

*Ecole Doctorale de l'Université de Savoie*

*N° attribué par la bibliothèque*

/ / / / / / / / / / / / / / /

**THESE**

Pour obtenir le grade de

**DOCTEUR de L'UNIVERSITE de SAVOIE**

*Discipline : Génie Industriel*

Présentée et soutenue publiquement  
par

**Jihad REAIDY**

Le 21 juillet 2003

**Titre :**

**Etude et mise en œuvre d'une Architecture d'Agents en Réseau dans les Systèmes  
Dynamiques Situés : Pilotage des Systèmes de Production Complexes**

*Directeur de thèse : Alain HAURAT*  
*Co-directeur de thèse : Pierre MASSOTTE*

**JURY**

M. Pierre LADET	Professeur, INPG-ENSGI de Grenoble	Président
M. Jean-Paul KIEFFER	Professeur, ENSAM d'Aix-en-Provence	Rapporteur
M. Stephano CERRI	Professeur, Université de Montpellier II	Rapporteur
M. Alain HAURAT	Professeur, Université de Savoie	Directeur de thèse
M. Pierre MASSOTTE	Directeur Adj., LGI2P-EMA-Nîmes	Co-directeur de thèse
M. Alexandre DOLGUI	Professeur, Ecole des Mines de Saint Etienne	Examineur
M. Pascal PONCELET	Professeur, Ecole des Mines d'Alès	Examineur - Invité
M. Daniel DIEP	Enseignant-Chercheur, Ecole des Mines d'Alès	Examineur - Invité

*A mes Parents,*

*à mon frère Jean*

*et à Myriam ...*

## *Résumé*

L'objet de la thèse est de concevoir et développer de nouvelles approches pour la gestion des systèmes de production complexes, en incluant des notions telles que la grande variabilité des produits, les aspects dynamiques de la demande et la reconfiguration de produits ou de procédés. La mise en œuvre de ces approches requiert la conception d'une architecture spécifique permettant de créer des agents autonomes, communicants et doués d'intelligence. La structure de pilotage et de la conduite utilisée est non centralisée et hétérarchique ; elle intègre plusieurs concepts et mécanismes tels que l'auto-organisation, des protocoles de négociation à base de « coopération » (coopération et compétition) et la théorie de jeux. En terme de mise en œuvre, le pilotage du système de production distribué est effectué à travers l'allocation dynamique des ressources entre les agents « produits et ressources ».

Notre contribution concerne en premier lieu l'étude et l'analyse des structures de pilotage existantes des systèmes de production et la proposition d'approches décentralisées pour un pilotage auto-organisé entre les entités autonomes du système.

En second lieu, l'apport concerne la conception de nouveaux protocoles de négociation pour l'allocation dynamique des ressources. Ces derniers sont fondés sur un nouveau concept de fonctionnement et de communication en temps réel nommé « Session de Demande », faisant appel à de nouveaux principes utilisés en économie, et basés sur la « coopération » et la théorie des jeux.

Enfin, et en dernier lieu, notre contribution a conduit à la réalisation d'un simulateur basé sur une architecture décentralisée de pilotage. Cet outil nous a permis de tester, valider, comparer et évaluer les performances des différents concepts développés dans cette thèse. Les protocoles de négociation développés ont fait émerger des solutions réactives stables et adaptables dans des environnements dynamiques situés.

*MOTS CLES* : Pilotage et Conduite des Systèmes de Production, Système Multi-Agents, Auto-Organisation, Coopération, Protocole de Négociation.

# Table des Matières

<b>CHAPITRE 1 : INTRODUCTION</b> .....	<b>9</b>
<b>1 PREAMBULE</b> .....	<b>9</b>
<b>2 CONTEXTE – EVOLUTION DE L’ENVIRONNEMENT</b> .....	<b>9</b>
2.1 PRINCIPAUX CHALLENGES DES FUTURS SYSTEMES DE PRODUCTION.....	10
<b>3 RAPPELS ET DEFINITIONS</b> .....	<b>12</b>
<b>4 PROBLEMATIQUE</b> .....	<b>13</b>
<b>5 SYNTHESE DE LA DEMARCHE PROPOSEE</b> .....	<b>15</b>
<b>6 PLAN ET CONTRIBUTIONS DE LA THESE</b> .....	<b>16</b>
6.1 PLAN DE CE MEMOIRE.....	16
6.2 LA CONTRIBUTION DE CETTE THESE .....	16
<b>CHAPITRE 2 : ETAT DE L’ART</b> .....	<b>19</b>
<b>1 INTRODUCTION</b> .....	<b>19</b>
<b>2 LE PILOTAGE DES SYSTEMES DE PRODUCTION</b> .....	<b>19</b>
2.1 STRUCTURE DES SYSTEMES DE PILOTAGE DE PRODUCTION .....	22
2.1.1 <i>Structure centralisée</i> .....	23
2.1.2 <i>Structure non centralisée</i> .....	24
2.1.2.1 Structure hiérarchique .....	24
2.1.2.2 Structure hétérarchique .....	26
2.1.2.3 Structure hybride.....	29
2.1.2.4 Discussion.....	31
2.2 TECHNIQUES ET ALGORITHMES POUR LE PILOTAGE DES SYSTEMES DE PRODUCTION HETERARCHIQUES.....	33
2.2.1 <i>Algorithmes de « dispatching » ou de répartition</i> .....	33
2.2.2 <i>Algorithmes d’ordonnancement</i> .....	33
2.2.3 <i>Algorithme à flux tendu ou tiré « Pull Algorithm »</i> .....	35
<b>3 LES SYSTEMES MULTI-AGENTS POUR LE PILOTAGE DECENTRALISE</b> .....	<b>36</b>
3.1 POURQUOI ?.....	36
3.2 APPLICATIONS.....	38
<b>4 LES SYSTEMES MULTI-AGENTS – RAPPELS ET CONCEPTS</b> .....	<b>40</b>
4.1 INTRODUCTION.....	40
4.2 QU’EST-CE QU’UN AGENT ?.....	40
4.3 QU’EST-CE QU’UN SYSTEME MULTI-AGENT ?.....	42
4.4 AGENTS ET OBJETS.....	43
4.5 DIFFERENTES CATEGORIES ET MODELES D’AGENTS.....	44
4.5.1 <i>Agents cognitifs</i> .....	44
4.5.2 <i>Agents réactifs</i> .....	45
4.5.3 <i>Agents hybrides</i> .....	47
4.5.4 <i>Synthèse</i> .....	48

4.6	DOMAINES D'APPLICATION DES SMA.....	48
4.7	PROTOCOLES DE NEGOCIATIONS.....	49
4.7.1	<i>Les types de Protocoles de Négociation (PN)</i> .....	50
4.7.1.1	PN à base d'enchères .....	50
4.7.1.2	PN à base de médiation.....	52
4.7.1.3	PN à base des argumentations et de jeux de dialogues .....	53
4.7.1.4	PN à base de détermination de stratégies .....	54
4.7.1.5	Discussion.....	56
4.7.1.6	Quelques éléments de la théorie des jeux.....	56
4.7.2	<i>langage de communication dans les protocoles de négociations</i> .....	61
4.7.3	<i>Plateformes multi-agents</i> .....	62
<b>5</b>	<b>CONCLUSION .....</b>	<b>63</b>
<b>CHAPITRE 3 : APPROCHE DECENTRALISEE POUR LE PILOTAGE ET LA CONDUITE DES SYSTEMES DE PRODUCTION .....</b>		<b>65</b>
<b>1</b>	<b>INTRODUCTION .....</b>	<b>65</b>
<b>2</b>	<b>APPROCHE DE SOLUTIONS : LA GESTION INVERSE DE PRODUCTION.....</b>	<b>65</b>
2.1	ORDONNANCEMENT OU RECONFIGURATION ?.....	65
2.2	SOLUTION INVERSE .....	66
<b>3</b>	<b>EXEMPLE D'ARCHITECTURE DE PILOTAGE DECENTRALISE : PABADIS .....</b>	<b>69</b>
3.1	INTRODUCTION.....	69
3.2	CONTEXTE, OBJECTIFS ET PARTENAIRES.....	69
3.3	APERÇU DE L'IDEE PABADIS.....	70
3.3.1	<i>Pourquoi des agents ?</i> .....	71
3.3.2	<i>Topologie du système</i> .....	72
3.3.3	<i>le principe de Plug-and-Participate</i> .....	73
3.3.4	<i>Intégration des composants PABADIS</i> .....	74
3.4	STRUCTURE DES AGENTS.....	74
3.5	CYCLE DE PRODUCTION.....	76
3.6	ALLOCATION DES RESSOURCES .....	77
3.7	APPORTS ET PERSPECTIVES.....	78
<b>4</b>	<b>CONCLUSION .....</b>	<b>79</b>
<b>CHAPITRE 4 : NOUVEAUX CONCEPTS ET MECANISMES POUR LA GESTION DE PRODUCTION .....</b>		<b>81</b>
<b>1</b>	<b>INTRODUCTION .....</b>	<b>81</b>
<b>2</b>	<b>LES AGENTS POUR LA MODELISATION DES INTERACTIONS ENTRE LES ENTITES D'UN SYSTEME DE PRODUCTION.....</b>	<b>81</b>
<b>3</b>	<b>AGENTS POUR L'ALLOCATION DES RESSOURCES DANS LES SYSTEMES DE PRODUCTION DECENTRALISES .....</b>	<b>84</b>
3.1	ALLOCATION DES RESSOURCES A TRAVERS DES PROTOCOLES DE NEGOCIATIONS.....	85
3.2	HORIZON DE LA DECISION DE L' «AGENT -PRODUIT» .....	86
3.3	CATEGORIE DE LA DECISION DE L' «AGENT -PRODUIT» .....	87
<b>4</b>	<b>CONCEPTS SPECIFIQUES .....</b>	<b>88</b>
4.1	COMMENTAIRES LIES AU CADRE DE L'ETUDE.....	89
4.2	DISCUSSION.....	89

<b>5</b>	<b>CONCEPT DE «SESSION DE DEMANDE» .....</b>	<b>92</b>
<b>6</b>	<b>COOPETITION OU COMPERATION ?.....</b>	<b>93</b>
<b>7</b>	<b>UN EXEMPLE D'APPROCHE POUR LA MODELISATION DU SYSTEME : PABADIS</b> <b>.....</b>	<b>96</b>
7.1	ARCHITECTURE DE L'«AGENT-PRODUIT ».....	97
7.2	AGENT DE « SESSION DE DEMANDE » (ASD) OU AGENT COORDINATEUR.....	98
7.3	CYCLE DE PRODUCTION.....	99
<b>8</b>	<b>METHODOLOGIE DE NEGOCIATION.....</b>	<b>101</b>
<b>9</b>	<b>PROTOCOLES DE NEGOCIATION DEVELOPPES .....</b>	<b>103</b>
9.1	PROTOCOLE «PRODUCTION RESERVATION » (PR).....	104
9.2	PROTOCOLE « SINGLE-STEP PRODUCTION RESERVATION » (SSPR) .....	107
9.3	PROTOCOLE « CURRENCY ».....	109
9.3.1	<i>Protocole de l' «Agent-Produit ».....</i>	<i>109</i>
9.3.2	<i>Protocole de l' « Agent Ressource ».....</i>	<i>112</i>
9.4	PROTOCOLE « CONSENSUEL SELON LA PRIORITE ».....	113
9.5	PROTOCOLE « CONSENSUEL AVEC CONTRAINTES».....	114
9.6	PROTOCOLE « COOPETITION » LOCALE.....	120
9.7	PROTOCOLE « COOPETITION GLOBALE OU CENTRALISEE ».....	126
9.8	CONCLUSION.....	130
	<b>CHAPITRE 5 : APPLICATION ET RESULTATS .....</b>	<b>131</b>
<b>1</b>	<b>INTRODUCTION .....</b>	<b>131</b>
<b>2</b>	<b>CONCEPTION DU DEMONSTRATEUR EXPERIMENTAL.....</b>	<b>131</b>
2.1	DESCRIPTION DU DEMONSTRATEUR.....	132
2.2	MODELE GENERIQUE.....	133
2.3	CONCEPTION DES EXPERIENCES.....	135
2.4	PLAN DE TEST - LES « BENCHMARKS » .....	136
<b>3</b>	<b>ELABORATION DU PROTOTYPE DU TEST.....</b>	<b>137</b>
3.1	OUTILS ET ENVIRONNEMENT INFORMATIQUES DE BASE.....	137
3.2	PLATE-FORME D'AGENT MOBILE : GRASSHOPPER.....	138
3.2.1	<i>Qu'est ce que Grasshopper ?.....</i>	<i>138</i>
3.2.2	<i>Composantes de la plate-forme .....</i>	<i>138</i>
3.2.3	<i>Aperçu technique de Grasshopper.....</i>	<i>140</i>
3.3	UTILISATION DE L'OUTIL DE TEST .....	141
<b>4</b>	<b>TESTS, RESULTATS, COMPARAISONS ET SYNTHESES .....</b>	<b>143</b>
4.1	ETUDE DU PARAMETRE « TEMPS DE FABRICATION » (TF) .....	145
4.2	ETUDE DU PARAMETRE « TEMPS D'ATTENTE » (TA).....	146
4.3	ETUDE DU PARAMETRE « TEMPS DE RETARD » (TR) .....	147
4.4	ETUDE DU PARAMETRE « TAUX D'UTILISATION DU SYSTEME » (TUS) .....	148
4.5	ETUDE DU PARAMETRE « LONGUEUR DES FILES D'ATTENTE » (LFA).....	149
4.6	ETUDE DU PARAMETRE « NOMBRE DES PRODUITS EN RETARD » (NPR).....	150
4.7	ANALYSE STATISTIQUE MULTIVARIEE DES RESULTATS.....	151
4.7.1	<i>Rappel sur l'analyse en composantes principales.....</i>	<i>151</i>
4.7.2	<i>Discussion .....</i>	<i>152</i>
4.8	CONCLUSION.....	155
	<b>CHAPITRE 6 : CONCLUS IONS ET PERSPECTIVES .....</b>	<b>157</b>

<b>1</b>	<b>SYNTHESE</b> .....	<b>157</b>
1.1	DISCUSSION GENERALE.....	158
<b>2</b>	<b>PERSPECTIVES ET TRAVAUX FUTURS A MENER</b> .....	<b>160</b>
	<b>REFERENCE</b> .....	<b>163</b>
	<b>GLOSSAIRE</b> .....	<b>179</b>
	<b>ANNEXES</b> .....	<b>181</b>

## Liste des Figures

Figure 2.1 : Modèle de référence ISO.....	20
Figure 2. 2 : Structure centralisée.....	23
Figure 2.3 : Structure hiérarchique .....	24
Figure 2.4 : Structure hétérarchique .....	26
Figure 2.5 : « Cellular Manufacturing System » .....	27
Figure 2.6 : Structure hybride .....	29
Figure 2.7 : Modèle d'un agent cognitif.....	44
Figure 2.8 : Modèle d'un agent réactif.....	45
Figure 2.9 : Structures des modèles d'agents.....	46
Figure 2.10 : Modèle d'agents InteRRap.....	47
Figure 2.11 : Protocole de négociation avec des offres non alternatives entre deux agents... 53	
Figure 3.1 : Approches conventionnelles pour les systèmes de production .....	67
Figure 3.2 : Approches mises en œuvre pour la gestion d'un système complexe .....	68
Figure 3.3 : Structure conventionnelle et structure PABADIS.....	71
Figure 3.4 : Topologie d'un système de production PABADIS .....	73
Figure 3.5 : Communication avec le service de Look Up.....	74
Figure 3.6 : Modèle structurel d'un Agent .....	75
Figure 3.7 : Hiérarchie des classes d'agents dans PABADIS.....	76
Figure 3.8 : Cycle de production d'un ordre de fabrication dans PABADIS .....	77
Figure 4.1 : Interactions entre les différentes entités d'un système de production .....	82
Figure 4.2 : Interactions entre produits et ressources .....	91
Figure 4.3 : Interactions entre produits et ressources par l'intermédiaire d'un agent coordinateur .....	92
Figure 4.4 : Prise de décision des «Agent-Produit» dans une « Session de Demande ».....	92
Figure 4.5 : Catégories de négociations et de décisions utilisée par l'«Agent-Produit » .....	97
Figure 4.6 : Schéma global des interactions entre les différentes entités du système .....	99
Figure 4.7 : Modélisation expérimentale d'un atelier de production.....	100
Figure 4.8 : Calcul du Temps de traitement permis ( $T_{p_{i+1}}$ ) pour la prochaine tâche.....	102
Figure 4.9 : Exemple pour le modèle « Production Reservation » (PR).....	105
Figure 4.10 : Algorithme du modèle « Production Reservation » (PR).....	106
Figure 4.11 : Algorithme du modèle « Single-Step Production Reservation » (SSPR).....	108
Figure 4.12 : Algorithme du modèle « Currency» utilisé par l' «Agent-Produit » .....	111
Figure 4.13 : Algorithme du modèle « Currency» utilisé par l'«Agent-Ressource».....	112
Figure 4.14 : Prise de décision d'un APZ par concession selon la priorité .....	114
Figure 4.15 : Prise de décision consensuelle d'un APZ selon la contrainte de temps de traitement permis .....	116
Figure 4.16 : Prise de décision consensuelle d'un APZ respectant la contrainte de temps de traitement permis .....	118
Figure 4.17 : Prise de décision consensuelle d'un APZ ne respectant pas la contrainte de temps de traitement permis .....	119
Figure 4.18 : Prise de décision d'un APZ dans la Session de Demande.....	125
Figure 4.19 : Approche centralisée pour la prise de décision des agents dans un système de production.....	126
Figure 4.20 : Influence et profit d'une population d'individu sur la prise de décision d'un individu. ....	127
Figure 4.21 : Prise de décision d'un APZ par l'intermédiaire d'une approche centralisée..	129
Figure 5.1 : Structure physique du démonstrateur du Laboratoire LGI2P-EMA.....	132

Figure 5.2 : Structure du démonstrateur expérimental théorique. ....	134
Figure 5.3 : Structure hiérarchique des composantes dans Grasshopper .....	139
Figure 5.4 : Interface utilisateur pour la création des jeux d'expériences.....	141
Figure 5.5 : Interface utilisateur pour le lancement de la simulation. ....	142
Figure 5.6 : Capture d'écran pour la simulation du modèle GT associé à l'expérience « Experiment 3 ». ....	142
Figure 5.7 : Histogramme de la moyenne des temps de fabrication des produits. ....	145
Figure 5.8 : Histogramme de la moyenne de temps d'attente des produits. ....	146
Figure 5.9 : Histogramme de la moyenne des temps de retard des produits. ....	147
Figure 5.10 : Histogramme de la moyenne de taux d'utilisation du système.....	148
Figure 5.11 : Histogramme de la moyenne des files d'attente dans le système.....	149
Figure 5.12 : Histogramme de nombre des produits en retard.....	151
Figure 5.13 : Analyse en Composantes Principales. Représentation des deux axes principaux F1 et F2. ....	153
Figure 5.14 : Analyse en Composantes Principales. Représentation des deux axes principaux F1 et F3. ....	154

## Liste des Tableaux et des Algorithmes

Tableau 2.1 : Exemples des projets utilisant la technologie agent pour le pilotage des systèmes de production .....	39
Tableau 2.2 : Dilemme du prisonnier .....	59
Tableau 2.3 : Liste non exhaustive des plate-formes multi-agent .....	62
Tableau 4.1 : Evolution des concepts, dans notre équipe de recherche, à travers les interactions entre les différentes composantes d'un système de production .....	82
Tableau 4.2 : Comparaison entre des travaux de recherche existant utilisant les protocoles de négociation pour le pilotage des systèmes de production. ....	87
Tableau 4.3 : Différentes stratégies d'interactions entre les entités du marché .....	94
Tableau 4.4 : Représentation de la matrice des retard attendu par les «Agent-Produit» pour les vecteurs de stratégies choisis durant le jeu. ....	122
Tableau 5.1 : Combinaisons des benchmarks utilisés dans le système.....	137
Tableau 5.2 : Equivalence terminologique entre Grasshopper et PABADIS.....	139
Tableau 5.3 : Moyenne et écart type des temps de fabrication des produits .....	145
Tableau 5.4 : Moyenne et écart type des temps d'attente des produits.....	146
Tableau 5.5 : Moyenne et écart type des temps de retard des produits.....	147
Tableau 5.6 : Moyenne et écart type de taux d'utilisation du système.....	148
Tableau 5.7 : Moyenne et écart type des files d'attente dans le système.....	149
Tableau 5.8 : Nombre de produits en retard dans le système.....	150
Tableau 5.9 : Pourcentage d'échec des produits respectant leur date d'échéance.....	150
Algorithme 4.1 : Protocole « Production Reservation » (PR) .....	104
Algorithme 4.2 : Protocole « Single-Step Production Reservation » (SSPR) .....	107
Algorithme 4.3 : Protocole de l'« Agent-Produit » selon le principe «Currency».....	110
Algorithme 4.4 : Allocation consensuelle selon la priorité .....	113
Algorithme 4.5-a : Consensuel avec Contraintes .....	115
Algorithme 4.5-b : Consensuel avec Contraintes.....	116
Algorithme 4.6 : « Coopétition » locale .....	124
Algorithme 4.7 : « Coopétition » globale .....	128



# Chapitre 1 : Introduction

## 1 Préambule

L'objet de cette thèse est d'étudier le pilotage et la conduite des systèmes de production décentralisés dans un environnement dynamique situé. Gérer les interactions et la prise de la décision entre les composantes autonomes du système, pour obtenir une meilleure cohérence et coordination, dans un tel environnement est un problème complexe. Ce contexte s'inscrit dans le cadre des missions générales de notre centre de recherche LGI2P- Ecole des Mines d'Alès. Par "complexe", nous désignons des systèmes au niveau organique, structurel et évolutionniste, comprenant des entités ou agents en interaction structurelle non-linéaire et réalisant des activités en vue d'atteindre un objectif donné. Cette complexité est de plusieurs types [Massotte, 1995] :

- Structurelle : la structure, l'organisation, le nombre d'entités et la nature des liens sont pris en compte ensemble, ce qui rend difficile la modélisation et le traitement de l'information associée.
- Algorithmique : la complexité est ici intrinsèque et s'adresse à des problèmes difficilement appréhendables et compréhensibles. Dans ce cas, il faut faire appel à des méthodes particulières de résolution (par exemple heuristiques).
- Comportementale : dans ce point de vue, l'évolution du système, bien que faisant appel à des modèles de représentation et à des algorithmes simples, est imprédictible et se traduit, par l'émergence spontanée de formes et des comportements complexes (par exemple : chaos déterministe).

Dans le cadre de cette thèse, nous partons du principe que ces types de complexité peuvent se présenter ensemble ou séparément. Nous les intégrerons donc implicitement dans notre approche. Cette dernière se doit d'être novatrice, en effet, les approches conventionnelles ne sont pas actuellement en mesure de couvrir cet aspect des choses de façon satisfaisante.

## 2 Contexte – Evolution de l'environnement

L'évolution du monde industriel est continue. Elle est fonction des besoins croissants et variés des clients, et doit répondre à l'augmentation de la compétitivité inhérente à la mise en œuvre de l'économie globale et de la Net Economie. Désormais, le modèle des Systèmes de Production (SP) centralisés n'a plus la place stratégique qu'il occupait dans les anciens contextes industriels car il doit pouvoir supporter un processus de fabrication flexible, réactif et économique. La flexibilité dont il est question ici, se traduit par une facilité de réadaptation et de ré-affectation des fabrications variant dans le temps, et ceci de manière compétitive. Ainsi, les

systèmes de production correspondants doivent être associés à des systèmes de pilotage adaptables et autonomes. Ces derniers doivent rester ouverts, c'est-à-dire évolutifs et capables de supporter de manière dynamique une reconfiguration ou un réajustement des paramètres opérationnels tant au niveau du produit, du procédé que du marché. Pour mémoire et afin de limiter le champs du problème, cette reconfiguration concernera avant tout le procédé ou le système de production et non le produit lui-même. Dans ce qui suit, nous allons identifier les défis pour les systèmes de production à venir.

## **2.1 Principaux challenges des futurs systèmes de production**

Dans cette partie, nous détaillons les tendances et caractéristiques nécessaires pour les futurs systèmes de production avancés. Ces tendances permettent aux organisations manufacturières d'être plus flexibles en management, et en travail pratique, mais aussi en développement, production, réutilisation et/ou rénovation. Ils peuvent être identifiés comme suit [Massotte et al., 2002] :

- **Ingénierie** : avec la progression des capacités de calcul et des technologies d'information ou des télécommunications, la conception, le processus de fabrication et l'utilisation des produits et des services ont évolué. De même, dans les années récentes, l'impact du principe Ordinateur-Intégré pour la production ou Computer-Integrated Manufacturing (CIM) est passé de la philosophie traditionnelle d'intégration d'usine à une philosophie de gestion d'usine virtuelle [Lee, 1996]. Ces progrès technologiques permettent de réaliser un environnement collaborateur et rapide basé sur i) les outils multimédia de génie logiciel et les systèmes de communication très fiables, pour permettre des procédures distribuées de conception et de développement concourants des produits et des services, et ii) les opérations dans les systèmes de production distribués.
- **Environnement et écologie** : le recyclage consiste à arranger et à réutiliser des matériaux. Ils peuvent être fournis par les produits retournés (équivalents à nouveau, réparés, reconstitués ou reconstruits, etc.). Comparée à la conception bien connue pour la fabrication ou « design for manufacturing », cette approche couvre beaucoup plus de contraintes. En effet, elle signifie la réutilisation des pièces et des produits assemblés, pour l'entretien, pour le démantèlement et/ou le démontage faciles, et pour la longévité (avec de nouveaux concepts tels que l'entretien et la réparation automatique, le rétablissement d'erreur, etc.). Ces items sont tous inclus dans le programme de GNOSIS-VF (Esprit#28448) [Massotte et al., 2001c ; Diep et al., 2001].
- **Processus et gestion de chaînes d'approvisionnements** : l'industrie est aujourd'hui confrontée à une économie globale et distribuée. La concurrence et les fusions mondiales réduisent les marges entre les industries. Ceci a un impact important, non seulement sur la conception du produit et des processus de fabrication, mais également sur les systèmes sociaux et

logistiques : comment améliorer l'approvisionnement et le transport des matériaux et des pièces ? Comment partager la fabrication des produits finaux complexes ? Comment organiser les récupérations, le gaspillage et la réutilisation des produits et des marchandises ? Comment développer de nouvelles opportunités pour l'évolution du travail ? Comment tenir compte de la valeur ajoutée de l'économie pour la nouvelle logistique et trouver l'équilibre le plus adapté pour le pilotage et la gestion des perturbations résultantes en termes de flux des produits ?

- **Évolution des besoins de la production :** la réduction des coûts de fabrication et l'amélioration de la satisfaction du client amènent un changement de paradigme partant de la 'Qualité de la conception et de la fabrication de produit' vers la 'qualité du service' avec un délai de livraison décroissant. En conséquence, la réactivité et la flexibilité des systèmes de production deviennent la clé principale des systèmes de production. Cette approche est étroitement liée au «Lean production» pour la minimisation des coûts résultant de la variabilité de produit, en particulier les temps requis pour changer la taille des inventaires par le principe du Juste-à-Temps (Just-In-Time) [Flik, 1995]. Dans la plupart des cas, on a démontré que la taille des lots a dû être réduite autant que possible pour augmenter le nombre de variantes de produits pour pouvoir répondre aux exigences des clients dans chaque secteur du marché.
- **Evolution des besoins des clients :** Actuellement, la place donnée au client progresse dans la gestion de la production et devient, au fur et à mesure, prioritaire. Les exigences du client tirent la production vers de nouveaux concepts de fabrication réactifs, adaptatifs et évolutifs. Dans ce contexte, de nouveaux principes de production ont émergé, comme la réduction de cycle de vie d'un produit et la production personnalisée . Cette dernière devient le fondement principal de ces nouvelles approches, dans lesquelles les notions de prix et de temps de livraison ou «Dynamic Pricing» deviennent plus flexibles, modulables et adaptables aux besoins du client. De plus, la tendance à produire un seul type ou «One-of-a-Kind-Production» (OKP) représenté par les principes de l'«Agile Manufacturing» devient indispensable pour les futurs systèmes de production. En fin de compte, la réduction du cycle de vie du produit diminue la durée de son introduction dans le marché, ce qui donne plus de dynamique et de variété de choix pour répondre aux attentes et souhaits des clients.
- **E-Business :** le Net-Economie est caractérisée par deux notions : la rapidité et la réactivité des processus. Le E-Business rend plus perceptible l'exacerbation des besoins des clients. Dès qu'une demande se manifeste, il faut indiquer rapidement quand, à quel coût, dans quelle quantité la commande sera fournie. Ici, le temps de réponse est primordial. D'ailleurs, en raison du volume élevé d'individus et de demandes spécifiques, comportant beaucoup de variations de produits, la «mass customization» est exigée pour

de tels systèmes de production. Ces caractéristiques combinées ensemble impliquent une grande réactivité.

### 3 Rappels et définitions

Nous précisons ici quelques termes qui vont être utilisés couramment dans le cadre de cette thèse.

**Système de production :** l'Institut National de la Statistique et des Etudes Economiques (INSEE) définit la production comme : "l'activité économique socialement organisée consistant à créer des biens et des services s'échangeant habituellement sur le marché et / ou obtenus à partir de facteurs de production s'échangeant sur le marché". La production est donc une activité qui crée ou ajoute de la valeur. Seront étudiés dans le cadre de la thèse les systèmes de production proprement dits, qui sont des assemblages des entités (produits, ressources, clients, producteurs, équipements, fournisseurs, etc.), sollicités de manière coordonnée en vue d'assurer la production sous des contraintes données.

**Auto-organisation :** L'auto-organisation caractérise tout processus dans lequel des structures émergent au niveau collectif (ou plus généralement apparition d'une structure à l'échelle N+1 à partir d'une dynamique définie à échelle N), à partir de la multitude d'interactions entre individus, sans être codées explicitement au niveau individuel [Bonabeau et Theraulaz, 1994]. L'objectif de l'auto-organisation est de permettre l'évolution dynamique et autonome d'un système existant, en fonction du contexte de façon à en assurer la viabilité. Elle permet aux entités composant le système de s'adapter à leur environnement soit par spécialisation des fonctions (apprentissage), soit par modification de la topologie du groupe et des interactions correspondantes. Ceci donne naissance à un nouveau modèle d'organisation.

**Coopétition :** Ce terme est la contraction de *coopération* et *compétition*. Il a été introduit par Nalebuff et Brandenburger pour montrer comment la théorie des jeux peut s'appliquer au monde des affaires ou « business » [Nalebuff et Brandenburger, 1996]. Pour ces deux auteurs « ce principe est une nouvelle façon de réfléchir et de voir le monde des affaires d'aujourd'hui. Certains le voient comme une concurrence totale. Ils pensent que faire du business, c'est comme déclarer la guerre et ils assument qu'ils ne peuvent pas gagner sans que les autres perdent. D'autres le voient comme une coopération totale de groupes et de partenaires. Mais le business est la coopération et la compétition en même temps. » La clé principale de ce principe est d'assimiler le business à un « jeu » dans lequel nous pouvons spécifier les aspects de *coopération* et de *compétition*. Une fois cela fait, l'utilisation des principes de la théorie des jeux permettra de changer et d'adapter le jeu selon nos convenances et nos besoins.

**Environnement dynamique situé :** Nous appelons *environnement dynamique* un milieu dont les propriétés physiques ou structurelles sont susceptibles d'être modifiées immédiatement à tout moment. Un milieu dynamique se modifie par l'évolution des interactions entre les entités agissant dans cet environnement (actions sur leur milieu) ou par sa propre dynamique interne (évolution de ses composants). Il est dit situé s'il est considéré comme un sous-environnement local de l'environnement global ayant ses propres caractéristiques, tout en gardant la relation d'influence réciproque avec les autres sous-environnements. La dynamique de l'environnement concerne toutes les parties qui le composent. Les interactions des entités sont considérées comme situées ou locales si elles prennent seulement en considération les caractéristiques de leur propre sous-environnement.

## 4 Problématique

Actuellement, les systèmes de production existants ne permettent pas de répondre aux besoins, exigences, évolutions et aux nombreux challenges du monde industriel à venir. Les approches traditionnelles de pilotage sont souvent centralisées et hiérarchiques, ce qui rend la conduite et le pilotage du système rigide et peu adaptable. Ces systèmes sont conçus seulement pour des environnements respectant des conditions nominales ou des perturbations limitées. Leurs comportements deviennent insatisfaisants dans les environnements dynamiques et ouverts. Les inconvénients majeurs de ces approches peuvent être résumés ainsi :

- toute perturbation locale dans le système remet en cause tout le processus de décision au niveau global,
- l'amélioration des performances du système exige une remontée d'informations et des traitements lourds à réaliser,
- les explosions combinatoires liées au grand nombre de contraintes et paramètres nécessitent des algorithmes complexes et difficiles à mettre en œuvre.

Dans la littérature, différentes approches ont été proposées pour résoudre ce problème avec des résultats variables en termes d'efficacité, d'efficience et de généralité. Par exemple, au niveau organique du pilotage, nous pouvons distinguer des approches hiérarchiques modifiées, hétérarchiques (ou décentralisées) et hybrides. Dans ces approches, différentes activités opérationnelles pouvant être utilisées tels que la planification, l'ordonnancement et l'allocation des ressources. Toutefois, dans un environnement dynamique soumis à de fortes perturbations, le choix de l'architecture de pilotage, des activités opérationnelles, des mécanismes et des technologies compatibles pour l'obtention des meilleures performances reste la tâche la plus ardue à réaliser. Ce problème est considéré dans la littérature comme étant un problème Np-dur [Blackstone et al., 1982 ; Brucker, 1995].

Ainsi, le problème sera posé à trois niveaux :

1. quelle est la stratégie à utiliser et les objectifs à poursuivre pour aboutir à des changements structurels, culturels, fonctionnels et comportementaux à travers de nouvelles approches non centralisées et non hiérarchiques ?
2. comment mettre en œuvre de nouveaux mécanismes, concepts et méthodes pour l'auto-organisation et la reconfiguration automatique de systèmes en faisant appel à des protocoles et des mécanismes de négociation entre les entités autonomes du système ?
3. comment mettre en œuvre ces concepts à travers de nouvelles technologies adaptées et flexibles comme les agents ?

Dans cette thèse, nous avons choisi d'utiliser une approche hétérarchique et décentralisée pour répondre, même partiellement, aux différents besoins et exigences actuelles et futures du pilotage des systèmes industriels. Cette approche apparaît comme la plus adaptée au pilotage des systèmes de production dans les environnements dynamiques situés, grâce à sa réactivité et son adaptabilité dans un tel environnement. Elle fait appel à des concepts issus du domaine de l'intelligence collective et de l'E-business comme l'auto-organisation, la coopération et la compétition entre les composantes ou les entités autonomes du système.

La gestion des interactions entre les différentes entités pour obtenir une meilleure cohérence et coordination, est effectuée dans le but d'améliorer les performances du système. C'est le plus grand défi pour la réussite de cette approche. Cette dernière nécessite le développement de nouveaux concepts pour auto-organiser, adapter et faire évoluer les interactions vers plus ou moins de linéarité et vers plus ou moins de satisfaction. De plus, ils nécessitent de nouveaux mécanismes pour le pilotage fonctionnel du système. Actuellement, la planification et l'ordonnancement sont les deux principes de base utilisés pour les activités opérationnelles du pilotage du système. Dans un environnement dynamique trop fortement perturbé, l'utilisation de ces principes conventionnels peut se révéler illusoire. En effet, la différence entre le déroulement réel requis par la demande et l'ordonnancement préalable est alors trop importante pour faire l'objet d'ajustements, et pour obtenir une solution consistant en des réactualisations permanentes dans un temps raisonnable. Ce fonctionnement montre plusieurs limites face aux impératifs liés au besoin de réactivité, rapidité et de flexibilité, et est extrêmement lourd et difficile à gérer. Pour cela, des méthodes d'allocation dynamique, et en temps réel, des ressources peuvent apporter des solutions à ce type de problème. L'auto-organisation du pilotage qui en résulte requiert l'utilisation de nouveaux principes basés sur la coopération, la compétition et la « coopétition » ; ils sont fondés sur des règles locales ou des protocoles de négociation locaux que nous détaillerons.

Finalement, la mise en œuvre de ces concepts et de ces mécanismes exige des techniques adaptables, modulables et décentralisées. Sur ce dernier point, la

technologie « agents » a été appliquée dans la mise en œuvre de telles approches décentralisées. Actuellement, les interactions entre les agents sont souvent réalisées à travers des protocoles de négociation simples et à base d'enchères ; les performances de ces derniers restent instables face aux perturbations du système. Pour cela, la conception de nouveaux protocoles de négociation utilisant d'autres mécanismes s'avère indispensable pour améliorer la stabilité des résultats de l'approche décentralisée.

Dans ce qui suit, nous détaillerons la démarche suivie tout au long de ce travail de recherche pour résoudre les différents problèmes cités ci-dessus et nous présenterons des résultats qui ont été obtenus à partir de simulateurs représentatifs de cas industriels concrets.

## **5 Synthèse de la démarche proposée**

En résumé de cette thèse, la démarche qui s'impose et qui a été explorée peut se résumer selon les différentes étapes suivantes :

- pour comprendre les différents domaines concernés dans le cadre la thèse une revue détaillée, une synthèse et des comparaisons ont été faits sur les travaux existants. Cela a été réalisé au niveau des structures de pilotage des systèmes de production, des protocoles de négociation dans les systèmes multi-agents et de l'utilisation de ces protocoles pour l'allocation des ressources dans les systèmes décentralisés,
- pour avoir une structure de pilotage des systèmes de production plus réactive et flexible dans un environnement dynamique situé, nous avons choisi et utilisé l'approche hétéraçhique et décentralisée. Le pilotage est réalisé à travers l'allocation dynamique de ressources. Il est effectué entre entités autonomes « produits et ressources » pour faire face aux différentes perturbations dans le système. La prise de décision de ces entités est faite à base d'informations en temps réel fournies par un nouveau principe d'anticipation, développé dans la thèse, nommé « session de demande »,
- pour gérer les interactions entre les entités autonomes dans le système, de nouveaux protocoles de négociation ont été conçus. Ces derniers reposent sur la détermination de stratégies et l'anticipation des sessions de demande, tout en utilisant des mécanismes tels que la compétition, la coopération, la coopération et la théorie des jeux,
- pour mettre en œuvre ces différents concepts et mécanismes nous avons utilisé la technologie d'agent à travers des agents hybrides (réactif et cognitif).
- Finalement, pour tester, valider et comparer ces différentes approches dans un environnement dynamique imprévu et à forte perturbation, nous avons développé un simulateur représentatif du projet Européen PABADIS.

## **6 Plan et contributions de la thèse**

### **6.1 Plan de ce mémoire**

Ce plan découle de la démarche proposée précédemment. Après l'introduction du contexte, de la problématique, de la démarche et des contributions de la thèse, le chapitre 2, sur l'état de l'art abordera les différentes approches de pilotage et de conduite des systèmes de production. Ces dernières concernent les différentes structures organiques et approches fonctionnelles du pilotage. De plus, une revue détaillée des protocoles de négociation utilisés dans les systèmes multi-agents, sera faite.

Dans le chapitre 3, nous proposerons des solutions à base de reconfiguration automatique de ressource et de ce que nous appelons « gestion inverse » de la production. En outre, nous détaillerons les différents concepts et composantes de l'architecture décentralisée du projet européen PABADIS.

Au chapitre 4, une analyse et une critique des perspectives, horizons et types de décision entre les entités produits et ressources pour l'allocation des ressources en temps réel dans les structures hétérarchiques est présentée. Ensuite, nous développerons et présenterons des nouveaux concepts et mécanismes pour la gestion de production dans un système dynamique situé à forte perturbation. L'approche de la coopération entre les entités autonomes du système sera appliquée ; elle sera intégrée dans des protocoles de négociation à base de l'anticipation à un nouveau concept, nommé « session de demande », et aux principes de la théorie des jeux.

Le chapitre 5 sera consacré à l'application de ces différents concepts à travers un simulateur développé dans le cadre de la thèse. Les tests, résultats, comparaisons et synthèses seront ensuite étudiés dans cette partie.

En conclusion, nous discuterons des perspectives d'avenir de ce travail.

### **6.2 La contribution de cette thèse**

La contribution de notre travail de recherche concerne principalement les points suivants :

*I. Une revue détaillée et un état de l'art des différents points existants :*

- des structures de pilotages existantes telles que les structures centralisées, hiérarchiques, hétérarchiques et hybrides,
- types de protocoles de négociation utilisés entre entités autonomes ou agents,

- perspectives, horizons et types de décision entre les entités produits et ressources pour l'allocation des ressources en temps réel dans les structures hétérarchiques. Des comparaisons et des analyses sur les avantages et inconvénients des travaux existants dans ce domaine permettront de proposer des éléments de solutions.

## *II. Les concepts de fonctionnement :*

- l'étude et l'analyse des principes de **compétition, coopération, coopétition** et la **compération**, utilisés dans l'économie, dans un **contexte industriel** (allocation dynamique des ressources), ont permis l'exploration et la définition de nouveaux concepts,
- la conception de nouvelles approches à base des principes de la coopération et de la coopétition pour le pilotage **auto-organisé** et hétérarchique du système de production,
- le développement de nouveaux **mécanismes d'interactions** entre les entités autonomes basées sur des protocoles de négociation, enchères et la théorie des jeux selon des approches compétitives, coopératives et coopétitives.

## *III. La communication :*

- la conception du principe de la **session de demande** offre une information en temps réel sur les entités en compétition dans un système.
- la mise en œuvre de la **technologie agent** pour le pilotage décentralisé du système. Ces agents sont conçus selon une approche hybride,
- le développement de nouveaux **protocoles de négociation** entre agents à base d'anticipation à des sessions de demande et de la théorie des jeux,

## *IV. L'application et réalisation :*

- le développement d'un **simulateur** basé sur une architecture hétérarchique afin de **tester, valider et comparer les performances** des différents concepts développés dans ce travail,
- une analyse et une synthèse des résultats obtenus, accompagnées d'une **perspective** d'avenir sont réalisées en fin de la thèse.



# Chapitre 2 : Etat de l'art

## 1 Introduction

Ce chapitre présente un état de l'art sur le pilotage et la conduite des systèmes de production (SP), basée de préférence sur l'utilisation de la technologie d'agent. Il se décompose en deux parties.

Dans la première partie, nous détaillons d'abord les différentes structures de pilotage des systèmes de production existant avec leurs avantages et leurs inconvénients. Ces structures peuvent être classées en centralisées et non centralisées. Les structures non centralisées comportent les structures hiérarchiques, hétérarchiques (ou décentralisées) et hybrides. Ensuite, nous détaillons les différentes techniques et algorithmes utilisables pour le pilotage des structures hétérarchiques, comme le dispatching, l'ordonnancement et les algorithmes à flux tendus ou « Pull algorithms ».

La technologie d'agent est prise en compte dans la seconde partie de ce chapitre comme nécessaire pour le pilotage et la conduite des systèmes de productions. Elle est détaillée à travers la définition du concept d'agent, la présentation des différentes architectures d'agent (réactive, cognitive et hybride), ses domaines d'application et les différents types de protocoles de négociation utilisés entre agents. Concernant les protocoles de négociation, nous nous focalisons sur un type de négociation basé sur la théorie des jeux pour la détermination des stratégies à utiliser entre agents. Ensuite, quelques éléments de la théorie des jeux seront rappelés.

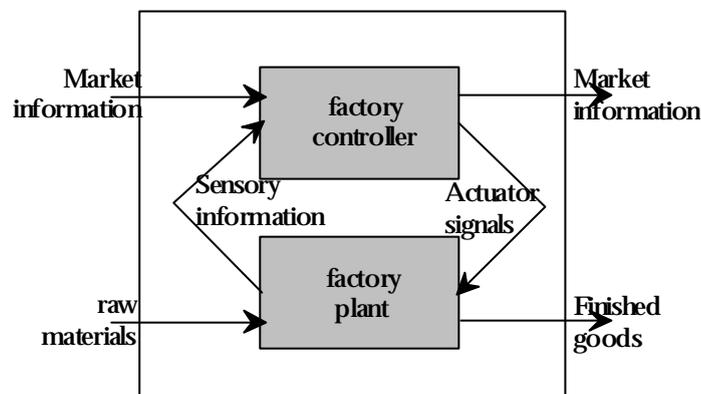
Finalement, des remarques et des choix sont établis dans la conclusion.

## 2 Le pilotage des systèmes de production

Selon le Grand Robert de la langue française et le dictionnaire encyclopédie Hachette, piloter réfère à «*conduire, diriger, gouverner et guider tout en induisant de plus la notion de difficulté et de contingence*» [Pujo et kieffer, 2002]. A l'heure actuelle, il n'existe pas de définition unique ni stable du terme pilotage ou « Control ». L'Organisation Internationale de Normalisation « International Standard Organization » (ISO) propose pour le pilotage la définition suivante : "Factory control is defined as the actuation of a plant to make products, using the present and past observed state of the plant and demand from the market" (le pilotage des systèmes de production est défini comme l'activation du système pour fabriquer des produits, en utilisant le présent et les états passés observés et les demandes à partir du marché.) [ISO, 1986 ; Baker, 1998] voir également [Broissin, 1999] et [Ounnar, 1999] pour plusieurs définitions similaires. Pour Trentesaux et al., « *le*

*pilotage consiste à décider dynamiquement des consignes pertinentes à donner à un système soumis à perturbation pour atteindre un objectif donné décrit en termes de maîtrise de performances. La notion de maîtrise intègre non seulement celle de maintien d'un niveau de performance donné, mais également celle de progrès (évolution vers un niveau de performance souhaité ou avec une amélioration continue) » [Trentesaux et al., 2000b ; Berrah, 1997]. Dans la thèse, nous admettons cette dernière définition pour le terme pilotage et conduite des systèmes de production.*

Il existe de nombreux modèles pour décrire l'environnement du système de pilotage des Systèmes de Production (SP) [ISO, 1986 ; Lemoigne, 1994 ; Landaux, 1996]. Ces modèles représentent l'organisation des inter-relations entre le système physique (appelé système opérant ou « factory plant ») et un système de décision (appelé système de pilotage ou « factory controller »). Parmi ceux-ci, le modèle de référence traditionnel issu de ISO est le plus répandu [Baker, 1998], (voir figure 2.1).



**Figure 2.1 : Modèle de référence ISO**

Les typologies des systèmes de pilotage peuvent être considérées selon des vues fonctionnelle, organique et évolutionniste. La vue fonctionnelle décrit *ce que fait et doit faire* un système. Les concepts liés à ce point de vue sont essentiellement ceux de fonction, tâche, activité et processus. Dans le cadre du pilotage des systèmes manufacturiers, la vue fonctionnelle comporte les activités opérationnelles suivantes : la planification, l'ordonnancement, le cheminement des produits et l'allocation des ressources [Dilts, 1991].

La vue organique décrit *qui fait quoi comment et avec qui*. Cette vue fait référence à un ensemble de sous-systèmes fonctionnels composant un système donné. Les concepts relatifs à la vue organique d'un système sont principalement ceux d'agencement, de structure et d'organisation. Deux types de systèmes de pilotage peuvent être mis en évidence au niveau organisationnel : centralisé et non centralisé. La distinction entre les deux types tient à l'unicité ou la multiplicité de systèmes capables d'exercer un pouvoir de décision [Mintzberg, 1982].

La vue évolutionniste d'un système traduit son évolution, c'est-à-dire les changements opérés dans ce système sur une échelle de temps différente, plus étendue, que celle relative aux vues fonctionnelle et organique. Les concepts relatifs à la vue évolutionniste d'un système sont essentiellement ceux d'évolution, d'adaptativité et d'auto-organisation. Cette vue se base sur l'hypothèse qu'un système de pilotage évolue entre plusieurs zones de fonctionnement. Au sein zone de fonctionnement, les rétroactions de pilotage sont nécessairement « négatives ». Elles permettent au système de pilotage de maintenir sa structure ou certains de ses paramètres dans des intervalles restreints (homéostasie). Les rétroactions « positives » conduisent, elles, à perturber et à faire évoluer un système de pilotage d'une zone de fonctionnement à une autre (morphogenèse) [Massotte, 1999]. Homéostasie et morphogenèse sont les deux notions qui permettent d'établir la typologie évolutionniste.

Dans la thèse, différentes structures organisationnelles pour le pilotage d'un système de production sont présentées. Toutefois, nous porterons notre attention sur la dispersion des approches et des fonctions de pilotage, de façon à faire émerger des ordres ou configurations globales des systèmes de production. Ainsi, l'allocation locale des ressources est choisie au niveau opérationnel de pilotage du système de production. L'auto-organisation du système sera effectuée à travers des interactions et négociations entre des agents autonomes représentant les différentes entités du système. Dans le paragraphe suivant nous allons détailler les différentes structures de pilotage existants dans de tels systèmes de production.

## **2.1 Structure des systèmes de pilotage de production**

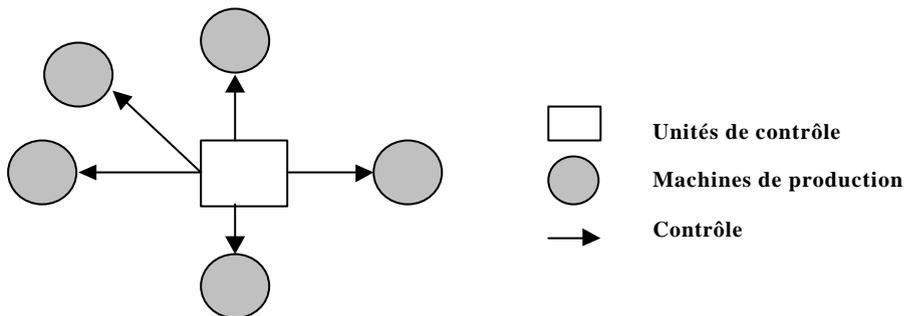
Les systèmes de pilotage de production peuvent être classés en fonction de leur structure. Divers travaux de recherche ont analysé l'évolution des différentes structures existantes, leurs avantages et leurs inconvénients. De nouvelles architectures ont été proposées pour améliorer les performances des applications industrielles existantes et répondre aux besoins des futurs systèmes de production.

Duffie et Piper ont présenté une série d'architecture de contrôleurs centralisés, hiérarchiques utilisant l'ordonnancement dynamique et totalement distribué ou hétérarchique avec des composantes intelligentes [Duffie et Piper, 1987]. Lin et Solberg ont montré quatre paradigmes de pilotage de la production : information centralisée – aide à la décision centralisée, information distribuée – aide à la décision centralisée, information centralisée – aide à la décision distribuée et information distribuée – aide à la décision distribuée [Lin et Solberg, 1991]. Dilts a proposé une vue d'ensemble de l'évolution des structures de pilotage existantes de la structure centralisée, à hiérarchique, jusqu'au contrôle hétérarchique. Il a mis en évidence les caractéristiques, les avantages et les inconvénients de chaque structure [Dilts et al., 1991]. Pels a, quant à lui, souligné l'influence et l'importance de l'architecture d'un système pour la flexibilité de son pilotage et son contrôle. Les structures hiérarchiques, hétérarchiques, modulaires, holonique, et à base d'agent sont détaillés dans son travail [Pels et al., 1997]. Kim a distingué trois types d'architecture hiérarchique, hétérarchique et hybride. Il en a signalé les limites et les problèmes principaux, pour les deux premières architectures. Il a en conséquence proposé une architecture hybride basée sur le concept holonique comme solution à ces différents problèmes [Kim, 2002]. Finalement, Pujo et Kieffer ont dressé un panorama des principales architectures possibles pour le pilotage des systèmes de production. Ils ont distingué les structures centralisée, hiérarchisée, coordonnée, distribuée, décentralisée et distribuée supervisée [Pujo et Kieffer, 2002].

En fin de compte, différentes classifications peuvent être envisageables pour les structures de pilotages comme centralisée et non centralisée, hiérarchique et non hiérarchique, etc. Dans la thèse, nous considérons que les structures de pilotage peuvent être classées en centralisées et non centralisées. Les structures non centralisées comprennent des structures hiérarchiques, hétérarchiques ou hybrides. L'architecture hiérarchique comprend des structures « hiérarchisées » et « hiérarchisées modifiées ». L'architecture hétérarchique comprend des structures décentralisées et distribuées. L'architecture hybride inclut en même temps les deux structures hiérarchique et hétérarchique. Dans les paragraphes suivants, nous citerons quelques travaux utilisant ces différentes architectures, tout en signalant les avantages et les inconvénients de chacune d'entre elles.

### 2.1.1 Structure centralisée

La structure centralisée est proposée par un nombre limité de chercheurs [Achatz et Parrish,1987 ; Hammer, 1987 ; Titli,1979]. Dans cette structure, une unité de contrôle, ou entité, pilote toutes les machines de production et détient le pouvoir de décision; elle maintient l'information globale de toutes les activités des entités dans le système. Cette unité gère la production, traite en temps réel les événements qui surviennent et réalise la synchronisation et la coordination de toutes les tâches (voir figure 2. 2).



**Figure 2. 2 : Structure centralisée**

#### *Avantages et inconvénients*

Les avantages de cette architecture peuvent être résumés ainsi :

- facilité d'accéder à la base de données globale et complète de l'information, (information unique ; cohérence du système d'information),
- nombre limité d'unités de contrôle, ou de moyens de traitement et de gestion de l'information),
- optimisation globale possible, en effet, l'information de l'état global de système peut être facilement référencée et extraite.

Cependant, nous pouvons relever plusieurs inconvénients :

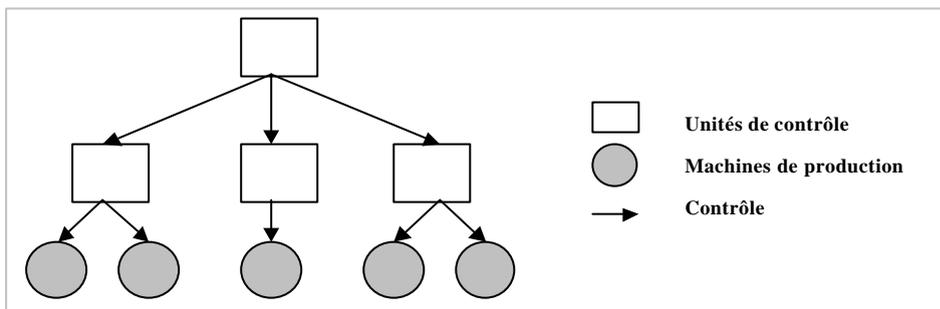
- la vitesse de réponse décroît lorsque le système grandit,
- le système présente une grande vulnérabilité aux pannes, un petit problème pouvant provoquer l'arrêt total,
- la difficulté d'appliquer des modifications au niveau du logiciel utilisé à cause de la non modularité,
- l'accès au système d'information est compliqué : une seule entité doit être capable d'appréhender rapidement et de manière pertinente un grand nombre d'informations et de contraintes.

Pour palier aux inconvénients de l'architecture centralisée, les chercheurs ont élaboré le concept de non centralisation de la décision. Celui-ci s'est manifesté à travers plusieurs types d'architectures où le contrôle de la décision peut être hiérarchique, hétérarchique (ou décentralisé), ou hybride.

## 2.1.2 Structure non centralisée

### 2.1.2.1 Structure hiérarchique

La présence naturelle de la hiérarchie dans la société et les structures des systèmes complexes ont mené les chercheurs à concevoir des architectures hiérarchiques. La hiérarchie établie est utilisée pour la structuration et pour le contrôle du système. Cette structure définit une relation maître-esclave entre les niveaux supérieurs et inférieurs du contrôle. Chaque niveau coordonne les unités de contrôle du niveau inférieur, et ce jusqu'au niveau le plus bas (voir figure 2.3). Chaque niveau a des relations de dépendance vis-à-vis du niveau supérieur, et de domination vis-à-vis du niveau inférieur [Pujo et Kieffer, 2002]. Les décisions sont prises souvent par l'unité de contrôle centrale.



**Figure 2.3 : Structure hiérarchique**

Dans cette partie, figurent des travaux de recherches qui ont contribué au développement et aux modifications apportées à la structure hiérarchique.

Jones, Maclean et Jackson ont ainsi présenté un modèle de contrôle hiérarchique pour les systèmes manufacturiers automatisés [Jones et Maclean, 1986 ; Jackson et Jones, 1987]. L'objectif poursuivi par leur travail est de limiter la taille, la complexité, et la fonctionnalité des modules de contrôle individuels dans les structures hiérarchiques. Leur modèle fonctionne avec les cinq couches suivantes : usine, atelier, cellule, poste de travail et équipement (facility, shop, cell, workstation and equipment). Chaque module décompose la commande d'entrée provenant du superviseur en des sous-tâches simples, les assigne aux modules subordonnés appropriés, gère leurs exécutions et fournit enfin l'état du « Feed-back » au superviseur. Celui-ci dispose d'un superviseur et de plusieurs subordonnés, et aucune communication directe entre les modules de même niveau n'existe. L'Institut National de Standard et Technologie (NIST) appliquait le premier l'architecture AMFR (Automated Manufacturing Research Facility) décrit ci-dessus par Jones et al.

Chryssolouris et al., en conformité avec les architectures standardisées de type CIM (Computed Integrated Manufacturing), ont décrit le modèle MADEMA (Manufacturing Decision Making). Ce dernier comporte quatre niveaux de hiérarchie : usine, atelier, « work center » et ressources (factory, job-shop, work center and ressource) [Chryssolouris et al., 1984 ; Chryssolouris et al., 1988]. Le

premier niveau représente l'usine toute entière et contrôle la capacité d'entrée des demandes dans l'usine. Le niveau atelier comporte les « work center » et assigne le travail à ces différents groupements. Le niveau « work center » représente le groupement des ressources de production. Le dernier niveau se rapporte aux unités de ressource de production. MADEMA reçoit les demandes de fabrication ( type, quantité, date d'échéance, etc.) du niveau atelier, détermine les alternatives envisageables des paires tâche-ressource, les critères appropriés, leurs conséquences avec des critères multiples, les règles d'aide à la décision et il choisit enfin la meilleure alternative.

Boulet et al., ont comparé les deux architectures de contrôle hiérarchique décrites ci-dessus et concluent que le modèle proposé par Jones et al. est le modèle le plus flexible [Boulet et al., 1991]. Par contre, le modèle MADEMA permet de meilleures implémentations pratiques et complètes de ce modèle dans l'industrie. Les deux modèles manquent toutefois de réactivité et des bonnes performances en temps réel face aux événements imprévus.

Les structures hiérarchiques récentes peuvent être représentées par de nouvelles structures nommées hiérarchiques modifiées ou « modified hierarchical control ». Elles permettent la communication et la coordination entre les entités de même niveau hiérarchique. Dans cette catégorie figurent par exemple « Manufacturing Systems Integration » (MSI) [Senehi et al.,1994], « Production Activity Control » (PAC++) [Andersson, 1997] et « Factory Activity Control » (FACT) [Arentsen, 1995].

Après cette revue de ces quelques travaux de recherche utilisant la structure hiérarchisée pour le pilotage des systèmes de production, nous pouvons en dégager les avantages et les inconvénients.

#### *Avantages et inconvénients*

La structure hiérarchique a été adoptée de façon quasi-systématique dans les systèmes de taille importante, jusque dans les années 1980. Les avantages principaux de cette structure peuvent être résumés ainsi :

- facilité de compréhension,
- conformité par rapport à la résolution classique des problèmes,
- rapidité d'obtention des réponses due au couplage maître/esclave entre les entités,
- optimisation globale.

