'identité de chaque monde car, rappelons-le, ils correspondent à deux environnements mathématiques différents, deux vues différentes et complémentaires du système. Il s'agit donc d'une approche de nature séparée basée sur l'utilisation conjointe d'un modèle à événements discrets et d'équations algébro-différentielles. Le modèle à événements discrets retenu est les réseaux de Petri.

Si nous retenons bien la notion de superviseur, l'interface entre le continu et le discret n'est pas limitée uniquement à l'interaction entre le superviseur et un modèle continu représentant le système et sa commande locale. Une interaction entre une partie continue et une partie discrète est également possible au sein du superviseur. Nous retiendrons pour cela l'approche «moniteur d'équations».

Notre approche pour la modélisation de la supervision des procédés discontinus nous amène à définir un modèle hybride indispensable, notamment pour la surveillance comme nous l'avons précédemment constaté. Ce modèle hybride sur lequel est fondée la supervision sera structuré selon trois vues : une vue discrète correspondant aux changements de configuration requis par la recette (la commande), une vue discrète établissant une représentation «qualitative» de l'état du procédé (les configurations de l'installation), et une vue continue traduisant les phénomènes continus du procédé.

## Deuxième Partie

### Une nouvelle approche pour la conduite des procédés discontinus





## Chapitre 4

# Principes de l'approche

Après avoir présenté le cadre de notre étude sous ses divers aspects nous allons tout d'abord mener quelques réflexions concernant l'influence de la notion de configuration dynamique sur la conception d'un système de commande/supervision, notamment sur l'architecture répartie qui en est le support.

Au sein d'une architecture répartie la communication entre les différentes entités du système est un facteur influant sur sa performance. Cette communication est donc une activité fondamentale pour le bon fonctionnement d'une application de commande temps-réel répartie, car elle doit respecter un ensemble de contraintes temporelles dont celles imposées par les évolutions dynamiques du procédé [Bergé 96]. L'architecture informatique et la communication temps-réel sont par conséquent les points que nous étudierons dans une deuxième section.

Nous aborderons ensuite la supervision des procédés discontinus en considérant, pour plus de clarté, seulement deux niveaux : d'une part un niveau de supervision et d'autre part un niveau de commande locale piloté par le précédent. Le passage d'une supervision en «boucle ouverte» à une supervision en «boucle fermée» nous amènera alors à l'introduction d'une interface appelée le *Générateur d'Événements*. Nous terminerons ce chapitre en évoquant la nécessité d'un modèle hybride pour la supervision.

### 1 Réflexions sur l'influence de la notion de configuration dynamique

La flexibilité de production ne repose pas seulement sur la flexibilité de l'installation mais également sur la capacité du système de conduite à gérer cette flexibilité. La **modularité** du système est par conséquent essentielle afin de s'adapter au besoin de **configuration dynamique** du procédé. Nous avons en effet précisé que l'évolution de la production implique de nombreux changements de configuration sur le procédé et de fréquents changements de stratégie de conduite.

Nous allons donc aborder l'influence de la notion de configuration dynamique sur la modularité, la hiérarchisation, et la répartition de l'application.

## 1.1 La commande modulaire

La modularité de la commande est un des facteurs de flexibilité. Elle repose, par définition (cf Larrousse), sur "la décomposition en entités fonctionnelles de base susceptibles d'être utilisées conjointement". Dans la «norme», ainsi que dans les travaux que nous avons présentés, la modularité est basée sur la hiérarchisation de l'aspect procédural, étroitement liée à la hiérarchisation de l'équipement.

Or cette hiérarchisation n'est pas toujours applicable comme dans le manufacturier. En effet, la notion de configuration dynamique implique un couplage dynamique des équipements (éléments et/ou modules de contrôle dans la norme), qui n'est pas nécessairement strictement hiérarchique (pas de composition successive).

Nous allons traiter trois exemples dont les deux derniers sont basés sur les procédés que nous avons présentés au premier chapitre : il s'agit d'un procédé de distillation batch, et d'un réacteur batch exothermique multifluide. Le premier exemple illustre un cas simple, car purement discret, pour lequel une hiérarchie procédurale semble pouvoir être établie par composition successive, il paraît donc «contradictoire» avec les propos précédents. Cependant, les exemples suivants nous montreront qu'il est préférable de raisonner en terme de configurations au sein desquelles sont alors définies les opérations ou les fonctions. Nous ferons également apparaître que la notion de configuration se retrouve à deux niveaux d'abstraction : la configuration de l'installation par connexion des unités entre elles, et la configuration des équipements au sein d'une unité selon la fonction exécutée.

### 1.1.1 Un exemple simple

Considérons l'installation représentée sur la figure 4.1.

Il est possible de composer un module d'équipement, en associant la vanne  $V_1$  et la pompe  $P_1$ <sup>1</sup>. Il s'agit d'un des modules associés au réacteur  $R_1$ . La séquence opératoire "ouvrir la vanne  $V_1$ " puis "activer la pompe  $P_1$ " constitue la fonction élémentaire de déchargement du réacteur  $R_1$ . Le lien entre  $V_1$  et  $P_1$  est *permanent* puisque le couplage reste figé quelle que soit l'évolution de la conduite du procédé, et quelle que soit sa configuration. De manière similaire, la vanne  $V_2$  peut composer à elle seule un des modules d'équipement du réacteur  $R_2$ , et supporter la fonction élémentaire de chargement du réacteur  $R_2$ .

La phase de transfert de matière de  $R_1$  à  $R_2$ , qui correspond à une configuration dans laquelle  $R_1$  et  $R_2$  sont connectés, s'effectue par la composition de leurs fonctions respectives de déchargement et de chargement. Le lien entre  $\{V_1, P_1\}$  et  $\{V_2\}$  est *temporaire* puisqu'il est établi dynamiquement et qu'il n'existe que pendant la durée du transfert. Cependant, la définition de la fonction permettant ce transfert de matière ne se limite pas à cette

---

1. association de la pompe  $P_1$  au réacteur  $R_1$  par rapport au sens du flux.

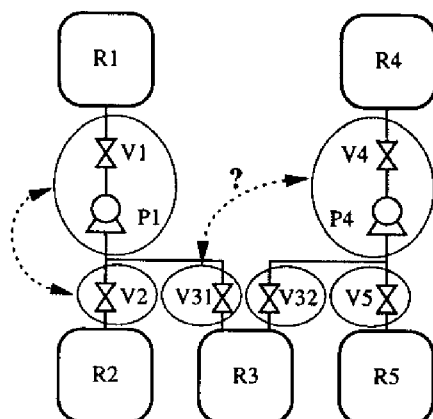


FIG. 4.1 – Exemple de constitution de fonctions élémentaires

composition : toutes les vannes aux *frontières* de cette connexion doivent également être positionnées et maintenues, c'est-à-dire *verrouillées*, dans un état donné durant l'activité. La vanne  $V_{31}$ , par exemple, doit être fermée et se trouve par conséquent impliquée dans la configuration.

Ces exemples de composition de modules d'équipement offrant un ensemble fini de fonctions élémentaires, reposent sur la composition d'équipements selon la *structure physique* du procédé : six modules d'équipement peuvent ainsi être définis sur l'exemple de la figure 4.1. Les couplages possibles entre ces modules ne dépendent pas que de la *structure logique* du procédé. Par exemple, les modules  $\{V_4, P_4\}$  et  $\{V_{31}\}$  semblent, d'après la structure logique, ne pas avoir de lien. Et pourtant, si  $R_3$  est un appareil au sein duquel est effectué un mélange en parallèle,  $\{V_1, P_1\}$ ,  $\{V_{31}\}$ ,  $\{V_{32}\}$ ,  $\{V_4, P_4\}$  se trouvent tous couplés afin d'assurer un mélange dans des proportions et profils définis.

A travers cet exemple, les compositions effectuées pour définir à la fois les modules d'équipement et les fonctions élémentaires qu'ils supportent, puis l'expression de leurs couplages par des liens temporaires, ne sont pas évidentes mais restent possibles dans la mesure où elles reposent principalement sur l'expression des relations de précedence ou de synchronisation d'actions élémentaires de nature discrète, comme dans le cas du manufacturier.

Or, dans le cadre des procédés discontinus, il est également nécessaire de coupler un ensemble de boucles de régulation continues selon une loi de commande déterminée, souvent exprimée par un système multivariable. Cette loi de commande a parfois un domaine de validité déterminé par des critères de performance (la concentration de la solution par exemple), et un changement de loi de commande peut s'avérer nécessaire en cours d'exécution d'une fonction : donc le couplage entre les boucles évolue. Par conséquent, il n'est pas toujours possible de définir une fonction élémentaire selon la démarche précédente.

C'est ce que nous allons illustrer à travers les deux exemples qui suivent.

### 1.1.2 La conduite d'un procédé de distillation batch

Cet exemple a été présenté au premier chapitre (Figure 1.1). Nous avons précisé que la conduite optimale de la distillation d'un lot (séparation des constituants *A* et *B* pour l'exemple traité) implique de changer de stratégie à des instants de temps particuliers qui sont déterminés en fonction de la concentration du produit considéré. Détaillons son fonctionnement.

Tout d'abord, il y a chargement d'un lot de produit dans la colonne, cette dernière est connectée au réservoir d'alimentation.

La première étape consiste alors à "monter la colonne en spécification" en utilisant un reflux presque total. La configuration des appareils et des équipements qui en résulte est décrite sur la figure 4.2.1. La quantité de matière condensée est en partie stockée dans le condenseur et le reste est réinjecté dans la colonne. La proportion est établie par une boucle de commande au sein de laquelle la consigne de la vanne  $V_1$  dépend de la mesure de niveau du condenseur (travail à niveau constant dans le condenseur). La colonne et le condenseur sont isolés dans cette configuration (les vannes  $V_2$  et  $V_3$  sont fermées).

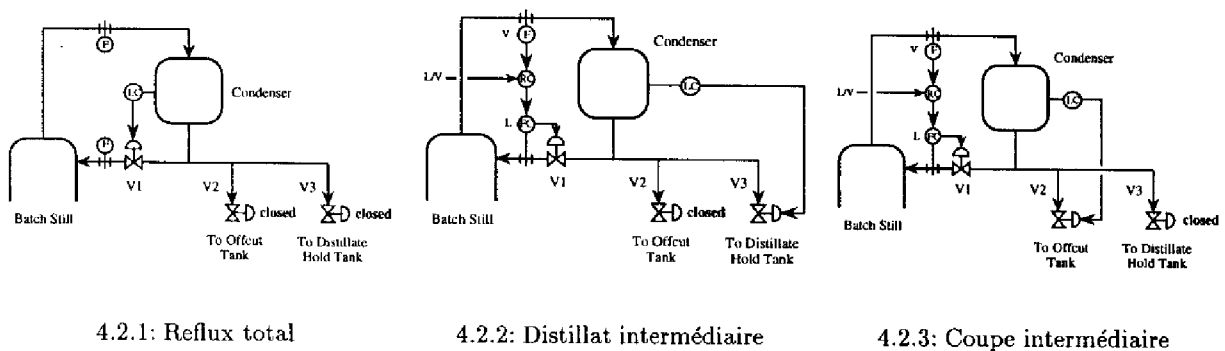


FIG. 4.2 – Les différentes stratégies de conduite [Macchietto et al. 89]

Dès l'obtention d'une certaine concentration en constituant *A* dans le condenseur, la stratégie de conduite est changée. Une partie du distillat est dirigée vers le stock intégrateur et un reflux variable est effectué pour le maintien de la concentration. La configuration correspondante est représentée sur la figure 4.2.2. Deux boucles de commande sont établies. La première agit sur le taux de reflux : la consigne de la vanne  $V_1$  (débit de liquide réinjecté) est calculée par une loi de commande en fonction du taux de reflux choisi et du débit de vapeur en tête de la colonne. La seconde agit sur le stockage du

distillat : la consigne de la vanne  $V_3$  dépend de la variation de niveau dans le condenseur (travail à niveau constant). La colonne, le condenseur, et le réservoir intégrateur (distillat) sont couplés dans cette configuration (seule la vanne  $V_2$  est fermée).

Lorsque le constituant  $A$  est quasiment épuisé, la concentration en  $A$  ne peut plus être maintenue en tête de colonne. Il est nécessaire de changer à nouveau de stratégie de conduite : l'objectif est de concentrer le constituant  $B$ . Cela implique une période durant laquelle la solution en tête de colonne doit être coupée car elle contient à la fois du  $A$  et du  $B$ . Cette période de coupe (fin de séparation de  $A$  et début de séparation de  $B$ ), qui est maintenue tant que la concentration en constituant  $B$  en pied de cuve n'atteint pas la valeur requise, correspond à la configuration représentée sur la figure 4.2.3. Elle fait apparaître deux boucles de commande comme pour la stratégie précédente, mais cette fois-ci le distillat est dirigé vers le réservoir de coupe. Ce sont la colonne, le condenseur, et le réservoir de coupe qui se retrouvent couplés dans cette configuration (seule la vanne  $V_3$  est fermée).

Lorsque la concentration du pied de cuve en constituant  $B$  atteint la valeur désirée, la distillation est arrêtée mais le chauffage est maintenu pour éviter la chute des rétentions sur les plateaux et la vidange du pied de cuve dans le réservoir  $T_6$  (Figure 1.1) est effectuée. Dans cette configuration la colonne n'est couplée qu'au réservoir  $T_6$ .

La quantité de matière stockée dans le réservoir de coupe sera transférée dans la colonne pour être mélangée au chargement du prochain lot. Une fois le lot chargé l'opération de séparation est à nouveau lancée. Après quelques lots la distillation batch atteint quasiment un régime permanent au sens où les concentrations et les volumes obtenus deviennent quasiment identiques.

A travers cet exemple il est possible de distinguer deux types de configurations en définissant une unité constituée par l'association, dans une même ressource, de la colonne, du condenseur, du réservoir de coupe, et du réservoir intermédiaire (distillat). D'une part, il y a la configuration de l'installation, qui repose sur la connexion de l'unité avec son environnement, composé dans ce cas des réservoirs d'alimentation et de stockage des produits finaux. D'autre part, à un niveau d'abstraction plus bas, il y a la configuration au sein de l'unité, qui repose sur la connexion entre les appareils et les équipements dont elle est constituée.

Les fonctions impliquant un transfert de matière pour charger ou décharger l'unité nous ramènent à l'exemple précédent. Par contre les fonctions de transformation de la matière au sein de l'unité doivent être définies par rapport à ses configurations internes, pour lesquelles les boucles de régulation et leurs couplages sont différents.

L'exemple suivant, dont la complexité des couplages est plus importante car il y a combinaison d'équipements discrets et continus, met encore plus en évidence la nécessité d'associer fonction et configuration.

### 1.1.3 La conduite d'un réacteur batch multifluide

Cet exemple du réacteur exothermique multifluide a également été présenté au premier chapitre (Figure 1.2). Le réacteur est supposé chargé, il est donc dans une configuration où il est isolé de l'environnement (aucune connexion avec des unités amont et aval), et nous nous intéressons à la transformation de la matière au sein du réacteur. Rappelons que cette transformation repose sur la conduite thermique de la réaction.

Le choix du fluide à utiliser pour suivre le profil de température imposé est déterminé par le calcul du flux thermique à appliquer : il s'agit d'une conduite thermique du réacteur par analyse des flux thermiques [Louleh 96]. Chaque fluide caloporteur est caractérisé par les limites (maximum et minimum) du flux thermique qu'il peut fournir. Lorsque le choix du fluide est déterminé, si le fluide en cours ne convient plus il faut effectuer une commutation qui peut exiger une purge de la double enveloppe pour des raisons de compatibilité ou de choc thermique à éviter, puis appliquer la (ou les) consigne(s) calculée(s) aux différentes boucles de régulation impliquées dans la configuration choisie.

Par exemple considérons qu'il soit nécessaire de passer d'une phase de maintien à une température élevée, à une phase de refroidissement. Admettons que la première s'effectue à l'aide du fluide intermédiaire (eau et vapeur) en mode recyclé, et que la seconde exige l'utilisation de l'eau glycolée.

Dans la première configuration (phase de maintien), les vannes discrètes  $V_3$ ,  $V_4$ ,  $V_8$ ,  $V_9$  sont ouvertes (les autres sont fermées), et les vannes  $V_A$ ,  $V_B$ , la vanne de coulée, et la pompe sont couplées.

La seconde configuration (phase de refroidissement) se traduit par les vannes discrètes  $V_1$ ,  $V_2$  ouvertes (toutes les autres sont fermées), et la vanne  $V_C$  et la vanne de coulée couplées.

Pour le passage de la première à la seconde configuration, c'est-à-dire lors du changement de fluide, il est nécessaire de purger le circuit : il faut fermer certaines vannes selon une séquence déterminée qui dépend de la configuration précédente, arrêter la pompe dans ce cas, et arrêter la coulée si elle est en cours à cause de l'exothermicité de la réaction (cette dernière n'est pas contrôlée lors de la purge). Puis il faut passer dans la configuration de purge jusqu'à ce que la double-enveloppe soit vide.

Cet exemple met à nouveau en évidence la correspondance entre fonction élémentaire et configuration. Chaque fonction définit un couplage entre un ensemble de boucles de régulation et d'actions discrètes, qui correspond à une configuration interne à l'unité.

Revenons maintenant sur l'influence, d'une manière générale, de cette notion de configuration dynamique sur la conception du système de commande/supervision, en considérant l'aspect architectural.

## 1.2 La commande répartie

Nous avons vu que la maîtrise des différentes activités qui se déroulent au niveau du procédé, de nature hybride, passe par l'utilisation conjointe des techniques d'automatisation séquentielle, afin de gérer la séquence d'événements traduisant les phases discrètes successives, et continue, pour le contrôle de trajectoires de variables d'états continues. La commande se traduit alors par des actions discrètes supportées par des automates programmables, et par des boucles de régulation locales supportées et/ou coordonnées par des contrôleurs.

Il n'est donc pas envisageable de baser la structure du système de contrôle sur une architecture centralisée supportant la globalité de l'application logicielle. Par conséquent, l'application est fréquemment répartie et coordonnée par un superviseur.

D'autre part, la modularité de la commande impose de pouvoir coupler dynamiquement des boucles de régulation et des actions discrètes qui ne sont pas nécessairement liées à un équipement particulier (les opérations de nettoyage en sont un exemple puisqu'elles s'appuient sur l'ensemble de l'installation). Une application logicielle, une fonction de l'application globale par exemple, qui peut être amenée à travailler sur un ensemble de variables continues et discrètes dynamiquement établi, doit pouvoir accéder à toute variable à tout moment.

En conséquence, associer directement les automates et les contrôleurs locaux à des équipements particuliers (Figure 4.3), va à l'encontre de la flexibilité requise pour répondre au besoin de configuration dynamique. Il faut donc éviter la perte de flexibilité qui serait induite par une répartition figée des capteurs et actionneurs sur l'ensemble des automates et des contrôleurs. Ainsi les communications et les synchronisations complexes entre ces organes de commande locale dans le cadre de leurs interactions [Akamatsu *et al.* 89], seront également limitées.

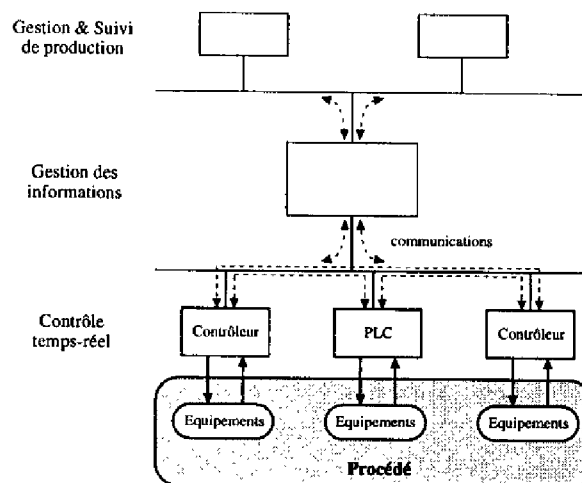


FIG. 4.3 – Association directe d'organes de commande aux équipements



### 1.3 La commande hiérarchisée

Revenons, dans le cadre de cette section uniquement, à une hiérarchie classique constituée de trois niveaux (supervision, coordination, commande locale) [Andreu *et al.* 94a].

La hiérarchie fonctionnelle que nous avons décrite est proche de celle classiquement utilisée pour les systèmes manufacturiers. Cependant cette hiérarchie est basée sur l'agrégation du temps et des fonctions, mais pas des données. En effet, la hiérarchie fonctionnelle de la «norme» (Figure 2.3) montre clairement que toutes les informations sont accessibles à chaque niveau, ce qui la remet partiellement en cause.

Le problème de modularité que nous avons évoqué confère un rôle important à la coordination qui doit, rappelons le, coordonner un ensemble d'actions discrètes et de boucles de régulation dans une configuration dynamiquement établie.

Dans les systèmes de commande/supervision présentés au second chapitre, les nombreux échanges qui passent par la base de données font que les interactions entre les niveaux ne sont pas explicites. La coordination ne semble pas vraiment exister en tant que niveau mais uniquement en tant que fonction.

Le contrôle étant réparti de manière figée (organes de commande directement associés à un ensemble d'équipements), la coordination des phases repose sur la communication entre les organes de commande, ce qui induit de nombreux échanges et synchronisations entre eux.

Une coordination supportée par l'ensemble des organes de commande locale n'est valable que pour les procédés dont la flexibilité et/ou la polyvalence est limitée. Pour une installation comportant une simple ligne de fabrication, coordonner les phases permettant un transfert de matière entre deux unités est relativement simple. Par contre, pour une installation flexible au sein de laquelle un ensemble de configurations (couplages entre unités) sont possibles par définition, la complexité de coordination d'un ensemble de phases est renforcée lorsqu'elle est ainsi répartie sur l'ensemble des organes de commande.

Par conséquent, l'existence d'un niveau de coordination nous semble indispensable. Elle permet une meilleure répartition des rôles, du moins de manière plus flexible. Le changement de configuration du procédé (liens temporaires entre les unités) est établi par la supervision, et le changement de configuration au sein d'une unité (liens temporaires entre les appareils et les équipements la constituant) est établi par le niveau de coordination lui-même, qui agit alors comme une «supervision locale» (supervision d'une activité ayant lieu sur une unité).

A titre d'exemple nous pouvons reprendre le cas de la conduite du réacteur exothermique multifluide. L'enchaînement des configurations nécessaire à l'exécution de la recette dépend de la supervision : chargement du réacteur (réacteur connecté à une unité amont), transformation du lot (réacteur isolé), et déchargement du réacteur (réacteur connecté à une unité aval). Lors de la transformation du lot, le changement de fluide, qui revient à

changer la configuration des équipements, est du ressort de la coordination. Elle décide des instants de commutation, et agit en ce sens comme une «supervision locale».

## 1.4 Conclusion

Sur un procédé discontinu, la notion de configuration est essentielle, au point que fonctions et configurations sont apparues indissociables, à travers les exemples étudiés. Rappelons par exemple que la définition d'une fonction pour le transfert d'un lot entre deux unités nécessite d'exprimer les verrouillages aux frontières de la connexion entre ces unités.

Exécuter la séquence d'opérations d'une recette, ainsi que les différentes phases dont sont constituées ces opérations, peut être comparé à établir une séquence de configurations au sein desquelles les flux de matière et d'énergie sont pilotés selon une stratégie déterminée. Supervision et coordination sont donc chargées de la gestion des configurations, du procédé et/ou de l'unité, et de l'exécution d'une stratégie de commande dans la configuration courante.

Appliquer une stratégie de commande dans une configuration dynamiquement établie, impliquant par conséquent un ensemble d'équipements défini dynamiquement, n'est guère possible en se basant sur une architecture répartie où les organes de commande locale sont directement associés aux équipements, comme cela se fait souvent dans le cas des procédés complètement discrets. Il est apparu nécessaire d'adapter l'architecture. C'est l'objet de la section qui suit.

## 2 L'architecture du système de commande/supervision

Notre système de commande/supervision s'appuie sur le "niveau bas" existant : ce niveau bas est constitué de l'instrumentation (capteurs et actionneurs), et de l'ensemble hétérogène de systèmes numériques de commande tels que les automates programmables industriels et les régulateurs continus, pilotés par le superviseur. L'utilisation effective de ces organes d'automatisation nécessite leur connexion par l'intermédiaire d'un système de communication, qui doit permettre, comme nous l'avons souligné, d'accéder à toutes les variables du procédé. La connexion la plus directe passe par l'utilisation d'un réseau de communication unique (Figure 4.4).

Cependant, le réseau de communication doit nécessairement assurer que les contraintes de cohérence temporelle et spatiale de la base de données temps réel<sup>2</sup> sont respectées [Rosa *et al.* 94].

---

2. lien logique entre les applications

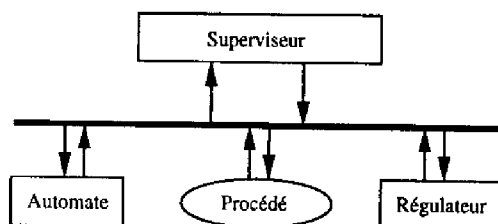


FIG. 4.4 – Architecture organisée autour d'un réseau unique

La contrainte de cohérence temporelle est double : cohérence temporelle de production et cohérence temporelle de transmission [Noury & Laine 92]. La cohérence temporelle de production signifie que les données doivent être produites dans une même fenêtre de temps : cette contrainte assure que les algorithmes des applications utilisatrices de ces données travaillent sur des valeurs de même date. La cohérence temporelle de transmission impose une perception correcte de l'environnement : l'image d'une variable d'état continue du procédé doit être mise à jour périodiquement selon une fréquence convenable assurant ainsi l'adaptation à la dynamique du procédé. Dans le cas contraire, un état intermédiaire de la variable ne serait pas reflété par les observations et serait donc perdu [Kopetz 92]. La contrainte de cohérence spatiale impose une vue cohérente du procédé dans tout le système de commande : l'image d'une variable doit être identique dans toutes les entités de contrôle du procédé.

Or, dans le cas d'une architecture basée sur un réseau unique, les mesures échantillonnées des variables d'état continues transmises périodiquement (mesures des capteurs), les commandes émises par les organes de commande de bas niveau (contrôleurs et automates) vers les actionneurs, ainsi que les échanges événementiels entre les organes de commande et le superviseur transmis de manière asynchrone, transitent sur le même bus.

Afin de respecter les contraintes de cohérence, il est préférable d'éviter le transfert simultané sur un même réseau de données périodiques et de messages aperiodiques : l'aspect aperiodique ne permet pas une prédéfinition et dimensionnement du réseau, donc le respect des contraintes est difficile à garantir. L'utilisation d'un réseau unique n'est donc pas adaptée [Pleinevaux & Decotignie 88] (risque de surcharge du réseau).

Par conséquent la communication doit être organisée autour de deux réseaux afin d'assurer un échange performant d'informations entre les organes de commande de bas niveau et le superviseur. L'architecture proposée est basée sur l'utilisation conjointe d'un bus à caractère « continu », c'est-à-dire dédié aux variables continues échantillonnées, pour la communication entre les organes de commande et les capteurs du procédé (le bus de terrain<sup>3</sup>), et d'un bus à caractère « discret » (les messages sont des événements) pour la communication entre le superviseur et les organes de commande (le bus d'atelier).

3. la possibilité d'une transmission aperiodique « exceptionnelle », comme un message d'alarme d'un actionneur, n'est pas exclue. Le réseau *FIP* le permet par exemple.

L'architecture résultante est représentée sur la figure 4.5.

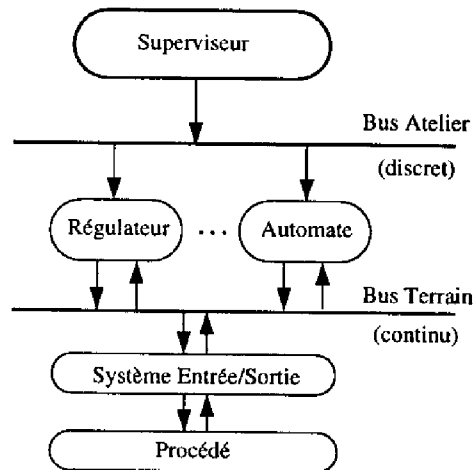


FIG. 4.5 – Architecture informatique

Si l'on raisonne en terme de contraintes temps réel, ce type d'architecture s'impose car le superviseur n'a pas les mêmes contraintes strictes que celles de la commande locale (horizon temporel différent), et il se situe de plus à un niveau d'abstraction et de compétence différent (objectif et perception du procédé différents). Les systèmes de contrôle distribué sont d'ailleurs généralement basés sur une architecture constituée d'une hiérarchie de réseaux locaux industriels [Decotignie & Pleinevaux 93].

### 3 Supervision des procédés hybrides

Dans le cas des systèmes de production purement continus fonctionnant en régime permanent la supervision est essentiellement chargée de surveiller le fonctionnement pour détecter tout écart par rapport au régime permanent optimal. Lorsqu'un écart est détecté, un diagnostic doit être effectué et une action de compensation doit être faite par l'intermédiaire des régulateurs locaux. Dans le cas des procédés de fabrication hybrides, la supervision est également chargée de détecter la fin de chaque opération de la recette, c'est-à-dire de chaque configuration du procédé. La supervision doit alors commander le changement de configuration souhaité (spécifié par la recette), par l'intermédiaire des automatismes locaux. Cela induit des régimes transitoires spécifiques qui doivent être pris en charge par les régulateurs locaux sous le contrôle de la supervision.

De façon générale, la supervision est chargée de maintenir le procédé près de son point de fonctionnement nominal, et d'assurer le bon déroulement de la production. Ceci implique l'optimisation et la surveillance du fonctionnement. Ces deux objectifs se basent sur l'évolution du procédé, qui est observée à travers un ensemble de variables pertinentes,

discrètes et continues. Le modèle sur lequel est fondée la surveillance doit naturellement être hybride, c'est-à-dire exprimer l'état discret et l'état continu du procédé, ainsi que leur dynamique, ce qui nous replonge dans la problématique détaillée dans le chapitre sur la modélisation.

Nous allons d'abord présenter une «construction» progressive d'un système de commande/supervision reposant sur l'architecture informatique précédente, puis nous aborderons la prise en compte de l'aspect continu au sein de la supervision.

### 3.1 Supervision en boucle ouverte

Considérons d'abord le cas de la supervision en boucle ouverte. Soit  $x$  le vecteur des variables d'état, il est formé de  $x_c$  qui est le vecteur des composantes continues (variables prenant leurs valeurs dans l'ensemble des réels comme le volume, la température, etc.) et  $x_d$  qui est le vecteur des composantes discrètes (capteurs et actionneurs binaires, configurations du procédé, etc.).

Une recette élémentaire est décrite par le réseau de Petri de la figure 4.6. Partant du réacteur vide, 100kg de produit  $A$  sont chargés, puis 50kg de produit  $B$ . Le mélange subit alors un traitement physico-chimique : “mélanger et chauffer pendant 10 minutes”. La réaction terminée, le produit final est disponible dans le réacteur et peut être utilisé (soutirage du produit). Cette recette fait apparaître 4 phases qui correspondent à 4 configurations différentes de l'installation : réacteur connecté au réservoir de matière première  $A$ , réacteur connecté au réservoir de matière première  $B$ , réacteur isolé pendant la réaction, et enfin réacteur connecté à une unité aval pour le soutirage du produit final.

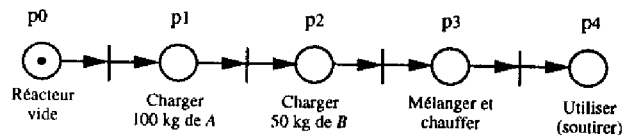


FIG. 4.6 – Une recette élémentaire

La figure 4.7 montre le schéma de la supervision en boucle ouverte. L'entrée du superviseur est un plan de fabrication  $\Sigma_L$ , c'est-à-dire une suite d'ordres de lancement de lots  $l \in \Sigma_L$  avec leurs dates et les équipements requis. Nous supposons évidemment que les recettes des produits à fabriquer (les recettes principales) ont été décrites au préalable. Le premier rôle du superviseur est d'exécuter ces recettes, donc de générer des recettes de contrôle pour chaque lot, à partir du plan de fabrication  $\Sigma_L$ , en tenant compte des équipements virtuellement alloués par le plan pour chaque opération.

Les recettes de contrôle sont traduites sous la forme de deux types de messages,  $U_c, U_d \in \Sigma_U$ , qui représentent l'ensemble des requêtes envoyées par le superviseur aux organes de commande. Les messages  $U_c$  sont des consignes envoyées aux régulateurs

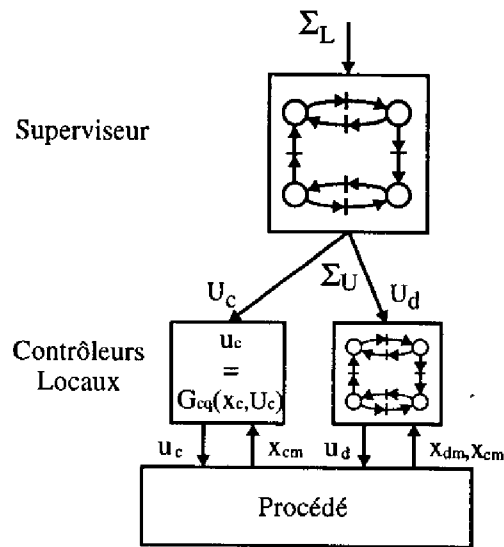


FIG. 4.7 - Supervision en boucle ouverte

continus, et les messages  $U_d$  sont des événements qui provoquent des changements d'états des automates programmables industriels. Ces automates, connectés au procédé, reconfigurent physiquement l'installation lors de chaque changement de configuration par des ouvertures et des fermetures de vannes «tout-ou-rien», entre autres. Ces messages peuvent être considérés comme des commandes symboliques traduites en signaux de contrôle par les organes de commande locaux [Antsaklis *et al.* 93].

Les régulateurs continus et les automates programmables industriels fonctionnent en boucle fermée.

Les régulateurs continus ont accès aux variables d'état continues  $x_{cm}$  mesurées par des capteurs. Ils reconstituent l'état continu,

$$x_c = F_c(x_{cm}) \quad (4.1)$$

et génèrent périodiquement les commandes continues :

$$u_c = G_{c_q}(x_c, U_c) \quad (4.2)$$

Les lois de commande continue  $G_{c_q}$  sont supposées établies par rapport à leur domaine de validité (domaine pouvant être exprimé par une région dans l'espace d'état ou simplement par un seuil sur une variable). En plus de préciser la consigne à suivre, le superviseur a pour rôle de sélectionner la loi à appliquer lorsque qu'il y a changement de domaine, selon la commutation de mode de contrôle qui a été spécifiée (changement de la loi active  $q$ ). Il sélectionne ainsi différentes trajectoires du système continu (commutation de structure ou

modification des paramètres). Le message  $U_c$  ne précise plus uniquement la consigne mais également le mode de contrôle  $q$ .

Nous n'aborderons pas les problèmes de régulation car nous supposons<sup>4</sup> que le fonctionnement à l'intérieur des différents domaines ressort de l'approche continue classique et peut être traité au niveau de la commande locale.

Les automates programmables industriels détectent les variations de variables d'état discrètes mesurées  $x_{dm}$  et les franchissements de seuils par les variables d'état continues  $x_{cm}$ . Par un ensemble de fonctions logiques combinatoires ( $B$ ), ils déterminent l'occurrence d'événements  $e$ ,

$$e = B(x_{d_{n-1}}, x_{dm}, x_{cm}) \quad (4.3)$$

$x_{d_{n-1}}$ , étant l'état discret précédent.

A partir des événements, l'état discret suivant est obtenu,

$$x_{d_n} = F_d(x_{d_{n-1}}, e) \quad (4.4)$$

et les commandes discrètes sont déduites et appliquées au cycle suivant :

$$u_d = G_d(x_{d_n}) \quad (4.5)$$

Une requête du superviseur (message  $U_d$ ) peut ainsi donner lieu à l'application de plusieurs commandes discrètes: la requête est affinée. Considérons l'exemple de deux réservoirs  $ST_1$  et  $ST_2$  connectés en série (Figure 4.8). La requête de transfert de matière de  $ST_1$  vers  $ST_2$  émise par le superviseur est décomposée par l'automate en deux commandes discrètes  $u_{d_1}$  et  $u_{d_2}$  correspondant respectivement à l'ouverture de la vanne et l'activation de la pompe.