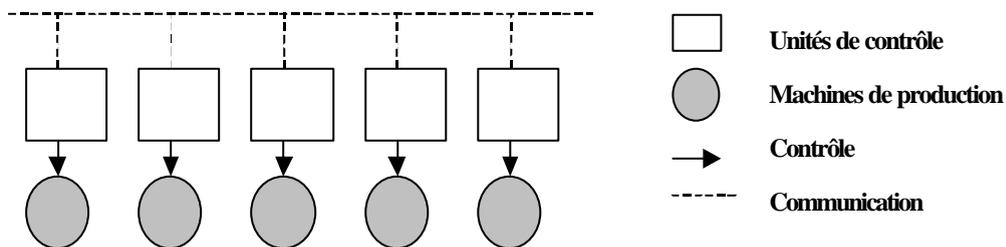
Les méthodes d'optimisation ont été appliquées intensivement, et avec succès dans les approches traditionnelles hiérarchiques. Cependant, n'ayant pas pris en compte tous les aspects, notamment l'incertitude et la complexité du système réel, ces approches ne sont pas adéquates pour la modélisation exacte des environnements distribués et complexes des systèmes actuels de production [Kim, 2002].

La plupart des architectures hiérarchiques exigent une structure fixe durant le fonctionnement du système et assument le comportement déterministe des composants. Ces rigidités engendrent les inconvénients principaux des architectures hiérarchiques, qui peuvent être résumés de la manière suivante :

- la difficulté d'ajouter, de modifier ou de supprimer des ressources. En effet, pour effectuer une modification, il faut arrêter le système, mettre à jour les structures de données relatives aux hauts niveaux dans la structure [Brussel et al., 1998],
- la difficulté dans le design ou conception de la structure, chaque contrôleur considère toutes les situations possibles des composantes de niveaux inférieurs à lui [Dilts et al., 1991],
- toute perturbation imprévue, comme la panne d'une ressource, rend la planification et l'ordonnancement pour le contrôleur de niveau élevé invalide [Duffie et Prabhu, 1994],
- la panne du contrôleur central de haut niveau exige l'arrêt total du système [Krothapalli et Deshmukh, 1999].

### 2.1.2.2 Structure hétérarchique

Le substantif hétérarchie (en anglais, Heterarchy) et son adjectif hétérarchique (en anglais, heterarchical) constituent des néologismes à la fois en français et en anglais. Ce terme est formé à partir de deux termes grecs : *heteros* (autre) et *Arckhein* (commander), signifiant ainsi à l'origine «*commandement par les autres* ». L'hétérarchie renvoie à l'idée d'acteurs différents qui assument en collégialité la coordination d'une action collective donnée et s'oppose par essence au terme *hiérarchie* [Trentesaux, 2002]. La structure hétérarchique est appelée aussi structure décentralisée. Dans cette structure il n'existe pas d'unité de contrôle de niveau hiérarchique supérieur pour coordonner l'ensemble des unités (voir figure 2.4).



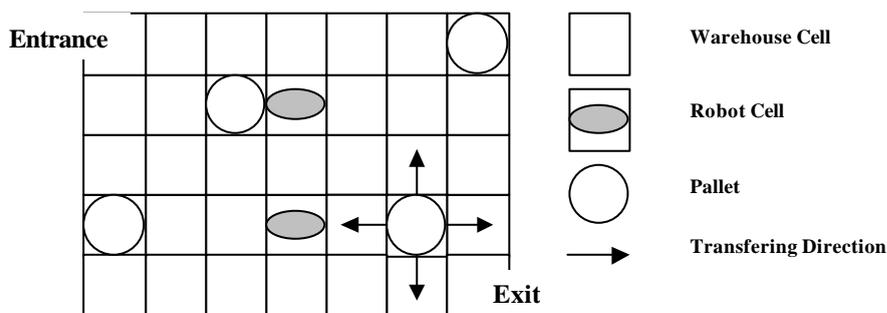
**Figure 2.4 : Structure hétérarchique**

Les unités de contrôle sont multiples et en interaction ; elles peuvent alors s'auto-organiser pour assurer des tâches globalement cohérentes. Ces unités disposent des propriétés suivantes [Cho, 1993] :

1. égalité du droit d'accès aux ressources,
2. accès et accessibilité mutuelle entre elles,
3. mode de fonctionnement indépendant,
4. conformité stricte aux règles et protocoles du système global.

Divers travaux ont étudié les structures décentralisées dans les systèmes de productions. Leurs contributions se situent au niveau du développement de l'architecture, de la conception des protocoles de négociation entre les entités et d'applications industrielles. Hatvany suggère l'utilisation des architectures hétérarchique et coopérative comme alternatives à l'architecture hiérarchique [Hatvany, 1985]. Cet auteur précise la conception des règles de comportement, des buts locaux, et des buts globaux (que les entités autonomes suivent), pour empêcher l'anarchie et le chaos dans le système.

Shaw décrit une structure de contrôle distribué pour l'ordonnancement dynamique dans un « Cellular Manufacturing System » (CMS) [Shaw,1987]. L'architecture d'un CMS se compose de trois types de « cell » (Warehouse cell, pallet et robot cell) (voir figure 2.5) ; chaque « cell » agit indépendamment en échangeant des messages. Chaque contrôleur de « cell » maintient ces informations locales, mais il n'y a aucun contrôle global. L'affectation des tâches est effectuée dynamiquement par la négociation entre les contrôleurs des « cells ». L'ordonnancement des tâches dans chaque « cell » est fait localement. Des travaux récents établis par Kondoh et al. ont proposé une structure hétérarchique pour le CMS [Kondoh et al. 1999 ; Kondoh et al. 2000 ; Tomiyama, 1997 ; Massotte et al., 2001b]. L'auteur considère le principe CSM comme un prototype rapide dans la conception et outil d'aide à la décision pour la configuration et l'affectation des tâches aux niveaux des ressources et des produits.



**Figure 2.5 : « Cellular Manufacturing System »**

Les configurations et les affectations sont déterminées par des mécanismes d'auto-organisation entre les différentes entités au niveau opérationnel. Cette évolution vers plus d'autonomie des unités de production, conduit naturellement à considérer cette approche comme un cadre d'application des systèmes de pilotage hétérarchique [Chan et al., 1999].

La coordination entre les entités indépendantes d'une structure hétérarchique est un point essentiel à évoquer pour le pilotage d'un système. Nous pouvons l'assurer au travers des règles prédéfinies (de manière centralisée ou non), nous pouvons aussi laisser «ouverte» cette coordination. Généralement, c'est le cas de notre approche et de celle mise en œuvre dans les structures en réseau, les agents utilisent des protocoles de communication basés sur des principes de marché d'enchères «market paradigm bidding» pour satisfaire leurs objectifs. L'exemple canonique de protocole de négociation type est le réseau contractuel ou «Contract Net Protocol» (CNP) développé par Smith [Smith, 1980 ; Smith et Davis, 1981]. Il a été largement utilisé par les différents travaux utilisant des architectures hétérarchiques. Il a été étendu à travers différents mécanismes d'enchères [ FIPA, 2000 ; Liu, 2002] et protocoles de négociation [Lin et Solberg, 1991 ; Krothapalli et Deshmukh, 1999].

Lin et Solberg utilisent une architecture hétérarchique pour le pilotage d'un atelier de production. Ils considèrent les produits et les ressources comme des agents. Chaque agent négocie avec les autres agents, en temps réel, à travers le principe de marché d'échange pour satisfaire ses objectifs individuels [Lin et Solberg, 1992]. Lorsqu'un client demande un service fourni par une organisation, un coût en monnaie d'échange est exigé par l'organisation. Ce modèle utilise un mécanisme de construction générique des offres d'échange durant la négociation entre les agents, en se basant sur le principe de la combinaison du prix et de l'objectif (temps, coût, qualité, etc.). Lin et Solberg ont montré à travers des simulations que l'architecture proposée montre beaucoup de flexibilité et d'adaptabilité au niveau du pilotage en temps réel d'un atelier de production. Ce travail a inspiré à plusieurs auteurs de nouveaux protocoles de négociation. [Saad et al., 1997, Krothapalli, et Deshmukh, 1997 ; Krothapalli, et Deshmukh, 1998 ; Usher et Wang, 2000a].

Parunak et al., et Baker et al. ont décrit une architecture décentralisée pour le pilotage d'un atelier de production [Parunak et al., 1998 ; Baker et al., 1997]. Ressources, manager, type de produits et unité des traitements sont modélisés par des entités intelligentes ou agents intelligents qui connaissent « la façon de combiner produits et ressources pour la fabrication d'autres produits ». Dans cette structure, les auteurs fournissent un mécanisme de dialogue direct entre les clients et l'atelier de production, pour la « mass customization », en utilisant la technologie d'agents intelligents.

Finalement, le projet Européen PABADIS « Plant Automation Based on Distributed System » constitue un exemple récent d'architecture hétérarchique conçu pour des applications industrielles. PABADIS utilise une organisation décentralisée pour la reconfiguration automatique et dynamique des chaînes de production [PABADIS, 2000]. Il vise à améliorer le pilotage d'un système de production décentralisé en utilisant la notion de "plug-and-participate" et la suppression totale ou partielle des tâches de planification et d'ordonnancement. Les composants de base dans PABADIS sont des agents et des services. Les agents et les services fonctionnent en coopération afin d'accomplir les tâches à exécuter. L'architecture de PABADIS sera détaillée dans le chapitre 3 de la thèse.

### Avantage et inconvénients :

Les avantages principaux des structures hétérarchiques peuvent être résumés dans les points suivants :

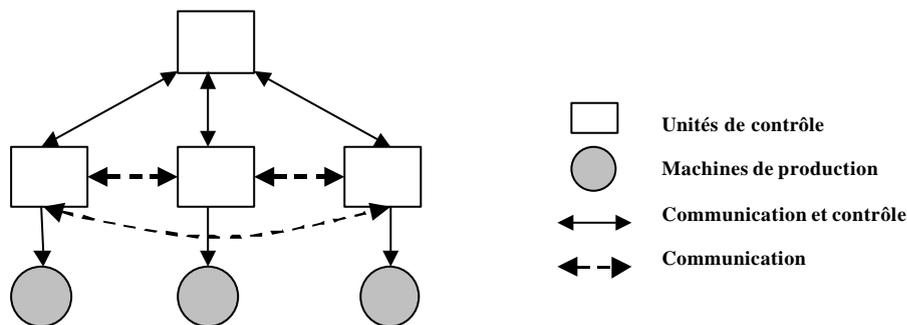
- complexité réduite des logiciels,
- tolérance aux fautes améliorées,
- maintenance, modification, reconfiguration et intervention humaine faciles,
- connaissance facile des caractéristiques des parties.

La décentralisation de la décision dans les structures hétérarchiques présente aussi quelques inconvénients. Nous pouvons souligner que :

- la prédiction des performances globales du système et du comportement individuel des entités est difficile, sinon impossible à réaliser [Duffie et Prabhu, 1996 ; Massotte et al., 2001d],
- les performances globales du système sont extrêmement sensibles aux définitions et au choix des règles locales et des protocoles de négociation entre les entités [Dilts et al., 1991]. Ceci est dû aux interactions qui rendent le Système sensible aux Conditions Initiales (SCI) et le font rapidement converger dans un bassin d'attraction difficilement pré-déterminable.

### 2.1.2.3 Structure hybride

Les deux structures hiérarchique et hétérarchique présentent des avantages et des inconvénients pour le pilotage des SP. Certains travaux de recherche ont essayé de préserver les avantages des deux structures en proposant une nouvelle structure hybride en combinant les deux autres (voir figure 2.6).



**Figure 2.6 : Structure hybride**

Dans la structure hybride, les unités de contrôle de même niveau hiérarchique sont interconnectées via un même moyen de contrôle. Elles sont capables de communiquer et de coopérer pour satisfaire leurs objectifs locaux. Lors de perturbations, l'ensemble des unités de contrôle peut demander de l'aide à leur moyen de contrôle pour résoudre les problèmes détectés.

Tawegoum et al. présentent une architecture hybride de pilotage pour un atelier de production flexible [Tawegoum et al., 1994]. Dans leur modèle, les unités de contrôle requièrent l'aide aux unités de contrôle de niveau plus élevé en cas de non-respect de leurs objectifs locaux à cause de perturbations inattendues.

Ottaway et Burns proposent un système adaptatif de pilotage de la production « adaptive production control system » (APCS), dans lequel la transition entre la structure hétérarchique et hiérarchique se produit dynamiquement et est basé sur la charge de travail de système [Ottaway et Burns, 1997 ; Ottaway et Burns, 2000]. Dans leurs modèles, ces auteurs considèrent les trois types d'agents : agents tâches, ressources et superviseurs. Chaque agent a une connaissance au niveau de la coordination, production, interface et moteur d'inférence. Quand un agent constate que la ressource qu'il représente n'est pas utilisée correctement, il demande un agent de surveillance pour le pilotage de la ressource. Ainsi un niveau de hiérarchie sera introduit dynamiquement dans le système.

Maturana et al. proposent MetaMorph, une architecture hybride à base d'agent pour le pilotage des systèmes de production distribués [Maturana,1997 ; Maturana et al., 1999]. Ils emploient deux types d'agents : agent ressource pour représenter les ressources physiques, et agent médiateur pour la coordination entre les agents ressources. Shen et al. étendent l'architecture de MetaMorph pour intégrer des activités d'entreprise avec ceux de ses fournisseurs, partenaires et clients dans leur projet de MetaMorph II [Shen et al., 1998]. Ils emploient des médiateurs hiérarchiques et des mécanismes de négociation coopérative entre les agents ressources.

Parmi les structures hybrides récentes figurent les Systèmes Holoniques de Production « Holonic Manufacturing System » (HMS). Les principes de base des systèmes holoniques furent introduits en 1967 par Arthur Koestler dans un livre intitulé "The Ghost in the Machine " [Koestler, 1967]. Koestler introduisit l'idée qu'un petit nombre de principes essentiels suffit à expliquer la capacité qu'ont les systèmes sociaux et biologiques à s'autoréguler. il proposa le terme « holon » pour décrire l'élément de base de ces systèmes. Ce mot combine la racine grecque « holos » qui signifie « entier », avec le suffixe « on »,« partie », comme dans proton ou neutron.

HMS apparaît comme un compromis recherché entre des organisations intégrées ou hiérarchiques, en tant qu'ensemble cohérent au sein d'un système, et des organisations distribuées ou hétérarchiques, en tant que parties réactives à l'environnement de ce système. Cette interprétation récente [Valckenaers et al., 1994a ; Valckenaers et al., 1994b ; Brussel et al., 1995] pour les systèmes de production du concept a pour objet de satisfaire les critères d'adaptabilité requis par la nouvelle génération de systèmes de production (Next Generation Manufacturing System) [Kurihara et al., 1996], proposée de façon générale par l'initiative I.M.S. (Intelligent Manufacturing Systems) initiée par le Japon en 1989 [IMS, 1994 ; 1997]. La structure « holarchique » qui résulte de l'agencement de ces holons de production

peut être considérée comme le compromis entre une structure hiérarchique et une structure hétérarchique dans la mesure où la coopération des holons « intelligents » de bas niveau reste néanmoins coordonnée par des holons hiérarchiquement supérieurs.

Brussel et al., Valckenaers et al. et Wyns présentent l'architecture PROSA (Product-Ressource-Order-Staff Architecture) comme une structure holonique pour le pilotage de la production [Brussel et al., 1998 ; Wyns, 1999].

Parmi les exemples d'architecture holonique se trouve l'architecture PROSA (Product-Ressource-Order-Staff Architecture) pour le pilotage de la production développée par le groupe de recherche de l'Université Catholique de Louvain [Brussel et al., 1998 ; Valckenaers et al., 1999 ; Wyns, 1999]. PROSA est un système holonique de production (HMS) ayant pour but d'atteindre la stabilité malgré les perturbations, la flexibilité, l'adaptabilité au changement, et l'utilisation efficace des ressources. Cette architecture comporte trois types d'holons de base : ordre, produit, et ressource, et un « staff » holon. Chacun des holons de base est responsable respectivement de la logistique, de la planification technologique (planification de processus y compris), et de la détermination des capacités de la ressource. Le « staff » holon est considéré comme un expert externe qui prodigue des conseils aux holons de base. Il peut fournir des algorithmes centralisés pour l'ordonnancement, et aider les holons de base. En incluant des « staff » holon dans la structure, le système dispose d'un comportement de pilotage hiérarchique pouvant améliorer les performances globales du système.

#### *Avantages et inconvénients :*

Grâce à la combinaison des deux structures hiérarchique et hétérarchique, la structure hybride peut profiter des avantages des deux structures en même temps. Comme ça, elle combine la robustesse contre des perturbations, à travers les interactions locales entre les agents hétérarchiques, et l'optimisation globale et la prédiction à travers des agents hiérarchiques de supervision. L'inconvénient majeur de cette structure est la grande difficulté à trouver le bon compromis entre la supervision au niveau hiérarchique et le degré d'autonomie attribué aux niveaux hétérarchiques. Le compromis cherché permet la stabilité et l'adaptation au changement, (dynamique, perturbation, chaos, etc.), dans un environnement complexe.

#### *2.1.2.4 Discussion*

Après avoir examiné les caractéristiques des les travaux de recherche portant sur ces organisations, ainsi que les avantages et inconvénients des différentes structures de pilotage existantes, nous constatons qu'il n'existe pas de modèle idéal de pilotage qui puisse être utilisé à tout moment et dans tous les environnements. Chaque structure peut être efficace pour certains types de problèmes et

d'environnements, suivant le contexte, les contraintes dynamiques et temporelles, etc.

Notre travail portera sur les mécanismes d'auto-organisation entre entités autonomes à travers des protocoles de négociation et des mécanismes de coopération (coopétition = compétition + coopération). Ces approches seront appliquées pour l'allocation dynamique des ressources, dans un environnement dynamique et situé, entre les entités produits et ressources d'un atelier de production, considérées localement.

L'architecture la plus adaptée à ces concepts est l'architecture hétérarchique. Ses caractéristiques propres nous ont permis d'appliquer et de valider différents concepts et mécanismes avec plus de liberté d'interaction et de simplicité de modélisation. Ces concepts ont été élaborés pour mettre en jeu des interactions entre entités autonomes dans des environnements dynamiques situés. Toutefois, nous pouvons aussi appliquer ces concepts avec des architectures hybrides de pilotage, et avec un degré élevé d'autonomie des unités de production.

Comme exemple pratique d'architecture hétérarchique nous avons choisi celle du projet européen PABADIS, dont notre centre de recherche LGI2P-EMA est partenaire. Cette architecture nous a permis de tester et de valider les concepts développés (voir chapitre 3 et chapitre 4).

Dans les prochains paragraphes nous allons expliquer les techniques et algorithmes pouvant être utilisés pour le pilotage des systèmes de production hétérarchiques.

## **2.2 Techniques et algorithmes pour le pilotage des systèmes de production hétérarchiques**

Une revue rapide des algorithmes et des concepts existants et pouvant être appliqués pour le pilotage, au niveau fonctionnel, des structures hétérarchiques est établie. Les algorithmes sont utilisés pour décider du type de produit et de la quantité à produire, pour prévoir la durée de la production, comment et quand utiliser les ressources disponibles, quand et quel type de tâche à introduire dans le système et comment effectuer le cheminement des différents flux, etc. Trois types d'algorithmes de base sont étudiés [Baker, 1998] : les algorithmes de dispatching, les algorithmes d'ordonnement et les algorithmes à flux tiré (Pull algorithms).

### 2.2.1 Algorithmes de « dispatching » ou de répartition

Un algorithme de dispatching décide comment utiliser une ressource dans un système en tenant en compte de la disponibilité de cette ressource, des autres ressources voisines et les tâches introduites dans le système [Baker, 1998]. Les algorithmes de dispatching sont généralement à base de règles de production : « dispatching rules ». Elles permettent de répartir les prochaines tâches à effectuer sur les ressources disponibles. Plusieurs auteurs ont étudié les différents types des règles de dispatching [Panwalker et Iskander, 1977 ; Blackstone et al., 1982; Wisner, 1993 ; Baker, 1998]. La décision séquentielle dans ces règles peut être basée sur les notions suivantes : date d'échéance, priorité de la demande ou de la tâche, temps de « setup » des ressources, temps de traitement le plus court (Shortest Processing Time), etc. Ces règles sont largement utilisées dans l'industrie en raison de leur simplicité, efficacité, et leur nature d'algorithme on-line [Holthaus, 1997]. Elles permettent l'ordonnement instantané au niveau local des systèmes de production. Les systèmes de production flexibles et distribués, et les systèmes d'assemblage sont les meilleurs candidats pour l'utilisation de ce type d'algorithmes. Comme exemple d'algorithmes de dispatching nous pouvons citer les travaux récents faits par [Saad et al., 1997 ; Pesenti et al., 2001] pour le pilotage décentralisé à base d'algorithmes de dispatching et des systèmes multi-agents.

### 2.2.2 Algorithmes d'ordonnement

Les algorithmes d'ordonnement déterminent à l'avance l'allocation des tâches sur les ressources. Ils sont utilisés normalement dans les industries « deterministic » pour des problèmes statiques parce qu'ils permettent la réduction de l'aspect aléatoire d'allocation des tâches sur les ressources en réduisant les « inventories » et le « lead-time ». L'ordonnement devient un problème complexe à modéliser et à résoudre, en particulier quand il est effectué dans un environnement ouvert et dynamique. Les problèmes d'ordonnement dans les systèmes de production décentralisée souvent de nature dynamique. Ainsi, les ordres de fabrication peuvent

survenir à n'importe quel moment, la durée de la fabrication et le temps de service ne sont pas connus avec précision, les machines peuvent devenir indisponibles et des ressources additionnelles peuvent être introduites dans le système, le temps de traitement des tâches peut subir des variations inattendues, en effet, une tâche peut prendre plus ou moins de temps que prévu. En raison de ces aspects fortement combinatoires et de leur intérêt pratique pour des systèmes de production, le problème d'ordonnancement a été largement étudié dans la littérature par diverses méthodes et techniques alternatives (voir annexe A).

Nous pouvons différencier les algorithmes d'ordonnancement selon le degré d'optimalité de leurs résultats. Nous distinguons alors quatre catégories d'algorithmes d'ordonnancement [Baker, 1998] :

- « Optimal Scheduling » est utilisé pour les problèmes simples, combinatoires et à programmation mathématique [Lenstra et Kan, 1978],
- « Nearly Optimal Scheduling » fonctionne pour les problèmes à forte complexité, dits NP-complets par l'intermédiaire des algorithmes de probabilité : numerical probabilistic algorithms, Monte Carlo probabilistic algorithms, Las Vegas probabilistic algorithms, Sherwood probabilistic algorithms. [Brassard et Bratley, 1988] ,
- « Towards Optimal scheduling » est utilisé pour optimiser un coût métrique comme une fonction d'évaluation et ne garanti pas des solutions proches des solutions optimales. Ce type d'ordonnancement utilise des technique comme le recuit simulé « Simulated annealing », les algorithmes génétiques « genetic algorithms » et l'ordonnancement neuronal « neuro scheduling » [Johnson et al., 91 ; Goldberg, 94 ; Rabelo et Alptekin, 1989],
- « Heuristic Scheduling » emploie des techniques fournissant de bonnes performances, mais sans toutefois que son placement dans une zone d'optimalité ou même d'évolution vers une solution optimale soit démontré. Un coût métrique est souvent inclus dans l'ensemble d'heuristique et sert à motiver leur utilisation. Les méthodes heuristiques sont largement utilisées dans l'industrie parce qu'elles garantissent une solution aux problèmes dans un délai d'exécution raisonnable et sans être affectées par la taille et la complexité du problème. Comme exemple nous pouvons citer : le « Forward/Backward Scheduling » , le « deterministic simulation » et l' « Intelligent Scheduling » [Wright, 1983 ; MacFarland et H. Grant, 1987 ; Kusiak et Chen, 1988 ].

La mise en œuvre de ces algorithmes d'ordonnancement dans les systèmes de pilotage hétéroarchique est généralement à base heuristique : « Forward / Backward Scheduling » [Baker, 1988 ;1991], « deterministic simulation »[Duffie et Prabhu, 1994] et l' « Intelligent Scheduling » [Sycara et al., 1991 ; Zweben et Fox, 1994].

### 2.2.3 Algorithme à flux tendu ou tiré « Pull Algorithm »

Dans les modèles à flux tendus, la production est dictée par les besoins, les usines produisent uniquement le strict nécessaire, afin de réduire au maximum les gaspillages. Nous nous plaçons ici dans le contexte essentiel de la production à la demande. C'est le principe initialement utilisé dans le domaine du e-business qui est étendu au contexte de l'économie globale et mondialisé. Il a été repris par un certain nombre de grandes entreprises sous le terme « Production on Demand » et dépasse bien sûr le simple cadre de la « Mass customization ». La méthode à flux tendus la plus connue est sans doute la méthode « Kanban » (« étiquette » en japonais). C'est une méthode décentralisée de gestion de flux basée sur des ordres de fabrication implicites, les étiquettes [Shingo, 1989].

Ces méthodes s'insèrent dans le contexte plus global de la gestion de l'outil de production : Juste A Temps (JAT). La philosophie du JAT reprend les principes de lutte contre les stocks intempestifs et contre toute forme de gaspillage au niveau de l'appareil de production. Par gaspillage nous entendons ici non seulement toute opération non productrice de valeur ajoutée, mais aussi toute non-utilisation du potentiel productif des opérateurs. On ne produit que ce qui est nécessaire pour satisfaire un besoin de production à un instant donné. L'objectif est donc l'amélioration des performances du système de production, avec bien sûr, un impact indirect sur la qualité et l'efficacité de ce même système.

Généralement, les modèles de flux tendu sont utilisés davantage comme des techniques de gestion des stocks (inventaires) que comme des algorithmes de pilotage de la production. Ils sont essentiellement de nature distribuée, ce qui facilite leur utilisation dans les environnements hétérarchiques. Les premières implémentations de ces algorithmes dans les systèmes hétérarchiques étaient tentées en 1992 par Flavors Technology [Baker, 1998]. Un autre essai a été fait récemment à base de simulation et d'agents à l'aide de l'outil DAPS « Dynamic Adaptation Of Complex Production Systems » [Massotte et al., 2001a].

En raison de l'aspect distribué et des caractéristiques parallèles de ces différentes méthodes, leur traitement exige une technologie spécifique et bien adaptée : la technologie d'agent a été dernièrement considérée comme une tentative réussie pour résoudre ce problème. Dans les prochaines parties, nous expliquerons les principes de base des systèmes multi-agents, les différentes catégories d'agents, les protocoles de négociation utilisés entre les agents et l'utilité de l'utilisation des systèmes multi-agent pour les Systèmes de production décentralisés.

### 3 Les systèmes multi-agents pour le pilotage décentralisé

En 1998, à l'issue d'un workshop et un «Delphi<sup>1</sup> survey» réalisé par des experts internationaux en systèmes manufacturiers, la communauté « Visionary Manufacturing Challenges For 2020 », a identifié les agents « autonomes » et « intelligents comme l'une des clés technologiques permettant aux organisations manufacturières de surmonter les besoins et les enjeux des futurs systèmes de production. Les agents sont perçus comme une technologie d'aide à la modélisation et comme un système de support de décision [NRC, 1998].

#### 3.1 Pourquoi ?

Parunak, dans son article intitulé « What can Agents do in Industry, and Why ? », considère que les agents ne sont pas une panacée pour le logiciel industriel. Comme n'importe quelle autre technologie, ils ont certaines capacités et leurs propres propriétés, et ils sont mieux employés pour les problèmes dont les caractéristiques exigent ces capacités et ces propriétés. Cet auteur indique que les agents sont plus adaptés pour les applications qui sont modulaires, décentralisées, variables, moins structurées, et complexes. De même, ils peuvent rapporter une solution plus robuste et adaptable que celles utilisées par d'autres technologies.

Dans ce contexte, les caractéristiques principales justifiant le pilotage décentralisé par l'intermédiaire de l'approche multi-agents se résument à ces différents points [Parunak, 1998a ; Jennings, 1996 ; Massotte, 1999] :

- **la modularité** : décomposition en sous-systèmes ou sous-produits dotés d'interfaces bien définies. Outre une réduction de la complexité, ceci permet éventuellement la répartition aux différentes entreprises du réseau et la conception de gammes de produits modulables pour être plus réactives par rapport à la demande. La modularité est une propriété fondamentale des agents héritée de la technologie objet ;
- **la décentralisation** : décomposition en processus indépendants capables d'effectuer des tâches sans nécessiter un contrôle permanent d'un autre processus. Caractéristique fondamentale pour supporter la notion d'entreprise virtuelle, elle est également pertinente au sein d'une entreprise unique en regard de l'objectif de flexibilité entraînant la multiplication des ressources et des contraintes à gérer. Le caractère autonome et proactif des agents les rend particulièrement adaptés pour aborder ce type de problème ;

---

<sup>1</sup> La méthode Delphi consiste en une enquête menée auprès d'experts afin d'apporter l'éclairage de ces derniers sur des zones d'incertitude en vue d'une aide à la décision. Cette dernière se proposait notamment de forger un nouveau partenariat de travail entre la science et l'industrie, afin d'évaluer les nouveaux débouchés commerciaux ainsi que les évolutions technologiques, et de diffuser les décisions relatives à l'équilibre et à l'orientation des activités scientifiques et technologiques financées par l'État.

- **L'adaptabilité** : capacité à répondre aux fluctuations de l'environnement. Aujourd'hui, la pression du marché conduit les entreprises à évoluer dans un environnement beaucoup plus versatile qu'auparavant. La quantité et la diversité des paramètres à prendre en compte nécessitent le développement de systèmes informatiques complexes et donc très coûteux. Dans ces systèmes, la recherche d'adaptabilité est directement liée aux caractéristiques précédentes. La modularité permet de modifier seulement une partie (un module) du système à un instant donné, mais elle n'est pas suffisante pour supporter des changements fréquents si le contrôle de l'ensemble est centralisé. La décentralisation minimise l'impact que produit le changement d'un module sur les autres modules en découplant les modules les uns des autres. Ainsi une erreur dans un module n'a de conséquence que sur les modules qui interagissent avec lui, le reste du système n'étant pas affecté. De part leurs caractéristiques, les architectures d'agents permettent la réutilisation du code et l'auto-configuration réduisant ainsi les coûts et les temps de développement ;
  
- **L'absence de structuration et la reconfiguration automatique** : dans les approches de conception conventionnelle, on commence par identifier les entités d'un système et les relations entre elles. Ensuite, on essaie de déterminer la structure interne de chaque entité en fonction des données et des traitements qu'elles doivent accomplir. Finalement, on écrit le programme chargé de gérer les flux d'information et de coordonner ces différentes entités. Cette démarche suppose la connaissance dès le départ de la description précise et complète de la structure du système, des informations qui circulent, des acteurs présents et de l'expertise de ces acteurs en terme de savoir et de savoir-faire. L'acquisition de savoir et savoir-faire est déjà un problème en soi, lié au caractère subjectif et souvent informel de ce type de connaissance. Si en plus on est dans un système dynamique où les acteurs sont susceptibles de changer, introduisant ou supprimant par là-même des informations et des compétences on se rend compte que, dès le départ, les approches traditionnelles vont entraîner des coûts de maintenance prohibitifs. Comme nous l'avons indiqué précédemment, les méthodes de conception traditionnelles analysent toujours un système en terme d'entités, qu'elles soient fonctionnelles ou opérationnelles, qu'il faut ensuite structurer pour construire le modèle d'interaction. Or, l'information structurelle n'est pas toujours disponible au départ et l'information disponible n'est pas toujours facilement structurable et formalisable. Même un système bien structuré au départ peut être considéré comme mal structuré, si on se projette sur l'ensemble de son cycle de vie en considérant les évolutions qui seront nécessaires en fonction des besoins des utilisateurs. Les applications sur Internet, en particulier le commerce électronique, sont l'illustration typique aujourd'hui de ce type de problème. En ayant une démarche inverse partant du modèle d'interaction, les agents peuvent découvrir la structure du système pendant son fonctionnement au lieu de connaître sa conception dès le départ. Ceci permet de concevoir le système comme un ensemble de classes qui vont générer une structure pour un domaine donné, plutôt que la structure elle-même, augmentant

ainsi la durée de vie du système et réduisant les coûts de maintenance et de reconfiguration ;

- **la complexité structurelle** : l'augmentation de la complexité des produits rend la tâche de conception plus difficile et augmente le nombre de machines en interaction qui doivent être contrôlées. L'évolution vers l'entreprise virtuelle ne fait qu'aggraver cette tendance et conduit à des méthodes d'optimisation conventionnelles soumises à l'explosion combinatoire des calculs. En effet, le nombre des interactions dans un ensemble d'éléments croît plus rapidement que le nombre d'éléments, ou le nombre de fonctions, dans le système. En associant un agent à chaque élément interactif, il est possible de remplacer le codage explicite de l'ensemble des interactions par sa génération au moment de l'exécution. Le caractère combinatoire des comportements n'est plus envisagé au moment de la conception, ce qui réduit de façon drastique la quantité de code à produire et donc le coût du système à développer ;
- **la complexité comportementale** : est très importante car elle est due à la présence de non-linéarités et boucles de 'feed-back' amplificatrices, au niveau du flot de produits. Le flot d'informations, quant à lui, est aussi soumis à des fluctuations perturbantes. L'ensemble de ces facteurs conduit à des systèmes sujets à des SCI (Sensibilité aux Conditions Initiales) dans lesquels les variables essentielles présentent du Chaos Déterministe. La conséquence directe de tels comportements complexes est due au fait que des systèmes même «simples» ont un comportement complexe et sont «non-prédictibles». Ainsi, toute planification préalable ne pourra que permettre d'initialiser le système de production, d'en prévoir les ressources et moyens, mais en aucun cas un ordonnancement pourra être exécuté de manière fiable, il ne sera donc d'aucune utilité [Massotte, 1999].

### 3.2 Applications

Plusieurs travaux de recherche et projets industriels ont appliqué le pilotage à base d'agents dans les systèmes de production. Shen et Norrie citent plus de 30 travaux et projet utilisant la technologie agent pour le pilotage des SP [Shen et Norrie, 1999].

Les premiers travaux ont été réalisés par Shaw (1983) et Parunack (1987). Shaw a proposé l'utilisation des agents pour le pilotage des systèmes de production. Il a émis l'idée innovante qu'une cellule de production pourrait sous-traiter le travail à d'autres cellules par le mécanisme d'enchères [Shaw et Whinston, 1983 ; Shaw, 1988]. YAMS (Yet Another Manufacturing System) a été parmi les premiers projets industriels à base d'agent [Parunak, 1987]. Les agents dans YAMS représentent les usines et leurs composantes. Chaque agent a une collection de plan représentant les capacités de ces composantes. Les négociations s'effectuent par l'intermédiaire du Contract-Net Protocol. D'autres projets ont été développés durant ces dernières

années utilisant différentes architectures de pilotage, différents types d'agent et différents mécanismes de négociation (voir tableau 2.1).

Projet	Groupe	Caractéristiques principales
<b>YAMS</b>	Parunack, 1987	premier projet industriel a base d'agent, agents représentent les usines et leurs composantes Contract-Net Protocol pour les négociations .
<b>DAS</b>	Burk et Prosser, 1991	Architecture hiérarchique agents représentent les ressources et les groupes de ressources
<b>ADDYMS</b>	Butler et Ohtsubo, 1992	Agents representent les ressources physiques ordonnancement dynamique et locale
<b>IFCF</b>	Lin et Solberg, 1992	agents representent les ressources, market-like control model, combinaison des mécanismes de coût et d'objectif
<b>AARIA</b>	Parunak et al., 1998	agents autonomes pour représenter, les entités physique, processus et opérations, ordonnancement à base de "Forward/Backward"
<b>ABACUS</b>	McEleney et al., 1998	Agents BDI pour l'ordonnancement
<b>MASCADA</b>	Valckenaers, 1999	architecture en couches décisionnelles multi-niveaux approche de type distribuée supervisée agents représentent les équipements de production
<b>PABADIS</b>	PABADIS, 2000	notion de "plug-and-participate" pour les ressources, composants de base sont les agents et les services, suppression totale ou partielle des tâches de planification et d'ordonnancement,

**Tableau 2.1 : Exemples des projets utilisant la technologie agent pour le pilotage des systèmes de production**

## **4 Les Systèmes Multi-Agents – Rappels et concepts**

### **4.1 Introduction**

Les Systèmes Multi-Agent (SMA) sont apparus au carrefour des recherches sur l'intelligence artificielle distribuée et sur la vie artificielle. Ces systèmes sont développés à partir de schémas de raisonnement ou d'organisations empruntées aux domaines de la vie et de la société [Ferber, 1995]. Ils empruntent à l'intelligence artificielle distribuée les modes de communication et de concertation entre agents. Ils reprennent les idées d'autonomie et d'émergence du résultat final à partir des interactions individuelles liées à la vie artificielle.

Les SMA sont particulièrement adaptés pour proposer des solutions réactives et robustes à des problèmes complexes pour lesquels il n'existe pas de contrôle centralisé [Jennings, 1999 ; Casteran et al., 2000]. Ils sont des systèmes idéaux pour représenter des problèmes possédant de multiples méthodes de résolution, de multiples perspectives et/ou de multiples résolveurs (Chab-draa et al., 2001). Ces systèmes possèdent les avantages traditionnels de la résolution distribuée et concurrente de problèmes comme la modularité, la vitesse (avec le parallélisme), et la fiabilité (due à la redondance). Ils héritent aussi des bénéfices envisageables de l'Intelligence Artificielle comme le raisonnement symbolique (au niveau des connaissances), la facilité de maintenance, la réutilisation et la portabilité, mais surtout ils ont l'avantage de faire intervenir des schémas d'interaction sophistiqués. Les types courants d'interaction incluent :

- la coopération qui implique une activité commune d'un ensemble d'agents afin d'atteindre un but commun,
- la coordination qui suppose l'organisation de l'activité de résolution de chaque agent afin d'éviter les interactions inutiles et d'exploiter celles qui sont bénéfiques,
- la négociation dont l'objet est d'arriver à un compromis acceptable entre tous les agents engagés en compétition.

Les concepts « Agent » et « Multi-agent » sont relativement récents et jusqu'à maintenant nous n'avons pas de définition formelle de ce que sont un agent et un système multi-agent, qui soit acceptée par tout le monde [Jennings et al., 1998]. Dans ce qui suit nous donnerons quelques définitions proposées autour de ces deux concepts.

### **4.2 Qu'est-ce qu'un agent ?**

Dans la littérature, on trouve une multitude de définitions du concept agent. Elles présentent certaines similitudes et dépendent du type d'application pour

laquelle est conçu l'agent. Dans le cadre de notre travail, nous porterons notre attention sur les définitions qui suivent :

Ferber a proposé assimiler le terme agent à une entité physique ou virtuelle [Ferber, 1995] :

- qui est capable d'agir dans un environnement,
- qui peut communiquer directement avec d'autres agents,
- qui est mue par un ensemble de tendances, sous la forme d'objectifs individuels, d'une fonction de satisfaction ou de survie, qu'elle cherche à optimiser,
- qui possède des ressources propres,
- qui est capable de percevoir partiellement son environnement,
- qui ne dispose que d'une représentation partielle de l'environnement,
- qui possède des compétences et offre des services,
- qui peut éventuellement se reproduire,
- dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit.

Cette définition introduit des caractéristiques exploitables, comme la notion d'objectif et de perception limités ou locale de l'environnement, comme c'est le cas dans plusieurs applications. En effet, un robot mobile a généralement un certain rayon de perception et d'influence, pour observer et agir sur son environnement physique. Un agent mobile dans le réseau Internet ne peut pas observer tout le réseau ; il ne peut en observer qu'une partie. De même, dans un atelier de production décentralisé l'affectation des tâches des produits sur des ressources se fait de façon locale ; en conclusion, les interactions entre les agents produits et les agents ressources dépendent en grande partie de leurs propres objectifs et des contraintes dans leur environnement local.

Wooldridge et Jennings ont défini l'agent comme étant un Système Informatique, **Situé** dans un environnement, et qui agit d'une façon **Autonome** et **Flexible** pour atteindre certains objectifs pour lesquels il a été conçu [Wooldridge et Jennings, 1995 ; Jennings et al., 1998 ] :

1. **Situé** : un agent est dit situé, s'il est capable d'agir sur son environnement à partir des entrées sensorielles qu'il reçoit de ce même environnement,
2. **Autonome** : un agent est dit autonome s'il est capable d'agir sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne,
3. **Flexible** : un agent est dit flexible s'il est capable d'agir de manière réactive, proactive et sociale :
  - **Réactif** : être capable de percevoir son environnement et de réagir dans le temps,
  - **Proactif** : prendre l'initiative et être opportuniste au bon moment,

- Social : être capable d'interagir avec les autres agents quand la situation l'exige (pour compléter ses tâches ou coopérer avec eux).

Un agent est donc une entité physique ou virtuelle en situation dans un environnement avec lequel il interagit de façon autonome et flexible. Cette notion de situation ou d'agent situé implique que les agents autonomes doivent agir dans un monde réel, en perpétuel changement, partiellement observable et intrinsèquement imprévisible. De même ils doivent réagir en temps réel parce que l'environnement change constamment, et parce qu'ils doivent éventuellement tenir compte des actions d'autres agents [Drogoul et Meyer, 1999].

Dans cette thèse, nous nous intéresserons au pilotage décentralisé des systèmes de production. Les produits et les ressources sont représentés par des agents en interaction permanente pour une meilleure allocation des tâches sur les ressources. Ces agents doivent être capables de réagir à n'importe quel moment, dans n'importe quel contexte géographique ou physique et pour toutes les perturbations rencontrées tout en respectant leurs objectifs, et les contraintes du système. Pour cela, les notions de perception partielle de l'environnement, de situation, d'autonomie et de réactivité introduites par les précédents auteurs représentent bien la définition que nous adoptons pour notre agent.

### **4.3 Qu'est-ce qu'un Système Multi-agent ?**

Comme pour les agents, plusieurs définitions ont été proposées, et nous allons commencer par présenter la définition formelle introduite par Ferber. Un SMA est composé des éléments suivants :

- un Environnement  $E$  c'est à dire un espace disposant généralement d'une métrique,
- un ensemble d'objets  $O$ , ces objets sont situés, c'est à dire que pour tout objet, il est possible, à un moment donné, d'associer une position dans  $E$ . ils sont passifs, ils peuvent être perçus, détruits, créés et modifiés par les agents,
- un ensemble d'Agents  $A$  qui sont les entités actives du système ( $A \subseteq O$ ),
- un ensemble de Relations  $R$  qui unissent les objets entre eux,
- un ensemble d'Opérations  $O_p$  permettant aux agents de percevoir, de détruire, de créer, de transformer et de manipuler les objets de  $O$ ,
- un ensemble d'opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification (les lois de l'univers).
- Une autre définition est celle donnée par Wooldridge et Jennings, qui présente un SMA comme étant un ensemble d'agents en interaction afin de réaliser leurs buts ou d'accomplir leurs tâches [Wooldridge et Jennings, 1995]. Les interactions peuvent directes par l'intermédiaire des

communications, comme elles peuvent être indirectes via l'action et la perception de l'environnement. Les interactions peuvent être mises en œuvre dans un but de :coopération entre les agents, lorsqu'ils ont des buts communs,

- coordination, c'est à dire d'organisation pour éviter les conflits et tirer le maximum de profit de leurs interactions afin de réaliser leurs buts,
- compétition, lorsque les agents ont des buts antagonistes.

Dans la thèse nous adoptons la définition de Woldridge et Jennings, un SMA étant un ensemble d'agents ayant des buts ou des tâches, et qui interagissent pour les accomplir en mode de coopération et/ou de compétition.

#### **4.4 Agents et Objets**

Un **objet** est défini par un ensemble de services offerts (ses méthodes) qu'il ne peut refuser d'exécuter si un autre objet le lui demande. Les **objets** exécutent des tâches ; ils n'ont ni but, ni recherche de satisfaction. En revanche les **agents**, bien plus élaborés, disposent d'objectifs qui leur donnent une autonomie de décision vis-à-vis des messages qu'ils reçoivent.

Les **objets** utilisent un mécanisme d'envoi de message qui se résume à un simple appel de méthode. Pour les **agents**, les interactions sont plus complexes et font intervenir des communications de haut niveau, où l'important est que l'agent décide par lui-même comment interagir et réagir aux messages qu'il reçoit.

Un **agent** est une entité capable de recevoir et d'émettre des messages. Ce comportement est minimaliste, et correspond d'ailleurs tout à fait à la définition d'un objet. Néanmoins, un objet est contraint de répondre aux requêtes qui lui sont soumises, contrairement à l'agent qui dispose de son propre libre arbitre. Concrètement, les objets font ce qu'on leur demande, tandis qu'avec les agents, il faut négocier. Donc, il est important que les agents contrôlent eux-mêmes leur comportement et les ressources qu'ils possèdent. C'est ce qui caractérise leur autonomie, et d'une certaine manière leur donne conscience de leurs possibilités.

Dans un **SMA**, l'originalité et la source de toute la problématique **SMA** résident dans l'hypothèse d'autonomie de décision attribuée aux agents, c'est-à-dire dans la conservation d'un maximum de degrés de liberté pour chacun de ces agents. Cette perspective d'autonomie introduit une dynamique qui complexifie le maintien de la cohérence et de l'efficacité du système. Il est ici indispensable de combiner une communication plus évoluée que les objets avec une recherche de satisfaction des objectifs de chaque agent.

Néanmoins, le lien existant entre la notion d'objet et celle d'agent reste fort. Il n'y a pas de frontière nette : un objet peut être considéré comme un agent dont le langage d'expression se résume à l'emploi de mots clés correspondant à ses

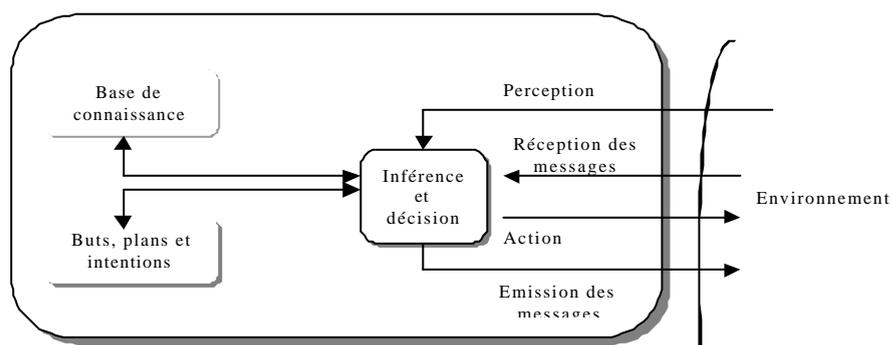
méthodes. Un autre aspect qui rajoute un peu plus à la confusion entre la notion d'objet et celle d'agent, est que souvent un agent est implémenté, pour des raisons pratiques, sous la forme d'objets. Par conséquent, on peut dire que les agents sont définis à un niveau conceptuel tandis que les objets sont définis à un niveau d'implémentation et d'exécution.

#### 4.5 Différentes catégories et modèles d'agents

Après avoir défini les agents et les systèmes multi-agents, nous présentons dans cette partie les différents modèles d'agents, afin de comprendre leurs caractéristiques et leurs modes de fonctionnement. Nous distinguons deux grandes familles d'agents les agents réactifs et agents cognitifs.

##### 4.5.1 Agents cognitifs

C'est le premier modèle d'agents qui a été proposé. Il est nommé aussi agent délibératif. Il est basé sur l'IA symbolique, et il permet de planifier les actions d'un agent au sein de son environnement [Fikes et Nilsson, 1971]. En effet, les agents cognitifs sont capables à eux seuls de réaliser des opérations relativement complexes. Généralement, ils coopèrent les uns avec les autres pour atteindre un but commun (résolution d'un problème, une tâche complexe, etc.). Ils possèdent un ensemble de représentations explicites (sur l'environnement, sur les autres agents et sur eux-mêmes) décrits dans une base de connaissances sur laquelle ils peuvent raisonner. Ils réagissent en fonction de leurs connaissances, leurs buts, de leurs échanges d'informations avec les autres agents et de la perception de l'environnement (voir figure 2.7). Ils sont dotés de moyens et mécanismes de communication pour gérer les interactions avec d'autres agents (coopération, coordination et négociation).



**Figure 2.7 : Modèle d'un agent cognitif**

Ce modèle d'agent est une métaphore du modèle humain et s'appuie sur la sociologie des organisations [Liu, 2002]. Nous pouvons aussi trouver son origine dans la volonté de faire coopérer des systèmes experts classiques dans le domaine d'IA.

L'agent cognitif, en terme de BDI pour Beliefs (croyances), Désires (désirs) et Intention (intentions), [Bratman, 1987 ; Bratman et al., 1988], postule que ses actions, tout comme celles d'un être humain, doivent être dictées par des représentations abstraites du monde, incluant la représentation de ses propres capacités, ses buts ainsi que ceux des autres agents. Ce modèle est fondé sur des extensions de la logique, et est basé sur les états mentaux suivants :

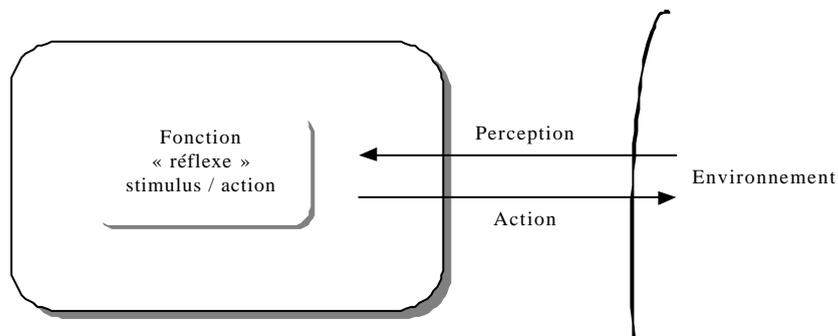
- les croyances : les connaissances de l'agent sur l'environnement,
- les désirs : les « options » de l'agent, c'est à dire les différents objectifs vers lesquels l'agent peut vouloir transiter,
- les intentions : les états vers lesquels l'agent a choisi de transiter.

Le fonctionnement d'un agent BDI consiste donc à actualiser ses croyances à partir des informations qu'il reçoit de son environnement, ensuite, à déterminer les différents choix qu'il a en face, les analyser et fixer ses états buts en fonction d'eux (intentions), et enfin, à définir ses actions en tenant en compte de ses intentions.

L'agent cognitif traite généralement des informations qualitatives tout en utilisant un raisonnement qualitatif ou symbolique. Ces traitements peuvent être établis par l'intermédiaire des outils comme les Classifieurs Génétiques/ Neuronaux (CG/ CN) ou des Systèmes d'Expert (SE). Ces agents peuvent utiliser des mécanismes comme les systèmes à base de raisonnement par cas « Case Based Reasoning » (CBR), les systèmes à base de connaissances « Knowledge Base System » (KBS), les théories des jeux « Game Theory »(GT) ou à base de la logique utilisée dans les agent BDI, voir (C) et (D) dans la figure 2.9.

#### 4.5.2 Agents réactifs

Parmi les critiques du raisonnement symbolique figure Brooks, qui, par le biais de plusieurs papiers [Brooks, 1986 ; 1991], manifesta son opposition au modèle symbolique et proposa une approche alternative appelée aujourd'hui IA active. Selon lui, le comportement intelligent devrait émerger de l'interaction entre divers comportements plus simples.



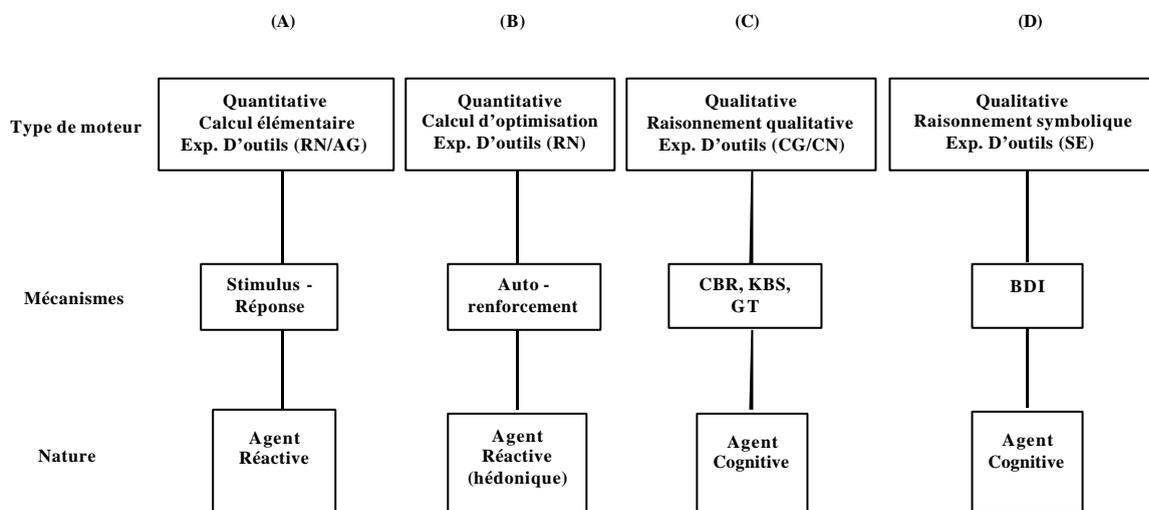
**Figure 2.8 : Modèle d'un agent réactif**

Dans ce même contexte, nous considérons que les agents réactifs n'ont de connaissance explicite, ni de l'environnement, ni des autres agents, ni de leur passé, pas plus que de leurs buts (pas de planification de leurs actions). Ce sont des agents qui réagissent uniquement à leur perception de l'environnement et qui agissent en fonction de cette perception (voir figure 2.8).

Ce modèle d'agent est une métaphore du modèle «fourmi», il s'appuie sur les sciences de la vie et l'intelligence collective [Bonabeau et Theraulaz, 1994 ; Bonabeau et al. 1999]

Dans la même catégorie que les agents réactifs, mais avec davantage de rationalité, se trouvent les agents hédoniques. Ces agents apprennent, par auto-renforcement, à modifier leur comportement afin d'augmenter leur " plaisir ou satisfaction ". Ils sont capables d'anticipations " hédoniques" et d'adaptation lente à partir de leur expérience historique, ce qui suppose un niveau de rationalité plus élevé que l'agent purement réactif.

Les agents réactifs, traitent généralement des informations quantitatives tout en utilisant des calculs élémentaires ou d'optimisation. Ils peuvent être construits par des réseaux connexionnistes comme les Réseaux de Neurones (RN) ou en utilisant des simples algorithmes de calcul comme les Algorithmes Génétiques (AG). Leurs capacités répondent à la loi stimulus/action et apprentissage par auto-renforcement, voir (A) et (B) dans la figure 2.9.



**Figure 2.9 : Structures des modèles d'agents**

Les agents réactifs sont les plus simples à implémenter, et leur efficacité a été vérifiée dans plusieurs applications. Ils présentent toutefois des limites dues aux points suivants :

- l'agent n'a pas la représentation mentale de l'environnement, et doit choisir ses bonnes actions à partir des données locales uniquement,

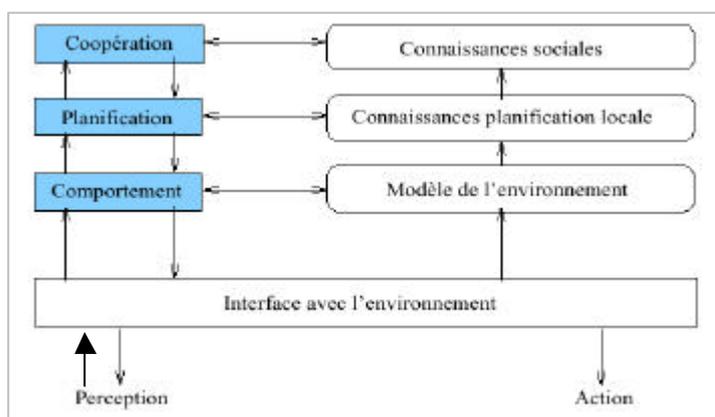
- le comportement global de l'agent ne peut pas être facilement prévu, par conséquent, il n'est pas toujours possible de concevoir un comportement d'agent en fonction du but spécifié.

Contrairement aux agents réactifs, les agents cognitifs sont beaucoup plus complexes et plus difficiles à mettre en œuvre. Chaque agent se fonde sur ses propres compétences de façon isolée pour résoudre un problème, ce sont les principales limites de cette architecture.

Pour faire face aux inconvénients de ces deux modèles, les chercheurs ont combiné ces deux facettes, opposées mais complémentaires, de la conception des agents. Cette combinaison fait apparaître les agents hybrides.

#### 4.5.3 Agents hybrides

Les agents hybrides sont conçus pour combiner des capacités réactives à des capacités cognitives, ce qui leur permet d'adapter leur comportement en temps réel à l'évolution de l'environnement [Ferguson, 1992 ; Fischer, 1999]. Dans le modèle hybride, un agent est composé de plusieurs couches, rangées selon une hiérarchie. La plupart des architectures considèrent que trois couches suffisent amplement. Ainsi, au plus bas niveau de l'architecture, on retrouve habituellement une couche purement réactive, qui prend ses décisions en se basant sur des données brutes en provenance des senseurs. La couche intermédiaire fait abstraction des données brutes et travaille plutôt avec une vision des connaissances de l'environnement. Finalement, la couche supérieure se charge des aspects sociaux de l'environnement (communication, coopération, négociation), c'est-à-dire du raisonnement tenant compte des autres agents. Le modèle InteRRap proposé par Müller et Pischel, voir figure 2.10 [Müller et Pischel, 1994] figure parmi les exemples du modèle hybride. Dans ce modèle, chaque couche comporte une base de connaissance en fonction de son degré d'abstraction : modèle de l'environnement au niveau de la couche réactive, modèle mental au niveau de la couche de planification locale et modèle social pour la couche de coopération et de planification globale.



**Figure 2.10 : Modèle d'agents InteRRap**

Dans ce mémoire, nous utilisons un modèle évolutif d'agent qui peut être considéré comme un agent hybride. Ce modèle est appliqué aux agents « produits » et « ressources » pour l'allocation dynamique des ressources dans un atelier de production. L'architecture de notre modèle d'agent sera présenté dans le chapitre 4.

#### 4.5.4 Synthèse

En résumé, l'ensemble des architectures et concepts proposés pour mettre en œuvre de nouvelles approches de gestion et de contrôle de système de production est représenté par la figure 2.9. De gauche à droite, nous avons fait figurer des modèles d'abord quantitatifs puis qualitatifs et cognitifs, dont la mise en œuvre requiert des soins particuliers.