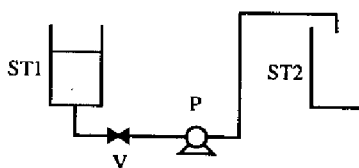


FIG. 4.8 – Exemple de deux réservoirs connectés en série

Cette supervision en boucle ouverte n'est possible que si le modèle utilisé pour générer les recettes de contrôle est uniquement discret (pas de transition impliquant des seuils). En effet, les dates de changement de phase (date d'émission des messages  $U_c$  et  $U_d$ ) doivent être prédéfinies par le plan  $\Sigma_L$  et ne peuvent pas dépendre de l'état courant du procédé. Il

4. supposition caractéristique des approches de type «superviseur» [Quénech'hdu *et al.* 95]

n'y a effectivement aucun retour d'état du procédé vers le superviseur. Il est donc possible de mettre en œuvre une opération de type "mélanger et chauffer pendant 10 minutes" mais pas une opération du type "remplir le réacteur avec 100kg de matière première A". Cela rend difficile l'utilisation de ce type de supervision pour des procédés de fabrication discontinus pour lesquels les changements de phases (déroulement de la recette) sont souvent dépendants de l'état du procédé [Andreu *et al.* 96a].

### 3.2 Supervision en boucle fermée

Nous venons de voir que la supervision d'un procédé discontinu nécessite l'accès à des informations sur l'état continu du procédé afin d'exécuter la recette, c'est-à-dire de décider des instants de changement de phase. Il s'agit de supervision en boucle fermée.

Le premier point important est qu'il est exclu que la fonction supervision soit directement connectée sur le bus de terrain à «caractère continu». En effet, elle utilise des modèles agrégés, et doit assurer l'interface avec les opérateurs humains et la gestion de la fabrication. Il ne serait pas sain de lui imposer les mêmes contraintes de temps réel strict que celles de la commande locale comme nous l'avons précisé. Le superviseur étant connecté au bus à «caractère discret» (Figure 4.5), il n'accède pas directement aux variables d'état continues, et par conséquent la supervision ne peut pas déterminer l'évolution de son état discret, c'est-à-dire déterminer l'occurrence d'événements, selon le schéma de l'équation 4.3.

En fait il faut introduire une nouvelle fonction, chargée de faire l'interface entre le monde des régulateurs continus et des automates programmables, autour du procédé, et celui de la supervision : le **générateur d'événements (GE)** [Andreu *et al.* 94b]. Si l'on se réfère à nouveau à l'architecture informatique proposée (Figure 4.5), nous pouvons constater que le premier monde fonctionne dans un environnement à échantillonnage rapide et périodique, celui du bus de terrain. Le second fonctionne dans un environnement asynchrone de messages sporadiques, moins rapide, celui du bus d'atelier. L'élément qui supporte la fonction de générateur d'événements se trouve nécessairement localisé entre les deux bus, et va manipuler d'un côté des événements (messages sporadiques), et de l'autre des variables continues (variables d'état échantillonnées). De ce fait, le générateur d'événements étant une fonction, il peut soit être localisé sur un processeur spécialisé (Figure 4.9), soit être réparti sur les régulateurs et les automates (auquel cas il n'apparaît plus dans l'architecture puisqu'il est distribué sur les organes de commande).

Toutefois il y a une grande différence entre sa fonction et celle des régulateurs et des automates. Intéressons nous à l'utilisation qui est faite des mesures permettant d'observer le fonctionnement du procédé (les variables d'état continues échantillonnées). Cette utilisation est naturellement différente selon le niveau d'abstraction considéré. Localement, ces mesures servent directement ou indirectement de valeurs d'entrée des organes de commande destinés à réguler des variables (asservissement d'ouverture de vanne, régulation de la puissance de chauffe, etc.), soit en maintien de consigne, soit en poursuite de trajectoire.



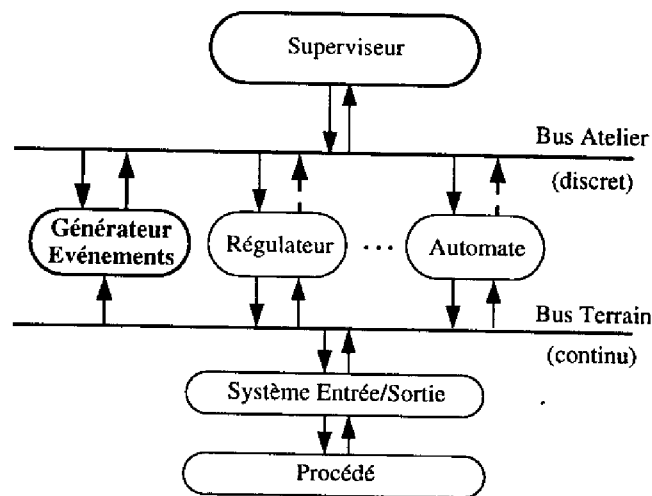


FIG. 4.9 – Position du générateur d'événements dans l'architecture informatique

Pour la supervision, ces mesures servent uniquement à détecter les **événements d'état**<sup>5</sup> provoquant un changement de phase ou de configuration tels que les franchissements de seuil par exemple.

L'ensemble des variables à surveiller pour détecter ces événements correspond à celles impliquées dans les opérations lancées par le superviseur, cet ensemble évolue selon les opérations en cours, c'est-à-dire selon la configuration. Par conséquent, le générateur d'événements ne doit s'attacher qu'aux **variables significatives** pour la supervision **dans la configuration courante**. Par exemple, si dans une batterie de réacteurs, un seul réacteur  $R_i$  est en phase de remplissage alors que les autres sont inutilisés (aucune activité en cours), seule la variable  $V_{R_i}$  du volume de produit contenu dans le réacteur  $R_i$  sera significative pour la supervision.

Cette notion de **configuration dynamique** implique qu'à chaque changement de configuration (c'est-à-dire à chaque changement d'état discret  $X_d$  pour le modèle utilisé par la supervision), le générateur d'événements doit être reprogrammé. En se basant sur le réseau de Petri décrivant la recette, cette reprogrammation est systématique. En effet, à partir de l'état discret courant  $X_d$ , c'est-à-dire du marquage du réseau de Petri qui indique la configuration active, nous pouvons déduire l'ensemble des transitions de seuil sensibilisées. C'est précisément l'ensemble des événements  $\Sigma_E$  à observer par le générateur d'événements ( $\Lambda_P$ ). Sur le schéma de la figure 4.10, l'arc étiqueté  $\Sigma_E$  correspond à l'ensemble des événements à observer, et  $\Sigma_P$  a pour signification l'ensemble des événements détectés.

5. en opposition aux événements de temps qui définissent la date de fin des activités dont la durée est spécifiée a priori.

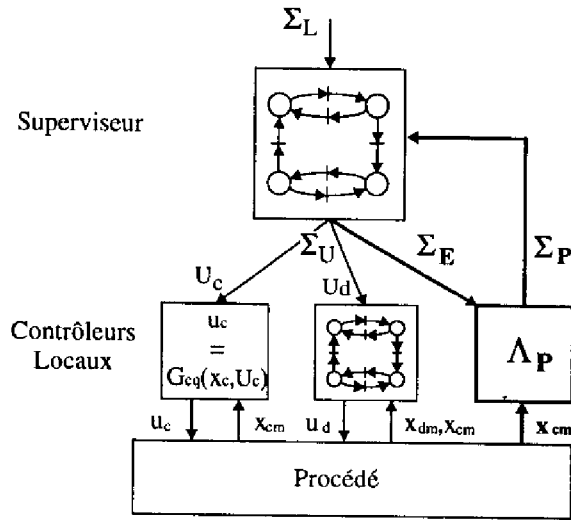


FIG. 4.10 – Supervision en boucle fermée

Revenons sur l'exemple de recette élémentaire (Figure 4.11). La phase active de la recette est la phase de chargement de 100kg de produit A, place  $p_1$  marquée. Dans cette configuration la transition sensibilisée est la transition  $t_1$ , donc la variable à considérer est celle décrivant la rétention massique du réacteur. Le générateur d'événements est programmé lors du lancement de la phase (marquage de  $p_1$ ), et le changement de phase, impliquant un changement de configuration, s'effectuera dès la réception de l'événement d'état associé à la transition  $t_1$ . Lors du franchissement de la transition  $t_1$  la phase de chargement en produit A est désactivée (démarquage de  $p_1$ ), la phase de chargement en produit B est activée (marquage de  $p_2$ ), et le générateur d'événements est reprogrammé. L'événement à détecter porte à nouveau sur la variable de rétention massique du réacteur, mais le seuil est différent.

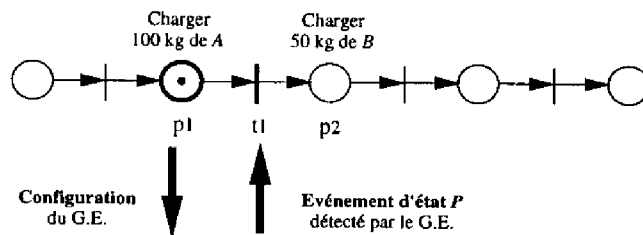


FIG. 4.11 – Explicitation du retour d'état

Jusqu'ici, notre approche repose sur un superviseur discret qui pilote un procédé hybride, typique des approches de type «superviseur» évoquées au chapitre précédent. Toutefois, il y a une différence notable avec celles-ci en ce qui concerne l'interface avec le procédé.

En effet, l'introduction d'un générateur d'événements nous permet de *définir dynamiquement* les événements, ou symboles (pour reprendre le terme utilisé dans ces approches), à considérer au niveau de l'interface. Nous allons maintenant évoquer le caractère hybride du modèle utilisé pour la supervision.

### 3.3 Nécessité d'un modèle hybride pour la supervision

La supervision en boucle fermée telle que nous venons de la décrire est restreinte au rôle d'enchaînement des phases constituant la recette. Ce rôle est bien évidemment restrictif car, comme nous l'avons précisé, la supervision doit assurer le bon déroulement de la production, notamment en surveillant le fonctionnement du procédé, et en l'optimisant. Ces objectifs nous amènent à considérer les différents aspects continus au sein du superviseur. La composante continue est de nature différente selon qu'il s'agit de commande et d'optimisation, ou de surveillance. Dans le cadre de la surveillance, le modèle continu correspond à un modèle de comportement (flux, phénomènes physico-chimiques, etc.). Dans un objectif de commande et d'optimisation, le modèle continu correspond généralement à l'équation d'un critère à optimiser, ou à une loi de commande (pour le calcul de consignes par exemple).

Tous ces modèles continus servent à piloter l'évolution du modèle à événements discrets représentant la stratégie de commande.

#### 3.3.1 Aspect continu et commande

L'aspect continu au sein de la commande ne se limite pas aux régulation continues de la commande locale. Une loi de commande dont l'utilisation se justifie dans un cadre d'optimisation d'une opération unitaire [Liu & Macchietto 95], en est une illustration possible.

Dans le cadre de l'optimisation, la supervision se base sur un ensemble de concepts importants. Utilisés souvent comme critères d'optimisation ou de choix<sup>6</sup>, les concepts sont exprimés par des variables continues autres que les grandeurs mesurées, traduisant des caractéristiques plus générales comme la qualité du produit, le rendement de l'unité, etc., qui ne servent pas à décrire le fonctionnement physique du procédé mais forment les objectifs de la supervision [Féray-Beaumont & Gentil 91]. Une grandeur continue caractéristique telle que le rendement d'une unité, par exemple, peut constituer un critère dans la décision d'affectation des ressources aux opérations (optimisation de la gestion de l'installation).

Dans ce cadre, les équations du modèle continu (à temps continu ou à temps discret) font parfois intervenir des images de variables d'état mesurées dont les valeurs doivent être

---

6. critères de choix pour lever les indéterminismes issus des degrés de liberté de la commande liés à la flexibilité du site, comme par exemple la polyvalence des appareils, ou le partage de ressources.

mises à jour régulièrement. Ces valeurs doivent être actualisées selon une fréquence bien moins élevée que l'échantillonnage de bas niveau, et cette actualisation n'est nécessaire que lors de l'utilisation du modèle continu (résolution d'une indétermination par exemple). L'actualisation est effectuée par le générateur d'événements. Nous n'avons considéré jusqu'ici qu'une seule **règle de génération** d'événements : "générer un événement *sur franchissement de seuil* par une variable continue". Le cas présent nous amène à définir deux autres règles de génération d'événement : "générer un événement *selon le taux de variation* de la variable continue" (par exemple une variation relative de 10 % de la valeur de la variable), et "générer un événement *périodiquement*" (par exemple toutes les minutes).

### 3.3.2 Aspect continu et surveillance

La surveillance repose sur la détection des déviations par comparaison entre le comportement attendu et le comportement effectif. Cette comparaison repose donc sur la confrontation des observations issues du générateur d'événements, avec les résultats de la simulation.

La simulation est fondée sur un modèle du procédé qui, à son image, est hybride. Nous devons préciser dès à présent que la résolution des équations du modèle est supposée ne pas poser de problème particulier, et nous renvoyons le lecteur aux travaux de Daubas sur les simulations continues et discontinues [Daubas 94]. Si nous considérons sans problème cette résolution, par contre la comparaison entre l'état du procédé de fabrication et celle du modèle simulé n'est pas simple car il faut comparer à la fois les variables d'état continues et les variables d'état discrètes.

En ce qui concerne une variable d'état continue une anomalie est détectée lorsque l'écart entre la valeur mesurée et la valeur simulée dépasse un certain seuil compte tenu de la précision et de la dynamique de la variable, et de la précision du modèle. Dans le cas d'une variable d'état discrète une anomalie sera détectée lorsque la date d'un événement d'état observé est à l'extérieur de la fenêtre temporelle estimée pour l'occurrence de cet événement : l'événement observé se produit trop tôt ou trop tard. Cette fenêtre temporelle est obtenue à partir de la simulation du modèle continu du procédé. Le modèle à événements discrets utilisé est nécessairement un réseau de Petri temporel.

Conformément à l'approche choisie au chapitre précédent, nous abordons ce caractère hybride par une approche du type «moniteur d'équations». Rappelons que cette approche consiste à associer des équations aux états discrets du modèle événementiel. Le modèle continu considéré ici est le modèle comportemental du procédé. En conséquence, ce sont les équations traduisant le comportement continu du procédé qui sont associées aux différents états du modèle événementiel, le réseau de Petri représentant la recette en l'occurrence [Valette *et al.* 95]. Les équations associées à une place d'activité du réseau expriment le phénomène continu ayant lieu sur le procédé pendant cette activité. Ces équations permettent donc de calculer les fenêtres temporelles d'occurrence des

événements correspondant à des franchissements de seuils.

Pour chaque transition de seuil sensibilisée par l'état discret courant (marquage du réseau), le système d'équations algébro-différentielles est utilisé pour évaluer la date au plus tôt et la date au plus tard de franchissement de seuil. Chaque fois qu'un événement d'état est détecté par le générateur d'événements (événement  $P \in \Sigma_P$  sur la figure 4.10), il est transmis à la supervision qui vérifie si sa date d'occurrence est compatible avec la fenêtre temporelle. Si c'est le cas, alors l'état discret suivant est calculé :

$$X_{d_n} = F_d(X_{d_{n-1}}, P) \quad (4.6)$$

Le nouveau système d'équations algébro-différentielles est généré par l'évolution du marquage, et le générateur d'événements est reprogrammé.

Si l'occurrence de l'événement se produit en dehors de la fenêtre temporelle, la transition correspondante n'est pas franchissable, et l'état discret suivant ne peut pas être atteint. La défaillance qui a été détectée est propagée vers un module de diagnostic.

La surveillance fine des variables continues est faite au niveau des régulateurs si l'on désire comparer la valeur réelle à la valeur nominale à chaque échantillonnage. Le modèle de supervision étant normalement un modèle agrégé, la surveillance et la mise à jour de ses composantes continues ( $X_c$ ) se fait simplement lors de l'occurrence de chaque événement. Par le fait qu'un seuil ait été franchi, la valeur de la variable continue correspondante est connue à cet instant.

### 3.3.3 Détermination de la fenêtre temporelle

Le modèle générique comportemental d'une opération physico-chimique est établi à partir des équations de conservation de la matière, de la quantité de mouvement et de l'énergie [Daubas 94]. Les liens entre une unité et son environnement, lors de l'opération, passent par des bilans de conservation écrits généralement, pour une phase<sup>7</sup>, sous une forme :

$$\begin{array}{ccccccc} \text{vitesse} & & \text{débit} & & \text{débit} & & \text{vitesse} \\ \text{d'accumulation} & - & \text{d'entrée} & + & \text{de sortie} & - & \text{de production} & = 0 \end{array}$$

Ce modèle générique est évolutif, ses termes dépendent de la configuration de l'unité. Le premier terme correspond au comportement dynamique pendant l'opération, le dernier correspond à une composante statique précisant une transformation physico-chimique, et les deux autres termes expriment les flux avec l'environnement selon les connexions avec d'autres unités.

---

<sup>7</sup>. une phase au sens multiphasique de la matière

Revenons à notre recette élémentaire (Figure 4.12), et intéressons-nous à la phase de remplissage du réacteur  $R_i$  dont la mesure de sa rétention massique est  $M_{R_i}$ . Le modèle générique du réacteur dans cette configuration se limite à :

$$\text{vitesse d'accumulation} - \text{débit d'entrée} = 0$$

Les paramètres de l'activité de remplissage sont connus, ils sont donnés par la recette : "charger 100kg de produit A avec un débit  $q_A = 100\text{kg/h}$ ". La durée estimée de l'activité peut être calculée<sup>8</sup> à partir de l'équation qui est associée à la place correspondante  $p_1$  :  $\Delta M_{R_i} = \int q_A \times dt$ .

Précisons que le seuil définissant l'événement ne peut pas être une valeur précise, notamment de par l'imprécision de la mesure. Le générateur d'événement doit détecter le franchissement d'un seuil défini par un intervalle de valeurs.

De manière identique, la date estimée de réception de l'événement ne saurait être une date précise (c'est-à-dire un instant de temps précis), d'une part à cause de l'imprécision du modèle et d'autre part à cause de la fluctuation du débit que nous supposons constant sur la période d'intégration (le transitoire est à prendre en compte par exemple). Par conséquent, la date estimée de réception de l'événement est traduite par une fenêtre temporelle correspondant à un intervalle de temps délimité par deux bornes. Ces bornes de l'intervalle correspondent aux dates de fin au plus tôt ( $d_1$ ) et de fin au plus tard ( $d_2$ ). La fenêtre temporelle formée par ces deux dates  $[d_1, d_2]$ , représente l'intervalle de temps pendant lequel l'événement d'état  $P$  "réacteur rempli" ( $\Delta M_{R_i} = 100\text{kg} \pm \text{tolérance}$ ) doit être reçu. Il s'agit de l'intervalle de franchissement de la transition de fin de l'activité  $t_1$ .

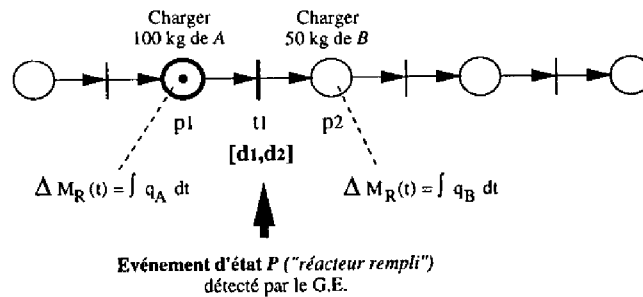


FIG. 4.12 - Surveillance du déroulement de la recette

La génération du nouveau système d'équations et la mise à jour de la composante continue, s'effectuent lors du franchissement de la transition  $t_1$ . Le modèle à événements discrets modifie la structure du modèle continu. Le système d'équations associé à la place dont le jeton est enlevé (place  $p_1$ ) est effacé, et celui associé à la place qui reçoit le jeton (place  $p_2$ ) est ajouté au solveur.

8. résolution de  $\Delta M_{R_i} - 100 = 0$

## 4 Bilan de l'approche

La démarche que nous avons suivie est fondée sur l'utilisation d'un réseau de Petri pour représenter la partie discrète, et d'un système d'équations algébro-différentielles pour représenter la partie continue. Le réseau de Petri décrit les enchaînements des opérations et configurations, nécessaires à la réalisation de la recette. Il **"pilote"** le système d'équations puisque, lors de chaque changement de configuration, la structure du système, et donc le système d'équations, est modifiée. De façon symétrique le système d'équations **"contrôle"** l'évolution du réseau de Petri puisque c'est lui qui permet de savoir à quelles dates devraient être franchies les transitions correspondant à des franchissements de seuils par des variables d'état continues.

Cependant, comme nous l'avons fait remarquer en étudiant l'aspect modélisation, cette représentation «fusionne» l'état de la commande, la configuration du procédé, et l'ensemble des équations traduisant les phénomènes continus en cours, dans une même vue (même état discret). Nous allons raffiner cette vision, et structurer autrement le modèle afin de séparer les trois vues précédentes.

La dissociation de l'aspect commande et de l'aspect procédé, de manière identique à la démarche suivie dans les travaux sur la surveillance des systèmes à événements discrets (cf. chapitre 2), nous permet de définir deux vues : une première vue discrète correspondant à la stratégie de commande, et une seconde établissant une vue «qualitative» de l'état du procédé à travers la représentation de ses configurations. A chacune de ces configurations nous associons alors une vue purement continue par le biais d'équations algébro-différentielles [Andreu *et al.* 95].

Un des points essentiels devient par conséquent d'assurer une évolution cohérente de l'ensemble de la structure, afin de maintenir une description cohérente de l'état du procédé.

## Chapitre 5

# Un modèle hybride pour la supervision

Le modèle hybride utilisé pour la supervision est le résultat d'une structuration du modèle selon les trois vues précédemment évoquées, en se basant sur la séparation de l'aspect commande et de l'aspect procédé: une vue discrète correspondant à la stratégie de commande, une vue discrète représentant les configurations du procédé, et une vue continue dans chaque configuration, par l'association d'équations algèbro-différentielles.

Nous allons nous intéresser tout particulièrement à la modélisation hybride du procédé et à son utilisation dans le cadre de la commande et de la surveillance. Nous définirons alors un **modèle de référence hybride**. Ce modèle ne correspond qu'à un premier niveau de la représentation discrète (des configurations) utile pour composer dynamiquement le système d'équations. Si la représentation des évolutions internes du procédé est nécessaire pour l'obtention d'événements de temps traduisant les instants de changement d'état du procédé (changements d'état hydro-dynamiques ou thermodynamiques non observables par exemple), des modèles discrets moins agrégés doivent être décrits afin d'accéder à ce niveau de raffinement.

Il est essentiel d'assurer une évolution cohérente de l'ensemble de la structure constituant le modèle utilisé pour la supervision, nous présenterons par conséquent les règles d'évolution de cette structure.

Nous reviendrons ensuite sur la «construction» du système de commande/supervision afin de situer le modèle de référence hybride au sein du système.

Enfin, nous résumerons les interactions entre les aspects continus et discrets du système de commande/supervision proposé.

### 1 Un modèle hybride du procédé

Comme nous l'avons précisé à plusieurs reprises le procédé discontinu doit nécessairement être caractérisé par son état discret et son état continu. Le modèle du procédé doit



par conséquent être de nature hybride. La problématique de la surveillance des procédés discontinus que nous avons abordée au chapitre 2, a mis l'accent sur l'importance de la représentation des configurations du procédé. La connaissance de la configuration dans laquelle se situe chaque unité du procédé est nécessaire pour surveiller son comportement en terme de tendance. Cette représentation «qualitative» permet d'indiquer l'évolution attendue d'une grandeur caractérisant l'unité: par exemple lors du transfert d'un lot vers une unité de stockage, la variation de la rétention massique de l'unité doit être croissante. Le modèle de l'unité doit également permettre de décrire la dynamique de son comportement continu dans chacune de ses configurations.

Le modèle hybride du procédé est constitué d'un ensemble de modèles hybrides, *un modèle par unité*. Chaque unité est représentée par un modèle hybride basé sur un réseau de Petri dont les places correspondent aux configurations de l'unité par rapport à son environnement. A chaque état discret du réseau (chaque place) est associé un système d'équations traduisant le phénomène continu correspondant. La configuration de l'unité est caractérisée par la position du jeton dans le réseau (le marquage) et son état continu est «porté» par le jeton. Ce modèle hybride offre par conséquent une *vue «qualitative»* du comportement de l'unité ainsi qu'une représentation de son comportement continu.

Notre objectif d'utilisation du modèle est multiple :

- d'une part, dans le cadre de la surveillance, le modèle doit permettre la détection de comportements anormaux du procédé par rapport au comportement attendu selon la commande appliquée. Il est donc utilisé pour valider l'exécution d'une commande.
- d'autre part le modèle doit permettre de s'assurer qu'une commande à appliquer est cohérente avec l'état du procédé. Il est donc consulté par la commande avant l'émission de toute requête.
- enfin, nous avons parfois besoin de pouvoir émuler le modèle<sup>1</sup> pour vérifier que la commande à appliquer n'entraînera pas de violation de contraintes. En effet, dans certains cas que nous illustrerons ultérieurement, la connaissance de l'état statique (discret et continu) du procédé est insuffisante pour garantir la cohérence d'une commande. Il est alors nécessaire de prendre en compte la dynamique continue, c'est l'objectif de cette émulation du modèle du procédé.

Dans le cadre d'une architecture hiérarchisée et modulaire, le modèle du procédé doit être structuré pareillement et se situer en parallèle de la commande. De ce fait notre approche va s'inspirer des travaux de Combacau sur la surveillance des systèmes à événements discrets basée sur une approche à modèle de référence [Combacau 91]. Nous appellerons par conséquent ce modèle du procédé hybride **le modèle de référence hybride**, exprimant ainsi sa nature et son utilisation.

---

1. projection de son comportement sur un horizon temporel limité.

## 2 Un exemple de modèle de référence hybride : une unité de stockage

Afin d'illustrer le concept de modèle de référence hybride, considérons l'exemple simple d'une unité de stockage, constituée en fait d'un réservoir  $R$  pour lequel la variable d'intérêt est sa rétention massique  $M_R$  (Figure 5.1).

Si nous reprenons le modèle comportemental précédemment décrit, en considérant le réservoir comme isotherme et parfaitement agité, le bilan de conservation de la matière s'écrit :

$$\text{vitesse d'accumulation} - \text{débit d'entrée} + \text{débit de sortie} = 0$$

Le modèle du réservoir est composé de 4 états discrets, les 4 places du réseau de Petri (Figure 5.1). Ces états correspondent aux configurations suivantes : "réservoir inactif" (place  $p_1$ ), "réservoir en remplissage" (place  $p_2$ ), "réservoir en soutirage" (place  $p_3$ ), et "réservoir simultanément en remplissage et soutirage" (place  $p_4$ ).

Ces 4 états discrets sont les configurations accessibles du réservoir. A chacun d'entre eux est associé le modèle comportemental selon une structure dépendant de l'état. Dans l'état "réservoir en remplissage" le modèle se limite à :

$$\text{vitesse d'accumulation} - \text{débit d'entrée} = 0$$

C'est ce qui est exprimé par l'équation associée à la place  $p_2$ .

La configuration "réservoir simultanément en remplissage et soutirage" est fréquente pour un «stock intermédiaire». Si des contraintes technologiques n'autorisent pas l'utilisation de cet état alors il doit être supprimé du modèle de référence (Figure 5.2.1). Par contre, si cette contrainte est issue de la stratégie de commande, alors elle doit être spécifiée sur le modèle de la commande : la ressource est affectée exclusivement à une activité de remplissage ou à une activité de soutirage. Tous les états décrits sur le modèle de référence doivent être les configurations accessibles, et cet ensemble fini d'états discrets ne doit pas être restreint par des considérations de pilotage. Ce modèle doit par définition être indépendant de la stratégie d'utilisation du procédé.

Le degré de finesse du modèle de référence dépend du degré de «détail» requis. Il ne s'agit pas de précision numérique sur les variables mais d'abstraction. Reprenons l'exemple du réservoir précédent. Il est caractérisé par sa configuration, les états discrets du modèle de référence, et son état continu, la valeur de sa variable continue de rétention massique. Affiner la vision continue correspond à faire apparaître la composition molaire du produit contenu dans le réservoir (équations de bilan matière par constituant  $c_i$ ), et affiner la vision discrète correspond à séparer l'état "réservoir inactif" en deux états distincts "réservoir

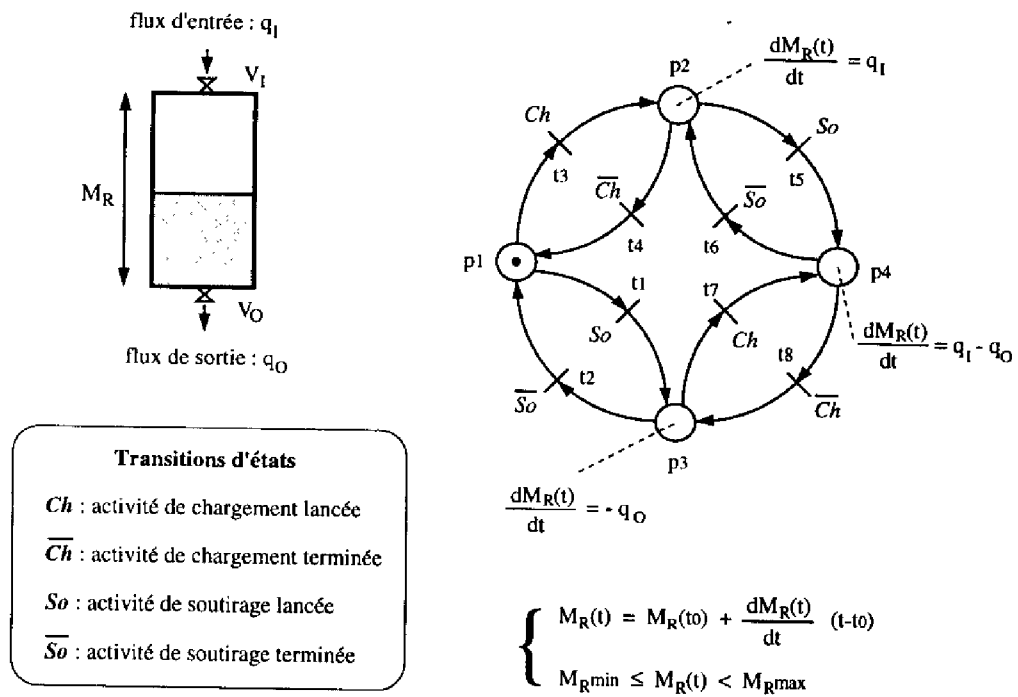
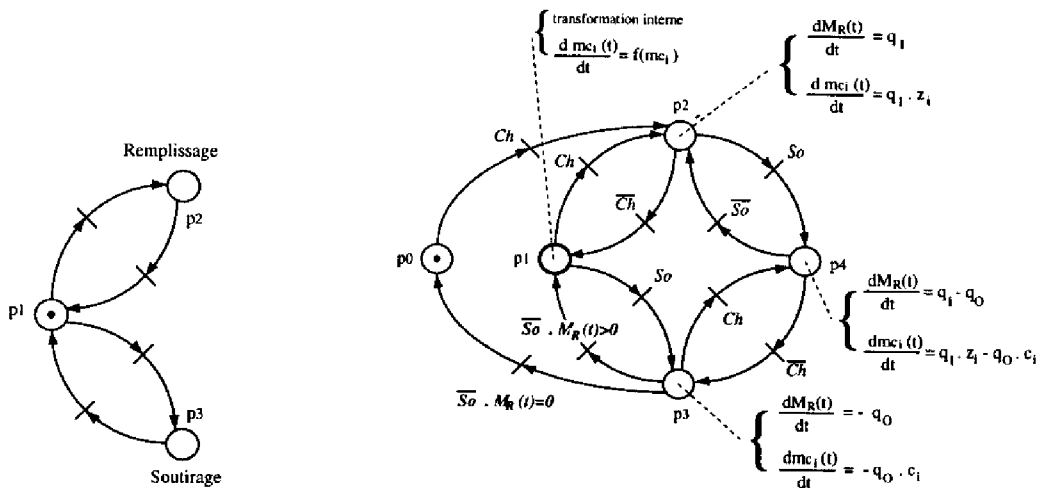


FIG. 5.1 – Le modèle de référence d'une unité de stockage

inactif vide” et “réservoir inactif non vide”. Cette distinction n’apporte rien sur la configuration du réservoir (liens avec son environnement), il est isolé dans les deux cas, mais elle permet de représenter un état au sein duquel il y a évolution du produit: il s’agit du terme que nous avons ignoré en supposant le réservoir isotherme et parfaitement agité. Ce terme traduit la réaction interne entre les composantes du produit (en supposant cette réaction modélisée). Ce modèle de référence «détaillé» est présenté sur la figure 5.2.2.



5.2.1: Modèle de référence contraint

5.2.2: Modèle de référence «détaillé»

FIG. 5.2 - Exemples de modèles de référence

Dans tous les cas la structure de l'équation traduisant le comportement continu diffère selon l'état, seul l'état "réservoir inactif (vide)" n'a pas d'équation associée, aucun phénomène continu n'ayant lieu dans cette configuration.

Une précision est nécessaire en ce qui concerne l'indépendance du modèle de référence par rapport à la commande. Prenons le cas d'un réservoir dont l'état continu est caractérisé par le volume de matière qu'il contient. Si à travers le modèle de référence, nous ne surveillons que le comportement normal du procédé, surveiller l'évolution normale de la composante continue se limite à surveiller que le volume maximum  $V_{max}$  du réservoir ne soit pas atteint ou dépassé. Toute variation de la variable continue entre 0 (réservoir vide) et  $V_{max}$  peut être considérée comme normale. Dans notre approche cette formulation est insuffisante puisque nous désirons surveiller que le comportement du procédé soit conforme au comportement attendu selon la commande appliquée. C'est ainsi que nous allons associer des fenêtres temporelles de franchissement aux transitions du modèle de référence dont le calcul est effectué en fonction des paramètres de la commande (volume à transférer, débit, etc.). Les fenêtres temporelles étant déterminées selon la commande appliquée, il n'y

a plus rigoureusement indépendance. En fait le modèle de référence est indépendant de la commande mais ce qui est surveillé sur le modèle ne l'est pas.

### 3 Les règles d'évolution

Le modèle hybride utilisé pour la supervision doit évidemment être cohérent avec l'état du procédé (configurations, opérations en cours, etc.). Il est essentiel d'assurer une évolution cohérente de l'ensemble de la structure constituant ce modèle hybride. Nous devons donc assurer un changement de configuration **automatique et cohérent** sur le modèle de référence, lors de l'évolution de la commande.

Les équations étant associées aux configurations des unités il en résulte une détermination et sélection, automatique et cohérente des équations actives<sup>2</sup>.

Le changement de configuration résulte des décisions de lancement et d'arrêt des opérations ainsi que de l'affectation des unités aux opérations : c'est la stratégie de pilotage exprimée sur le modèle de la commande. L'évolution du modèle de référence doit être par conséquent *couplée* à l'évolution du modèle de la commande et leurs évolutions doivent être *simultanées*.

La simultanée de l'évolution de ces deux modèles discrets repose sur le principe de la fusion des transitions<sup>3</sup> [Alanche *et al.* 84]. Nous étudierons plus en détail la communication entre les deux modèles au chapitre suivant, considérons pour l'instant cette évolution comme simultanée.

Les transitions fusionnées sont désignées par un label exprimant la nature du changement d'état (cf. légende figure 5.1). A titre d'exemple considérons la figure 5.3.1 : le label (*Ch*) associé à la transition  $t_{10}$  du modèle de la recette indique que la transition devant être fusionnée sur le modèle de référence est une transition sensibilisée portant le même label. Dans ce cas il s'agit de la transition  $t_3$ .

Nous allons décrire plus en détail l'évolution des modèles en considérant tout d'abord l'exécution d'une recette, puis de deux recettes en parallèle.

#### 3.1 Exécution d'une seule recette

Reprenons l'exemple de recette représentée sur la figure 4.12, et intéressons-nous à l'opération de chargement du réacteur en produit A. L'exécution de cette opération

2. la sélection est dite cohérente si l'ensemble des équations sélectionnées correspond aux équations traduisant le phénomène continu ayant lieu sur le procédé à cet instant.

3. La transition résultant d'une fusion de transitions est une transition dont l'ensemble des places d'entrée (respect. des places de sortie) est l'union des places d'entrée (respect. des places de sortie) de chacune des transitions fusionnées. L'interprétation associée à cette transition est le ET logique des conditions de chacune des transitions

peut être décomposée en trois étapes : le lancement de l'opération, le déroulement de l'opération, et la fin de l'opération. La figure 5.3 représente l'évolution du marquage des deux modèles discrets lors de chaque étape. Les extraits de réseaux de Petri représentés sur cette figure correspondent aux branches actives du réseau de la commande et du réseau de référence.

Avant le lancement de l'opération, la recette est dans l'état  $p_{10}$  et le modèle de référence du réacteur est dans l'état discret  $p_1$  (Figure 5.3.1). Les transitions sensibilisées sont la transition  $t_{10}$  sur le modèle de la commande, et les transitions  $t_1$  et  $t_3$  sur le modèle de référence. Le lancement de la phase de chargement du réacteur correspond au franchissement de la transition  $t_{10}$ . Ce franchissement n'est possible que si la requête de remplissage du réacteur est autorisée par son modèle de référence, c'est-à-dire si la transition  $t_3$  qui est fusionnée avec la transition  $t_{10}$  ( $Ch$ ) peut être simultanément franchie.

L'évolution des modèles est initiée par le réseau de la commande : le lancement d'une opération provoque un changement de configuration.

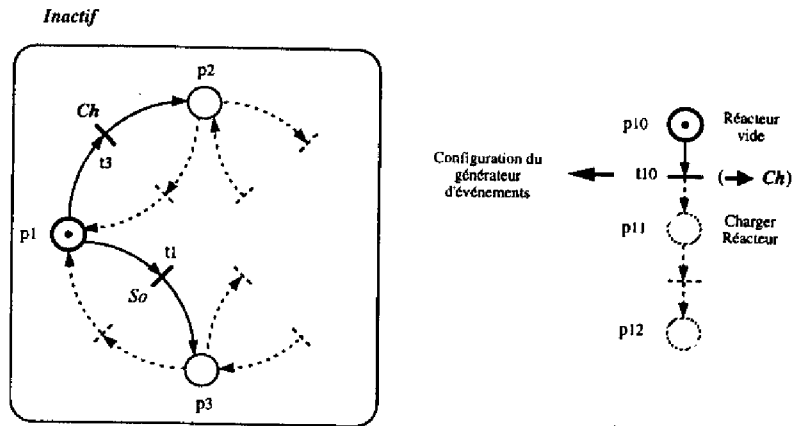
Le lancement effectué (tir des transitions  $t_{10}$  et  $t_3$ ), le générateur d'événements a été programmé par la recette (tir de  $t_{10}$ ), et le nouvel état atteint par les modèles discrets est traduit par le marquage des places  $p_{11}$  et  $p_2$  (Figure 5.3.2). L'ensemble des transitions sensibilisées devient  $t_{11}$ ,  $t_4$ , et  $t_5$ . La place  $p_2$  étant marquée, le modèle de référence calcule la fenêtre temporelle de réception de l'événement d'état associée à la transition  $t_4$  ( $[d_1, d_2]$ ), ainsi que la valeur attendue de la rétention massique du réacteur (valeur simulée).

Dès réception de l'événement "réacteur chargé", la transition  $t_{11}$  est franchie seulement si les deux conditions suivantes sont satisfaites :

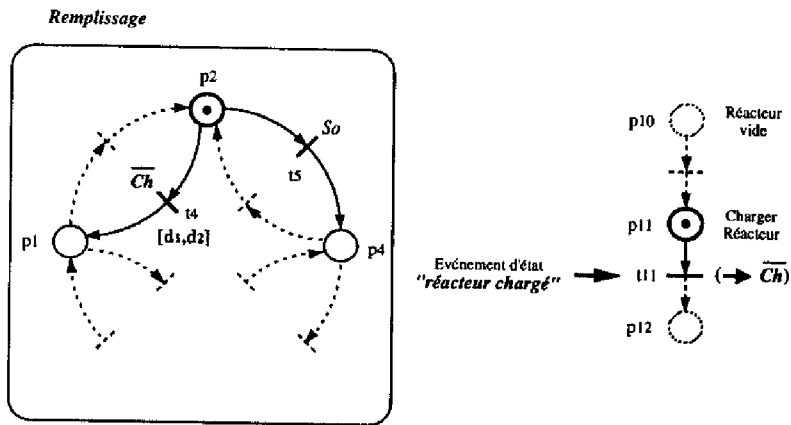
1. l'occurrence de l'événement a eu lieu dans l'intervalle de temps prévu : il s'agit de l'intervalle porté sur la transition  $t_4$ . En effet, le tir de la transition  $t_{11}$  n'est possible que si la transition  $t_4$  avec laquelle elle est fusionnée est franchissable, or la transition  $t_4$  n'est franchissable que dans l'intervalle de temps  $[d_1, d_2]$ .
2. l'écart entre la valeur mesurée de la rétention massique du réacteur, qui a été retournée par le générateur d'événements, et la valeur simulée est inférieur à un certain seuil. Dans ce cas, la composante continue du modèle de référence du réacteur est mise à jour : le modèle conserve ainsi une image fidèle de l'état effectif du procédé (état discret et état continu). L'état continu ainsi mis à jour constitue les valeurs initiales des prochains calculs.

L'évolution du modèle de la commande induit celle du modèle de référence : la fin de l'opération provoque un changement de configuration. L'état discret atteint par les deux modèles est celui décrit sur la figure 5.3.3.

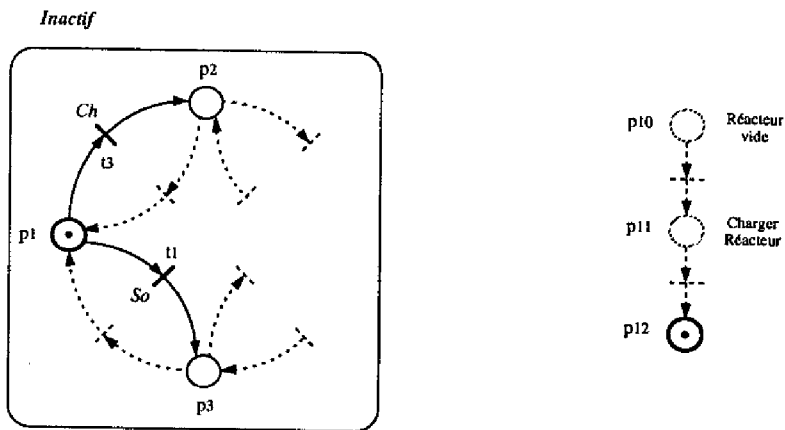
*Les modèles sont indépendants mais sont dynamiquement couplés lors de l'exécution de la recette.*



5.3.1: Avant l'opération



5.3.2: Pendant l'opération



5.3.3: Après l'opération

FIG. 5.3 – Evolution des modèles discrets

Si l'événement se produit en dehors de la fenêtre temporelle déterminée, ou si l'écart entre les valeurs est supérieur au seuil spécifié, la transition  $t_4$  n'est pas franchissable. Le modèle de référence ne peut donc pas évoluer, par conséquent le modèle de la commande est bloqué puisque la transition  $t_{11}$  n'est pas franchissable. Une défaillance a été détectée. Le mécanisme de comparaison permettant cette détection sera traité au prochain chapitre.

### 3.2 Exécution de plusieurs recettes en parallèle

Nous allons décrire l'évolution du modèle de référence d'une unité de stockage (un réservoir intermédiaire) utilisée simultanément par deux recettes qui s'exécutent en parallèle.

Afin d'illustrer la possibilité d'une modification dynamique<sup>4</sup> des fenêtres temporelles associées aux transitions du modèle de référence, selon les opérations en cours, nous considérerons que le débit de sortie de l'unité est une fonction du volume et non un paramètre fixé.

L'unité de stockage  $ST$  est située entre deux réacteurs  $R_1$  et  $R_2$  (Figure 5.4). Les deux recettes considérées sont représentées par les réseaux de Petri de la figure 5.5. La première recette consiste à produire une solution  $C$  qui sera stockée dans le réservoir intermédiaire  $ST$ . La seconde recette fabrique un produit  $D$  à partir de la solution  $C$  contenue dans  $ST$ , et stocke ce produit dans un réservoir aval. Ces recettes utilisent deux réacteurs distincts qui sont respectivement  $R_1$  et  $R_2$ .

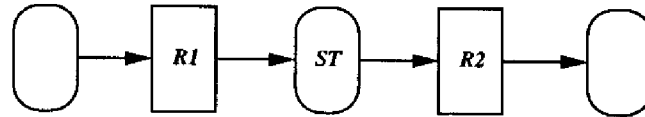


FIG. 5.4 - Diagramme de flux

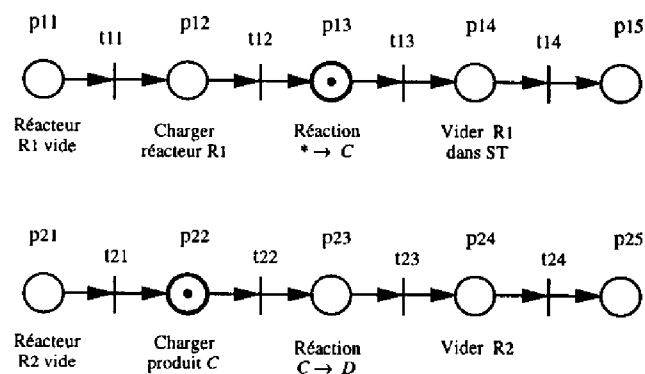


FIG. 5.5 - Les recettes

4. les bornes de la fenêtre temporelle sont recalculées.



A l'instant  $t$  considéré la première recette est supposée être en phase de transformation d'un lot dans le réacteur  $R_1$  (réacteur  $R_1$  isolé) et la seconde en phase de chargement du réacteur  $R_2$  (réacteur  $R_2$  connecté à  $ST$ ). Le modèle de référence du réservoir  $ST$  est par conséquent dans la configuration "réservoir en soutirage". Cet état est traduit sur leurs modèles respectifs par les places marquées  $p_{13}$ ,  $p_{22}$ , et  $p_3$  (Figure 5.6.1).

La seule opération qui implique le réservoir intermédiaire  $ST$ , à l'instant de temps considéré, est la phase de chargement du réacteur  $R_2$  en produit  $C$ . La durée de cette opération est calculée à partir du système d'équations algébro-différentielles suivant :

$$\begin{aligned} q_o &= f(V_{ST}) \\ \frac{dV_{ST}}{dt} &= -q_o \\ V_{ST}(t_0) &= V_{ST(t_0)} \end{aligned} \quad (5.1)$$

avec  $V_{ST}$  le volume du réservoir  $ST$ , et  $q_o$  son débit de sortie.

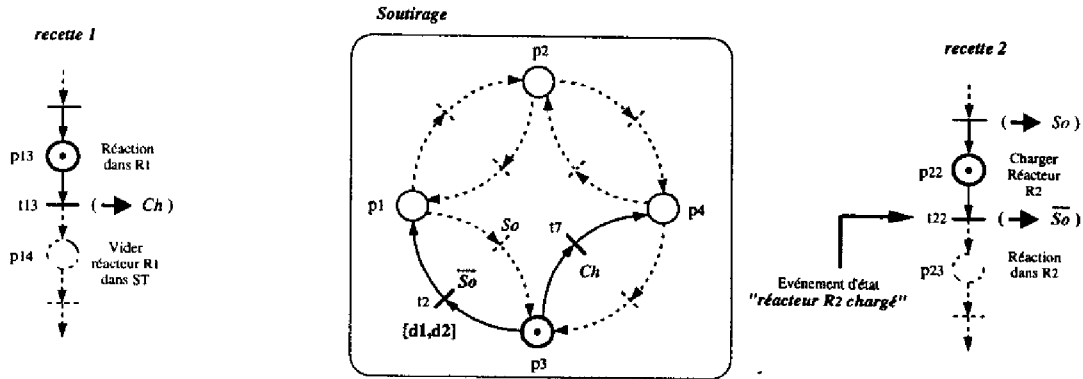
La fenêtre temporelle associée à la transition de fin de configuration, la transition  $t_2$  sur le modèle de référence, en est déduite : il s'agit de la fenêtre  $[d_1, d_2]$  (Figure 5.6.1).

Pendant le chargement du réacteur  $R_2$  en produit  $C$ , le traitement du lot dans le réacteur  $R_1$  sur la première recette est terminé. Cette recette passe donc à la phase de transfert du lot dans le réservoir  $ST$  (démarquage de  $p_{13}$  et marquage de  $p_{14}$ ) (Figure 5.6.2). Les deux réseaux de Petri de la commande, correspondant aux deux recettes, sont couplés avec un même modèle de référence, celui du réservoir intermédiaire  $ST$ .

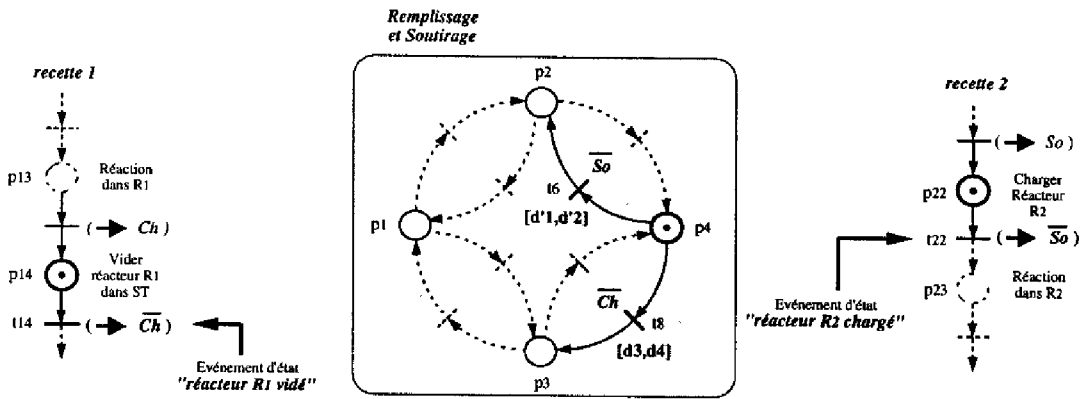
Cette évolution de la commande induit un changement de configuration sur le modèle de référence : le réservoir est maintenant dans la configuration "remplissage et soutirage simultané" (Figure 5.6.2). Lorsque l'état discret correspondant (place  $p_4$ ) est marqué, les fenêtres temporelles de fin des opérations en cours sont recalculées. L'intervalle de temps prévu pour la fin du chargement du réacteur  $R_2$  (seconde recette) devient  $[d'_1, d'_2]$  au lieu de  $[d_1, d_2]$ . En effet le remplissage du réservoir modifie l'évolution de son volume, ce qui implique un débit de sortie différent (équations 5.1). La modification du débit de sortie modifie alors les bornes de la fenêtre temporelle.

La fenêtre  $[d'_1, d'_2]$  est maintenant associée à la transition  $t_6$  alors que la fenêtre  $[d_1, d_2]$  était associée à la transition  $t_2$ . Ces deux transitions sont des transitions de sortie de deux états différents, mais leur franchissement correspond au même phénomène : la fin du transfert de matière de  $ST$  vers  $R_2$ , traduite par l'événement "réacteur  $R_2$  chargé" ( $\bar{S}_0$ ).

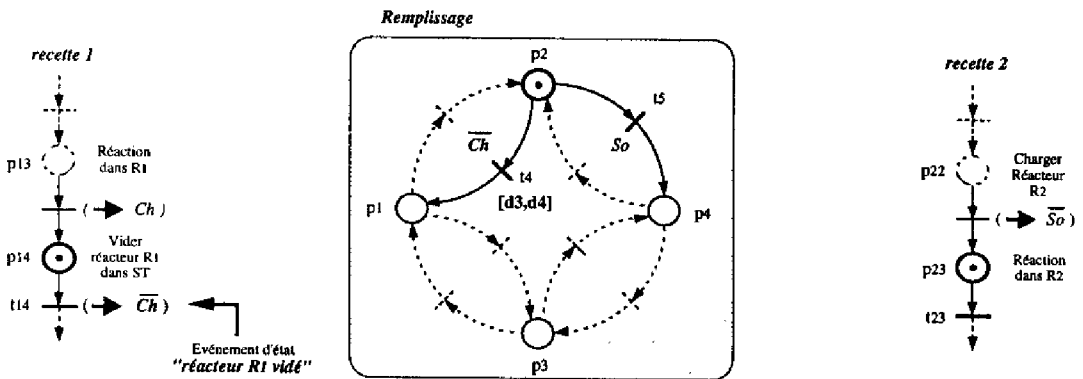
Dans la configuration courante (Figure 5.6.2), plusieurs changements d'état sont possibles sur le modèle de référence : soit l'événement "réacteur  $R_1$  vidé" se produit avant l'événement "réacteur  $R_2$  chargé", auquel cas le modèle de référence revient dans son état



5.6.1: Réservoir connecté au réacteur  $R_2$



5.6.2: Réservoir connecté aux réacteurs  $R_1$  et  $R_2$



5.6.3: Réservoir connecté au réacteur  $R_1$

FIG. 5.6 – Unité de stockage utilisée simultanément par deux recettes

décrit sur la figure 5.6.1, soit c'est l'inverse et le modèle de référence atteint l'état décrit sur la figure 5.6.3.

## 4 Une détection affinée

La séparation de l'aspect procédé et l'aspect commande permet d'affiner la détection d'une défaillance en ce sens que la *nature de la défaillance* détectée sur chacun des modèles est *différente*. Sur le modèle de référence, comme nous venons de le voir, ce sont les déviations de comportement du procédé qui sont surveillées. Sur le modèle de la commande nous allons prendre en compte des déviations propres à la commande.

Illustrons la différence entre les comparaisons portées sur le modèle de référence et celles portées sur le modèle de la commande, en considérant l'exemple du transfert entre un stock et un réacteur.

Pour cela revenons à la recette élémentaire que nous avons prise comme exemple (Figure 4.6). Cette recette comporte une phase de chargement d'un réacteur avec quantité de  $100kg$  d'un produit  $A$ . Cette phase impose une configuration dans laquelle le réacteur  $R$  est connecté à un réservoir  $S$  contenant le produit  $A$  à charger. L'exemple met donc en jeu deux modèles de référence, le modèle de référence du réacteur et celui du réservoir, et le modèle de la recette (partiellement représentés sur la figure 5.7).

Le modèle de la recette, qui coordonne les deux modèles de référence (changement de configuration simultané), fixe de façon cohérente les paramètres de chacune de ces configurations. La recette garantit en effet que les paramètres de l'activité de soutirage du réservoir et ceux de l'activité de remplissage du réacteur sont identiques : par exemple le débit d'entrée du réacteur est le débit de sortie du réservoir.

Les modèles de référence du réacteur et du réservoir sont respectivement dans les configurations "réacteur en remplissage" (place  $p_{22}$  sur la figure 5.7.2) et "réservoir en soutirage" (place  $p_{13}$  sur la figure 5.7.1). Le modèle de la commande, la recette, a pour phase active le chargement du réacteur en produit  $A$  (place  $p_2$  sur la figure 5.7.3).

Sur chacun des modèles de référence, une fenêtre temporelle est associée à la transition sensibilisée de fin d'activité. Nous avons déjà défini le sens de ces fenêtres, elles indiquent l'intervalle de temps prévu pour la fin de l'activité sur l'unité considérée. La fin du transfert entre le réservoir et le réacteur correspond à l'événement  $E$  traduisant la variation de la rétention massique du réacteur telle que  $\Delta M_R = +100kg$ . Précisons que dans ce cas le générateur d'événement retourne, lors de la détection de l'événement ci-dessus, à la fois la valeur de la rétention massique du réacteur et celle du réservoir.

Par un raisonnement similaire à celui mené par Combacau dans le cadre de la surveillance des systèmes à événements discrets [Combacau 91], il est possible d'établir une comparaison entre la date d'occurrence de cet événement de fin d'activité et celle

prévue par le plan de fabrication (date issue de l'ordonnancement). Il est ainsi possible de s'assurer du respect des dates d'exécution prévues par le plan de fabrication par exemple.

La phase de transfert a des dates d'exécution prévues par le plan de fabrication : ces dates sont les dates de début au plus tôt ( $Début_{tôt}$ ), et de fin au plus tard ( $Fin_{tard}$ ) de l'activité. De plus le plan de fabrication a associé à l'activité de transfert une durée estimée ( $D_{transfert}$ ).

Comme le souligne Combacau, l'intervalle  $[Début_{tôt}, Fin_{tard}]$  est généralement plus long que la durée estimée de l'activité  $D_{transfert}$  afin de laisser une marge temporelle à la commande. Les dates déterminant la fin prévue de l'activité en sont déduites, constituant ainsi la fenêtre temporelle de fin d'activité  $[Fin_{tôt}, Fin_{tard}]$ , avec  $Fin_{tôt} = Début_{tôt} + D_{transfert}$ .

Cet intervalle peut être réduit en calculant la date  $Fin_{tôt}$  à partir de la date effective de lancement de l'activité ( $Début_{effectif}$ ) : il s'agit de la date à laquelle la place correspondante devient marquée. Ce calcul de la date de fin au plus tôt de l'activité, permet de détecter immédiatement l'infaisabilité du plan dans le cas où  $Fin_{tôt} = Début_{effectif} + D_{transfert}$  est telle que  $Fin_{tôt} > Fin_{tard}$ . Cette situation peut être due à un retard pris dans l'exécution d'activités<sup>5</sup> exécutées en parallèle sur d'autres recettes impliquant la ressource (unité) nécessaire pour l'activité considérée. Cette situation ne peut pas provenir d'un retard sur une des activités précédentes se déroulant sur la même recette car le dépassement aurait été détecté auparavant.

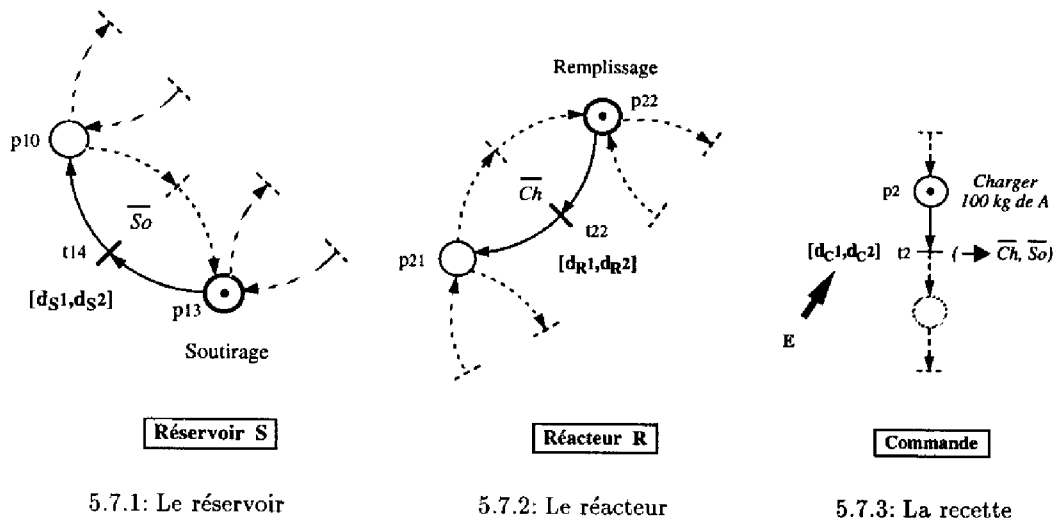


FIG. 5.7 – Interaction entre les modèles

5. par exemple, ce retard peut provenir du fait que l'état continu des équipements n'est pas tout à fait l'état nominal, ce qui «ralentit» les opérations (leur durée n'est plus tout à fait celle prévue).

Revenons à notre exemple. L'évolution des trois modèles discrets est nécessairement liée. Cette évolution vers l'état discret suivant sur chacun des modèles, dépend du franchissement simultané de leurs transitions sensibilisées respectives. Les transitions que nous allons examiner sont la transition  $t_{14}$  sur le modèle de référence du réservoir  $S$  (Figure 5.7.1), la transition  $t_{22}$  sur le modèle de référence du réacteur  $R$  (Figure 5.7.2), et la transition  $t_2$  sur la recette (Figure 5.7.3). Les fenêtres temporelles associées à ces transitions sont respectivement  $[d_{S_1}, d_{S_2}]$ ,  $[d_{R_1}, d_{R_2}]$ , et  $[d_{C_1}, d_{C_2}]$  (équivalente à  $[\text{Fin}_{\text{tôt}}, \text{Fin}_{\text{tard}}]$ ).

Si ces fenêtres temporelles ne se chevauchent pas, il est évident qu'au moins une de ces transitions ne sera pas franchissable, et par conséquent, leur franchissement devant être simultané (en mode nominal), une anomalie sera nécessairement détectée. Dans l'hypothèse d'un fonctionnement normal (les fenêtres déterminées correspondant au fonctionnement normal), la faisabilité de l'ordonnancement peut ainsi être examinée. Déterminer que le plan de fabrication n'est pas faisable évite de chercher l'origine d'une défaillance qui aurait conduit au non-respect des contraintes temporelles. En écartant la possibilité que les intervalles de temps des modèles de référence ne se chevauchent pas, ce qui traduirait une mauvaise modélisation du phénomène continu (non respect des bilans matière), trois scénarios restent possibles. Comparons pour cela l'intervalle  $[d_{R_1}, d_{R_2}]$  de fin de remplissage du réacteur porté sur le modèle de référence du réacteur, et l'intervalle  $[d_{C_1}, d_{C_2}]$  de fin de la phase porté sur le modèle de la recette (Figure 5.8) :

1.  $d_{R_2} < d_{C_1}$  : la date de fin au plus tard du remplissage du réacteur précède la date de fin au plus tôt de la phase (Figure 5.8.1). L'événement d'état  $E_2$  ("réacteur rempli"), se produit avant la date prévue par le plan de fabrication : l'activité a été lancée trop tôt.
2.  $d_{R_1} > d_{C_2}$  : la date de fin au plus tard de la phase précède la date de fin au plus tôt du remplissage du réacteur (Figure 5.8.2). L'événement d'état  $E_2$  se produit après la date prévue par le plan de fabrication : l'activité n'a pas été lancée trop tard puisque cela aurait été détecté lors du calcul des bornes de l'intervalle, c'est donc que le plan proposé est infaisable (durée de l'activité mal estimée par exemple).
3.  $d_{R_1} < d_{C_2}$  et  $d_{R_2} > d_{C_1}$  : la date de fin au plus tôt du remplissage du réacteur précède la date de fin au plus tard de la phase, et la date de fin au plus tard du remplissage du réacteur est précédée par la date de fin au plus tôt de la phase (Figure 5.8.3). Il y a donc chevauchement des intervalles de temps, le plan de fabrication est, normalement, faisable.

Ce mécanisme offre la possibilité de valider la faisabilité d'une activité dans l'intervalle temporel prévu par le plan de fabrication. De plus, cette «modularité» dans la fonction de détection permet une *localisation spatiale*<sup>6</sup> et une *localisation fonctionnelle*<sup>7</sup> de la

6. connaissance de l'unité impliquée dans la défaillance.

7. connaissance de la fonction en cours sur cette unité.

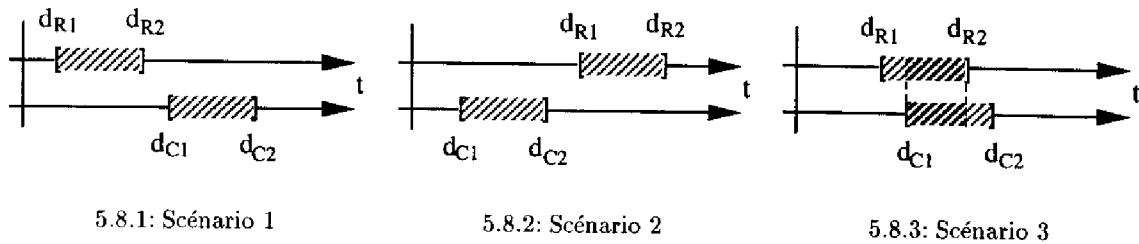


FIG. 5.8 - Trois scénarios possibles

défaillance qui pourrait permettre, dans certains cas, un traitement local.

Cependant, la défaillance constatée peut être une conséquence d'une déviation de comportement préalable qui n'aurait pas été détectée. Les informations disponibles sont alors insuffisantes pour mener à bien le processus de diagnostic. Ce cas fait apparaître la nécessité de mémoriser le parcours du lot lors du déroulement de la recette afin de faciliter le diagnostic. Une «double mémorisation» est possible: la mémorisation du parcours du lot sur le modèle de la commande et la mémorisation des opérations effectuées par chaque unité sur les modèles de référence. Ce problème de mémorisation, qui a déjà été abordé dans les travaux de [Combacau 91] sur les systèmes à événements discrets, sera traité au prochain chapitre.

La mémorisation du parcours du lot et des traitements effectués lors du déroulement de la recette, est d'autant plus importante qu'elle est nécessaire dans le cadre de la traçabilité de la production, obligatoire dans la certification<sup>8</sup> ISO 9002.

La séparation de la détection permet d'affiner la surveillance des variables indépendamment sur le modèle du procédé et sur le modèle de commande. Il est ainsi parfaitement possible de surveiller certaines variables à certains moments ou bien autour de certaines valeurs. Les seuils surveillés ne correspondent plus à des fins de configuration du procédé industriel, ils sont associés à des anomalies. L'exemple représenté sur la figure 5.9 illustre cet aspect. Lorsque le réservoir est dans une configuration isolée, c'est à dire sans connexion avec d'autres unités (appareils), la valeur de la variable de pression est surveillée. La prise en compte de l'événement d'état correspondant au dépassement du seuil spécifié est effectué par la transition  $t_x$ . Si l'événement se produit lorsque le réservoir est dans cette configuration (place  $p_1$  marquée), la transition  $t_x$  est franchie et un message est émis à destination du module prévu pour traiter cette information. Par contre si l'événement est détecté dans une configuration autre que celle considérée (place  $p_1$  vide), l'événement est ignoré puisque sa transition associée n'est pas sensibilisée.

De manière identique il est possible de rajouter à la partie commande du modèle de

8. certification de la qualité imposant notamment l'archivage des données techniques.

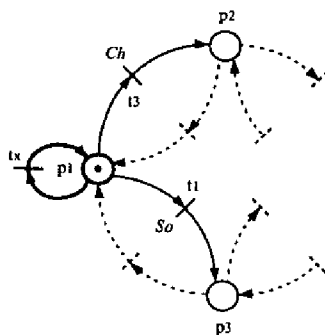


FIG. 5.9 – Détection d'anomalies exprimée sur le modèle de référence

supervision, une partie spécifique de surveillance. Un exemple d'illustration est donné sur la figure 5.10. Le schéma représente un morceau de réseau de Petri modélisant un module quelconque qui doit traiter une requête. La requête est reçue dans la place  $p_a$ . Afin de s'assurer que la requête sera traitée dans un délai maximum  $T_{max}$ , un chien de garde, exprimé par l'intervalle  $[0, T_{max}]$ , est associé à la transition de sortie de la place  $p_a$ . Si la requête n'a pas été considérée avant  $T_{max}$ , la transition  $t_a$  n'est plus franchissable, un dysfonctionnement de la commande a été détecté.

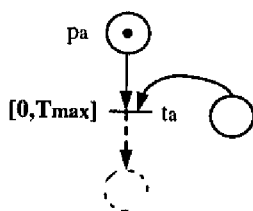


FIG. 5.10 – Détection d'anomalies exprimée sur le modèle de la commande

Nous avons jusque là séparé l'aspect commande et l'aspect procédé, cependant la mise en œuvre de stratégies intermédiaires sur le modèle de la commande est possible (surveillance de seuils ne correspondant pas à des fins d'activités).

Nous avons illustré l'utilisation du modèle de référence dans le cadre de la surveillance du comportement du procédé. Précisons bien que la détection d'une *alarme*, c'est-à-dire d'un dépassement d'une valeur seuil correspondant à une sécurité, est effectuée au plus bas niveau, et n'est donc pas du ressort du modèle de référence.

Intéressons-nous maintenant à l'utilisation du modèle de référence en émulation. Dans ce cadre l'objectif est de s'assurer de la cohérence (la compatibilité) d'une commande par rapport à l'état du procédé, en prenant en compte sa dynamique et non plus seulement son état statique.

## 5 Emulation du modèle de référence

Le modèle est émulé afin de vérifier que l'application de la commande n'engendre pas de violation de contraintes (celles-ci étant exprimées sur le modèle de référence). La faisabilité de l'opération, pour un fonctionnement normal, peut ainsi être assurée.

Pour illustrer l'utilisation du modèle dans ce cadre reprenons l'exemple précédemment évoqué de deux recettes qui s'exécutent en parallèle (Figure 5.5), et qui utilisent un même réservoir intermédiaire  $ST$  (Figure 5.4). Considérons à nouveau qu'à l'instant  $t$  considéré la première recette est en phase de traitement du lot dans le réacteur  $R_1$ , et la seconde en phase de chargement du réacteur  $R_2$ .

Pendant son chargement le réacteur  $R_2$  est couplé au réservoir intermédiaire  $ST$  qui contient le produit. Le modèle de référence du réservoir se trouve alors dans la configuration "réservoir en soutirage" (place  $p_3$  sur la figure 5.11).

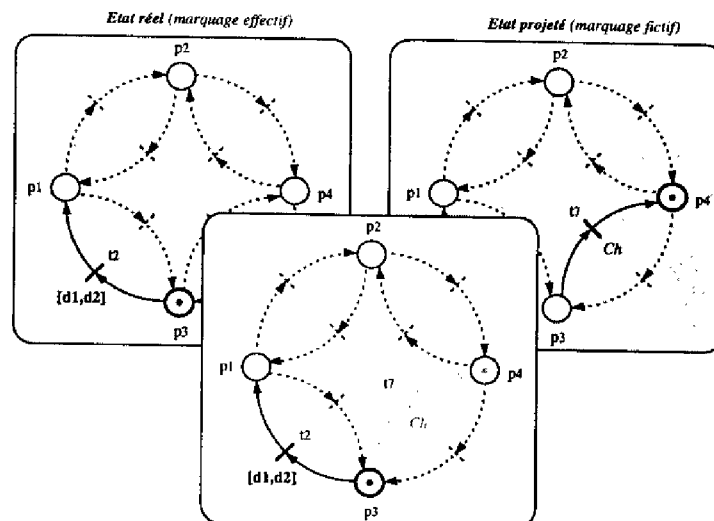


FIG. 5.11 – Emulation du modèle de référence du réservoir  $ST$

Supposons que la réaction dans le réacteur  $R_1$  soit terminée. La phase suivante de cette recette peut être activée : il s'agit du transfert du lot contenu dans le réacteur  $R_1$  vers le réservoir  $ST$ .

L'activation de cette phase de transfert est cependant conditionnelle : elle dépend des états discret et continu du réservoir. En ce qui concerne son état discret le réservoir doit être disponible pour cette opération et la configuration "remplissage-soutirage" doit être autorisée : dans l'état actuel ces conditions sont vérifiées. Au niveau de son état continu, son volume doit permettre de réceptionner le lot à transférer.

Or, tant que la transition  $t_2$  n'est pas franchie, la valeur de la variable de volume du réservoir  $ST$  n'est pas mise à jour (la valeur connue est celle avant le chargement du



réacteur  $R_2$ ). A ce niveau d'abstraction, comme nous l'avions précisé, l'état continu n'est connu avec «précision» que lors des événements de changement de configuration.