### **4.6 Domaines d'application des SMA**

Les systèmes multi-agents sont à l'intersection de plusieurs domaines scientifiques tels que : l'information répartie, le génie logiciel, l'intelligence collective, l'intelligence artificielle et la vie artificielle [Chaib-draa, 2001]. Par conséquent, ils font appel à plusieurs autres disciplines, telles que la sociologie, la psychologie sociale, les sciences cognitives, la biologie, etc. Il en découle que les SMA sont appliqués dans divers domaines, comme l'industrie, le commerce, le divertissement et la médecine. Parmi les applications industrielles nous pouvons citer le pilotage des Système de Production [Parunack, 1987 ; PABADIS, 1999], la télécommunication comme le contrôle de réseaux [Schoonderwoerd et al., 1997], le contrôle de trafic aérien [Zeghal, 1993] et la gestion du trafic et du transport [Burmeister et al., 1997]. Parmi les applications commerciales à base d'agent, on trouve la gestion de l'information [Chen et Sycara, 1998], et le commerce électronique [Chavez et Kasbah, 1996]. Un grand domaine d'application des SMA est celui des divertissements, comme dans les jeux [Grand et Cliff, 1998], le théâtre interactif et la réalité virtuelle [Bates, 1994]. Finalement, les applications à base d'agents développés dans le cadre de la médecine ont été utilisées pour les surveillances des patients [Durand et al., 2001] et pour la gestion des soins aux patients [Huang et al., 1995].

Après une revue générale des SMA, nous nous intéressons maintenant à la partie essentielle dans les SMA, les interactions entre les différents agents dans le système. Généralement, ces interactions entre agents sont effectuées à travers des protocoles de négociation afin de satisfaire les intérêts locaux des agents et ceux globaux du système. Dans les prochains paragraphes nous détaillons les différents protocoles de négociation existant dans la littérature des SMA.

## **4.7 Protocoles de négociations**

Un Protocole de Négociation (PN) dans un SMA est constitué de l'ensemble des règles utilisées pour organiser la communication, la négociation, la séquence de conversation [Chen, 1998] et la prise de décision entre les agents [Krothapalli et Deshmukh, 1997]. Il sélectionne également les agents qui peuvent communiquer entre eux. Comment procéder dans chaque communication et conversation? Quel type de message est-il permis d'avoir durant la négociation? Un protocole de négociation spécifie comment les agents peuvent converger vers un accord dans une négociation en cas de conflit. La conception et l'analyse des PN sont des tâches importantes pour obtenir un PN efficace et efficient. Rosenschein et Zlotkin spécifient les propriétés et les attributs pour le jugement des PN comme suit [Rosenschein et Zlotkin, 1994 ; Kraus, 2001] :

- Temps de négociation : les négociations qui se terminent sans délai sont préférées. En effet, le type de négociation qui utilise un délai pour trouver un accord durant la négociation augmente le coût de communication et le temps de calcul d'une façon considérable.
- Efficience : un bénéfice efficace est souhaité. il se traduit par une satisfaction globale liée à l'augmentation de nombre d'agents profitant de ce bénéfice dans la négociation,
- Stabilité : les agents ne doivent pas dévier de leurs engagements et leurs accords pour les stratégies à utiliser,
- Simplicité : le protocole doit faire de faibles calculs pour chaque demande émanant des agents ; il utilise le moins possible de communications entre eux,
- Distribution : les règles d'interactions ne doivent pas avoir besoin d'une prise de décision centralisée,
- Symétrie : aucun processus de négociation ne doit être appliqué arbitrairement et partialement contre certains agents.

Dans cette thèse nous avons considéré quatre types de PN. Ils se distinguent entre eux par les mécanismes utilisés durant la négociation entre les agents. Ces mécanismes sont à base d'enchères, de médiation, d'argumentations et de détermination de stratégies.

## 4.7.1 Les types de Protocoles de Négociation (PN)

### 4.7.1.1 PN à base d'enchères

Parmi les PNs à base d'enchère le plus connu, que nous avons déjà cité, est le réseau contractuel ou « Contract Net Protocol » (CNP) développé par Smith [Smith, 1980 ; Smith et Davis, 1981]. Ce protocole se compose de cinq étapes et se déroule entre clients et fournisseurs, de la manière suivante :

- Le client sollicite des propositions de fournisseurs potentiels en émettant un appel d'offre décrivant la tâche et les conditions nécessaires à son exécution,
- Les fournisseurs potentiels évaluent les appels d'offres provenant des différents clients,
- Les fournisseurs potentiels fournissent des propositions pour accomplir la tâche,
- Le client évalue les propositions et fait son choix sur un, ou plusieurs fournisseurs pouvant accomplir la tâche. Dans certains cas il ne peut pas y avoir de choix,
- Finalement client et fournisseur établissent un contrat pour l'exécution de la tâche.

Ce protocole a été largement utilisé dans la littérature des SMA. Il a été étendu à différents mécanismes d'enchères [FIPA, 2000 ; Liu, 2002] et autres protocoles de négociation [Lin et Solberg, 1992 ; Krothapalli et Deshmukh, 1999]. Parmi les applications utilisant ce protocole, le Commerce électronique utilise largement ce modèle à travers les différents sites d'achats et de vente sur Internet. De même les nouveaux principes des entreprises virtuelles ou « virtual factory » emploient ce protocole au niveau de la gestion des chaînes d'approvisionnement (Supply chain Management) [Shen et Norrie, 1998; Liu, 1999]

Nous distinguons aussi cinq types d'enchères. L'enchère Anglaise ou (Ascendante), Hollandaise ou (Descendante), Scellées au premier prix ou (premier prix), scellées au second prix (ou Vickrey) et l'enchère inversée (ou « reverse auction »). Dans l'enchère Anglaise, le prix du bien vendu débute bas et croît jusqu'à ce qu'il n'y ait qu'une seule offre émise pour le dernier prix choisi. Par contre, dans l'enchère Hollandaise le prix du bien vendu débute haut et décroît jusqu'à ce qu'un acheteur accepte le prix choisi. Dans les enchères Scellées au premier prix, les soumissions d'offres s'effectuent sous plis scellés, et la meilleure offre, au sens du prix, gagne l'enchère. L'enchère Veckry utilise le même principe que la précédente, sauf que le gagnant paye le prix de la seconde meilleure offre. Finalement, dans l'enchère inversée les vendeurs sont en concurrence, et proposent des prix de plus en plus bas, au-dessus de leurs prix de réserve en respectant un décrétement fixé par l'acheteur. Chaque vendeur connaît à tout instant la meilleure offre venant de ses concurrents. L'enchère dure une période de temps fixée à l'avance. Le gagnant est

celui qui a proposé le dernier meilleur prix. Ce type d'enchères est en plein développement sur Internet (e-intelligence, e-bidding, e-auction), il est avantageux pour l'acheteur parce qu'il oblige les vendeurs à consentir leurs meilleurs tarifs.

Les enchères peuvent être classées en différentes catégories selon les critères du nombre d'issues (unique ou multiple) et du nombre de parties participantes (acheteur unique / plusieurs vendeurs, plusieurs acheteur / vendeur unique, plusieurs acheteurs / plusieurs vendeurs).

Actuellement, l'« issue unique » est la catégorie d'enchère en ligne la plus utilisée (le moteur de recherche Yahoo liste 100 sites d'enchères). Le point commun entre ces différents sites se résume à la négociation de l'issue unique de « prix » seulement. De même, les cinq types d'enchères cités ci-dessus et les deux premières catégories (acheteur unique / plusieurs vendeurs, plusieurs acheteur / vendeur unique) sont les plus utilisés.

Comme exemple de la troisième catégorie (plusieurs acheteurs / plusieurs vendeurs) nous pouvons citer la double enchère. Dans un système de double enchère, les vendeurs et les acheteurs soumettent le prix auquel ils souhaitent vendre ou acheter. Offres et demandes peuvent se faire à n'importe quel moment. Les participants sont libres d'accepter n'importe quelle demande ou offre, et d'annoncer de nouvelles demandes et offres. De cette manière, une multitude d'échanges de négociation seront effectués. Il s'agit en fait d'une situation identique à celle des marchés financiers où convergent de nombreux vendeurs et acheteurs, et où l'offre et la demande changent de façon dynamique. Comme exemple connu à cette catégorie nous avons le New York Stock Exchange (NYSE) proposé par Walch et Wellman [Walch et Wellman, 1998].

Bien que l'utilisation de l'issue unique soit largement répandue dans le commerce électronique, la vente au détail reste l'inconvénient majeur de ces catégories [Guttman et Maes, 1998]. En différenciant seulement les produits selon leurs prix, nous négligeons les autres facteurs importants pour l'acheteur et le vendeur, comme le délai, la maintenance, la qualité, etc. Comme exemple d'enchère à issue multiple, le site « optimark » supporte deux issues et « web-based auctions » qui supporte des issues multiples [Teich et al., 1998].

Outre le nombre d'issues et le nombre des parties participantes, nous pouvons ajouter l'identité des participants (acheteurs ou vendeurs) comme un nouveau critère à prendre en compte. De cette manière, nous aurons deux nouvelles catégories (acheteur unique / plusieurs acheteurs, vendeur unique / plusieurs vendeurs). Il ne faut pas confondre entre ces nouvelles catégories et celles utilisées dans le commerce électronique, C2C (Consumers to Consumers ) ou B2B (Business to Business ), pour désigner des achats et des ventes entre particuliers ou entre entreprises.

Ce critère permet d'insérer la notion de coo-pétition (coopération + compétition) intra-acheteurs et intra-vendeurs en introduisant la notion «enchères intra-organisation» (acheteurs / acheteurs et vendeurs / vendeurs), où les membres de la même organisation (acheteurs ou vendeurs) ont la même identité et le même objectif, mais se différencient par rapport à leur capacité d'achat ou de vente. Ces critères reflètent la négociation réelle dans la société où les acheteurs et les vendeurs sont en compétition entre eux pour obtenir le meilleur prix en fonction de leur liste d'achat ou de vente (quantité, prix, délai, qualité, etc.) et en coopération pour acheter avec le prix le plus bas ou pour augmenter leur capacité de vente (organisation des mutuelles, entreprise virtuelle en réseau, etc.).

Ce critère a été utilisé pour la conception de nouveaux protocoles de négociations entre les entités produit (acheteur) et ressource (vendeur) pour le pilotage de la production dans ce travail.

#### *4.7.1.2 PN à base de médiation*

Dans ce protocole, un agent nommé «agent médiateur » est utilisé pour l'aide à la coordination entre les agents dans le processus de négociation. Un agent médiateur prend le rôle de la troisième partie neutre qui coordonne les autres parties de négociation. Les applications de ce PN sont E-commerce, le pilotage des SPs à base d'agents.

Les marchés d'enchères peuvent être considérés comme des PNs à base de médiation quand le processus de négociation passe par un intermédiaire, médiateur, courtier ou « commissaire-priseur », etc. [Liu, 2002]

Ehtalmo et al. utilisent un médiateur pour obtenir les préférences des agents à partir de leur prise de décision dans le but d'identifier les solutions pareto-optimales de prise de décision entre eux [Ehtala, 1999].

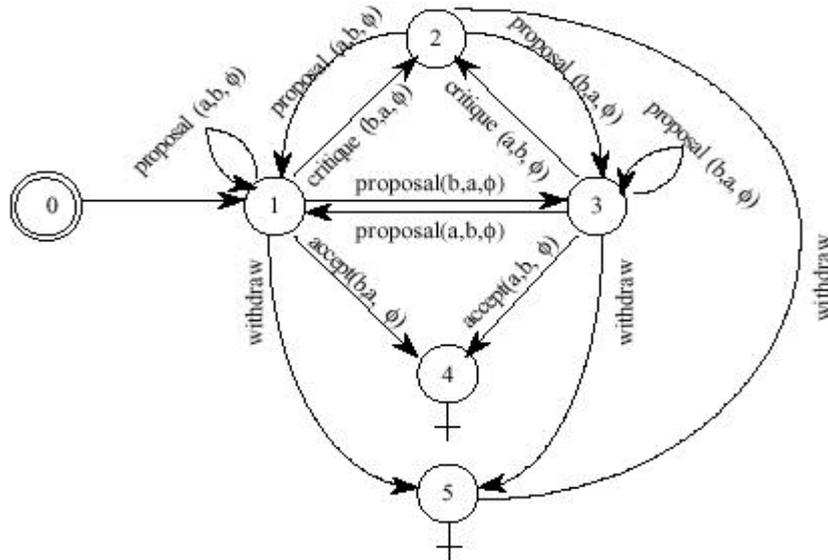
Sycara, dans son modèle PERSUADER, emploie un médiateur pour coordonner les négociations à travers des simulations et en se basant à un raisonnement à Base De Cas [Sycara, 1988].

Maturana, Norrie et Shen introduisent les agents médiateurs pour la coordination entre les agents dans le but d'établir des sous-systèmes d'agent collaborateurs (coordination clusters ou virtual clusters). Le mécanisme utilisé consiste à recevoir une demande d'un agent, la comprendre, trouver les récepteurs appropriés pour le message, et annoncer le message au cluster d'agent choisi. Ce mécanisme a été utilisé dans l'architecture Methamorph [Maturana et Norrie 1996; Shen et al., 1998 ; Maturana et al., 1999] dans le but de piloter des entreprises virtuelles à base d'agents.

#### 4.7.1.3 PN à base des argumentations et de jeux de dialogues

Ce type de PN est utilisé aussi dans différents travaux de conception de PN et d'automatisation de négociation dans les SMA [Kraus, 2001 ; Sierra et al., 1997]. Dans ce modèle les agents sont doués de notions d'attitudes mentales comme Beliefs, Desires and Intentions (BDI) [Rao et Georgeff, 1995] associées à des mécanismes de raisonnement sur leur propre état mental et les états mentaux des autres. A travers la notion de négociation à base d'argumentation, le principe de négociation est vu comme un processus de changement des buts des autres agents par l'intermédiaire de la persuasion. Les applications de ce PN sont limitées au niveau de travaux de recherche théorique.

Parson, Sierra et Jennings ont également spécifié un protocole générique de négociation à base d'argumentation en forme de diagramme de transition d'état de 0 jusqu'à 5, entre deux agents (a et b) (voir figure 2.11).



**Figure 2.11 : Protocole de négociation avec des offres non alternatives entre deux agents**

Le processus de négociation développé, procède par échange de propositions ( $\phi$ ), des critiques, des explications et de m étainformation entre les agents. Il commence à l'état 0 (State0), quand un agent (*agent a*) fait une proposition à un autre agent (*agent b*), noté par  $\text{proposal}(a, b, \phi)$ , la proposition  $\phi$ , inclut les conditions de la proposition. Une fois la proposition envoyée (State1), le premier agent peut faire une seconde proposition sans attendre la réponse. Le second agent peut réagir en acceptant la proposition, ou en faisant une critique ou une contre-proposition ou en se retirant du processus. Si une critique (State2) ou une contre-proposition (State3) est faite, l'agent peut se retirer ou continuer le processus en faisant d'autres propositions, ce qui le ramène à l'état initial. Le processus continue jusqu'à ce qu'un de ces agents accepte ou se retire. Il faut souligner que ce PN est un

modèle non alternatif d'offres, parce qu'un agent peut faire une contre-proposition sans attendre la réponse de l'autre agent à la première proposition. Les explications sont des informations additionnelles ou une forme de meta-information, qui rend intelligible les propositions, les critiquent et font des contre-propositions (dans ce PN ils ne sont pas explicitement inclus).

Finalement, dans ce même type de protocole de négociation nous pouvons mentionner les protocoles à base de jeux de dialogues ou «Dialogue Games» entre agents. Un jeu de dialogue est un ensemble de règles qui définissent le comportement de l'agent en fonction de son contexte cognitif. Lorsque l'état mental d'un agent correspond à l'état requis par un jeu, l'agent peut entrer dans ce jeu et son comportement est dicté par les règles qui définissent le jeu [Levin et Moore, 1980]. Les jeux de dialogue modélisent le comportement des agents lors du dialogue. Ils doivent donc spécifier :

- le passage d'un jeu à un autre (conditions de changement de jeux),
- le traitement des coups et des états mentaux (règles des jeux).

De plus, les jeux peuvent être imbriqués pendant le dialogue, les agents peuvent donc jouer plusieurs jeux en même temps (ce qui permet par exemple de décomposer des problèmes en sous-problèmes dans un dialogue coopératif) [Dastani et al., 2000].

#### *4.7.1.4 PN à base de détermination de stratégies*

Une fois les préférences relatives à un agent et le protocole de négociation choisis, il reste à déterminer quel type de stratégie doit être utilisé par un agent durant le processus de négociation. Parmi les approches employées actuellement figure encore le raisonnement à base de cas [Sycara, 1990], l'apprentissage adaptatif / l'approche évolutionniste et la théorie des jeux. Dans ce qui suit, nous allons préciser quelques points de stratégie.

##### *4.7.1.4.1 Raisonnement à Base de Cas*

Sycara introduit une approche de modélisation de négociation, nommée PERSUADER, fondée sur le raisonnement à base de cas, «cased-based reasoning» (CBR) [Sycara, 1988 ; Sycara, 1990]. L'idée première d'un CBR est d'effectuer un raisonnement en se basant sur les cas passés similaires au cas actuel rencontré. Dans PERSUADER, chaque cas décrit les attributs des participants à la négociation, leurs « settlement » (engagements, accords), le contexte de leurs désaccords, les processus dans lesquels l'accord entre les parties a réussi ou a échoué. Il est utilisé pour construire et proposer un compromis d'engagement approprié en cas de dispute entre les parties.

#### 4.7.1.4.2 Apprentissage adaptatif / approche évolutionniste

Certains auteurs emploient l'apprentissage adaptatif [Priest, 1998] ou l'approche évolutionniste [Gerding, 1999] pour déterminer la meilleure stratégie de négociation. Dans l'approche apprentissage adaptatif, les agents utilisent des règles d'apprentissage basées sur le déroulement de leurs stratégies antérieures pour adapter leurs stratégies actuelles. L'approche évolutionniste utilise les algorithmes génétiques pour faire évoluer les stratégies de négociation pour les agents. Le classifieur génétique peut être considéré comme un exemple pour ce type d'approche [Goldberg, 1994 ; Reaidy, 1999].

#### 4.7.1.4.3 La théorie de jeux

La théorie des jeux est la branche des mathématiques appliquées, qui a été développée pour analyser les situations avec des interactions entre des entités ou agents avec des intérêts semblables, opposés, ou mélangés. Elle a vu le jour grâce à l'ouvrage désormais classique « The Theory of Games and Economic Behaviour », écrit par Von Neumann, publié en 1944 et qui jette alors les bases de la microéconomie.

Dans un jeu typique, la prise de décision des "joueurs", ayant chacun leurs propres buts, aide à prévoir les décisions de l'un par rapport aux autres. La solution du jeu est la conséquence des décisions prises par les joueurs. La théorie des jeux a été appliquée dans toutes les activités dites stratégiques, parce qu'elles portent sur la coordination et l'agencement des forces face à l'adversité. Toutes les disciplines sont concernées, qu'il s'agisse de l'informatique, de la biologie, de la psychologie, de la philosophie, de la diplomatie, de la politique, des choix industriels, ou de la conduite des affaires en matière économique et militaire.

Son succès est dû au fait qu'il peut modéliser un ensemble important d'interactions entre de nombreuses d'entités. L'indépendance entre la structure d'un jeu et le contexte choisi permet d'avoir un outil théorique applicable dans n'importe quel type d'interactions entre les sociétés, les espèces, les individus, les compagnies, les bactéries ou les agents, etc. Ainsi, dans le système multi-agent, le jeu peut être vu comme un processus d'interaction entre les agents ou les joueurs.

Rosenschein, Genesereth et Zlotkin ont analysé plusieurs hypothèses sur le comportement et les interactions entre les agents et ont suggéré que l'approche de la théorie des jeux permet une grande flexibilité au niveau de la coordination dans les modèles d'agents autonomes [Rosenschein et Genesereth, 1985 ; Rosenschein et Zlotkin, 1994].

Cicirello a développé une série de systèmes multi-agents inspirés des comportements sociaux d'insectes pour modéliser le problème des cheminements dynamiques des produits dans un atelier de production « shop floor routing ». Les interactions entre les agents ont été calquées sur la théorie des jeux et adaptées comme un jeu en forme normale [Cicirello, 2000 ; Cicirello et Smith, 2001].

Le « dilemme itéré du prisonnier », exemple issu de la théorie des jeux, a été utilisé pour modéliser et étudier la coopération entre agents. Le but de l'application de ce modèle est de proposer de nouveaux critères de qualité pour les stratégies à utiliser par les agents pour favoriser l'émergence et l'évolution de la coopération entre eux [Bicchieri et Green, 1997 ; Bicchieri et al., 1998 ; Beaufils et al., 1998 ; Beaufils, 2000].

#### 4.7.1.5 Discussion

Parmi les quatre types de PN détaillés, le PN à base d'enchères est considéré comme le protocole le plus simple à implémenter. Ses inconvénients se résument à ses limitations pour prendre en considération toutes les contraintes imposées par le système. Le PN à base d'argumentations présente une complexité assez importante au niveau de la modélisation des mécanismes de raisonnement sur l'état mental propre d'un agent par rapport aux autres agents dans le système. Concernant le PN à base de médiation, la résolution des conflits entre agents durant leurs négociations et la possibilité de prendre en considération un grand nombre de contraintes du système en sont les principaux avantages, tandis que la multiplication des interactions entre agents peut être considérée comme un inconvénient. Finalement, le PN à base de détermination de stratégie présente l'avantage d'exploiter et d'explorer de nouveaux mécanismes économiques à base de coopération et de la théorie des jeux entre agents, tandis que son inconvénient réside au niveau de la possibilité d'introduire et d'adapter ces différents mécanismes dans des applications réelles. Dans la thèse nous avons utilisé un nouveau PN qui consiste à combiner les deux PN à base de médiation et de détermination de stratégie afin de profiter, simultanément des avantages de ces deux PN.

Avant de poursuivre par la partie langage de communication pour les PN, nous devons d'expliquer, dans les paragraphes qui suivent, quelques concepts de la théorie des jeux utilisés dans des nouveaux PN développés dans le cadre de cette thèse.

#### 4.7.1.6 Quelques éléments de la théorie des jeux

Sans revenir sur la définition donnée précédemment, nous allons détailler un peu mieux ces éléments pour définir nos choix stratégiques et tactiques.

##### 4.7.1.6.1 Les différents types de jeux

Un jeu est dit à information *complète* si chacun des participants connaît :

- ses possibilités d'actions (l'ensemble des choix qu'il peut faire),
- l'ensemble des choix des autres joueurs,
- les issues possibles et la valeur des gains qui en résultent,
- les motifs des joueurs : chacun sait se mettre à la place des autres et sait ce que l'autre déciderait s'il était dans la même situation.

On dit qu'un jeu est à information incomplète s'il manque de l'information (lorsqu'il n'y a pas de connaissance des gains, ou de certaines règles). Les jeux de dés, bridge, poker etc. sont classés dans cette catégories.

Dans les jeux à information complète, l'ordre des coups permet de distinguer deux types de jeu :

1. s'il y a *simultanéité des coups*, comme dans le jeu d'enfant où chacun choisit simultanément ciseaux / pierre / papier ou pair / impair ou le dilemme de prisonnier. On ne peut alors pas se décider en fonction de ce que joue l'adversaire puisqu'on joue en même temps. On dit qu'il y a *information imparfaite*,
2. Dans le cas du jeu d'échecs ou de dames, par contre, les coups n'étant pas simultanés, mais *successifs*, on dispose d'une information supplémentaire qui est le coup de l'adversaire. On dit alors qu'il y a *information parfaite*.

#### 4.7.1.6.2 Somme des jeux

##### *Jeux à somme nulle*

Se dit des jeux dans lesquels la somme des gains de tous les joueurs pour chaque situation est nulle. En d'autres termes, tout gain positif pour certains joueurs est compensé par autant de perte pour d'autres joueurs.

##### *Jeux à somme non nulle*

Il s'agit des jeux dans lesquels la répartition des gains et des pertes peut ne pas être nulle. Par exemple, on a longtemps considéré que les rapports liant deux partenaires économiques étaient « formalisables » à l'aide de jeux à somme nulle. Cela sous-entendait qu'il devait y avoir un « gagnant » et un « perdant ». Depuis, on a introduit la notion de gains économiques partagés pour lesquels les deux acteurs peuvent être gagnants sous certaines conditions.

#### 4.7.1.6.3 Jeux coopératifs et non coopératifs

Au niveau des jeux *coopératifs*, on peut assimiler la coopération grâce à des contrats qui ne peuvent pas être remis en cause. Les joueurs peuvent également transférer les gains d'un joueur à l'autre. Dans les cas extrêmes de coopération où les gains sont répartis équitablement entre les joueurs (on recherche alors une somme de gains maximum), on peut considérer les joueurs comme un joueur unique qui chercherait à dégager un *intérêt général* et qui ensuite répartirait les gains entre les joueurs.

Partant de l'hypothèse que chaque joueur garde sa liberté d'engagement, l'objectif de la théorie des jeux *non coopératifs* est de caractériser les issues possibles d'une interaction stratégique lorsque les joueurs abordent cette interaction de manière rationnelle, c'est-à-dire finalement de la manière la plus *égoïste* qui soit (ils

veulent seulement maximiser leur propre bien). Les jeux non coopératifs ont les caractéristiques suivantes :

- un petit nombre d'agents (les joueurs) interagissent,
- les décisions de chaque agent influencent les gains des autres,
- chaque agent dispose au moment de prendre sa décision d'une certaine information, dont il faut tenir compte
- le déroulement des décisions dans le temps (décisions simultanées ou séquentielles) doit également être intégré.

Les décisions simultanées sont généralement représentées sous la forme d'un tableau (jeux en forme normale) et les décisions séquentielles, sous la forme d'un arbre de jeu (jeux en forme extensive). Dans cette partie, nous détaillons seulement les jeux en forme normale, dilemme de prisonnier, équilibre de Nash, l'optimum Pareto et les autres principes liés à cette forme de jeux, utilisé dans le chapitre 4 de la thèse, pour plus de détails sur les autres formes de jeux ou d'autres principes dans la théorie de jeux on peut consulter [Osborne et Rubinstein, 1994 ; Andreu et al., 1995].

#### 4.7.1.6.4 Jeux en forme normale

Dans ce qui suit, nous allons aborder quelques définitions, pour introduire les composantes mises en œuvre pour caractériser certaines propriétés d'un jeu.

**Définition 1** un jeu est décrit par les éléments suivants :

1. un ensemble de  $N$  joueurs :  $I = \{1, 2, \dots, N\}$ ,
2. pour chaque joueur  $i$ ,  $i \in I$ , un ensemble de stratégies  $A^i$  qui contient toutes les stratégies possibles de ce joueur.  $a^i \in A^i$  est une stratégie particulière du joueur  $i$ . Par conséquent,  $A^i = \{a_1^i, a_2^i, \dots, a_{k_i}^i\}$  si  $K^i$  stratégies sont disponibles pour le joueur  $i$ . Si chaque joueur  $i$  choisit une stratégie  $a^i$  nous pouvons représenter le résultat (ou profil de stratégies) du jeu par un vecteur qui contient toutes ces stratégies :  $a \equiv (a^1, a^2, \dots, a^N)$ .
3. pour chaque joueur  $i$  une fonction de gain,  $\pi^i$ , qui donne la valeur pour le joueur  $i$  de chaque résultat du jeu :  $\pi^i(a)$ . C'est un nombre réel :

$$\pi^i : A = \prod_{i \in I} A^i \rightarrow R$$

$$a \equiv (a^1, a^2, \dots, a^N) \rightarrow \pi^i(a)$$

**Exemple** le dilemme du prisonnier

Deux individus (Jacques et Pauline) sont arrêtés par la police pour complicité de vol avec arme. Ils sont enfermés dans deux cellules séparées sans possibilité de communication. Chaque prévenu est interrogé séparément et a le choix entre nier d'avoir commis le vol ou avouer l'avoir commis avec son complice. Nous avons donc un jeu non-coopératif avec  $N = 2$  joueurs,

$$I = \{1, 2\} = \{Jacques, Pauline\}.$$

L'ensemble de stratégies de chaque joueur est  $A^1 = A^2 = \{nier, avouer\}$ . Il y a donc 4 résultats possibles du jeu :

$$A = \left[ \begin{array}{l} (a^1 = nier, a^2 = nier), (a^1 = nier, a^2 = avouer), \\ (a^1 = avouer, a^2 = avouer), (a^1 = avouer, a^2 = nier) \end{array} \right]$$

Les gains des individus représentent la situation qui résulte des années de prisons auxquelles ils peuvent être condamnés en fonction de leurs aveux. La durée de la peine est négativement liée aux déclarations faites au cours de l'enquête. Ainsi :

- si Jacques et Pauline avouent tous les deux leur crime, ils seront condamnés à 8 ans de prison,
- s'ils nient tous les deux, ils écoperont d'1 année de prison en raison de l'absence de preuves accablantes,
- si un seul avoue, il sera relâché en récompense de sa coopération et l'autre sera condamné à 10 ans de prison.

Nous avons donc les gains (symétriques) suivants :

$$\begin{aligned} \pi^1(nier, nier) &= \pi^2(nier, nier) = -1, \\ \pi^1(nier, avouer) &= \pi^2(avouer, nier) = -10, \\ \pi^1(avouer, nier) &= \pi^2(nier, avouer) = 0, \\ \pi^1(avouer, avouer) &= \pi^2(avouer, avouer) = -8. \end{aligned}$$

Nous pouvons alors représenter ce jeu en forme normale, sous la forme d'un tableau :

		<b>Pauline</b>	
		<i>nier</i>	<i>avouer</i>
<b>Jacques</b>	<i>nier</i>	(-1,-1)	(-10,0)
	<i>avouer</i>	(0,-10)	(-8,-8)

**Tableau 2.2 : Dilemme du prisonnier**

Remarques :

1. il ne faut pas confondre la stratégie d'un joueur individuel  $a^i$  et le résultat  $a$  qui est une combinaison particulière des stratégies de tous les joueurs.
2. dans notre définition d'ensemble de stratégies, il y a un nombre fini de  $k^i$  stratégies pour chaque agent. Mais en économie, les ensembles de stratégies sont en général continus et contiennent une infinité de stratégies possibles (choix de quantités, de prix, etc...).

3. les gains représentent en général des utilités ordinales et non des sommes monétaires. En économie industrielle, néanmoins, les gains des firmes correspondent souvent à des profits.

La formulation sous la forme d'un jeu permet de clarifier une situation conflictuelle. Il nous faut, en plus comprendre à quel type de solution ce jeu peut nous conduire. Pour déterminer cette solution, nous devons étudier l'équilibre du jeu.

### *Concepts d'équilibre*

Parmi l'ensemble des résultats possibles nous devons déterminer ceux auxquels le jeu peut aboutir : les résultats d'équilibre. Nous pouvons alors prédire les situations auxquelles ce jeu va conduire. La solution idéale correspond à un équilibre unique. Dans ce cas nous pouvons précisément prédire la solution de cette situation conflictuelle. Néanmoins, on a souvent des équilibres multiples. Parfois il n'existe même pas d'équilibre. Dans ce paragraphe nous intéressons à l'équilibre de Nash. Avant de passer à l'étude de cet équilibre, introduisons une notation supplémentaire. Considérons le résultat du jeu qui contient les stratégies de tous les joueurs sauf  $i$ . Nous pouvons alors le noter de la manière suivante :

$$a^{-i} = (a^1, a^2, \dots, a^{i-1}, a^{i+1}, \dots, a^N), \quad a^{-i} \in \prod_{j \neq i} A^j$$

$$\Rightarrow a = (a^i, a^{-i})$$

Nous pouvons maintenant introduire les concepts de l'équilibre de Nash.

### *Equilibre de Nash (EN)*

L'équilibre de Nash est l'un des *concepts clés* dans la théorie des jeux défini par John Nash : il permet en effet de trouver les solutions à un jeu que l'on pose en forme normale (sous forme de tableaux de gains) et lorsque l'on a un jeu à information complète et imparfaite. C'est le concept d'équilibre le plus couramment utilisé, il est à la base d'autres concepts d'équilibre qui ont été développés plus récemment.

**Définition 2** Un résultat  $a^* = (a^{*1}, \dots, a^{*N})$  où  $a^{*i} \in A^i \quad i = 1 \dots N$ , est un équilibre de Nash si aucun joueur n'a intérêt à dévier unilatéralement de sa stratégie  $a^{*i}$  quand les autres joueurs continuent à jouer  $a^{*-i}$ . Par conséquent, pour tout joueur  $i = 1 \dots N$  nous devons avoir :

$$\pi^i(a^{*i}, a^{*-i}) \geq \pi^i(a^i, a^{*-i}), \quad \forall a^i \in A^i$$

Pour tester si un résultat  $a$  est un équilibre de Nash, nous devons vérifier si un des joueurs au moins n'a pas intérêt à choisir une autre stratégie. Si ce n'est pas le cas alors  $a$  est un EN. L'EN n'est pas nécessairement unique dans un jeu. Dans l'exemple du dilemme du prisonnier :

$(nier, nier)$  n'est pas un équilibre de Nash car :

$$\pi^1(nier, nier) = -1 < 0 = \pi^1(avouer, nier)$$

$(avouer, avouer)$  est un équilibre de Nash car :

$$\pi^1(avouer, avouer) = -8 > -10 = \pi^1(nier, avouer)$$

$$\pi^2(avouer, avouer) = -8 > -10 = \pi^2(avouer, nier)$$

Les concepts d'équilibre correspondent à des mécanismes particuliers de coordination des stratégies individuelles. Dans les jeux non-coopératifs, chaque joueur cherche unilatéralement à améliorer sa situation individuelle. Est-ce que la solution qui est donnée par l'équilibre est un mécanisme de coordination efficace ? Pour répondre à cette question nous allons définir le concept d'optimum de Pareto.

**Définition 3** Le résultat  $\hat{a}$  Pareto-domine le résultat  $a$  si :

$$\pi^i(\hat{a}) \geq \pi^i(a), \quad \forall i \text{ et } \exists j, \quad \pi^j(\hat{a}) > \pi^j(a).$$

Un résultat  $a^*$  est un optimum de Pareto s'il n'existe aucun autre résultat qui le « Pareto-domine ».

Dans le dilemme du prisonnier  $(avouer, avouer)$  est un EN mais le résultat  $(nier, nier)$  Pareto-domine cet équilibre.

Ces différents éléments de la théorie de jeux ont été détaillés parce que ce principe a été appliqué dans des protocoles de négociations conçus dans la thèse (voir chapitre 4).

#### 4.7.2 langage de communication dans les protocoles de négociations

En théorie de la communication, il existe principalement trois dimensions du langage : syntaxique, sémantique et pragmatique. Si la plupart des langages informatiques traitent des deux premières, la dimension pragmatique est aussi nécessaire pour structurer des conversations entre agents. C'est pourquoi les efforts du DARPA ont conduit successivement à la spécification des langages KIF (Knowledge Interchange Format) permettant de décrire et de représenter des connaissances au niveau syntaxique nous distinguerons par exemple :

- ONTOLINGUA permet de décrire des connaissances partagées en terme d'ontologie (niveau sémantique),
- KQML (Knowledge Query and Manipulation Language) définit un langage et un protocole de haut-niveau pour échanger et interpréter des connaissances (niveau pragmatique),

- STROBE (STReams, OBjects and Environments) : présente un modèle de Conversation inter Agent. Il est centré sur le besoin fondamental d'avoir un modèle de dialogue du partenaire pour être en mesure de comprendre quel message suivra dans le cours du dialogue (niveau sémantique et pragmatique) [Cerri, 1996 ; Cerri, 1997].

Cette partie n'est pas détaillée dans la thèse parce qu'elle n'est pas exploitée dans les contributions effectuées dans notre travail de recherche. Finin et Labrou ont proposé une étude complète du langage de communication pour les PNs [Finin et Labrou, 1999]. La standardisation de protocoles et de langages est en cours à la communauté FIPA [FIPA, 2002].

#### 4.7.3 Plateformes multi-agents

Un certain nombre d'environnements de développement a vu le jour ces dernières années. La liste que nous donnons ici est loin d'être exhaustive et Internet est riche d'une quantité de travaux importante sur le sujet (voir tableau 2.3). Il faut remarquer que la plupart sont développés autour de Java pour profiter notamment des possibilités d'interopérabilité offertes par ce langage. Elles sont souvent dédiées à un domaine d'application ou à un modèle d'agent particulier.

Plate-forme	URL
AGLETS	<a href="http://www.trl.ibm.com/aglets/">http://www.trl.ibm.com/aglets/</a>
FIPA-OS	<a href="http://www.nortelnetworks.com/fipa-os">http://www.nortelnetworks.com/fipa-os</a>
GRASSHOPER	<a href="http://www.grasshopper.de/">http://www.grasshopper.de/</a>
JADE	<a href="http://sharon.csel.it/projects/jade">http://sharon.csel.it/projects/jade</a>
JAFMAS	<a href="http://www.ececs.uc.edu/~abaker/JAFMAS/">http://www.ececs.uc.edu/~abaker/JAFMAS/</a>
JATLITE	<a href="http://java.stanford.edu/">http://java.stanford.edu/</a>
MADKIT	<a href="http://www.madkit.org">http://www.madkit.org</a>
ODISSEY	<a href="http://www.genmagic.com/agents/odyssey.html">http://www.genmagic.com/agents/odyssey.html</a>
SWARM	<a href="http://www.swarm.org/intro.html">http://www.swarm.org/intro.html</a>
ZEUS	<a href="http://www.labs.bt.com/projects/agents/zeus/">http://www.labs.bt.com/projects/agents/zeus/</a>

**Tableau 2.3 : Liste non exhaustive des plate-formes multi-agent**

En grande majorité, les plates-formes SMAs sont issues de projets de recherche, mais on trouve aussi des produits commerciaux. Certains d'entre eux sont orientés vers une communication entre systèmes distribués (par exemple [Madkit](#), JAFMAS, Hive), d'autres sont axés vers la construction de modèles de simulation (par exemple [Swarm](#)).

En règle générale, les plates-formes disponibles actuellement sont difficiles à mettre en œuvre car elles ne couvrent pas les fonctionnalités requises par les environnements industriels actuels (globalisation de l'économie, E-business, intégration des réseaux internet, systèmes de production et de décision distribués et

délocalisés) et ne possèdent pas une fiabilité suffisante. Pour plus de détails sur ce sujet, la thèse de LIU offre un aperçu plus complet [Liu, 2002].

## 5 Conclusion

Dans ce chapitre, nous avons présenté une vue détaillée de deux domaines clés pour la conduite et le pilotage de système de production dans un environnement dynamique situé.

Dans le premier domaine, nous avons détaillé les différents concepts et approches pour la gestion de la production au niveau organique et fonctionnel. La structure décentralisée a été préférée en raison de sa réactivité, flexibilité et adaptabilité dans les environnements dynamiques et ouverts.

Dans le deuxième domaine, nous avons abordé l'utilisation des systèmes multi-agent pour le pilotage décentralisé du système de production. En effet, grâce aux caractéristiques propres des agents au niveau de la modularité, la décentralisation, l'adaptabilité, mais aussi de leur potentiel à pouvoir représenter et faire fonctionner des réseaux programmables, nous sommes en mesure de justifier l'utilisation des agents pour le pilotage décentralisé des systèmes de productions.

Entre modèle d'agent réactif, cognitif et hybride nous emploierons un modèle d'agent évolutif qui peut être classé « hybride » pour profiter, en même temps, des avantages de la réactivité et de la rationalité chez un agent.

Enfin, parmi les plusieurs types de protocoles de négociation cités dans ce chapitre, nous utiliserons un nouveau protocole qui consiste à combiner ceux à base de médiation et la théorie des jeux pour déterminer les stratégies à employer entre agents.



# **Chapitre 3 : Approche décentralisée pour le pilotage et la conduite des systèmes de production**

## **1 Introduction**

Dans cette partie, nous proposons des approches décentralisées à base de gestion inverse de production. Le principe de l'ordonnancement est remplacé par la configuration automatique des ressources pour faire face à la dynamique et aux perturbations liées aux besoins des clients et aux exigences du marché. L'approche étudiée est fondée sur l'auto-organisation entre les entités autonomes du système. Ensuite, le projet européen PABADIS illustrera cette architecture de pilotage et de conduite décentralisée.

## **2 Approche de solutions : la gestion inverse de production**

Suite aux différents besoins et challenges mentionnés dans l'introduction de la thèse, et en tenant compte des difficultés actuelles rencontrées pour répondre à ces différents besoins, de nouvelles approches doivent être étudiées et appliquées dans nos futurs systèmes de production. Parmi les points évoqués antérieurement nous considérons dans notre travail les exigences et les besoins des clients par rapport à la production et nous essayons d'y répondre par le développement des nouvelles approches au niveau structurel et fonctionnel du système.

En effet, les exigences du marché au niveau compétitivité, dynamique, réactivité et urgence demandent une réponse rapide et flexible. La notion de « temps limité de réponse » est primordiale, mais nous n'avons pas suffisamment de temps pour faire la planification et l'ordonnancement à l'avance. Pour cela, il faut décentraliser la structure de décision et partir de ce qui existe dans le système en essayant de satisfaire le plus vite possible les besoins. La demande est clé, pour cela les étapes les plus proches de l'expédition du produit qui priment. Au niveau fonctionnel nous travaillons en reconfiguration permanente des moyens/systèmes de production par l'intermédiaire de l'allocation dynamique et en temps réel des ressources. Dans ce qui suit nous essayons de justifier la substitution de l'ordonnancement par la reconfiguration dynamique.

### **2.1 Ordonnancement ou Reconfiguration ?**

Dans le domaine de la gestion d'un système de production complexe et décentralisé, un problème consiste à déterminer comment appliquer un ordonnancement réaliste, avec des capacités limitées et variables, et une demande fluctuante. En général, certains spécialistes tentent de concevoir et développer des

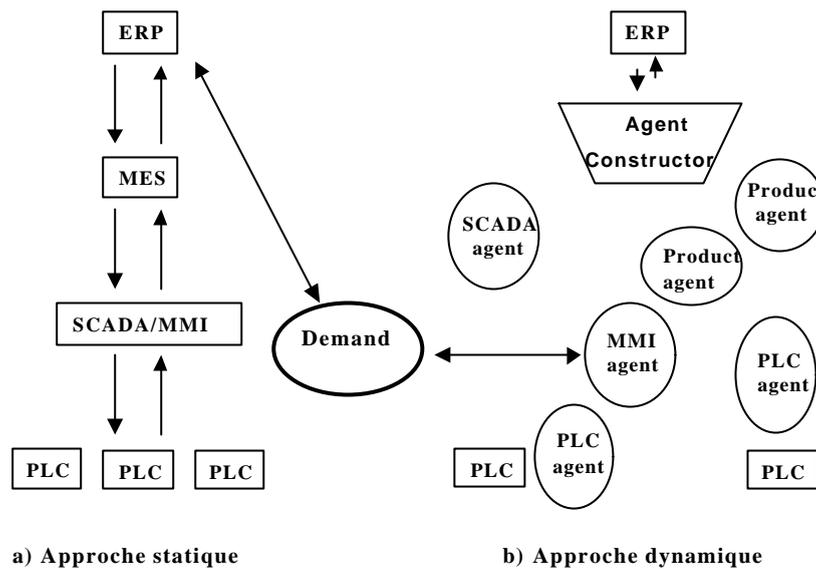
solutions compliquées, qui donneront de toute façon des réponses obsolètes, au moment où elles seront sollicitées, compte tenu des perturbations permanentes auxquelles un système de production est soumis. Ces solutions consistent avant tout à effectuer de l'allocation de tâches sur un ensemble de ressources identifiées en termes de quantité et de spécificité. On adapte donc et/ou on ordonne dans le temps des produits (ordres de fabrication) compte tenu des contraintes fortes existantes au niveau des ressources / outils de la production. Une autre forme de raisonnement consiste à porter son attention sur les ressources et les capacités d'un système à se reconfigurer ; dans ce cas, nous pourrions dire : si l'on est en mesure de reconfigurer en continu (en temps réel) les ressources d'un système de production, les mobiliser et les adapter à chaque demande versatile. La contrainte n'est plus au niveau des ressources ; elle est reportée sur la demande, et il n'y a pas de problème d'ordonnancement. Il s'agit donc de résoudre un problème d'allocation de ressources. Tout problème d'ordonnancement se ramène à un problème de reconfiguration dynamique du système de production.

En termes de méthodes, la configuration automatique d'un système et/ou sa reconfiguration sont considérées comme une amélioration majeure. Cette démarche est particulièrement bien adaptée à la production personnalisée de masse "tirée" et "contrainte" par la demande. Par contre celle-ci implique la mise en œuvre de nouvelles techniques que nous appellerons "gestion inverse de production" qui remplace un ordonnancement conventionnel. C'est la seule façon, à notre avis, d'assurer une réactivité telle que requise par les nouveaux environnements industriels.

## **2.2 Solution inverse**

Le comportement d'un système de production est issu de sa structure avec la connaissance, ou les mécanismes fonctionnels, inclus dans les interactions entre les ressources (que nous modélisons par des entités nommées agents), plutôt que le résultat direct d'une fonction donnée prédéfinie et complexe. Ainsi, la manière de contrôler un système de production est totalement différente de ce que nous considérons habituellement dans l'industrie. Dans la figure 3.1, nous comparons deux approches fonctionnelles, sur l'exemple du modèle de projet européen PABADIS, qui sera détaillé dans ce chapitre.

Dans la partie 'a' de la figure 3.1, nous pouvons observer une organisation déterministe, statique et conventionnelle du travail; la connaissance / savoir-faire et le traitement associé sont hiérarchiquement structurés dans les modules spécifiques et les couches. Dans cette approche nous distinguons les modules suivantes : Enterprise Resource Planning (ERP), Manufacturing Execution System (MES), Supervisory Control And Data Acquisition (SCADA), Man-Machine Interface (MMI) and Programmable Logic Controllers (PLC's). Les demandes et/ou les ordres de client ont un seul point d'entrée en haut de ce graphique et la production est organisée d'une manière de haut en bas ou « Top-down way ».



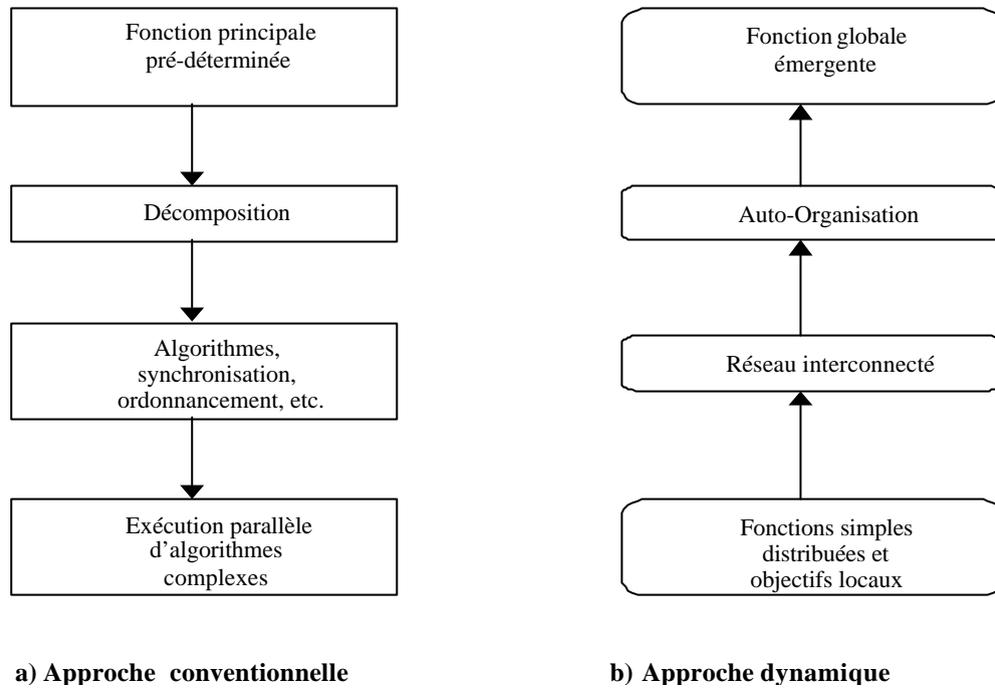
**Figure 3.1 : Approches conventionnelles pour les systèmes de production**

Dans la partie ‘b’ de la figure 3.1, il y a un découplage entre ERP et le niveau opérationnel comprenant les mêmes modules de SCADA et de PLC. Les liens entre les modules sont supprimés puisqu'ils ne sont plus autonomes et ils peuvent communiquer directement l'un avec l'autre. Cependant, pour des raisons pratiques nous pouvons définir un échange local entre les modules voisins. Ici, le modèle fonctionnel est différent; les interactions entre les modules ou agents sont dynamiques et font appel à des mécanismes comme la coopération et/ou la compétition pour le pilotage du système.

D'un point de vue ‘logique’, le mode de raisonnement sous-jacent à chacune de ces approches est très différent (voir figure 3.2) :

- a- dans la partie gauche de la figure 3.2 (approche conventionnelle), on implique généralement des fonctions statiques (en termes de transformation ou de valeur ajoutée au produit). Cela consiste à analyser, à étudier et à décomposer l'ensemble du système considéré en fonctions et sous fonctions (principe de décomposition). On simplifie ensuite le processus et on l'automatise grâce à l'emploi des technologies de l'information en exécutant les fonctions ainsi décrites sur des machines parallèles (éventuellement),
- b- dans la partie droite de la figure 3.2 (approche dynamique), on considère le système de production comme un réseau de ressources distribuées dans lequel s'exercent des interactions entre agents autonomes. Ce qui est important est de décrire la nature exacte de ces interactions. Ce réseau interconnecté sera soumis à des phénomènes d'auto-organisation, suivant les règles en vigueur dans les automates cellulaires. Au lieu de procéder à un ordonnancement détaillé, l'appel des pièces se fera de manière ascendante ‘upward’ à partir d'une demande client initiale appliquée à la sortie du modèle. La mobilisation

des ressources est effectuée de manière locale en fonction des besoins immédiats (ordres de réapprovisionnement aval). Le rôle de la planification est réduit à celui de préparation ou de pré-organisation des ressources dans le système de production.



**Figure 3.2 : Approches mises en œuvre pour la gestion d'un système complexe**

Dans ce qui suit, nous détaillons l'architecture de PABADIS appliquant une approche dynamique pour le pilotage des systèmes de production.

## **3 Exemple d'architecture de pilotage décentralisé : PABADIS**

### **3.1 Introduction**

PABADIS est un projet européen conçu pour définir les principes des systèmes de production à venir, basés sur l'organisation dynamique des ressources dans un environnement changeant et distribué [PABADIS, 2002d]. Pour cela, une nouvelle approche est définie pour un pilotage décentralisé de la production. La mise en œuvre de cette approche est réalisée à travers des plate-formes d'agents mobiles. En effet, les agents représentent les différentes composantes organiques et physiques du système dont le fonctionnement est fait à travers des interactions auto-organisationnelles et coopératives. Dans ce qui suit, nous donnerons un aperçu de l'idée PABADIS, la structure des agents, du cycle de production, de l'allocation des ressources et des plate-formes utilisées dans ce projet.

### **3.2 Contexte, objectifs et partenaires**

Dans un marché en constante évolution, piloté par la technologie, les industries se situent aujourd'hui dans un environnement turbulent, caractérisé par :

- une difficulté à prévoir les événements,
- des cycles de production courts, imposés par la nécessité d'innovation,
- une personnalisation des produits, et donc une extrême variabilité des gammes et des volumes de production.

Dans le même temps, l'apparition des réseaux à tous les niveaux d'une entreprise pour le management, la logistique, la production, la vente, amorce une tendance vers une économie globale et distribuée, où compétition et coopération s'organisent dynamiquement à travers des entreprises virtuelles.

Dans ce contexte, le projet Européen PABADIS (Plant Automation BAsed on DIstributed Systems) s'attache à concevoir et développer des outils de gestion de production basés sur de nouveaux paradigmes. PABADIS est un projet financé par la Communauté européenne dans le cadre du programme Information Society Technology (IST- #60016). Il tente de définir les principes des systèmes de production à venir, basés sur l'organisation dynamique des ressources dans un environnement changeant et distribué. En termes d'architecture, il s'adresse à des systèmes de production flexibles et re-configurables, ouverts, connectés en réseau et dont l'intelligence est distribuée. Sur le plan technique, PABADIS s'appuie sur des outils de reconfiguration automatique, sur le modèle « Plug-and-Participate » ainsi que sur une technologie « multi-agents ».

Les partenaires de ce projet européen comprennent des laboratoires de recherche et des industriels. Il s'agit de :

- Ecoles des Mines d'Ales, LGI2P-EMA, France
- Université Otto Von Guericke de Magdeburg , Allemagne
- Institute of Strategic Manufacturing (IMS GmbH) , Allemagne
- JETTER AG , Allemagne,
- PHOENIX Electronics GmbH, Allemagne
- Technische Universitaet (ICT) de Vienne , Autriche
- Phillips-Universitaet Marburg (PUM), Allemagne
- UNISOFT Software Applications , Grèce
- HATZOPOULOS, Grèce
- P2I Engineering Productique et Informatique Intégrée, France

Associé au projet européen PABADIS, un consortium IMS (Intelligent Manufacturing Systems) a pour objectif de valider les concepts. Il se compose de :

- SUN Microsystems, USA
- Centor, Canada
- Prevost Motor Coach, Canada
- University of Calgary, Canada
- PebbleAge, Suisse
- Université de Genève (CUI), Suisse
- Université de Zurich (ZHW), Suisse

PABADIS peut être appliqué aux systèmes de production traitant des produits légèrement différents ou des produits avec une variation élevée, tous les deux avec un lot de petites tailles. Comme exemples typiques à ces applications nous pouvons citer les industries :

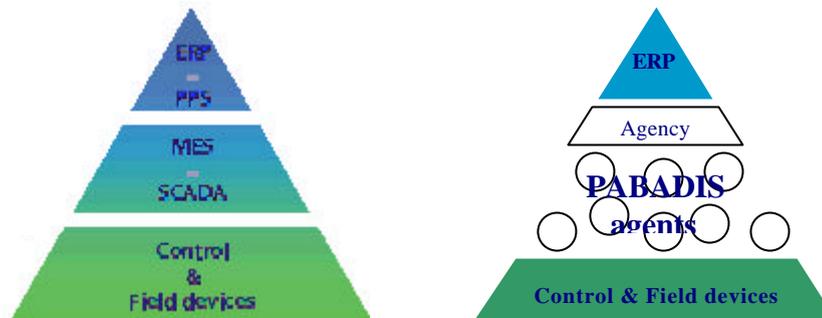
- de fabrication des meubles avec des matériaux de haute qualité,
- de fabrication de voiture avec un traitement spécial,
- industrie auxiliaires d'aéronautique,

En ce qui concerne l'industrie auxiliaire on peut affirmer que de telles compagnies fournissent normalement des marchandises à plus d'un client à partir de plus d'un champ industriel. Par conséquent, le nombre de variations de produit est élevé.

### **3.3 Aperçu de l'idée PABADIS**

En général, un système de production conventionnel possède une structure entièrement centralisée. Une telle structure s'étend depuis le progiciel de gestion intégrée des ressources (ERP, Enterprise Resource Planning), jusqu'au contrôle-commande des différentes unités et cellules de production, en passant par les

logiciels de contrôle d'exécution (MES, Manufacturing Execution System) et de supervision (SCADA, Supervisory Control And Data Acquisition). La figure 3.3 montre la structure conventionnelle conforme au principe CIM (Computer Integrated Manufacturing), comparée à la structure PABADIS à base d'agents.



**Figure 3.3 : Structure conventionnelle et structure PABADIS**

Dans l'approche PABADIS, la couche intermédiaire MES-SCADA a été remplacée par un ensemble d'agents autonomes et communicants, capables de gérer les flux de produits et de services. Fonctionnellement, l'exécution et la supervision restent assurées de façon globale et macroscopique ; elles ne sont cependant plus réalisées de façon hiérarchique, centralisée et planifiée, mais de façon indépendante, autonome, et réactive, à l'aide de processus élémentaires.

Les bénéfices attendus d'un tel mode de fonctionnement sont de plusieurs ordres :

- tirer parti de l'organisation en réseau pour faire une reconfiguration automatique de l'atelier ;
- gérer la complexité toujours croissante des opérations d'ordonnancement grâce à des règles d'allocation locales et une distribution des processus de décision ;
- augmenter l'efficacité de la production par une diminution des temps de réaction face à des situations imprévues ou conflictuelles ;
- avoir un suivi de la production «au plus près», grâce à des agents logiciels associés individuellement à un produit et responsables de sa réalisation.

### 3.3.1 Pourquoi des agents ?

La programmation orientée agent a été introduite récemment comme une extension à celle orientée objet. Elle confère à l'objet des propriétés d'autonomie d'exécution ainsi que des possibilités de mobilité du code tout à fait adaptées à des systèmes à intelligence distribuée. La technologie à base d'agents constitue une aide pour la conception des systèmes, offrant une approche naturelle et une appréhension aisée. Des agents peuvent être assignés à des instances physiques dont ils sont responsables, à des machines, des ressources ou des fonctions. Ils cohabitent au sein

d'un système multi-agents, communiquent et coopèrent afin de remplir leur mission. La mobilité du code apporte une flexibilité supplémentaire : les agents mobiles sont transférés de place en place, exécutant leurs tâches dans l'environnement local de l'hôte qui les reçoit.

L'utilisation de l'agent mobile est avantageuse au niveau technique (la mobilité de code) et conceptuel (les notions sociales de coopération et de compétition). En effet, même en cas de rupture ou de panne de réseau, l'emploi d'un agent mobile, équipé de toutes les informations sur l'ordonnancement et le plan d'exécution de ses tâches, permet la réalisation de ces dernières. Cette indépendance technique est importante dans l'industrie. En outre, la mobilité des agents fait émerger des comportements de coopération et de compétition entre eux. Ils peuvent être considérés comme des amis, pour donner de l'aide sur place à travers des stratégies coopératives ou « gentilles », ou comme des ennemis pour envahir l'environnement des autres à travers des stratégies compétitives ou « méchantes » tout en imposant leurs priorités, exigences et lois.

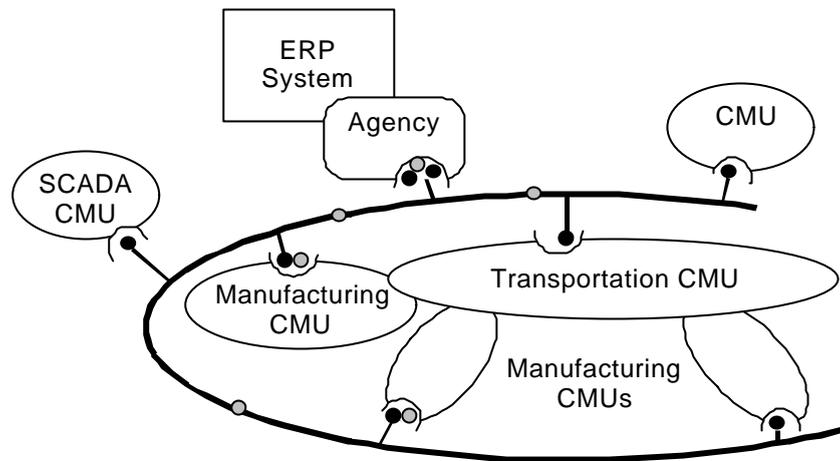
### 3.3.2 Topologie du système

Un système PABADIS est organisé autour d'unités de fabrication appelées CMU (Co-operative Manufacturing Units), qui fournissent au réseau des fonctions de fabrication. On distinguera ainsi :

- des CMU physiques, utilisés pour le traitement physique des produits. Ces unités peuvent être indifféremment des machines individuelles, des cellules flexibles ou des lignes de production.
- des CMU logiques, qui fournissent des services de calcul, d'archivage, etc.