La décision de lancement du transfert du lot vers  $ST$  nécessite par conséquent la prise en compte de la dynamique continue afin de déterminer la faisabilité de l'opération dans le respect des contraintes physiques. La connaissance sur l'état «statique» n'est pas suffisante. Ces contraintes se limitent dans ce cas à la contrainte sur le volume maximum du réservoir à ne pas dépasser.

Le modèle peut alors être émulé afin de vérifier que l'évolution du volume du réservoir induite par l'application de la commande (opération de transfert du lot de  $R_1$  sur  $ST$ ) n'engendre pas de dépassement du volume maximum.

Il est important que l'émulation soit effectuée sur le modèle de référence. La projection temporelle est ainsi effectuée à partir de l'état courant du modèle du procédé ce qui évite le problème de maintien de la cohérence rencontré lorsque différents modèles sont utilisés à des fins différentes (un modèle pour la détection, un deuxième pour l'émulation, etc...).

L'utilisation du modèle de référence dans ce cadre nous amène alors à considérer deux marquages sur une même structure. Cela justifie la présence de deux jetons dans le modèle de référence décrit sur la figure 5.11. Le jeton de la place  $p_3$  (marque noire) correspond à l'état «effectif» du modèle de référence alors que le jeton contenu dans la place  $p_4$  (marque grise) représente l'état «fictif» du modèle, c'est-à-dire l'état du modèle projeté.

Lors de l'émulation, la commande est appliquée sur le modèle de référence mais pas sur le procédé. Ceci se traduit dans notre exemple par le franchissement de la transition  $t_7$  sur le modèle de référence du réservoir. Le jeton qui franchit cette transition est le jeton «fictif» qui change d'état discret : il passe de son état initial qui est l'état courant du modèle, la place  $p_3$ , à l'état discret imposé par la commande (simulée), la place  $p_4$ .

A partir du système d'équations associé à cette place (place  $p_4$ ), et de la valeur de la variable continue correspondant au volume (variable portée par le jeton), il est possible d'étudier le comportement dynamique du réservoir suite au lancement du transfert du lot contenu dans  $R_1$ .

Deux scénarios sont alors possibles :

- Les variations de volume induites par le remplissage et le soutirage simultanés du réservoir se «compensent» partiellement donc le transfert n'engendre pas de dépassement du seuil maximum du volume du réservoir. Le lot complet peut être transféré en une seule étape.
- La variation de volume induite par le remplissage du réservoir provoque un dépassement du seuil maximum. Dans ce cas deux décisions sont possibles : soit celle de

transférer le lot par morceaux soit d'attendre qu'un transfert du lot complet soit possible. Cependant, le transfert du lot en plusieurs étapes n'est possible que si ce fonctionnement a été modélisé. En effet, si le transfert du lot par étapes n'a pas été prévu les dates de fin d'activité de transfert précisées sur le réseau de la recette ne seront pas respectées (dates calculées pour un transfert en une étape) et une défaillance sera signalée.

Ce type de projection temporelle du comportement du procédé peut être également utile dans le cadre de la fonction de recouvrement de défaillances [Valette *et al.* 95]. Cela peut permettre effectivement de tester différentes procédures de recouvrement en s'assurant qu'elles respectent les contraintes imposées.

Les différents aspects du modèle de référence ayant été détaillés, revenons à la «construction» du système de commande/supervision afin de situer le modèle de référence.

## 6 Le modèle de référence au sein du système de commande/supervision

La position du modèle de référence est décrite sur la figure 5.12.

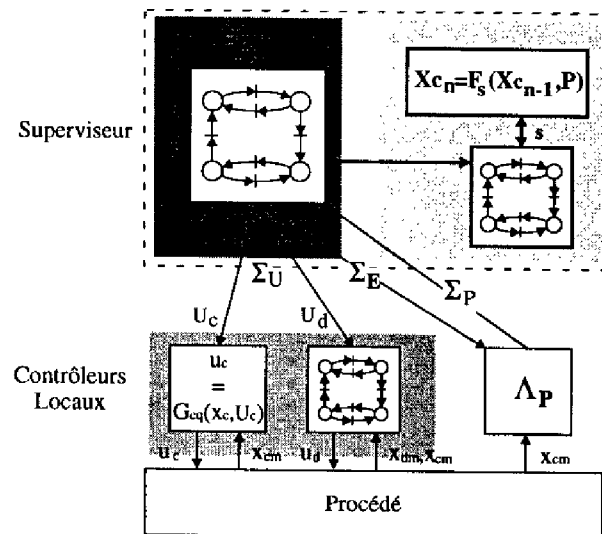


FIG. 5.12 – Supervision basée sur un modèle hybride

Le couplage dynamique entre le modèle de référence et le modèle de la commande garantit que l'état discret  $X_s$  du modèle de référence ( $X_s$  sous-vecteur de  $X_d$  qui est le vecteur d'état discret du superviseur) est cohérent avec la configuration du procédé. A chacun de ces états discrets  $X_s$  correspond un ensemble d'équations algèbro-différentielles  $F_s$  traduisant la dynamique de l'évolution de la composante continue du superviseur  $X_c$ .

$$X_{c_n} = F_s(X_{c_{n-1}}, P) \quad (5.2)$$

La composante continue  $X_c$  est mise à jour sur l'occurrence d'événements d'état  $P$  issus du générateur d'événements. L'occurrence de l'événement  $P$  provoque l'évolution simultanée du modèle de la commande et du modèle de référence (si la date d'occurrence de l'événement est compatible avec la fenêtre temporelle, et si l'écart entre les valeurs simulées et mesurées est inférieur au seuil spécifié). L'état discret suivant  $X_{d_n}$  du modèle de la supervision est alors calculé :

$$X_{d_n} = F_d(X_{d_{n-1}}, P) \quad (5.3)$$

## 7 Résumé des interactions discret-continu

Le système dynamique hybride peut être considéré, dans une représentation simplifiée, comme un ensemble de «blocs» ayant des interactions entre eux [Quénec'hdu *et al.* 95]. Il est possible de distinguer un bloc «continu», un bloc «événementiel» et un bloc «procédé», dont les interactions s'effectuent à travers des interfaces. Les interactions sont représentées sur la figure 5.13.

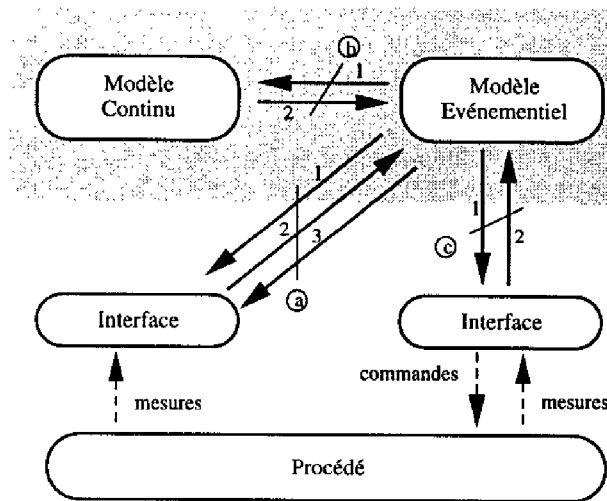


FIG. 5.13 – L'interface discret-continu

### Interaction entre le bloc événementiel et le bloc continu

Cette interaction est interne au superviseur, toutes les autres sont externes. Rappelons que le bloc événementiel correspond à un ensemble de réseaux de Petri, et le bloc continu

à un ensemble d'équations algébro-différentielles. Son interaction avec le bloc événementiel s'effectue en deux temps (respectivement les arcs 1 et 2 de l'interaction  $\textcircled{b}$  sur la figure 5.13) :

1. La sélection des équations actives du bloc continu. En ce qui concerne le modèle continu du procédé une sélection automatique et cohérente est effectuée par le modèle de référence. Pour le modèle continu de la commande son activation est effectuée, au besoin, directement par le modèle événementiel de la commande.
2. La prise en compte du résultat de la résolution du système d'équations. Ce résultat est utilisé différemment selon la nature du modèle continu. Cette interaction portée par les transitions peut avoir deux finalités : soit le forçage du franchissement (prise de décision cf. section 3.3.1), soit l'association d'une fenêtre temporelle de franchissement (surveillance cf. section 3.3.2).

#### Interaction entre le bloc événementiel et le procédé

Du procédé vers le bloc événementiel, l'interface est représentée par le générateur d'événements. L'interaction est constituée de deux ou trois étapes selon la règle de génération d'événement considérée (respectivement les arcs 1, 2, et 3 de l'interaction  $\textcircled{a}$  sur la figure 5.13) :

1. La configuration dynamique du générateur d'événements. Le générateur d'événements est programmé par le bloc événementiel : la règle de génération, l'identificateur de la variable, et les paramètres de la détection sont indiqués, ainsi que l'identificateur des variables dont la valeur doit être retournée avec l'événement. Les paramètres dépendent de la règle : soit un taux de variation dans le cas d'une génération sur taux, soit un temps dans le cas d'une génération périodique, soit un seuil dans le cas d'une génération sur franchissement de seuil.
2. La réception de l'événement s'effectue par l'intermédiaire des transitions du modèle à événements discrets (bloc événementiel). Il s'agit soit d'un événement d'état soit d'un message de mise à jour des images des variables d'état utilisées dans certains modèles continus comme les lois de commandes (cf. section 3.3.1).
3. Dans le cas où il s'agit de mise à jour d'images de variables d'état continues, le générateur d'événements doit être reconfiguré dès que l'actualisation n'est plus nécessaire (suppression de la variable de la liste des variables à considérer).

Du bloc événementiel vers le procédé, l'interface est constituée par les organes de commande continus et discrets. L'interaction peut être bidirectionnelle (respectivement les arcs 1 et 2 de l'interaction  $\textcircled{c}$  sur la figure 5.13) :

1. Activation, ou programmation des organes de commande locaux (changement de loi de commande sur les contrôleurs).
2. Emission de compte-rendus d'exécution des organes de commande (dans la mesure où les organes peuvent supporter cette fonction).

## 8 Conclusion

A travers la «construction» d'un système de commande/supervision adapté aux procédés discontinus nous avons proposé une architecture informatique répartie au sein de laquelle une interface entre le procédé et le superviseur a été introduite: le *générateur d'événements*. D'autre part nous avons mis en évidence la nécessité de prendre en compte l'aspect continu (modèle comportemental du procédé) au sein du superviseur, ce qui nous a amené à définir un modèle hybride pour la supervision.

Conformément à notre choix d'une approche de modélisation séparée, l'homogénéité de chaque composante est respectée. Cette coexistence des aspects événementiels et continus nous a amené à considérer les problèmes de la coopération des modèles et de la cohérence de leurs évolutions.

L'introduction du *modèle de référence* de nature *hybride* offre une vue qualitative du comportement du procédé. Son évolution est couplée à celle du modèle événementiel de la commande et assure ainsi une sélection automatique et cohérente du modèle comportemental continu du procédé. Dans le cadre de la surveillance ce modèle permet la détection de déviations dans le comportement du procédé. La possibilité de l'émuler rend possible son utilisation aussi bien dans le cadre de la vérification de la cohérence d'une commande, que dans celui du test des procédures de recouvrement de défaillances.

La séparation de l'aspect procédé et de l'aspect commande offre modularité et ré-utilisabilité (généricité) des modèles.

Ce chapitre a présenté les divers aspects de notre approche de la supervision des procédés discontinus sans aborder l'aspect mise en œuvre des principes évoqués dont, entre autres, la simultanéité d'évolution des modèles basée sur le principe de la fusion de transitions, le mécanisme de comparaison dans le cadre de la détection, et la mémorisation du parcours des lots et des traitements effectués sur les unités. Nous allons donc nous pencher maintenant sur ces différents points.

# Chapitre 6

## Les détails de la méthode

### 1 Le joueur de réseaux de Petri

Il est possible de classer les méthodes de mise en œuvre d'un réseau de Petri suivant qu'elles sont fondées sur une approche procédurale ou déclarative. La première approche consiste à écrire des programmes dont le comportement sera identique à celui spécifié par réseau de Petri (l'approche procédurale correspond à une mise en œuvre compilée). La seconde consiste à considérer qu'un réseau de Petri est un système de règles, et à baser sa mise en œuvre sur l'utilisation d'un mécanisme d'inférence spécialisé appelé *joueur de réseau de Petri* (l'approche déclarative correspond à une mise en œuvre interprétée) [Bako *et al.* 90].

Les avantages principaux de cette dernière approche sont l'évitement des erreurs dues à la traduction dans un langage de programmation, l'accès possible à l'état global du système à travers le marquage du réseau, et la possibilité d'analyser des solutions envisageables à un instant donné. De plus l'approche procédurale n'est pas particulièrement adaptée pour la détection et le diagnostic [Valette & Künzle 94]. Par contre l'approche déclarative permet la mise en œuvre automatique de la détection.

Le réseau de Petri peut être considéré comme un système de règles particulier décrivant des changements d'états. Chaque transition est une règle décrivant une transformation entre un ensemble d'états de départ (marquage courant) vers un ensemble d'états suivants (marquage obtenu après le déplacement des jetons). Le joueur déplace les jetons en respectant les règles de franchissement des transitions, jusqu'à atteindre un état stable (Figure 6.1). L'état stable d'un réseau est un marquage pour lequel seules des transitions associées à des événements externes sont franchissables. Le joueur se place donc en attente de l'occurrence d'un événement externe. Quand un événement externe se produit (réception d'un message), le joueur cherche la transition à laquelle il est associé, et teste si elle est franchissable. Si la transition est franchissable il la franchit, ainsi que toutes celles rendues franchissables par le nouveau marquage, jusqu'à atteindre un nouvel état stable. Par contre si la transition ne peut pas être tirée, le joueur détecte une anomalie et reste dans l'état stable : c'est un

mécanisme de détection directement intégré à la commande.

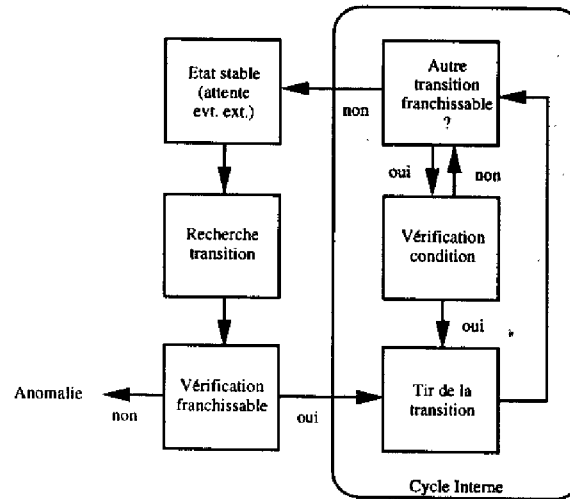


FIG. 6.1 – Principe du joueur

## 2 La partie commande

### 2.1 Homogénéité de la modélisation

La possibilité d'utiliser différentes classes de réseaux de Petri à différents niveaux offre une modélisation homogène de l'aspect discret du système de commande/supervision.

La classe de réseaux de Petri utilisée à chaque niveau dépend bien évidemment des besoins de modélisation : prise en compte explicite de l'aspect temporel, manipulation de structures de données, etc...

Ainsi les réseaux de Petri utilisés pour modéliser la recette sont les réseaux de Petri à Objets. La structure du réseau correspond à la procédure de la recette et les objets manipulés sont d'une part les objets <lot>, et d'autre part les objets <ressource>. Les premiers contiennent l'ensemble des informations de la recette principale concernant la fabrication d'une classe de produit et les caractéristiques du lot à produire dont son identité, son volume, etc. Un ensemble d'informations concernant les opérations à exécuter est encapsulé dans cette structure de données : les paramètres de chaque opération comme par exemple les débits pour les transferts de matière, les profils de température pour la conduite thermique, etc., ainsi que des informations issues de l'ordonnancement décrivant le plan de production à suivre, avec notamment l'identité des unités à utiliser<sup>1</sup> et les dates d'exécution pour chaque opération.

1. selon la flexibilité décisionnelle laissée à la commande soit une classe d'unité, soit l'identité de l'unité est désignée.

Les objets <ressource> correspondent à une description de chaque ressource en précisant par exemple la liste des opérations exécutables et les caractéristiques techniques. Sont considérées comme ressource toutes les unités de l'installation ainsi que les vannes de commutation<sup>2</sup> et les sections<sup>3</sup>. L'objet associé à une section comprend son identité et ses caractéristiques comme son flux maximum, ses connexions aux autres sections et/ou unités, la dernière classe de produit transférée, la liste des vannes aux frontières, etc.

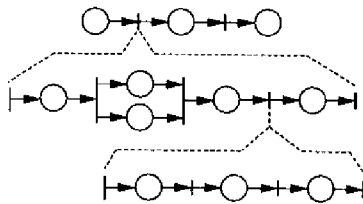
La possibilité de modification des ressources à travers les objets, indépendamment de la structure du réseau (procédure de la recette), permet une adaptation rapide d'une recette principale à une autre installation ou aux changements sur l'installation courante.

Les réseaux de Petri à objets sont également utilisés pour la description des fonctions. Les objets manipulés sont les requêtes émises par la recette, contenant l'identité de la requête et les paramètres opératoires (débits, temps, profils de température, etc.), et les appareils constituant les unités qui sont considérés comme des ressources à ce niveau (leur description est similaire à celle des ressources du niveau recette).

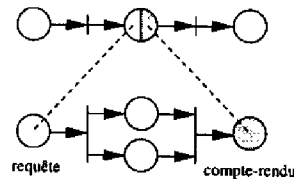
Au plus bas niveau, la classe de réseaux de Petri utilisée est le réseau de Petri binaire interprété ou le grafcet qui, sous certaines restrictions, est équivalent [David & Alla 89].

## 2.2 La communication entre les niveaux

Le système de commande/supervision est hiérarchisé et modulaire, le modèle de la commande correspond donc à un ensemble de réseaux de Petri qui communiquent entre eux. La technique de décomposition structurelle du réseau est fréquemment une décomposition par «substitution», c'est-à-dire par affinement des places ou des transitions (Figure 6.2).



6.2.1: Affinement des transitions



6.2.2: Affinement des places

FIG. 6.2 – Décomposition par «substitution»

Pour la représentation des flux de requêtes et de compte-rendus, nous utiliserons le

- 
2. vannes deux états permettant d'établir les configurations du flowsheet.
  3. une section correspond à un ensemble de connexions utilisables de manière isolée, c'est-à-dire délimitée par des vannes de commutation ou des unités.



formalisme de la figure 6.2.2. La requête émise est modélisée par un jeton (un objet) dont les attributs contiennent les informations nécessaires à son exécution. Le compte-rendu est modélisé par le même jeton, permettant ainsi une identification directe du compte-rendu (attribut identificateur de la requête contenu dans le jeton retourné).

La plupart des interactions se résume par l'envoi d'une requête au niveau inférieur et l'attente du compte-rendu retourné dès la fin d'exécution de la requête. Toutefois lorsque la requête est destinée à l'activation d'une boucle de commande, il n'y a pas de compte-rendu de fin d'exécution. C'est au niveau supérieur de décider des instants de désactivation de la boucle de commande, ce qui est le cas par exemple de l'interaction avec les régulateurs locaux.

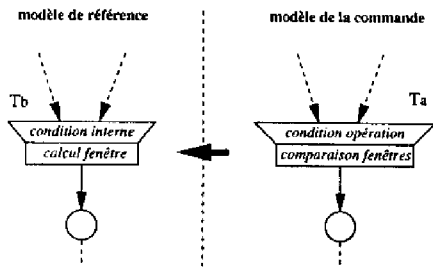
### 2.3 La communication entre la partie commande et le modèle de référence

Le mécanisme de communication entre le modèle de la commande et le modèle de référence conditionne le fonctionnement de la structure de commande-surveillance. Nous avons considéré jusqu'ici qu'elle est basée sur le principe de la fusion de transitions. Si nous fusionnons les transitions nous obtenons un modèle global qui peut alors être validé par les techniques classiques d'analyse des réseaux de Petri (réduction, arbre des marquages accessibles, invariants, etc.). La validation de l'ensemble des réseaux n'apparaît donc pas comme une source de problème.

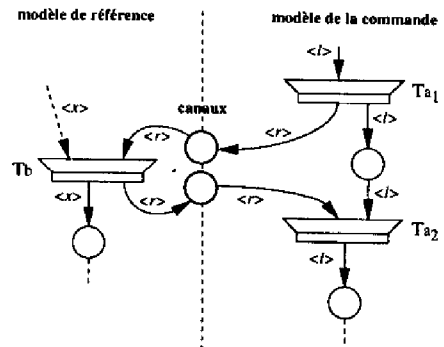
Pour la spécification du lien entre les transitions nous allons nous appuyer sur la communication entre le bloc de référence et le bloc de commande définie dans les travaux de Combacau [Combacau 91].

Par le fonctionnement que nous avons choisi le lien entre les transitions n'est pas symétrique puisqu'il y a une transition appelante (modèle de la commande) et une transition appelée (modèle de référence) (Figure 6.3.1). La spécification de ce lien par réseau de Petri est décrite sur la figure 6.3.2. La transition appelante  $T_a$  est décomposée en deux transitions  $T_{a_1}$  et  $T_{a_2}$  connectées à la transition  $T_b$  du modèle de référence par des places qui constituent les canaux à travers lesquels les informations sont échangées (places contenant les messages). Deux canaux unidirectionnels suffisent pour établir une communication entre l'ensemble des transitions d'un modèle de référence et l'ensemble des transitions du modèle de la commande (places © et ®). Chaque canal peut être considéré comme une boîte aux lettres dans laquelle chacun des modèles dépose ses requêtes et ses réponses (Figure 6.3.3).

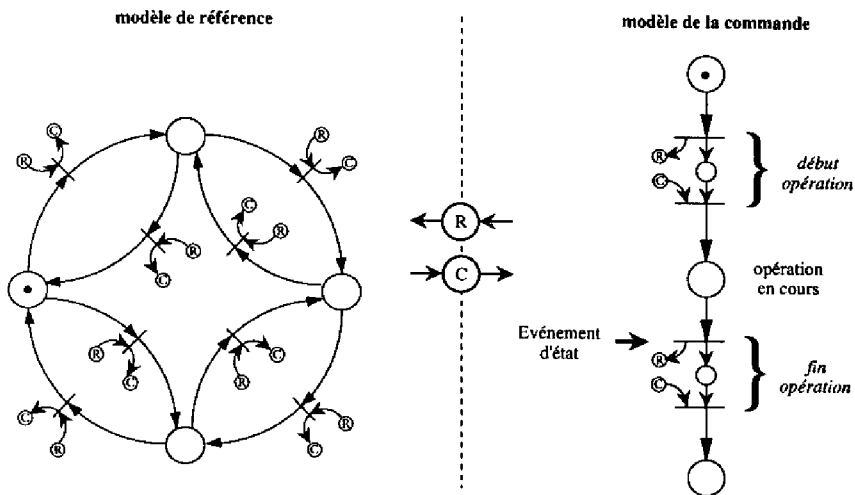
Le jeton transmis (la requête) est porteur des informations caractérisant l'opération que la commande veut exécuter sur l'unité considérée, telles que les paramètres de l'opération : le volume et le débit d'une opération de transfert par exemple. Ces informations sont nécessaires au modèle de référence afin d'autoriser ou non l'exécution de l'opération (test



6.3.1: Lien commande-référence



6.3.2: Spécification du lien



6.3.3: Deux canaux unidirectionnels pour un modèle de référence

FIG. 6.3 – Communication entre les modèles

des conditions internes), ainsi que pour calculer la fenêtre temporelle de fin d'opération si celle-ci est autorisée. Le jeton retourné comprend notamment les bornes de la fenêtre temporelle calculée, permettant au modèle de la commande d'établir la comparaison avec la fenêtre temporelle issue de l'ordonnancement. Les informations échangées permettent également l'identification des requêtes et des réponses. Cette identification est indispensable puisque plusieurs opérations étant parfois exécutables simultanément sur une même unité (cas du réservoir intermédiaire par exemple), plusieurs requêtes et réponses peuvent être échangées parallèlement.

Cette spécification de la synchronisation du franchissement des transitions correspond à un mode appel/réponse. Ce mode de synchronisation est important car, étant bloquant, il garantit la prise en compte de la réponse fournie par le modèle de référence. En effet, la deuxième transition  $T_{a_2}$  de la transition appelante  $T_a$  ne peut pas être franchie tant que la réponse n'a pas été retournée. Spécifier cette synchronisation par un mode appel simple, c'est-à-dire par l'intermédiaire d'une seule place de communication, ne serait pas conforme au fonctionnement désiré car le modèle de la commande pourrait évoluer sans attendre l'autorisation du modèle de référence.

L'évolution des deux réseaux de Petri n'est pas strictement synchrone mais elle a tout de même lieu dans le même cycle du joueur puisque ce dernier ne peut pas être dans un état stable tant qu'il n'a pas franchi les transitions appelante et appelée.

## 2.4 La mémorisation du parcours du lot traité

Nous nous intéressons au principe de mémorisation des informations mais nous n'aborderons pas leur utilisation dans le cadre du diagnostic ni dans celui de l'analyse statistique de la production.

Nous avons adopté un mécanisme similaire à celui proposé par Combacau, qui repose sur le principe de confier au moteur d'inférence (le joueur) le rôle de gestionnaire de l'historique des traitements effectués [Combacau 91]. La mémorisation est basée sur la création de traces associées à chaque transition de début d'activité (Figure 6.4). Une trace est comparable à une liste dans laquelle le joueur stocke<sup>4</sup> les contextes d'exécution constitués par une copie des jetons utilisés pour le franchissement de la transition considérée.

La destruction des traces, utilisées à des fins de diagnostic uniquement dans le cadre de ces travaux, est effectuée après l'exécution correcte de la dernière activité de la séquence de commande, c'est-à-dire au moment où le compte-rendu d'exécution est retourné au niveau supérieur de commande (la supervision). La destruction est prise en charge par le joueur.

En se basant sur ces travaux, nous proposons de mémoriser l'historique de traitement sur les jetons (Figure 6.5), car une fois le lot traité, c'est-à-dire la recette terminée, ces

---

4. rangement par ordre chronologique.

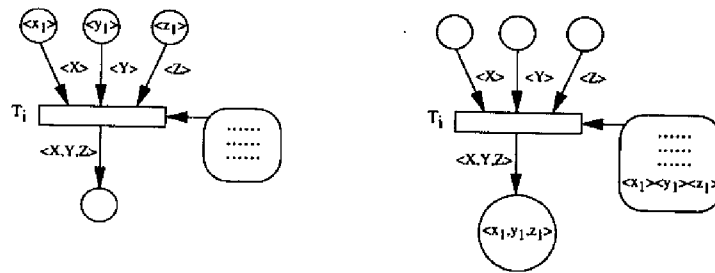


FIG. 6.4 – Association de traces aux transitions [Combacau 91]

informations, qui ne sont peut-être plus utiles dans le cadre du diagnostic, sont requises pour la traçabilité de la production et doivent être archivées.

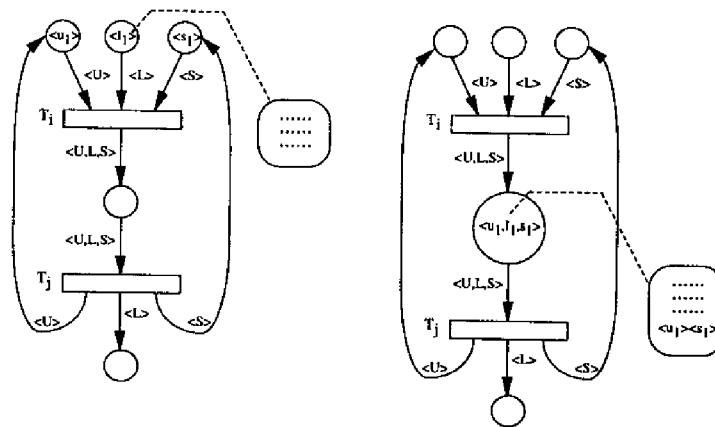


FIG. 6.5 – Association de traces au jeton

La destruction des informations concernant le parcours du lot traité est donc remplacée par l'archivage. Afin de ne pas pénaliser le temps de cycle du moteur d'inférence, la gestion de l'archivage est confiée à un gestionnaire autre que le joueur qui n'est alors en charge que de la mémorisation du traitement du lot sur le jeton.

A chaque étape du traitement du lot, l'ensemble des unités impliquées (les ressources), les opérations effectuées, et les conditions d'exécution (conditions opératoires) sont mémorisées. Cette mémorisation du contexte d'exécution s'effectue lors du franchissement des transitions de début d'activité. Le passage d'information qui a lieu entre le modèle de la commande et le modèle de référence (cf. section 6.3), permet d'établir des liens entre l'historique des traitements effectués sur les unités, et l'historique de la fabrication du lot.

Dès l'instant où la recette de contrôle d'un lot donné est terminée, les informations mémorisées qui sont portées par le jeton correspondant au lot traité, sont archivées.

### 3 Le modèle de référence hybride

#### 3.1 Le modèle à événements discrets utilisé

Le modèle utilisé pour la représentation de l'aspect discret du modèle de référence hybride doit permettre la manipulation d'informations, ainsi que l'expression du mécanisme de détection basé sur le chien de garde : le modèle à événements discrets est donc nécessairement un réseau de Petri de "haut-niveau" sur lequel le temps est pris en compte de manière explicite. Les réseaux de Petri temporels à objets [Cardoso 90] sont un modèle adéquat.

Ce modèle permet d'associer à chaque transition un intervalle de temps, une fenêtre temporelle  $[d_i, d_j]$ , dont les bornes dépendent des attributs des objets sensibilisant la transition. Le calcul des bornes est effectué lorsque la transition est sensibilisée, à partir des attributs des objets qui apparaissent dans la substitution consistante, et des variables globales du réseau de Petri. Ces dates étant des durées de sensibilisation et non des dates absolues, il est nécessaire de prendre en compte la date de marquage des places auxquelles sont associées des transitions avec chien de garde.

Comme nous l'avons indiqué, le jeton déplacé dans ce modèle hybride est porteur d'information : il s'agit d'un objet dont les attributs sont entre autres les contraintes physiques de cette dernière, et la valeur des variables d'état continues caractérisant l'unité.

Le jeton comporte également un ensemble d'attributs permettant la désignation de l'ensemble des équations associées à l'état au sein duquel il se trouve, ainsi que la description de l'événement de fin de configuration en indiquant la variable considérée, la valeur du seuil et son sens de franchissement. La valeur du seuil est précisée dans la requête transmise par le réseau de la commande, par contre le sens de franchissement de ce seuil est connu implicitement, il est déduit de la configuration courante de l'unité (état discret marqué).

Précisons à nouveau que la technique de résolution numérique du système d'équations n'est pas abordée dans nos travaux, nous renvoyons le lecteur aux travaux de Daubas [Daubas 94]. Nous considérons le réseau de Petri comme un moniteur d'équations qui désigne l'ensemble des équations actives, ces dernières étant mises à disposition d'un moniteur de calcul comprenant un intégrateur pour la résolution numérique, et un interpréteur assurant l'interface<sup>5</sup> entre ce moniteur de calcul et le moteur d'inférence du réseau de Petri.

#### 3.2 La mémorisation des activités exécutées sur l'unité

Le principe de mémorisation des activités exécutées sur une unité est identique à celui concernant la mémorisation de la fabrication d'un lot. Cependant le moment d'archivage est différent.

---

5. gestion et mise en forme des informations échangées entre la partie continue et la partie discrète.

Reprenons la figure 6.3.2. Lors du lancement d'une activité, c'est à dire lors du franchissement de la transition  $T_b$ , le moteur d'inférence édite un lien entre la requête (objet  $\langle r \rangle$ ) et l'unité (objet  $\langle x \rangle$ ) en recopiant les identificateurs respectifs dans chacun des objets, ainsi que la date. C'est par l'intermédiaire de cet échange daté qu'un lien peut être établi entre l'historique des traitements effectués sur les unités et l'historique de la fabrication des lots, permettant ainsi ultérieurement une reconstitution précise de l'historique d'une campagne de production.

Plusieurs activités pouvant avoir lieu sur une même unité l'archivage n'est effectué que lorsque toutes ces activités sont terminées. Il est facile de déterminer sur le modèle de référence l'instant auquel les activités de transfert sont achevées, il s'agit du retour dans la configuration isolée. Cependant sur le modèle de référence il n'est pas toujours possible de savoir si un traitement a lieu au sein de cette configuration. Cela dépend de l'unité que représente le modèle de référence. Dans une configuration isolée, aucun traitement n'a lieu dans le cas d'un réservoir intermédiaire, alors qu'une réaction peut être en cours dans le cas d'un réacteur. Afin de rendre systématique la décision d'archivage nous archivons l'objet (le jeton) dès son retour dans la configuration isolée. Lors de cet archivage un nouvel objet est créé pour remplacer l'objet archivé. Certains de ses attributs sont mis à jour dès sa création comme la valeur des variables d'état continues par exemple, et un lien est édité avec l'objet archivé afin de pouvoir chaîner l'historique des activités exécutées sur l'unité durant la campagne.

Nous avons détaillé certains points de la méthode, abordons maintenant quelques considérations sur le générateur d'événements qui constitue un point sensible du système.

## 4 Le générateur d'événements

### 4.1 La notion d'événement

L'identification et la détermination de l'occurrence d'un événement est un aspect délicat [Lemmon *et al.* 93]. Notre approche nous a permis de définir les événements à partir de la notion de configuration active, nous allons considérer maintenant quelques aspects importants quant à la détection de leur occurrence.

Un événement est essentiellement défini comme le franchissement d'un seuil par une variable d'état continue (ou encore la norme d'un vecteur changeant de région).

D'autre part le sens par lequel le seuil est atteint est également important, il est un des éléments définissant l'événement [Engell *et al.* 95]. Prenons le cas du niveau d'un réservoir : un seuil atteint par croissance ou par décroissance n'a pas la même signification, il correspond respectivement à un réservoir se remplissant et se vidant.

Toute combinaison d'événements doit être spécifiée sur le modèle à événement discret utilisé par la supervision (réseau de Petri), la notion de macro-événement

[Peleties & DeCarlo 89] basée sur la composition logique d'événements n'existe pas au niveau du générateur d'événements.

Il est évident que la pertinence de la détection d'un événement est fortement dépendante de la qualité de la mesure. Toute mesure bruitée (mesure de viscosité par exemple) risque d'engendrer la détection d'un «faux» événement et induire en conséquence une évolution incohérente de la commande par rapport à l'état réel du procédé. Un seuil ne doit pas être défini par une valeur stricte, il est nécessaire d'accorder une tolérance.

La détection d'un événement suppose également que la variable d'intérêt est mesurée. Or ce n'est pas toujours le cas car une des caractéristiques de ces procédés est justement que l'observation en est limitée. L'utilisation de modèles permettant la reconstitution de variables d'état non mesurées à partir des mesures disponibles est une solution possible comme nous avons pu le constater à travers la conduite de l'unité de distillation. Dans cet exemple la concentration du produit est déduite à partir de la mesure de température. L'événement d'état n'est alors plus détecté par le générateur et la pertinence de l'événement dépend alors de la qualité du modèle.

Lorsque la détection d'un événement implique un ensemble de variables d'état mesurées, ce qui est le cas par exemple du changement de région sur la norme d'un vecteur, un problème de cohérence temporelle se pose. L'événement ne peut être pertinent que si toutes les mesures nécessaires à sa détermination correspondent à des échantillons de même date. D'où l'importance de garantir le respect des contraintes spatiales et temporelles sur le bus de terrain (bus de nature continue).

Cependant il est des cas pour lesquels le domaine de validité de chaque mesure doit alors être pris en compte. Considérons par exemple deux mesures dont la première est échantillonnée toutes les secondes et la deuxième toutes les 10 secondes. Le calcul effectué pour la détection de l'événement ne peut être effectué que toutes les 10 secondes car rien ne garantit que cet échantillon de la deuxième mesure reste valable pendant 10 secondes, c'est-à-dire jusqu'au prochain échantillon.

La période d'échantillonnage de chacune des mesures est connue par l'arbitre du bus de terrain<sup>6</sup> (table de l'arbitre), ainsi que par certaines applications utilisant ces valeurs, mais elle n'est pas censée être connue par le générateur d'événements. Nous considérons alors que le générateur ne peut évaluer l'occurrence d'un événement que lorsque toutes les mesures impliquées ont été simultanément (dans le même micro-cycle de communication) mises à jour, c'est-à-dire que tout calcul impliquant un ensemble de mesures n'est effectué que si elles sont toutes de la même date.

---

6. L'arbitre est un gestionnaire de la communication, il s'agit d'un contrôle d'accès au bus centralisé comme par exemple pour le réseau de terrain FIP [Pleinevaux & Decotignie 88].

## 4.2 L'algorithme du générateur

Le générateur doit exécuter séquentiellement un ensemble de tâches dans un cycle dont la durée est finie. Ce cycle comprend deux phases, la première concerne l'acquisition des mesures, et la seconde leur traitement (calculs). Comme pour toute application temps-réel il est impératif de garantir que le générateur est capable d'effectuer tous les traitements nécessaires entre deux acquisitions de mesures.

L'algorithme du générateur peut être décrit comme suit :

1. Acquisition des mesures sur le bus. Il s'agit de la lecture et de la mémorisation de l'image de chaque variable d'état continue de l'ensemble des variables sur lequel travaille le générateur.
2. Traitement des listes d'événements à détecter. A chaque type d'événement, caractérisé par la règle de génération qui lui est associée (génération sur seuil, génération périodique, génération sur taux), correspond une liste que le générateur parcourt lors du traitement.

Le traitement de l'ensemble des événements peut donc être décomposé en trois étapes :

- (a) Traitement des événements périodiques. Il s'agit simplement de «renvoyer» la valeur de la variable tous les  $n$  cycles (la période correspond à un multiple de cycles).
  - (b) Traitement des événements sur seuils. Le générateur effectue une comparaison entre la valeur mesurée de la variable et l'intervalle définissant le seuil (l'intervalle est défini par une valeur seuil précise  $\pm$  la tolérance fixée pour la variable considérée).
  - (c) Traitement des événements sur taux de variation. Le calcul est effectué par rapport à deux échantillons successifs de la variable examinée (la mémorisation de la valeur des variables d'état est limitée à deux échantillons consécutifs).
3. Mise à jour des listes d'événements, puis retour à l'étape 1. Lors de cette étape le générateur effectue une scrutation des nouvelles demandes de configuration (ajout et/ou suppression d'événements à surveiller), après avoir supprimé de la liste des événements sur seuil ceux détectés dans le cycle courant.

## 5 Conclusion

Ce chapitre nous a permis de clarifier certains points concernant la mise en œuvre de l'approche proposée.



Tout d'abord, nous pouvons retenir que l'utilisation possible de différentes classes de réseaux de Petri à chaque niveau de la hiérarchie représente un avantage important pour l'homogénéité de la modélisation, qu'il s'agisse du modèle de la commande ou du modèle du procédé. De plus, le mécanisme de communication entre ces deux modèles est un mécanisme relativement simple basé sur un mode appel/réponse.

En outre, le concept de joueur de réseaux de Petri est particulièrement intéressant puisqu'il permet une adaptation rapide de l'application aussi bien aux modifications de la stratégie de commande (nouvelles recettes) ou de la commande elle-même (nouvelles fonctions), qu'à celles concernant le procédé (changement d'un appareil). Cette adaptation, effectuée par la modification des modèles et/ou des objets manipulés, n'implique aucune programmation.

Enfin, les considérations que nous avons abordées au sujet du générateur d'événements montrent l'importance de cet élément dans notre système de commande/supervision. Il est l'articulation entre le système de supervision d'une part, et le procédé et sa commande locale d'autre part.

Après avoir présenté notre conception du système de commande/supervision des procédés batch jusqu'aux détails de mise en œuvre, nous allons illustrer son application à travers deux exemples. L'objectif de cette prochaine partie est la description du fonctionnement de notre approche notamment en ce qui concerne les interactions discret-continu internes à chaque niveau, ainsi que celles entre les différents niveaux de la hiérarchie.

# Troisième Partie

## Exemples d'applications



## Chapitre 7

# La conduite d'un procédé de distillation

Le procédé de distillation considéré est un procédé que nous avons présenté au premier chapitre (cf. section 2.2.1) et dont nous avons évoqué les différentes stratégies de conduite au quatrième chapitre (cf. section 1.1.2). Rappelons simplement que le processus consiste à séparer deux constituants  $A$  et  $B$  d'un produit initial.

Nous nous intéressons à nouveau à ce procédé afin d'illustrer la modélisation de la commande et du procédé aux niveaux supervision (recette) et coordination (fonctions). Nous détaillerons les interactions entre le modèle de la commande et le modèle de référence au sein de chaque niveau<sup>1</sup>, ainsi que les interactions entre ces deux niveaux d'agrégation.

L'objectif est également d'illustrer la gestion d'une équation au sein du modèle de la commande, cet aspect n'ayant pas été détaillé lors des précédents chapitres. Par contre nous ne reviendrons pas sur les calculs des fenêtres temporelles qui ont déjà été traités.

Les modèles que nous allons présenter sont basés sur les réseaux de Petri à objets, le lecteur non familier avec cette classe de réseaux de Petri en trouvera une description succincte en annexe. Nous utilisons en fait les réseaux de Petri temporels à objets, mais l'aspect temporel ne sera pas étudié à travers ces exemples.

### 1 La modélisation au niveau supervision

Le niveau supervision est le plus haut niveau temps-réel de la hiérarchie, il offre une vue agrégée aussi bien de la commande, à travers le modèle de la recette, que du procédé, à travers les modèles de référence des unités.

---

1. nb: pour la clarté des réseaux, ni les interprétations associées aux transitions, ni les interactions avec le générateur d'événements n'ont été représentées, et seulement quelques fusions de transition ont été décrites.

La vision du procédé à ce niveau est celle de quatre unités interconnectées : une unité de stockage de la matière première (unité d'alimentation), une unité de distillation, et deux unités de stockage des produits finaux.

La recette à exécuter pour la distillation d'un lot est la suivante :

- charger un lot dans la colonne, et s'il existe un résidu de coupe (issu de la distillation d'un lot précédent), charger le contenu du réservoir de coupe dans la colonne,
- chauffer le contenu, et maintenir la chauffe jusqu'à la vidange du pied de cuve,
- distiller le lot en changeant de stratégie à des instants définis en fonction de la concentration du distillat (cf. chapitre 2 section 1.1.2).
- vidanger le pied de cuve,
- vidanger le réservoir de distillat intermédiaire.

## 1.1 La recette et les ressources

La recette de contrôle modélisée par réseaux de Petri à objets est représentée sur la figure 7.1.3. Le réseau fait clairement apparaître la séquentialité et/ou le parallélisme des différentes phases de la recette. Le formalisme utilisé est décrit sur la légende associée à la figure.

Un lot est représenté par un objet  $\langle l_i \rangle$  qui correspond à une instance de la classe d'objet *Lot*  $\langle l \rangle$ . Un objet  $\langle l_i \rangle$  contient l'ensemble des informations de la recette de fabrication du lot  $i$  de la classe de produit considérée (cf. chapitre 6).

Sur le réseau de Petri représentant la recette est ajoutée la gestion des ressources. Les ressources peuvent être partagées par un ensemble de recettes identiques ou différentes s'exécutant en parallèle sur l'installation, il est donc indispensable de gérer leurs affectations en temps-réel. Toutefois la recette de commande est générée par les couches de gestion et les intervalles de temps affectés aux opérations doivent normalement correspondre à des instants pour lesquels les ressources nécessaires sont disponibles.

L'ensemble des ressources de cet exemple est représenté par deux classes d'objet : la classe d'objet *Unité de Stockage*  $\langle s \rangle$  représente les ressources de stockage de la matière, et la classe d'objet *Unité de Traitement*  $\langle u \rangle$  les ressources de traitement de la matière. Les instances d'objet utilisées sont données dans la légende de la figure 7.1.3 : par exemple, l'objet  $\langle c \rangle$ , représentant l'unité de distillation, est une instance de la classe d'objet  $\langle u \rangle$ . Une seule place<sup>2</sup> est utilisée sur le réseau de Petri pour représenter l'état disponible des ressources, la différence entre elles étant établie par l'individualisation offerte par le concept d'objet. Cette place joue le même rôle que celui du superviseur dans l'approche de Ramadge et Wohnam : elle interdit un ensemble d'états qui sont incohérents, de façon

<sup>2</sup> pour la clarté du réseau, celle-ci a été dupliquée sur la figure 7.1.3.

évidente.

L'unité de distillation (objet  $\langle c \rangle$  de la classe  $\langle u \rangle$ ) ne peut être utilisée que pour la fabrication d'un seul lot à un instant donné, elle est donc réservée dès le lancement de la recette, et elle est libérée dès la fin de la recette. Par contre, deux stratégies d'utilisation des unités de stockage ont été modélisées<sup>3</sup>:

- l'une exprime une utilisation exclusive d'une unité de stockage soit en soutirage soit en remplissage. C'est le cas par exemple du réservoir de matière première dont l'objet correspondant (objet  $\langle a \rangle$  de la classe  $\langle s \rangle$ ) est réservé dès le lancement de l'opération "*Charger lot*", et n'est libéré qu'une fois l'opération terminée. Ce réservoir, en soutirage de par l'opération qu'il subit, n'est pas disponible pour une opération de remplissage.
- l'autre exprime la possibilité d'une utilisation simultanée d'une unité de stockage à la fois en soutirage et en remplissage. Le seul exemple modélisé est celui du réservoir de produit final distillé dont l'objet correspondant (objet  $\langle d \rangle$  de la classe  $\langle s \rangle$ ) est pris et immédiatement restitué lors du lancement de l'opération "*Vider distillat*". L'implication de cette ressource dans une activité est alors mémorisée sur un des attributs de l'objet qui lui est associé. Cet objet étant rendu à nouveau disponible après le lancement de l'activité de soutirage (objet restitué après le franchissement de la transition), la ressource peut être utilisée par ailleurs pour une activité de remplissage, mais pas pour une autre activité de soutirage. Si une nouvelle requête de soutirage venait tout de même à être émise, le modèle de référence n'autoriserait pas l'exécution de cette requête (aucune transition n'est franchissable sur le modèle de référence pour cette requête). Dès que l'opération "*Vider distillat*" est terminée, l'objet associé au réservoir de produit final distillé (objet  $\langle d \rangle$  de la classe  $\langle s \rangle$ ), est à nouveau pris et restitué après mise à jour de ses attributs.

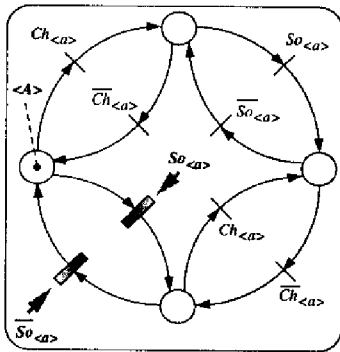
Lors du déroulement de la recette, l'implication d'un ensemble d'objets au sein d'un même état discret suite au tir d'une transition, est exprimée par la création d'un n-uplets, qui est constitué à partir des objets entrants de la transition (jetons contenus dans les places amont de la transition franchie). C'est ainsi que le 2-uplet  $\langle l, u \rangle$  est constitué à partir des singletons  $\langle l \rangle$  et  $\langle u \rangle$ , exprimant le lien dynamique, donc temporaire, entre un lot et une unité.

## 1.2 Les modèles de référence des unités

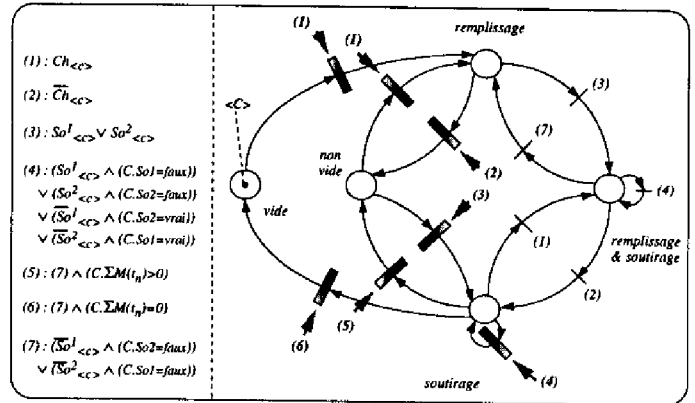
Chaque unité est représentée par un modèle de référence hybride, qui offre une vue agrégée de l'unité en représentant sa configuration par rapport à son environnement. La généralité du modèle de référence montre que les trois unités de stockage possèdent un modèle dont la structure et les règles d'évolution sont identiques, par conséquent

---

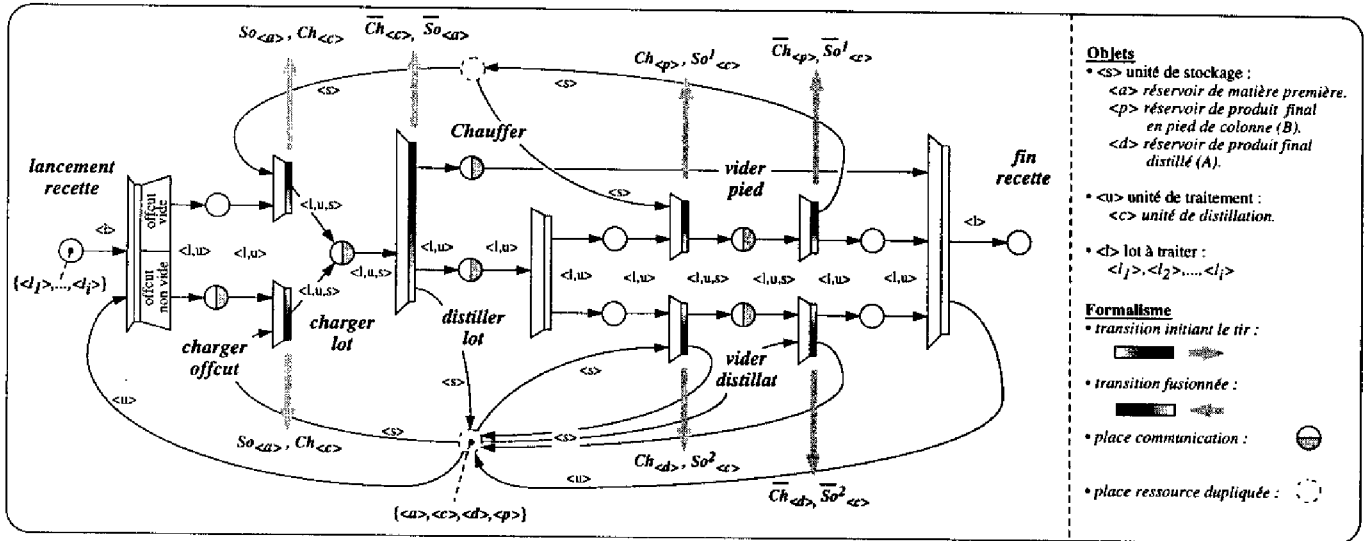
3. à titre d'illustration.



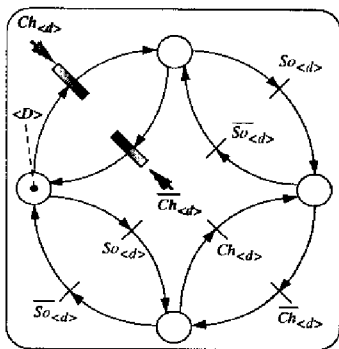
7.1.1: Le stock de matière première



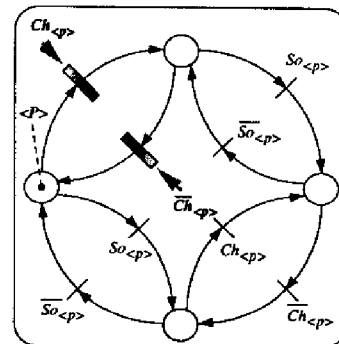
7.1.2: L'unité de distillation



7.1.3: La recette



7.1.4: Le stock de produit final A



7.1.5: Le stock de produit final B

FIG. 7.1 – Recette et modèles de référence : leurs interactions

un repliage de ces réseaux est possible (Figures 7.1.1, 7.1.4, 7.1.5). Nous conserverons cependant trois modèles distincts afin d'illustrer plus clairement les interactions entre les modèles de référence et la recette. Les différences entre ces modèles reposent sur les objets manipulés, et les labels des interactions avec la recette.

Rappelons que le jeton de chaque modèle de référence est un objet contenant un ensemble d'informations sur l'état et les caractéristiques de l'unité qu'il représente (cf. chapitre 6 section 3.1). Par exemple, le jeton représentant l'unité d'alimentation est l'objet  $\langle A \rangle$  (Figure 7.1.1). Afin de rendre plus claire la correspondance des réseaux et de leurs interactions, nous avons associé le même nom aux objets représentant une même unité:  $\langle A \rangle$  est l'image de l'unité d'alimentation au sein du modèle de référence et  $\langle a \rangle$  est l'image de l'unité en tant que ressource sur le réseau de la recette. Les labels associés aux interactions indiquent la nature de l'interaction et le nom de l'unité concernée: par exemple,  $So_{\langle a \rangle}$  et  $Ch_{\langle a \rangle}$  correspondent respectivement à une requête de soutirage et de remplissage sur l'unité d'alimentation (ressource  $\langle a \rangle$ ),  $\overline{So}_{\langle a \rangle}$  et  $\overline{Ch}_{\langle a \rangle}$  correspondant respectivement à une fin de soutirage et de remplissage.

Le modèle de référence de l'unité de distillation possède un ensemble de règles d'évolution plus complexes que celles des modèles de référence des autres unités (Figure 7.1.2). Ces règles sont issues du nombre d'activités possibles sur l'unité qui comprend un flux entrant et deux flux sortants: trois activités sont donc possibles, une de remplissage ( $Ch_{\langle c \rangle}$ ) et deux de soutirage ( $So^1_{\langle c \rangle}$  et  $So^2_{\langle c \rangle}$ ). Le nombre d'états discrets correspondant aux configurations n'est pas dépendant du nombre de flux entrants/sortants. Considérons par exemple ce cas de l'unité de distillation: que l'un des deux, ou les deux, flux sortants soient actifs simultanément, l'unité se trouve dans la configuration "unité en soutirage". La distinction entre l'origine des flux sortants actifs est effectuée par l'intermédiaire de l'objet représentant l'unité ( $\langle C \rangle$ ), au sein duquel l'état de chaque flux est mémorisé.

La logique de changement d'état du modèle de référence est alors décrite par des équations impliquant des attributs de l'objet qui évolue dans le réseau. Considérons par exemple l'équation (4) sur la figure 7.1.2. Cette équation comprend des termes impliquant des informations portées par l'objet  $\langle C \rangle$ , elle décrit la logique de maintien dans la configuration courante lors de la réception d'une requête de soutirage ou d'une fin d'activité de soutirage en cours. Décrivons cette équation, qui n'est valable que dans les configurations "unité en soutirage" et "unité en soutirage/remplissage":

- si une requête de soutirage  $So^1_{\langle c \rangle}$  ou  $So^2_{\langle c \rangle}$  est reçue, elle n'est autorisée que si cette activité n'est pas déjà en cours:

$$(So^1_{\langle c \rangle} \wedge C.SO1=\text{faux}) \vee (So^2_{\langle c \rangle} \wedge C.SO2=\text{faux})$$

Si cette requête est autorisée, l'unité reste dans la même configuration puisque l'un des soutirages est déjà actif (l'état discret correspondant étant déjà marqué).



- si la fin d'une activité de soutirage est indiquée, elle ne provoquera un changement de configuration que dans la mesure où plus aucune activité de soutirage n'est active. Dans le cas contraire l'unité reste dans la même configuration :

$$(\overline{So}_{<c>}^1 \wedge C.SO2=vrai) \vee (\overline{So}_{<c>}^2 \wedge C.SO1=vrai)$$

Cette logique est valable quel que soit le nombre de flux de l'unité.

### 1.3 Interactions entre la recette et les modèles de référence

Dans cette approche, les états de chaque unité sont représentés par les modèles de référence, qui correspondent à des automates finis, et c'est le réseau de Petri décrivant la recette qui va coordonner les évolutions de ces automates. Le parallélisme explicite n'existe donc qu'au niveau d'une recette et de la gestion temps-réel d'une ressource entre les recettes.

Les interactions entre la recette et les modèles de référence des unités de stockage suivent toutes le principe évoqué aux chapitres précédents.

Prenons par exemple le lancement de l'opération "Charger lot" (Figure 7.1.3). Lors du franchissement de l'une ou l'autre des transitions amont de la place activité correspondante, la recette émet deux requêtes (flèche grise) : une requête de soutirage à destination du modèle de référence de l'unité d'alimentation ( $So_{<a>}$ ), et une requête de remplissage à destination du modèle de référence de l'unité de distillation ( $Ch_{<c>}$ ). Ces deux requêtes sont alors examinées respectivement par chacun des modèles de référence.

Le soutirage de l'unité de distillation n'est autorisé que dans la mesure où le changement de configuration induit par la requête est possible (existence d'une transition franchissable amenant l'unité dans cette configuration) et les paramètres de l'opération sont acceptables : le débit de transfert requis est inférieur au débit maximum de l'unité, la rétention de matière dans l'unité est supérieure à la taille du lot. Si ces conditions sont remplies, la fenêtre temporelle de fin de soutirage est calculée.

La faisabilité de l'opération est garantie en ce qui concerne cette unité.

Par contre, l'examen de la requête par le modèle de référence de l'unité de distillation est plus délicat. Ce modèle est en fait une représentation agrégée de l'unité de distillation qui comprend quatre appareils : la colonne, le condenseur, le réservoir de coupe, et le réservoir intégrateur. Par définition, ce modèle agrégé ne représente pas les états continus et discrets des appareils constituant l'unité. Seule sa rétention massique globale est connue, ainsi que sa configuration par rapport à son environnement.

Or si la requête de chargement du lot est destinée à l'unité de distillation, l'appareil impliqué dans cette opération est en fait la colonne. A ce niveau d'agrégation, le modèle de référence de l'unité de distillation peut autoriser le chargement du lot en s'assurant que

les paramètres de l'opération sont acceptables et que le passage dans une configuration de remplissage est possible, mais il ne peut en aucun cas garantir la faisabilité de l'opération sur la colonne. En effet, ce modèle n'est pas en mesure de répondre aux interrogations concernant l'état de la colonne: "la rétention massique de la colonne permet-elle le chargement du lot?" - la répartition de la rétention massique de l'unité sur ses appareils n'est pas connue, "la colonne est-elle dans une configuration permettant cette opération?" - la seule information connue est qu'elle n'est pas en remplissage puisque l'unité elle-même ne l'est pas, cependant une distillation peut être en cours. Autant d'interrogations dont les réponses ne sont disponibles qu'au niveau inférieur.

La requête est autorisée sur l'unité de distillation puisque son état (discret et continu) le permet, mais sa faisabilité n'est pas garantie et la requête peut par conséquent être rejetée au niveau inférieur. Si un rejet se produit, il s'exprime par le refus de franchissement d'une transition au niveau inférieur. Si ce rejet n'est pas synchrone avec la requête issue de la recette, il a toutefois lieu immédiatement après l'émission de la requête. Il n'est pas nécessaire d'attendre la date au plus tard de fin d'exécution de l'opération pour se rendre compte qu'elle ne peut avoir lieu dans la configuration courante et dans le délai prévu.

*Il n'y a pas de garantie de cohérence parfaite entre les niveaux.* Seule la simulation permet l'obtention d'une réponse à priori, c'est-à-dire avant la propagation de la requête vers le niveau inférieur.

Une incohérence de cette nature ne peut provenir que d'une mauvaise spécification de la recette, un enchaînement de deux chargements de lots par exemple.

Il ne peut pas s'agir de la conséquence d'un plan de fabrication incorrect, un plan prévoyant la distillation de plusieurs lots simultanément sur l'unité par exemple. L'unité étant réservée pour le traitement d'un lot exclusivement, un plan prévoyant le traitement de deux lots sur l'unité serait qualifié d'infaisable lors de l'exécution de la recette (ressource non disponible dans l'intervalle de temps prévu par l'ordonnancement). L'infaisabilité d'un plan de fabrication doit être résolue au niveau de la gestion prévisionnelle, par une nouvelle génération du plan ou un réajustement de ce dernier s'il s'agit d'un simple décalage temporel, et non au niveau de la supervision lors du pilotage en temps-réel.

Revenons à l'exécution de l'opération "*Charger lot*". Suite à l'autorisation de cette opération de transfert, il y a changement de configuration sur chacun des modèles de référence. Les fenêtres temporelles respectives de fin d'activité de soutirage et de remplissage sont calculées, et comparées à celle issue de l'ordonnancement portée sur la recette. La requête est alors propagée vers le niveau inférieur.

La configuration du générateur d'événements nécessaire à la détection de la fin de cette opération de transfert ne peut pas être effectuée au niveau de la recette. La définition de l'événement, avant la configuration du générateur, requiert deux éléments qui sont le seuil et la variable d'état sur laquelle le détecter. Or cette variable d'état est la variable de rétention massique de la colonne dont l'identificateur n'est pas connu à ce niveau. C'est par conséquent au niveau inférieur que s'effectue la configuration du générateur, ainsi que

la réception de l'événement d'état correspondant.

La fin de l'opération de chargement du lot dans l'unité de distillation est alors indiquée à la recette par l'intermédiaire du compte-rendu d'exécution retourné par le niveau inférieur.

La section ultérieure concernant la description de l'interaction entre les niveaux nous permettra de revenir sur ce fonctionnement.

Dès la fin de l'opération, la recette signale la fin de soutirage au modèle de référence de l'unité d'alimentation, et la fin de remplissage à celui de l'unité de distillation (flèche grise sortant de la transition de fin d'opération "*Charger lot*" portant les interactions  $\overline{So}_{\langle a \rangle}$  et  $\overline{Ch}_{\langle c \rangle}$ ).

Intéressons-nous maintenant à la modélisation des fonctions et des modèles de référence du niveau inférieur, le niveau de coordination.

## 2 La modélisation au niveau coordination

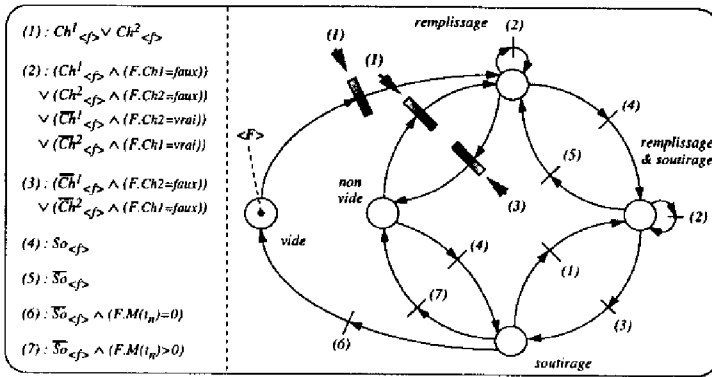
### 2.1 Les fonctions et les ressources

Ces modèles sont issus de la désagrégation des places activités de la recette. Les ressources présentées à ce niveau sont les appareils dont l'association constitue l'unité de distillation. Les fonctions sont représentées sur la figure 7.2.3. L'ensemble des places partagées entre les modèles des fonctions représente la stratégie d'utilisation de ces ressources (ressources partagées de manière exclusive entre les fonctions). Par exemple, toutes les fonctions s'exécutant sur la colonne, "*Charger lot*", "*Charger offcut*", "*Distiller lot*" et "*Vider pied de colonne*", sont mutuellement exclusives.

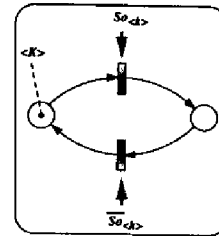
Deux types de jetons sont utilisés, d'une part les instances d'objets de la classe *Requête*  $\langle r \rangle$  transmises par le niveau supérieur pour l'activation de fonctions, et retournées au niveau supérieur lors des comptes-rendus d'exécution, et d'autre part les jetons représentant les appareils.

Le modèle de la fonction "*Distiller lot*" représente l'enchaînement séquentiel des différentes stratégies de conduite de l'unité de distillation (Figure 7.2.3). Il établit dans un premier temps la configuration nécessaire des appareils, puis active un ensemble de boucles de régulation au sein de cette configuration (changement de loi de commande), en précisant les consignes, comme le taux de reflux par exemple. Le compte-rendu retourné indique la prise en compte du changement de loi de commande mais pas un compte-rendu d'exécution. Il s'agit d'une boucle de régulation, donc la loi de commande sélectionnée reste active tant qu'aucune requête de changement de loi ou d'arrêt de la régulation n'est émise.

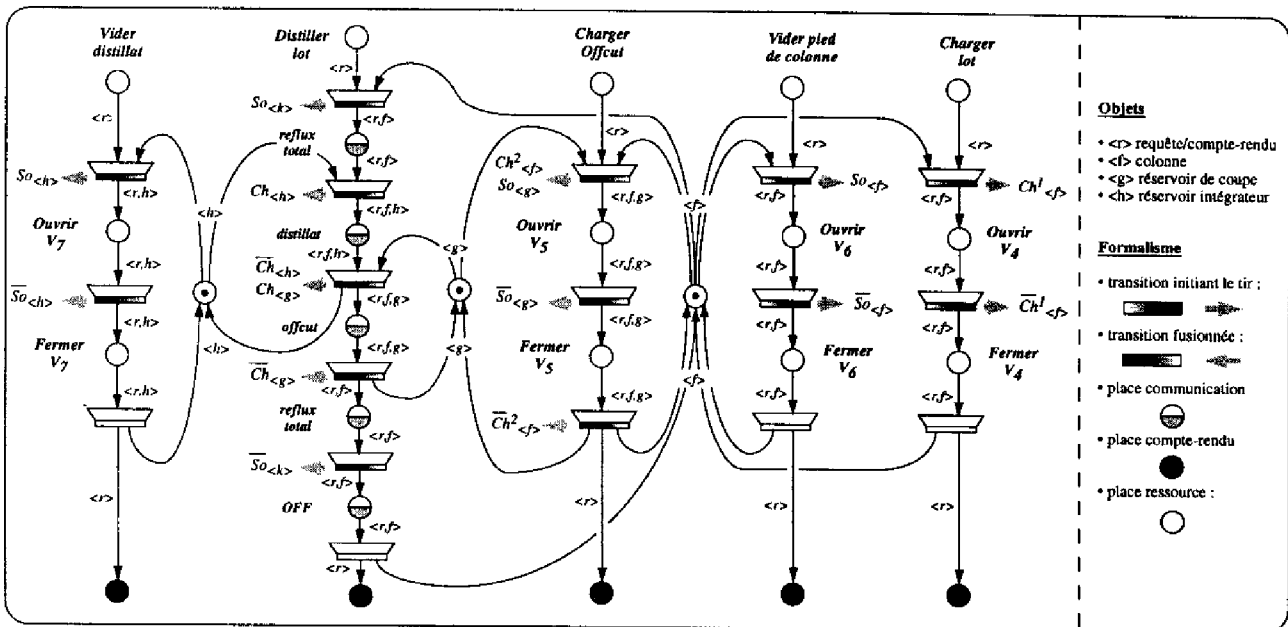
Chaque changement de stratégie correspond au franchissement d'une transition sur le réseau de Petri modélisant la fonction "*Distiller lot*". Un changement de stratégie



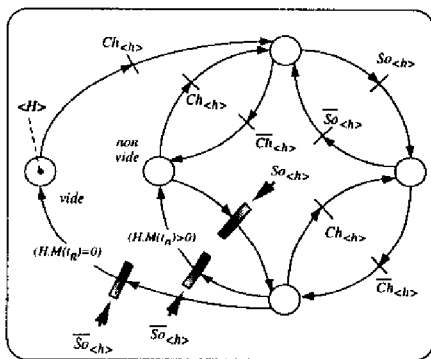
7.2.1: La colonne à distiller



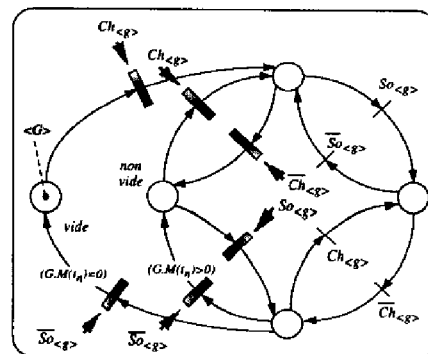
7.2.2: Le condenseur



7.2.3: Les fonctions



7.2.4: Le réservoir intégrateur



7.2.5: Le réservoir de coupe

FIG. 7.2 – Fonctions et modèles de référence : leurs interactions

s'effectue à un instant de temps particulier déterminé en fonction de la concentration du produit, dont le seuil est spécifié à travers la requête. La condition associée à chaque transition porte donc sur la concentration.

Cependant, la concentration n'est pas une mesure directe, elle est obtenue par corrélation avec la température, qui est mesurée en tête de colonne. Un modèle continu (équation) est nécessaire pour décrire cette corrélation. Il s'agit d'une fonction définie par intervalles, qui a comme paramètres une valeur seuil correspondant à la température de bulle de la solution et le taux de reflux, considéré constant sur un intervalle, et comme variable d'entrée, la mesure de température.

De manière identique au modèle de référence au sein duquel le réseau de Petri pilote un système d'équations, le modèle de la commande doit piloter cette équation. Cette dernière est donc associée à une place du réseau de Petri représentant la fonction<sup>4</sup> "Distiller lot" (Figure 7.3).

Dès le lancement de la distillation, l'équation est activée. Cette activation s'effectue par le marquage de la place à laquelle est associée l'équation. Le générateur d'événements est configuré selon une règle de génération périodique afin d'actualiser régulièrement la valeur de la variable de température. Lors d'un changement de stratégie, un nouveau taux de reflux est fixé. Changer de taux de reflux implique de mettre à jour le paramètre de l'équation, et d'indiquer la nouvelle consigne de la boucle de reflux au régulateur concerné : ceci est effectué lors de chaque changement de stratégie.

Une fois la distillation terminée, l'équation ne doit plus être active. La place à laquelle est associée cette équation est donc démarquée dès le franchissement de la transition de fin de distillation. Ensuite, les régulateurs sont désactivés, et le générateur d'événements est reconfiguré afin d'annuler la génération périodique.

## 2.2 Les modèles de référence des appareils

Ces modèles sont issus de la désagrégation de l'unité de distillation en un ensemble d'appareils la constituant. Les quatre modèles de référence, un modèle pour chaque appareil, sont présentés sur la figure 7.2.

Rappelons que les modèles de référence correspondent à une représentation agrégée de l'état de l'appareil. La représentation de l'évolution interne de son état, qui pourrait être intéressante pour l'étude du transitoire de démarrage par exemple (changements d'état sur les plateaux, passage du point de bulle au point de rosée), n'est pas traitée.

Seul le modèle de référence du condenseur est différent des modèles décrits jusque-là. Ce modèle ne comporte que deux états, les deux configurations possibles du condenseur, un seul flux avec son environnement étant commandable (son flux de sortie). Le condenseur

---

4. cette place n'existe pas sur la représentation initiale de la fonction décrite sur la figure 7.2.3.