L'ensemble des CMU se connecte à l'ERP par l'intermédiaire d'une interface appelée Agence (Agency), un CMU particulier qui offre un certain contrôle au niveau de la production (analyse des ordres de fabrication et leurs décomposition en ordre de travail, construction de report,...) et au niveau des agents (création, fin, ...). La partie Agence du projet a été développée par notre laboratoire LGI2P-EMA-Nîmes [PABADIS, 2002c]. Nous pouvons distinguer un autre type de CMU nommé SCADA CMU. Ce dernier type fournit un interface avec le système SCADA. Il traduit aux services utilisés directement par les agents les fonctions de ce système, comme l'acquisition des données, l'alarme, etc.

Un CMU comprend une, ou plusieurs fonctions fournissant des capacités manufacturières de production, un ensemble de support de services communs pour exploiter l'environnement de PABADIS et un Agent hôte. De plus, il peut inclure n'importe quelle ressource matérielle ou logicielle utile pour effectuer ses fonctions et ses services. Les CMU offrent leurs services à l'ensemble du réseau, utilisant une technique de PnP (Plug-and-Participate) qui permet à chaque CMU de se connecter ou de se déconnecter sans reconfigurer ni réinitialiser le système. La figure 3.4 donne un bref aperçu de ces composants.



**Figure 3.4 : Topologie d'un système de production PABADIS**

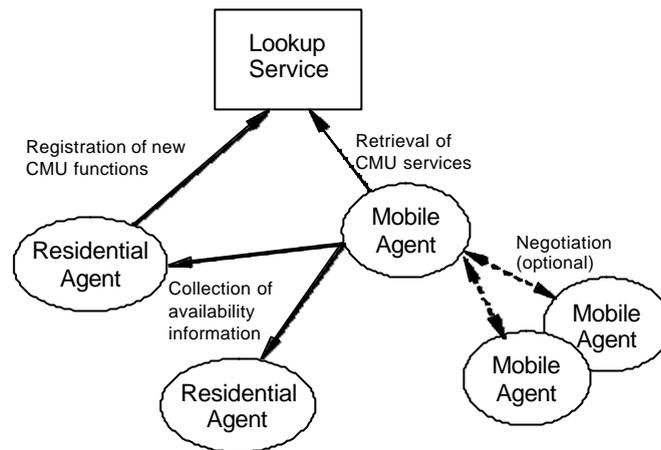
Les agents mis en jeu dans PABADIS sont de trois types :

- Les agents résidentiels sont attachés à un CMU et permettent à d'autres agents d'utiliser les ressources du CMU
- Les agents de production sont attachés à un produit et sont responsables de sa réalisation. Ces agents sont mobiles et se déplacent de CMU en CMU en accompagnant le produit en cours d'élaboration.
- Les agents de management (PMA) servent à collecter de l'information ou à contrôler des actions en cours.

### 3.3.3 le principe de Plug-and-Participate

Introduite par SUN avec le logiciel JINI, cette fonctionnalité permet, du point de vue de l'utilisateur, une organisation automatique des services en réseau, c'est-à-dire sans configuration manuelle des interfaces[PABADIS, 2002a].

Dans PABADIS, la fonction PnP met en oeuvre un service de « Look Up », dont la tâche est d'enregistrer les CMU actifs ainsi que les fonctions de production offertes. Le service de Look Up est ainsi un lieu d'échanges qui permet à un agent de production de trouver le service requis (figure 3.5).



**Figure 3.5 : Communication avec le service de Look Up**

### 3.3.4 Intégration des composants PABADIS

L'intégration d'une configuration PABADIS dans un système de production existant requiert la mise en place d'interfaces avec les autres outils de gestion de la production :

- L'interface avec un ERP est assurée par l'Agence, au moyen d'une base de données. Les ordres de fabrication, les requêtes particulières sont transmises sous forme de documents au format XML.
- L'interface avec un SCADA est gérée par un CMU particulier, qui met à jour la base de données du SCADA et lui transmet les requêtes.
- Enfin, l'interface d'un CMU avec une machine (par exemple un automate programmable) est constituée principalement :
  - d'une interface agent résidentiel – fonctions de contrôle-commande
  - d'une description des capacités de la machine

## 3.4 Structure des agents

Dans PABADIS, les agents sont considérés comme des programmes logiciels ayant un interface pour communiquer et pour fournir des services à la communauté. Cela reflète d'une façon générale le principe orienté-objet d'un programme logiciel (méthodes, données et interfaces pour les autres objets). Les méthodes et les données sont encapsulées par l'objet et ils sont accessibles seulement à partir d'un interface.

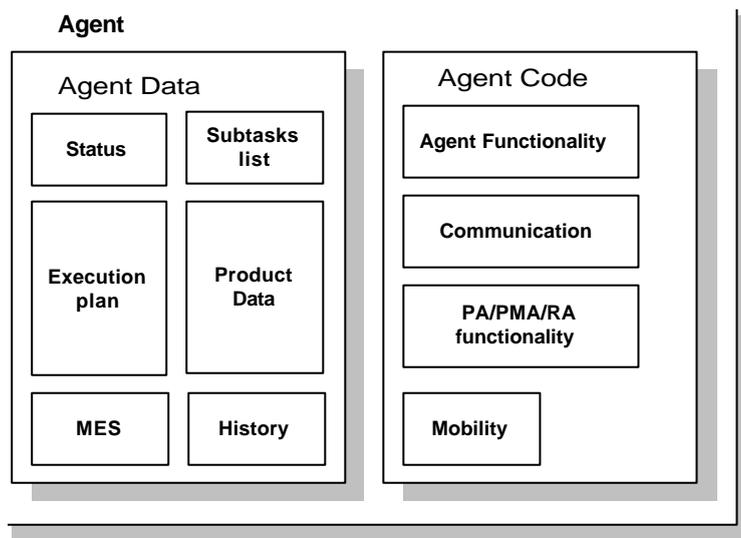
L'«Intelligence» du système figure seulement au niveau des agents de production, et les autres types d'agents résidentiels et PMA servent uniquement d'interface pour la communauté des agents de production. Un agent résidentiel

représente un CMU dans lequel il réside tandis qu'un PMA représente un interface pour le système ERP.

Chaque agent se compose de deux parties : code et données. Le code d'un agent inclut des méthodes standard, qui sont les mêmes pour tous les agents, et des méthodes spécifiques concernant des tâches concrètes pour un agent comme la mobilité, les services de communication, etc.. Les données dans un agent peuvent concerner son état, son historique, ses plans d'exécution, etc. (voir figure 3.6). Dans PABADIS, les agents sont codés en langage de programmation JAVA.

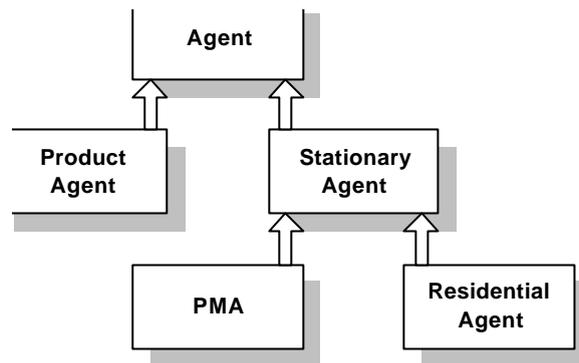
Tous les agents sont créés à partir d'une classe de base, nommée « Agent ». Cette classe inclut toutes les méthodes standard utiles pour un agent, et est aussi la super-classe pour tous les types d'agents dans PABADIS contenant les méthodes suivantes :

- Init : l'initialisation d'un agent,
- Live : la méthode principale,
- Action : les actions spéciaux,
- Create, Terminate : la création et la fin d'un agent.



**Figure 3.6 : Modèle structurel d'un Agent**

Il y a deux sous-classes dans la classe « Agent » (voir figure 3.7) : « Product Agent » et « Stationary Agent ». Cette dernière sous-classe comporte aussi deux sous-classes, nommées « PMA » et « Residential Agent ». La classe « Product Agent » implémente les propriétés nécessaires pour l'agent de production (la plupart concerne la mobilité de l'agent) et la classe « Stationary Agent » est utilisé pour la création des PMA et des « Residential Agent ».



**Figure 3.7 : Hiérarchie des classes d'agents dans PABADIS**

### **3.5 Cycle de production**

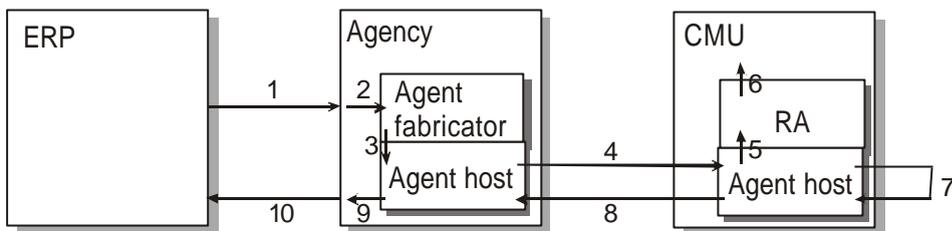
Un cycle de production dans PABADIS démarre de façon classique avec l'émission d'un ordre de fabrication émanant de l'ERP. Cet ordre de fabrication décrit la séquence de fabrication du produit, sans toutefois spécifier l'affectation des machines, ni l'ordonnancement exact des tâches. Après analyse de l'ordre de fabrication, l'agence crée un ou plusieurs Agents de Production (PA) ; chaque agent exécute alors son plan de fabrication individuel, en procédant de la façon suivante :

- recherche d'un service CMU adéquat
- négociation avec les agents résidentiels
- sélection du CMU
- migration vers l'hôte sélectionné

Tout au long du cycle de production, l'agent de production accompagne le produit en cours d'élaboration. Il élabore pour finir un rapport d'exécution, conservant ainsi une trace de son activité.

Le cycle de production d'un ordre de fabrication comporte les étapes suivantes : Au début, l'ERP produit un ordre de fabrication et l'envoie à l'Agency (1). Ensuite, il sera transformé par l'Agency en un ordre de travail contenant plusieurs tâches à effectuer selon un ordre séquentiel et sera transmis à l'Agent fabricant pour la création d'un agent de production (2). L'agent fabricant crée un agent de production dans l'Agent Host et il sera initialisé à partir de la description de l'ordre de travail (3). Après la création, l'agent de production cherche un service CMU adéquat en demandant les adresses de ces derniers au LUS. Le choix du service est effectué à travers la négociation avec les agents résidentiels. Une fois fait, l'agent de production migre à travers le réseau au CMU choisi et réside dans la partie Agent host du CMU dans le but de réaliser les tâches qui lui étaient attribuées (4). L'agent de production accède à l'agent résidentiel pour le traitement de la tâche (5). Cet accès sera transmis au niveau fonctionnel par l'agent résidentiel représentant la ressource choisie (6). Quand le traitement de la tâche est achevée, l'agent de production migre à la prochaine CMU (7) pour effectuer sa prochaine tâche en

passant par les étapes précédentes (5, 6). Lorsqu'il a fini toutes les tâches spécifiées dans son ordre de travail, l'agent de production retourne à l'Agent hot dans l'Agency (8) pour remettre les informations récoltées, pour signaler la fin de traitement de son ordre de travail (9) et pour s'arrêter. Enfin l'Agency transmet les informations remises par l'agent de production à l'ERP (10). Ce cycle de production est détaillé dans la figure 3.8.



**Figure 3.8 : Cycle de production d'un ordre de fabrication dans PABADIS**

### **3.6 Allocation des ressources**

Le pilotage au niveau fonctionnel dans PABADIS est effectué à travers le principe de l'allocation des ressources. En effet, ce principe introduit réactivité et adaptabilité face aux perturbations pouvant survenir dans les ressources (pannes, nouvelles configuration, etc.) dans le système. L'allocation des ressources dans PABADIS est effectuée entre les agents de production ou «Agent-Produit» et les agents résidentiels ou les «Agent-Ressource». Un «Agent-Produit» contient toutes les informations nécessaires concernant les tâches et les opérations à faire pour réaliser un produit, tandis qu'un «Agent-Ressource» possède les informations nécessaires sur la disponibilité des ressources dans un seul CMU [PABADIS, 2002b].

L'allocation des ressources dans le système est effectuée à travers un protocole de négociation entre ces deux types d'agents à base de Contact-Net Protocol. Le protocole de négociation proposé utilise une approche de réservation orientée PR, (c.a.d l'allocation des ressources est effectuée par l'«Agent-Produit» pour ses prochaines tâches à effectuer). Le choix des ressources fait par l'agent se base sur la règle de dispatching du temps de traitement le plus court ou Shortest Processing Time (SPT).

Le protocole de négociation proposé entre les agents suit les étapes suivantes :

- l'«Agent-Produit» envoie une requête aux «Agent-Ressource» fournissant le service voulu, pour effectuer une tâche à un moment donné,
- l'«Agent-Ressource», après avoir vérifié son état envoie une offre concernant sa disponibilité et la durée de traitement de la tâche demandée par l'«Agent-Produit»,

- le choix d'un «Agent-Ressource» pour effectuer une tâche donnée par un «Agent-Produit» est en fonction du temps de traitement le plus court proposé par les «Agent-Ressource»,
- si la requête envoyée par un «Agent-Produit» est prioritaire par rapport aux allocations existantes sur le même «Agent-Ressource», ce dernier essaiera de re-allouer sa file d'attente. L'«Agent-Ressource» demande alors aux «Agent-Produit» qui ont déjà réservé à l'avance un temps d'exécution de le retarder ou de l'effacer de sa file d'attente,
- l'«Agent-Produit» vérifie son propre ordonnancement et fait son choix. S'il estime impossible de retarder l'exécution de sa tâche, il envoie des requêtes aux autres «Agent-Ressource» pour trouver une meilleure solution et il retire son ancienne allocation. Dans le cas contraire, il accepte le retardement de l'exécution.

La partie d'allocation des ressources dans PABADIS est toujours en cours de développement. Le protocole de négociation proposé est considéré comme un protocole simple utilisant une approche orientée PR comportant de nombreux inconvénients dans un environnement dynamique à forte perturbation, d'autres propositions seront étudiées dans l'avenir pour obtenir un protocole de négociation plus élaboré.

### **3.7 Apports et perspectives**

PABADIS peut être considéré comme un projet révolutionnaire pour la production manufacturière automatisée. L'objectif du projet est d'apporter une preuve de la faisabilité de ses concepts. Ceci conduit à développer une plate-forme générique qui peut être appliquée et adaptée selon différents contextes industriels. De plus, cette plate-forme générique dispose de nouveaux outils au niveau industriel compatibles en communication à travers un réseau ouvert (Plug-and Play, JAVA, Internet).

Actuellement ce projet est encore un prototype. Il s'achèvera en octobre 2003 avec la mise en place d'un démonstrateur situé à Magdeburg en Allemagne. Celui-ci est destiné à mettre en évidence les concepts qui ont été développés, sur la base d'un système de production pilote, et comportant des automates connectés en réseau (automates programmables JETTER et PHOENIX), une plateforme multi-agents (GRASSHOPPER), une fonction PnP (JINI), un ERP (ATLANTIS) et un SCADA (InTouch). Des prolongements sont en cours d'élaboration notamment avec une implémentation du code avec la plateforme multi-agents LANA, et la mise en place de démonstrateurs à Zurich et à Nîmes.

## 4 Conclusion

L'approche décentralisée de gestion inverse de production sera utilisée dans cette thèse à travers la reconfiguration automatique des ressources. Elle remplacera l'approche conventionnelle basée sur le principe de l'ordonnancement. Le projet européen est donné comme exemple d'architecture de pilotage décentralisée. L'approche PABADIS offre plus de flexibilité, modularité et réactivité au niveau fonctionnel (interactions entre ses composantes) et physique interne (reconfiguration automatique des composantes à travers la notion Plug-and Participate). La mise en œuvre de ces approches par des agents a rendu la plate-forme développée modulable, générique et ouverte pour l'évolution permanente du principe de pilotage dans l'industrie.



# **Chapitre 4 : Nouveaux Concepts et Mécanismes pour la Gestion de Production**

## **1 Introduction**

Le pilotage fonctionnel d'un système de production décentralisé dans un environnement dynamique situé nécessite le développement de nouveaux concepts et mécanismes. Dans cette thèse, le pilotage est réalisé à travers l'allocation dynamique des ressources à base d'interactions auto-organisées entre les différentes entités du système. L'allocation est réalisée par l'intermédiaire des protocoles de négociations entre agents représentant les produits et les ressources du système.

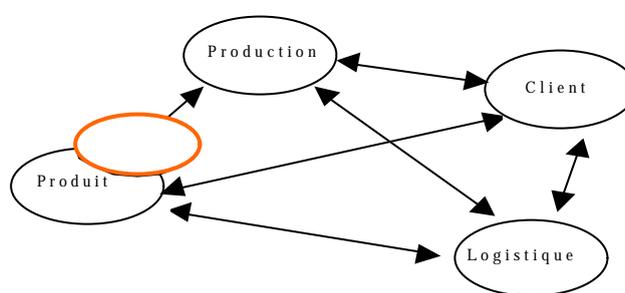
Dans ce chapitre, une analyse et une critique des modèles existants traitant le problème de l'allocation dynamique de ressource à base d'agents est présentée. Ensuite, de nouveaux concepts seront détaillés, comme la session de demande pour l'organisation de la décision, en temps réel, entre agents en compétition, la notion de « coopération » et de « compération », une méthodologie de négociation entre agents pour un système de production décentralisé. De plus, quatre nouveaux protocoles de négociation seront développés à base de coopération, coopération et de compétition. Dans la suite, il existe un nombre important de sigles. Pour faciliter la lecture, le lecteur se reportera au glossaire situé en fin de thèse.

## **2 Les agents pour la modélisation des interactions entre les entités d'un système de production**

L'approche multi-agents peut être appliquée à de nombreux domaines des systèmes complexes de production : pilotage, ordonnancement, gestion de la production, conception, ingénierie concurrente, système d'information, logistique, etc. Les systèmes multi-agents ainsi conçus ont pour but de résoudre des problèmes et/ou coordonner des actions. Dans le but de prendre en compte les différentes contraintes liées aux systèmes de production modernes, dans le cadre du 'E-business' par exemple, quatre types d'entités (ou agents) doivent être considérés dans les processus de prise de décision (voir figure 4.1) :

- Agent de Production : il représente un processus de fabrication, un partenaire ou un fournisseur, une file d'attente, etc. Il comprend une ou plusieurs ressources matérielles ou humaines. Il est associé à un ordre de « production » ou un ordre d' « approvisionnement ».
- Agent Client : il modélise également une requête et traduit une demande client. Ce dernier étant maintenant partie intégrante des systèmes de production, on se doit de prendre en compte son rôle. En effet il intervient directement au niveau du système de décision et/ou du choix des approches stratégiques dans l'entreprise considérée. Il est généralement associé à un ordre 'Client'.

- Agent Logistique : L'environnement nouveau requis par la globalisation des marchés, les délocalisations et les systèmes de production distribués fait que la notion de logistique, (avec ses stratégies et ses tactiques) ne peut être considérée comme une activité marginale dans un système de production complexe. Il s'agit donc d'une entité à part qui permettra de modéliser les contrats/approches client-fournisseur, les démarches de transport, les tailles de Kanban, etc. On y associera un ordre 'logistique'.
- Agent 'Produit' : celui ci est destiné à représenter un produit ou un service. C'est un agent généralement mobile associé à une demande ou commande. Au cours du processus de transformation, un produit évolue et peut choisir en fonction de ses caractéristiques/ priorités/ spécifications, etc. les ressources et parfois le type de tâche qui lui seront prescrites ou attribuées. On lui associe un ordre 'produit'.



**Figure 4.1 : Interactions entre les différentes entités d'un système de production**

Dans cette thèse, nous nous intéressons seulement aux interactions entre les entités produits et productions. En fait, les autres interactions ont été étudiées au LGI2P-EMA-Nîmes à travers le développement d'un simulateur (DAPS), la participation à un projet européen (PABADIS) et des travaux de thèse (Voir tableau 4.1) [Massotte et al., 2001a ; Diep et al., 2001 ; Liu, 2002 ; Reaidy et al., 2003a, 2003b].

Interactions entre les composantes d'un système de production				
Evolutions des concepts	Produit	Production	Client	Logistique
DAPS			**	**
VFDCS		*	*	
PABADIS	*	*	*	
Thèse-Reaidy	**	**		

\* : interactions simples ; \*\* : interactions sophistiquées.

**Tableau 4.1 : Evolution des concepts, dans notre équipe de recherche, à travers les interactions entre les différentes composantes d'un système de production**

DAPS «Dynamic Adaptation of Complex Production Systems» est un simulateur conçu pour implémenter de nouveaux concepts concernant la gestion de chaîne d'approvisionnement « Supply Chain Management » (SCM) et la technologie de flux de demande « Demand Flow Technology » (DFT). Il est basé sur la gestion inverse de production dans le domaine du pilotage et de la simulation. Il utilise la technologie agent pour modéliser les interactions entre les entités clients et logistiques dans le système de production. Le principe juste à temps « Just In Time (JIT) » est appliqué dans ce simulateur pour la gestion de production. Finalement, DAPS permet une analyse ascendante « Upward » des effets de la demande dynamique du client dans le système. De plus, DAPS détecte les anomalies logistiques et il fournit les données de diagnostic pour améliorer la logistique du système. [Massotte et al., 2001a].

VFDCS « Virtual Factory Dynamic Configuration System » est une plateforme de développement, innovante, permettant de modéliser et de traiter et d'élaborer une approche méthodologique de la réorganisation / reconfiguration dynamique de systèmes de production distribués. Elle intègre plusieurs techniques modernes relatives au 'Supply Chain Management', le 'Pull Production Control', le E-commerce basé sur Internet, les enchères basées sur des règles de raisonnement, et enfin une coordination basée sur le 'contact net protocol'. L'organisation des tâches se fait au travers d'enchères et de négociations, sous couvert d'un objectif global, ce qui rend caduc le rôle d'un superviseur de pilotage [Liu, 2002 ; Liu et Massotte, 1999].

PABADIS « Plant Automation BAsed on DIstributed Systems » est un projet européen qui a pour objectifs la reconfiguration automatique et dynamique des chaînes de production, basée sur la notion de "plug-and-participate", et la suppression totale ou partielle des tâches de planification et d'ordonnancement, par une organisation décentralisée de la production. Les concepts qui sont à l'origine du projet PABADIS s'appuient sur des technologies distribuées et sur un modèle multi-agents. Grasshopper est la plate-forme de développement utilisée dans ce projet qui est une plate-forme à base d'Agents Mobiles [PABADIS, 1999 ; Diep et al., 2001].

Thèse-REIDY : l'objet de cette thèse est de concevoir et développer des approches nouvelles pour la gestion des systèmes de production complexes, incluant des notions telles que la grande variabilité des produits, les aspects dynamiques de la demande et la reconfiguration de produits ou de procédés. La démarche utilisée est non centralisée et hétérarchique ; elle intègre plusieurs concepts et mécanismes tels que l'auto-organisation, les protocoles de négociation à base de coopération (coopération et compétition), la théorie de jeux et les enchères. En terme de mise en oeuvre, les agents représentent les entités physiques (produits et ressources) du système. Le pilotage du système de production est effectué à travers l'allocation dynamique des ressources entre les agents « produits et ressources » [Ready, 2003a].

Dans les paragraphes qui suivent nous détaillerons les différents types d'ordonnancement à base d'agents pour le pilotage décentralisé des systèmes de production.

### **3 Agents pour l'allocation des ressources dans les Systèmes de Production décentralisés**

Dans un système de production décentralisé, les agents peuvent être utilisés comme modules d'encapsulation selon deux approches : l'approche de décomposition fonctionnelle et l'approche de décomposition physique [Shen et Norrie, 1999]. Dans la première approche les agents sont assignés ou affectés à des fonctions dans le système (la planification, l'ordonnancement, le cheminement des produits, etc.). Tandis que dans la seconde approche les agents sont utilisés pour représenter des entités physiques (les travailleurs, les ressources, les produits, les opérations, etc.). Pour cela, l'ordonnancement du système peut être réalisé selon deux types.

Dans le premier cas, les agents sont responsables de l'ordonnancement des ordres de fabrication. Cette approche est liée aux savoir-faire en ordonnancement (méthode de placement, technique de recherche incrémentale, de backtracking, etc.). l'ordonnancement global du système est obtenu à travers l'émergence des ordonnancements locaux. [Sycara et al., 1991 ; Mceleney et al., 1998]. Le second cas est lié au savoir-faire ayant trait à l'ordonnancement local des ressources du système opérant (machines, cellules, etc.). Un agent représente une ressource ; il est responsable de son propre ordonnancement. Pour ce faire, il utilise des mécanismes de négociation avec les autres agents dans le système. [Baker, 1991 ; Parunak et al., 2001 ; Shen and Norrie, 1998].

Finalement, nous pouvons mentionner un autre type d'ordonnancement local, ou plutôt d'allocation dynamique des ressources locales, pour le pilotage décentralisé et auto-organisé d'un système de production. Dans ce type l'ordonnancement est effectué en temps réel, totalement distribué et organisé au niveau du produit à fabriquer. En effet, l'allocation dynamique des ressources est effectuée par un agent représentant le produit physique dans le système. Il est responsable de l'accomplissement de l'ensemble de ses tâches sur des ressources appropriées durant son cycle de vie [Krothapalli and Deshmukh, 1999 ; PABADIS, 1999 ; Diep et al., 2001 ; Reaidy et al., 2002a ; 2002b].

L'allocation dynamique des ressources nous paraît l'approche la mieux adaptée pour les environnements de production dynamiques situés et fortement perturbés. Ce dernier type est détaillé dans les prochains paragraphes à travers des protocoles de négociation et selon des horizons et catégories de décision pour les agents représentant les produits dans le système.

### **3.1 Allocation des ressources à travers des protocoles de négociations**

Le réseau contractuel ou « Contract Net Protocol » est le protocole de négociation le plus utilisé pour les mécanismes d'allocation entre agents dans un SP [Shen et Norrie, 1999]. Ce protocole peut être appliqué par les entités indépendantes produits et ressources de différentes manières :

- les ressources étant des clients ou des fournisseurs se partagent le travail entre eux à travers des appels d'offres et des propositions subséquentes,
- les produits étant des clients demandent des tâches et les ressources deviennent des fournisseurs potentiels à ces tâches.

L'avantage principal de ce protocole se résume à sa réactivité dans un environnement dynamique et à sa facilité à être implémenté dans n'importe quel système décentralisé. Tandis que ses points faibles sont liés aux décisions locales effectuées par les agents qui ne prennent pas en considération les performances globales du système.

Les mécanismes de négociation pour la génération et le maintien de l'allocation des ressources dans ce protocole sont développés comme des processus d'enchères entre les «Agent-Produit» et les «Agent-Ressource». Ces processus peuvent être appliquées selon les perspectives suivantes :

- « Orienté-Produit » (O-P) : l'«Agent-Produit» est considéré comme l'arbitre final de la décision. Dans ce processus, les «Agent-Ressource» sont en compétition entre eux et font des propositions d'enchères pour l'exécution d'une tâche demandée par un «Agent-Produit». Finalement, l'«Agent-Produit» choisit l'«Agent-Ressource» qui propose la meilleure offre [Lin et Solberg, 1992],
- « Orienté-Ressource » (O-R) : l'«Agent-Ressource» est considéré étant l'arbitre final de la décision. Les «Agent-Produit» sont en compétition pour une même tâche fournie par un «Agent-Ressource». Ce dernier choisit l'«Agent-Produit» selon ses propres critères liés à la priorité de la tâche, date d'échéance, etc. [Shen et Norrie, 1998],
- « Bi-Directionnelle » (B-D) : la décision finale est partagée entre les deux types d'agents selon leurs propres critères de choix [Pesenti et al., 2001].

Le choix d'un processus selon ces différentes perspectives dépend du type (produit/ressource) et du nombre (un/plusieurs) des entités en compétition dans le système. La perspective «Orienté-Produit» est préférable dans une compétition entre les ressources et propose la meilleure offre à un produit « un produit et plusieurs ressource ». A l'inverse, la perspective «Orienté-Ressource» peut être appliquée dans une compétition entre les produits pour obtenir le service proposé par une ressource (une ressource et plusieurs produits). Finalement, pour des compétitions entre plusieurs produits qui cherchent les meilleures offres proposées par plusieurs

ressources (plusieurs produits et plusieurs ressources), la perspective «Bi-Directionnelle» peut être utilisée. Dans ce dernier cas, la distribution de la décision peut être compliquée à réaliser et impose un traitement lourd pour gérer toutes les interactions entre les différents types d'agents. L'utilisation d'un médiateur est considérée comme une solution pour la gestion de ces types d'interaction. Dans cette thèse nous utiliserons la perspective «Orienté-Produit» avec l'utilisation d'un médiateur pour la gestion du cas cité précédemment.

### **3.2 Horizon de la décision de l' «Agent-Produit»**

Dans cette partie nous détaillons les approches utilisées pour le processus «Orienté-Produit». Pour définir le plan d'horizon que la décision d'«Agent-Produit» couvre pour l'allocation de ses différentes tâches, nous distinguons deux approches à base d'allocation ou de réservation [Saad et al. , 1997] :

- la première est nommée Production Reservation (PR), elle consiste à allouer ou à réserver toutes les tâches nécessaires à effectuer pour un produit en même temps dès son arrivée dans le système,
- tandis que, dans la seconde approche, qui est nommée Single-Step Production Reservation (SSPR), l'«Agent-Produit» procède par l'allocation seulement de sa prochaine tâche. L'allocation est faite durant l'exécution de sa tâche actuelle. L'«Agent-Produit» répète ce processus durant sa fabrication pour toutes les tâches nécessaires à effectuer.

Les principaux avantages de ces approches sont respectivement :

- de rendre la prédiction possible, surtout pour les produits à haute priorité,
- de favoriser également une meilleure réactivité et adaptabilité dans un environnement dynamique à forte perturbation.

Différents travaux de recherches ont utilisé ces deux approches (voir tableau 4.2). L'inconvénient de l'approche PR est son manque d'adaptation dans les environnements dynamiques, en raison du taux d'arrivée des produits dans le système ou des pannes des ressources. En effet, le plan d'allocation établi par le produit dès son introduction, dans le système, et pour toutes ses tâches à effectuer sera, remis en cause à chaque perturbation dans le système. L'inconvénient de l'approche SSPR est l'impossibilité pour les produits de faire de bonnes évaluations de leur temps de traitement à l'avance (en début de production), parce que ceux-ci dépendent de la dynamique du système et de la cadence d'arrivée des autres produits. Dans ce système, les mécanismes d'émergence conduisent toujours à une satisfaction dynamique de la configuration produit/processus, mais qu'on ne peut pas définir à l'avance [Massotte et al., 2002]. En effet, quelle que soit la configuration initiale, on converge toujours vers un attracteur. Toutefois le choix du bassin d'attraction reste difficile à définir à priori. Finalement, Dans un environnement dynamique et situé,

l'approche SSPR est préférée à l'approche PR parce qu'elle est plus réactive et adaptative.

Travaux	Horizon de décision	Mécanismes d'enchères	Caractéristiques principales
Lin et Solberg 1992 (IFCF)	SSPR	O-P	« Market-like control model », combinaison des mécanismes de coût et d'objectif.
Saad et al. 1996	PR	B-D	Règle de dispatching approche PR et SSPR.
Ben-Arieh et Chopra 1997	PR	O-P	Utilisation de la théorie des jeux, les tâches sont considérées comme des joueurs et les machines comme des stratégies à choisir.
Sousa & Ramous 1997	PR	O-R	Utilisation de l'approche holonique (HMS), ordonnancement dynamique à base de forward / backward.
Parunak et al 1998 (AARIA)	PR	O-R	Ordonnancement à base de "Forward/Backward", broker Agent.
Shen et Norrie 1998 (Metamorph II)	PR	O-R	Agents ressources se coordonnent dynamiquement par l'intermédiaire d'un agent médiateur.
Krothapalli et Deshmukh 1999	SSPR	O-P	Fonction d'Echange ou « Currency function » basée sur une relation non linéaire entre la date d'échéance du produit et le taux d'utilisation des machines.
Usher et wang 2000	SSPR	B-D	Les agents ressources utilisent des informations-additionnelles récupérées, par le protocole de négociation, avant de proposer leurs offres.
Cicirello 2001	SSPR	O-P	Utilisation de la théorie des jeux, Dynamic shop floor routing.
Pesenti et al. 2001	SSPR	B-D	Négociation à base des règles de dispatching.

PR: Production Reservation; SSPR: Single-Step Production Reservation;  
O-R : Orienté-Ressource; O-P : Orienté-Produit; B-D: Bi-Directionnelle.

**Tableau 4.2 : Comparaison entre des travaux de recherche existant utilisant les protocoles de négociation pour le pilotage des systèmes de production.**

### 3.3 Catégorie de la décision de l'«Agent-Produit»

L'allocation des ressources utilisée par un «Agent-Produit» selon l'approche SSPR peut être classée en trois catégories [Ready et al., 2002a, 2002b] :

1. Egoïste : l'agent fait son choix sans prendre en considération les demandes des produits en compétition avec lui dans le système. Par exemple, l'agent choisi toujours la meilleure offre proposée par les ressources,

2. Altruiste : l'agent effectue son choix en considérant les demandes des autres produits. Par exemple, l'agent choisit l'offre qui correspond le mieux à son degré de priorité par rapport aux autres agents en compétition avec lui,
3. Consensuelle : le choix de l'agent prend en considération, simultanément, ses propres critères et contraintes (date d'échéance, coût, etc.) et les demandes effectuées par les autres agents. Dans cette catégorie, les agents négocient entre eux pour trouver le meilleur compromis pour toutes les situations. Pour cela, l'agent essaye de se procurer des informations en temps réel des états et des contraintes de ses concurrents ou d'utiliser des méthodes à base d'heuristique, de statistique ou de la théorie des jeux pour trouver la meilleure décision à prendre dans n'importe quelle situation.

Ces différentes catégories ont leurs avantages et leurs inconvénients. La première catégorie est simple à mettre en œuvre, mais ne s'adapte pas à la dynamique du système. En effet, elle garde toujours le même comportement et ne réagit pas pour n'importe quelle perturbation effectuée dans le système. La deuxième catégorie ne peut être appliquée que dans les environnements où la notion de priorité est privilégiée. Bien que la dernière catégorie soit moins simple à implémenter, elle a l'avantage de l'adaptabilité dans n'importe quelle situation dans un environnement dynamique parce qu'elle utilise un compromis d'intérêts entre les deux autres catégories.

Finalement, pour le pilotage décentralisé à base d'agent dans un environnement dynamique situé, nous avons choisi l'allocation dynamique des ressources par l'intermédiaire des protocoles de négociation. Les agents représentent les produits et les ressources du système. Le processus de négociation est orienté produit. L'horizon de la décision de l'agent produit est local, selon l'approche SSPR, et effectué d'une façon consensuelle.

## **4 Concepts spécifiques**

Dans cette partie, nous développons des nouveaux concepts pour la décision entre agents et de nouveaux protocoles de négociations à base de certains mécanismes cités ci-dessus. Ces concepts portent, de plus, sur l'allocation entre produits et ressource selon l'approche «Orienté-Produit». Nous privilégions cette approche en raison des liens étroits entre la dynamique du système et les changements ou perturbations effectuées par l'arrivée imprévue et aléatoire des demandes de fabrication des produits dans le système. De plus, les agents produits sont les responsables directs de l'allocation de toutes tâches de chaque produit dans le système. Nous signalons aussi que l'allocation dynamique des ressources est effectuée à travers l'approche SSPR parce qu'elle est plus réactive et adaptative que l'approche PR dans un environnement dynamique situé. Avant de passer à ces nouvelles réalisations, l'analyse critique des différents travaux existants permettre d'introduire des changements et des adaptations pour l'amélioration du fonctionnement du système.

#### **4.1 Commentaires liés au cadre de l'étude**

Dans l'approche SSPR, le moment de la prise de décision des agents est un critère essentiel à prendre en considération pour limiter l'impact sur les résultats et les performances du système. Cet instant peut requérir un temps d'attente bénéfique. Ce temps permet à l'agent d'avoir accès à toutes les informations utiles et nécessaires pour réagir efficacement et en temps réel face à la dynamique du système.

Ce point est évoqué dès maintenant, dans cette partie, pour mettre en évidence un nouveau concept développé dans le cadre de la thèse répondant aux besoins détaillés antérieurement. Ce concept sera appliqué sur plusieurs protocoles de négociations et précisé dans les prochains paragraphes.

Ayant besoin d'accomplir plusieurs tâches lors de son processus de fabrication, un produit entre en compétition avec d'autres produits pour l'allocation de la prochaine même tâche. Dans ce contexte, nous distinguons trois modèles développés dans la littérature, pour définir le moment de la prise de décision, tels que :

- modèle1 : le produit prend sa décision lors de son entrée dans le système ou lorsqu'une ressource commence le traitement de sa tâche. La décision prise dans ce modèle est souvent de type « égoïste ». Dans ce cas, seul l'ordre de demande d'un service ou l'arrivée d'un produit dans le système est pris en compte. Par exemple, un produit choisit toujours la meilleure offre proposée par les ressources sans prendre en considération les autres produits dans le système. Ce modèle est appelé « First-Come-First-Serve » (FCFS) et il est utilisé souvent dans les règles de dispatching ou « Dispatching rules » parce qu'il est simple et réactif [Baker, 1998] .
- modèle 2 : le produit fait son choix après attente d'une durée de temps constant fixé à l'avance. Cette période de temps permet aux ressources de proposer une meilleure offre tout en prenant en compte des considérations statistiques comme le nombre de produits demandant le même service et leurs pourcentages de réussite par demande [Krothapalli et Deshmukh, 1998].
- modèle3 : la durée d'attente est limitée par le temps de fabrication de la tâche sur une ressource. Cette période de temps permet à chaque ressource, après avoir reçu les demandes des tâches pour des offres concernant le temps total de la fabrication, de proposer séparément une offre pour chaque produit. L'offre proposée prend en considération la priorité de chaque produit en compétition pour la même tâche dans le [Usher et Wang, 2000a ; Usher, 2001].

#### **4.2 Discussion**

L'utilisation du modèle basé sur le FCFS ne donne pas de bonnes performances dans un environnement dynamique. En effet, la priorité des produits, liée à leurs

coûts ou à leurs dates d'échéance «Due Date », y est prise en considération. Ceci est justifié par le fait que le choix du produit ne prend en considération que sa propre valeur et non les valeurs des autres produits dans le système.

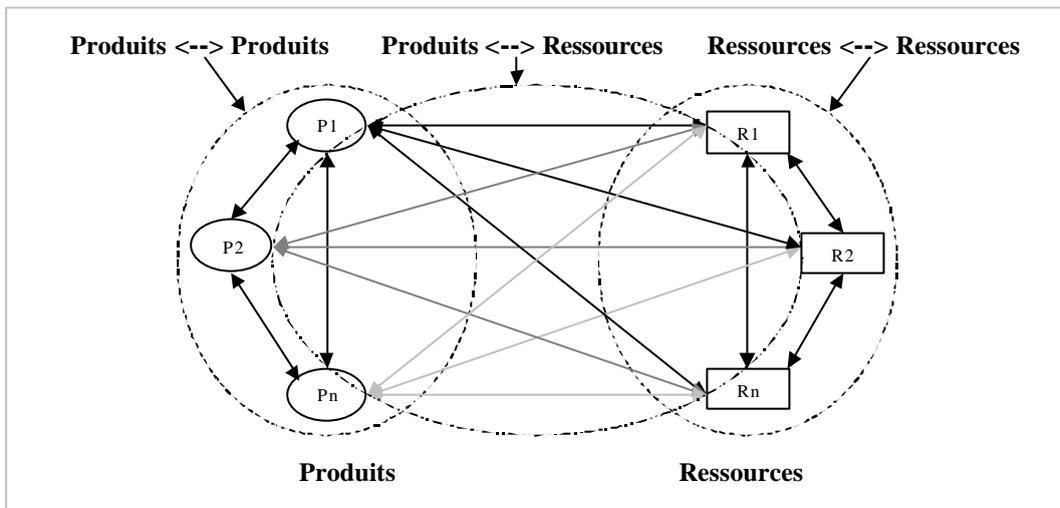
Le model proposé par N. Krothapalli est intéressant et peut être considéré comme une source d'inspiration pour la conception de protocoles de négociations et de la modélisation des interactions Produits-Ressources dans la méthode SSPR [Usher et Wang, 2000a ; Pesenti at al., 2001) ; Reaidy et al., 2003a]. N. Krothapalli a établi une relation non linéaire entre la monnaie et le temps pour la négociation entre produits et ressources dans le système considéré. Le produit se comporte comme un acheteur de services muni d'une quantité de monnaie correspondant à sa priorité (ou à sa due date). La machine est un vendeur de service, avec un prix proposé selon des considérations statistiques et heuristiques. L'absence d'établissement de relation entre les produits eux-mêmes, pour les évaluer les uns par rapport aux autres, reste le point faible de ce modèle. Cette faiblesse se manifeste par un manque de stabilité dans un environnement dynamique à forte variation de priorité entre les produits.

Dans le troisième modèle cité, la possibilité pour chaque ressource d'ordonner les produits par leur ordre de priorité, et de proposer séparément une offre pour chaque produit selon sa priorité, tout en bloquant les produits moins prioritaires, tue toute forme de réactivité et adaptabilité entre les produits dans un environnement dynamique. De plus, la réservation des meilleures offres pour des produits ayant des priorités élevées cause un problème majeur, pour répartir les tâches à effectuer sur les machines d'une part, et pour respecter l'ordre d'entrée des produits dans les files d'attente des machines selon l'ordre d'allocation, d'autre part.

Ces différents modèles ne prennent pas en considération la relation « Produits—Produits » en envisagent seulement la relation « Produits—Ressources ». Or, la multiplication des relations entre les entités complexifie le problème, car il y a davantage d'interactions à gérer entre les différentes entités (voir figure 4.2). Le temps de calcul et de réponse entre les entités augmente donc en conséquence. L'absence de relations essentielles entre les entités, qui dépendent étroitement des caractéristiques des problèmes posés, amène à utiliser des approches heuristiques ou statistiques compliquées, pour combler le manque d'information révélé par des relations omises. Ceci peut pénaliser les résultats des stratégies ou algorithmes utilisés. La relation « Ressources—Ressources » n'est pas prise en compte dans le cadre de la thèse, parce que les deux relations « Produits—Produits » et « Produits—Ressources » suffisent à modéliser et proposer une bonne solution en temps réel au problème posé (voir figure 4.3).

Les interactions dynamiques, dans le système, entre produits et ressources proviennent, en grande partie, de la variation des paramètres liée aux produits (« workload », quantité, « due date », coût, etc.) et aux ressources (quantité, capacité de production, panne, etc.). Un produit peut, à tout moment, subir des changements au niveau de la demande, de la quantité introduite dans le système, des

options, du temps d'arrivée dans le système et de sa priorité des produits liée à son coût, de sa due date, etc. Pour cela, le choix la relation « Produits—Produits » permet de faire face à la dynamique du système. Ce choix est appliqué dans la thèse et va être utilisé dans la conception du principe de la « Session de Demande » et de la méthodologie et protocole de négociation. Dans cette thèse, seule la variation de la demande au niveau des produits est prise en compte. De cette façon, le but de l'allocation des ressources devient de servir seulement la demande des produits. Chaque fois qu'une ressource devient inaccessible, les produits en attente dans la file d'attente de la ressource subiront une modification de leurs priorités. Donc gérer les pannes des ressources, équivaut à gérer les compétitions entre des produits ayant différentes priorités.



**Figure 4.2 : Interactions entre produits et ressources**

Dans notre recherche, nous nous sommes inspirés de ces différents modèles et nous avons développé un nouveau concept nommé « Session de Demande » pour l'aide à la décision. Celui-ci est « Orienté-Produit » tout en prenant en considération l'état en temps réels des produits, l'état des ressources et l'état global du système. Nous avons de plus pris en compte la relation « Produits—Produits » dans laquelle les produits se comparent, s'évaluent et s'organisent entre eux, par l'intermédiaire d'un agent coordinateur, avant de prendre leurs décisions durant leurs processus de fabrication (voir figure 4.3).

Après une analyse et une critique des modèles existants traitant du problème de l'allocation dynamique de ressource dans un système de production décentralisé, nous détaillerons dans les prochains paragraphes quelques concepts et protocoles de négociation développés dans le cadre de la thèse. Nous commençons par le concept de Session de Demande qui est un nouveau concept créé et implémenté dans notre recherche.

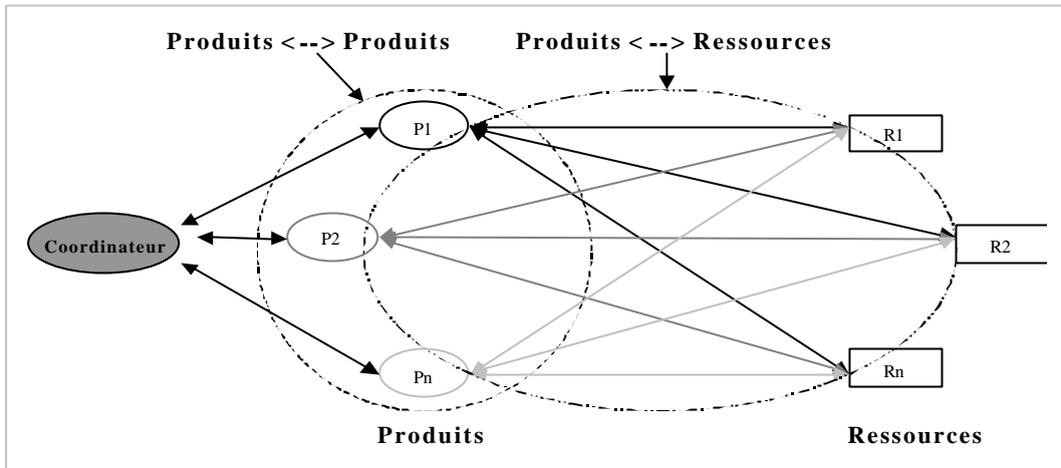


Figure 4.3 : Interactions entre produits et ressources par l'intermédiaire d'un agent coordinateur

## 5 Concept de «Session de Demande»

Le but de l'utilisation de la «Session de Demande» (SD) est de fournir en temps réel aux agents, en phase de négociation, des informations sur l'état des différentes entités associées dans le système lors de leur prise de décision. Ces informations peuvent concerner les caractéristiques (date d'échéance, priorité, coût, etc.) des autres agents produits, l'état des ressources dans le système (disponibilités, charges, etc.) ou l'état global du système (flux d'entrée des produits, flux de sorties des produits, etc.). Au cours de leur processus de fabrication, plusieurs «Agent-Produit» ou (AP) sont en compétition locale pour une même prochaine tâche, qui peut être fournie par plusieurs ressources ; ils participeront ensemble à une même « Session de Demande ».

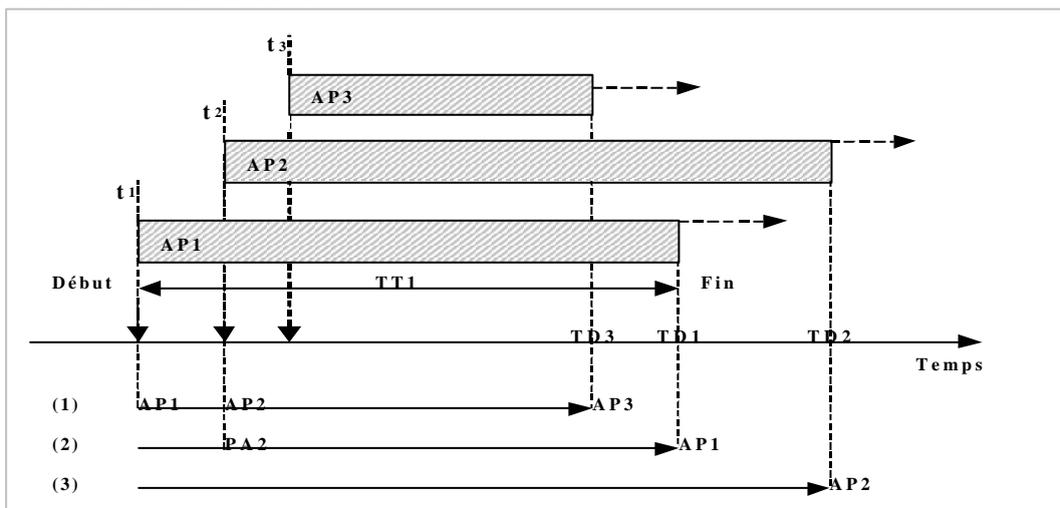


Figure 4.4 : Prise de décision des «Agent-Produit» dans une « Session de Demande »

La période de temps considérée dans la session pour chaque agent est limitée par le temps d'envoi de la demande dans la session. Elle est liée au temps du commencement (Début) de l'exécution de sa tâche sur une ressource, mais aussi au temps d'achèvement prévu de l'exécution de sa tâche sur la même ressource (Fin). Cette période de temps est appelée Temps de Traitement ( $TT$ ) de la tâche actuelle. L'instant de la prise de décision, ou le Temps de Décision ( $TD$ ) d'un agent, concernant le choix d'une ressource pour sa prochaine tâche est situé à l'achèvement de la tâche en cours de d'exécution sur lui-même. Ainsi, son choix prend en considération les autres agents dans la session, en compétition avec lui pour une même prochaine tâche, et qui ont commencé leur processus de fabrication avant, en même temps, et après lui.

Dans l'exemple décrit sur la figure 4.4, nous distinguons trois produits. Ils sont représentés par les agents  $AP_1$ ,  $AP_2$  et  $AP_3$ ; ils se sont respectivement inscrits dans la session de demande à l'instant  $t_1$ ,  $t_2$ ,  $t_3$  qui correspond au début de leurs processus de fabrication sur les ressources. Le point commun entre ces différents produits est le besoin d'exécuter la même prochaine tâche. Le Temps de Décision ( $TD_i$ ) de chacun de ces agents est lié à la durée de leurs Temps de Traitement ( $TT_i$ ) et se situe à la fin de chaque traitement. Par exemple, l'agent  $AP_3$  qui a fini le premier sa tâche prend sa décision en prenant en compte l'existence, les caractéristiques et les décisions potentielles des agents  $AP_1$  et  $AP_2$ . Grâce au principe de la «Session de Demande», le moment de la prise de décision des agents est situé juste avant le déplacement physique des produits aux ressources choisies, pour effectuer leur prochaine tâche. Cela diminue le risque des perturbations et des imprévus par rapport à la décision prise dans un environnement dynamique. De plus, l'échange des informations concernant l'état de chaque  $AP_i$  en temps réel permet une prise de décision coopérative et/ou compétitive pour satisfaire les intérêts locaux ou globaux des agents.

Dans ce qui suit nous définissons les stratégies de décisions utilisables par les agents pour satisfaire leurs propres intérêts et l'intérêt de l'environnement qu'ils représentent.

## 6 Coopétition ou Compération ?

La compétition et la coopération sont les deux principes de base des interactions entre les entités, dans un environnement où les intérêts locaux ou globaux sont pris en considération. En effet, les acteurs économiques sont amenés, à tout moment, à prendre des décisions qui sont aiguillonnées par leur intérêt individuel, tout en suivant néanmoins l'intérêt collectif. Le principe de la compétition consiste ainsi à employer des stratégies égoïstes de préservation de leurs intérêts locaux par les entités. A l'inverse, la coopération pousse les entités à mettre en commun leurs capacités et leurs besoins, afin d'aboutir à une satisfaction globale pour tous.

L'application de ces principes varie selon le contexte et les stratégies utilisées dans chaque environnement. Le marché d'affaire, ou le commerce, est l'exemple direct de la mise en situation de ces différents principes. Dans un marché constitué de fournisseurs et de clients, les fournisseurs peuvent être en compétition tout en utilisant des stratégies égoïstes pour attirer plus de clients et augmenter le plus possible leur gain. Dans le cas où le marché présente un certain décalage entre fournisseurs, au niveau de la capacité de production et des prix vente, la tendance chez les petits fournisseurs serait de coopérer pour obtenir une place significative de compétiteur face aux autres grands fournisseurs. Le phénomène des mutuelles et des syndicats est considéré comme une tendance coopérative entre les petits fournisseurs.

Actuellement, les exigences du marché ont évolué grâce à l'économie globale et à l'influence directe et explicite des exigences du client sur le marché. Pour survivre et être actives dans ce nouvel environnement, les stratégies doivent être adaptées à ces évolutions. Pour cela, de nouvelles propositions stratégiques ont été faites et consistent à combiner les deux principes de compétition et de coopération en même temps. De cette combinaison deux nouveaux principes : la « Coopétition » et la « Compération » ont émergé (voir tableau 4.3).

<b>Etape 1 \ Etape 2</b>	<b>Coopération</b>	<b>Compétition</b>
<b>Coopération</b>	Coopération	<b>Coopétition</b>
<b>Compétition</b>	<b>Compération</b>	Compétition

**Tableau 4.3 : Différentes stratégies d'interactions entre les entités du marché**

Ces deux principes sont des combinaisons de stratégies que nous pouvons ainsi définir :

1. La **coopétition** est la contraction de *coopération* et *compétition*. Selon ce principe, les fournisseurs ne prennent initialement pas le risque d'effectuer une compétition directe avec les autres compétiteurs sur le marché. Pour cela, ils commencent par coopérer entre eux, ne serait-ce que partiellement, pour la construction d'un bien commun global ; ensuite, ils redeviennent des compétiteurs lors de la décomposition et de l'attribution de ce bien. « L'activité de l'entreprise est affaire de coopération lorsqu'il s'agit de confectionner le gâteau et de compétition quand il vient le moment de le partager » [Nalebuff et Brandenburger, 1996]. Les accords accomplis entre les grandes entreprises comme Intel et Compac pour la fabrication des PC personnels, Motorola et AT&T au niveau télécommunication et au niveau de la recherche entre les universités et les industries fournissent autant d'exemples basés sur ce principe.

2. La **compération** est la contraction de *compétition* et *coopération*. L'inversion de l'ordre d'utilisation de ces deux stratégies nous conduit à un nouveau principe complètement différent de la coopération. En effet, les fournisseurs choisissent la compétition comme stratégie de départ et essayent d'abord de profiter le plus possible de leur capacité et d'assurer leur intérêt individuel. Ces mêmes fournisseurs modifient leur stratégies lorsque la concurrence de prix et de production devient difficile à tenir. Ils adoptent alors, en partie, le principe de la coopération pour peser sur le marché, fédérer des moyens de pression et de négociations tant au niveau de la distribution de la production ; enfin, elle permet le partage des gains entre eux. Le principe de l'Usine Virtuelle est un véritable exemple à l'application de ce principe. Dans ce dernier cas, par exemple, des PME/PMI et/ou sous-traitants tentent de récupérer une charge de travail maximale, compte tenu du contexte de leurs intérêts et de leurs capacités. Une fois effectuée l'allocation des tâches, les PME/PMI vont fédérer leurs moyens de production pour réaliser l'objectif global par entraide mutuelle, pour couvrir une perturbation physique inattendue, pour supporter une variation de la demande, etc.

L'application de ces deux principes au niveau fonctionnel des systèmes de production peut donc être effective entre produits et ressources pour l'allocation des ressources selon différentes stratégies de décision et scénarii. En effet, l'allocation des ressources durant la production peut être réalisée par les produits eux-même, par les ressources ou résulter d'un échange entre les deux en même temps. Dans un même contexte, plusieurs scénarios peuvent être élaborés. Nous nous limitons ici à deux scénarios possibles relativement significatifs pour l'application de ces deux principes dans un contexte industriel. Le premier est dit «Orienté-Ressource» ; il utilise la compération tandis que le second est de type «Orienté-Produit» et applique le principe de la coopération. Les deux scénarios sont utilisés pour satisfaire globalement et individuellement toutes les entités interagissant dans le système sont maintenant décrits :

- Scénario 1 : la **compération** peut être utilisée comme une stratégie de décision au niveau des ressources. En effet, celles-ci répondent aux demandes des produits pour la réalisation d'une tâche donnée en proposant, en fonction de leur disponibilité et du contexte de leurs meilleures offres. Une fois la compétition devenue ingérable, à cause des conflits permanents, des pannes et de la distribution asymétrique entre elles (au niveau des charges et de l'information), les ressources changent de stratégie et acceptent de se répartir les tâches afin de satisfaire aux exigences et aux performances globales du système,
- Scénario 2 : la **coopération** peut être appliquée au niveau des produits. Le produit, entité dynamique du système, possède des informations sur les tâches à exécuter, les priorités, les dates d'échéances, etc., il peut interagir directement et activement pour l'allocation des tâches au niveau des ressources. Durant leurs interactions, les produits coopèrent en mettant en commun les informations sur leurs propres intérêts, exigences et

contraintes, tout en organisant la décision entre eux. La coopération de cette manière rend le domaine de choix large et organisé rationnellement. Elle diminue le risque de décisions égoïstes et imprévues entre les produits. Au moment de la prise de la décision, chaque produit entre en compétition avec les autres pour le choix des ressources afin de préserver ses propres intérêts et satisfaire ses exigences.

On voit ici, que le choix du scénario est d'abord fonction de ce qu'on désire mettre en avant : qui est statique ou dynamique ? le produit ou la ressource ? qui gère l'autre ?

Dans la thèse, nous privilégions l'influence de la demande des clients dans le système, ce qui se traduit par l'arrivée dynamique des produits dans le système. Pour cela nous choisirons le second scénario, représenté par le principe de la coopération entre les produits dans le système.

## **7 Un exemple d'approche pour la modélisation du système : PABADIS**

Les travaux de la thèse se sont déroulés en parallèle et en complément de la conception et du développement des approches et des outils destinés au projet Européen PABADIS. Compte tenu de son évolution, PABADIS ayant gardé certains aspects de décision hiérarchiques pour la gestion de la production (ERP, SCADA, etc.), notre approche globale en a un peu divergé. Dans cette thèse, nous utilisons une structure organique de pilotage et de conduite totalement décentralisée et nous conservons le fonctionnement opérationnel développé dans PABADIS. En effet, le pilotage fonctionnel y est effectué à travers l'allocation dynamique des ressources, d'une façon auto-organisée, entre des agents représentant les produits et les ressources du système.

Dans cette partie, nous présentons une architecture d'un système multi-agent inspiré du projet PABADIS pour modéliser le pilotage et la conduite d'un système de production décentralisé. Elle comporte les entités suivantes : un « LookUp Service » (**LUS**), un générateur d'agents ou « Agency », des « Agent-Produit » (**AP**), et des « Agent-Ressource » (**AR**). Dans notre modèle, le « LookUp Service » est utilisé pour communiquer aux agents des informations concernant les adresses et les types des agents dans le système. L'Agency peut générer des Agents Produits dans le système. Les « Agent-Produit » et les « Agent-Ressource » y représentent les produits et les ressources physiques. Les modèles structurels généraux des agents aux niveau code et donnée ont été détaillés dans la partie PABADIS du chapitre 3.

Dans ce chapitre, nous nous intéressons plus spécialement à la méthodologie et aux protocoles de négociations entre les agents dans le but d'améliorer la gestion de la production dans un système décentralisé dynamique et situé. Dans ce qui suit, nous détaillons les aspects d'interaction, de négociation et de la prise de décision des agents produits et des agents de la « session de demande ». Ces deux derniers jouent

un rôle essentiel dans la conception et le fonctionnement des protocoles de négociation conçus ici.