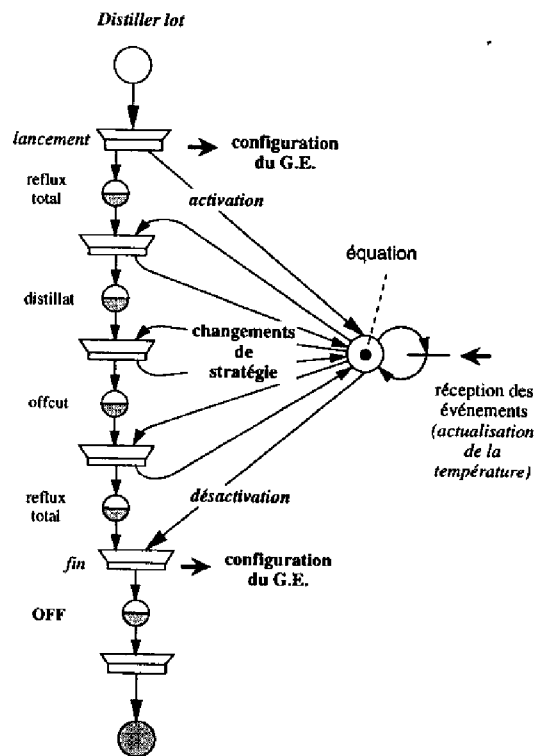


FIG. 7.3 – Gestion d'une équation au sein du modèle de la commande

est directement connecté à la colonne à distiller, il est donc possible de le considérer initialement dans la configuration “remplissage”. Cette configuration signifie, rappelons-le, que l'entité considérée (appareil ou unité) est connectée à une autre entité amont. La seconde configuration du condenseur est par conséquent “remplissage et soutirage simultanés”. Le passage de l'une à l'autre dépend de l'état des vannes  $V_1$ ,  $V_2$ , et  $V_3$  qui établissent respectivement une connexion aval avec la colonne, le réservoir de coupe, et le réservoir intégrateur. La représentation du condenseur est nécessaire pour l'expression du bilan matière au sein de l'unité de distillation. Par ailleurs, l'expression de ses deux configurations peut s'avérer utile à des fins de surveillance, pour distinguer les tendances d'évolution de sa rétention massique.

### 2.3 Interactions entre les fonctions et les modèles de référence

Les interactions représentées à ce niveau sont celles entre les fonctions et les appareils constituant l'unité.

Considérons l'interaction entre la fonction “*Vider distillat*” (Figure 7.2.3) et le modèle de référence du réservoir intégrateur (Figure 7.2.4).

Il est possible de constater qu'il y a un décalage entre l'instant de détection de fin de phase et l'instant de fin de phase effectif. Le premier instant correspond à l'occurrence de l'événement d'état envoyé par le générateur d'événements suite à la configuration effectuée lors du franchissement de la transition amont de la place “Ouvrir  $V_7$ ”. Cet instant se traduit par le franchissement de la transition de sortie de la place “Ouvrir  $V_7$ ”. Le second correspond à la fermeture de la vanne, il est traduit par le franchissement de la transition de sortie de place “Fermer  $V_7$ ”.

Il y a par conséquent un écart entre la valeur de la rétention massique du réservoir intégrateur sur le modèle de référence, mise à jour dès la réception de l'événement d'état, et la valeur de cette rétention massique à l'instant où la vanne  $V_7$  est effectivement fermée.

Même si le principe de la détection de déviation est basé en partie sur la comparaison de l'état continu calculé (simulé) et l'état continu effectif (mesuré), cet écart peut être négligé. D'une part, la grandeur mesurée n'est pas exacte, c'est une image d'une grandeur physique dont la précision dépend de celle du capteur, du bruit de mesure, etc... D'autre part, la grandeur calculée souffre également d'imprécision due à la fois au modèle et aux approximations de calculs lorsqu'ils sont complexes. C'est au seuil de comparaison entre ces grandeurs «d'absorber» cet écart.

## 3 Interactions entre les niveaux

L'interaction qui nous intéresse plus particulièrement est celle entre la recette et la fonction “*Charger lot*”. Elle nous permet d'illustrer la détection d'une fin de configuration (fin d'activité) au niveau coordination, qui est alors signalée à la recette par l'intermédiaire

du compte-rendu retourné. Il s'agit du fonctionnement que nous avons évoqué dans le cadre de l'interaction entre la recette et les modèles de référence au niveau supervision (cf. section 1.3).

L'objectif étant uniquement d'illustrer les interactions entre les deux niveaux, nous nous plaçons dans le cadre d'un fonctionnement normal. Par conséquent, nous considérons que la requête émise est cohérente, c'est-à-dire compatible avec l'état des modèles de référence et de leurs contraintes.

Par ailleurs, dans un souci de clarté de la représentation, les réseaux ne sont que partiellement représentés, et les canaux de chaque modèle de référence sont dupliqués.

L'illustration est basée sur l'évolution du marquage respectif des réseaux impliqués (Figure 7.4). Nous avons pour cela numéroté le marquage global selon l'ordre chronologique de son évolution, en attribuant le même numéro à tous les marquages ayant lieu entre deux états stables (c'est-à-dire pendant un même cycle).

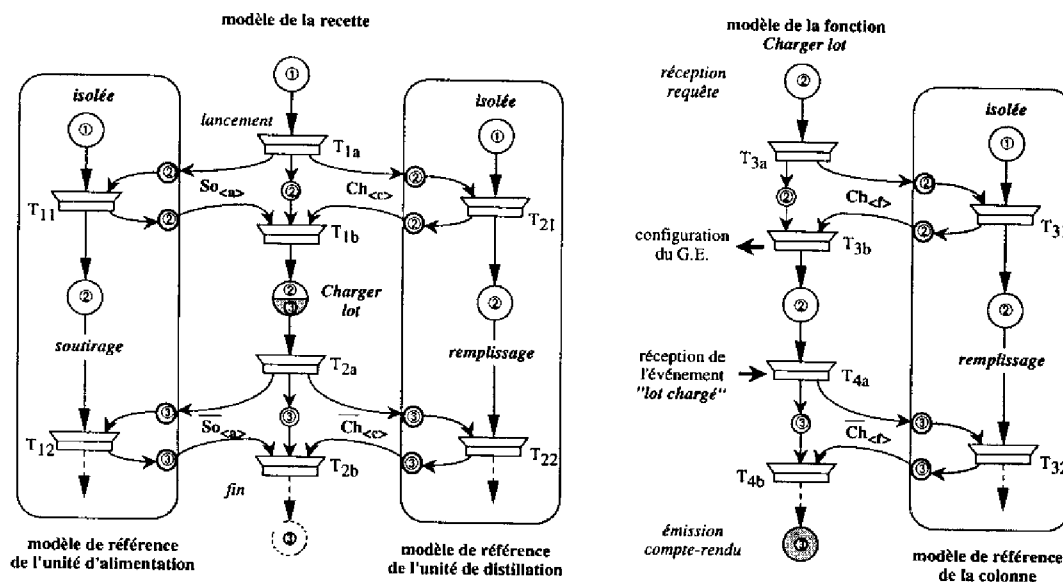


FIG. 7.4 – Interactions entre les niveaux

Le marquage ① correspond à l'état des modèles à l'instant de temps initial considéré.

Le lancement de l'opération "Charger lot", traduit par le franchissement simultané des transitions  $T_1$ ,  $T_{11}$ , et  $T_{21}$  au niveau supervision, engendre dans le même cycle le franchissement des transitions  $T_3$  et  $T_{31}$  au niveau coordination<sup>5</sup>. Le franchissement de ces dernières fait suite à la réception de la requête du niveau supérieur. Ces tirs de transition successifs amènent les modèles dans l'état stable ②. La phase active de la recette est alors

5. les transitions  $T_1$  et  $T_3$  ont été décomposées pour expliciter les interactions (fusion de transitions).



celle de chargement du lot dans l'unité de distillation, et la fonction "Charger lot" est en attente de l'occurrence de l'événement de fin d'activité correspondant à "lot chargé" dans la colonne.

La réception de cet événement provoque le franchissement simultané des transitions  $T_4$  et  $T_{32}$  (actualisation des états continu et discret du modèle de référence de la colonne), dont résulte l'émission d'un compte-rendu d'exécution à destination de la recette.

La réception de ce compte-rendu sur la recette, induit le franchissement simultané des transitions  $T_2$ ,  $T_{12}$  et  $T_{22}$  (mise à jour des états continu et discret des modèles de référence de l'unité d'alimentation et de l'unité de distillation). Le nouvel état stable atteint est l'état ③.

## 4 Conclusion

A travers cet exemple, nous avons mis en évidence la modularité et la généralité des modèles de référence. Les interactions entre le modèle de la commande et les modèles de références ont été détaillées au sein de chaque niveau, ainsi qu'entre les deux niveaux hiérarchiques. L'agrégation dans la représentation à la fois du procédé et de la commande a été illustrée, elle clarifie le rôle de chaque niveau. En effet, il apparaît clairement que la configuration des unités est gérée au niveau supervision, et que la configuration au sein d'une unité l'est au niveau inférieur de coordination.

L'utilisation des réseaux de Petri pour modéliser les différents niveaux facilite la communication entre les ingénieurs chargés de la conception des différents niveaux (outil graphique). Elle facilite en fait la détection d'incohérence entre ces niveaux mais cependant elle n'en garantit pas une cohérence parfaite. Comme dans tous les cas pour lesquels il y a une interaction entre des niveaux agrégés et désagrégés, une politique cohérente au niveau agrégé ne l'est plus forcément au niveau désagrégé.

Par ailleurs, cet exemple nous a également permis d'illustrer l'association d'un modèle continu à un état discret du modèle de la commande. L'activation, l'actualisation, et la désactivation de ce système d'équations sont pilotées par le modèle à événements discrets, dont l'évolution est contrôlée par le résultat des calculs menés dans la partie continue (calcul de la concentration).

De manière générale, notre structuration du modèle hybride pour la conduite du procédé batch exprime clairement les interactions entre les parties continues et discrètes.

Pour compléter cet exemple d'application, nous allons maintenant étudier une opération unitaire portant sur la conduite du système de chauffe/refroidissement d'un réacteur exothermique.

## Chapitre 8

# Commande et supervision de la conduite thermique d'un réacteur exothermique multifluide

Le procédé considéré à travers cet exemple d'application est le réacteur batch exothermique présenté au premier chapitre (cf. section 2.2.2).

Nous allons nous intéresser à l'exécution d'une opération de transformation d'un lot au sein du réacteur (supposé chargé). La réalisation de cette opération se ramène à un problème de conduite thermique (cf. chapitre 4 section 1.1.3).

La température du milieu réactionnel est contrôlée par l'utilisation de plusieurs fluides, caloporteurs (vapeur) ou réfrigérants (eau glycolée), à température fixée et d'un fluide intermédiaire à température réglable (vapeur injectée dans l'eau du réseau). La conduite thermique nécessite donc la gestion dynamique des différentes configurations issues du changement de fluide, la gestion des transitoires induits par ces commutations, ainsi que la détermination de la commande à appliquer dans la configuration active.

L'objectif de ce chapitre est d'illustrer le rôle de «supervision locale» joué par la coordination, ainsi que les interactions entre les parties discrète et continue du modèle de la commande. Nous n'illustrerons pas les modèles de référence de l'unité de réaction et leurs interactions avec le modèle de la commande, le chapitre précédent ayant été consacré à cet aspect.

## 1 La modélisation

### 1.1 La recette de l'opération

Le traitement de la matière au sein du réacteur est initiée par la présence d'un catalyseur puis la réaction est maintenue selon des profils de température et de coulée donnés (Figure 8.1). La transformation est donc constituée de la régulation de la température

pour suivre le profil imposé, de l'agitation du contenu pour avoir un mélange homogène et de l'ajout de réactifs, avec une quantité et un taux adéquats, à des instants de temps précis afin de maintenir un taux de réaction. Il s'agit d'un ensemble de tâches qui doivent être menées simultanément.

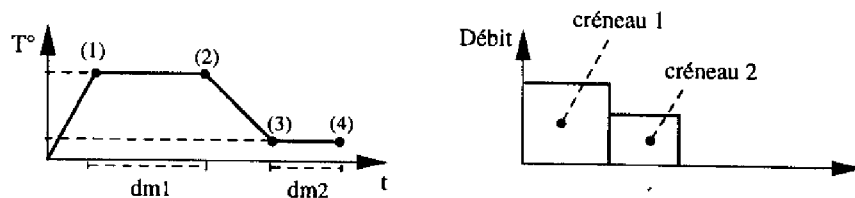


FIG. 8.1 - Profils de température et de coulée

La coordination, chargée de l'exécution et de la synchronisation de ces tâches, peut être perçue comme une «supervision locale» [Andreu *et al.* c]. Par exemple dans le cadre du suivi du profil de température, elle assure la gestion des configurations au sein de l'unité de réaction (unité de traitement) et elle décide des instants de commutation.

A partir de ces profils de température et de coulée, il est possible de définir la procédure de la recette décrivant la transformation du lot.

La modélisation de cette procédure, représentée sur la figure 8.2, est basée sur la décomposition des profils en plusieurs phases. Le profil de température peut être découpé en quatre phases (Figure 8.1) : une phase de chauffe jusqu'au point 1, une phase de maintien jusqu'au point 2, une phase de refroidissement jusqu'au point 3, et enfin une phase de maintien jusqu'au point 4. Le profil de coulée est également séparable en deux phases (Figure 8.1), qui correspondent à deux créniaux.

Une place étant associée à chacune de ces phases nous obtenons les séquences représentées sur la Figure 8.2. La phase active de la procédure correspond aux places marquées des deux séquences en parallèle. L'évolution du marquage est engendrée par :

- des *événements d'état* comme par exemple le franchissement des transitions  $t_2$  et  $t_4$  par rapport à des seuils de température (points 1 et 3 sur la Figure 8.1),
- des *événements de temps* telles les dates de franchissement des transitions  $t_3$  et  $t_5$  obtenues à partir des durées, fixées ou déterminées à l'aide d'un modèle physico-chimique, des phases de maintien (durées de maintien  $dm_1$  et  $dm_2$  sur la Figure 8.1).

Cette représentation fait clairement apparaître le parallélisme et la synchronisation dans l'exécution de l'opération. A chaque changement de phase, qu'il s'agisse de la température ou de la coulée, correspond un changement de consigne. Ces changements de

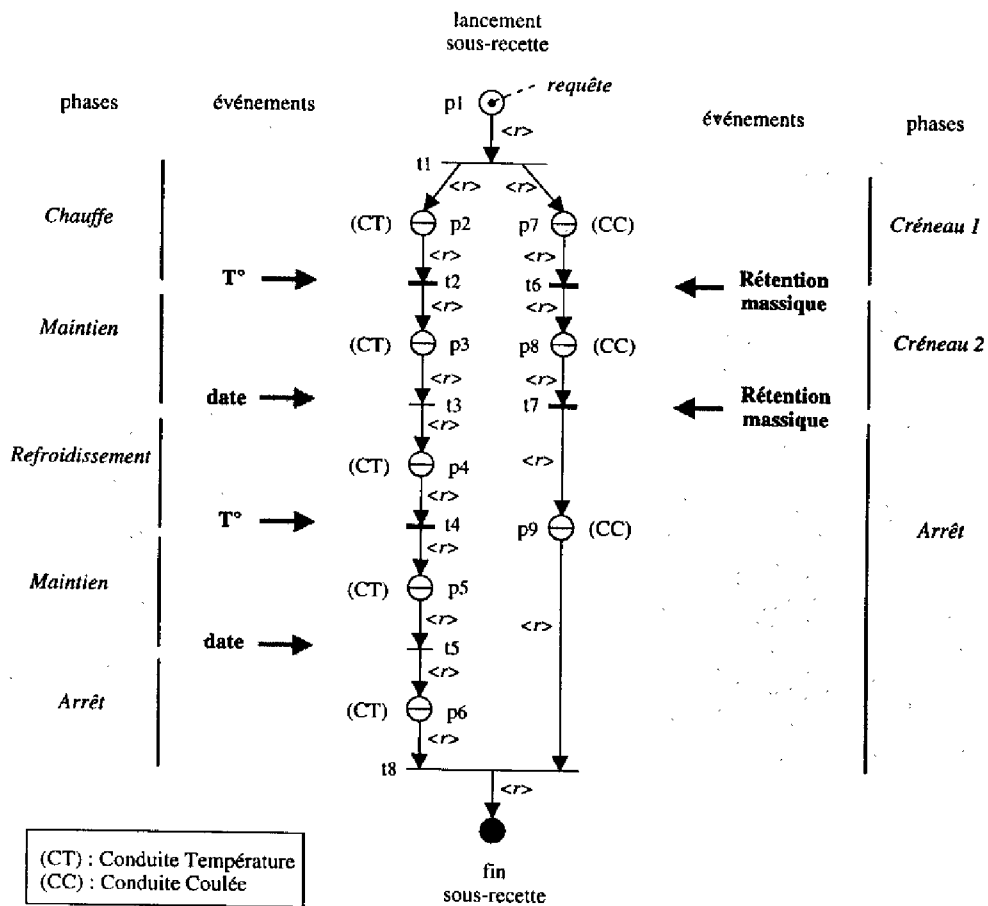


FIG. 8.2 – Modélisation de la procédure de la recette

phase sont mis en œuvre par l'intermédiaire des fonctions utilisées.

## 1.2 Les fonctions

Un ensemble de fonctions peut être construit de manière modulaire (Figure 8.4). Cet ensemble de fonctions du niveau coordination se constitue :

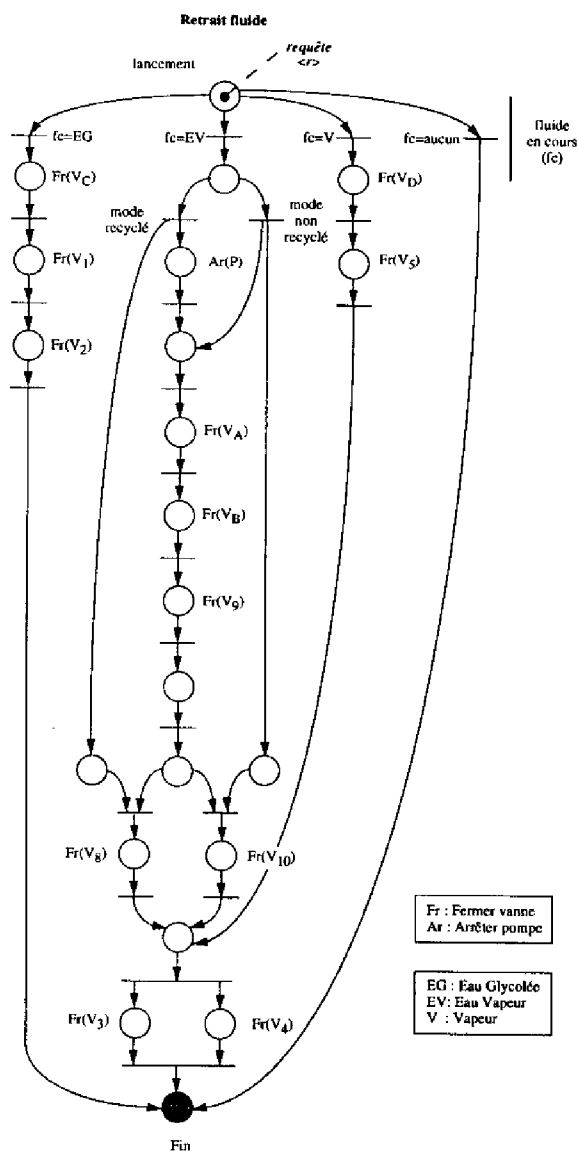
- de *fonctions de configuration* qui sont indépendantes de la stratégie de pilotage de la production, comme la fonction de purge (coordination d'un ensemble de vannes discrètes) et les fonctions d'application ou de retrait d'un fluide (respectivement ouverture et fermeture d'un ensemble de vannes discrètes et continues selon une séquence particulière).
- de *fonctions de conduite* qui utilisent les fonctions de configuration comme par exemple la fonction "*Conduite thermique*", la fonction "*Conduite coulée*" et la fonction "*Commutation*"; cette dernière utilise la fonction de purge et celles d'application et de retrait d'un fluide. Ces fonctions de conduite intègrent la stratégie de pilotage.

### 1.2.1 Les fonctions d'application et de retrait de fluide

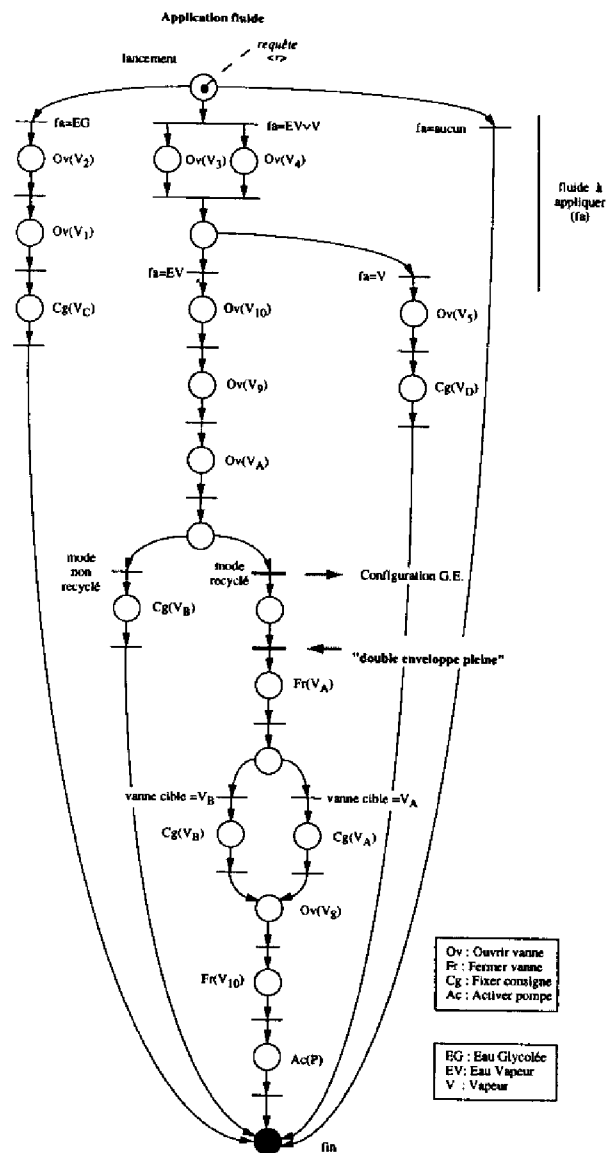
Ces fonctions ne font que mettre en œuvre les requêtes issues de la fonction de commutation. Une requête de ce type désigne le fluide à appliquer ou à retirer, le mode de fonctionnement dans la configuration choisie (mode recyclé ou non), ainsi que les consignes des vannes (dites vannes cibles) dont le débit doit être régulé.

Considérons par exemple le retrait et l'application du fluide intermédiaire Eau/Vapeur.

Dans le cas du retrait du fluide Eau/Vapeur, qui correspond donc au fluide en cours à l'instant de temps considéré, le début de la séquence de fermeture dépend du mode de fonctionnement (Figure 8.3.1). Dans un mode recyclé, la pompe étant active dans la boucle du système de chauffage/refroidissement, elle doit être arrêtée avant d'enchaîner la fermeture des vannes impliquées dans la configuration courante (celle correspondant à l'utilisation de l'Eau/Vapeur). Dans le cas d'un fonctionnement en mode non recyclé cette étape n'est pas nécessaire et le retrait du fluide commence directement par la fermeture des vannes. L'ordre de fermeture est déterminé de manière précise, il peut dépendre de contraintes temporelles strictes (non représentées sur la figure) telle que "fermer la vanne  $V_A$  cinq secondes après avoir arrêté la pompe" par exemple. Cet ordre de fermeture dont la représentation séquentielle de la figure 8.3.1 traduit les contraintes de précédence, est : couper l'alimentation en eau du réseau (Fermer  $V_A$ ), couper l'ajout de vapeur (Fermer  $V_B$ ), fermer la vanne de sortie du mélangeur statique (Fermer  $V_9$ ), puis selon le mode de fonctionnement fermer la vanne  $V_8$  établissant la boucle de recyclage ou fermer la vanne  $V_{10}$  qui établit une connexion avec l'égout (configuration sans recyclage). La dernière étape



8.3.1: La fonction "Retrait fluide"



8.3.2: La fonction "Application fluide"

FIG. 8.3 – Exemples de fonctions de configuration

de retrait du fluide Eau/Vapeur consiste à fermer les vannes d'accès à la double enveloppe du réacteur (vannes  $V_3$  et  $V_4$ ).

Une fois la séquence terminée, la fonction "*Retrait fluide*" retourne un compte-rendu d'exécution à destination de la fonction appelante (celle ayant émis la requête) qui est en l'occurrence la fonction de commutation.

Afin de faciliter la gestion des configurations, toutes les vannes impliquées dans la configuration à quitter sont systématiquement fermées, même si certaines d'entre elles doivent être rouvertes dans la configuration nouvelle (c'est-à-dire après le changement de fluide).

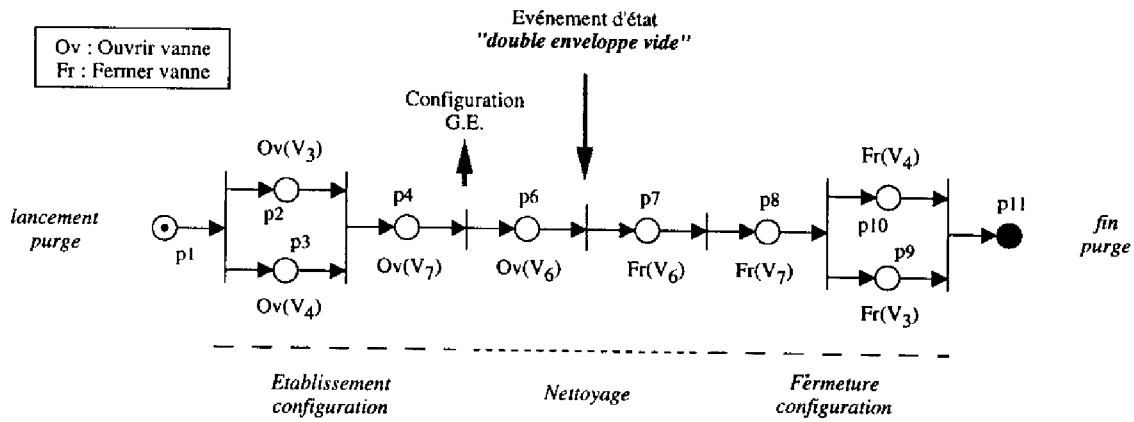
L'application du fluide intermédiaire Eau/Vapeur s'effectue quasiment dans l'ordre inverse de celui que nous venons de décrire (Figure 8.3.2). L'établissement de la configuration du système de chauffe/refroidissement est composée principalement de trois étapes. Tout d'abord l'accès à la double enveloppe est établi (ouverture des vannes  $V_3$  et  $V_4$ ). Ensuite la configuration est ouverte d'une part vers l'égout (ouverture de la vanne  $V_{10}$ ) et d'autre part vers le mélangeur statique à travers lequel est appliqué le fluide Eau/Vapeur (ouverture de la vanne  $V_9$ ).

La troisième étape, qui consiste à achever l'établissement de la configuration, dépend du mode de fonctionnement choisi. La vanne  $V_A$  contrôlant le débit de l'eau du réseau est entièrement ouverte. En mode non recyclé la configuration est terminée et l'action permettant d'atteindre le flux thermique requis porte sur le débit de la vanne de vapeur  $V_B$ . En mode recyclé, il est nécessaire d'attendre que la double enveloppe soit pleine avant d'agir soit sur le débit de l'eau soit sur celui de la vapeur pour atteindre le flux thermique requis. Ensuite, la boucle de recyclage est refermée (fermeture de la vanne  $V_{10}$  et ouverture de la vanne  $V_8$ ) et la pompe est activée.

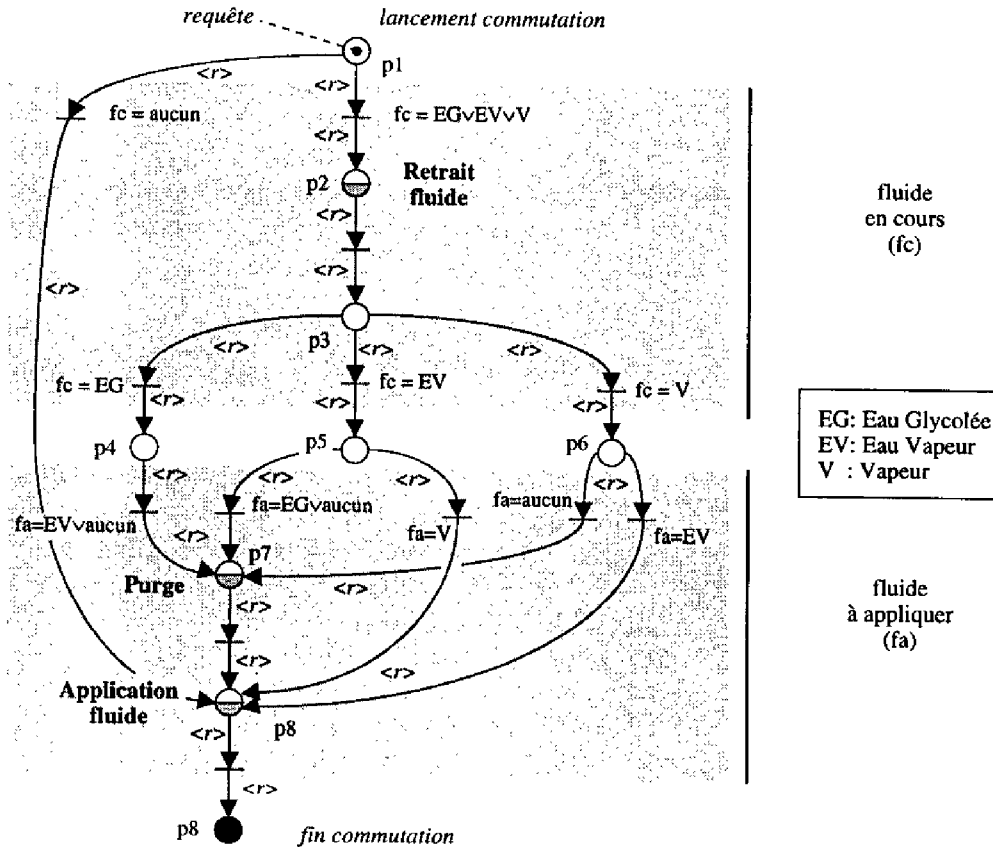
La fin de cette séquence se conclut par l'envoi d'un compte-rendu d'exécution à destination de la fonction de commutation.

### 1.2.2 La fonction de purge

La fonction "*Purge*" est représentée sur le réseau de Petri de la figure 8.4.1. La purge, lancée dès que la place d'entrée  $p_1$  est marquée, débute par l'ouverture des vannes  $V_3$  et  $V_4$  (passage du jeton de la place  $p_1$  aux places  $p_2$  et  $p_3$ ) afin d'établir une connexion entre la source d'air comprimé, la double enveloppe et la sortie vers l'égout. La séquence se poursuit par l'ouverture de la vanne  $V_7$  (passage du jeton des places  $p_2$  et  $p_3$  à la place  $p_4$ ) ouvrant l'accès à l'égout. La configuration nécessaire à la purge est établie. La configuration étant établie, la vanne d'application de l'air comprimé, la vanne  $V_6$ , peut être ouverte (passage du jeton de la place  $p_4$  à la place  $p_6$ ). L'air comprimé est envoyé dans la double enveloppe du réacteur tant que l'événement d'état "*double enveloppe vide*" n'a pas été détecté. Suite à l'occurrence de cet événement, l'ensemble des vannes impliquées dans cette configuration sont fermées, dans un ordre inverse à celui de l'établissement de la configuration.



8.4.1: (a) La fonction "Purge"



8.4.2: (b) La fonction "Commutation"

FIG. 8.4 – Exemples de fonctions de configuration (a) et de conduite (b)



Tous les franchissements de transition de cette séquence sont indépendants de la stratégie de pilotage de l'unité de réaction.

### 1.2.3 La fonction de commutation

La fonction de commutation, représentée sur la figure 8.4.2, séquence :

- le retrait du fluide en cours : le marquage de la place  $p_2$  correspond à l'appel de la fonction "*Retrait fluide*". Toutefois si aucun fluide n'est appliqué (aucun fluide en cours), ce qui est le cas au démarrage, elle passe directement à l'application du fluide sélectionné.
- la purge de la double enveloppe du réacteur : selon le fluide actuel et celui sélectionné, la purge n'est pas forcément nécessaire. L'ensemble des transitions de sortie des places  $p_4$ ,  $p_5$  et  $p_6$  constitue les différentes combinaisons. Par exemple si le fluide en cours est l'eau glycolée, la purge est nécessaire avant d'appliquer un nouveau fluide. Ce n'est pas le cas pour passer de la vapeur à l'eau/vapeur (et vice-versa).
- et l'application du fluide sélectionné : le marquage de la place  $p_7$  engendre l'exécution de la fonction "*Application fluide*".

Cette fonction intègre la stratégie de pilotage : elle exprime par exemple les commutations autorisées entre les fluides. Il est possible de remarquer sur le réseau de Petri de la figure 8.4.2, qu'il est interdit de passer de la vapeur à l'eau glycolée, de même qu'il est interdit de passer de l'eau glycolée à la vapeur. Ces contraintes de pilotage sont issues notamment de considérations de sécurité de l'installation : il faut éviter le choc thermique que pourrait produire la commutation d'un fluide fortement réfrigérant à un fluide fortement calorifique.

Ces interdictions sont exprimées par la restriction de l'ensemble des transitions de sortie des places  $p_4$  et  $p_6$ . Ces places ne possèdent qu'une seule transition de sortie dont l'interprétation associée désigne le fluide dont l'application est autorisée. Par exemple, à partir de la place  $p_4$ , dont le marquage résulte de l'arrêt d'utilisation de l'eau glycolée, le seul fluide qui peut être appliqué est l'eau/vapeur. Si le fluide à appliquer est autre que l'eau/vapeur, la vapeur en l'occurrence, la transition de sortie de la place  $p_4$  n'est pas franchissable, une anomalie est alors détectée. L'anomalie qui serait détectée dans ce cas correspond à une erreur de commande.

Détaillons maintenant la fonction de conduite thermique afin d'illustrer les interactions entre les parties discrète et continue au sein du modèle de la commande.

### 1.2.4 La fonction de conduite thermique

La fonction de conduite thermique est chargée de déterminer le flux thermique à appliquer pour atteindre le point de température désiré (selon les capacités réfrigérantes ou

calorifiques des différents fluides<sup>1</sup>), de sélectionner un fluide<sup>2</sup> et enfin d'engager une commutation ou un ajustement du débit du fluide courant.

Cette fonction est représentée<sup>3</sup> sur la figure 8.5.

Le suivi du profil de température s'effectue à travers l'utilisation d'un ensemble de fluides dont le choix est déterminé par le flux énergétique à fournir  $Q$ . Un algorithme de commande avancée<sup>4</sup> calcule ce flux thermique à fournir au milieu réactionnel (exprimé par  $Q_r$ ) pour suivre le profil de température de consigne désiré. Ce flux est ensuite interprété à l'aide d'un modèle thermique du système de chauffage/refroidissement pour déterminer les flux thermiques limites (capacités thermiques minimale et maximale, de chauffage et/ou de refroidissement d'un fluide: les flux thermiques  $Q_{EG_{min}}$  et  $Q_{EG_{max}}$  pour l'Eau Glycolée par exemple) [Louleh 96].

Le système d'équation correspondant est associé à la place  $p_4$  (Figure 8.5). Simultanément à l'activation du système d'équations, le générateur d'événements est configuré lors du franchissement de la transition  $t_1$ , afin d'assurer une actualisation périodique des variables d'état impliquées dans les équations. Il s'agit des mesures de la température de masse (réacteur), ainsi que des températures d'entrée et de sortie de la double enveloppe. La réception de l'événement est associée à la transition  $t_4$ , son franchissement périodique correspond à l'actualisation des variables.

Dès l'arrêt de la conduite thermique du réacteur, le générateur d'événements est reconfiguré afin d'annuler la génération périodique. Cet arrêt, traduit par le franchissement de la transition  $t_3$ , induit également la désactivation du système d'équations.

Lors du changement de phase, les paramètres du système d'équations sont mis à jour (tir de la transition  $t_2$ ).

Les résultats des calculs sont impliqués dans les interprétations associées aux transitions de sortie de la place  $p_9$ . La sélection du fluide dépend alors du flux thermique requis et des flux thermiques limites de chaque fluide.