### 7.1 Architecture de l'«Agent-Produit»

La notion d'«agent» utilisée dans ce mémoire correspond à une entité autonome, active et douée d'«intelligence» (au sens de la capacité d'initiatives, moyen de communication et de décision) représentant les produits et les ressources dans un système de production. L'agent est conçu de façon à disposer d'une capacité à négocier (coopération, compétition) avec les autres agents et de décider selon des protocoles et stratégies de négociation (théorie des jeux) utilisés afin de respecter les contraintes imposées dans le système et de satisfaire ses propres buts (temps d'échéance).

Les agents considérés sont de nature hybride (cognitive et réactive). L'agent le plus développé, dit « Intelligent », est l'«Agent-Produit» : Il est le responsable direct du pilotage fonctionnel auto-organisé du système. La prise de la décision et les mécanismes de négociation avec les autres agents pour l'allocation dynamique des ressources sont considérés comme une partie fondamentale de sa structure. Le modèle structurel général de l'agent garde les mêmes caractéristiques (code et données) que celui développé dans le projet PABADIS ; toutefois il comporte quelques changements au niveau des protocoles de négociation utilisés et des mécanismes de prise de la décision. La nature hybride de l'«Agent-Produit» peut être vue au niveau de ses interactions avec son environnement et de ses choix de stratégies. Différents types de stratégies sont utilisés par l'agent comme la coopération, la compétition et la coopétition. Dans ce dernier type, l'agent peut basculer d'une stratégie de coopération à une stratégie de compétition durant son exécution pour satisfaire ses intérêts et réaliser ses objectifs.

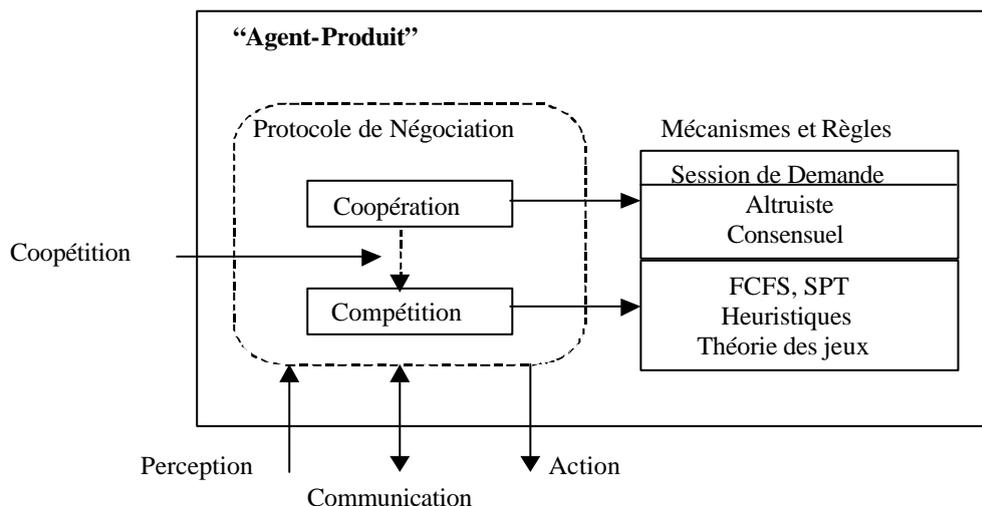


Figure 4.5 : Catégories de négociations et de décisions utilisée par l'«Agent-Produit»

La figure 4.5 montre les différentes catégories de négociation et de décision qui peuvent être employées par les «Agent-Produit» dans le système. Nous distinguons trois types de protocoles de négociation pour l'agent selon les approches suivantes : la coopération, la compétition et la coopération. Dans la première catégorie, les décisions des agents prennent en considération les priorités des autres agents, en participant à des « Session de Demande ». Leurs décisions peuvent être altruistes ou consensuelles. Tandis que dans la seconde, les décisions sont égoïstes ; elles peuvent être basées sur des règles de dispatching telles que « First-Come-First-Serve » (FCFS) ou le temps de traitement le plus court « Shortest Processing Time » (SPT), ou sur des méthodes heuristiques, ou même sur la théorie des jeux. Dans la dernière catégorie, l'agent réagit selon le principe de la coopération en commençant par coopérer, en anticipant une « Session de Demande », pour préparer l'allocation de sa prochaine tâche, puis en complétant le processus avec les règles et les mécanismes propres à la compétition durant sa prise de décision.

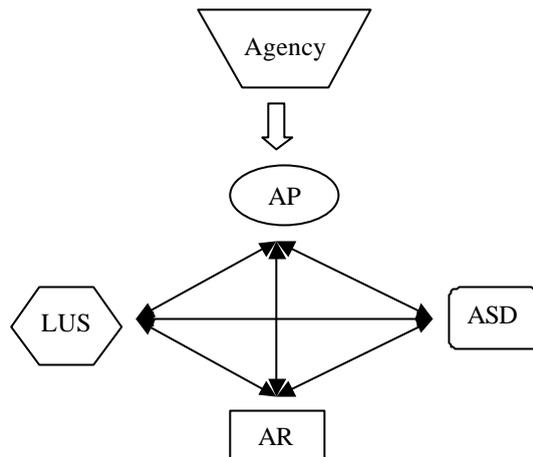
Nous avons développé un nouvel agent pour la gestion de la session de demande, nommé Agent de la «Session de Demande » (**ASD**) distinct des différents agents précités. Dans ce qui suit, nous détaillons les différentes caractéristiques de cet agent.

## **7.2 Agent de « Session de Demande » (ASD) ou agent coordinateur**

Le processus de la «Session de Demande » est géré par l'Agent de «Session de Demande » (ASD). Il est généré automatiquement dans le système chaque fois qu'un nouveau service ou tâche est signalé par le Look-Up Service. Le nombre de ce type d'agent dépend du nombre de tâches fournies par les ressources. Cet agent a pour but d'organiser et de gérer les négociations entre agents dans la «Session de Demande ». Il possède ses propres spécifications liées à des opérations de base interne dans la session comme :

- il est le moniteur,
- il ouvre et ferme la session,
- il reçoit, inscrit, accepte, efface les requêtes envoyées par les agents,
- il trie les requêtes selon leur ordre de priorité,
- il transmet aux agents les informations relatives aux autres agents (produits, ressources, etc.) dans le système en temps réel,
- il organise la prise de décision entre les agents en cas de conflit selon leur priorité,
- il garde un contact permanent avec l'entité look-up service pour mettre à jour les nouvelles informations sur les ressources nouvelles ou en panne, fournissant le service qui le concerne.

Dans cette thèse, l'ASD intègre l'architecture inspirée de PABADIS et joue un rôle important dans la méthodologie et les protocoles de négociation des agents dans le système. Afin de récapituler et d'éclaircir les différentes interactions de base entre les différents agents nous proposons ce schéma :



**Figure 4.6 : Schéma global des interactions entre les différentes entités du système**

La figure 4.6 montre les différentes interactions entre les différentes entités du système. L'Agency génère les «Agent-Produit» (AP) et garde le contact avec eux jusqu'à la fin de leurs tâches à exécuter. Les «Agent-Produit» communiquent avec le Look-Up Service (LUS) pour avoir les adresses des autres agents dans le système, ils communiquent également avec l'Agent de «Session de Demande» (ASD) et l'«Agent-Ressource» (AR) pour l'allocation des ressources. Une fois qu'ils ont terminé leur mission, ils transmettent les résultats à l'Agency. Les «Agent-Ressource» communiquent avec les «Agent-Produit» pour répondre à des demandes de service par l'intermédiaire des protocoles de négociation, de même ils communiquent avec le LUS et l'ASD pour les informer sur leur statut durant la production. Les ASD gèrent les «Session de Demande» tout en communiquant avec les «Agent-Produit» et l'«Agent-Ressource», et gardent un contact permanent avec le LUS pour mettre à jour les nouvelles informations concernant les «Agent-Ressource». Finalement, le LUS communique avec les différents types d'agents en leurs transmettant les adresses et les types d'agents dans le système.

### **7.3 Cycle de production**

Le processus de la production débute, quand une « Demande de Production » est introduite dans le système. Cette dernière, concerne la fabrication d'un certain nombre de produits par le système. Elle peut être envoyée directement par une entité externe au système, par exemple un client, ou par un ordre d'approvisionnement selon le mode de fonctionnement « Pull » ou par un ordre de fabrication issue de l'ERP. La « Demande de Production » sera analysée par l'« Agency » en prenant en considération l'état global du système avant de générer un certain nombre d'«Agent-Produit» pour répondre à cette demande. Le nombre d'agent générés est lié seulement au nombre de produits mentionnés dans la demande.

Une fois généré, l'«Agent-Produit» attend dans le système l'introduction du Produit Physique (PP) auquel il sera associé. Il effectue ensuite l'allocation des prochaines tâches à exécuter par le produit à travers des protocoles de négociation avec les «Agent-Ressource» fournissant les services voulus. Etant mobile et associé à un produit physique, l'«Agent-Produit» l'accompagne réellement durant les différentes étapes de sa fabrication. Après l'achèvement de traitement de toutes ses tâches, il transmet à l'Agency le résultat qu'il a réalisé et s'arrête.

Dans la figure 4.7, nous montrons le déroulement graphique du cycle de production dans un atelier expérimental. Nous considérons dans le cas général un atelier de production composé de  $m$  ressources réparties en trois groupes. Chaque groupe est composé de  $n$  ressources fournissant le même type de tâche. L'atelier considéré ici fourni trois types de tâches : Percer, Nettoyer et Assembler.

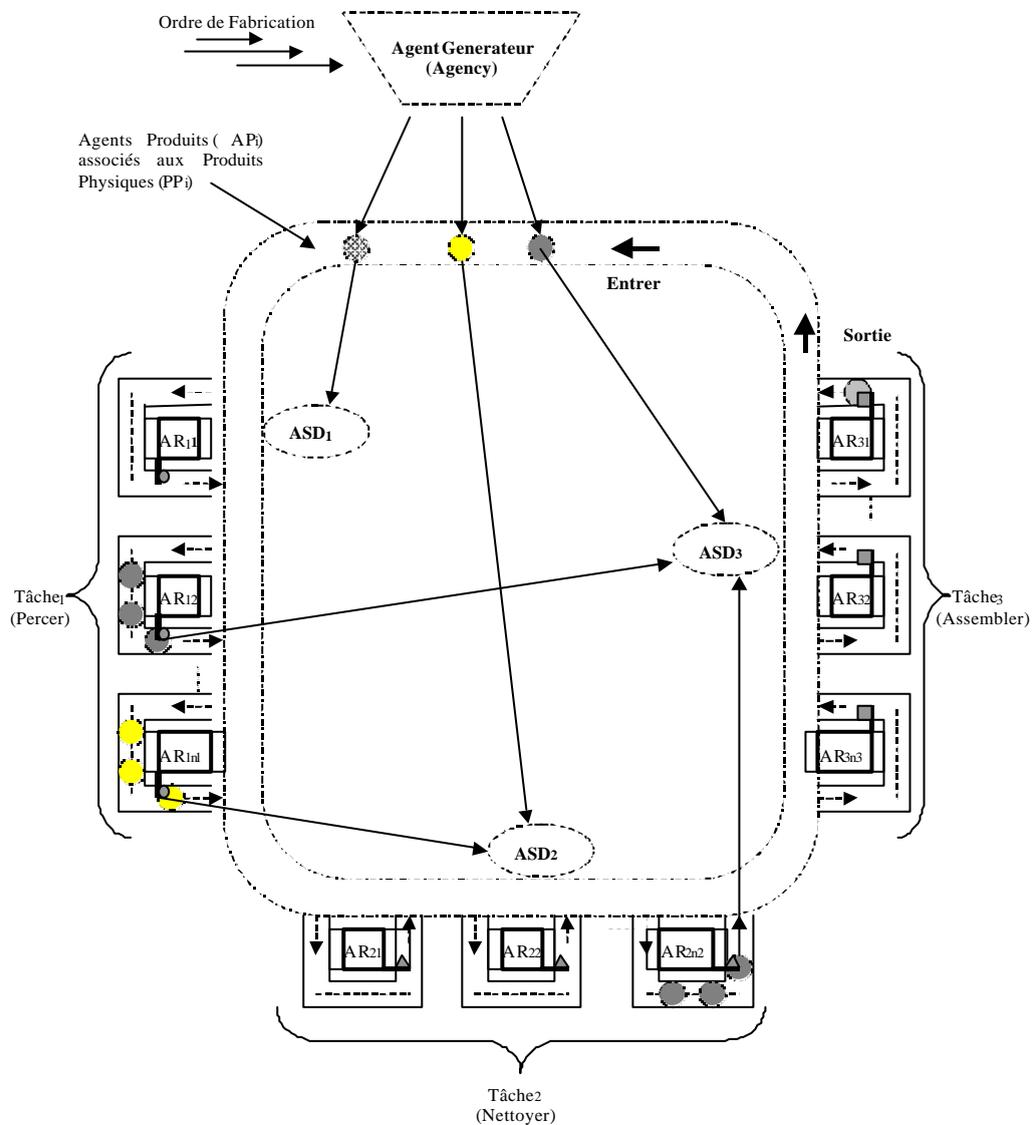


Figure 4.7 : Modélisation expérimentale d'un atelier de production

La « Demande de Production » est représentée par un ordre de fabrication, les  $AP_i$  et  $AR_{jk}$  représentent respectivement les « Agent-Produit » et les « Agent-Ressource » dans le système ( $i$  correspond au numéro du produit généré ;  $j = 1$  à  $3$  correspond au numéro des tâches fourni dans le système ;  $k = 1$  à  $m$  ( $m=9$  dans cet exemple) correspond aux numéros de ressource ;  $n_1, n_2, n_3$  correspondent respectivement au nombre des ressources pour la tâche<sub>1</sub>, tâche<sub>2</sub>, tâche<sub>3</sub>,  $n_1+n_2+n_3 = m$ ). Dans cet atelier, nous distinguons trois Agent de « Session de Demande » ( $ASD_j$ ). Leur nombre correspond au nombre des tâches fourni par les « Agent-Ressource ». Chaque « Agent-Produit » a besoin d'exécuter un certain nombre de tâche selon un ordre de tâche et une date d'échéance fixé à l'avance. Le temps d'arrivé d'un « Agent-Produit » dans le système de n'est pas connu à l'avance. Cet exemple expérimental est choisi pour faciliter la compréhension de la modélisation du système utilisé.

Le temps des traitements de tâche estimé ( $T_e$ ) des produits sur les ressources est une information essentielle pour l'allocation dynamique des ressources. En effet, il sert en grande partie comme « estimation » du temps nécessaire au produit pour respecter une date d'échéance ou « Due Date » (DD) (le temps limite pour accomplir ses tâches). Ce temps peut être calculé en utilisant une base de données de connaissances ou Knowledge Base System (KBS) [Usher et Wang, 2000b] selon le statut des ressources dans le système, ou en utilisant des estimations nominales de temps de traitement fixé à l'avance [Krothapalli et Deshmukh, 1999]. Dans cette thèse, ce temps est calculé dynamiquement, et en temps réel. Sa valeur dépend du temps total de traitement pris pour une tâche donnée, au même instant de la demande, par la ressource la plus lente fournissant la tâche demandée. Le temps de transport d'une ressource à une autre est pris en compte dans ce modèle et calculé par les « Agent-Ressource » comme une estimation de temps de transport entre deux adresses ou deux points dans un plan. Les produits sont traités dans les files d'attente des ressources selon le principe « First-In-First-Out » (FIFO). Enfin, il reste à signaler qu'on utilise un buffer auxiliaire pour chaque ressource. Celui-ci sert à faire attendre les produits qui arrivent à l'avance sur une ressource, en ne respectant pas l'ordre d'entrée des produits dans les files d'attente des ressources selon l'ordre d'allocation déclaré.

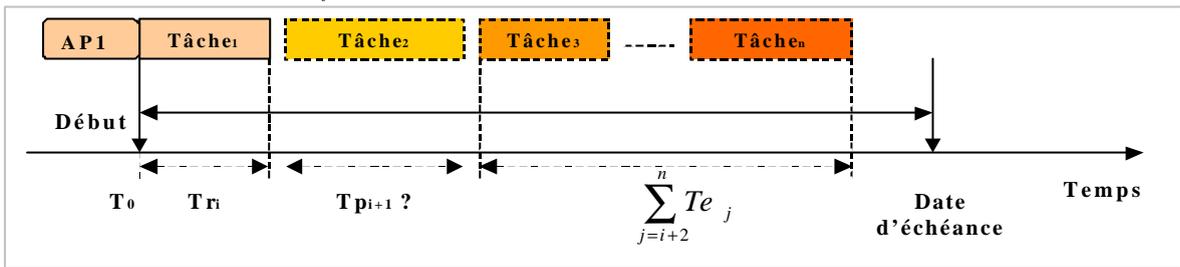
## 8 Méthodologie de Négociation

Durant l'allocation des tâches relatives aux produits, l'« Agent-Produit » utilise une méthodologie de négociation avec les autres agents qui est étroitement liée à l'architecture du système et des protocoles de négociations utilisés. Dans ce paragraphe, nous développons une méthodologie de négociation adaptée à l'architecture proposée d'un système de production décentralisée et qui sera appliquée dans plusieurs protocoles de négociation à base de l'approche SSPR.

L'« Agent-Produit » commence son processus d'allocation des tâches une fois le produit physique qui lui est associé introduit dans le système. Il essaie alors

d'allouer sa prochaine tâche. Pour cela, il demande au «Look-Up Service» l'adresse de l'Agent de la «Session de Demande»(ASD) relative à la tâche voulu ; une fois trouvée, il s'inscrit dans la session auprès de lui, et lui transmet le Temps de traitement restant ( $Tr_i$ ) pour sa tâche actuelle  $i$  et le Temps de traitement permis ( $Tp_{i+1}$ ) pour sa prochaine tâche. L'«Agent-Produit» calcule le  $Tp_{i+1}$  chaque fois qu'il participe à une «Session de Demande» en utilisant la formule suivante ( $n$  étant le nombre de tâche à effectuer ;  $i, j \in N$ ) :

$$Tp_{i+1} = Tdd - (Tr_i + \sum_{j=i+2}^n Te_j) \quad (4.1)$$



**Figure 4.8 : Calcul du Temps de traitement permis ( $Tp_{i+1}$ ) pour la prochaine tâche**

Dans la figure 4.8, nous supposons que l'«Agent-Produit» ( $AP_1$ ) a besoin d'effectuer  $n$  tâches. A l'instant  $T_0$ ,  $AP_1$  commence sa première tâche «Tâche<sub>1</sub>» sur un «Agent-Ressource» (AR). Le temps permis  $Tp_2$  pour sa Tâche<sub>2</sub> est calculé selon la formule 4.1 tout en prenant en considération le temps restant  $Tr_1$  pour la Tâche<sub>1</sub>, le temps restant avant la Due Date ( $Tdd$ ) de l'«Agent-Produit» et le Temps de traitement estimé ( $Te$ ) à effectuer pour les tâches restants sans la Tâche<sub>2</sub>.

L'Agent de la «Session de Demande» ajoute ou inscrit la demande de l'«Agent-Produit» à la session. Il continue à recevoir les demandes des agents jusqu'au temps de décision dans la session. Ce temps dépend de l'achèvement de traitement (temps restant de traitement proche de Zero) de la tâche d'un agent. Dorénavant, nous désignons par le nom APZ, l'«Agent-Produit» déclenchant le temps de décision dans la «Session de Demande». Ainsi, L'APZ demande aux «Agent-Ressource» de lui fournir une de la Durée du traitement estimée de sa prochaine tâche ( $Dt_{i+1}$ ). Ces derniers calculent la durée en prenant en considération le Temps de traitement restant  $Tr_i$  pour la tâche  $i$  en cours de fabrication, le Temps de transport ( $T$ ) entre les deux ressources, le Temps de traitement ( $Tt$ ) pour effectuer une seule tâche et le Nombre des agents dans sa File d'attente ( $NF$ ). L'«Agent-Ressource» utilise la formule suivante pour calculer la durée du traitement demandée :

$$Dt_{i+1} = Tr_i + T + (NF+1) * Tt \quad (4.2)$$

L'APZ utilise des règles et des méthodes d'aide à la décision comme les règles de dispatching, la théorie des jeux, etc. pour choisir la meilleure offre proposée par

les «Agent-Ressource» pour satisfaire ses intérêts. Ainsi, il alloue un «Agent-Ressource», quitte la « Session de Demande » et migre vers l'agent alloué.

Cette méthodologie de négociation sera utilisée dans les nouveaux protocoles de négociation développés dans la thèse.

## **9 Protocoles de négociation développés**

Les Protocoles de Négociations (PN) développés dans la thèse appartiennent aux catégories de décisions suivantes : coopératives, compétitives et coopétitives.

Dans le cadre de la compétition, nous utilisons trois protocoles existants qui diffèrent entre eux par rapport à l'approche utilisée et les algorithmes implémentés (simple ou sophistiqué). Les deux premiers protocoles de négociations utilisent des stratégies égoïstes, (représentées par le principe « First-Come-First-Serve (FCFS)»). Quant au dernier, il utilise l'approche SSPR tout en appliquant des algorithmes sophistiqués pour l'aide à la décision.

Concernant l'aspect coopératif, deux nouveaux protocoles de négociation seront détaillés. Le premier applique le principe de la «Session de demande» pour le partage des information et des intérêts entre les « Agents-Produits ». La prise de la décision de ces agents est altruiste et basée sur la notion de priorité. Le second protocole essaye d'apporter la notion de concession en prenant en considération les intérêts des autres agents tant que son propre intérêt est respecté.

Finalement, la notion de coopétition sera appliquée dans deux nouveaux protocoles de négociation par le biais de la coopération, à travers l'anticipation des « Agent-Produit » à la «Session de Demande» et la compétition, lors de la prise de la décision en appliquant la théorie des jeux. Dans le premier protocole, le principe d'anticipation à la session sera local et décentralisée. Tandis que dans le second l'anticipation sera globale et centralisée. Ceci est fait permet de comparer les performances d'une stratégie appliquée selon les deux façons décentralisée et centralisée.

La démarche utilisée en variant les catégories de décision dans ces protocoles de négociation à pour but d'élargir le domaine de décision, et d'explorer de nouveaux aspects stratégiques dans un environnement dynamique à forte perturbation. En effet, l'allocation dynamique des ressources à travers une approche décentralisée et auto-organisée dans un tel environnement nécessite de nouveaux types de protocoles pour faire face à l'instabilité des performances des protocoles existants. Les sept protocoles de négociation mentionnés ci-dessus seront testés, validés et comparés dans le prochain chapitre.

## 9.1 Protocole «*Production Reservation*» (PR)

Ce modèle propose une nouvelle approche pour l'allocation des ressources d'un système de production décentralisé. En fait, l'allocation ou la réservation est réalisée pour chaque Ordre de Fabrication (OF) séparément dans le système de production et non pour tous les ordres en même temps. Le terme «réserver» utilisé ici signifie aussi allouer. Différents travaux ont appliqué ce modèle pour l'allocation des ressources dans un système décentralisé [Baker, 1991 ; Shaw, 1987 ; Saad et al., 1997 ; etc.]. Dans ce paragraphe, nous détaillons l'approche développée par Saad parce qu'il s'agit d'une approche plus complète que les autres. En plus, elle prend en considération la dynamique du système en permettant aux nouveaux ordre de fabrication, arrivant dans le système, d'alterner l'allocation des ordres de fabrication existants ce qui n'est pas le cas dans les autres travaux. En effet, Saad et al. donne aux ressources la possibilité de sélectionner les produits à traiter selon des règles de dispatching (FIFO, EDD, SPT, etc.) et à partir d'un «buffer» auxiliaire. Le mécanisme de ce modèle se base sur le principe consistant à faire «pousser» le plus vite possible des ordres de fabrication à travers le processus de fabrication, en se basant sur l'état du système au moment de l'entrée de l'ordre de fabrication, de sorte que chaque Produit puisse finir sa tâche au plus tôt.

Conçue sur le principe «Contract-Net Protocol», l'allocation des ressources est réalisée par des négociations entre les Produits et les Ressources. La prise de décision est partagée et bidirectionnelle entre les deux parties. Ce modèle sera détaillé à travers un algorithme et des schémas représentatifs du fonctionnement de l'algorithme. Pour cela, nous citons quelques notions utiles pour le développement de l'algorithme. Chaque ressource possède une Liste de Réservation (LR) et chaque item dans la LR est composé des champs suivants : Numéro de Produit (NP), Début de Temps de Traitement (DTT) et le Temps de Traitement Estimé (TTE). L'ordre d'arrivée des ordre de fabrication n'est connu à l'avance et chaque ordre de fabrication génère un nombre d'«Agents-Produits» équivalent au même nombre de produits demandé dans l'ordre de fabrication. Chaque agent à besoin d'effectuer une liste de tâche pour accomplir sa fabrication. Dans cette approche le terme réservation et allocation ont la même signification.

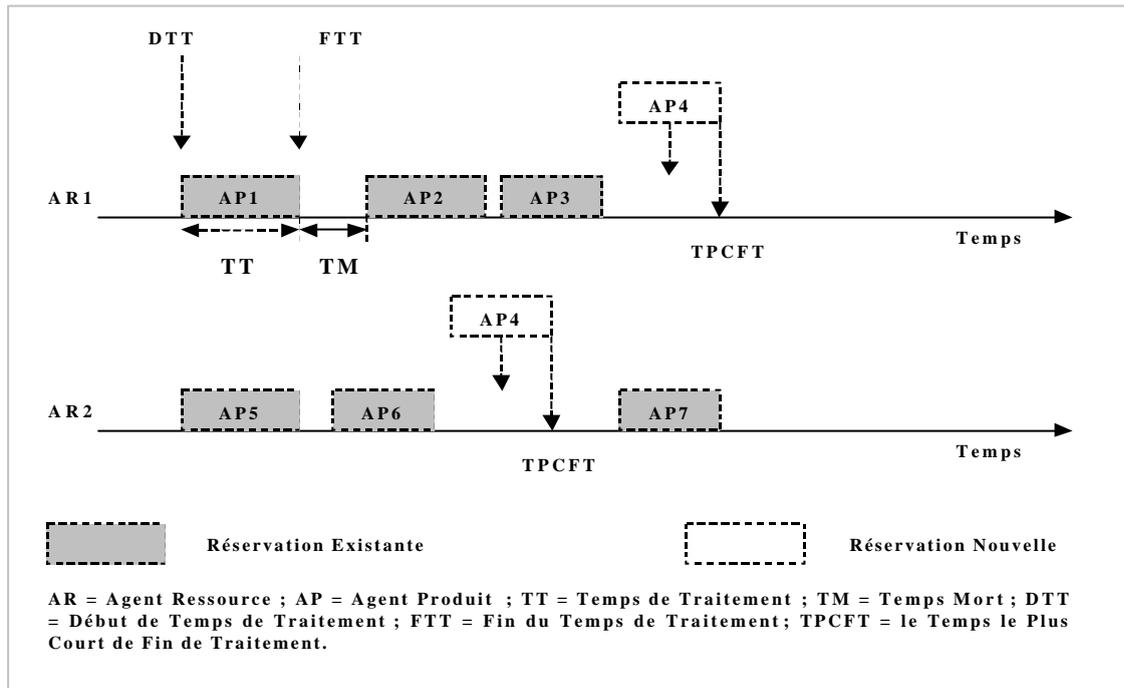
### Algorithme 4.1 : Protocole «*Production Reservation*» (PR)

Pour chaque «Agent-Produit» (AP), le processus d'enchère (Bidding Processus) est effectué pour chaque tâche demandé par lui.

- a) il demande aux «Agent-Ressource» (AR) de fournir des offres pour sa prochaine tâche à effectuer. L'offre est composée de ces deux champs :
  - i. le Temps le Plus Court de Fin de traitement (TPCFT) de la tâche. L'«Agent-Ressource» fait son offre en se basant sur le premier intervalle de temps dans la liste de réservation qui peut être accommodé pour un «Agent-Produit». Par exemple, dans la figure 4.9, l'AR2 trouve une possibilité de traiter la tâche demandée par l'AP4 avant la tâche demandé par l'AP7. Tandis que l'AR1 peut

seulement traiter la tâche demandée après toutes les réservations existantes dans la liste de réservation. Dans l'implémentation de ce modèle, les «Agent-Produit» font leurs réservations selon le principe de «First-In-First-Out». Aucun changement ne sera effectué pour prendre en compte les priorités des Produits selon un nouvel ordre dans la Liste de l'«Agent-Ressource».

- ii. la taille de la liste de réservation pour un «Agent-Ressource», c'est-à-dire le nombre de agents tenant une réservation sur sa liste.



**Figure 4.9 : Exemple pour le modèle « Production Reservation » (PR)**

- (b) il compare les différentes offres données par les «Agent-Ressource» et prend sa décision de la manière suivante :
- il choisit la meilleure offre donnée,
  - En cas d'égalité entre les offres, il choisit l'«Agent-Ressource» avec la plus petite liste de réservation.
- (c) l'«Agent-Ressource» ajoute la demande de l'«Agent-Produit» à sa liste.
- (d) l'«Agent-Produit» répète les étapes (a) .. (c) pour réserver les «Agent-Ressource» pour toutes les tâches restantes à effectuer.
- (e) il commence le processus de traitement de ses tâches sur les «Agent-Ressource» qui leurs sont réservés (voir figure 4.10).

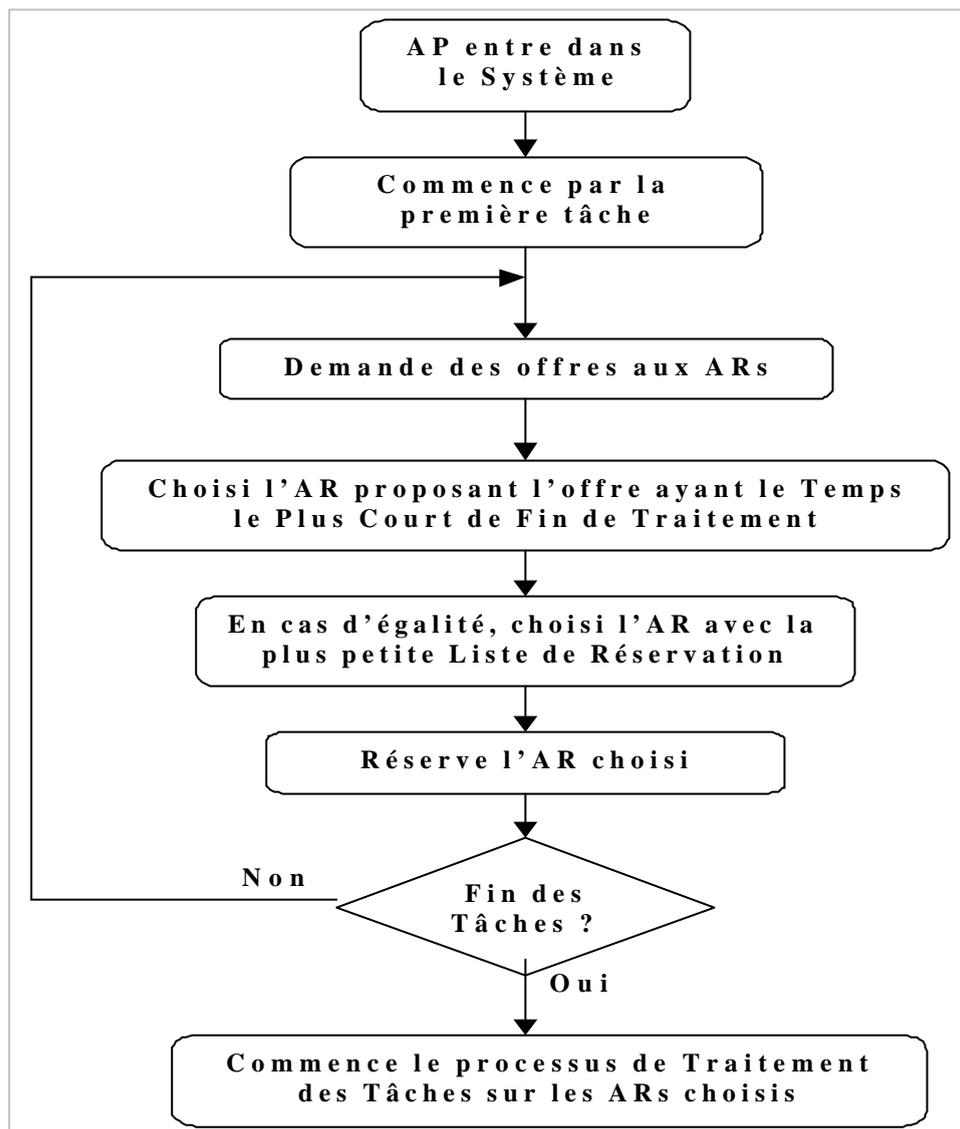


Figure 4.10 : Algorithme du modèle « Production Reservation » (PR)

## 9.2 Protocole « Single-Step Production Reservation » (SSPR)

Ce modèle est appliqué dans plusieurs travaux de recherche pour la pilotage décentralisé des systèmes de production [Holthaus, 1997 ; Saad et al., 1997 ; Krothapalli, 1999 ; Wang and Usher, 2000a].

Il est largement utilisé pour sa simplicité et sa réactivité dans les environnements dynamiques et complexes. L'allocation de ressource dans ce protocole est faite suivant l'approche SSPR en utilisant le principe « First-Come-First-Serve (FCFS) ».

Une fois créé et associé à un produit physique, l'«Agent-Produit» (AP) commence son processus de fabrication par allouer un «Agent-Ressource» (AR) pour sa prochaine tâche. Ensuite, il répète le même processus pour ses prochaines tâches, jusqu'au la dernière tâche à faire.

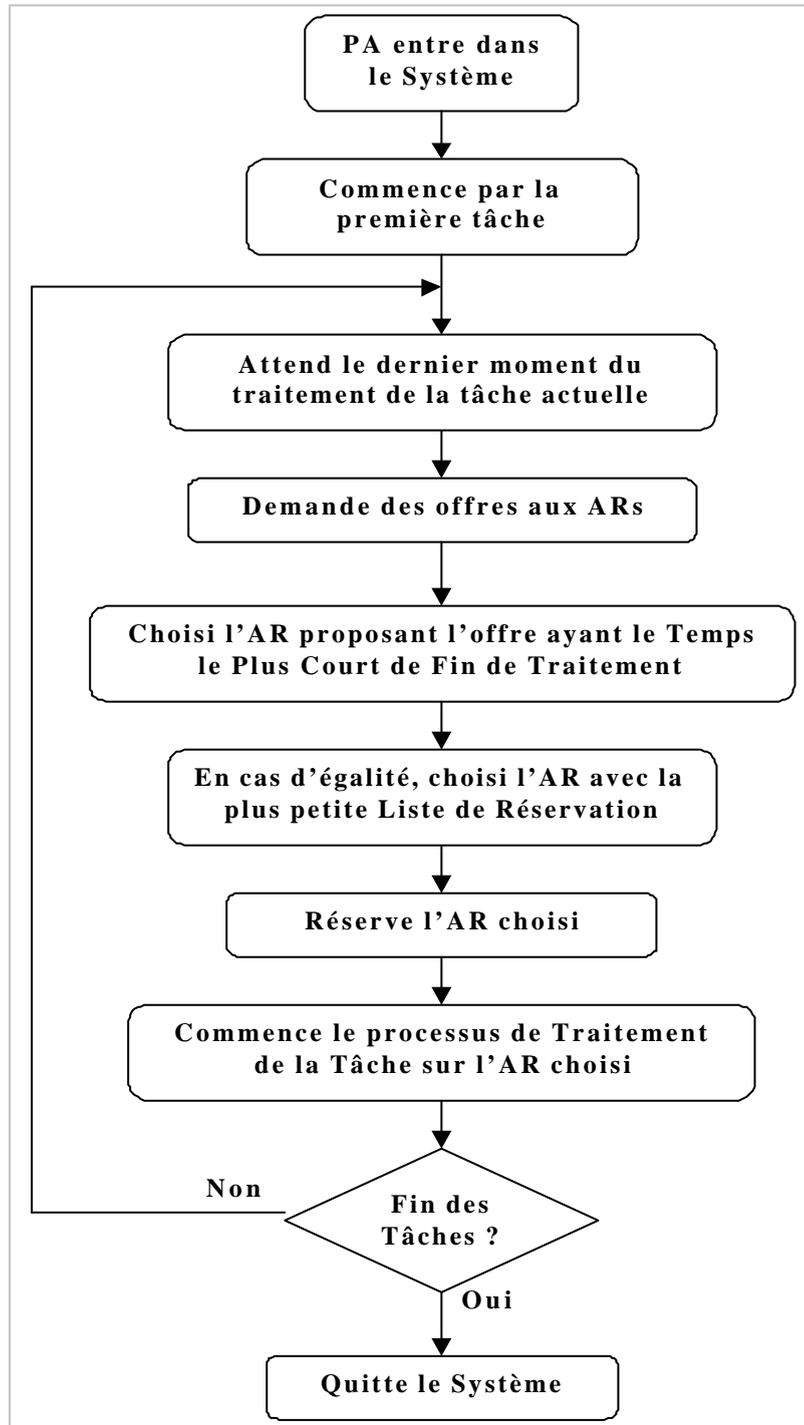
Le moment de la prise de décision, ou l'allocation d'un «Agent-Ressource», par un «Agent-Produit», peut être situé au début, à la fin ou à n'importe quel instant durant le traitement de sa tâche sur un «Agent-Ressource». En raison de la dynamique du système, nous considérons pour ce modèle que le meilleur instant pour allouer une «Agent-Ressource» pour sa prochaine tâche est au dernier moment de son traitement, de cette manière l'«Agent-Produit» prend en considération les variations au niveau des offres et des états (panne, charge, etc.) des «Agent-Ressource» avant de faire son choix.

### Algorithme 4.2 : Protocole « Single-Step Production Reservation » (SSPR)

Comme dans l'algorithme 4.1, Pour chaque «Agent-Produit», le processus d'enchère (Bidding Processus) est effectué pour sa prochaine tâche tâche.

- a) il attend le dernier moment du traitement de la tâche actuelle,
- b) il demande aux «Agent-Ressource» de lui fournir des offres pour sa prochaine tâche à effectuer. L'offre est composée de ces deux champs :
  - i. le temps de fin de traitement de la tâche demandée,
  - ii. la taille de la liste de réservation, c a d le nombre de agents tenant une réservation dans la liste.
- (c) il compare les différentes offres données par les «Agent-Ressource» et prend sa décision de la manière suivante :
  - il choisit la meilleure offre donnée,
  - En cas d'égalité entre les offres, il choisit l'«Agent-Ressource» avec la plus petite liste de réservation.
- (d) l'«Agent-Ressource» choisi, ajoute la demande du l'«Agent-Produit» à sa liste.
- (e) l'«Agent-Produit» commence le processus de traitement de sa tâche sur l'«Agent-Ressource» choisi.

- (f) il répète les étapes (a) .. (e) pour toutes les tâches restantes à effectuer.
- (g) il quitte le système une fois qu'il a accompli toutes ses tâches à effectuer (voir figure 4.11).



**Figure 4.11 : Algorithme du modèle « Single-Step Production Reservation » (SSPR)**

### 9.3 Protocole « Currency »

G. Lin et J. Solberg étaient les premiers à introduire la notion monnaie d'échange ou « Currency » entre les produits et les ressources dans le système de production décentralisé [Lin et Solberg, 1991]. Ce principe est basé sur le principe de marché d'échange ou « market pardigme » ; si un client demande un service fourni par une organisation, un coût ou monnaie d'échange sera exigé par l'organisation. Ce modèle utilise un mécanisme de construction générique des offres d'échange durant la négociation entre les agents, tout en se basant sur le principe de la combinaison du prix et de l'objectif (temps, coût, qualité, etc.).

N. Krothapalli et A.Deshmukh ont complété ce dernier modèle en considérant la relation non linéaire entre la « Currency » et le temps [Krothapalli et Deshmukh, 1997]. Pour cela, ils utilisent une fonction de calcul et d'attribution de « Currency » aux agents en se basant sur les notions de « Slack » disponible pour les produits et les charges de travail pour les machines.

Pour cela, considérons les «Agent-Produit» comme des clients et les «Agent-Ressource» comme des fournisseurs de service ou des tâches dans le système. Le temps de traitement de tâche est fixé à l'avance dans le système et considéré comme une estimation nominale prise par la ressource la plus lente fournissant la tâche voulue. L'«Agent-Produit» entre dans le système avec la monnaie dépendant de sa priorité. L'«Agent-Ressource» calcule sa charge de travail en se basant sur le temps de fin de traitement et sur son taux de réussite pour les appels d'offre et les engagements faits par les «Agent-Produit». Ce dernier converti le temps de fin de traitement donné par les «Agent-Ressource» en monnaie et considère seulement les offres des «Agent-Ressource» ayant un temps de fin de traitement plus petite que son « Due Date » (DD) et la plus petite charge de travail respectant la valeur de son « Currency ».

Ce modèle utilise l'approche SSPR, il est développé dans ce paragraphe à travers deux types de protocoles utilisés par les «Agent-Produit» et les «Agent-Ressource» et des schémas représentatifs du fonctionnement de ces protocoles. La possibilité que les ressources tombent en panne n'est pas évoquée dans ce modèle-ci comme elle était faite dans le modèle de N. Krothapalli et A.Deshmukh.

#### 9.3.1 Protocole de l' «Agent-Produit »

L'«Agent-Produit» entre dans le système avec la monnaie dépendant de son DD et de son temps de traitement nominale de ses tâches. Il demande aux «Agent-Ressource» de lui envoyer des offres concernant leurs temps de fin de traitement. Il évalue les offres envoyées après un temps d'attente constant fixé à l'avance dans le système. L'«Agent-Produit» ayant n tâches à accomplir, calcul le due date pour chaque tâche en utilisant cette formule :

$$d_i = D \left( \frac{m_i}{M} \right) \quad \forall i = 1, \dots, n \quad (4.3)$$

Où  $d_i$  est la due date pour la tâche  $i$ ,  $D$  est le due date pour les  $n$  tâches,  $m_i$  est le temps de traitement nominale de la tâche  $i$  et  $M$  est la somme de temps de traitement nominale pour les  $n$  tâches de l'«Agent-Produit».

En considérant que  $C$  est la «Currency» pour les  $n$  tâches de l'«Agent-Produit», La «Currency»  $C_i$  attribuée à chaque tâche est calculée de la manière suivante :

$$C_i = C \left( \frac{m_i}{M} \right) \quad \forall i = 1, \dots, n \quad (4.4)$$

L'«Agent-Produit» convertit le temps de fin de traitement des «Agent-Ressources» en «Currency» par l'intermédiaire de cette fonction :

$$C_p = 2.0 \tan^{-1} \left( e^{Aw} \right) \quad 0.01 \leq A \leq 0.1 \quad (4.5)$$

Où  $C_p$  est équivalent à la Currency par Unité de Temps (UT),  $A$  est une fonction qui détermine la courbature de la fonction  $C_p$ ,  $W$  est égale à la valeur du le temps de fin de traitement de l'«Agent-Ressource».  $A$  est calculée en fonction de  $d$  et  $m_i$  de la manière suivante :

$$A = 0.11 - \frac{1 + \tanh \left( \frac{d_i}{m_i} \right)}{20} \quad (4.6)$$

$C_p$  et  $A$  ont été choisi dans ce protocole pour refléter l'asymétrie du temps quand nous comparons l'avance ou le retard d'un produit. Ces deux fonctions ont été utilisées dans le domaine de la prise de décision humaine sous risque [Kahneman and Tversky, 1979]. L'algorithme de ce protocole est représenté par l'algorithme 4.3 :

**Algorithme 4.3 : Protocole de l'«Agent-Produit» selon le principe «Currency»**

le processus d'enchère (Bidding Processus) est effectué pour chaque tâche demandé par l'«Agent-Produit» (AP) de la manière suivante :

- a) il demande aux «Agent-Ressource» (AR) de lui fournir des offres pour sa prochaine tâche. L'offre est composée de ces deux champs :
  - i. le temps de fin de traitement,
  - ii. la charge de travail.
- b) il évalue les offres après un temps d'attente constant,
- c) il compare les différentes offres données et considère les offres ayant :
  - un temps de fin de traitement plus petit que son Due Date,
  - une charge de travail plus petite que son «Currency».
- (d) il choisit l'«Agent-Ressource» ayant la plus petite charge de travail.
- (e) l'«Agent-Ressource» choisi ajoute la demande de l'«Agent-Produit» à sa liste de réservation.

- (f) l'«Agent-Produit» commence le processus de traitement de sa tâche sur l'«Agent-Ressource» choisi.
- (g) il répète les étapes (a) .. (f) pour toutes les tâches restantes à effectuer.
- (h) il quitte le système une fois qu'il a accompli toutes ses tâches à effectuer (voir figure 4.12).

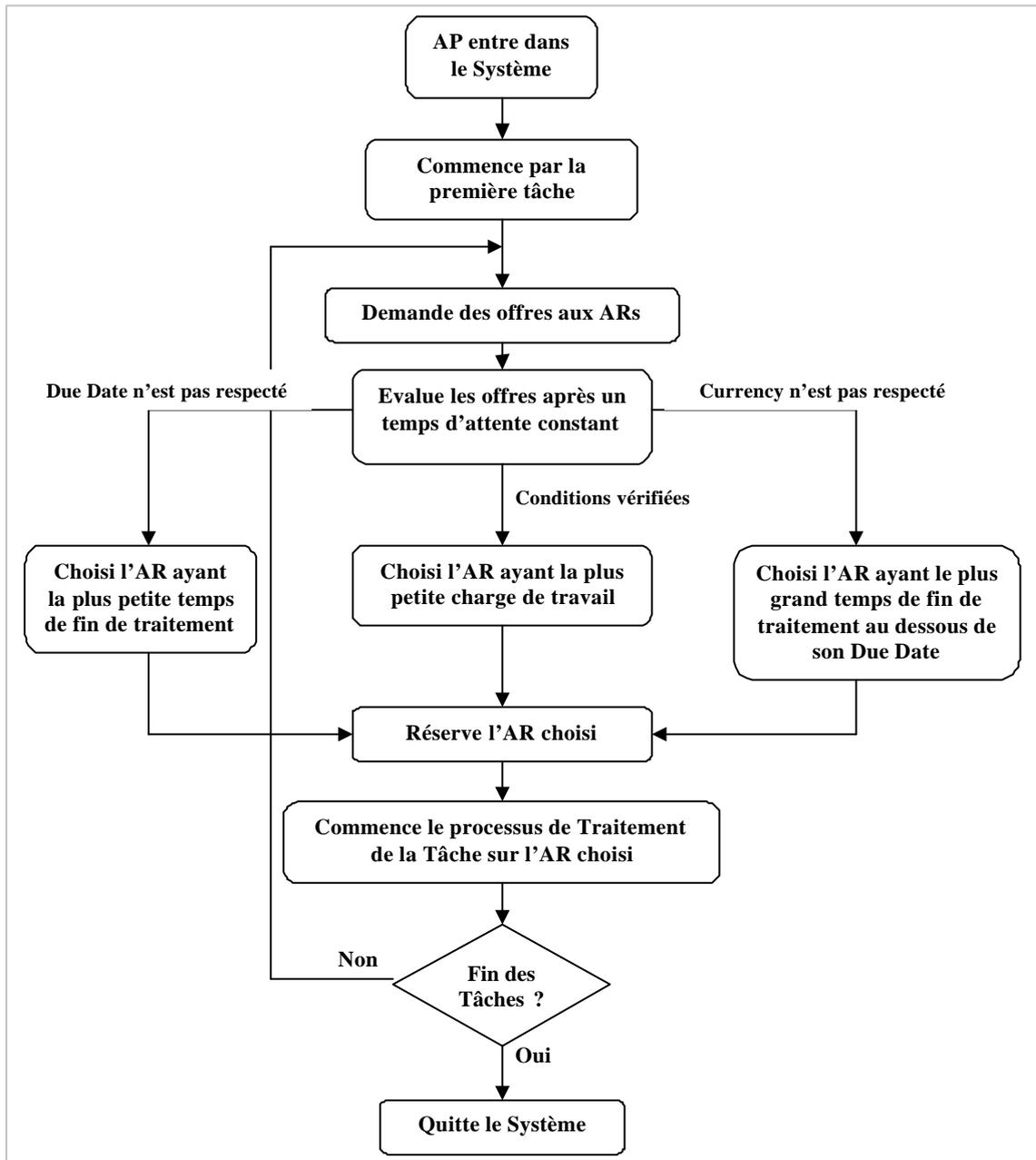


Figure 4.12 : Algorithme du modèle « Currency » utilisé par l' «Agent-Produit »

### 9.3.2 Protocole de l' « Agent Ressource »

Les « Agent-Ressource » attendent dans le système les demandes des tâches des «Agent-Produit». Ensuite, il propose des offres d'achat de service correspondant à leurs temps de fin de traitement pour une tâche donnée. L'«Agent-Ressource» calcule la valeur de son offre de la manière suivante :

$$w = Q + \frac{Q}{b} \Psi \beta_K + \frac{m_i}{S} \quad (4.7)$$

Où  $Q$  est le temps d'attente dans la Queue ou file d'attente,  $b$  est le nombre des «Agent-Produit» dans la file d'attente de l'«Agent-Ressource»,  $\hat{\alpha}_k$  est le nombre de demandes effectuées par les «Agent-Produit» dans les derniers  $K$  unité de temps,  $K$  est le temps d'attente d'un «Agent-Produit» avant d'évaluer les offres des «Agent-Ressource»,  $\phi = \hat{\alpha}_k / \hat{\alpha}_T$  est le taux de succès d'un «Agent-Ressource»,  $\phi = \hat{\alpha}_w / \hat{\alpha}_T$  où  $\hat{\alpha}_w$  est le nombre des demandes qui ont été conclu par l'«Agent-Ressource» dans la dernière heure et  $\hat{\alpha}_T$  est le nombre total des demandes effectuées par les «Agent-Produit»,  $S$  est la proportion de temps de temps de traitement d'une ressource par rapport à la ressource la plus lente fournissant la même tâche.

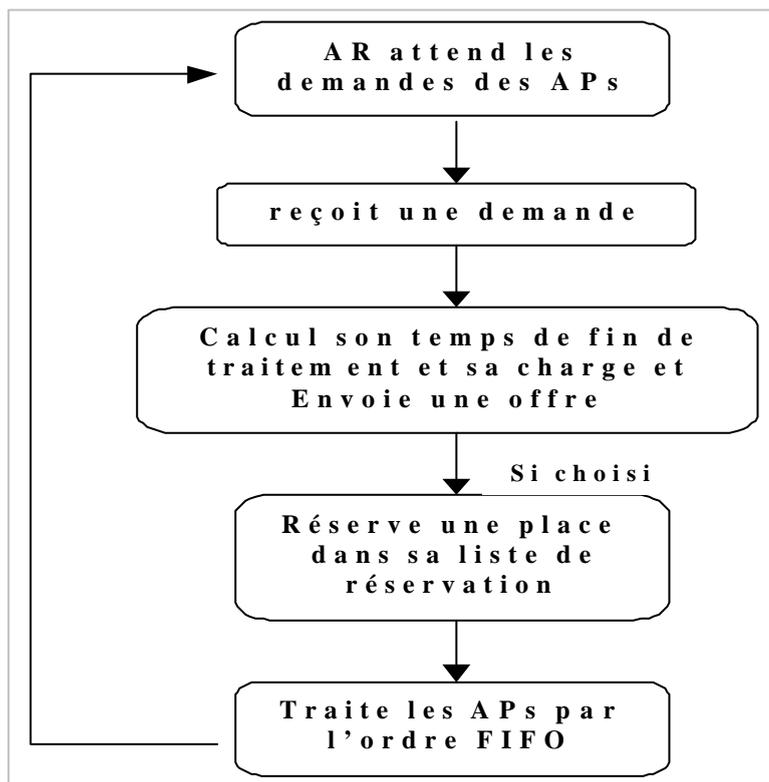


Figure 4.13 : Algorithme du modèle « Currency » utilisé par l'«Agent-Ressource»

Le coût d'un service fourni par une ressource est donné par cette fonction :

$$C_r = 4.7 - \left( 2.0 \tan^{-1} \left( e^{(0.11-\beta)\omega} \right) \right) \quad 0.01 \leq \beta \leq 0.1 \quad (4.8)$$

Où  $C_r$  est la charge de travail d'une Ressource par Unité de Temps,  $\hat{\alpha}$  est une fonction qui détermine la courbure de la fonction  $G_r$ ,  $\hat{\alpha}$  dépend de  $\varnothing$  et de  $w$  et elle est calculée de cette manière :

$$B = 0.1 - \frac{1 + \tanh(1 - 0.1 \Psi)}{20} \quad (4.9)$$

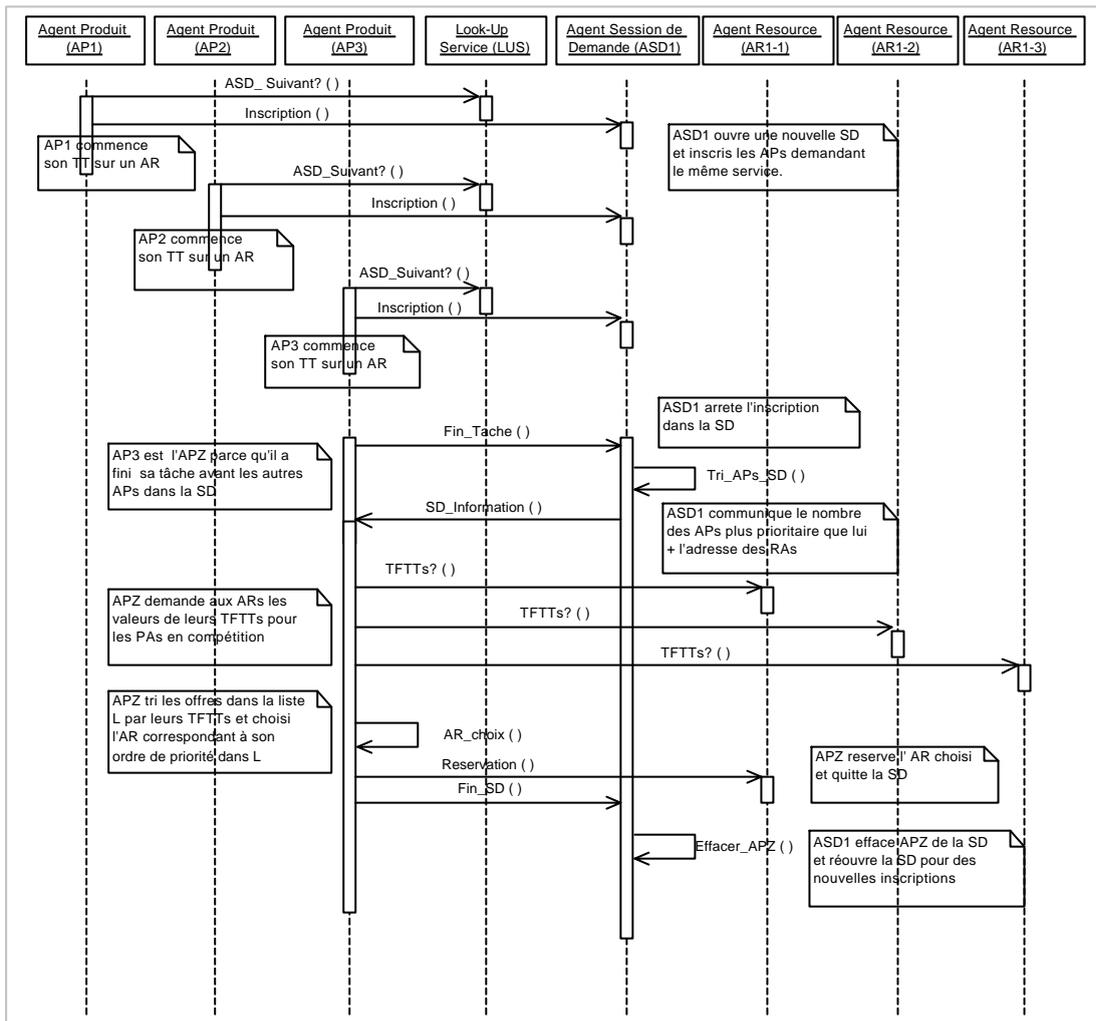
Une fois que l'«Agent-Produit» a choisi l'«Agent-Ressource», ce dernier réserve pour lui une place dans sa liste de réservation. L'«Agent-Ressource» traite les agents dans sa file d'attente par l'ordre FIFO. L'algorithme de ce protocole est représenté par la figure 4.13.

#### **9.4 Protocole « Consensuel selon la priorité »**

Dans ce modèle, la notion de priorité entre les «Agent-Produit» dans la «Session de Demande» est privilégiée. Celle-ci est représentée par le paramètre Temps de Traitement Permis (Tp) pour la prochaine tâche de chaque «Agent-Produit» et elle est calculée par l'intermédiaire de la formule 4.1. Ce protocole utilise la même méthodologie de négociation développée dans le paragraphe 4.5. L'agent (ayant le temps restant de son traitement actuel proche de Zero) ou **APZ** prend sa décision en faisant des concessions pour tous les «Agent-Produit» prioritaires dans la «Session de Demande». La prise de la décision de l'APZ est détaillée dans l'Algorithme 4.4 et la figure 4.14.

##### **Algorithme 4.4 : Allocation consensuelle selon la priorité**

- a) l'APZ signale la fin de son traitement à l'Agent de la «Session de Demande» (**ASD**),
- b) l'ASD arrête l'inscription dans la Session de Demande (**SD**),
- c) il trie tous les «Agent-Produit» par leurs temps de traitement permis,
- d) il communique à l'APZ le nombre des «Agent-Produit» ayant des temps de traitement permis plus petits que lui (ou prioritaire) et les adresses des «Agent-Ressource» fournissant la tâche voulue,
- e) l'«Agent-Produit» demande à chaque «Agent-Ressource» de lui fournir une offre pour sa prochaine tâche à effectuer.
- f) il trie dans une liste les offres fournies par un ordre croissant et choisit l'offre qui correspond à son ordre de priorité dans la liste triée. Si le nombre des agents prioritaires dépasse le nombre d'offres fournies, l'APZ choisira alors la dernière offre dans la liste ayant le plus grand temps de fin de traitement,
- g) l'APZ alloue l'«Agent-Ressource» choisi, signale la prise de sa décision à l'ASD et quitte la session.



**Figure 4.14 : Prise de décision d'un APZ par concession selon la priorité**

Les sigles utilisés dans les diagrammes de séquence UML, figures 4.14, 4.15, 4.16, 4.17, 4.18, 4.19, correspondent aux notions suivantes. AP= «Agent Produit» ; AR=«Agent Ressource» ; APZ=«Agent Produit»(ayant le temps restant de son traitement actuel proche de Zero) ; ASD=Agent de la «Session de Demande» ; SD=«Session de Demande» ; TT=Traitement de Tache ; TTP=Temps de Traitement Permis ; TFFT= Temps de Fin de Traitement de Tâche ;

### 9.5 Protocole « Consensuel avec contraintes »

Ce protocole de négociation ajoute une contrainte à respecter pour la prise de la décision dans la « Session de Demande » au protocole précédent. En effet, L'APZ fait des concessions pour les agents bénéficiant d'une priorité, **Tant Que** son temps de traitement permis pour sa prochaine tâche est respecté (ou n'est pas dépassé). Ce modèle est détaillé selon deux algorithmes développés dans le cadre de notre travail de recherche. Dans le premier algorithme la prise de la décision de l'APZ est faite à

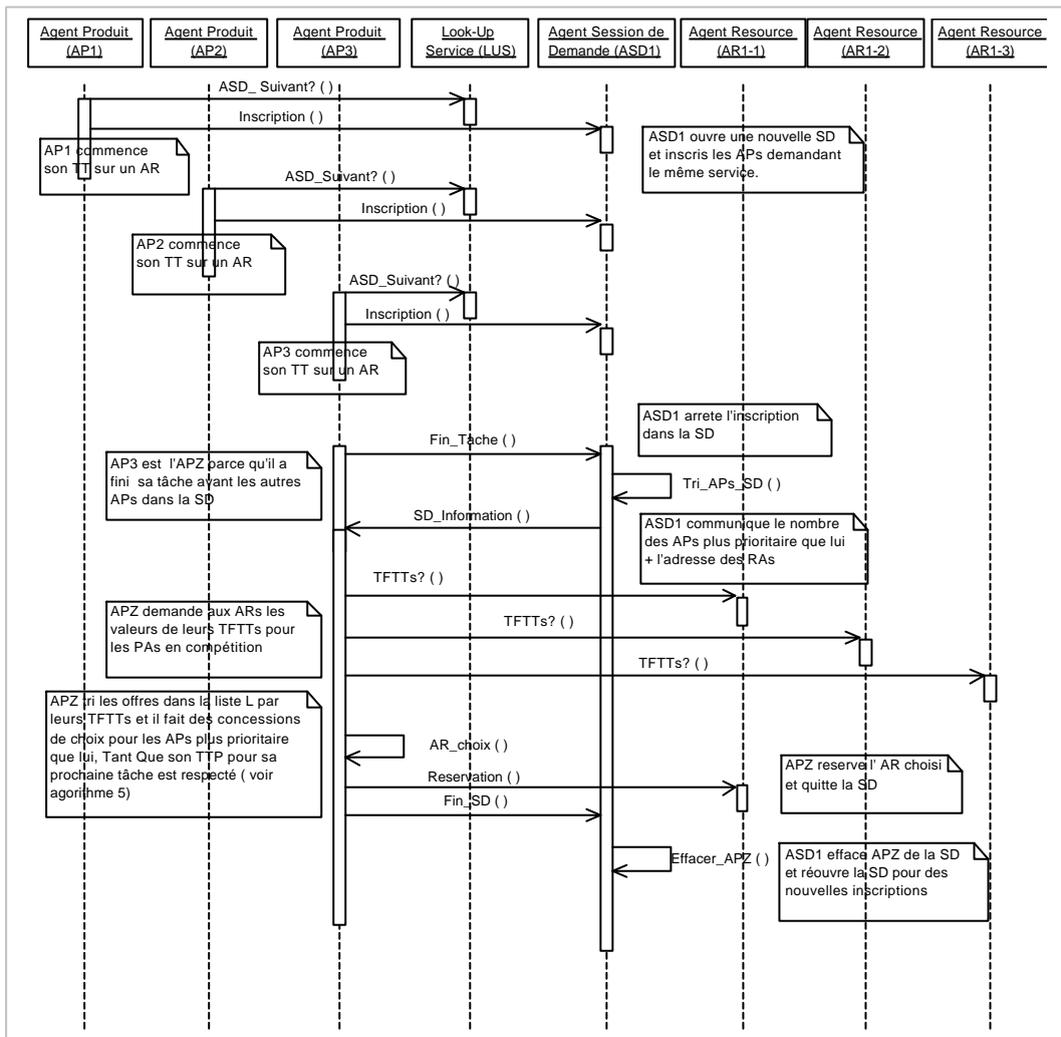
travers une fonction de tri et de sélection des agents prioritaires dans la session (voir algorithme 4.5-a et la figure 4.15). Tandis que dans le second algorithme, la concession de l'APZ est établie par l'intermédiaire d'un échange de négociation et de prise de la décision entre les agents dans la session. Dans ce dernier algorithme, l'ASD joue un rôle primordial dans la répartition de la décision entre les «Agent-Produit», tout en respectant priorités et contraintes de chacun d'eux (voir algorithme 4.5-b, la figure 4.16 et la figure 4.17).

**Algorithme 4.5-a : Consensuel avec Contraintes**

- a) l'APZ signale la fin de son traitement actuel à l'ASD,
- b) l'ASD arrête l'inscription dans la «Session de Demande » (SD),
- c) il trie tous les «Agent-Produit» par leurs temps de traitement permis,
- d) il communique à l'APZ le nombre des «Agent-Produit» ayant des temps de traitement permis plus petits que lui (ou prioritaire) et les adresses des «Agent-Ressource» fournissant la tâche voulue,
- e) l'«Agent-Produit» demande à chaque «Agent-Ressource» de lui fournir une offre pour sa prochaine tâche à effectuer.
- f) Soit  $m$  le nombre des «Agent-Ressource» dans le système,  $NP$  le nombre des «Agent-Produit» plus Prioritaires que APZ dans la session,  $i$  est l'indice dans la liste des offres,  $OF_i$  représente l'offre  $i$  dans la liste des offres et  $Choix$  correspond à l'indice choisi. Nous initialisons à 1 l'indice  $i$  ( $i = 1$ ). L'APZ choisit l'«Agent-Ressource» convenable à sa contrainte de la manière suivante :
  - 1. Tri les offres des « Agent-Ressource » par ordre croissant dans une liste L, chaque élément de la liste est composé par le nom (ou adresse) et la valeur de l'offre,
  - 2. **Tant Que** ( $(i \leq m)$  et  $(i \leq NP)$  et  $(TTP - OF_i \geq 0)$ ) **Faire**
    - Début*
    - $i = i + 1$
    - Fin*
    - Si** ( $i == 1$ ) **Alors**
      - Début*
      - $Choix = OF_1$
      - Fin*
    - Sinon**
      - Début*
      - $Choix = OF_{i-1}$
      - Fin*
- g) L'APZ choisit l'«Agent-Ressource» correspondant à la valeur de l'indice  $Choix$  dans la liste,

h) L'APZ alloue l'«Agent-Ressource» choisi, signale la prise de sa décision à l'ASD et quitte la « Session de Demande ».

Dans la partie expérimentale de la thèse l'algorithme 7-a est utilisé pour représenter ce modèle et non l'algorithme 7-b. Bien que les deux algorithmes donnent le même résultat, le premier algorithme a été préféré parce qu'il est plus simple à implémenter et plus rapide à réagir en temps réel.



**Figure 4.15 : Prise de décision consensuelle d'un APZ selon la contrainte de temps de traitement permis**

#### Algorithme 4.5-b : Consensuel avec Contraintes

- l'APZ signale la fin de son traitement actuel à l'ASD,
- l'ASD arrête l'inscription dans la «Session de Demande » (SD),
- il trie tous les «Agent-Produit» par leurs temps de traitement permis,

- d) L'ASD communique à l'APZ les adresses des «Agent-Ressource» fournissant la tâche voulue, et il lui demande de commencer l'étape de négociation potentielle sur eux,
- e) l'«Agent-Produit» demande à chaque «Agent-Ressource» de lui fournir une offre pour sa prochaine tâche à effectuer.
- f) Les «Agent-Ressource» calculent les valeurs de ses offres et les transmettent à l'APZ,
- g) L'APZ réserve potentiellement l'«Agent-Ressource» proposant la meilleure offre et signale ça à l'ASD,
- h) L'ASD demande aux agents plus prioritaires que l'APZ de commencer leur processus de réservation potentiel en répétant les étapes (d)...(g) l'un à la suite de l'autre par ordre décroissant de priorité tant que les contraintes de concession de l'APZ sont respectées. Le processus est effectué de la manière suivante :
  - 1. l'«Agent-Produit» plus prioritaire que l'APZ demande une concession de choix à ce dernier pour la réservation du même «Agent-Ressource»,
  - 2. L'APZ vérifie que sa contrainte est respectée pour les autres offres données par des «Agent-Ressource» de cette manière :
    - Si elle est respectée, l'APZ fait une concession en réservant potentiellement un autre «Agent-Ressource» (voir figure 4.16),
    - Sinon il refuse la demande de concession et signale ça à l'ASD pour arrêter le processus (voir figure 4.17).
- i) Une fois que le processus de réservation potentiel est arrêté, l'ASD demande à l'APZ d'effectuer une réservation réelle sur l'«Agent-Ressource» choisie. Le processus peut être arrêté pour une violation de contrainte ou pour l'achèvement de réservations potentielles effectuées par les «Agent-Produit»,
- j) L'APZ réserve réellement l'«Agent-Ressource» choisie, signale la prise de sa décision à l'ASD et quitte la session,
- k) L'ASD efface l'APZ de la «Session de Demande» et la réouvre pour des nouvelles inscriptions, de plus il demande aux «Agent-Ressource» d'effacer toutes les réservations effectuées durant le processus de réservation potentiel.

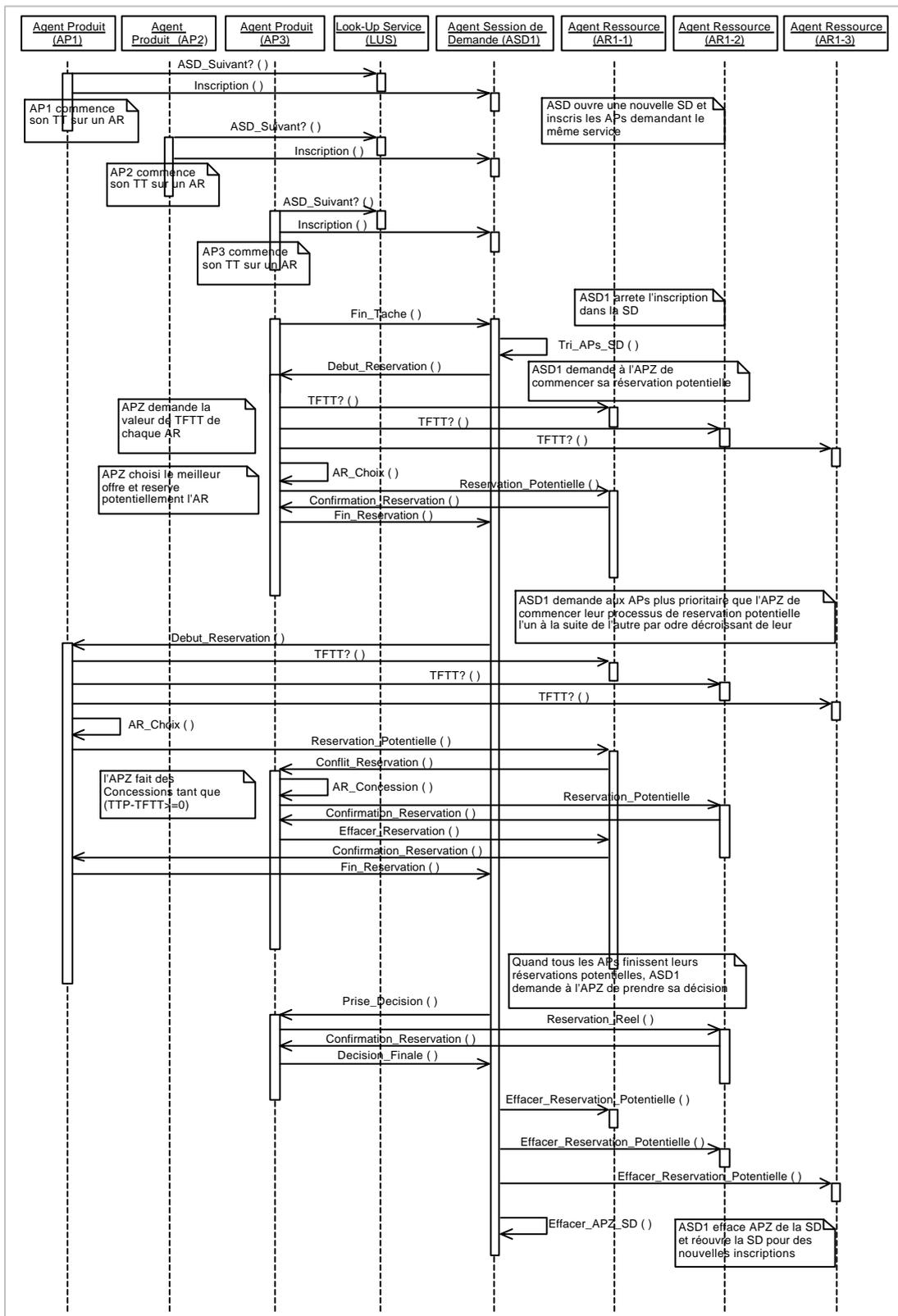
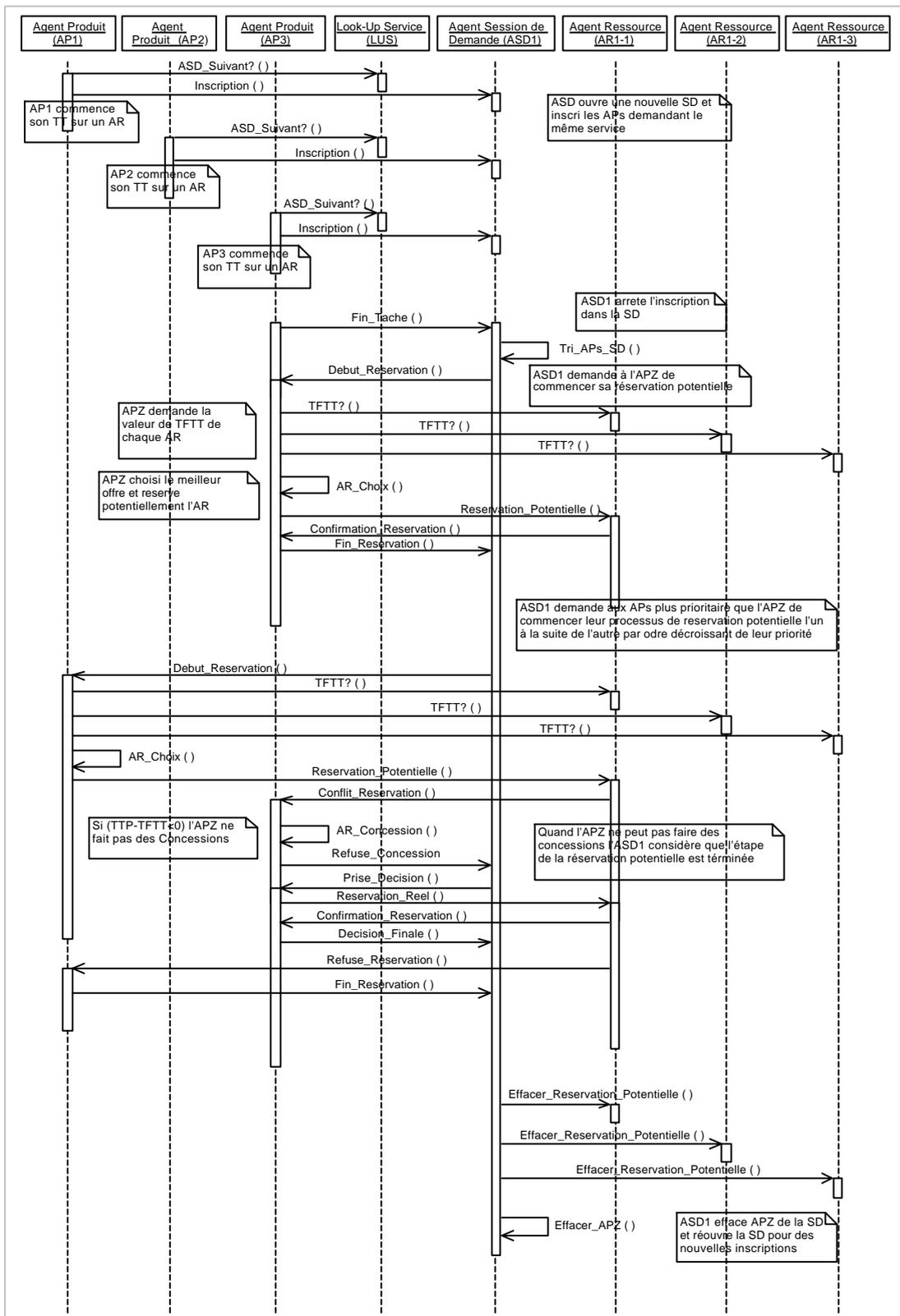


Figure 4.16 : Prise de décision consensuelle d'un APZ respectant la contrainte de temps de traitement permis



**Figure 4.17 : Prise de décision consensuelle d'un APZ ne respectant pas la contrainte de temps de traitement permis**

## **9.6 Protocole « Coopétition » locale**

Dans ce modèle la « Coopétition » est utilisée à travers l'attribution des intérêts globaux sous forme de « Coopération » et l'attribution des intérêts locaux, sous forme de « Compétition », aux agents dans le système. La coopération sera prise en compte au niveau de l'anticipation à une « Session de Demande » chaque fois qu'ils essayent d'allouer leurs prochaines tâches. Cette anticipation permet de mettre en commun leurs propres intérêts, contraintes, et satisfactions afin d'organiser la décision entre eux. Tandis que la compétition sera considérée au niveau de la prise de décision des « Agent-Produit » pour l'allocation des ressources chaque fois qu'ils veulent quitter leurs « Session de Demande » pour commencer l'exécution de leurs nouvelles tâches.

Les agents dans le système utilisent la méthodologie de négociation pour la gestion de la production développée dans le paragraphe 4.6.

Grâce à l'anticipation à une session de demande, le risque de la décision arbitraire d'un « Agent-Produit », à n'importe quel moment dans un environnement dynamique, diminue d'une façon significative. De plus, le partage des informations entre agents dans la session permet de construire un profil réel pour chaque participant. La prise de la décision de l'APZ, se base en grande partie sur ce profil réel construit dans la session. Cette sûreté de décision par l'APZ peut être dérangée par d'autres types de perturbations et imprévus dans le système. A cause de la variété de temps de transport entre les ressources, due à la dynamique du système, à la structure physique de système et au « Dynamic queuing », l'ordre de l'arrivée de l'APZ dans les files d'attente des machines n'est pas connu à l'avance. Un autre facteur important est la notion des propres intérêts à respecter par chaque « Agent-Produit » dans le système qui résume son but d'existence initiale et finale.

La prise de décision de l'APZ doit prendre en considération l'influence et le risque des différents facteurs cités ci-dessus. Pour cela, l'APZ demande à l'Agent de la « Session de Demande » de lui communiquer les informations concernant les temps de traitement permis des autres agents dans la session avec les plus petits temps de traitement restant pour leurs tâches actuelles (Le nombre de ces agents sera réduit à trois dans notre modèle expérimental pour limiter l'influence de toutes les agents dans la session au voisin proche de l'APZ). Finalement, l'APZ recourt aux principes de la théorie des jeux pour obtenir une décision bénéfique et sûre pour faire face à n'importe quelle influence ou perturbation dans le système.

La compétition entre les « Agent-Produit » dans la « Session de Demande » lors de la prise de la décision peut être modélisée sous forme d'un « jeu en forme normale » de la théorie des jeux. Chaque jeu est interprété comme une partie d'une série de jeux non séparables gardant un certain lien ou une continuité entre eux du début jusqu'à la fin de la production. Ceci reflète le fonctionnement réel et dynamique du système et crée une relation « locale—globale » entre les intérêts des agents dans le système de production décentralisé. Ce phénomène est concrétisé dans

ce modèle grâce à l'association du principe de la «Session de Demande» à la théorie des jeux. En fait, la session permet d'établir la continuité et le lien entre les intérêts des «Agent-Produit» participant à la même session et la théorie des jeux permet de modéliser la compétition entre les ces derniers sous forme d'un jeu.

Le jeu est décrit par les éléments suivants :

1. un ensemble de N joueurs  $I = \{1, 2, \dots, N\}$ ,
2. pour chaque joueur  $i, i \in I$ , un ensemble de stratégie  $S^i$ , qui contient toutes les stratégies possibles de ce joueur.  $s^i \in S^i$  est une stratégie particulière du joueur i. Par conséquent,  $S^i = \{s_1^i, s_2^i, \dots, s_{k^i}^i\}$  si  $k^i$  stratégies sont disponibles pour le joueur i. Si chaque joueur i choisit une stratégie  $s^i$ , nous pouvons représenter le profil de stratégies du jeu de tous les joueurs par un vecteur qui contient toutes ces stratégies  $s \equiv (s^1, s^2, \dots, s^N)$ .
3. Pour chaque joueur i, une fonction de gain,  $u^i$ , qui donne la valeur pour le joueur i de chaque résultat de jeu  $u^i(s)$ .  $u^i$ , est une fonction de l'ensemble de vecteur de stratégie  $\prod_{i \in I} S^i$  dans  $\mathfrak{R}$  représenté de la manière suivante :

$$u^i: S = \prod_{i \in I} S^i \rightarrow \mathfrak{R}$$

$$s \equiv (s^1, s^2, \dots, s^N) \mapsto u^i(s)$$

L'objectif de chaque joueur est de minimiser sa perte ou de maximiser son gain. Ce protocole de négociation est modélisé comme un jeu à information complète et imparfaite. Les joueurs connaissent, en effet, l'ensemble des choix possibles et leurs conséquences, mais ne connaissent pas le comportement immédiat des autres joueurs (problème de l'asymétrie dans les approches économiques).

Dans ce modèle nous considérons les «Agent-Produit» comme des joueurs et le choix des «Agent-Ressource» ou le Temps de Fin de Traitement de Tâche (TFTT) comme des stratégies utilisées par les joueurs durant le jeu. Le comportement des interactions des «Agent-Produit» dans le jeu peut être expliqué en termes d'Equilibre de Nash (EN) et la performance globale du système se corrèle d'une certaine façon avec les profits de cet équilibre. Le modèle coordonne sur l'Optimum de Pareto (OP) dans un scénario d'un « jeu continu » fourni par la « Session de Demande ».

Le but du jeu sera alors de choisir le meilleur vecteur de stratégies qui pénalise le moins possible l'ensemble des «Agent-Produit» dans le jeu. Pour cela nous procédons de la manière suivante :

- 1- nous cherchons le /ou les Equilibres de Nash dans le jeu,
- 2- à partir de l'Equilibre de Nash nous calculons l'Optimum de Pareto du jeu.
- 3- Nous attribuons à APZ la stratégie correspondant à l'Optimum de Pareto calculé.

Notre démarche peut être expliquée à travers l'exemple suivant. Nous considérons trois «Agent-Produit» en compétition dans la « Session de Demande » pour la même prochaine tâche, représentée par l'ensemble  $P = \{P1, P2, P3\}$ . Ces agents ont respectivement les Temps de traitement permis  $Tp = \{25, 35, 45\}$  selon une Unité de Temps (UT) et les Temps restants pour l'exécution de leurs tâches en cours  $Tr = \{0, 2, 5\}$  (UT). Les stratégies de choix des «Agent-Ressource» utilisées dans le jeu sont représentés par l'ensemble  $R = \{R1, R2, R3\}$ , ayant respectivement les Temps de Fin de Traitement de Tâche ou les Durées de traitement suivantes :  $Dt = \{30, 35, 40\}$  (UT). Ces valeurs correspondent à une seule offre proposée par chaque «Agent-Ressource» fournissant la tâche voulue. L'«Agent-Ressource» peut proposer plusieurs « Offres » aux «Agent-Produit» dans la session dépendant du Temps de Traitement restant de la tâche en cours ( $Ttr$ ), l'Ordre d'arrivée dans la file d'attente ( $Ordre$ ) et le Temps Traitement de la tâche ( $Tt$ ). Les valeurs de ces offres sont calculées de la manière suivante :

$$Offre_{ij} = Ttr + Ordre_i * Tt_j \quad (4.10)$$

Où  $i$  représente le numéro d'offre correspondant à l'ordre d'arrivé des «Agent-Produit» en compétition dans la file d'attente et  $j$  est le numéro de l'«Agent-Ressource».

Le Retard attendu ( $Ra$ ) pour un «Agent-Produit» est calculé à travers cette formule :

$$Ra_{kij} = Tp_k - Offre_{ij} \quad (4.11)$$

Où  $k$  représente le numéro de l'«Agent-Produit».

Le tableau des valeurs des  $Ra_{ij}$  pour les «Agent-Produit» est le suivant :

	P3		
P1,P2	M1	M2	M3
M1,M1	(-5,-25,-45)	(-5,-25,10)	(-5,-25,5)
M1,M2	(-5,0,-15)	(-5,0,-25)	<b>(-5,0,5)</b>
M1,M3	(-5,-5,-15)	<b>(-5,-5,10)</b>	(-5,-5,-35)
M2,M2	(-10,-35,15)	(-10,-35,-60)	(-10,-35,5)
M2,M1	(-10,5,-15)	(-10,5,-25)	<b>(-10,5,5)</b>
M2,M3	<b>(-10,-5,15)</b>	(-10,-5,-25)	(-10,-5,-35)
M3,M3	(-15,-45,15)	(-15,-45,10)	(-15,-45,-75)
M3,M1	(-15,5,-15)	<b>(-15,5,10)</b>	(-15,5,-35)
M3,M2	<b>(-15,0,15)</b>	(-15,0,-25)	(-15,0,-15)

**Tableau 4.4 : Représentation de la matrice des retard attendu par les «Agent-Produit» pour les vecteurs de stratégies choisis durant le jeu.**

Chaque cellule détaillée dans le tableau 4.4 montre les retards attendus pour les «Agent-Produit» une fois que la combinaison de ces différentes stratégies est jouée. Les colonnes représentent les stratégies utilisées respectivement par les joueurs P1 et P2 et les lignes correspondent aux stratégies de P3. Le vecteur (R1,R1,R1) signifie que les trois «Agent-Produit» P1, P2 et P3 ont choisi la même R1. Ce même vecteur (-5,-25,-45) a comme valeur de retard attendu pour les «Agent-Produit» correspondant à l'ordre d'arrivé estimé dans la file d'attente de R1 suivant leur Temps restant (Tr) pour l'exécution de leurs tâches en cours.

Le profit (P) attribué à un «Agent-Produit» selon son Retard attendu (Ra) pour une stratégie choisie est calculé de la manière suivante :

$$P = \begin{cases} Ra * M & \text{Si } Ra < 0 \\ Ra & \text{Sinon} \end{cases} \quad (4.12)$$

M est une constante choisie pour pénaliser fortement les retards négatifs, avec une grande valeur dépassant la somme des valeurs maximales des retards non négatifs (M=1000 dans notre modèle).

L'ensemble de L'Equilibre de Nash (EN) du jeu  $E$ , correspond à toutes les permutations possibles des N meilleurs offres donnés par les «Agent-Ressource». Il est représenté par les six vecteurs en gras dans le tableau 4.4 (N est le nombre des joueurs, N = 3 dans cet exemple, et E est représenté par  $3! = 6$  vecteurs).

La fonction d'évaluation  $F$ , pour un vecteur  $EN_i = (P_{i1}, P_{i2}, \dots, P_{iN})$  de  $E$ , est calculée de la manière suivante :

$$F (EN_i) = \sum_{j=1}^N P_{ij} \quad (4.13)$$

Où  $P_{ij}$  correspond aux profits des joueurs,  $i$  l'indice des vecteurs dans  $E$ , et N est le nombre des joueurs.

L'Optimum de Pareto de l'Equilibre de Nash considéré est un vecteur EN ayant la valeur maximale de la fonction dévaluation F. Il est représenté par le vecteur (-5,0,5) dans le tableau 4.4.

Ce modèle est utilisé pour la prise de décision de l'APZ dans la «Session de Demande». Dans ce modèle, le Temps de traitement restant ( $Tr$ ) pour un «Agent-Produit»,  $Tr < \ddot{e}$  est le critère de sélection utilisé pour le choix des agents en compétition avec l'APZ dans la session,  $\ddot{e}$  est une constante et le nombre de produits vérifiant ce critère est limité à trois. La stratégie utilisée par L'APZ correspond à l'Optimum de Pareto de l'Equilibre de Nash dans le jeu effectué avec les autres «Agents-Produits» dans la session. Ce protocole de négociation est détaillé à travers l'algorithme 4.6 et diagramme de séquence dans la figure 4.18.

#### **Algorithme 4.6 : « Coopétition » locale**

- a) L'APZ signale la fin de son traitement actuel à l'ASD,
- b) L'ASD arrête l'inscription dans la « Session de Demande » (SD),
- c) L'ASD trie les «Agent-Produit» par leurs temps de traitement permis et sélectionne les agents selon le critère de voisinage  $\bar{e}$  utilisé dans la session,
- d) L'ASD communique à l'APZ le nombre des agents sélectionnés avec leurs temps de traitement permis et les adresses des «Agent-Ressource» fournissant la tâche voulue,
- e) l'APZ demande à chaque «Agent-Ressource» de lui fournir un certain nombre d'offres correspondant à leurs durée de traitement et aux nombre des agents choisi par l'ASD et qui sont en compétition avec lui,
- f) Les « Agent-Ressource » calculent les valeurs de leurs offres et les transmettent à l'APZ,
- g) L'APZ simule le déroulement de la compétition de la manière suivante :
  - il cherche les vecteurs correspondant à l'Equilibre de Nash (EN) dans le jeu,
  - il calcule l'Optimum de Pareto (OP) de ces vecteurs,
  - il choisit l'« Agent-Ressource » dans le vecteur correspondant à l'Optimum de Pareto calculé.
- h) L'APZ alloue l'« Agent-Ressource » choisi, signale la prise de sa décision à l'ASD et quitte la « Session de Demande ».

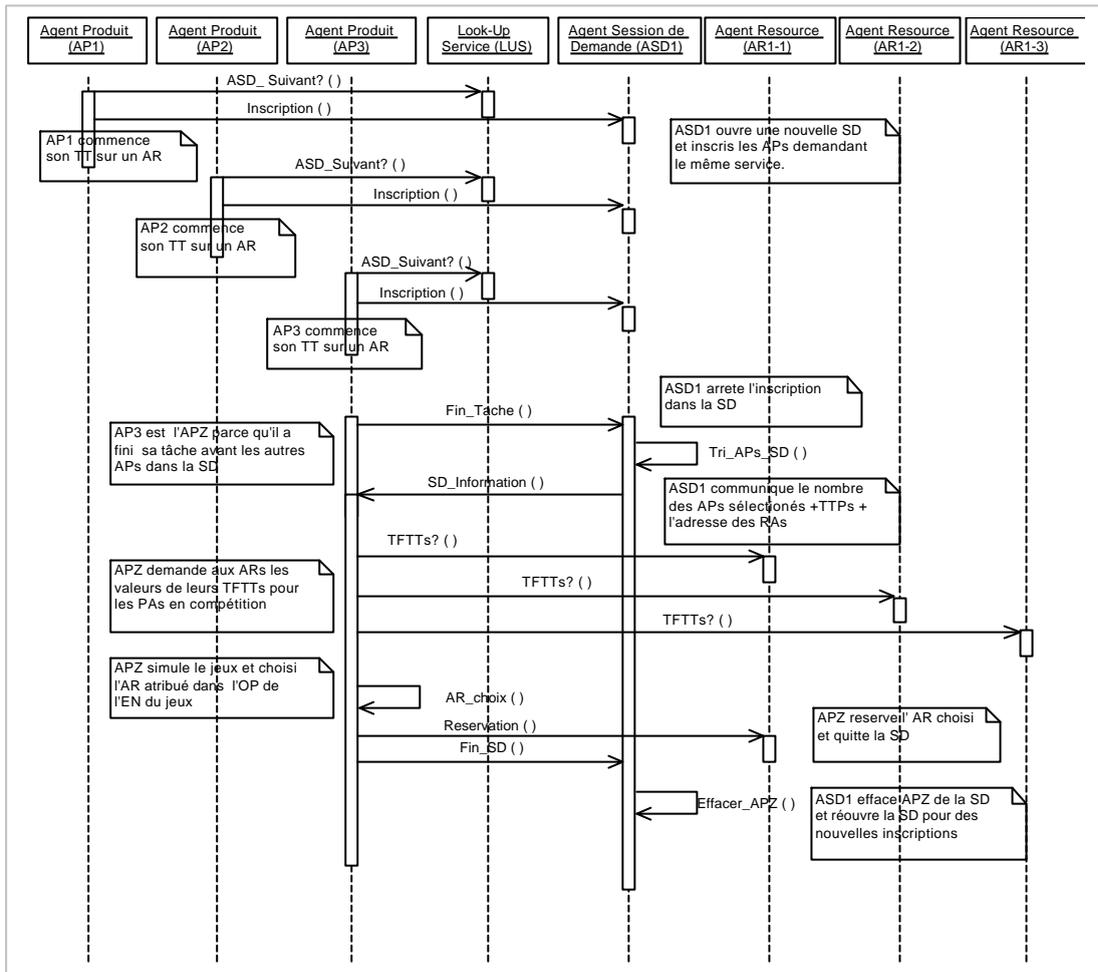
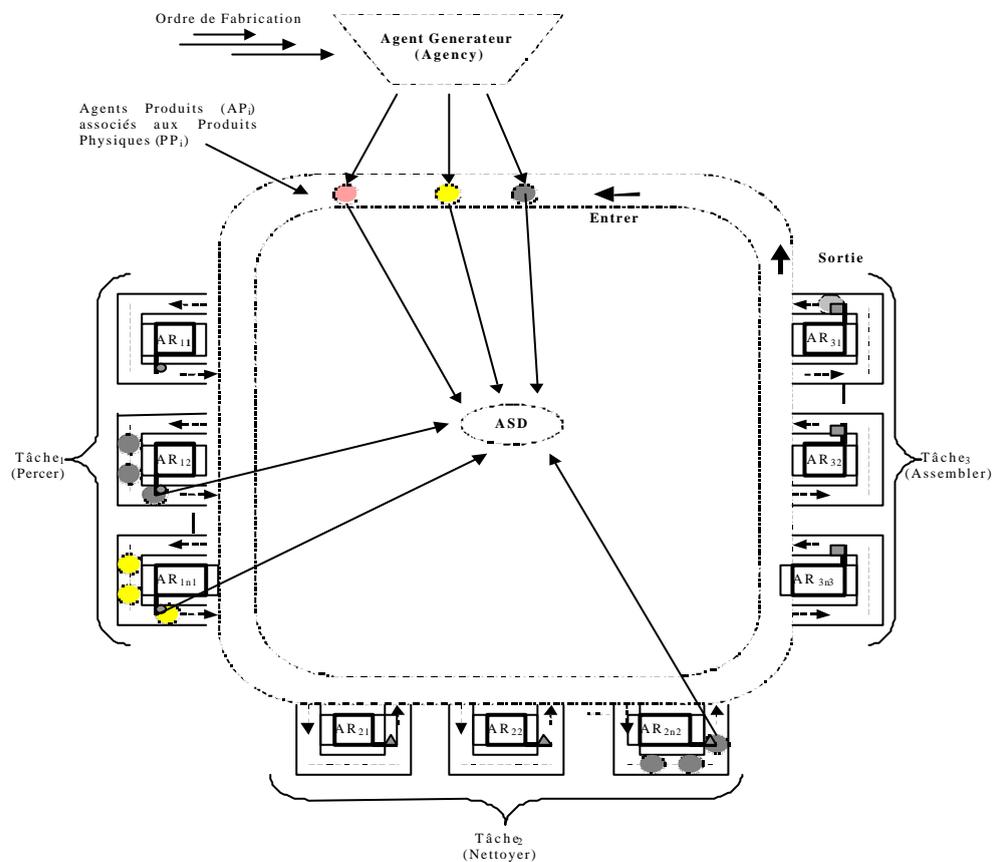


Figure 4.18 : Prise de décision d'un APZ dans la Session de Demande

## 9.7 Protocole « Coopétition globale ou centralisée »

Ce protocole de négociation utilise une approche centralisée pour la prise de décision dans un environnement dynamique afin d'allouer les ressources. Dans ce modèle nous considérons une seule « Session de Demande » et un seul Agent de Session de Demande (ASD) pour tous les « Agent-Produit » (AP) et pour toutes les tâches fournies par les « Agent Ressource » (AR). A cause de la dynamique du système, les « Agent-Produit » participant à la « Session de Demande » sont ceux qui sont seulement en cours d'exécution sur les « Agent-Ressource » et non dans des files d'attente. Ainsi, la prise de décision d'un agent dans la session englobe toutes ses tâches restantes à effectuer (voir figure 4.19).

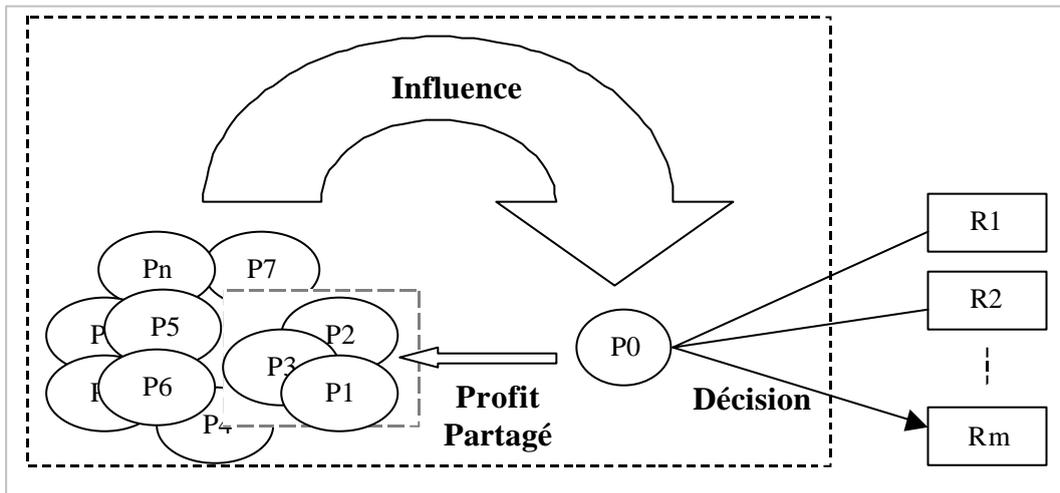


**Figure 4.19 : Approche centralisée pour la prise de décision des agents dans un système de production**

Comme dans le modèle précédent, la « Session de Demande » est associée à la théorie des jeux pour modéliser le système comme un jeu de « coopétition » entre les agents. Dans ce protocole, le choix de l'APZ dans la session est attribué par l'ASD et correspond à la stratégie utilisée dans le vecteur correspondant à l'Optimum de Pareto de l'Equilibre de Nash du jeu. Dans ce protocole de négociation, le nombre des « Agent-Produit » participant à la « Session de demande » est élevé à cause de

l'utilisation d'une seule session dans le système. De même, le nombre d'offres proposé par les «Agent-Ressource» est élevé et concerne toutes les tâches fournies dans le système. Le nombre de «Agent-Produit» en compétition sera limité à un certain voisinage  $\ddot{e}$  correspondant à leur intervalle de Temps de traitement restant ( $Tr$ ). Ce choix est fait pour l'amélioration des performances du modèle et pour des besoins de calcul et de réponse en temps réel.

En effet, dans un environnement dynamique à forte perturbation, l'influence exercée par une grande population de produits, dépendants les uns des autres, sur la prise de décision d'un produit, est grande. Le profit souhaité partagé par la population d'influence sera perdu avec le nouveau changement introduit dans le système. La population de produits perd alors le partage de profit attendu, à travers son influence, et le produit P0 qui a pris sa décision a subi une énorme influence en vain (voir figure 4.20). Pour cette raison, la meilleure solution dans une telle situation est de sélectionner une sous population de voisins du produit P0 influençant directement sa décision.



**Figure 4.20 : Influence et profit d'une population d'individu sur la prise de décision d'un individu.**

Dans la figure 4.20, l'ensemble  $\{P1, P2, \dots, Pn\}$  correspond aux «Agent-Produit» dans la «Session de Demande» et l'ensemble  $\{R1, R2, \dots, Rm\}$  correspond aux «Agent-Ressource» fournissant les services dans le système. P0 correspond à l'APZ au moment de sa prise de décision.

Trouver les critères de choix des voisins de P0 dans la population est une tâche difficile, surtout quand elle est utilisée pour la prise de décision dans un environnement dynamique. Le principe de la «Session de Demande» peut faciliter cette tâche en nous fournissant une adaptation dynamique entre les interactions des produits dans la session. En plus, grâce aux liens continus entre les séries de jeux durant la session, un produit peut participer à plusieurs jeux durant son exécution et il

peut être exclu de certains jeux. Son influence dans la population varie selon ses caractéristiques et l'état des autres produits dans la session.

Les critères de sélection de la sous population des « Agent-Produit » utilisés sont le Temps de traitement restant ( $Tr$ ) de la tâche en cours et le Temps de traitement permis ( $Tp$ ) pour l'exécution de la prochaine tâche. Dans ce modèle, le  $Tr < \ddot{e}$ ' est le critère utilisé pour le choix de la sous population des « Agent-Produit »,  $\ddot{e}$ ' est une constante et le nombre de produits vérifiant ce critère est limité à cinq pour des raisons de calcul en temps réel.

Soit un vecteur d'offre  $V = (Dt_1, Dt_2, \dots, Dt_N)$  correspondant aux Durées de traitement ( $Dt_i$ ) des « Agent-Ressource », pour les  $N$  tâches restant à exécuter par un « Agent-Produit ». Le retard attendu pour cet agent est égal à :

$$Ra = Tdd - \sum_{i=1}^N Dt_i, \quad (4.14)$$

Où  $Tdd$  est le temps restant avant la Due Date de l'« Agent-Produit ».

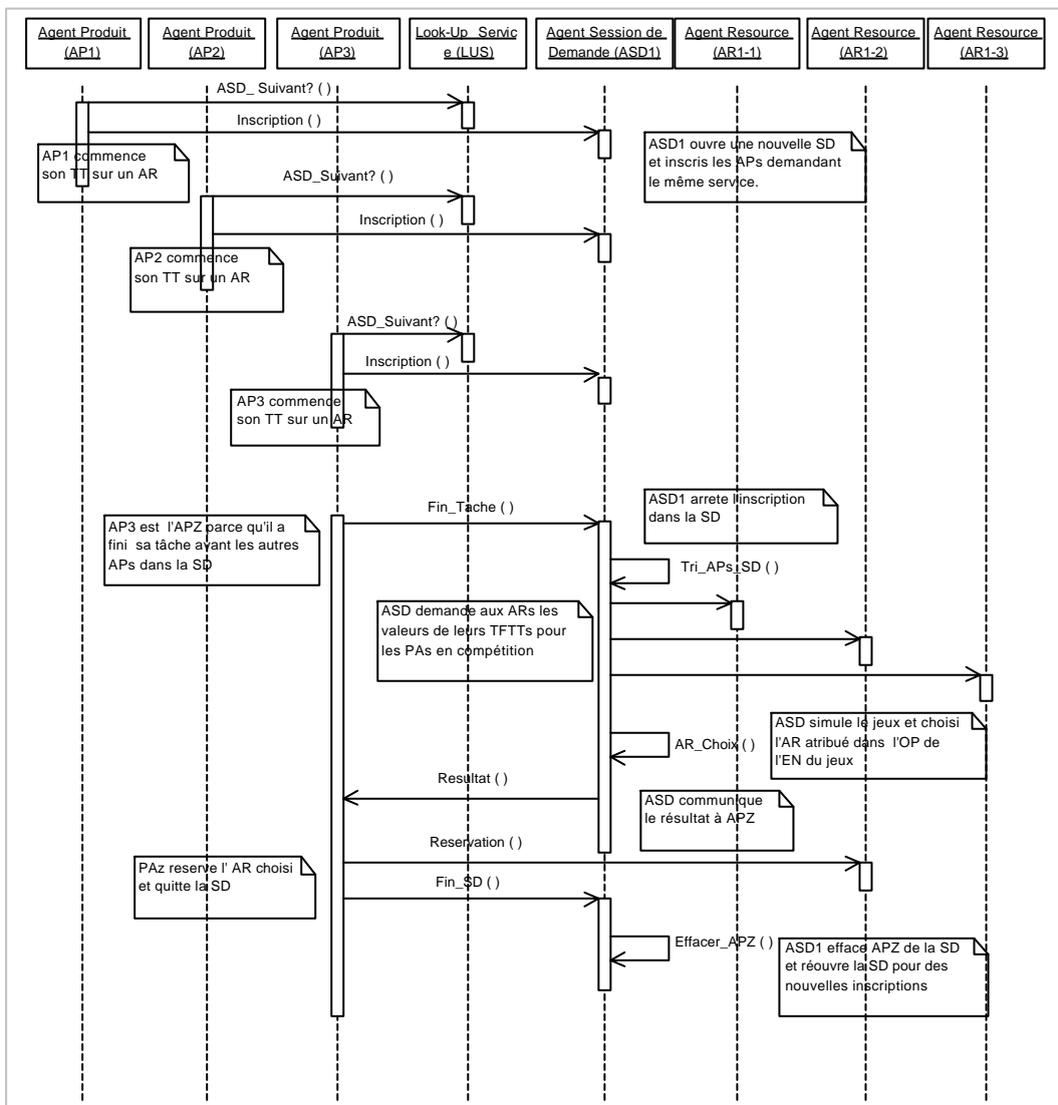
Comme dans le modèle précédent, l'ensemble de L'Equilibre de Nash (EN) du jeu  $E$ , correspond à toutes les permutations possibles des  $N$  meilleurs offres donnés par les « Agent-Ressource ». Le profit attribué à un « Agent-Produit » pour un vecteur  $V$  est donné par la formule 4.12. La fonction d'évaluation  $F$ , pour un vecteur  $EN_i = (P_{i1}, P_{i2}, \dots, P_{iN})$  de  $E$ , est calculée par la formule 4.13. Finalement, l'Optimum de Pareto de l'Equilibre de Nash est un vecteur EN ayant la valeur maximale de la fonction dévaluation  $F$ .

La prise de décision d'un APZ par l'intermédiaire d'une approche centralisée utilisée par l'agent de la « Session de Demande » (ASD) est détaillée à travers les l'algorithme 4.7 et le diagramme de séquence UML dans la figure 4.21.

#### **Algorithme 4.7 : « Coopétition » globale**

- a) L'APZ signale la fin de son traitement actuel à l'ASD,
- b) L'ASD arrête l'inscription dans la session,
- c) L'ASD trie les « Agent-Produit » par leurs temps de traitement permis et sélectionne les agent selon le critère de voisinage  $\ddot{e}$ ' utilisé dans la session,
- d) il demande à chaque « Agent-Ressource » de lui fournir un certain nombre d'offres correspondant à leurs durée de traitement et aux nombre des agents choisi par l'ASD et qui sont en compétition avec lui,
- e) Les « Agent-Ressource » calculent les valeurs de leurs offres et les transmettent à l'ASD,

- f) L'ASD simule le déroulement de la compétition ou le jeu avec les «Agent-Produit» choisis de la manière suivante :
- il cherche les vecteurs correspondant à l'Equilibre de Nash (EN) dans le jeu,
  - il calcule l'Optimum de Pareto (OP) de ces vecteurs,
  - il choisit sa stratégie ou l'« Agent-Ressource » dans le vecteur correspondant à l'Optimum de Pareto calculé.
- g) L'ASD communique ce choix à l'APZ,
- h) L'APZ alloue l'« Agent-Ressource » choisi, signale la prise de sa décision à l'ASD et quitte la « Session de Demande ».



**Figure 4.21 : Prise de décision d'un APZ par l'intermédiaire d'une approche centralisée**

## **9.8 Conclusion**

Dans ce chapitre, nous avons présenté une analyse et une critique de certains concepts existants dans la littérature des systèmes de production décentralisés. L'adaptation et la stabilité des performances de ces concepts dans un environnement dynamique en est généralement le point faible. Notre approche a été développée pour combler en partie les lacunes constatées au moyen de nouveaux principes et mécanismes.

Le concept de la Session de Demande proposé dans ce chapitre nous a permis de lier les différentes interactions et influences locales en temps réel avec les entités dans le futur, le présent et le passé. Les agents, dans la session, profitent des informations fournies en temps réel par l'Agent de la Session de Demande pour leur prise de décision. Ce dernier joue le rôle d'un agent coordinateur entre les agents interagissant localement au cas de conflit ou d'un changement global dans le système.

Quatre nouveaux protocoles ont été détaillés intégrant des notions de concession et de « coopération », pour la prise de décision, à travers de plusieurs protocoles de négociation et le principe de la théorie des jeux.

La théorie des jeux a intégré deux des nouveaux protocoles conçus pour la prise de décision localement et globalement.

Finalement, les protocoles de négociation utilisés dans la thèse seront testés, validés et comparés sur un démonstrateur expérimental, et leurs performances évaluées, dans le chapitre suivant.

# Chapitre 5 : Application et Résultats

## 1 Introduction

Cette thèse présente différents concepts existants dans les systèmes de production décentralisés pour la gestion de la production. Compte tenu de l'étude critique qui suit, de nouveaux concepts, approches et mécanismes ont été développés pour améliorer les performances des systèmes de production dans un environnement dynamique situé.

Ce chapitre est consacré à la description de la structure des démonstrateurs expérimentaux mise en œuvre physiquement (en cours d'adaptation avec l'architecture de PABADIS), et théoriquement. Nous décrivons également la conception des fichiers expérimentaux (ou les benchmarks) développés, le prototype de test et les résultats obtenus.

L'architecture, les concepts et les algorithmes ont été validés grâce à des simulations sur un démonstrateur expérimental théorique. Un prototype de test a été conçu pour étendre la capacité du démonstrateur physique du LGI2P-EMA.

Ce Chapitre est organisé de la manière suivante : en première partie nous considérons le démonstrateur physique et théorique utilisé, ensuite nous développons différents benchmarks pour le test et la validation des concepts. Dans la seconde partie nous détaillons le prototype élaboré. Dans la troisième partie nous détaillons les résultats obtenus, suivis par une comparaison des performances des différents modèles. Finalement, nous concluons par quelques synthèses et de perspectives.

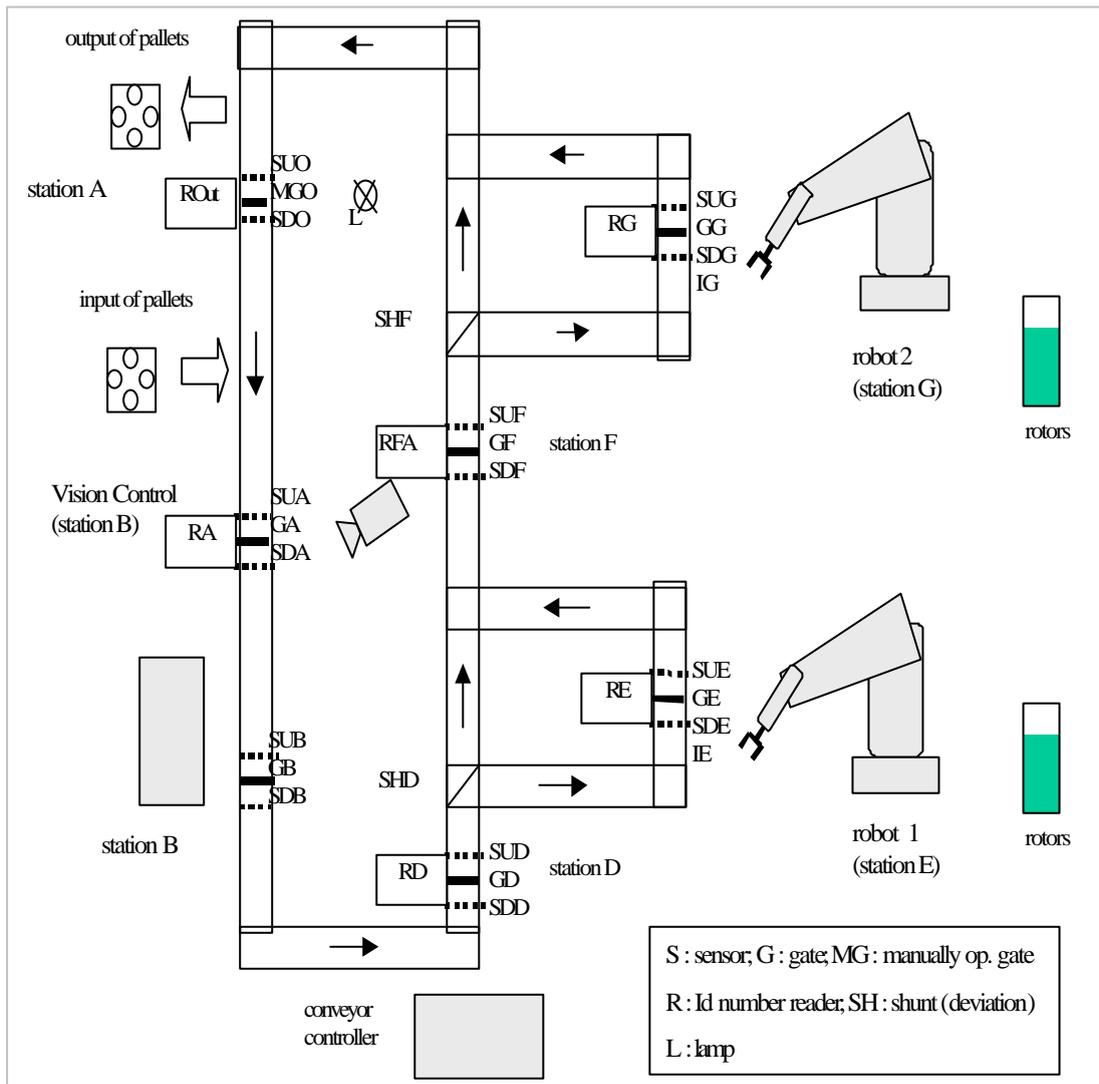
## 2 Conception du démonstrateur expérimental

Nous avons développé un démonstrateur expérimental théorique pour simuler, tester, évaluer et valider les performances des concepts développés dans notre travail de recherche.

Ce démonstrateur étend au niveau théorique et expérimental les approches de gestion de production mises en œuvre sur le démonstrateur expérimentale réel du centre de recherche LGI2P-EMA (voir figure 5.1).

## 2.1 Description du démonstrateur

Cet outil a été choisi par le projet PABADIS, avec d'autres démonstrateurs européens, pour tester et valider en mode réel les concepts définis pour ce projet. Dans ce démonstrateur, la production consiste à assembler 2 éléments personnalisés de moteurs électriques : stators et rotors. Des stators sont installés aux positions fixes sur les palettes mobiles et chaque palette peut porter jusqu'à 4 stators.



**Figure 5.1 : Structure physique du démonstrateur du Laboratoire LGI2P-EMA**

Dans la figure 5.1, nous distinguons les ressources suivantes : « Vision Control » pour détecter le nombre et les positions de stators sur une palette, « convoyeur » pour contrôler le cheminement des palettes sur les différentes stations, et les deux Robots « Robot1 et Robot2 » pour assembler sur une palette deux parties de moteurs électriques : stators et rotors.

Les opérations de production sur le démonstrateur sont les suivantes :

- les palettes sont présentées par un opérateur à la station A. Le nombre de stators peut changer et varier selon les demandes.
- sur la station B, les palettes sont inspectées par une caméra. Le nombre et les positions de stators sont détectés. Chaque palette peut être identifiée par un nombre « code barre » qui est stocké dans une mémoire magnétique embarquée par la palette.
- la station C est une station intermédiaire. Elle permet d'avoir la file d'attente de palettes sur le convoyeur. Entre les stations B et C, 4 palettes peuvent attendre.
- sur la station D, les palettes peuvent continuer leur trajet sur la bande du convoyeur principal ou prendre la dérivation et aller au robot 1.
- la station E est un lieu d'assemblage pour le robot 1. La palette est détectée grâce à un indexeur. Des rotors sont pris par le robot d'une pile de stock, et insérés dans des stators. Lors de l'assemblage sur une palette, les autres palettes attendent derrière elle sur la même dérivation.
- les stations F et G sont identiques aux stations D et E. Finalement, la station A sert également à l'extraction des palettes. Une lampe est utilisée pour signaler à l'opérateur que l'assemblage sur une palette est complet, ou qu'il faut recommencer un autre cycle.

L'adaptation du modèle centralisé de notre démonstrateur au modèle proposé par PABADIS est en cours de développement. Le modèle actuel du démonstrateur nous a posé des contraintes de **temps** (l'adaptation est en cours de développement) et des **limitations** de matérielles (nombre de robots = 2, nombre de tâche = 1, etc.) utilisés. Pour cela, nous avons développé un simulateur pour la **validation** théorique des concepts utilisés dans la thèse. L'implémentation de ces concepts sur le démonstrateur réel sera appliquée une fois que l'adaptation de ce dernier à l'architecture de PABADIS sera terminée.

## **2.2 Modèle générique**

L'idée de base est de développer un simulateur pour tester, valider et évaluer les performances des concepts développés dans le cadre de la thèse.

Dans ce simulateur nous avons augmenté le nombre des robots et le type des tâches à effectuer d'une façon intuitive tout en adaptant ce démonstrateur à l'architecture proposée dans PABADIS. La configuration de base utilisée comporte (voir figure 5.2) :

- neuf robots repartis en trois groupes selon le type de tâches fournies,
- en dehors de l'Assemblage (A), les robots peuvent effectuer le Perçage (P) et le Nettoyage (N),
- chaque robot ne peut faire qu'une seule tâche,

- les palettes sont traitées selon l'ordre FIFO dans les files d'attente des robots,
- les robots sont représentés par des Agents Ressources (AR),
- les palettes ou produits sont représentés par des Agents Produits (AP),
- sept types de produit sont fabriqués dans ce système. Ils possèdent chacun leur gamme opératoire propre. Chaque type est caractérisé par le nombre, le type et l'ordre de l'exécution de ses tâches, par exemple : Type1 = {P,N,A}, Type2 = {P,N}, Type3 = {P,A}, Type4 = {N,A}, Type5 = {P}, Type6 = {N }, Type7 = {A },
- chaque produit effectue un certain nombre de tâche ordonnées séquentiellement,
- le convoyeur est représenté par un Agent de Transport (AT).