Plusieurs stratégies de sélection du fluide peuvent être exprimées, seules les interprétations associées aux transitions sont alors à modifier. Les tableaux 8.1.1 et 8.1.2 donnent deux exemples de stratégies, sachant que le « bloc de transitions » considéré dépend du fluide en cours (Figure 8.5): par exemple la transition  $t_{10}$  est sensibilisée si le fluide courant est l'Eau/Vapeur et l'ensemble des transitions considérées est formé de  $t_{10a}$ ,  $t_{10b}$  et  $t_{10c}$ .

Une fois la sélection effectuée, la valeur de la commande à appliquer sur la vanne réglant le débit du fluide, pour obtenir le flux thermique requis, est calculée à l'aide d'un modèle d'échange thermique du fluide. Le système d'équation correspondant est associé à la place  $p_{16}$ . La valeur de cette commande constitue la consigne appliquée au régulateur contrôlant

1. la période d'applicabilité d'un fluide dépend du flux thermique requis.

2. règle en cas de choix multiple: priorité au fluide courant.

3. le réseau de Petri global étant relativement complexe, il n'a été que partiellement représenté.

4. régulateur prédictif adaptatif [Lann *et al.* 95].

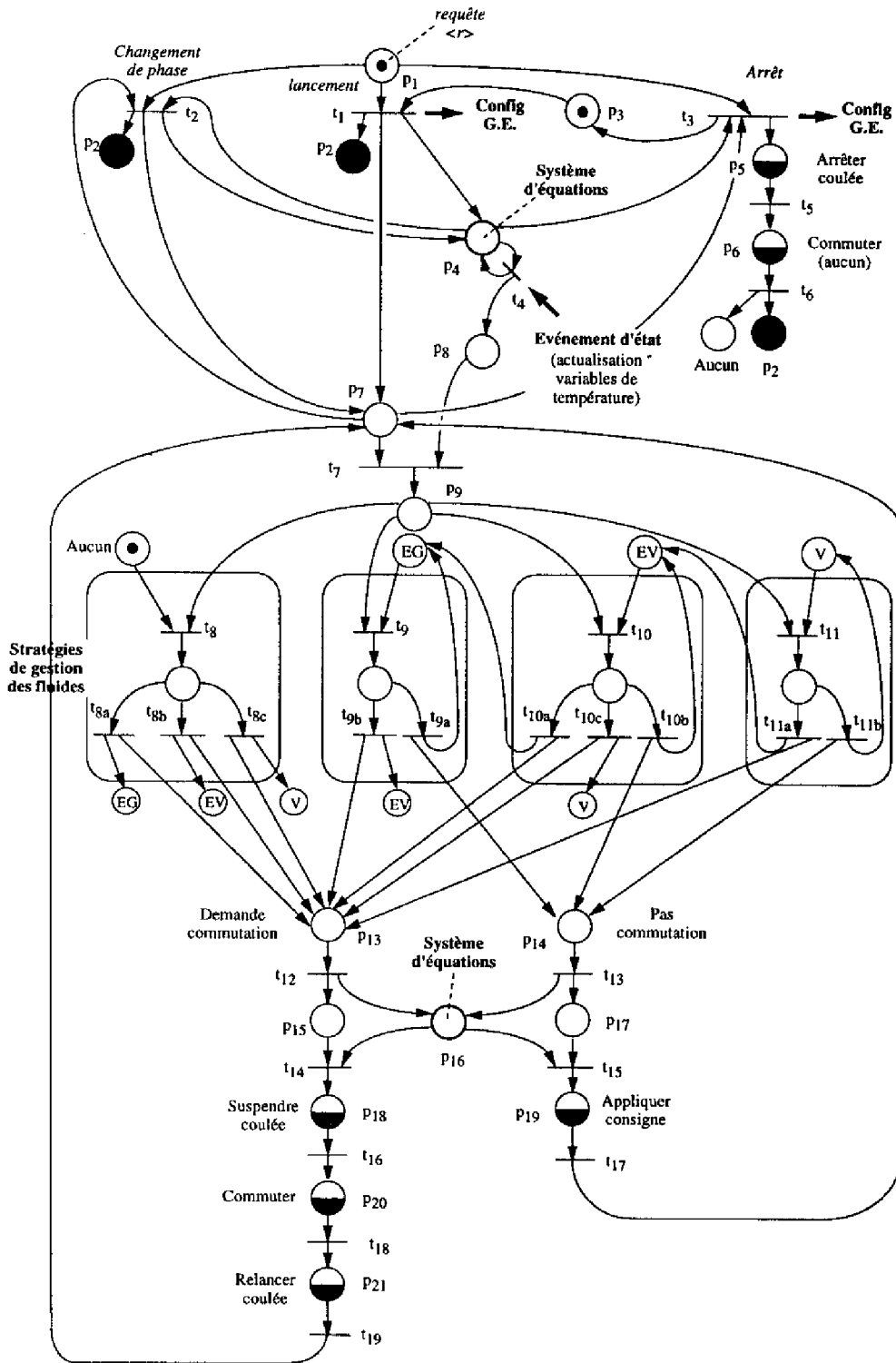


FIG. 8.5 - La fonction "Conduite thermique"

bloc	transitions	fluide	interprétation
$t_8$	$t_{8a}$	EG	$Q_r < Q_{EG_{max}}$
	$t_{8b}$	EV	$(Q_r > Q_{EG_{max}}) \wedge (Q_r < Q_{EV_{max}})$
	$t_{8c}$	V	$Q_r > Q_{EV_{max}}$
$t_9$	$t_{9a}$	EG	$Q_r < Q_{EG_{max}}$
	$t_{9b}$	EV	$Q_r > Q_{EG_{max}}$
$t_{10}$	$t_{10a}$	EG	$Q_r < Q_{EV_{min}}$
	$t_{10b}$	EV	$(Q_r > Q_{EV_{min}}) \wedge (Q_r < Q_{EV_{max}})$
	$t_{10c}$	V	$Q_r > Q_{EV_{max}}$
$t_{11}$	$t_{11a}$	EV	$Q_r < Q_{V_{min}}$
	$t_{11b}$	V	$Q_r > Q_{V_{min}}$

8.1.1: Priorité au fluide en cours

bloc	transitions	fluide	interprétation
$t_8$	$t_{8a}$	EG	$Q_r < Q_{EV_{min}}$
	$t_{8b}$	EV	$(Q_r > Q_{EV_{min}}) \wedge (Q_r < Q_{EV_{max}})$
	$t_{8c}$	V	$Q_r > Q_{EV_{max}}$
$t_9$	$t_{9a}$	EG	$Q_r < Q_{EV_{min}}$
	$t_{9b}$	EV	$Q_r > Q_{EV_{min}}$
$t_{10}$	$t_{10a}$	EG	$Q_r < Q_{EV_{min}}$
	$t_{10b}$	EV	$(Q_r > Q_{EV_{min}}) \wedge (Q_r < Q_{EV_{max}})$
	$t_{10c}$	V	$Q_r > Q_{EV_{max}}$
$t_{11}$	$t_{11a}$	EV	$Q_r < Q_{EV_{max}}$
	$t_{11b}$	V	$Q_r > Q_{EV_{max}}$

8.1.2: Priorité à l'Eau/Vapeur

TAB. 8.1 – Exemples de stratégies de gestion des fluides

la vanne correspondante, par l'intermédiaire de la fonction "Appliquer consigne" (non détaillée).

Par ailleurs, la sélection du fluide a permis d'établir la nécessité ou non d'un changement de fluide. Si ce dernier est nécessaire, la gestion du transitoire est assurée par la fonction de commutation (cf. section 1.2.3). Avant de changer de fluide, une suspension de la coulée est effectuée. La coulée, qui maintient un taux de réaction, doit nécessairement être arrêtée car le réacteur n'est pas contrôlé lors de la commutation. Une fois la commutation terminée, la coulée est relancée.

L'interaction entre les parties discrète et continue se concrétise alors par un choix de fluide dicté par l'algorithme de commande continue et par sa mise en œuvre effective par

la commande discrète [Andreu *et al.* 96b].

## 2 La simulation

Une maquette basée sur cette architecture a été développée pour la commande du procédé de chimie fine présenté. Cette maquette est composée de deux simulateurs (deux calculateurs) communiquant par une liaison série, l'un simulant la commande<sup>5</sup> et l'autre le procédé<sup>6</sup>. La synchronisation entre les simulateurs est assurée par le protocole<sup>7</sup> de communication. Un système multi-tâches temps-réel permet la mise en oeuvre de la recette et de l'ensemble des fonctions constituant la commande, dont certaines ont été présentées (cf. section 1.2). Le simulateur dynamique du réacteur discontinu repose sur l'écriture en régime transitoire des bilans de matière et d'énergie pour le milieu réactionnel, la paroi et la double enveloppe du réacteur [Cezerac *et al.* 93]. Il permet l'extrapolation à des réacteurs de taille industrielle quel que soit son mode de fonctionnement, ses caractéristiques géométriques, son type d'agitation et les réactions considérées. La modélisation des échanges thermiques est basée sur l'utilisation de modèles de connaissance du procédé qui, à partir des dimensions caractéristiques du réacteur, traduisent l'ensemble des phénomènes liés aux transferts thermiques. L'estimation des propriétés thermodynamiques fait appel à la banque de données Prophy [Lann *et al.* 88].

L'interface graphique (Figure 8.6) du simulateur de commande permet le suivi temps-réel de l'évolution du procédé. La configuration active et le fluide en cours d'utilisation sont mis en évidence par différentes colorations des connexions, l'état des organes de commande (agitateur, vannes, pompe) et les valeurs des capteurs (débits, températures, pression) sont indiqués. Une fenêtre de dialogue (en bas à gauche de l'écran) permet de signaler les requêtes incohérentes<sup>8</sup> et les erreurs de communication entre les simulateurs.

## 3 Conclusion

Nous pouvons constater l'importance de la notion de configuration. La prise en compte des configurations interdites, qui nous ramène à la problématique des états interdits, est traitée par la restriction de l'ensemble des changements d'états possibles. Un exemple a été illustré à travers la fonction "*Commutation*".

Par ailleurs, cet exemple met en évidence la clarté des interactions entre les aspects discrets et continus. Les changements de configuration, dont la combinatoire est exprimée sur le modèle à événements discrets, se concrétisent par des commutations dictées par le modèle de commande continu et exécutées par le modèle de commande discret.

---

5. implémenté en Turbo Pascal.

6. implémenté en Fortran.

7. protocole basé sur un contrôle des flots de données selon une solution matérielle (signaux DSR/DTR).

8. par exemple la requête "*activer pompe*" alors que la pompe est déjà en marche.

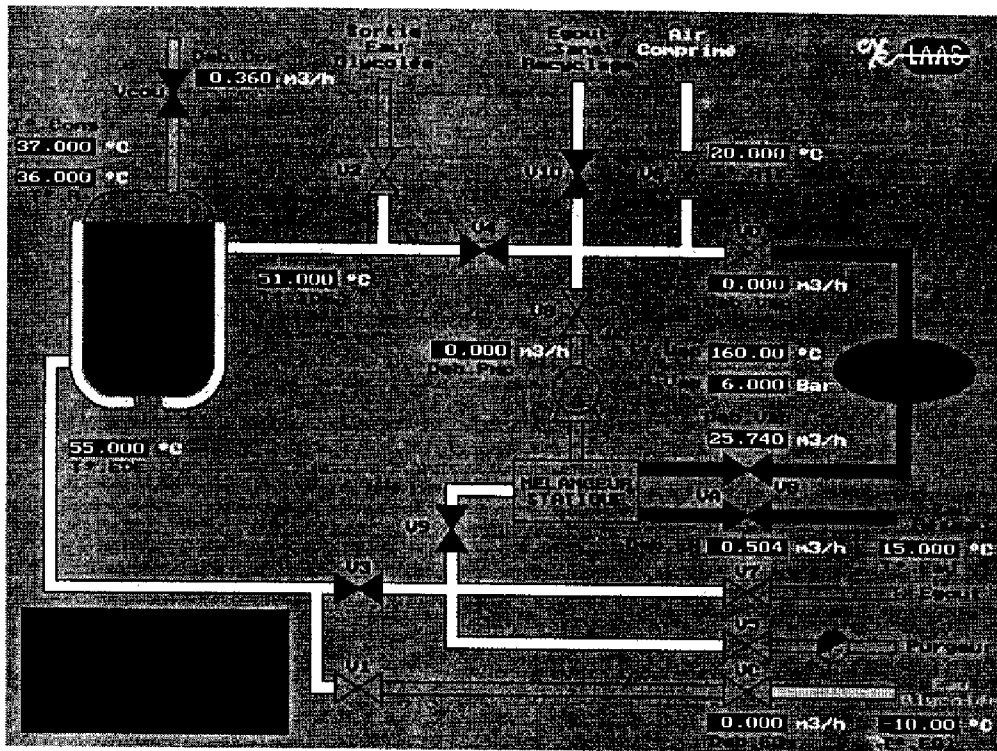


FIG. 8.6 - Interface du simulateur

Cela permet de découpler complètement la problématique de la modélisation des transformations physico-chimiques et celle du choix de la politique de commutation. Il est ainsi possible de modifier le modèle des transformations sans changer la politique de gestion des fluides et vice-versa.

Cette illustration du rôle de la coordination laisse apparaître la coordination comme un centre de décision, elle correspond à une «supervision locale». En effet, dans le cadre de l'exécution de requêtes du niveau supérieur, elle est amenée à prendre des décisions à son niveau local, de manière autonome.

Le développement d'un prototype pour la commande de ce procédé, selon l'architecture proposée, devrait nous permettre une validation partielle de notre approche. La mise en œuvre compilée des réseaux de Petri, par l'intermédiaire d'un système multi-tâches, est composée d'un ensemble de fonctions modulaires (les tâches) dont les interactions s'effectuent par communication (échanges de messages).

La phase d'interconnexion des simulateurs n'étant pas terminée, aucune simulation n'a pu être effectuée. Cependant, le système de commande, développé selon une approche hiérarchisée et modulaire, s'est avéré aisément adaptable aux évolutions des stratégies de commande<sup>9</sup> et à celles du procédé<sup>10</sup> que nous avons du effectuer.

---

9. autres politiques de gestion des fluides

10. modification du flowsheet du système de chauffage/refroidissement

## Conclusion générale

Les travaux exposés dans ce mémoire portent sur la commande et la supervision des procédés hybrides, plus particulièrement les procédés à traitements par lots (dits procédés batch). Une nouvelle architecture de commande et une approche de la problématique hybride basée sur le respect de l'identité de chaque monde, sont les points originaux de ces travaux.

L'objectif est d'avoir une approche globale, hiérarchisée et modulaire, des aspects temps-réel de la conduite de ces procédés de fabrication, reposant sur une modélisation homogène de chaque composante du système. Cette approche prend en compte les directives de la «norme» internationale ISA/SP88 concernant la commande des procédés batch.

L'architecture proposée repose sur la répartition du système de commande/supervision sur un ensemble de régulateurs continus et d'automates programmables pilotés par une station de supervision/coordination. Cette architecture est organisée autour de deux réseaux locaux : un bus de terrain à caractère continu, dédié aux variables d'état continues échantillonnées, et un bus d'atelier à caractère discret, consacré aux échanges sporadiques entre les éléments du système de commande. L'interface entre le monde des variables continues et celui des événements est assurée par un "générateur d'événements". Ce dernier ne s'attache qu'aux variables significatives pour la supervision dans la configuration courante du procédé. Ces variables significatives constituent l'ensemble des variables d'état du procédé à surveiller afin de détecter l'occurrence des changements d'états dont dépend l'évolution de la commande. Le générateur d'événements est par conséquent dynamiquement configuré par la supervision.

L'interface entre les aspects discrets et continus ne se limite pas au bas niveau de la hiérarchie. La considération de l'aspect continu, aussi bien de l'état que de la dynamique, est nécessaire au niveau de la supervision, notamment à des fins de surveillance. Le modèle utilisé par la supervision est donc nécessairement hybride, ce qui nous plonge dans la problématique de la modélisation hybride.

Le modèle du système, constitué du procédé et de sa commande, résulte de la combi-



raison de deux formalismes : les réseaux de Petri sont utilisés pour aborder l'aspect discret (événementiel), et les équations algèbro-différentielles pour représenter l'aspect continu. Ce modèle est structuré selon trois vues : une vue discrète traduisant la stratégie de commande, une vue discrète correspondant à une représentation qualitative de l'état du procédé (ses configurations) et une vue continue associée à chaque configuration. L'évolution cohérente de cette structure est garantie par une coopération des modèles dont les interactions sont clairement définies. Le modèle à événements discrets sélectionne l'ensemble des équations actives (c'est-à-dire celles traduisant les phénomènes continus en cours sur le procédé) qui pilotent alors son évolution.

Dans certains cas, une vue continue est associée directement à des états du modèle discret de la commande. Le modèle du système est alors structuré selon quatre vues.

Le modèle pour la supervision est en fait constitué d'un modèle de la commande (discrete et parfois continue) et d'un modèle hybride du procédé, appelé "modèle de référence hybride". Ce dernier est utilisé essentiellement dans le cadre de la surveillance. Piloté par le modèle de la commande, il permet la détection de déviations du comportement du procédé ainsi que la vérification de la faisabilité de l'ordonnancement établi.

Ce modèle de référence hybride correspond à une représentation agrégée du procédé à travers l'expression de ses configurations selon des changements d'état commandables et/ou observables.

Dans les cas les plus complexes, il peut être nécessaire de prendre en compte des changements de configuration du procédé ni commandables ni observables qui résulteraient de sa dynamique propre (évolution interne). Dans l'état actuel de nos travaux, nous avons supposé que ce type de changements de configuration pouvait être exprimé par des événements de temps. Il est probable que cette approximation sera suffisante dans le cadre de développements ultérieurs concernant les problèmes de diagnostic et de reprise.

Nous pouvons déjà, à partir de cette remarque, désigner une des continuités de nos travaux. Une étude approfondie de l'exemple sur la conduite de l'unité de distillation pourrait nous confronter à cette problématique de la représentation de l'évolution d'état non observable, notamment en abordant précisément les transitoires tels que les phases de démarrage et d'arrêt de la colonne à distiller entre chaque lot.

Les travaux exposés dans ce mémoire nous suggèrent d'autres réflexions. Celles-ci s'orientent selon deux axes : d'une part un approfondissement de nos travaux sur les aspects temps-réel dans le cadre d'un transfert technologique et d'autre part une étude de l'interaction avec les aspects hors temps-réel que sont la planification et l'ordonnancement.

Une mise en œuvre complète de notre approche sur une unité pilote en collaboration avec un industriel représenterait un très intéressant champ d'expérience. Aborder la conduite temps-réel d'une unité pilote selon l'approche actuelle serait une première étape permettant sans aucun doute de mettre en évidence les limitations et/ou les lacunes de nos

travaux.

La capacité de notre système à déterminer et à gérer les configurations dynamiques du procédé, nous permet d'envisager un certain nombre de perspectives de recherche portant sur :

- la sélection des variables d'état utiles au contrôle et à la supervision du procédé, afin de considérer un ordonnancement temps-réel de la table de scrutation de l'arbitre du bus de terrain (contrôleur central de la communication sur le bus, qui est quasiment réduite à un transfert périodique par l'architecture proposée). Dans une configuration donnée du procédé, l'ensemble des variables d'état dont l'image doit être mise à jour est effectivement restreint.
- la programmation dynamique des automates. Cet aspect concerne particulièrement le problème induit par le fonctionnement asynchrone du superviseur et synchrone des automates. Cette différence dans le mode de fonctionnement ne permet pas de garantir une bonne réactivité du système de commande face à la dynamique du procédé : l'exécution immédiate par un automate d'une requête issue du superviseur, suite à l'occurrence d'un événement d'état (fin de configuration et/ou changement de phase par exemple), n'est pas assurée. Le décalage temporel entre la requête et son exécution pourrait être annulé par une programmation dynamique des automates spécifiant les actions à mener dès l'occurrence d'un événement d'état particulier. Cette démarche est très délicate car il faut éviter de glisser vers une solution qui tendrait, par une remise en cause de la hiérarchie (description de la stratégie de pilotage au bas niveau), à inhiber les avantages de notre approche.
- la configuration dynamique des régulateurs. Dans le cadre de nos travaux nous avons simplement évoqué le changement de consigne et/ou de loi de commande sur les régulateurs, sans aborder les problèmes de discontinuité que peuvent induire ces changements.

Notre approche dont l'aptitude à traiter les configurations dynamiques laisse déjà apparaître des perspectives intéressantes, nous permet d'envisager également un apport sur un thème un peu plus éloigné de nos travaux : la réconciliation de données basée sur la redondance analytique. L'ensemble des équations sur lesquelles doit se baser cette redondance analytique, nécessaire à la correction d'erreurs dans l'acquisition des mesures (surveillance au niveau bas), pourrait être dynamiquement établi, offrant ainsi une réduction à la fois de l'ensemble des variables à réconcilier et de l'ensemble des équations à manipuler.

Le deuxième axe de réflexion concerne les interactions entre l'ordonnancement et le système de pilotage proposé. Nous avons souligné à travers le mémoire l'importance des recettes et le besoin de «souplesse» dans leur définition afin de pouvoir prétendre à un lien réactif entre l'ordonnancement et sa mise en œuvre. La recherche de flexibilité et de réactivité du système de production considéré globalement, c'est-à-dire en intégrant

l'ordonnancement, semble devoir être abordée par une génération automatique de la recette de contrôle à partir de la recette principale. Il s'agit d'un sujet étudié actuellement dans le cadre de la thèse de Ronan Champagnat, au sein de notre groupe de recherche.

Par ailleurs, afin d'éviter une remise en cause trop fréquente du plan de fabrication issue d'une définition «rigide» de la recette de contrôle, et pour favoriser l'interaction entre le superviseur et l'opérateur, il serait intéressant d'étudier l'apport de certaines techniques telles que le flou et le qualitatif par exemple, sur la définition des changements de phase de la recette. Nous avons en effet pu constater la difficulté et parfois l'impossibilité d'une représentation purement formelle des phénomènes étudiés, les changements d'état du procédé ne se limitent pas à des franchissements de seuil sur des variables mesurées comme nous avons pu le supposer à ce stade de nos recherches. La part d'information et les connaissances détenues par l'opérateur sont importantes, voire essentielles.

En conséquence, notre contribution doit être de proposer une approche pouvant aider l'opérateur dans l'exercice de sa tâche de conduite et de suivi de la production.

## Bibliographie

- [Akamatsu *et al.* 89] N. Akamatsu, Y. Sakaki, H.J. Brass, M. Tamaoka & K. Matsunaga. *Batch Sequence Control using a Distributed Control System with a Coordinator Station*. Dans IFAC Dynamics and Control of Chemical Reactors, pages 323–328, DYCORN+'89, Maastricht, The Netherlands, 1989.
- [Alanche *et al.* 84] P. Alanche, K. Benzakour, F. Dollé, P. Gillet, P. Rodrigues & R. Valette. *PSI: A Petri net based simulator for flexible manufacturing systems*. Lecture Notes in Computer Science, Advances in Petri Nets, vol. 188, pages 1–14, 1984.
- [Alla 94] H. Alla. *Les Réseaux de Petri: un outil particulièrement adapté à la modélisation des systèmes hybrides*. Dans IBRA, éditeur, Automatisation des processus mixtes : les systèmes dynamiques hybrides, ADPM'94, pages 17–25, Bruxelles, Belgique, Novembre 1994.
- [Andreu *et al.* 94a] D. Andreu, M. Combacau, P. Esteban & J.C. Pascal. *Commande temps-réel hiérarchisée: Application à une maquette d'atelier flexible*. Revue d'Automatique et de Productique Appliquées, vol. 7, no. 5, pages 627–639, 1994.
- [Andreu *et al.* 94b] D. Andreu, J.C. Pascal, H. Pingaud & R. Valette. *Batch Process Modeling using Petri Nets*. Dans IEEE Intern. Conf. on SMC, pages 314–319, San Antonio, USA, Octobre 1994.
- [Andreu *et al.* 95] D. Andreu, J.C. Pascal & R. Valette. *Interaction of Discrete and Continuous Parts of a Batch Process Control System*. Dans Workshop on Analysis and Design of Event-Driven Operations in Process Systems, ADEDOPS'95, London, UK, Avril 1995.
- [Andreu *et al.* 96a] D. Andreu, J.C. Pascal & R. Valette. *Events as a key of Batch Process Control System*. Dans CESA'96 IEEE-SMC IMACS Symposium on Discrete Events and Manufacturing Systems, pages 297–302, Lille, Juillet 1996.
- [Andreu *et al.* 96b] D. Andreu, H. Pingaud, J.C. Pascal & R. Valette. *Un Système de Commande Hiérarchisé Hybride: application à la commande d'un réacteur batch multifluide*. Dans Journées d'étude SEE/AFCEC, Automatisation des processus discontinus et à traitement par lots, Lyon, Février 1996.

- [Andreu *et al.* c] D. Andreu, J.C. Pascal, R. Valette, M. Cabassud, M.V. Le Lann & H. Pingaud. *Une architecture de commande hiérarchisée adaptée à la conduite des procédés batchs : application aux réacteurs exothermiques multifluide*. Dans SIMO'96 Colloque Simulation, Optimisation et Commande en Génie des Procédés, .
- [Antsaklis *et al.* 93] P.J. Antsaklis, J.A. Silver & M. Lemmon. *Hybrid System Modeling and Autonomous Control Systems*. Lecture Notes in Computer Science, vol. 736, 1993.
- [Arzen 95] K.E. Arzen. *Experiences of using G2 for Real-Time Process Control*. Dans Supervision de processus à l'aide du système expert G2, pages 21–40, Toulouse, France, Octobre 1995. Hermès.
- [Audry & Prunet 95] N. Audry & F. Prunet. *Simulation of production systems with the help of batches Petri nets*. Dans F. Breitenacker & I. Husinsky, éditeurs, EURO-SIM'95, pages 1193–1198, Vienne, Septembre 1995. Elsevier Science B.V.
- [Audry *et al.* 94] N. Audry, M.K. Ghabri, I. Demongodin, F. Prunet & P. Ladet. *Modelling and control of high throughput production lines*. Dans 1st IFAC workshop on New Trends in Design of Control Systems, pages 392–397, Smolenice, Slovak Republic, 7-10 Septembre 1994.
- [Azzopardi & Holding 95] D. Azzopardi & D.J. Holding. *Closed loop control of DEEDS using timed Petri net models*. Dans Workshop on Analysis and Design of Event-Driven Operations in Process Systems, ADEDOPS'95, London, UK, Avril 1995.
- [Bail *et al.* 91] J. Le Bail, H. Alla & R. David. *Hybrid Petri net*. Dans European Control Conference, pages 1472–1477, Grenoble France, Juillet 1991.
- [Bail *et al.* 93] J. Le Bail, H. Alla & R. David. *Asymptotic continuous Petri nets*. Discrete Event dynamic Systems: Theory and Applications, vol. 2, pages 235–263, 1993.
- [Bako *et al.* 90] B. Bako, R. Valette & M. Courvoisier. *A controlled rule-based system interpreter: an application to FMS simulation*. Dans 1st International Conference on Artificial Intelligence, Simulation and Planning in High Autonomy Systems, pages 22–24, Tucson, Arizona USA, Mars 1990.
- [Barbosa Póvoa & Macchietto 94] A.P. Barbosa Póvoa & S. Macchietto. *Redesign of a multipurpose batch pilot plant with cleaning in place (CIP) integration*. Dans Computers Chemical Engineering, éditeur, European Symposium on Computer Aided Process Engineering-3, volume 18, pages 277–281. Pergamon Press Ltd, 1994.
- [Bellorin 94] J. Bellorin. *Automatisation des systèmes continus à traitements par lots : étude des spécifications des fonctions de coordination et de leurs réactivités*. Thèse de doctorat, Ecole Supérieure d'Electricité, Rennes, Juin 1994.

- [Benveniste *et al.* 93] A. Benveniste, M. Le Borgne & P. Le Guernic. *Hybrid Systems: the SIGNAL approach*. Dans R.L. Grossman, A. Nerode, A.P. Ravn & H. Rischel, éditeurs, Hybrid Systems, volume Lecture Notes in Computer Science 736, pages 230–254. Springer Verlag, 1993.
- [Bergé 96] N. Bergé. *Modélisation au moyen des réseaux de Petri temporisés stochastiques d'une application de contrôle-commande de poste de transformation d'énergie électrique répartie sur le réseau de terrain FIP*. Thèse de doctorat, Université Paul Sabatier, Toulouse, Janvier 1996.
- [Brielman 95] M. Brielman. *Modelling Differential Equations by Basic Information Technology Means*. Dans Proceedings of the Fifth International Conference on Computer Aided Systems Theory and Technology, EUROCAST'95, volume Lecture Notes in Computer Science 1030, pages 163–174, Innsbruck, Mai 1995. Springer Verlag.
- [Cardoso 90] J. Cardoso. *Sur les réseaux de Petri avec marquage flou*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France, Octobre 1990.
- [Cassandras 93] C.G. Cassandras. *Discrete event systems: Modeling and performance analysis*. Aksen Associates, Boston, MA, USA, Irwin edition, June 1993.
- [CEP 92] *Chemical Engineering Progress*, Avril 1992.
- [CEP 93] *Chemical Engineering Progress*, Février 1993.
- [Cezerac *et al.* 93] J. Cezerac, M.V. Le Lann, M. Cabassud, G. Casamatta, J. Albert & J.M. Le Lann. *Développement d'un modèle de simulation dynamique en vue de la conduite automatisée d'un réacteur discontinu par la pression du ciel gazeux*. Dans Récents Progrès en Génie des Procédés C. Jallut, éditeur, 4ème Congrès Français de Génie des Procédés, volume 7 (29), pages 1–6, Grenoble, France, Septembre 1993. Lavoisier, Paris.
- [Chaillet-Subias 95] A. Chaillet-Subias. *Approche multi modèles pour la commande et la surveillance en temps réel des systèmes à événements discrets*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France, Décembre 1995.
- [Chérury & Flaus 94] A. Chérury & J.M. Flaus. *Des mesures indirectes à l'estimation en ligne*. Dans Lavoisier Tec&Doc, éditeur, Capteurs et mesures en biotechnologies, pages 444–487. Paris, 1994.
- [CM 95] *Automatisation de procédés batch. Chimie Magazine*, Avril 1995.
- [Combacau & Courvoisier 90] M. Combacau & M. Courvoisier. *A hierarchical and modular structure for F.M.S. control and monitoring*. Dans 1st International Conference on Artificial Intelligence, Simulation and Planning in High Autonomy Systems, Tuscon, Arizona, USA, Mars 1990.
- [Combacau 91] 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.

- [Committee 90] Working Committee. *Recommendation for a variable software for the configuration of batch controllers*. Rapport technique, Namur Work Committee, 1990.
- [Cott & Macchietto 88] B.J. Cott & S. Macchietto. *An integrated approach to Computer-Aided Operation of batch chemical plants*. Dans *Process Systems Engineering*, Sydney, Australia, Août 1988.
- [Cruette 91] D. Cruette. *Méthodologie de conception des systèmes complexes à événements discrets : application à la conception et à la validation hiérarchisée de la commande de cellules flexibles de production dans l'industrie manufacturière*. Thèse de doctorat, Université de Lille, Février 1991.
- [Daubas *et al.* 94] B. Daubas, A. Pages & H. Pingaud. *Combined simulation of hybrid processes*. Dans *IEEE Intern. Conf. on SMC*, pages 314–319, San Antonio, USA, Octobre 1994.
- [Daubas 94] B. Daubas. *Modélisation et simulation des procédés continus et discontinus*. Thèse de doctorat, Institut National Polytechnique de Toulouse, Toulouse, France, Novembre 1994.
- [David & Alla 87] R. David & H. Alla. *Continuous Petri Nets*. Dans *8th European Workshop on Application and Theory of Petri Nets*, pages 275–294, Zaragoza, Spain, Juin 1987.
- [David & Alla 89] R. David & H. Alla. *Du grafcet aux réseaux de Petri*. Paris, Hermès edition, 1989.
- [David & Alla 94] R. David & H. Alla. *Petri Nets for Modeling of Dynamic Systems - A Survey*. *Automatica*, vol. 30, no. 2, pages 175–202, 1994.
- [David 91] R. David. *Modeling of Dynamic Systems by Petri Nets*. Dans *1st European Control Conference*, pages 136–147, Grenoble, France, 1991.
- [de Bonneval 93] A. de Bonneval. *Mécanismes de reprise dans les systèmes de commande à événements discrets*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France, Septembre 1993.
- [Decotignie & Pleinevaux 93] J.D. Decotignie & P. Pleinevaux. *A survey on industrial communication networks*. *Ann. Telecomm.*, vol. 48, no. 9-10, pages 435–448, 1993.
- [Demongodin & Koussoulas 96] I. Demongodin & N.T. Koussoulas. *Modeling dynamic systems through Petri nets*. Dans *CESA'96 IEEE-SMC IMACS Symposium on Discrete Events and Manufacturing Systems*, pages 279–284, Lille, France, Juillet 1996.
- [Demongodin *et al.* 93] I. Demongodin, N. Audry & F. Prunet. *Batches Petri Nets*. Dans *IEEE International Conference on Systems, Man and Cybernetics*, pages 607–617, Le Touquet, France, Octobre 1993.

- [Demongodin *et al.* 94] I. Demongodin, N. Audry & F. Prunet. *Outil de Modélisation des Systèmes Hybrides: les Réseaux de Petri Lots à caractéristique variable*. Dans Ibra, editeur, Automatisation des processus mixtes: les systèmes dynamiques hybrides, ADPM'94, pages 227-232, Bruxelles, Belgique, Novembre 1994.
- [Destenave 91] G. Destenave. *Les spécificités de l'automatisation dans l'industrie agro-alimentaire*. Dans Journée SEE sur l'automatisation dans les IAA, Grenoble, Avril 1991.
- [Dubois & Gentil 90] D. Dubois & S. Gentil. *Intelligence Artificielle: un outil pour l'automatique*. Journées annuelles du GR automatique, Octobre 1990.
- [Dubois *et al.* 94] E. Dubois, H. Alla & R. David. *Les réseaux de Petri Continus à Vitesses Fonction du Temps*. APII, vol. 28, no. 5, pages 425-443, 1994.
- [Engell & Wöllhaf 94] S. Engell & K. Wöllhaf. *Dynamic Simulation for Improved Operation of Flexible Batch Plants*. Dans Rutgers, editeur, Conference on Computer Integrated Manufacturing in the Process Industries, pages 441-455, East Brunswick, Avril 1994.
- [Engell *et al.* 95] S. Engell, M. Fritz, C. Schulz & K. Wöllhaf. *Hybrid Simulation of Flexible Batch Plants*. Dans IFAC Symposium DYCORS'95, pages 33-38, Helsinki, Juin 1995.
- [Esquirol *et al.* 95] P. Esquirol, M.J. Huguet & P. Lopez. *Modelling and managing disjunctions in scheduling problems*. Journal of Intelligent Manufacturing, vol. 6, pages 133-144, 1995.
- [Féray-Beaumont & Gentil 91] S. Féray-Beaumont & S. Gentil. *Modèle qualitatif de comportement pour un système d'aide à la supervision*. APII, vol. 25, pages 325-348, 1991.
- [Fleck & Hanisch 96] S. Fleck & H.M. Hanisch. *A Resource Allocation Scheme for Flexible Batch Plants based on High-Level Petri Nets*. Dans CESA'96 IEEE-SMC IMACS Symposium on Discrete Events and Manufacturing Systems, pages 303-308, Lille, Juillet 9-12 1996.
- [Genrich & H.M.Hanisch 95] H.J. Genrich & H.M.Hanisch. *Modelling and Analysis of Recipes*. Dans Workshop on Analysis and Design of Event-Driven Operations in Process Systems, ADEDOPS'95, London, UK, Avril 1995.
- [Genrich *et al.* 94] H.J. Genrich, H.M.Hanisch & K. Wollhaf. *Verification of Recipe-Based Control Procedures by Means of Predicate/transition Nets*. Dans Lecture Notes in Computer Science 815, Application and Theory of Petri Nets, pages 278-297, Zaragoza, Spain, Juin 1994. Springer-Verlag.
- [Genrich 87] H.J. Genrich. *Predicate/transition Nets*. Dans Lecture Notes in Computer Science. Springer-Verlag, 1987.



- [Gentil 95] S. Gentil. *Systèmes d'aide à la supervision*. Dans *Supervision de processus à l'aide du système expert G2*, pages 7–20, Toulouse, Octobre 1995. Hermès, Paris.
- [Georges 96] D. Georges. *Aide à la décision dans la production d'énergie électrique*. Dans *Journées d'étude SEE/AF CET, Pôle Productique Rhône-Alpes, Analyse et Supervision des Systèmes Dynamiques Hybrides*, Lyon, Février 1996.
- [Gollu & Varaiya 89] A. Gollu & P. Varaiya. *Hybrid Dynamical Systems*. Dans *28th Conf. on Decision and Control*, pages 2708–2712, Tampa, Florida, USA, Décembre 1989.
- [Gomaa & Gentil 95] M. Gomaa & S. Gentil. *La Modélisation des Systèmes Industriels par les Réseaux de Petri Continus Causaux*. Rapport technique 95/100, Laboratoire d'Automatique de Grenoble, ENSIEG-INPG St Martin d'Hères France, Septembre 1995.
- [Gomaa & Gentil 96a] M. M. Gomaa & S. Gentil. *Systèmes Dynamiques Hybrides : Une nouvelle approche de modélisation pour la supervision*. Dans *Journées d'étude SEE/AF CET, Pôle Productique Rhône-Alpes, Analyse et Supervision des Systèmes Dynamiques Hybrides*, Lyon, Février 1996.
- [Gomaa & Gentil 96b] M. M. Gomaa & S. Gentil. *Hybrid industrial dynamical system supervision via Hybrid Continuous Causal Petri Nets*. Dans *CESA'96 IEEE-SMC IMACS Symposium on Discrete Events and Manufacturing Systems*, pages 285–290, Lille, Juillet 1996.
- [Guéguen 96] H. Guéguen. *Uses of StateCharts and Signal for the specification of control of hybrid systems*. Dans *CESA'96 IEEE-SMC IMACS Symposium on Discrete Events and Manufacturing Systems*, pages 380–384, Lille, Juillet 9-12 1996.
- [Hanisch 92] H.M. Hanisch. *Coordination Control Modelling in Batch Production Systems by means of Petri Nets*. *Computer Chemical Engineering*, vol. 16, no. 1, pages 1–10, 1992.
- [Holloway & Krogh 90] L.E. Holloway & B.H. Krogh. *Fault Detection and Diagnosis in Manufacturing Systems: A Behavioral Model Approach*. Dans *2nd IEEE International Conference on Computer Integrated Manufacturing*, Mai 1990.
- [ISA/SP88 94] ISA/SP88. *Batch Control, Models and Terminology*. Rapport technique Draft 10, Instrument Society of America, Research triangle Park, NC USA, Mars 1994.
- [Jallut & Gilles 90] C. Jallut & G. Gilles. *Automatique, productique et procédés chimiques discontinus*. *Revue Générale d'Electricité*, no. 6, pages 1–6, Juin 1990.

- [Jambois *et al.* 95] P. Jambois, D. Maquin & F.X. Renard. *Flownet: validation de mesures et diagnostic en ligne*. Dans *Supervision de processus à l'aide du système expert G2*, pages 51–56, Toulouse, Octobre 1995. Hermès, Paris.
- [Jensen 86] K. Jensen. *Coloured Petri Nets*. Dans *Lecture Notes in Computer Science*. Springer-Verlag, 1986.
- [Johnsson & Arzen 94] C. Johnsson & K.E. Arzen. *High-Level Grafset and Batch Control*. Dans Ibra, editeur, *Automatisation des processus mixtes: les systèmes dynamiques hybrides*, ADPM'94, pages 255–263, Bruxelles, Belgique, Novembre 1994.
- [Jost 92] R. Jost. *Flexibatch ou la maîtrise complète par l'exploitant des séquences de fabrication*. Dans *Automatisation des procédés mixtes continus et séquentiels: les processus «dis-continus»*, ADPM'92, pages 205–210, Janvier 1992.
- [Künzle *et al.* 94] L.A. Künzle, B. Pradin-Chézalviel, F. Girault & R. Valette. *Synthesis of monitoring functions based on a formal specification*. Dans INRIA European Workshop on Integrated Manufacturing System Engineering - IMSE'94, pages 287–295, Grenoble, France, Décembre 1994.
- [Kondili *et al.* 93] E. Kondili, C.C. Pantelides & R.W.H. Sargent. *A general algorithm for short-term scheduling of batch operations - MILP formulation*. *Computers Chemical Engineering*, vol. 17, no. 2, pages 211–227, 1993.
- [Kopetz 92] H. Kopetz. *Six difficult problems in the design of responsive systems*. Dans H. Kopetz & Y. Kakuda, éditeurs, *Dependable computing and fault tolerant systems: Responsive computer systems'92*, volume 7, pages 3–26. Springer-Verlag, 1992.
- [Lann *et al.* 88] J.M. Le Lann, Joulia X. & B. Koehret. *I.C.E.*, vol. 28, no. 1, pages 36–65, 1988.
- [Lann *et al.* 95] M.V. Le Lann, M. Cabassud & G. Casamatta. *Adaptive Model Predictive Control*. Dans R. Berber, editeur, *Methods of Model Based Process Control*, pages 426–458, Dordrecht, 1995. Kluwer Academic Publishers.
- [Lebourgeois 96] F. Lebourgeois. *La conception de la conduite des batchs par la méthode Astrid*. Dans *Journées d'étude SEE/AF CET*, Automatisation des processus discontinus et à traitement par lots, Lyon, Février 1996.
- [Lemmon *et al.* 93] M. Lemmon, J.A. Stiver & P.J. Antsaklis. *Event identification and intelligent hybrid control*. *Lecture Notes in Computer Science*, vol. 736, pages 268–296, 1993.
- [Leyval *et al.* 94] L. Leyval, S. Gentil & S. Feray-Beaumont. *Model-based Causal Reasoning for Process Supervision*. *Automatica*, vol. 30, no. 8, pages 1295–1306, 1994.

- [Lhoste 91] P. Lhoste. *Surveillance des M.S.A.P. : les atouts de la modélisation de comportement*. Dans Journées surveillance du pôle SED (GT2) du GR Automatique, Paris, Février 1991.
- [Liu & Macchietto 94] Z.H. Liu & S. Macchietto. *Dynamic modelling and simulation of a multi-purpose batch pilot plant*. Dans ADCHEM'94, Kyoto, Japan, Mai 25-27 1994.
- [Liu & Macchietto 95] Z.H. Liu & S. Macchietto. *An advanced operations support system for batch plants*. Dans Workshop on Analysis and Design of Event-Driven Operations in Process Systems, ADEDOPS'95, London, UK, Avril 1995.
- [Lloyd & Salleh 91] S. Lloyd & M. Salleh. *Modeling and control design of batch process plant by Timed Petri Net*. Dans Proceedings of the 30th Conference on Decision and Control, pages 1531–1536, Brighton, England, Décembre 1991.
- [Louleh 96] Z. Louleh. *Modélisation et conduite des réacteurs discontinus par analyse des flux thermiques*. Thèse de doctorat, Institut National Polytechnique de Toulouse, Toulouse, France, Juillet 1996.
- [Macchietto et al. 89] S. Macchietto, B.J. Cott, I.M. Mujtaba & C.A. Crooks. *Optimal Control and On-line Operation of Batch Distillation*. Dans Annual AIChE Meeting, San Francisco, USA, Novembre 1989.
- [Macchietto et al. 92] S. Macchietto, C. Crooks & K. Kuriyan. *An integrated system for batch processing*. Dans NATO Advance Study Institute on Batch Processing Systems Engineering, Antalya, Turkey, Mai-Juin 1992.
- [Macchietto 93] S. Macchietto. *Bridging the gap - integration of design. Operations scheduling and control*. Dans FOCAPO II Conference, pages 207–230, Colorado, USA, Juillet 1993.
- [Mehta 83] G.A. Mehta. *The Benefits of Batch Process Control*. Chemical Engineering Progress, pages 47–52, Octobre 1983.
- [Montmain 94] J. Montmain. *Raisonnement Approximatif et Graphes Causaux pour la détection et la localisation de défaillances*. Dans Journées nationales : Sécurité, Surveillance, Supervision. GR Automatique, Paris, 17-18 Novembre 1994.
- [Nerode & Kohn 93] A. Nerode & W. Kohn. *Models for Hybrid Systems: Automata, Topologies, Controllability, Observability*. Dans R.L. Grossman, A. Nerode, A.P. Ravn & H. Rischel, éditeurs, Hybrid Systems, volume Lecture Notes in Computer Science 736, pages 317–356. Springer Verlag, 1993.
- [Nicollin et al. 93] X. Nicollin, A. Olivero, J. Sifakis & S. Yovine. *An Approach to the Description and Analysis of Hybrid Systems*. Dans R.L. Grossman, A. Nerode, A.P. Ravn & H. Rischel, éditeurs, Hybrid Systems, volume Lecture Notes in Computer Science 736, pages 149–178. Springer Verlag, 1993.

- [Niel *et al.* 96] E. Niel, N. Rezg, M. Nourelfath & S. Boukohbza. *Supervisory Control in the context of Operational Safety Reactivity*. Dans CESA'96 IEEE-SMC IMACS Symposium on Discrete Events and Manufacturing Systems, pages 746–751, Lille, Juillet 1996.
- [Noury & Laine 92] P. Noury & T. Laine. *Le RLI temps critique pour l'automatisation des procédés discrets-continus*. Dans Automatisation des procédés mixtes continus et séquentiels: les processus «dis-continus», ADPM'92, pages 105–117, Janvier 1992.
- [Pages & Pingaud 94] A. Pages & H. Pingaud. *Un outil d'aide à l'ordonnancement d'ateliers batchs à production continue*. Dans Ibra, editeur, Automatisation des processus mixtes: les systèmes dynamiques hybrides, ADPM'94, Bruxelles, Belgique, Novembre 1994.
- [Pages & Pingaud 95] A. Pages & H. Pingaud. *An hybrid process model based on Petri nets applied on short term scheduling of batch/semi continuous plants*. Dans Workshop on Analysis and Design of Event-Driven Operations in Process Systems, ADEDOPS'95, London, UK, Avril 1995.
- [Pantelides 93] C.C. Pantelides. *Unified Frameworks for Optimal Process Planning and Scheduling*. Dans FOCAPO II Conference, pages 253–274, Colorado, USA, Juillet 1993.
- [Peleties & DeCarlo 89] P. Peleties & R. DeCarlo. *A Modeling Strategy with Event Structures for Hybrid Systems*. Dans 28th Conf. on Decision and Control, pages 1308–1313, Tampa, Florida, USA, Décembre 1989.
- [Peleties & Decarlo 94] P. Peleties & R. Decarlo. *Analysis of a Hybrid System Using Symbolic Dynamics and Petri Nets*. Automatica, vol. 30, no. 9, pages 1421–1427, 1994.
- [Pierreval 94] H. Pierreval. *La simulation de fonderies: une modélisation combinée discrète-continue*. Dans Ibra, editeur, Automatisation des processus mixtes: les systèmes dynamiques hybrides, ADPM'94, pages 159–165, Bruxelles, Belgique, Novembre 1994.
- [Pingaud *et al.* 92] H. Pingaud, B. Daubas & B. Koehret. *Conception d'un simulateur de procédés continus, semi-continus et discontinus: une évolution vers la conduite intégrée des procédés*. Dans Automatisation des procédés mixtes continus et séquentiels: les processus «dis-continus», ADPM'92, pages 195–204, Janvier 1992.
- [Pleinevaux & Decotignie 88] P. Pleinevaux & J.D. Decotignie. *Time critical communication networks: Field buses*. Dans IEEE Network, volume 2, pages 55–63, Mai 1988.
- [Pourcel 91] C. Pourcel. *Conduite des procédés continus et manufacturiers: une même approche?* Dans 3ème congrès International du Génie Industriel, pages 61–70, Tours, France, Mars 1991.

- [Puigjaner *et al.* 94] L. Puigjaner, A. Huercio & A. Espuna. *Batch Production Control in a computer integrated manufacturing environment*. Journal of Process Control, vol. 4, no. 4, pages 281–290, Novembre 1994.
- [Quénec'hdu & Buisson 92] Y. Quénec'hdu & J. Buisson. *Le génie automatique dans l'automatisation des processus continus*. Revue Générale d'Electricité, no. 4, pages 61–66, Avril 1992.
- [Quénec'hdu *et al.* 95] Y. Quénec'hdu, H. Guéguen & J. Buisson. *Les systèmes dynamiques hybrides : une nouvelle problématique*. Revue Générale d'Electricité, no. 1, pages 2–9, Janvier 1995.
- [Quénec'hdu *et al.* 96] Y. Quénec'hdu, H. Cormerais & J. Buisson. *Les systèmes dynamiques hybrides : l'approche continue appliquée à l'électronique de puissance*. Dans Journées d'étude SEE/AF CET, Pôle Productique Rhône-Alpes, Analyse et Supervision des Systèmes Dynamiques Hybrides, Lyon, Février 1996.
- [Rakotoson *et al.* 92] M.P. Rakotoson, C. Vercauter & J.C. Gentina. *Analyse des systèmes de production dits «continus-discrets» et critères de hiérarchisation du niveau coordination/supervision*. Dans Automatisation des procédés mixtes continus et séquentiels : les processus «dis-continus», ADPM'92, pages 11–22, Janvier 1992.
- [Ramadge & Wonham 89] P.J.G. Ramadge & W.M. Wonham. *The Control of Discrete Event Systems*. Proc. IEEE, Special issue on Discrete Event Dynamic Systems, vol. 77, no. 1, pages 81–98, Janvier 1989.
- [Rippin 89] D.W.T. Rippin. *Control of batch processes*. Dans IFAC Dynamics and Control of Chemical Reactors, pages 131–141, DYCORS'89, Maastricht, The Netherlands, 1989.
- [Rippin 93] D.W.T. Rippin. *Batch Process systems engineering: a retrospective and prospective review*. Computer and Chemical Engineering, vol. 17, pages s1–s13, 1993.
- [Rosa *et al.* 94] P.F. Rosa, F. Vasques & R. Valette. *The network transparency concept in fieldbus based distributed systems*. Dans ISIE94 - IEEE Int. Symp. on Industrial Electronics, Santiago, Chile, Mai 1994.
- [Rosenof & Ghosh 87] H.P. Rosenof & A. Ghosh. *Batch process automation : Theory and practice*. Van Nostrand Reinhold Company, New York, USA, 1987.
- [Rotstein *et al.* 95] G.E. Rotstein, A. Sanchez, N. Alops, L. Solfaroli Camillocci & S. Macchietto. *Synthesis of Procedural Controllers and the Automatic Generation of Sequential Control Code*. Dans Workshop on Analysis and Design of Event-Driven Operations in Process Systems, ADEDOPS'95, London, UK, Avril 1995.

- [Sanchez & Macchietto 93] A. Sanchez & S. Macchietto. *Synthesis of Operating Procedures for Chemical Processes using DES Theory*. Dans American Control Conference ACC'93, San Francisco, USA, 1993.
- [Sawyer 92] P. Sawyer. *Batch Standards on the move*. Dans The Chemical Engineer, pages 28-30, Juillet 1992.
- [Sawyer 93] P. Sawyer. Computer-controlled batch processing. J.W. Arrowsmith Ltd, Bristol, UK, 1993.
- [Sibertin-Blanc 88] C. Sibertin-Blanc. *Le prototypage des applications interactives à l'aide des réseaux de Petri*. Dans Journées internationales : Le génie logiciel et ses applications, pages 837-856, Toulouse, France, Décembre 1988. Springer-Verlag.
- [Soto-Fernandez 89] L. Soto-Fernandez. *Contribution à l'étude des moyens de description, simulation et conduite des processus industriels à traitement par lots*. Thèse de doctorat, Université de Rennes I, Rennes, Mai 1989.
- [Stromberg et al. 93] J.E. Stromberg, J. Top & U. Soderman. *Modeling mode switching in dynamic systems*. Dans 2nd ECC'93, volume 2, pages 848-853, Groningen, The Netherlands, 1993.
- [Tittus et al. 95] M. Tittus, M. Fabian & B. Lennartson. *Controlling and Coordinating Recipes in Batch Applications*. Dans 34th Conference on Decision and Control, pages 2484-2489, New Orleans, LA, USA, Décembre 1995.
- [Tittus 95] M. Tittus. *Control Synthesis for Batch Processes*. Thèse de doctorat, School of Electrical and Computer Engineering, Chalmers University of Technology, Goteborg, Sweden, Octobre 1995.
- [Valentin 93] C. Valentin. *Contribution à la modélisation et à la conduite des procédés mixtes (continus-discrets) : application à l'industrie papetière*. Thèse de doctorat, Institut National Polytechnique de Grenoble, Laboratoire d'Automatique de Grenoble, France, Février 1993.
- [Valette & Künzle 94] R. Valette & L.A. Künzle. *Réseaux de Petri pour la détection et le diagnostic*. Dans Journées nationales : Sécurité, Surveillance, Supervision. GR Automatique, Paris, 17-18 Novembre 1994.
- [Valette et al. 95] R. Valette, H. Pingaud, A. Pages, D. Andreu & J.C. Pascal. *Modeling, simulation and control of event-driven operations in process systems*. Dans INRIA-IEEE Symposium on Emerging Technologies and Factory Automation, volume 3, pages 119-128, Paris France, Octobre 10-13 1995.
- [Valette 90] R. Valette. Les réseaux de Petri (cours INSAT-DGE). LAAS-CNRS, Toulouse, 1990.

- [Valette 95] R. Valette. *Petri nets for control and monitoring: specification, verification, implementation*. Dans Workshop on Analysis and Design of Event-Driven Operations in Process Systems, ADEDOPS'95, London, UK, Avril 1995.
- [Vanbergen 94] J.P. Vanbergen. *Vue générale des problèmes de l'automatisation des batches*. Dans Ibra, editeur, Automatisation des processus mixtes: les systèmes dynamiques hybrides, ADPM'94, pages 265–270, Bruxelles, Belgique, Novembre 1994.
- [Venkatasubramanian & Stanley 93] V. Venkatasubramanian & G.M. Stanley. *Integration of Process Monitoring, Diagnosis and Control: Issues and Emerging Trends*. Dans FOCAPO II Conference, Colorado, USA, Juillet 1993.
- [Wieting & Sonnenschein 95] R. Wieting & M. Sonnenschein. *Extending High-Level Petri Nets for Modeling Hybrid Systems*. Dans A. Sydow, editeur, IMACS Symposium on System Analysis and Simulation, volume 18-19, pages 259–262, Berlin, Juin 1995. Gordon and Breach.
- [Yamalidou & Kantor 91] E.C. Yamalidou & J.C. Kantor. *Modeling and optimal control of discrete-event chemical processes using Petri nets*. Computer Chem. Engng, vol. 15, no. 7, pages 503–519, 1991.
- [Yamalidou et al. 90] E.C. Yamalidou, E.P. Patsidou & J.C. Kantor. *Modeling Discrete-Event Dynamical Systems for Chemical Process Control - A survey of several new techniques*. Computer Chemical Engineering, vol. 14, no. 3, pages 281–299, 1990.
- [Yazdi & Jorgensen 96] H. Yazdi & S.B. Jorgensen. *On PCT and Grafset - two approaches in event-driven systems*. Dans CESA'96 IEEE-SMC IMACS Symposium on Discrete Events and Manufacturing Systems, pages 407–412, Lille, Juillet 1996.
- [Yazdi et al. 95] H. Yazdi, M. Bahar, A. Koggersbol & S.B. Jorgensen. *Knowledge-based control structuring of a distillation plant start-up*. Control Engineering Practice, vol. 3, no. 3, pages 423–430, 1995.
- [Yi & Reklaitis 94] G. Yi & G.V. Reklaitis. *Model based feedback inventory controller for intermediate storage in batch processes*. Journal of Process Control, vol. 4, no. 4, pages 227–237, 1994.
- [Zamai 94] E. Zamai. *Commande et Surveillance des systèmes à événements discrets complexes: Utilisation d'un modèle du procédé*. Rapport de DEA, LAAS-CNRS et Université Paul Sabatier, Toulouse, Juillet 1994.
- [Zerhouni & Alla 92] N. Zerhouni & H. Alla. *Sur l'analyse des lignes de fabrication par réseaux de Petri continus*. APII, vol. 26, pages 253–276, 1992.

## Annexe

Cette présentation sommaire des RdP n'est pas exhaustive, les lecteurs pourront trouver les définitions formelles, les méthodes d'analyse des propriétés, ainsi que des exemples d'applications dans les références suivantes notamment [David & Alla 89], [Valette 90], [David 91], [David & Alla 94].

### Les réseaux de Petri

Le modèle réseaux de Petri est un outil de modélisation de la dynamique des systèmes. Il permet de décrire les relations existant entre des conditions et des événements, et de représenter des comportements comprenant séquentialité, parallélisme, synchronisation, et partage de ressource. Le modèle réseau de Petri interprété (non-autonome) est composé de :

- un ensemble fini de places auxquelles on peut associer la notion d'état. On représente généralement par des places les états des différentes entités du système modélisé (exemples de places : «réservoir libre», «réaction en cours»). Une place correspond donc à un état partiel du système auquel une action peut être associée.
- un ensemble fini de transitions décrivant l'évolution ou le changement d'état du système. Des conditions supplémentaires de franchissement peuvent être associées aux transitions, elles font intervenir des données et/ou des événements extérieurs (valeurs de capteurs, réceptions de messages par exemple). L'évolution du réseau est induite par le franchissement (ou tir) des transitions. Le tir d'une transition ne dépend généralement pas de l'état global du réseau mais plutôt de certains états partiels qui sont exprimés par les places en amont de la transition considérée. Lors du franchissement d'une transition, les jetons sont enlevés des places d'entrée pour être déposés dans les places de sortie.
- un ensemble fini d'arcs traduisant les relations de cause à effet entre l'état du système et ses évolutions possibles. Les arcs reliant des places à une transition indiquent les états partiels nécessaires pour que l'évolution modélisée par la transition puisse avoir lieu (pré-conditions). Les arcs reliant une transition à des places indiquent les états



partiels nécessairement atteints lorsque l'évolution modélisée par la transition a lieu (post-conditions).

- un ensemble de jetons qui représentent les marques du réseau. Ils permettent de modéliser les ressources, les produits, les messages, etc. La position du jeton dans le réseau donne l'état de l'entité (ou des entités) associée(s). L'état global effectif du modèle est défini par le marquage du réseau.

Le RdP non interprété représente donc les enchaînements potentiels des événements et des activités, il définit la *structure de contrôle* du système modélisé (partie contrôle). L'interprétation du RdP spécifie les liens entre la partie contrôle d'une part, et la partie opérative qui décrit les données internes (avec ses calculs) et l'environnement du système, d'autre part.

Le formalisme des RdP offre la possibilité d'analyser le modèle et de s'assurer ainsi de la cohérence de la représentation du système. En effet ses fondements mathématiques permettent de prouver certaines propriétés du modèle, qui, portant seulement sur la structure du réseau, donnent une validation partielle du modèle.

Pour des applications complexes, la taille du modèle devient rapidement difficile à maîtriser. Des abréviations, apportant individualisation du jeton et repliage du réseau ont permis d'augmenter la compacité du modèle (représentation simplifiée). Ces possibilités sont offertes par les RdP de Haut-niveau que sont les RdP colorés [Jensen 86], RdP à prédicats [Genrich 87] ou encore les RdP à objets [Sibertin-Blanc 88].

Les RdP colorés étant utilisés dans les approches séparées qui ont été présentées, rappelons-en le principe.

Dans les RdP colorés, des identificateurs appelés couleurs sont associés aux jetons, aux places et aux transitions. La couleur du jeton apporte une information sur l'identité du jeton, la couleur des places sur la couleur des jetons pouvant y séjourner, tandis que la couleur des transitions exprime les manières de les franchir. Des fonctions sur les arcs déterminent la couleur et le nombre de jetons qui doivent être retirés des places d'entrée de la transition suivant les couleurs de la place et de la transition.

La description faite jusqu'ici des RdP n'évoque pas la prise en compte du temps explicitement et quantitativement. Le réseau de Petri ordinaire décrit une relation de causalité entre des événements, ces relations de précédence expriment la notion de temps seulement de manière qualitative. Dans les approches présentées, et pour les besoins de surveillance abordés dans nos travaux, la prise en compte du temps est un aspect essentiel. Différents modèles de RdP intègrent explicitement la notion de temps dont les RdP temporisés et les RdP temporels.

Les RdP temporisés, plus précisément les RdP à transitions temporisées (dual des RdP à places temporisées), sont caractérisés par une durée de tir  $d_i$  associée à chaque transition  $T_i$ . Le tir de la transition n'est plus instantané, mais dure  $d_i$ . Ce n'est qu'après

écoulement de la temporisation  $d_i$  que les jetons sont enlevés des places d'entrée pour être déposés dans les places de sortie. Toutefois dès que l'on a décidé de franchir une transition, les jetons impliqués sont réservés et ne peuvent pas être utilisés pour le franchissement d'une autre transition. Pour les Rdp temporels, le modèle associe à chaque transition un intervalle  $[\tau_{min}, \tau_{max}]$  (tel que  $\tau_{min} \leq \tau_{max}$ ) dont les bornes sont des constantes. Une telle transition ne peut être franchie que lorsqu'elle est restée sensibilisée pendant  $\tau_{min}$  et doit être franchie avant d'avoir été sensibilisée pendant  $\tau_{max}$ . Une fenêtre temporelle de franchissement est ainsi définie. La notion de jetons réservés n'existe pas et à tout moment ils peuvent être utilisés pour le franchissement d'une autre transition.

## Les réseaux de Petri à Objets

### Définition formelle

#### Notations :

Soit  $E$  un ensemble  $\{e_1, e_2, \dots, e_n, \dots\}$ , on notera :

$E^*$  l'ensemble des n-uplets d'éléments de  $E$

$\langle e_1, \dots, e_m \rangle$  ( $e_i \in E^*$ ) les n-uplets éléments de  $E^*$ ,

$\mathcal{P}(E)$  l'ensemble des parties de  $E$ .

Un réseau de Petri à Objets est défini par :

1. un ensemble fini de places noté  $P$
2. un ensemble fini de transitions noté  $T$
3. un ensemble de classes d'objets noté  $C_o$ . Chacune de ces classes est définie par une liste d'attributs, chaque attribut étant lui même typé. Cet ensemble de classes peut être organisé hiérarchiquement : une sous-classe hérite alors des attributs des classes dont elle est une spécialisation.
4. un ensemble de variables noté  $V$ ,
5. deux fonctions définissant la classe de chaque variable et de chaque place :

$$C_v : V \longrightarrow C_o$$

$$C_p : P \longrightarrow C_o^*$$

La fonction  $C_p$  fixe les classes d'objets dont les instances pourront marquer chaque place.

La fonction  $C_v$  fixe les classes d'objets dont les instances pourront être capturées par chaque variable.

6. une fonction incidence avant  $Pre : P \times T \longrightarrow \mathcal{P}(V^*)$

7. une fonction incidence avant  $Post: P \times T \longrightarrow \mathcal{P}(V^*)$
8. un marquage initial  $M_0$ , défini par la donnée d'un ensemble de n-uplets d'instances d'objets et d'une fonction de répartition qui donne pour chaque place son marquage, c'est-à-dire l'ensemble des n-uplets d'instances d'objets qu'elle contient.
9. à chaque transition peut être associée une condition de franchissement constituée par une expression booléenne formée à partir des attributs des instances d'objets capturées par les variables d'entrée de la transition. Dans cette condition de franchissement peuvent aussi apparaître des variables globales (visibles par toutes les transitions du réseau de Petri simultanément) et qui constituent les verrous des transitions.
10. à chaque transition peut aussi être associée une action. Une action est constituée d'un ensemble d'affectations de la forme :

$$var.att_i \leftarrow Expr$$

-*var* étant une variable de sortie de la transition,

-*att<sub>i</sub>* étant un attribut défini par la classe typant la variable *var*,

-*Expr* étant une expression du même type que l'attribut *att<sub>i</sub>* et dont les paramètres sont les attributs des variables d'entrée et les verrous. Notons, enfin, que les affectations constituant les actions sont exécutées simultanément.

## Règles d'évolution

Les attributs des instances d'objets d'un réseau de Petri à Objets ne peuvent être modifiées que pendant le franchissement d'une transition. De plus, seules les instances d'objets qui participent au tir d'une transition peuvent être modifiées. Une transition  $T_i$  est franchissable lorsque :

1. les places d'entrée de cette transition sont marquées,
2. il existe au moins une substitution consistante des variables par les instances d'objets contenues dans les places d'entrée de la transition (une substitution est dite consistante si elle satisfait la condition de franchissement de la transition),
3. les verrous associés à la transition sont vrais.

Le franchissement d'une transition s'effectue en trois étapes successives mais indissociables :

1. les n-uplets d'instances d'objets qui forment la substitution consistante choisie pour le tir sont retirées des places d'entrée de la transition,

2. les affectations qui constituent l'action associée à la transition sont exécutées en parallèle. Les n-uplets capturés par les variables d'entrée sont modifiés en fonction des n-uplets de variables qui apparaissent sur les arcs de sortie de la transition. Notons qu'une action peut créer ou détruire des instances d'objet dans les cas où l'ensemble des variables d'entrée de la transition est différent de l'ensemble des variables de sortie de la transition.
3. les n-uplets d'instances d'objet résultant de l'exécution de l'action sont déposés dans les places de sortie.

Globalement, le résultat du tir d'une transition se traduit par un changement du marquage.

### Modélisation d'une activité

La spécification d'une activité par réseau de Petri à objets est simple comme le montre la figure 8.7.

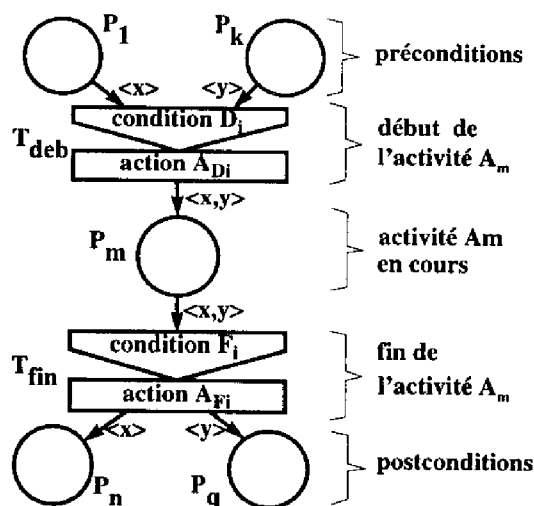


FIG. 8.7 – modélisation d'une activité par réseau de Petri

La place  $P_m$  modélise l'activité  $A_m$  en cours d'exécution : lorsqu'elle contient un jeton, une entité physique du procédé modélisée par un des objets constituant ce jeton est en train d'exécuter l'activité  $A_m$ .

Les places  $P_1$  et  $P_k$  sont appelées préconditions de l'activité  $A_m$ . Elles modélisent les états partiels dans lesquels les entités du procédé doivent se trouver pour pouvoir exécuter l'activité considérée.

De manière symétrique, les places  $P_n$  et  $P_q$  sont appelées postconditions de l'activité  $A_m$ , car elles modélisent les états partiels dans lesquels les entités qui participent à une activité se trouvent à la fin de son exécution.

Les transitions  $T_{deb}$  et  $T_{fin}$  représentent les événements début et fin d'activité. Les conditions  $D_i$  et  $F_i$  sont des conditions portant sur les attributs des objets constituant les jetons qui modélisent les entités devant exécuter l'activité  $A_m$ . Ces conditions traduisent les lois de fonctionnement de chacune de ces entités ainsi que les compatibilités devant exister entre elles pour pouvoir déclencher l'exécution de l'activité  $A_m$ .

Enfin, la modélisation par réseaux de Petri incluant la notion de n-uplet d'objets, il est très facile de représenter des liens dynamiques entre les entités physiques du procédé. On voit par exemple, apparaître explicitement sur le modèle de la figure 8.7, que les objets se substituant aux variables  $\langle x \rangle$  et  $\langle y \rangle$  sont indissociables (doublet  $\langle x, y \rangle$ ) pendant toute la durée de l'activité  $A_m$ .

### Commande et supervision des procédés discontinus : une approche hybride

Les travaux présentés s'inscrivent dans le cadre de la commande et de la supervision des procédés discontinus. Un système de commande/supervision, étudié dans le contexte particulier des systèmes de production à traitements par lots, est proposé. Notre démarche s'oriente vers une approche globale, hiérarchisée et modulaire, des aspects temps-réel de la conduite de ces procédés de fabrication, reposant sur une modélisation homogène de chaque composante du système. Les points originaux sont d'une part, une nouvelle architecture de commande et d'autre part, une modélisation séparée du procédé et de la commande, dont la nature hybride de chaque modèle obtenu est abordée selon le respect de l'identité des "mondes" discret et continu.

L'architecture proposée est organisée autour de deux réseaux locaux : un bus de terrain à caractère continu, dédié aux variables d'état continues échantillonnées, et un bus d'atelier à caractère discret, consacré aux échanges sporadiques entre les éléments du système de commande/supervision. L'interface entre le monde des variables continues et celui des événements est assurée par un générateur d'événements. Ayant mis en évidence la nécessité d'une représentation des aspects discrets et continus au niveau de la supervision, le modèle du système, constitué du modèle du procédé et du modèle de sa commande, résulte de la combinaison de deux formalismes. Les réseaux de Petri sont utilisés pour aborder l'aspect discret (événementiel) et les équations algébro-différentielles pour représenter l'aspect continu. Ce modèle est structuré selon trois vues : une vue discrète traduisant la stratégie de commande, une vue discrète correspondant à une représentation qualitative de l'état du procédé (ses configurations) et une vue continue associée à chaque configuration. L'évolution cohérente de cette structure, basée sur les événements, est garantie par une coopération des modèles dont les interactions sont clairement définies.

Deux exemples d'application sont présentés afin d'illustrer l'approche exposée. Ils insistent sur l'importance de la notion de configuration dynamique et illustrent la modélisation hybride du procédé, les interactions entre les modèles, ainsi que l'association directe d'une vue continue à des états du modèle discret de la commande.

**Mots clés :** systèmes discontinus, architecture hiérarchisée et répartie, générateur d'événements, supervision, modèle hybride.

### Control and supervision of discontinuous processes: an hybrid approach

The presented work belongs to the field of control and supervision of discontinuous processes. A control/supervision system, particularly developed for batch processes, is proposed. The work is turned towards a global approach of real-time aspects of such production systems control, in a hierarchical and modular way. Original points are on the one hand a new control architecture and on the other hand a process and control separated modeling in which hybrid aspects of each model is tackled according to the respect of the discrete and continuous "worlds" identity.

The architecture which is proposed is organised around two local area networks: a field bus of continuous nature, in charge of all the periodic data flow, and a cell network of discrete nature, dedicated to the sporadic flow of messages between the control/supervision system elements. The interface between the world of continuous state variables and that of events is ensured by an Event Generator. As the need of a model taking into account both discrete and continuous aspects at the supervisory level has been pointed out, the system model results of the combination of two formalisms. Petri nets are used to describe the discrete aspect, and algebraic-differential equations to represent the continuous one. This system model is broken down into three views: a discrete view describing the control strategy, a discrete view corresponding to a qualitative representation of the process state (its configurations), and finally a continuous view associated to each of these configurations. The consistent evolution of this structure, based on events, is assured by the models cooperation whose interactions are clearly defined.

Two application examples are developed to explain our proposals and to illustrate the main aspects of our approach.

**Keywords:** discontinuous systems, distributed and hierarchical architecture, event generator, supervision, hybrid model.