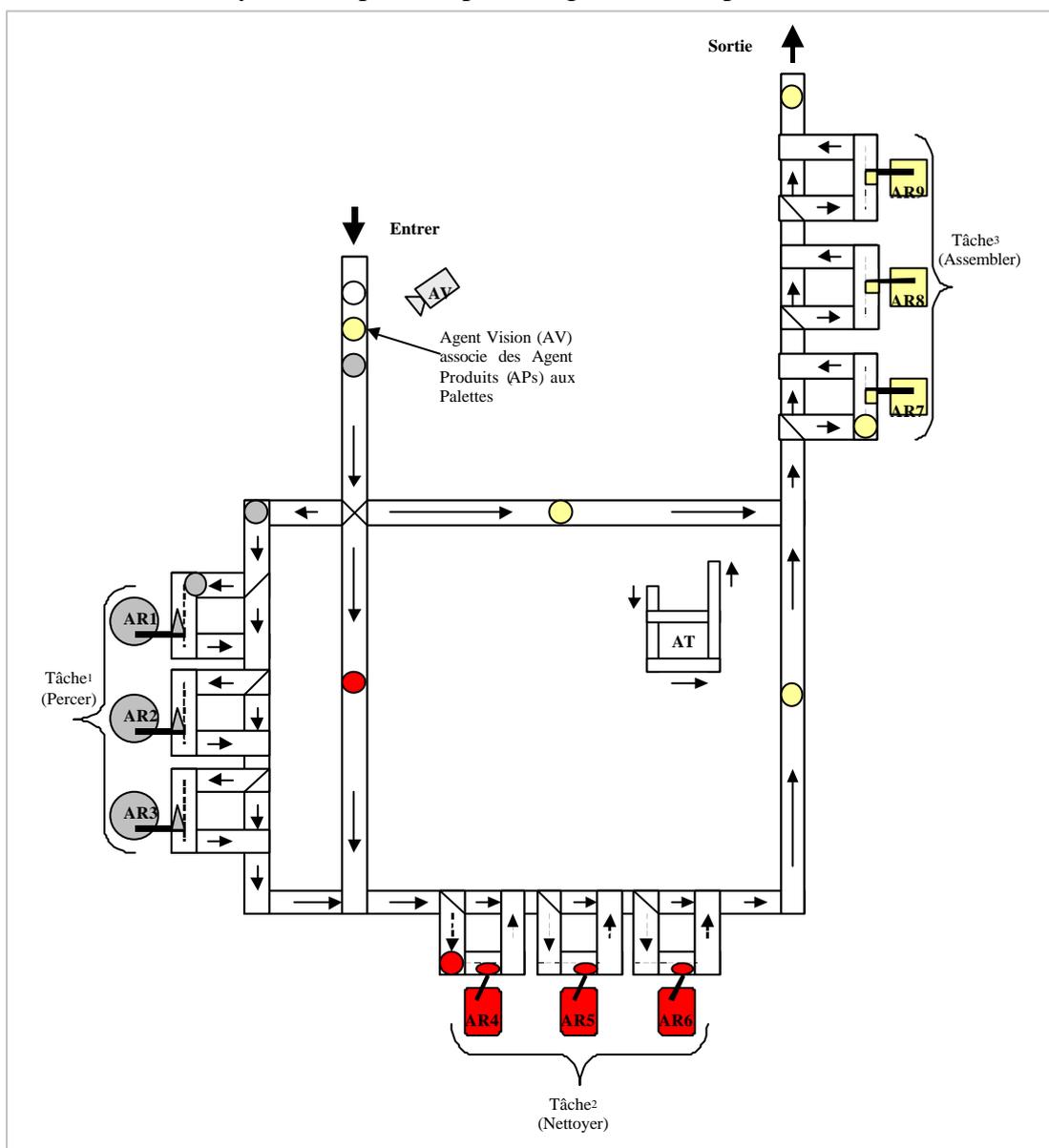


Figure 5.2 : Structure du démonstrateur expérimental théorique.

La limitation du nombre et du type de tâches effectuées par les ressources est établie de telle manière que le fonctionnement du simulateur reste simple et compréhensible (ces paramètres seront étendus dans le futur pour réaliser des tâches plus compliquées (type, nombre, etc.)).

### **2.3 Conception des expériences**

En littérature, le « design » des benchmarks expérimentaux pour les systèmes de production décentralisés sont conçus généralement selon des approches intuitives [Wyns, 1999]. Les résultats expérimentaux spécifiques pour la gestion de la production dans ces systèmes sont peu nombreux. Saad et al., détaillent des résultats expérimentaux pour l'application de leur approche « Production Reservation » pour la génération du plan de la production pour chaque produit dans le système [Saad et al., 1997] . Leur approche consiste à associer à la méthode de réservation PR des règles de dispatching (FIFO, Earliest Due Date, Short Processing Time, etc.) pour sélectionner les produits dans les buffers des machines.

Des résultats expérimentaux ont été calculés et comparés par N. Krothapalli pour les différentes approches développées dans son travail de recherche. Ce dernier a voulu mettre en évidence l'avantage de l'approche « Currency » dans un environnement dynamique pour la satisfaction locale des intérêts des produits et la satisfaction globale du système [Krothapalli et Deshmukh, 1998]. L. Bongaerts et B. I. Kim ont comparé dans leurs travaux de thèse les performances de l'architecture Holonic par rapport à l'architecture hétérarchique et hiérarchique pour l'ordonnancement et le contrôle de la production. Les résultats expérimentaux ont été calculés dans des environnements statiques, dynamiques et chaotiques [Bongaerts, 1999 ; Kim, 2002].

L'environnement choisi pour tester et valider les modèles développés dans la thèse est un environnement dynamique situé. L'architecture du système, les protocoles de négociation entre les entités et le « design » des benchmarks ont comme but majeur de représenter cet environnement. L'architecture de PABADIS nous a permis de représenter la distribution de l'interaction entre les différentes entités d'un système réel de production. Les modèles développés dans le chapitre précédent ont réussi à traduire la notion située, par le biais de la distribution de la décision entre les agents au niveau local dans le système. Il reste à créer la dynamique du système à partir du design des benchmarks et les fichiers d'expérimentations. Ceux-ci sont élaborés d'une façon intuitive tout en respectant la variation de la production des produits (type, quantité, etc.) pour des systèmes de production conçus pour PABADIS.

## 2.4 Plan de test - les « Benchmarks »

La configuration de base des différents benchmarks utilisés dans notre simulateur est composée de la manière suivante :

- quarante produits,
- sept types différents de produits, chaque type est distingué par la gamme opératoire c'est à dire le nombre, le type et l'ordre de l'exécution de ses tâches (voir le paragraphe 5.2.2),
- le pourcentage de produits effectuant un même nombre de tâche est égal à :
  - 52.5 % pour une seule tâche,
  - 30 % pour deux tâches,
  - 17.5 % pour trois tâches.
- la date d'échéance des produits varie entre 30 et 350 Unité de Temps (UT),
- Le temps de traitement pour chaque tâche sur les ressources peut être 30, 35 ou 40 UT,
- la charge de travail peut être basse (-) (égale à 1/8 UT) ou haute (+) (égale à 1/5 UT),
- la date d'échéance courte des produits peut varier entre petite (-) (10 %) et grande (45%),
- pour chaque demande, ou commande, la taille du lot des produits peut varier entre petite (-) [1..2] et grande (+) [4..8].

Chaque fichier d'expérimentation est composé d'une réplication de cinq fois la configuration de base citée dessus. Nous distinguons huit fichiers d'expérimentations dans l'ensemble des benchmarks utilisés. Ces fichiers sont déterminés par toutes les combinaisons (-) et (+) de ces trois paramètres : charge de travail, date d'échéance et taille du lot (voir tableau 5.1).

Cette variation introduite au niveau de ces huit benchmarks a comme but de créer un environnement dynamique évoluant selon ces derniers paramètres. Reste à signaler que cette perturbation est effectuée seulement au niveau des produits et non au niveau des ressources (pour des raisons de simplicité, suffisance, les pannes de ressources seront substituées par des variations au niveau de la date d'échéance des produits introduits dans le système). Finalement, nous considérons que la génération des produits dans le système à partir des benchmarks n'est pas connu à l'avance par les agents.

Les critères de performance de chaque protocole de négociation et pour chaque fichier d'expérimentation sont les suivants : le temps de fabrication, le temps d'attente, le temps de retard, le temps d'utilisation des ressources, la longueur des files d'attente et le nombre de produit en retard dans le système (ou en cours).

	Expériences							
	1	2	3	4	5	6	7	8
<b>Charge de Travail</b>	-	-	+	+	-	-	+	+
<b>Date d'Échéance</b>	-	-	-	-	+	+	+	+
<b>Taille du Lot</b>	-	+	-	+	-	+	-	+

**Tableau 5.1 : Combinaisons des benchmarks utilisés dans le système.**

Dans ce qui suit nous définissons et nous utilisons les huit jeux de tests détaillés dans le tableau 5.1.

### **3 Elaboration du prototype du test**

Cette section détaille essentiellement l'outil de test qui a été conçu et développé en vue de simuler, tester et valider les concepts que nous avons élaborés au cours de ce travail de recherche. Nous décrivons plus particulièrement la plate-forme Grasshopper et la façon dont nous l'avons utilisé.

#### **3.1 Outils et environnement informatiques de base**

La création des benchmarks, la simulation des différents protocoles de négociation et l'obtention des résultats ont été réalisés à l'aide d'un prototype de test, sur la base de la plate-forme Grasshopper 2.2 détaillée ci-dessous, et du langage JAVA 1.3. Les différentes simulations ont été réalisées sur un PC Pentium III de 800 MHZ, avec 327 MB de RAM et utilisant le système d'exploitation Windows NT version 4.0.

### **3.2 Plate-forme d'agent mobile : Grasshopper**

Une évaluation de plusieurs plate-formes d'agents existantes (Voyager, Grasshopper, Aglets, Lana, Odyssey) a été effectuée. Grasshopper a été choisi comme la plate-forme de base pour PABADIS parce qu'elle répond à différents besoins (disponibilité, mobilité, sécurité, communication, etc.) (voir annexe B) [PABADIS, 2002b]. De même, les différents avantages de cette plate-forme nous ont poussé à l'utiliser dans notre travail de recherche pour le développement du prototype de test. Dans les paragraphes qui suivent nous détaillons les principes et les concepts de base de la plate-forme Grasshopper.

#### **3.2.1 Qu'est ce que Grasshopper ?**

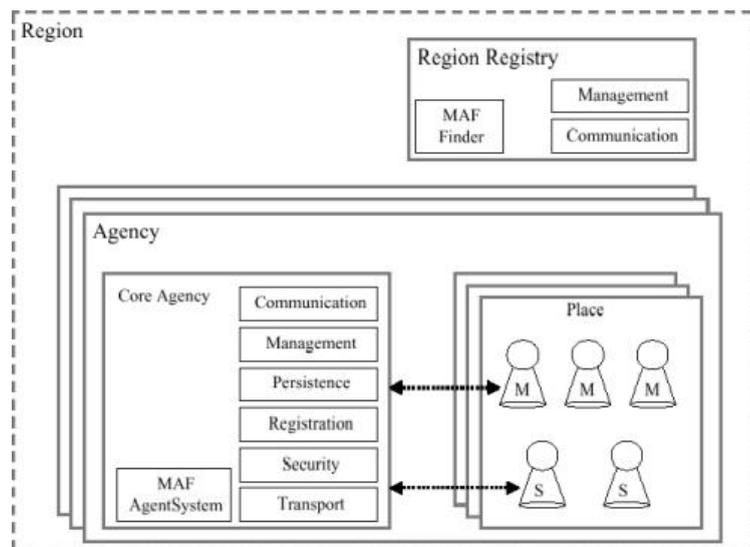
Grasshopper est une plate-forme d'agents mobiles développée par IKV++ (Innovation Know-how Vision ++) Technologies AG. Elle permet la création d'applications multi-agent en fournissant un environnement d'exécution approprié pour eux. Grasshopper a été définie selon le premier agent mobile standard MASIF « Mobile Agent System Interoperability Facility » de l'OMG (Object Management Group). Elle est entièrement construite sur la plate-forme Java 2 qui utilise les protocoles de communication standards et traditionnels comme IIOP, RMI ou Sockets. De même, Grasshopper supporte des mécanismes de sécurité externe, comme le SSL, et externe, fourni par JAVA, pour assurer la confidentialité voulue, l'intégrité, et l'authentification dans les applications. Finalement, Grasshopper est compatible avec des protocoles et des langages de communication standards tels que KQML et FIPA.

#### **3.2.2 Composantes de la plate-forme**

La plate-forme est constituée de plusieurs entités. Nous pouvons distinguer des régions, des agences, des places, des agents, des enregistrements de région « region registries » et des services de communication (voir figure 5.3).

L'utilisation simultanée de toutes ces composantes dans une même application n'est pas obligatoire sauf pour les agents et les agences ; leur existence est nécessaire, et doit être spécifiée, dans chaque application développée avec Grasshopper.

Une région peut être découpée en plusieurs « Agences » ; les agents sont regroupés en « place ». Lorsqu'un nouveau composant (agence, place, agent) est créé, il est automatiquement enregistré dans la région d'enregistrement correspondante. Une agence est l'environnement d'exécution des agents. Une agence est décomposée en deux parties : le noyau (ou core) et les places. Le noyau de l'agence contient les fonctions minimales requises pour l'exécution des agents.



**Figure 5.3 : Structure hiérarchique des composants dans Grasshopper**

Les différents services offerts par la plate-forme sont les suivants :

- la communication : toutes les interactions peuvent être exécutées par l'intermédiaire de CORBA IIOP, Java RMI, Connexions de sockets,
- l'enregistrement : chaque agence doit être capable de savoir tout au sujet des agents hébergés et des places,
- la gestion : Les services de gestion permettent aux utilisateurs humains la surveillance et le contrôle des agents et places,
- la sécurité: La sécurité interne protège les ressources des agences des accès interdits par des agents. La sécurité externe protège des interactions entre les agences et la région d'enregistrement,
- la persistance: Afin de sauvegarder les données en cas de crash du système, le service de persistance permet de sauvegarder l'état interne des agents s'exécutant. Les agents pourront continuer leurs tâches une fois l'agence relancée,

Par exemple, les composants de PABADIS peuvent être représentés par des entités équivalentes en Grasshopper, suivant le tableau 5.2 .

PABADIS components	Grasshopper components
PA	Mobile Agent
RA	Stationary Agent
PMA	Stationary Agent
CMU	Agency
Agent Fabricator	Subsystem made according to PABADIS and Grasshopper specifications.
Agency = CMU + PMA + Agent Fabricator.	

**Tableau 5.2 : Equivalence terminologique entre Grasshopper et PABADIS**

Dans le cadre de notre travail nous garderons une équivalence terminologique similaire.

### 3.2.3 Aperçu technique de Grasshopper

Tout comme la plate-forme, les agents sont développés en Java. Cela permet donc aux utilisateurs de se servir d'agents prédéfinis mais aussi aux programmeurs de créer leurs propres agents. Ces agents sont perçus comme un service car ils offrent diverses fonctionnalités à des tiers, qu'ils soient utilisateurs humains ou entités informatiques. Un agent est donc une classe Java qui hérite d'une super classe Service offrant les services de base de chaque agent. Cependant il existe deux types d'agents, les agents stationnaires, héritant de la classe StationnaryAgent et les agents mobiles héritant de la classe MobileAgent. Ces deux classes, sous classes communes de la classe Service, offrent, elles aussi, différentes fonctionnalités de base. Il suffit alors au programmeur de déterminer quel type d'agent il désire pour créer son agent propre. La différence entre les deux types d'agents réside dans le fait que le premier (Agent Stationnaire) est lié à un endroit spécifique (Agence, Place) et ne peut donc pas migrer de lui-même à travers le réseau, cependant il est toujours possible à un utilisateur de le déplacer explicitement. Le second type d'agent (Agent Mobile), est capable de se déplacer à travers le réseau, de place en place de manière autonome. De la sorte il peut donc facilement bénéficier des avantages des interactions locales. Mais il peut aussi effectuer des communications distantes de type RPC (Remote Procedure Call), selon que telle ou telle option semble préférable pour une bonne communication.

Un agent se structure en deux parties : une partie publique et une partie privée. La partie publique consiste à utiliser les classes et méthodes héritées (Service, StationnaryAgent et MobileAgent) et représente l'interface essentiel des interactions entre les agents et leur environnement. La partie privée est à la charge du programmeur et détermine en quelque sorte le comportement de l'agent. Nous pouvons par exemple l'utiliser comme serveur capable d'offrir des services aux autres agents.

Les agents sont donc capables de communiquer et d'interagir avec leur environnement. Il est en effet possible à un agent de communiquer avec un autre agent, mais aussi avec un utilisateur extérieur (Humain) ou une Agence. De plus, du fait la totale compatibilité MASIF de la plate-forme, les agents peuvent communiquer avec des agents extérieurs à la plate-forme GRASSHOPPER, à condition qu'ils soient eux aussi compatibles MASIF. La communication peut se faire de différentes manières :

- localement ou à distance ; une communication distante se fait indirectement en s'adressant d'abord à un serveur proxy,
- synchrone ou asynchrone,
- dynamiquement, ce qui permet de ne pas s'adresser à un serveur proxy, en spécifiant à la construction du message la signature du serveur appelé,

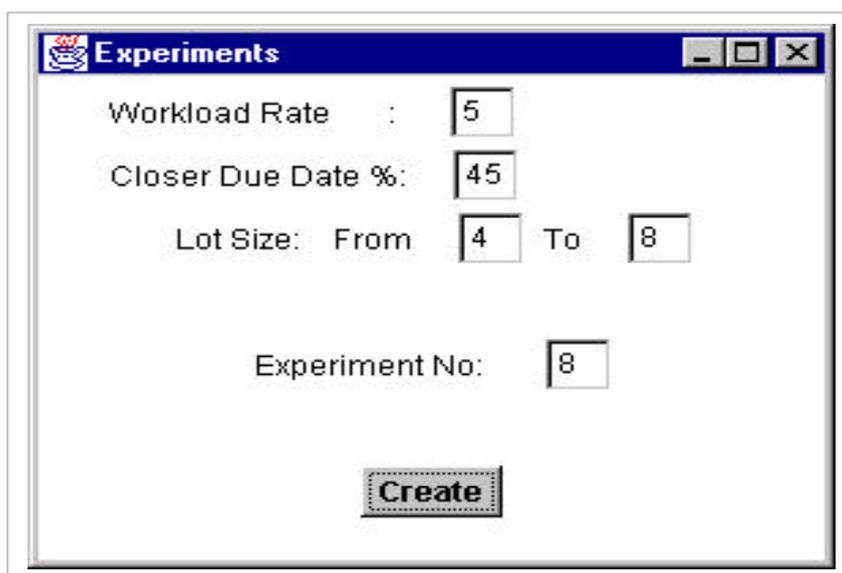
- en mode multicast permettant de s'adresser à plusieurs serveurs en parallèle.

Finalement, le modèle de Grasshopper est très proche du concept de PABADIS. Il fournit presque la même structure entre les composants du système, ce qui facilite l'implémentation des composantes de PABADIS dans Grasshopper. Un agent mobile ou stationnaire peut être considéré comme un PA ou RA suivant les conditions de fonctionnement de PABADIS. Des composants additionnels tels que le lookup service, représenté par la technologie Plug-and-Participate dans PABADIS, peuvent être facilement intégrés dans le système en tant qu'applications externes.

La plate-forme Grasshopper nous a permis de réaliser un outil de test efficace et pratique. Cet outil comporte des interfaces graphiques pour la création des benchmarks et le lancement des simulations.

### 3.3 Utilisation de l'outil de test

Le model simulé dans le cadre de notre travail a été décrit en section 2.2 de ce chapitre, En terme de réalisation, la création des benchmarks est faite par l'intermédiaire d'une interface d'utilisateur « Experiments » (voir figure 5.4). A travers cette interface, l'utilisateur spécifie la valeur de charge de travail par unité de Temps (UT), le pourcentage de la date d'échéance courte, l'intervalle de la taille de lot et le numéro de l'expérience (voir tableau 5.1).



**Figure 5.4 : Interface utilisateur pour la création des jeux d'expériences.**

La simulation est initiée par l'intermédiaire d'une interface d'utilisateur de type simulation « Simulation » (voir figure 5.5). L'utilisateur associe un modèle à un fichier d'expériences, puis il lance la simulation.

Une fois la simulation lancée, le prototype crée les entités suivantes : les neuf Agents Ressources représentés par ROBi, ROBii et ROBiii ; ils correspondent aux Robots fournissant les services de perçage « Drilling (D) », de nettoyage « Cleaning (C) » et d'assemblage « Assembly (A) » comme indiqué dans la figure 5.6.

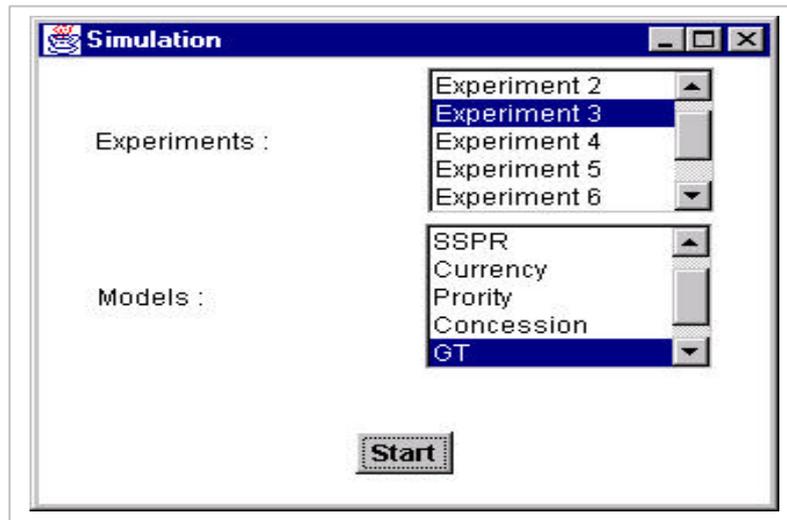


Figure 5.5 : Interface utilisateur pour le lancement de la simulation.

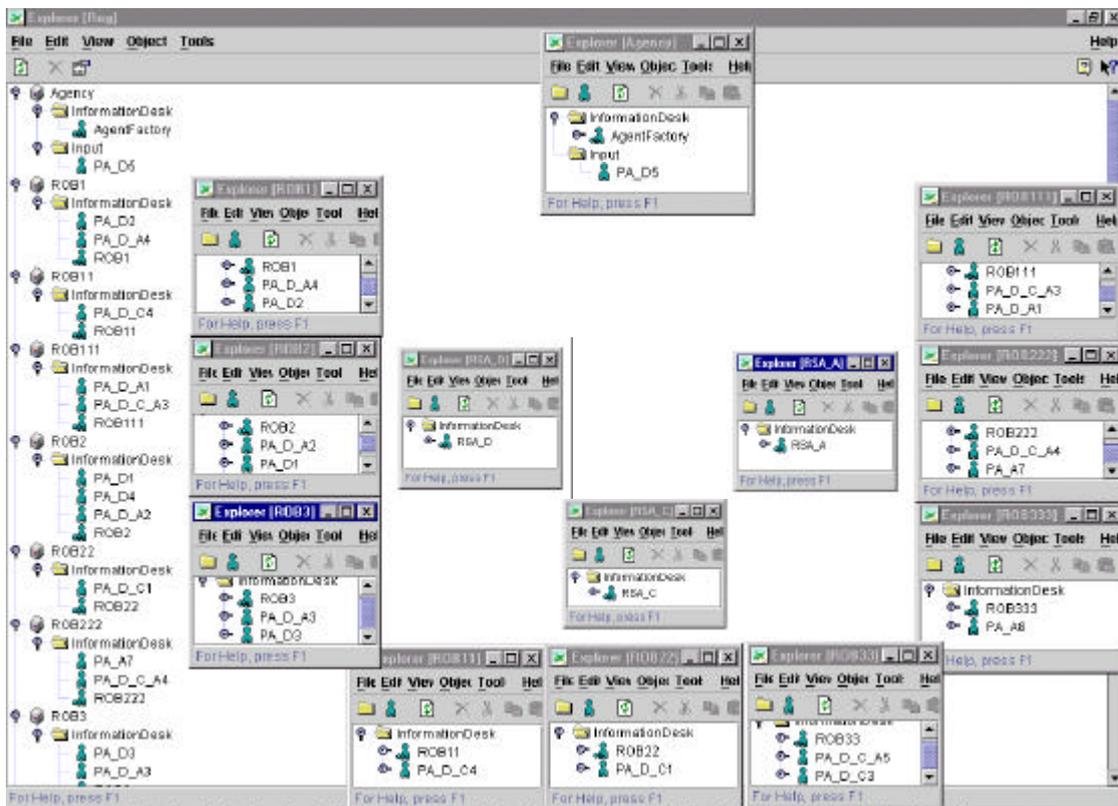


Figure 5.6 : Capture d'écran pour la simulation du modèle GT associé à l'expérience « Experiment 3 ».

Dans ce modèle, les RSA\_D, RSA\_C et RSA\_A représentent les Agents de la Session de Demande ASD pour les services déjà mentionnés ; l'Agency représente l'agent générateur des AP dans le système à partir des expériences choisies. Durant la simulation, les valeurs des différents paramètres utilisés pour les mesures de performance du système sont imprimées dans des fichiers résultats. Le nombre des AR et celui des services impliqués dans le système peut être modifié sans affecter le fonctionnement du système et cela grâce au principe de « Plug and Participate » détaillé dans le chapitre 3 de la thèse.

Grâce à cet outil nous avons pu réaliser de nombreuses simulations correspondantes aux différents scénarii suivant le plan d'expérience défini précédemment.

## **4 Tests, résultats, comparaisons et synthèses**

Dans cette partie, nous présentons les résultats obtenus pour différents Protocoles de Négociation (PN) suite aux simulations effectuées par le prototype. Le nom des protocoles utilisés dans les tableaux et les graphiques de résultats correspond, dans le chapitre 4, aux cas suivants :

- TJ : « Coopétition locale » utilisant la Théorie des Jeux (voir § 9.6),
- Currency : « Currency » (voir § 9.3),
- Concession : « Consensuel avec contraintes » (voir § 9.5),
- Centralisé : « Coopétition globale ou centralisée » (voir § 9.7)
- SSPR : « Single-Step Production Reservation » (voir § 9.2),
- Priorité : « Consensuel selon la priorité » (voir § 9.4),
- PR : « Production Reservation » (voir § 9.1).

Ces sept différents protocoles sont comparés en faisant varier des paramètres de mesure de performances (voir tableau 5.1). Ces paramètres sont les suivants :

- le temps de fabrication du produit,
- le temps d'attente du produit,
- le temps de retard du produit par rapport à une « due date »,
- le taux d'utilisation des ressources,
- la longueur des files d'attente,
- le nombre de produits en retard dans le système.

Plus précisément nous comparerons les valeurs moyennes et les écarts types des paramètres pour chaque protocole. Les mesures de performances sont rapportées dans des tableaux et des graphiques. Une analyse statistique détaillée, paramètre par paramètre, puis globale des résultats a été effectuée. Le numéro des huit expériences dans les tableaux et les graphiques correspondent aux numéros des expériences figurant dans le tableau 5.1. Dans ces graphes, nous avons classé les valeurs obtenues

et écarté les valeurs les plus mauvaises des protocoles et nous affichons seulement les valeurs des cinq premiers protocoles de négociation pour montrer clairement la différence de résultats existant entre les protocoles. Nous pouvons remarquer que les valeurs obtenues sont sensiblement les mêmes (à 1% de différence) pour les répétitions des mêmes simulations. Cette faible différence est liée à la répartition et à l'ordre d'exécution des tâches entre les agents lors de la simulation avec la plateforme de test. En effet, en cas de synchronisation d'une demande d'exécution pour la même tâche, l'ordre du choix entre les agents peut varier lors de la répétition de l'expérience.

Dans ce qui suit, nous allons analyser successivement et en détail chaque paramètre de performance, puis nous porterons notre attention sur une analyse de données pour nous permettre de synthétiser un ensemble de résultats obtenus.

#### 4.1 Etude du paramètre « Temps de Fabrication » (TF)

Dans le tableau 5.3, nous détaillons les valeurs de la moyenne et l'écart type de temps de fabrication des produits dans le système et pour les différents Protocoles de négociation. Nous constatons que le modèle TJ a les valeurs les plus basses des moyennes calculées sauf pour l'expérience numéro 1 et numéro 6. SSPR garde les mêmes valeurs moyennes pour les expériences à faible, (expérience no : 1, 2, 3, 4), et à fort pourcentage, (expérience no : 5, 6, 7, 8), de produits ayant des courtes date d'échéance. En effet, ce protocole ne prend pas en considération le paramètre «date d'échéance» pour le choix des Ressources. Les deux derniers modèles Priorité et PR donnent les plus mauvais résultats. Les autres modèles restants ont des performances voisines (voir figure 5.7). De plus, le temps de fabrication augmente selon la charge de travail et la taille du lot utilisée. Pour des lots de faible taille, nous obtenons des temps de fabrication courts qui augmentent pour les lots de grande taille.

		Expériences							
		1	2	3	4	5	6	7	8
TJ	Moyenne	44,441	49,422	80,185	86,290	42,821	50,758	87,369	92,912
	Ecart-Type	6,668	8,220	11,010	5,354	4,936	8,006	15,899	11,195
Currency	Moyenne	44,897	50,107	85,500	88,717	42,630	49,766	87,670	94,298
	Ecart-Type	7,815	8,863	13,052	8,227	5,106	8,426	15,208	13,288
Concession	Moyenne	47,847	54,466	87,912	87,842	53,972	49,213	107,196	95,090
	Ecart-Type	10,598	10,182	12,777	32,686	12,608	8,555	23,957	10,829
Centralisé	Moyenne	44,171	49,523	84,861	90,741	47,393	52,965	99,800	93,059
	Ecart-Type	6,718	7,800	18,159	11,297	7,096	10,159	20,281	10,553
SSPR	Moyenne	43,901	49,507	88,674	94,196	43,901	49,730	87,667	94,849
	Ecart-Type	5,965	8,103	14,042	10,564	5,965	8,251	14,344	18,557
Priorité	Moyenne	50,434	58,627	95,811	134,871	67,474	80,473	284,139	218,969
	Ecart-Type	16,954	14,775	27,639	89,954	34,383	39,240	272,760	205,148
PR	Moyenne	430,907	429,060	418,555	399,415	329,241	459,285	449,280	433,893
	Ecart-Type	326,015	314,493	275,390	248,940	359,988	309,595	292,019	283,322

Tableau 5.3 : Moyenne et écart type des temps de fabrication des produits

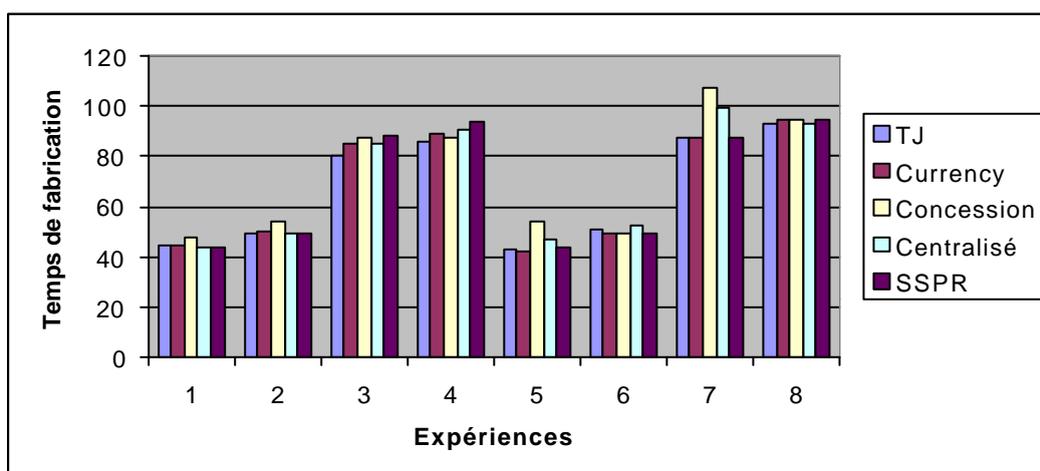


Figure 5.7 : Histogramme de la moyenne des temps de fabrication des produits.

De même, le temps de fabrication varie proportionnellement à l'importance de la charge de travail. Il reste à signaler que les valeurs des écart-types calculées sont élevées à cause de l'absence d'homogénéité de la population des produits.

#### 4.2 Etude du paramètre « Temps d'Attente » (TA)

Nous constatons que le modèle TJ a les valeurs les plus basses pour des moyennes calculées sauf pour l'expérience numéro 1 et 2 (voir le tableau 5.4). Les deux derniers modèles Priorité et PR ont les plus mauvais résultats. Les autres modèles restants ont des performances voisines (voir figure 5.8). De même le SSPR garde les mêmes valeurs moyennes pour des expériences à faible et à fort pourcentage de produits à courtes date d'échéance. Comme pour le temps de fabrication, le temps d'attente augmente selon la charge de travail et la taille du lot utilisée. Les valeurs des écarts types calculés sont élevées à cause de la non-homogénéité de la population des produits.

		Expériences							
		1	2	3	4	5	6	7	8
TJ	Moyenne	10,568	15,295	45,757	51,798	8,683	16,663	52,893	58,468
	Ecart-Type	6,325	7,588	11,615	6,640	3,381	7,179	16,150	11,570
Currency	Moyenne	10,824	15,822	50,922	54,278	8,442	15,517	53,313	59,714
	Ecart-Type	6,831	7,130	12,601	6,861	3,440	6,810	15,502	12,898
Concession	Moyenne	13,657	19,974	53,452	53,398	18,321	15,182	70,387	60,292
	Ecart-Type	9,223	9,542	13,098	10,409	8,519	7,901	24,083	11,215
Centralisé	Moyenne	10,171	15,253	50,339	56,265	12,440	18,448	63,860	58,555
	Ecart-Type	5,746	6,874	18,594	11,493	4,629	8,110	20,616	11,007
SSPR	Moyenne	9,917	15,380	54,230	59,720	9,917	15,603	53,206	60,422
	Ecart-Type	5,378	7,469	14,417	10,929	5,378	7,501	14,777	18,884
Priorité	Moyenne	16,153	24,184	61,287	100,030	33,111	45,584	248,520	183,746
	Ecart-Type	15,088	13,589	26,188	89,746	32,232	37,256	270,070	202,545
PR	Moyenne	396,447	394,568	384,047	364,955	295,003	424,682	414,804	399,290
	Ecart-Type	326,664	314,495	275,283	248,848	359,737	309,305	292,035	283,529

Tableau 5.4 : Moyenne et écart type des temps d'attente des produits

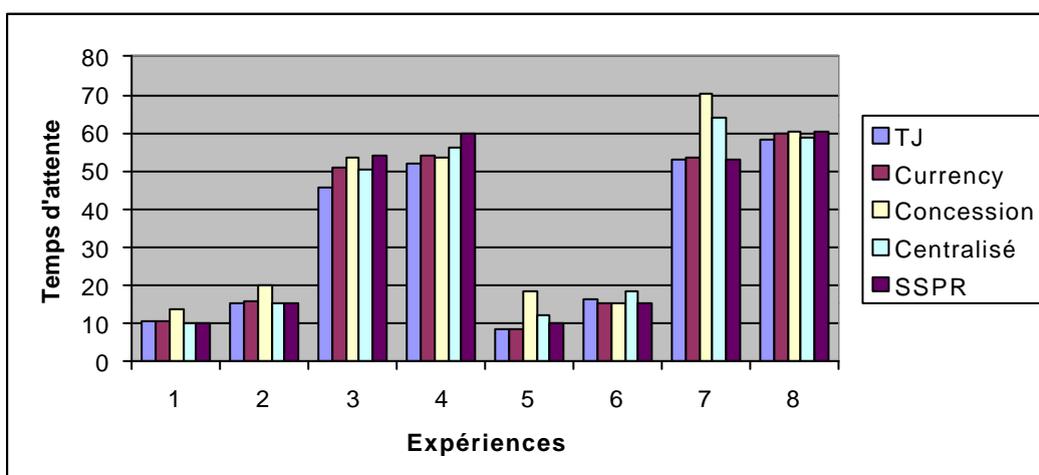


Figure 5.8 : Histogramme de la moyenne de temps d'attente des produits.

### 4.3 Etude du paramètre « Temps de Retard » (TR)

Le modèle TJ a les plus basses valeurs des moyennes calculées, sauf pour l'expérience numéro 5 (voir le tableau 5.5). Les deux derniers modèles Priorité et PR ont les plus mauvais résultats. Les autres modèles restants ont des performances voisines et conservent des bonnes performances pour les faibles pourcentages de courte date d'échéance et se dégradent quand le pourcentage devient plus fort (voir figure 5.9). Tandis que le modèle TJ maintient de bonnes performances dans toutes les situations et les expériences. Le temps de retard, quant à lui, augmente selon la charge de travail, la taille du lot utilisée et la date d'échéance.

		Expériences							
		1	2	3	4	5	6	7	8
TJ	Moyenne	0,365	1,175	2,890	5,578	9,645	12,715	58,268	61,193
	Ecart-Type	0,293	0,750	2,339	2,222	6,898	8,487	43,332	43,507
Currency	Moyenne	0,508	1,630	5,039	6,041	12,371	14,661	58,584	64,970
	Ecart-Type	0,801	1,339	3,966	2,398	8,490	8,469	42,484	43,236
Concession	Moyenne	0,605	1,310	5,270	6,278	11,269	10,027	64,919	66,806
	Ecart-Type	0,462	0,693	5,105	3,705	8,993	5,924	53,636	44,661
Centralisé	Moyenne	0,540	1,200	7,060	9,273	14,863	15,896	71,380	75,440
	Ecart-Type	0,603	0,726	7,190	5,183	9,915	10,822	53,959	50,206
SSPR	Moyenne	0,455	1,345	7,068	10,630	12,365	15,280	59,400	65,230
	Ecart-Type	0,325	1,012	6,263	5,952	10,266	9,569	42,334	43,600
Priorité	Moyenne	2,650	1,995	17,940	75,833	23,130	37,675	296,095	200,043
	Ecart-Type	3,614	1,149	18,117	83,372	14,807	23,380	202,837	142,514
PR	Moyenne	546,805	542,880	520,465	488,583	414,345	606,845	566,293	546,778
	Ecart-Type	389,940	379,600	362,042	320,998	336,246	379,568	362,265	341,568

Tableau 5.5 : Moyenne et écart type des temps de retard des produits

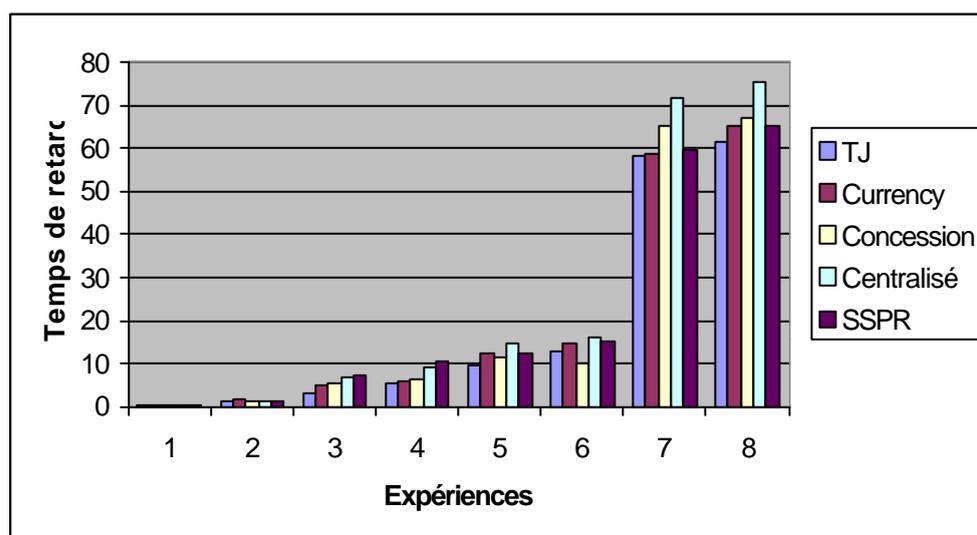


Figure 5.9 : Histogramme de la moyenne des temps de retard des produits.

#### 4.4 Etude du paramètre « Taux d'Utilisation du Système » (TUS)

Dans le tableau 5.6, le modèle Currency dispose des plus grandes valeurs des moyennes calculées. Les deux modèles TJ et SSPR ont des performances voisines de celles du modèle currency. Les deux derniers modèles Priorité et PR ont les plus mauvais résultats. Les autres modèles restants ont de mauvaises performances avec de grands écarts types (voir figure 5.10). Le modèle TJ offre de bonnes performances dans toutes les situations et sur toutes les expériences. Le taux d'utilisation du système augmente selon la charge de travail et la taille du lot utilisée.

		Expériences							
		1	2	3	4	5	6	7	8
TJ	Moyenne	0,341	0,380	0,616	0,663	0,327	0,390	0,671	0,714
	Ecart-Type	0,098	0,095	0,147	0,112	0,071	0,097	0,174	0,151
Currency	Moyenne	0,391	0,439	0,733	0,770	0,372	0,433	0,759	0,821
	Ecart-Type	0,098	0,078	0,136	0,107	0,067	0,074	0,202	0,139
Concession	Moyenne	0,172	0,195	0,315	0,346	0,199	0,182	0,480	0,477
	Ecart-Type	0,044	0,058	0,070	0,129	0,095	0,069	0,343	0,336
Centralisé	Moyenne	0,149	0,167	0,285	0,306	0,165	0,184	0,419	0,439
	Ecart-Type	0,036	0,033	0,082	0,064	0,053	0,061	0,288	0,319
SSPR	Moyenne	0,301	0,339	0,607	0,645	0,301	0,341	0,600	0,647
	Ecart-Type	0,077	0,084	0,148	0,127	0,102	0,084	0,149	0,167
Priorité	Moyenne	0,060	0,069	0,114	0,164	0,080	0,096	0,339	0,260
	Ecart-Type	0,021	0,021	0,033	0,159	0,044	0,066	0,631	0,364
PR	Moyenne	0,479	0,477	0,465	0,444	0,366	0,510	0,499	0,482
	Ecart-Type	0,384	0,374	0,323	0,298	0,478	0,413	0,399	0,385

Tableau 5.6 : Moyenne et écart type de taux d'utilisation du système.

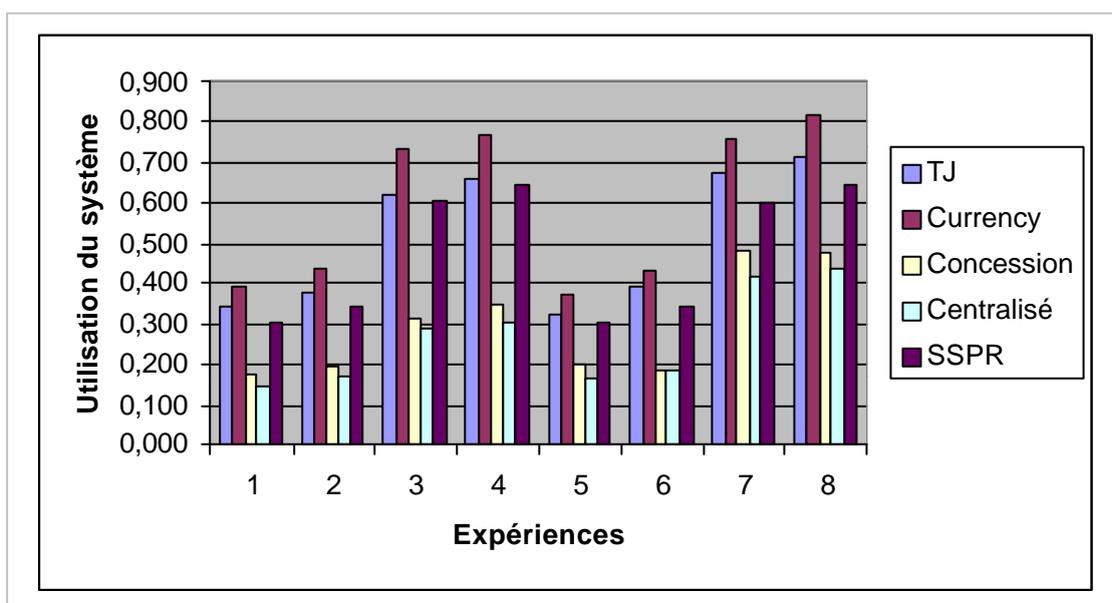


Figure 5.10 : Histogramme de la moyenne de taux d'utilisation du système.

#### 4.5 Etude du paramètre « Longueur des Files d'Attente » (LFA)

Le modèle TJ a les valeurs les plus basses des moyennes calculées sauf pour l'expérience numéro 1 et numéro 2 (voir le tableau 5.7). Les deux derniers modèles Priorité et PR ont les plus mauvais résultats. Les autres modèles restants ont des performances voisines (voir figure 5.11). Comme pour le temps de fabrication, le temps d'attente augmente selon la charge de travail et la taille des lots utilisés. De même, les valeurs des écarts types calculés sont élevés par rapport aux valeurs moyennes.

		Expériences							
		1	2	3	4	5	6	7	8
TJ	Moyenne	0,312	0,448	1,329	1,502	0,256	0,489	1,534	1,697
	Ecart-Type	0,187	0,222	0,337	0,192	0,010	0,211	0,468	0,336
Currency	Moyenne	0,320	0,461	1,506	1,591	0,248	0,456	1,572	1,738
	Ecart-Type	0,202	0,208	0,373	0,201	0,101	0,200	0,457	0,375
Concession	Moyenne	0,399	0,579	1,551	1,550	0,649	0,489	2,887	2,862
	Ecart-Type	0,270	0,277	0,380	0,302	0,429	0,315	2,092	2,078
Centralisé	Moyenne	0,299	0,445	1,463	1,632	0,437	0,589	2,621	2,779
	Ecart-Type	0,169	0,201	0,540	0,333	0,231	0,320	1,791	2,040
SSPR	Moyenne	0,292	0,451	1,574	1,732	0,290	0,483	2,558	2,740
	Ecart-Type	0,158	0,219	0,419	0,317	0,161	0,220	0,429	0,550
Priorité	Moyenne	0,474	0,704	1,775	2,871	0,967	1,307	6,977	5,233
	Ecart-Type	0,443	0,396	0,759	2,579	0,941	1,068	7,582	5,769
PR	Moyenne	11,504	11,439	11,129	10,591	8,616	12,273	12,032	11,539
	Ecart-Type	9,480	9,118	7,977	7,221	10,507	8,939	8,471	8,194

Tableau 5.7 : Moyenne et écart type des files d'attente dans le système.

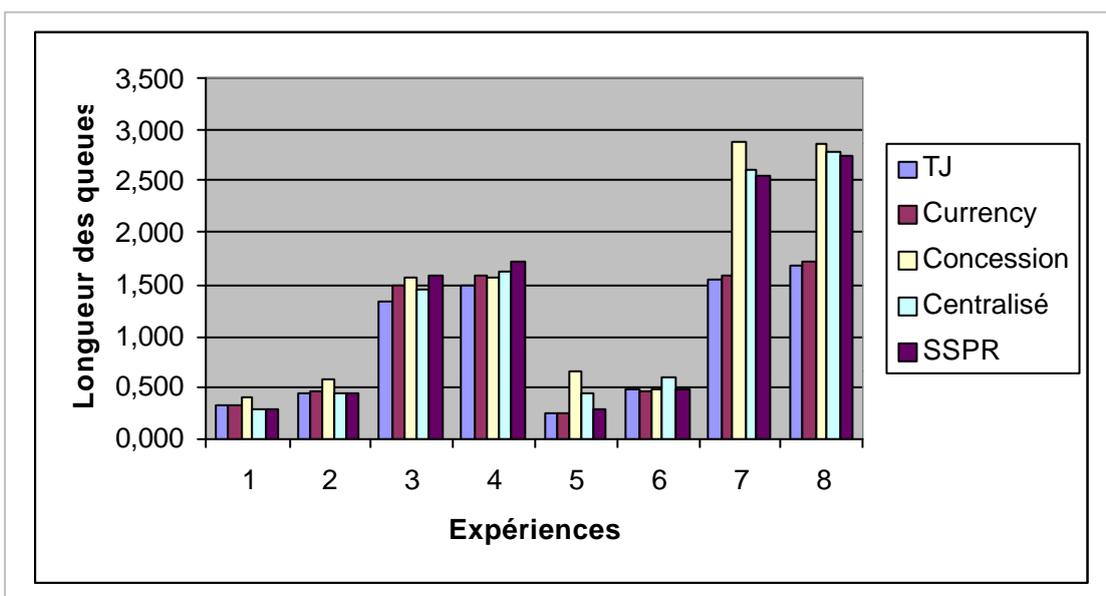


Figure 5.11 : Histogramme de la moyenne des files d'attente dans le système.

#### 4.6 Etude du paramètre « Nombre des Produits en Retard » (NPR)

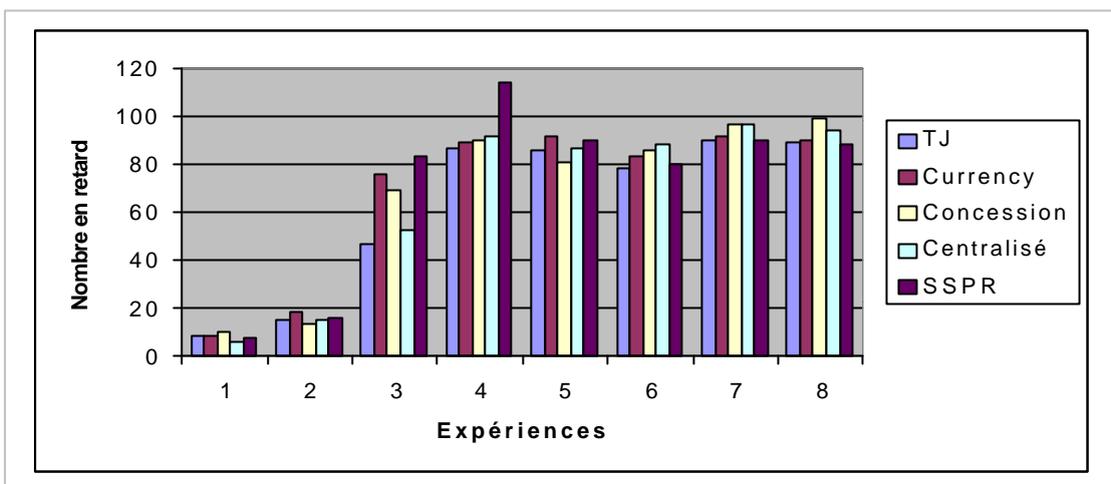
Le modèle TJ est bien classé par rapport aux autres modèles concernant le nombre de produits en retard ou «delayed parts»(voir le tableau 5.8). Pour chacune des huit expériences, le pourcentage d'échec varie entre 3 % et 45 % pour les meilleurs modèles (voir le tableau 5.9). Le modèle Concession a de bonnes performances pour les expériences à faible charge de travail, tandis que le modèle SSPR se distingue dans les expériences à fortes perturbations. Le protocole centralisé est bien classé au niveau somme totale des produits en retard pour toutes les expériences (voir tableau 5.8). Les deux derniers modèles Priorité et PR ont les plus mauvais résultats (voir figure 5.12). Le nombre des produits en retard augmente selon la charge de travail, la taille du lot utilisée et la date d'échéance.

	Expériences								Total /1600
	1	2	3	4	5	6	7	8	
TJ	9	15	47	87	85	78	90	89	500
Currency	9	18	76	89	92	83	92	90	549
Concession	10	14	69	90	81	85	97	99	545
Centralisé	6	15	52	91	87	88	96	94	529
SSPR	8	16	83	114	90	80	90	88	569
Priority	19	22	82	143	89	90	152	137	734
PR	152	135	149	174	115	120	137	138	1120

Tableau 5.8 : Nombre de produits en retard dans le système.

	Expériences								Total /1600
	1	2	3	4	5	6	7	8	
TJ	4,50%	7,50%	23,50%	43,50%	42,50%	39,00%	45,00%	44,50%	31,25%
Currency	4,50%	9,00%	38,00%	44,50%	46,00%	41,50%	46,00%	45,00%	34,31%
Concession	5,00%	7,00%	34,50%	45,00%	40,50%	42,50%	48,50%	49,50%	34,06%
Centralisé	3,00%	7,50%	26,00%	45,50%	43,50%	44,00%	48,00%	47,00%	33,06%
SSPR	4,00%	8,00%	41,50%	57,00%	45,00%	40,00%	45,00%	44,00%	35,56%
Priority	9,50%	11,00%	41,00%	71,50%	44,50%	45,00%	76,00%	68,50%	45,88%
PR	76,00%	67,50%	74,50%	87,00%	57,50%	60,00%	68,50%	69,00%	70,00%

Tableau 5.9 : Pourcentage d'échec des produits respectant leur date d'échéance.



**Figure 5.12 : Histogramme de nombre des produits en retard.**

#### **4.7 Analyse statistique multivariée des résultats**

Après la présentation détaillée séquentielle et partielle des résultats de chaque protocole de négociation par rapport aux paramètres de mesure de performances, il convenait de faire une Analyse de Données globale des informations obtenues. Nous avons utilisé l'Analyse en Composantes Principales (ACP) comme technique statistique pour étudier les relations entre les variables et paramètres impliqués dans ces expériences et faire ressortir des propriétés globales, des avantages et inconvénients attachés à chaque protocole.

##### **4.7.1 Rappel sur l'analyse en composantes principales**

L'analyse en composantes principales (ACP) permet de « réduire » un système complexe de corrélations et de le représenter en un plus petit nombre de dimensions. Le principe de cette méthode est d'obtenir une représentation approchée du nuage des  $n$  individus (ou Protocole de Négociation (PN)) caractérisés par  $m$  variables dans un sous espace de dimension plus faible [Saporta, 1990]. Pour ce faire, on procède à une projection de la population représentée dans un espace à  $m$  dimensions vers un hyperplan à  $m' < m$  dimensions.

Par exemple dans la thèse, nous cherchons à représenter les proximités des  $n$  individus, correspondant aux résultats des protocoles dans chacune des huit expériences, par rapport aux  $p$  paramètres de mesures de performances. L'Analyse en Composantes Principales se divise en quatre étapes :

- normalisation des données pour être indépendant des unités des  $p$  paramètres,
- calcul d'une matrice de similarité  $C$  (bien souvent la corrélation),
- recherche des éléments propres de  $C$ , qui donnent les axes principaux,

- représentation des individus dans le nouvel espace (en ne considérant que les valeurs propres expliquant une variance cumulée suffisante).

Géométriquement, le processus revient à placer des axes dans l'espace à  $m$  dimensions (un axe par dimension). Dans le cas où  $m = 100$ , il est peu probable que nous puissions obtenir, par projection orthogonale suffisamment d'informations pertinentes et significativement ordonnées sur un seul axe (axe1) appelé première composante principale. Nous aurons besoin d'axes supplémentaires pour compléter la représentation du nuage de points. Par convention, nous représentons la deuxième dimension synthétique (axe2) par une droite perpendiculaire à la première composante principale. Ce deuxième axe, ou deuxième composante principale, se définit comme la droite qui "explique" au mieux une grande partie de l'information restante. Toutes les droites suivantes ou axes factoriels seront orthogonales aux axes précédents et contiennent de moins en moins d'informations pertinentes.

#### 4.7.2 Discussion

Dans notre Analyse en Composantes Principales (ACP), la normalisation des valeurs de chaque individu  $PN_i^*$  dans une expérience donnée est obtenue en divisant cette dernière par la somme de tous les  $PN_i$  correspondant à une même expérience

( $PN_i^* = \frac{PN_i}{\sum PN_i}$ ). Les paramètres de mesure de performances seront les variables de

l'Analyse en Composantes Principales. Les Individus correspondent aux valeurs moyennes des protocoles des huit expériences.

Les **Variables** sont au nombre de six : TF, TA, TR, TUS, LFA, NPR qui correspondent respectivement aux paramètres de mesure de performances suivants : Temps de Fabrication, Temps d'Attente, Temps de Retard, Taux d'Utilisation du Système, Longueur des Files d'Attente, et le Nombre de Produit en Retard.

Les **paramètres** sont au nombre de cinq : S1, S2, S3, S4 et S5 qui correspondent respectivement aux cinq protocoles de négociation suivants : TJ, Currency, Concession, Centralisé et SSPR.

Les **Individus  $S_{ii}$** , au nombre de 40, correspondent aux couples associant un protocole de communication donné  $S_i$  à un cas expérimental  $i$  choisi parmi huit autres (voir tableau 5.1).

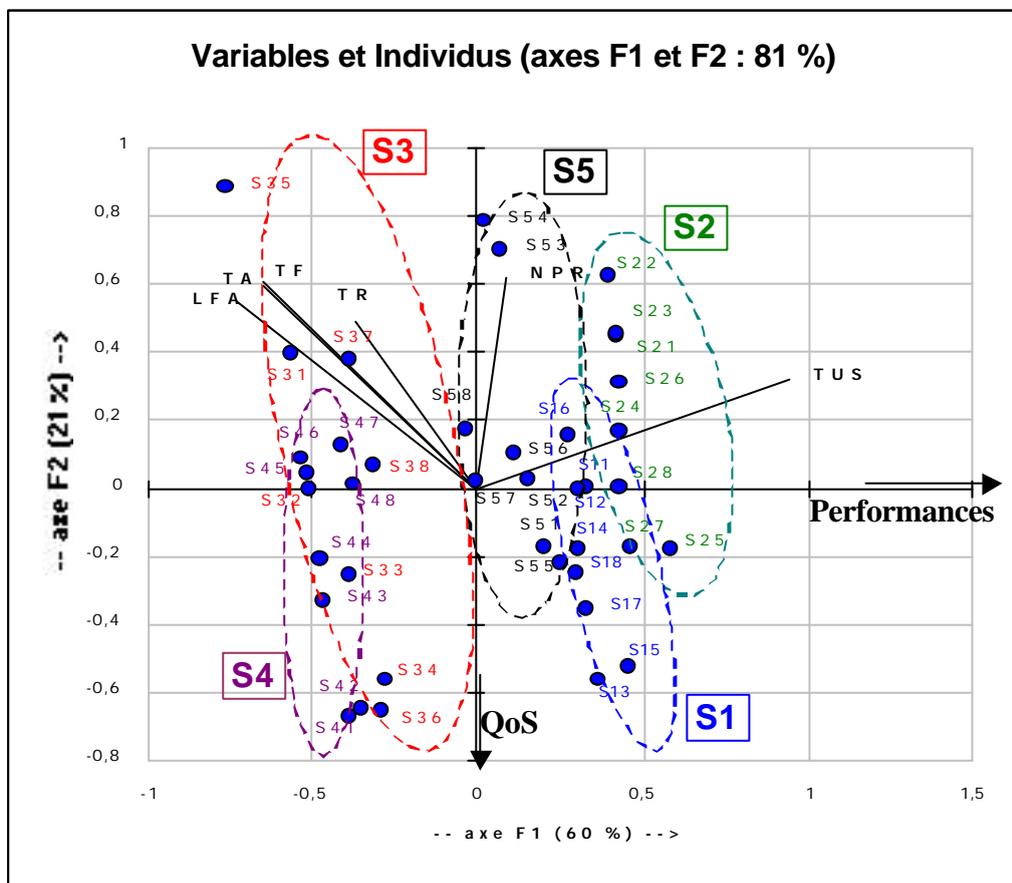
L'ACP porte sur les valeurs moyennes obtenues. Le logiciel utilisé permet de faire apparaître (clustering) des regroupements de cas de test homogènes (classes) pour en faciliter l'analyse détaillée (comportement des individus dans chaque classe). Nous ne la détaillerons pas dans ce paragraphe.

Dans la figure 5.13, les deux axes F1 et F2 sont les deux axes principaux choisis expliquant 81 % de l'information dans la population de test. TUS et LFA ont une forte contribution dans l'axe F1 tandis que F2 est influencé par TF, TR et NPR.

Compte tenu de la contribution de ces valeurs aux axes factoriels F1 et F2 nous en déduits que :

- F1 est représentatif des performances du processus ; à droite, nous aurons des observations correspondant à des bons taux d'utilisation, tandis qu'à gauche de l'axe, les files d'attente seront importantes.
- F2 correspondra quant à lui à une bonne qualité de service (QoS ; « Quality of Service ») du processus : en haut de cet axe, nous aurons des valeurs correspondant à de nombreux produits ayant de temps de retard importants.

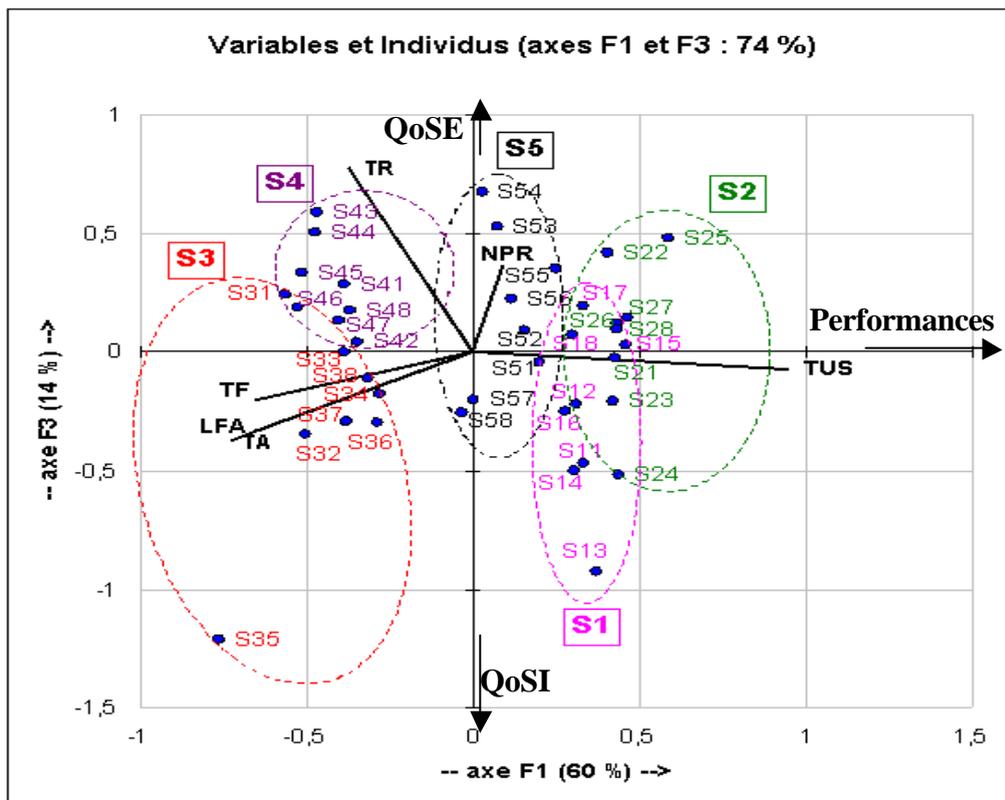
Cela nous permet donc de constater que S1 et S4 sont les protocoles de négociation qui permettent d'assurer la meilleure qualité de service. Toutefois en termes de performances du processus, S1 sera supérieur à S4.



**Figure 5.13 : Analyse en Composantes Principales. Représentation des deux axes principaux F1 et F2.**

Dans la figure 5.14, l'axe F1 reste marqué par TUS toutefois, l'influence de LFA est un peu moins marquée mais remplacée par le temps de fabrication TF. F1 reste donc un axe représentatif de la performance du processus. Par contre, F3 est influencé dans sa partie haute par TR et NPR (Qualité Client) et dans sa partie basse

par TA, TF et TFA : il s'agit de critères portant sur la qualité interne de processus. Donc F3 reste un axe de qualité mais de qualité externe dans sa partie haute (QoSE) et interne dans sa partie basse (QoSI). L'analyse du graphe montre que S1 continue de garder ses propriétés, tandis que S4 répond en fait plus mal à la notion externe vue par le client. S4 est au même niveau que S2, alors que S2 est supérieur en termes de performances. Ceci fait que S2 est globalement moins mauvaise que S4, etc. Nous pouvons aussi faire ressortir d'autres phénomènes à savoir que les individus S11 à S14 sont les mieux positionnés de même que (et dans une moindre mesure, S23 et S24).



**Figure 5.14 : Analyse en Composantes Principales. Representation des deux axes principaux F1 et F3.**

En résumé, la répartition des protocoles dans les deux figures 5.13 et 5.14 représente 95 % d'explication. Dans les deux graphes, des groupements/classes d'individus homogènes sont représentés par des pointillés mettant en avant des points particuliers pour chaque individu. Le protocole « S1, (équivalent à TJ) », reste bien classé pour les cinq axes de variables choisis. Le protocole « S2, (équivalent à Currency) » et « S5, (équivalent à SSPR) » montre qu'ils sont bien classés pour l'axe TUS et moins bien classés pour l'axe NPR. Pour les protocoles restants nous remarquons qu'ils ont des performances voisines. De plus, nous pourrions noter la forte corrélation entre les axes des variables TA, TF et LFA qui traduisent une mauvaise qualité interne du processus.

## 4.8 Conclusion

Dans ce chapitre, Un prototype expérimental a été conçu, élaboré et décrit. Il a permis de tester les différents concepts et mécanismes développés dans le chapitre 4 pour une gestion plus performante d'un système de production décentralisé.

Le plan de test a été conçu autour de huit cas expérimentaux impliquant de nombreuses combinaisons et variations de variables telles que la charge de travail, la date d'échéance et la taille du lot des produits. Le but principal de ce plan de test a été de créer un environnement dynamique correspondant à diverses situations et perturbations du système.

Le prototype élaboré nous a permis de tester facilement les différents protocoles de négociation développés et grâce à la simulation.

Les résultats des simulations de chaque protocole ont été analysés et comparés pour toutes les expériences effectuées par l'intermédiaire de tableaux et histogrammes graphiques. Cette analyse a été faite pour les six paramètres de mesure des performances du système que nous avons sélectionnés. L'analyse en composantes principales a été utilisée pour dégager une synthèse globale des performances de chaque protocole de négociation.

Nous avons constaté de bonnes performances pour le protocole, nommé TJ, utilisant l'approche de Coopétition locale et la Théorie des Jeux. Ces performances ont montré une stabilité et robustesse notables dans un environnement dynamique en évolution permanente. Pour les autres protocoles nous pouvons mettre en avant les avantages et les inconvénients de chacun d'eux. Par exemple, Le modèle Currency a obtenu des bons résultats au niveau du paramètre «Taux d'Utilisation du Système» mais pas pour le paramètre «Temps de Retard». Le Protocole SSPR présente de bons résultats avec le modèle Currency et TJ pour le paramètre «Taux d'Utilisation du Système» mais il n'est pas bien classé au niveau des autres paramètres. Les deux modèles «Concession» et «Centralisé» ont des performances moyennes globalement voisines. Les deux derniers modèles «Priorité» et PR ont les plus mauvais résultats dans toutes les expériences.

Au travers de ces différentes expériences, évaluations et comparaisons effectués, nous pouvons confirmer que l'utilisation du modèle TJ dans un système de production décentralisé est souvent une bonne solution dans un environnement dynamique situé.

Dans ce chapitre, également une comparaison des performances entre les différents protocoles a été réalisée. Toutefois celle-ci n'a pas été effectuée avec les approches conventionnelles ; En effet, notre attention a surtout porté sur la faisabilité et l'applicabilité de nouveaux concepts de gestion de production et le manque de temps ne nous a pas permis de comparer les avantages issus de notre démarche avec des techniques de type MRP mises en œuvres dans des cas concrets dans l'industrie.

Ce point à lui seul, comportant des analyses d'efficacité et de performance, peut constituer un travail de thèse.

Nous avons cependant utilisé ces concepts pour étendre au niveau expérimental les capacités de gestion de la production fournie par le démonstrateur du LGI2P-EMA.

Nous avons pu montrer que les principes de collaboration (au sens large du terme) sont applicables et qu'ils permettent aujourd'hui d'envisager des systèmes de production entièrement dirigés par la demande, qu'elle soit intérieure ou extérieure. Les gains réels financiers engendrés par une telle démarche doivent donc être encore évalués. Des cabinets d'études tels que le Gartner Research Group, ou Price Waterhouse Cooperation (IBM) les estiment non négligeables ; un ratio de l'ordre de 10 % pourrait être envisageable.

Finalement, les performances des protocoles développés peuvent être améliorées en approfondissant certaines notions au niveau des critères de choix des agents voisins, lors de la prise de la décision d'un « Agent-Produit » durant une « Session de Demande ». Cela fera l'objet de travaux complémentaires avant de déboucher sur un système plus complet et industrialisé.

# Chapitre 6 : Conclusions et perspectives

## 1 Synthèse

Dans ce travail de recherche, nous nous sommes intéressés au pilotage décentralisé et hétérarchique des systèmes de production complexes dans un environnement dynamique situé. L'approche de l'allocation dynamique des ressources a été préférée à la planification et l'ordonnancement préalable des ressources, ces dernières approches étant difficiles à gérer. L'allocation a été effectuée au moment de la demande, en temps réel et de manière locale « Situé » entre des entités autonomes ou agents représentant les produits et les ressources du système.

Les approches utilisées dans cette thèse sont basées sur les principes de l'intelligence collective et d'économie comme l'auto-organisation, la coopération, compétition, « coopétition » et la théorie des jeux. La mise en place de ces différents concepts est faite grâce à la technologie multi-agent. Les performances des différents protocoles de négociation développés ont été évaluées sur un simulateur basé sur la structure décentralisée du projet européen PABADIS.

Le problème qui se pose alors, est de faire émerger des solutions stables et performantes correspondant à un objectif prédéfini. Nous avons ainsi proposé de nouvelles approches, concepts et mécanismes pour l'obtention d'un comportement auto-organisé, réactif et adaptable entre les entités et répondant à ces contraintes de compétitivité.

Notre contribution porte, ici sur les quatre aspects suivants :

1. Revue détaillée et état de l'art : une analyse comparative est faite des structures de pilotages existantes des systèmes de production, des protocoles de négociation utilisés entre entités autonomes ou agents pour l'allocation dynamique des ressources dans les structures hétérarchiques, selon différentes perspectives, horizons et types de décision.
2. Concepts de fonctionnement : la conception de nouvelles approches à base des principes de la coopération, de la coopétition pour le pilotage auto-organisé et hétérarchique a été réalisée.
3. Communication : de nouveaux protocoles de négociation basés sur un nouveau concept de communication en temps réel nommé « Session de Demande » ont été développés ; ils sont basés sur des principes récents de l'économie comme la « coopétition », la théorie des jeux et le travail

collaboratif. La technologie agent a été mise en œuvre pour le pilotage décentralisé du système et selon une approche hybride (réactive et cognitive).

4. Application et réalisation : le test, la validation et l'évaluation des concepts cités ci-dessus ont été effectués à travers un simulateur d'un atelier de production basée sur une architecture de pilotage hétérarchique. Les résultats obtenus des approches testées ont montré une grande stabilité, adaptabilité et robustesse face à la dynamique et aux perturbations introduites dans le système.

Dans le travail de thèse de Yingjiu LIU, réalisé précédemment les mécanismes et principes généraux liés à une gestion hétérarchique et collaborative avaient été introduits. A partir de règles locales simples, il avait été démontré qu'il était possible, par émergence spontanée, de générer des configurations de produits et de systèmes de productions capables de satisfaire rapidement une demande impromptue émanant d'un client ; ceci correspond à une démarche de type : « production à la demande ».

Les présents travaux, se placent dans cette continuité, et notre effort a porté sur l'élaboration de mécanismes locaux de fonctionnement et de communication plus avancés et plus complets au niveau de chaque agent. Ceux-ci permettent aux phénomènes d'auto-organisation et de reconfiguration automatique de systèmes de production de s'exercer pleinement. L'efficacité et l'efficience des démarches ont été démontrées au travers de nombreux cas de test, et c'est en ce sens que nous dirons que les objectifs du travail ont été atteints. Actuellement nous pouvons donc procéder à une nouvelle étape consistant à intégrer ces démarches dans les approches « collaboratives ». Le marché est encore peu mature, mais avec le nouvel élan attendu dans le commerce « collaboratif » et les places de marché, les applications industrielles vont devoir faire appel à de telles technologies innovantes.

## **1.1 Discussion générale**

Dans le premier aspect de nos contributions, cité ci-dessus, l'analyse et la critique des différents concepts et approches existants pour la gestion de la production au niveau organique et fonctionnel nous ont permis de choisir la structure décentralisée pour sa réactivité, flexibilité et adaptabilité dans les environnements dynamiques et ouverts. L'allocation dynamique des ressources a été utilisée pour remplacer les approches conventionnelles, basées sur le principe de l'ordonnancement, qui ont montré des limites et des insatisfactions face aux exigences actuelles et futures des systèmes de production. Elle est réalisée à travers des protocoles de négociation entre entités autonomes représentant les produits et les ressources du système.

Parmi les perspectives, horizons et type de décision utilisés dans les protocoles de négociation nous avons privilégié les approches « Orienté-Produit » utilisant le principe d'allocation « Tâche par Tâche » ou « SSPR ». Ces approches ont montré

plus de réactivité et efficacité dans un environnement dynamique à forte perturbation..

Dans le second aspect, les principes de compétition, coopération, «coopétition» et «compération», utilisés en économie, ont été étudiés, explorés et classés selon un contexte industriel, et selon plusieurs scénarios pour l'allocation dynamique des ressources. Le scénario appliquant le principe de la «coopétition» entre les «Agent-Produit» pour un pilotage auto-organisé et hétérachique a été choisi et développé. Son principal avantage est la prise en considération l'influence de la demande des clients et l'arrivée dynamique des produits dans le système.

Dans le troisième aspect, un nouveau concept nommé «Session de Demande» a été développé. Ce dernier nous a permis de gérer les interactions et l'organisation de la décision entre les entités autonomes ou agents en temps réels. De même, la «Session de Demande» a créé un environnement favorable pour l'application du concept de «coopétition = coopération + compétition». En effet, les agents coopèrent au début de leur participation à la session et ils rentrent en compétition lors de leur prise de la décision à travers la théorie des jeux, à la fin de la session. Enfin, la méthodologie et les protocoles de négociation développés ont été conçus à base de ce concept et selon l'approche coopérative et «coopetitive».

Grâce aux caractéristiques propres des agents au niveau de la modularité, la décentralisation et de l'adaptabilité, la technologie d'agents a été adoptée pour la mise en œuvre du pilotage décentralisé des systèmes de productions. Entre modèle d'agent réactif, cognitif et hybride nous avons employé un modèle d'agent évolutif qui peut être désigné «hybride». Ce dernier, peut basculer d'une stratégie de coopération à une stratégie de compétition durant son exécution pour satisfaire ses intérêts et réaliser ses objectifs.

Le dernier aspect de nos contributions est réservé à l'application et la réalisation de cette thèse. Il concerne le développement d'un simulateur d'un atelier de production conçu selon une approche décentralisée inspirée de l'architecture du projet européen PABADIS. Ce simulateur nous a permis de tester, valider et évaluer les différents concepts de cette thèse.

L'évaluation des différents protocoles de négociation testés a montré que les performances du protocole utilisant l'approche de coopétition, en appliquant les principes de la théorie des jeux au niveau local, sont les meilleurs par rapport aux autres protocoles ; en effet, ce protocole a donné plus de stabilité et de robustesse que les autres protocoles existants dans un environnement dynamique situé. L'allocation dynamique de ressource à travers ce nouveau protocole de négociation, appliquant le principe de la coopétition, peut être considéré comme une bonne approche pour résoudre le problème d'émergence de solutions stables et performantes dans les systèmes de production décentralisés. Nous disposons donc maintenant d'une technologie qui apporte un avantage réel pour la conduite et le pilotage des systèmes complexes.

Finalement, l'application du modèle développé dans cette thèse nécessite des structures décentralisées de pilotage ou des structures hybrides, mais avec un degré d'autonomie locale élevé entre les entités, (par exemple les structures holoniques). De même, il peut être utilisé pour gérer les interactions entre les agents dans d'autres contextes comme le « Grid Computing » et l'E-business.

## 2 Perspectives et travaux futurs à mener

Les perspectives envisagées dans notre travail de recherche peuvent être réparties selon le niveau des concepts et applications.

Au niveau des concepts, des améliorations peuvent être introduites sur les points suivants :

1. Théorie des jeux : l'équilibre de Nash et l'optimum de Pareto ont été les seuls principes de la théorie des jeux utilisés dans cette thèse. L'exploration et l'exploitation d'autres principes d'aide à la décision, comme la valeur de Shapley, ou le compromis de Stek, pourrait également amener à des résultats intéressants.
2. Voisinage d'influence pour la prise de décision : le choix des «Agent-Produit» dans la « Session de Demande » pour la prise de la décision a été limitée à un nombre fixe, égal à trois, et selon un seul critère, lié au temps de traitement permis pour l'exécution de la prochaine tâche. Ce choix a été pris pour permettre une réponse et réaction en temps réel des agents, d'une part, et pour réduire le voisinage d'influence de la prise de décision des agents, d'autre part. Ce choix, qui dépend d'un seul critère, et qui est limité à un nombre fixe d'agents, peut présenter certaines limites dans des environnements, où le jeu de choix, selon plusieurs critères, est essentiel pour la prise de la décision. Pour cela, il serait intéressant de prendre davantage en compte les relations de voisinage d'influence, de dépendance et de la non-linéarité entre les critères pour améliorer les performances des modèles développés.
3. Modèle d'Agent « Global »: Les protocoles de négociation, testés dans cette thèse, à base de la coopération « TJ », le modèle « Currency » et le modèle SSPR ont obtenu de bonnes performances au niveau de paramètres de temps de retard et taux d'utilisation du système. Pour cela, la conception d'un modèle d'agent global intégrant ces différents protocoles peut contribuer à des résultats intéressants concernant l'adaptation des agents dans le système. Ces agents doivent être capables de modifier leurs stratégies, à fur et à mesure, en fonction des cas rencontrés dans le système pour l'amélioration des performances du système et pour le respect de leurs propres intérêts et de l'intérêt global du système.

4. Alternance de stratégies dans un même test : les stratégies utilisées par les agents, à travers des protocoles de négociation pour l'allocation dynamique de ressources, ont été testées et comparées séparément. L'évaluation de ces mêmes stratégies peut être effectuée, d'une autre façon, en mélangeant les stratégies utilisées par la population des agents au cours d'un même test et de la même simulation. Ce procédé peut apporter une nouvelle évaluation intéressante des protocoles et les stratégies développées dans un domaine plus large et ouvert comme l'E-business.
5. Architecture holonique : les différents concepts développés ont été appliqués dans des structures purement hétérarchiques. L'adaptation des modèles conçus pour d'autres structures de pilotage, comme la structure hybride « holonique », peut élargir leur domaine d'application et il peut même contribuer à l'amélioration des performances au niveau local.
6. Synthèse des travaux de recherche : les concepts et les avancées de chacun des travaux effectués dans notre groupe de recherche (DAPS, VFDCS, PABADIS, Thèse-READY) concernent des interactions non-linéaires, séparées, entre les composantes d'un système de production. Ces différentes réalisations et avancées sont à intégrer ensemble pour obtenir un modèle complet et global.

Parmi les domaines d'application des concepts développés, nous pouvons citer :

- E-business et Usine Virtuelle : grâce à l'utilisation de plusieurs concepts et stratégies de négociation provenant de l'économie, comme la Coopétition, les modèles développés peuvent être facilement adaptés et appliqués dans l'E-business pour les mécanismes d'achat et de ventes entre clients et fournisseurs. Le nouveau concept de la Compération, détaillé selon un scénario intéressant dans le chapitre 4 de la thèse, peut être utilisé pour la reconfiguration automatique des demandes des clients sur les ressources dans l'Usine Virtuelle.
- Grille de calcul ou « Computational grid » : l'allocation des tâches de calcul sur une grille « réseau spécifique » de nombre limité de ressources dans un environnement dynamique et ouvert est l'objet de ce nouveau domaine. Le fonctionnement et l'environnement décrit dans ce domaine ressemblent à celui choisi dans cette thèse.
- Projet d'avenir (restaurant ou/et service) : les modèles développés dans notre travail de recherche peuvent être appliqués au niveau de la restauration. Les services fournis par les humains (cuisinier, garçon, etc.), par les machines (four, réfrigérateur, etc.) peuvent être représentés par des « Agent-Ressource » et les clients internes et externes au restaurant par des « Agent-Produit ». La réalisation de ce projet demande l'automatisation de certains services dans la cuisine et le développement d'interface graphique

d'interactions entre le client et le restaurant. L'avantage dégagé dans ce projet peut être résumé à la réactivité et au suivi en temps réel de l'évolution de la demande d'un client et l'exploitation des services multi-média (conseil, traducteur, etc.) fournis par la technologie d'agent.

Nous pouvons néanmoins préciser, de manière plus concrète, que les secteurs les plus en pointe de l'industrie vont bénéficier de telles avancées, dans les domaines de la logistique, des approvisionnements et donc de la configuration des systèmes de production. Certains de ces principes commencent à être appliqués dans les systèmes concurrentiels de l'électronique et de l'informatique. Ils sont sur le point de s'imposer dans la distribution et l'industrie automobile.

Toutefois, même si les technologies liées aux principes décrits dans notre travail, sont amenées à évoluer encore, avant leur intégration dans des outils applicatifs, il est important de noter que les freins restent encore nombreux. En effet, le déploiement d'un système collaboratif implique de remettre à plat l'organisation des entreprises et les méthodes de travail, de façon à bénéficier pleinement des gains en compétitivité espérés. Finalement, toutes ces propositions sont à étudier de manière approfondie et constituent donc des thèmes de recherche porteurs pour des études à venir.

## Référence

- [Achatz, et Parrish, 1987] R.Achatz, et D. J. Parrish, «*Host computer controls FMS at all levels* », *The FMS Magazine* 5, 1, pp. 21-25, 1987.
- [Andersson, 1997] N.Andersson, « *On Modelling and Implementing Shopfloor Control Systems* » PhD thesis, Chalmers University of Technology, ISBN 91-7197-450-4, 1997.
- [Andreu et al., 1995] M.-C., Andreu, M. D. Whinston, J. R. Green. « *Microeconomic Theory* » *Oxford University Press*. 1995.
- [Arentsen, 1995] A.L., Arentsen, « *A generic architecture for Factory Activity Control* », PhD thesis, University of Twente, Enschede, 1995.
- [Baker, 1988] A. D. Baker, « *Complete Manufacturing Control Using a Contract Net: A Simulation Study* » in *1988 International Conference on Computer Integrated Manufacturing*, troy, NY, pp. 100-109, 1988.
- [Baker, 1991] A. D. Baker, « *Manufacturing Control with a Market-Driven Contract Net* », Electrical, Computer and Systems Engineering Department PhD Thesis, Troy, NY: Rensselaer Polytechnic Institute, 1991.
- [Baker et al., 1997] A. D. Baker, H. V. D. Parunak, and K. Krol, « *Manufacturing over the internet and into your living room: perspectives from the AARIA project* », ECECS Dept., University of Cincinnati, Technical Report TR208-08-27, Jan 13, 1997.
- [Bates, 1994] J. Bates., « *The role of emotion in believable agents, Communications of the ACM* », 37(7), 122-125. 1994.
- [Beaufils et al., 1998] B. Beaufils, P. Mathieu, J.-P. Delahaye « *Complete classes of strategies for the classical iterated prisoner's dilemma* » in *Proceedings of Evolutionary Programming VII*, LNCS vol 1447, pp 33-41, Springer-Verlag, 1998
- [Beaufils, 2000] B. Beaufils « *Modèles et simulations informatiques des problèmes de coopération entre agents* », Mémoire de thèse, 2000.
- [Ben-Arieh et Chopra, 1997] Ben-Arieh, D. et Chopra, M.. « *A Game Theoretic Approach to Real-Time Distributed Shop Floor Control* », In *Proceedings of Sixth Industrial Engineering Research Conference*, pages 304-309, 1997.
- [Berrah, 1997] L. Berrah, « *Une approche d'évaluation de la performance industrielle, Modèle d'indicateur et techniques floues pour un pilotage réactif* », Thèse de Doctorat de l'Institut National Polytechnique de Grenoble, 1997.

[Bicchieri et Green, 1997] C. Bicchieri et M.S. Green « Symmetry Arguments for cooperation in the Prisoner's Dilemma », in G. Holmstrom-Hintikka and R. Tuomela (eds.), *Contemporary Action Theory: The Philosophy and Logic of Social Action*, Kluwer, 1997.

[Bicchieri et al., 1998] C. Bicchieri, M. Pollack, C. Rovelli and I. Tsamardinos « *The Potential for the Evolution of Cooperation among Web Agents* », *International Journal of Human-Computer Studies* 48: 9-29, 1998.

[Blackstone et al., 1982] J. J.H. Blackstone, D. T. Phillips and G. L. Hogg, « *A State-of-the-Art Survey of Dispatching Rules for Manufacturing Job Shop Operations* », *International Journal of Production Research*, vol. 20, no. 1, pp. 27-45, 1982.

[Bonabeau et Theraulaz, 1994] E. Bonabeau, G. Theraulaz, « *Intelligence collective* ». Hermès, Paris, 1994.

[Bonabeau et al., 1999] E. Bonabeau, M. Dorigo, G. Theraulaz « *Swarm Intelligence : From Natural to artificial Systems* ». Oxford University Press, 1999.

[Boulet et al., 1991] B. Boulet, B. Chhabra, G. Harhalakis, I. Minis, and J. M. Proth, « *Cell controllers : analysis and comparison of three major projects* », *Computers in Industry*, vol. 16, pp. 239-254, 1991.

[Brassard et Bratley, 1988] G. Brassard et P. Bratley, « *Probabilistic Algorithms* », in *Algorithmics: Theory and Practice*, B. Brassard and P. Bratley, Eds. Englewood Cliffs, NJ: Prentice Hall, pp. 223-276, 341-351, 1988.

[Bratman, 1987] M. E. Bratman. « *Intention, Plans, and Practical Reason* ». Harvard University Press, Cambridge, MA, 1987.

[Bratman et al., 1988] M. E. Bratman, D. Israel, M. E. Pollack. « *Plans and resource-bounded practical reasoning* », *Computational Intelligence* 4 : 349-355. 1988.

[Brandenburger et Nalebuff, 1996] A.M. Brandenburger et B.J. Nalebuff. « *Co-opetition* ». Doubleday, Bantam Doubleday Dell Publishing Group. 1996.

[Broissin, 1999] N. Broissin, « *Contribution à l'amélioration de la réactivité des systèmes de production automatisés et flexibles grâce à un pilotage basé sur une génération de tâches décentralisée* », Thèse de Doctorat en Sciences, Université d'Aix-Marseille III, 1999.

[Brooks, 1986] R. A. Brooks. « *A robust layered control system for a mobile robot* ». *IEEE Journal of Robotics and Automation*, RA-2(1) :14-23, 1986.

[Brooks, 1991] R. A. Brooks. « *Intelligence without representation* ». *Artificial Intelligence* , 47(1-3) :139-159, 1991.

[Brussel et al., 1995] H.V. Brussel, P. Valckenaers, L. Bongaerts, J. Wyns, « *Architectural and System Design issues in Holonic Manufacturing Systems* », Preprints of the 3<sup>rd</sup> IFAC Workshop on Intelligent Manufacturing Systems (IMS'95), Bucarest, 1995.

[Brussel et al., 1998] H. V. Brussel, J. Wyns, P. Valckernaers, L. Bongaerts, « *Reference architecture for holonic manufacturing systems: PROSA* », *Computers in Industry*, vol. 37, pp.255-274, 1998.

[Burke et Prosser, 1994] P.Burke et P. Prosser. « *The Distributed Asynchronous Scheduler* ». Intelligent Scheduling, Zweben, M. and Fox, M.S., eds., Morgan Kaufman Publishers, San Francisco, CA, pp. 309-339, 1994.

[Brucker, 1995] Brucker, P., « *Scheduling Algorithms* », Springer, Berlin. 1995.

[Burmeister et al., 1997] B. Burmeister, A. Haddadi, G. Matyilis. « *Application of multi-agent systems in traffic and transportation* ». In IEE Proceedings of Software Engineering, pages 51—60, 1997.

[Butler et Ohtsubo, 1992] J.Butler et H.Ohtsubo. ADDYMS: « *Architecture for Distributed Dynamic Manufacturing Scheduling* ». Artificial Intelligence Applications in Manufacturing, Famili, A., Nau, D.S., and Kim, S.H., (eds.), The AAAI Press, pp. 199-214, 1992.

[Casteran et al., 2000] J.C. Casteran, M.P.Gleize, P.Glize, « *Des méthodologies orientées multi-agent* », *Actes des Journées Francophones sur les l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents, JFIADSMA'00* , Hermes, ISBN 2-7462-0176-3, p. 191-207, 2000.

[Chaib-draa et al., 2001] B.Chaib-draa, B. Moulin, I. Jarras. « *Agent et Systèmes Multiagents* », In *Principes et architecture des systèmes multi-agents*, JP. Briot et Y Demazeau , éditions Hermes, Lavoisier, 2001.

[Chan et al., 1999] C.Y. Chan, F.W. Lam et C.P. Lee, Considerations for using cellular manufacturing, *Journal of Materials processing technology*, vol. 96, pp. 182-187. 1999.

[Chavez et Kasbah, 1996] A. Chavez et P. M.. Kasbah: « *An agent marketplace for buying and selling goods* ». In *Proceedings of the first international Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, London, U.K., 1996.

[Cerri, 1997] S. A. Cerri . « *A simple language for generic dialogues: "Speech acts" for communication* », in *Journées Francophones des Langages Applicatifs - JFLA97*, M. Gengler and C. Queindec, Eds. Dolomieu, Isère, France: INRIA, pp. 145-168, 1997.

- [Cerri, 1996] S.A. Cerri. Cognitive environments in the STROBE model. Presented at *EuroAIED: the European Conference in Artificial Intelligence and Education*, Lisbon, Portugal, 1996.
- [Cho, 1993] H.Cho. «*Intelligent Workstation Controller for Computer Integrated Manufacturing* », Industrial Engineering, PhD thesis, Texas A&M University, 1993.
- [Chryssolouris et al., 1984] G. Chryssolouris, K. Wright, J. Pierce, W. Cobb, « Decision Making on the Factory Floor: An Integrated Approach to Process Planning and Scheduling », *Robotics and Computer-Integrated manufacturing*, vol. 1, no. 3/4, pp. 315-319, 1984.
- [Chryssolouris et al., 1988] G. Chryssolouris, K. Wright, J. Pierce, W. Cobb, “Manufacturing Systems Operation: Dispatch Rules Versus Intelligent Control,” *Robotics and Computer-Integrated manufacturing*, vol. 4, no. 3/4, pp. 531-544, 1988.
- [Cicirello, 2001] V.A. Cicirello. «*A Game-Theoretic Analysis of Multi-Agent Systems for Shop Floor Routing* ». tech. report CMU-RI-TR-01-28, Robotics Institute, Carnegie Mellon University, 2001.
- [Cicirello et Smith, 2001] V. A. Cicirello et S. F. Smith. «*Insect societies and manufacturing* ». In *The IJCAI-01 Workshop on Artificial Intelligence and Manufacturing, Working Notes*, pages 33-38. AAAI SIGMAN, 4-9 August 2001.
- [Dastani et al., 2000] M. Dastani, J. Hulstijn et L.V. der Torre. « Negotiation protocols and dialogue games ». In *Proceedings of the BNAIC*, 2000.
- [Diep et al.] D. Diep, P. Massotte, J. Reaidy, Y. Liu : «*Design and Integration of Intelligent Agents to Implement Sustainable Production Systems* » EcoDesign 2001, Second International Symposium on Environmentally Conscious Design and Inverse Manufacturing, IEEE Computer Society, Tokyo, JAPAN, 2001.
- [Dilts et al., 1991] D.M. Dilts, N.P. Boyd, et H.H. Whorms, «*The evolution of control architectures for automated manufacturing systems* », *J. Mfg. Sys.*, vol. 10, no. 1, pp.79-93, 1991.
- [Drogoul et Meyer, 1999] A. Drogoul, J. Meyer, «*Intelligence artificielle située*». Hermès Paris Science Publications, 1999.
- [Duffie et Piper, 1987] N. A. Duffie et R. S. Piper, «*Non-hierarchical control of a flexible manufacturing cell* », *Robotics & Computer-Integrated Manufacturing*, vol. 3, no. 2, pp.175-179, 1987.
- [Duffie et Prabhu, 1994] N. A. Duffie et V. V. Prabhu, «*Real-time distributed scheduling of heterarchical manufacturing systems* », *Journal of Manufacturing Systems*, vol. 13, no. 2, pp. 94 –107, 1994.

[Duffie et Prabhu, 1996] N. A. Duffie et V. V. Prabhu., «*Heterarchical control of highly distributed manufacturing systems*», *International Journal of Computer Integrated Manufacturing*, vol. 9, no. 4, pp. 270-281, 1996.

[Durand et al., 2001] P.Y. Durand, J. Chanliau et A. Mariot, M. Kessler, J.P. Thomesse, L. Romary, F. Charpillat, R. Hervy. «*Telemedicine and dialysis*». In 3rd International Workshop on Enterprise Networking and Computing in Health care Industry - HealthCom'2001. Italy. 2001.

[Ehtama, 1999] H. Ehtama., “*Generating Pareto Solutions in a Two-Party Set-ting: Constraint Proposal Methods*”. *Management Science*. 1999.

[Ferber, 1995] J. Ferber, *Les systèmes multiagents vers une intelligence collective*, InterEdition, Paris, France, 1995.

[Ferguson, 1992] I. A. Ferguson. «*TouringMachines : An Architecture for Dynamic, Rational, Mobile Agents*». PhD thesis, Clare Hall, University of Cambridge, UK, November 1992.

[Fikes et Nilsson, 1971] R.E. Fikes et N.J. Nilsson, «*STRIPS: a new approach to the, application of theorem proving to problem solving* ». *Artificial Intelligence*, 2 (3-4):189--208. 1971.

[Finin et Labrou, 1999] T. Finin et Y. Labrou. «*Tutorial on Agent Communication Languages* », First International Symposium on Agent Systems and Applications and the Third International Symposium on Mobile Agents, ASA/MA'99 , 1999.

[FIPA, 2000] FIPA, Foundation for Intelligent Physical Agent. «*FIPA Contract Net Interaction Protocol Library Specification* ». <http://www.fipa.org/specs/fipa00029/>, 2000.

[Fischer et al., 1999] K. Fischer, B. Chaib-draa, H. J. Müller, J. P. Müller, M.Pischel. «*A simulation approach based on negotiation and cooperation between agents*». *IEEE Trans. on Systems, Man, and Cybernetics*, 29(4), pp. 531-545, 1999.

[Flik, 1995] M.I, Flik. «*Optimization of Products Variants – Lean Standardization at a Radiator Supplier* ». Internal Report , Behr Automotive.1995

[Gerding, 1999] E. Gerding, “*Survival of the Richest: An Evolutionary model for Multi-Issue Negotiations*”. Master’s thesis, Free University Amsterdam (VU), Division Mathematics and Computer Science. 1999.

[Gnyawali et Madhavan, 2001] D.R. Gnyawali et R. Madhavan. «*Cooperative networks and competitive dynamics:A structural embeddedness perspective*». *Academy of Management Review*, 26(3): 431-445, 2001.

[Goldberg, 1994] D. E. Goldberg, «*Genetic and Evolutionary Algorithms Come of Age*», *Communications of the ACM*, vol.37, no. 3, pp. 113-119,1994.

[Grand et Cliff, 1998] S. Grand, D. Cliff : «*Creatures: Entertainment Software Agents with Artificial Life*». *Autonomous Agents and Multi-Agent Systems* 1 (1): 39-57, 1998.

[Grasshopper, 2002] Grasshopper. Web site, <http://www.grasshopper.de/>.

[Hatvany, 1985] J. Hatvany. «*Intelligence and Cooperation in Heterarchic Manufac-turing Systems*», *Manufacturing Systems, Proceedings of the CIRP Seminars*, Vol. 14, No. 1, 1985.

[Hammer, 1987] H. Hammer, «*Flexible manufacturing cells and systems with computer intelligence*», *Robotics and Computer Integrated Manufacturing* 3, 39-54,1987.

[Holthaus, 1997] O. Holthaus. «*Design of Efficient Job Shop Scheduling Rules*». *Computers & Industrial Engineering*, Vol. 33, No. 1-2, pp. 249 - 252. 1997.

[Huang et al., 1995] J. Huang, N., R. Jennings, J. Fox : «*Agent-based approach to health care management* ». *Applied Artificial Intelligence* 9 (4): 401-420, 1995.

[IMS, 1994] IMS : «*Intelligent Manufacturing Systems. A Program for International Cooperation in Advanced Manufacturing*», Final Report of the International Steering Committee, Hawaii, 1994.

[IMS, 1997] IMS'97, 4<sup>th</sup> IFAC Workshop on Intelligent Manufacturing Systems, Seoul, Korea Proceedings edited by Jongwon Kim, Seoul National University, PERGAMON, ISBN 0-08-043025-2, 1997.

[IOS, 1986] International Organization for Standards, The Ottawa Report on Reference Models for Manufacturing Standards, ISO TC184/SC5/WG1 N51, Version 1.1, 1986.

[Jackson et Jones, 1987] R. H. F. Jackson et A.T. Jones, «*An architecture for decision making in the factory of the future*» *Interfaces*, vol. 17, no. 6, pp. 15-28, 1987.

[Jennings, 1996] N.R. Jennings, «*Applying Agent Technology*», *Proceedings of the Forth International Conference on the Practical Application of Prolog/Constraints/Intelligent Agents and Multi-Agent Technology, PAAM'96* , 1996.

[Jennings et al., 1998] N. R. Jennings, K. Sycara, M. Wooldridge. «*A Roadmap of Agent Research and Development*» *Int Journal of Autonomous Agents and Multi-Agent Systems* 1 (1) 7-38, 1998.

[Jennings, 1999] Jennings N. R., « Agent-Oriented Software Engineering », *Multi-Agent System Engineering, 9<sup>o</sup> European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW'99*, F.J. Garijo and M. Boman, Springer Verlag, LNAI 1647, 1999.

[Jini, 1999] Jini, *Jini<sup>™</sup> Technology Architecture Overview*. Sun Microsystem Inc, available at <http://www.sun.com/jini/whitepaper/>.

[Johnson et al., 1991] D. S. Johnson, C. R. Aragon, L. A. McGeoch, « *Optimization by Simulated Annealing: An Experimental Evaluation* », Part II (Graph Coloring and Number Partitioning), *Operations Research*, vol. 39, no. 3, pp. 378-406, 1991.

[Jones et McLean, 1986] A. T. Jones et C. R. McLean, « A proposed hierarchical control model for automated manufacturing systems » *Journal of Manufacturing Systems*, vol. 5, no. 1, pp. 15-25, 1986.

[Kahneman et Tversky, 1979] D. Kahneman et A. Tversky. « *Prospect theory: An analysis of decision under risk* ». *Econometrica*, 47(2):263--291. 1979.

[Kim, 2002] B.I. Kim, « *Intelligent Agent Based Planning, Scheduling and Control: Warehouse Management Application* », PhD thesis, Rensselaer Polytechnic Institute, Troy, New York, 2002.

[Kondoh et al., 1999] S. Kondoh, R. K. Sato, Y.Umeda and T. Tomiyama : « *Self-organization of the Cellular Manufacturing System* ». The 32nd CIRP International Seminar on Manufacturing Systems, pp. 595-603, 1999.

[Kondoh et al., 2000] S. Kondoh, Y.Umeda, T. Tomiyama : Morphological Design of a Manufacturing System based on the Concept of Cellular Machines. The 33rd CIRP International Seminar on Manufacturing Systems, pp. 66-70, 2000.

[Koestler, 1967] Koestler, A.. « *The Ghost in the Machine* ». Arkana Books, London, UK. 1967.

[Kouiss et al., 2002] K. Kouiss, Éric Gouardères, Pierre Massotte, « Organisations distribuées des systèmes de pilotage », dans « *Fondements du pilotage des systèmes de production* », Hermes, Lavosier, 2002.

[Kraus, 2001] S. Kraus, "*Strategic Negotiation in Multiagent Environment*", ISBN: 0-262-11264-7, the MIT Press, 2001.

[Krothapalli et Deshmukh, 1997] N. Krothapalli, N. et A. Deshmukh, « *Effects of Negotiation Mechanisms on Per-formance of Agent-Based Manufacturing Systems* », Flexible Automation and Integrated Manufacturing, 1997.

- [Krothapalli et Deshmukh, 1998] N. Krothapalli, N. et A. Deshmukh. « Self-Regulating Negotiating Schemes for Robust Agent-Based Manufacturing Systems », 7th Industrial Engineering Research Conference, Banff, Alberta, Canada, May, 1998.
- [Krothapalli et Deshmukh, 1999] N. Krothapalli, N. et A. Deshmukh. « *Design of negotiation protocols for multi-agent manufacturing systems* », *International Journal of Production Research*, vol. 37, no. 7, pp. 1601-1624, 1999.
- [Kusiak and Chen, 1988] A. Kusiak and M. Chen, « *Expert Systems for Planning and Scheduling Manufacturing Systems* », *European Journal of Operational Research*, vol. 34, pp. 113-130, 1988.
- [Kurihara et al., 1996] T. Kuriharara, P. Bunce, J. Jordan, « *Next Generation Manufacturing Systems (NGMS) in IMS Program* », Actes de la 2<sup>ème</sup> Conférence Internationale DIISM (The Design of Information Infrastructure Systems for Manufacturing), Kaatsheuvel, 1996.
- [Lado et al., 1997] A.A.Lado, N.Boyd, S.C. Hanlon. « *Competition, cooperation, and the search for economic rents: A syncretic model* ». *Academy of Management Review*, 22(1): 110-141, 1997.
- [Landau, 1996] I. D. Landau, « *Identification et commande des systèmes* », Hermès, 2<sup>ème</sup> éd., Paris, 1996.
- [Lemoigne, 1994] J.L. Lemoigne, « *La théorie du système général- théorie de la modélisation* », éd. Presses universitaires de France, 1994.
- [Lee, 1996] J.Lee. « *Overview of Manufacturing Strategy, Production Practices, Emerging Technologies, and Education System in Japan* ». NSF/STA Study Report. 1996.
- [Lenstra et Kan, 1978] J. K. Lenstra et A. H. G. R. Kan, « *Complexity of Scheduling Under Precedence Constraints* », *Operations Research*, vol. 26, no. 2, pp. 22-35, 1978.
- [Levin et Moore] J. Levin, et J. Moore,.. « *Dialogue-games: meta-communication structure for natural language interaction* ». *Cognitive science*, 1(4):395-420. 1980.
- [Lin et Solberg, 1991] G.Y. Lin et J.J. Solberg, « *Effectiveness of flexible routing control* » *Int. J. Flex. Mfg. Sys.*, vol. 3, pp.189-211, 1991.
- [Lin et Solberg, 1992] Lin, G., Y-J., et Solberg, J. J., « *Integrated shop floor control using autonomous agents* », *IIE Transactions* 24, 3, 57-71.1992.
- [Linstone et Turoff, 1975] H.A.Linstone et M. Turoff. « *The Delphi Method: Techniques and Applications* ». Reading, Mass.: Addison-Wesley, 1975.

[Liu et Massotte, 1999] Y. J. Liu et P. Massotte. “*Self-adaptation and Reconfiguration of an Agent-Based Production System: Virtual Factory*” IAT’99: *Asia Pacific Conference on Intelligent Agent Technology*, Honk-Hong, Chine, 1999.

[Liu, 2002] Y.J. Liu, « *De la nécessité et de la façon de coopérer, de s’auto-organiser et de se reconfigurer dans des systèmes de production complexes : modélisation et gestion d’un système virtuel pour une approche multi-agents* », Thèse de Doctorat en Génie Industriel, INPG, France, 2002.

[Lüder et al., 2002] A. Lüder, J. Peschke, T. Sauter, S. Deter, D. Diep, “Distributed intelligence for plant automation based on multi-agent systems – the PABADIS approach”, *Journal on Production Planning and Control*, à paraître, 2002.

[MacFarland et Grant, 1987] D. MacFarland et H. Grant, « *Shop Floor Scheduling and Control Using Simulation Technology* », in *Shop Control’ 87 Conference Proceedings*, Cincinnati, OH, pp. 12-1, 1987.

[Massotte, 1995] P.Massotte, chapitre 12, "La modélisation systématique en entreprise", editeurs A.Haurat et C.Braesch, Hermes, Paris, 1995.

[Massotte, 1999] P. Massotte, « *Auto-organisation dans les structures et les systèmes complexes* », Conférence MOSIM’99, Modélisation et simulation des flux physiques et informationnels, G. Habchi & A. Haurat (eds), Annecy, octobre 1999, SCS International, Ghent, Belgique pp. 21-29, 1999

[Massotte et al., 2001a] P. Massotte, Y. Liu, J. Reaidy : "DAPS : Dynamic adaptation of complex production systems" ESS’01 - 13th european simulation symposium in simulation industry, Marseille, France, octobre 2001.

[Massotte et al., 2001b] P. Massotte, J. Reaidy, Y. Liu, D. Diep, T. Tomiyama, S. Kondoh : "*Evolution of management approaches devoted to the control of complex and distributed production systems*", rapport technique, 2001<sup>2</sup>.

[Massotte et al., 2001c] P. Massotte, P. Couturier, Y. Liu, J. Reaidy : "*Business network : rethinking "Supply chain management systems of virtual factories"*" IEPM01, Québec, Canada, 2001.

[Massotte et al., 2001d] P. Massotte, J. Reaidy, D. Diep, Y. Liu : "*Implementation of New Approaches Based on Intelligent Agents for the New Economy*" IEPM01, Québec, Canada, 2001.

---

<sup>2</sup> Ce rapport a été rédigé sur le Soft Artifact dans le cadre d’un projet de coopération entre le centre de recherche LGI2P et le Laboratoire RACE de l’Université de Tokyo.

[Massotte et al., 2002] P. Massotte, J. Reaidy, Y. Liu, D. Diep : "*Intelligent Agents for Production Systems*" Journal Européen des Systèmes Automatisés (JESA), 2002 (à paraître).

[Maturana et Norrie, 1996] F. Maturana, et D. Norrie, « *Multi-Agent Mediator Architecture for Distributed manufacturing* ». Journal of Intelligent Manufacturing, 7, 257-270. 1996.

[Maturana, 1997] F. Maturana, "*MetaMorph: an adaptive multi-agent architecture for advanced manufacturing systems,*" PhD thesis, University of Calgary, Canada, 1997.

[Maturana et al., 1999] F. Maturana, W. Shen, et D. H. Norrie, « *MetaMorph: an adaptive agent-based architecture for intelligent manufacturing* », *International Journal of Production Research*, vol. 37, no. 10, 2159-2173, 1999.

[McEleney et al., 1998] B. McEleney, G.M.P. O'Hare, et J. Sampson. « *An Agent Based System for Reducing Changeover Delays in a Job-Shop Factory Environment* ». In Proc. Of PAAM'98, London, 1998.

[Middelkoop et al., 1997] T. Middelkoop, N. Krothapalli, A. Deshmukh, et J. Solberg, « *Rapid Deployment of Agents to Facilitate Collaborative Engineering* », International CIRP Design Seminar - Multimedia and Virtual Reality Technologies for Collaborative Design and Manufacturing, pp. 208–213, 1997.

[Mintzberg, 1982] H. Mintzberg, « *Structure et dynamique des organisations, éd. Organisation* », Paris, 1982.

[Moulin et Chaib-draa, 1996] B. Moulin, et B. Chaib-draa, "A Review of Distributed Artificial Intelligence", in *Foundations of Distributed Artificial Intelligence*, O'Hare, G. and Jennings, N. (eds), Wiley, pp. 3-55, 1996.

[Müller et Pischel, 1994] J.P. Müller et M. Pischel. « *Modelling interacting agents in dynamic environments* ». In *Proceedings of the Eleventh European Conference on Artificial Intelligence (ECAI-94)*, Amsterdam, The Netherlands, pp. 709-713, 1994.

[NRC, 1998] National Research Council, « *Visionary Manufacturing Challenges for 2020* », Washington, DC: National Academy Press, 1998.

[Osborne et Rubinstein, 1994] M. J. Osborne et A. Rubinstein. « *A course in game theory* ». MIT Press, 1994.

[Ounnar, 1999] F. Ounnar, « *Prise en compte des aspects décision dans la modélisation par réseaux de Petri des systèmes flexibles de production* », Thèse de Doctorat de l'Institut National Polytechnique de Grenoble, 1999.

[Ottaway et Burns, 1997] T. A. Ottaway and J. R. Burns, « Adaptive, agile approaches to organizational architecture utilizing agent technology », *Decision Sciences*, vol. 28, no. 3, pp. 483-511, 1997.

[Ottaway et Burns, 2000] T. A. Ottaway and J. R. Burns, “An adaptive production control system utilizing agent technology,” *International Journal of Production Research*, vol. 38, no.4, 721-737, 2000.

[PABADIS, 1999] PABADIS, « Plant Automation Based on Distributed System », European Project : IMS IST-1999-60016-PABADIS, web site, <http://www.pabadis.org>.

[PABADIS, 2002a] PABADIS, « *Platform Layer and Host, Specification and Design* » Work Package 1, Task 1.3. 2002.

[PABADIS, 2002b] PABADIS, « *Development of the multiple agent system* ». Work Package 2, Task 2.2. 2002.

[PABADIS, 2002c] PABADIS, « *ERP – Interface and Agent Fabricator* », Work Package 4, Task 4.3. 2002.

[PABADIS, 2002d] PABADIS, « *Revolutionising Plant Automation, The PABADIS Approach* », Work Package 6, Task 6.3. 2002.

[Panwalkar et Iskander, 1977] S. S. Panwalkar et W. Iskander, « A Survey of Scheduling Rules », *Operations Research*, vol. 25, no. 1, pp. 45-61, 1977.

[Parunak, 1987] V.D. Parunak. « *Manufacturing Experience with the Contract Net. Distributed Artificial Intelligence* », Huhns, M.N. ed., Pitman, pp. 285-310. 1987.

[Parunak et al., 1998] H. V. D. Parunak, A. D. Baker, and S. J. Clark, “*The AARIA agent architecture: from manufacturing requirements to agent-based system design*,” the workshop on Agent-Based Manufacturing, ICAA '98, Minneapolis, MN, 1998.

[Parunak, 1998a] H.V.D Parunak., "Practical and Industrial Applications of Agent-Based Systems", Industrial Technology Institut publications, Environmental Research Institute of Michigan (ERIM) ,1998.

[Parunak et al., 2001] H. V. D. Parunak, , A. D. Baker, S. J. Clark. *The AARIA Agent Architecture: From Manufacturing Requirements to Agent-Based System Design*. Integrated Computer-Aided Engineering, Vol 8, No. 1. 2001.

[Parunak, 1998b] H.V.D Parunak., "What can Agents do in Industry, and Why?" An Overview of Industrially-Oriented R & D at CEC, *Proceedings of the Second International Workshop on Cooperative Information Agents* , CIA'98 , M. Klusch, G.

Weiß (Ed.), Springer-Verlag, Lecture Notes in Computer Science, Vol. 1435, ISBN 3540646760, p. 1-18, 1998.

[Pels et al., 1997] H. J. Pels, J.C. Wortmann, A. J.R. Zwegers, “*Flexibility in manufacturing: an architectural point of view,*” *Computers in Industry*, vol. 33, pp. 271-283, 1997.

[Pesenti et al., 2001] R. Pesenti, L., Castelli, et P. Santin. «*Scheduling in a Realistic Environment Using Autonomous Agents: A Simulation Study* ». Proceedings of Agent Based Simulation II Workshop, Passau, Germany, pp.149-154, 2001.

[Preist, 1998] C. Preist, “*Economic Agents for Automated Trading*”. Hewlett-Packard Labs, Bristol, UK.1998.

[Pujo et Kieffer, 2002] P. Pujo et J.-P. Kieffer, « *Concepts fondamentaux du pilotage des systèmes de production* », dans « *Fondements du pilotage des systèmes de production* », Hermes, Lavosier, 2002.

[Qiao et Zhu, 2000] B.Qiao et J. Zhu, Agent-Based Intelligent Manufacturing System for the 21<sup>st</sup> Century, Global Dialogue Expo 2000, Hannover, Germany, 2000.

[Rabelo et Alptekin, 1989] L. Rabelo et S. Alptekin, «Synergy of Neural Networks and Expert Systems for FMS Scheduling », in *Third ORSA/TIMS Conference on Flexible Manufacturing Systems*, K. E. Stecke and R. Suri, Eds., Cambridge, MA, pp. 361-366, 1989.

[Reaidy, 1999] J. Reaidy, «*Construction évolutive de connaissances du contrôle* », mémoire de DEA, ULP-Strasbourg, 1999.

[Reaidy et al., 2001] J. Reaidy, P. Massotte, Y. Liu, D. Diep : "*Product and process reconfiguration based on intelligent Agents*" TESADI'2001 - IEEE SMC 2001, Tucson, USA, 2001.

[Reaidy et al., 2002a] J. Reaidy, P. Massotte, D. Diep : "*Cooperation through Consensual Decision-Making in Multi-Agent Manufacturing Systems*" SCI 2002, 6th World Multi Conference On Systemics, Cybernetics And Informatics, Orlando , Florida, USA, 2002.

[Reaidy et al., 2002b] J. Reaidy, P. Massotte, D. Diep: "*Negotiation and Decision-Making in Agent-Based Manufacturing Systems*" KI2002, 25<sup>th</sup> German Conference on Artificial Intelligence, Workshop on Cognitive Agents, Aachen, Germany, 2002.

[Reaidy et al., 2003a] J. Reaidy, P. Massotte, D. Diep: "*Comparison of Negotiation Protocols in Dynamic Agent-Based Manufacturing Systems*" IEPM 2003, Porto, Portugal, 2003.

[Reaidy et al., 2003b] J. Reaidy, P. Massotte, D. Diep: "Co-opetition through Game Theory Principles and Agents Based Information Systems" INDIN 2003, 1<sup>st</sup> IEEE International Conference on Industrial Informatics, Alberta, Canada, 2003 (article soumis).

[Rao et Georgeff, 1995] A.S.Rao et M.P. Georgeff, « *BDI Agents : From theory to practice* », Proceedings of the first International Conference on Multi-Agent Systems, ICMAS'95, p. 312-319, 1995.

[Rosenschein et Genesereth, 1985] J. S. Rosenschein et M. R Genesereth. « *Deals among rational agents* », Proceedings of 9<sup>th</sup> International Joint Conference on Artificial Intelligence, Los Angeles, CA, pp. 91-99, 1985.

[Rosenschein et Zlotkin, 1994] J. Rosenschein and G. Zlotkin. « *Rules of Encounter* ». The MIT Press, Cambridge, Massachusetts, 1994.

[Saad et al., 1997] A.Saad, K. Kawamura, G. Biswas, « *Performance evaluation of contract net-based heterarchical scheduling for flexible manufacturing systems* ». Intelligent Automation and Soft Computing, 3(3), 229-248. 1997.

[Saporta, 1990] Saporta G., « *Probabilités et analyse des données et statistiques* », edition technip, paris, 1990.

[Schoonderwoerd, 1997] R. Schoonderwoerd, O. Holland, J. Bruten: « *Ant-Like Agents for Load Balancing in Telecommunications Networks* ». Agent 1997: 209-216.1997.

[Senehi et al., 1994] M.K.Senehi, T.R. Kramer, S.R. Ray, R. Quintero, J.S. Albus, « *Hierarchical control architectures from shop level to end effectors, Computer control of flexible manufacturing systems* », Eds. S.B. Joshi and J.S. Smith, Chapman & Hall, 1994.

[Shaw et Whinston, 1983] M.J. Shaw et A.B. Whinston, « *Distributed Planning in Cellular Flexible Manufacturing Systems* » Tech. Report, Management Information Research Center, Purdue University. 1983.

[Shaw, 1987] M. J. Shaw, "A distributed scheduling method for computer integrated manufacturing: the use of local area networks in cellular systems," *International Journal of Production Research*, vol. 25, no. 9, pp. 1285-1303, 1987.

[Shen et Norrie, 1998] W.Shen et D.H. Norrie. « *A Hybrid Agent-Oriented Infrastructure for Modeling Manufacturing Enterprises* ». In Proceedings of KAW'98, Banff, Canada. 1998.

- [Shen et Norrie, 1999] W. Shen et D.H. Norrie. « *Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey* ». Knowledge and Information Systems, an International Journal, 1(2), pp.129-156.1999.
- [Shen et al., 1998] W. Shen, D. Xue, and D. H. Norrie, "An agent-based manufacturing enterprise infrastructure for distributed integrated intelligent manufacturing systems," in *Proc. PAAM'98*, London, UK, pp. 533-548. 1998.
- [Sierra et al., 1997] C. Sierra, P. Faratin, N. R. Jennings . "A Service -Oriented Negotiation Model between Autonomous Agents". *MAMMAW'97*, Ronneby, Sweden.1997.
- [Shingo, 1989] S. Shingo, « *A Study of the Toyota Production System from and Industrial Engineering Viewpoint* », Revised Edition. Cambridge, MA: Productivity Press, 1989.
- [Smith, 1980] R.G. Smith, « *The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver* », *IEEE Transactions on Computers*, Vol. C-29, No. 12, 1980.
- [Smith et Davis, 1981] R.G. Smith, R. Davis, « *Frameworks for Cooperation in Distributed Problem Solving* », *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-11, No. 1, 1981.
- [Sohier, 1996] C. Sohier, "Pilotage des Cellules Adaptatives de Production : Apport des Systèmes Multi-Agents" Thèse de Doctorat de l'Ecole Normale Supérieure de Cachan, 1996.
- [Sycara, 1988] K.P. Sycara, "Utility theory in conflict resolution". *Annals of Operations Research* 12: 65-84. 1988.
- [Sycara, 1990] K.P. Sycara, "Negotiation planning: An AI approach". *European Journal of Operational Research* 46: 216-234.1990.
- [Sycara et al., 1991] K. P. Sycara, S. F. Roth, N. Sadeh, « *Resource Allocation in Distributed Factory Scheduling* », *IEEE Expert*, vol. 6, no. 1, pp. 29-40, 1991.
- [Tawegoum et al., 1994] R. Tawegoum, E. Castelain, and J.C. Gentina, "Hierarchical and dynamic production control in flexible manufacturing systems," *Robotics & Computer-Integrated Manufacturing*, vol. 11, no. 4, pp. 327-334, 1994.
- [Teich et al, 1998] J.Teich, H. Wallenius, J. Wallenius. "World Wide Web Technology in Support of Negotiation and Communication". Interim report, IIASA, Laxenburg, Austria. 1998.

[Teti et Kumara, 1997] R. Teti et S.R. Kumara, Intelligent Computing Methods for Manufacturing Systems, Annals of the CIRP, Vol. 46/2. 1997.

[Titly, 1979] A.Titly, Analyse et Commande des Systèmes Complexes, Cépaduès, 1979.

[Tomiyama, 1997] T. Tomiyama, "A Manufacturing Paradigm Toward the 21st Century, *Integrated Computer Aided Engineering*", Vol. 4, n° 3, pp. 159-178. 1997.

[Trentesaux et al., 2000] D. Trentesaux, J. P. Campagne et J. Erschler, "A generic design framework for decentralized control: the DMU model", *2nd IFAC/IFIP/IEEE Conf. On Management and Control of Production and Logistics, MCPL 2000* (Grenoble, juillet 2000), Z. Binder (ed), Pergamon, ISBN 0-08-043621 8, 1<sup>st</sup> edition, Oxford, Elsevier Science Ltd., vol. 3, 2001, pp. 1021-1026. 2000.

[Trentesaux, 2002] D. Trentesaux, « *Pilotage hétérarchique des systèmes de production* » (HDR), VHC, Valenciennes, 2002.

[Usher et Wang, 2000a] J.M. Usher et Y-C. Wang, *Judging the Value of Additional Information on the Performance of Intelligent Agents in Manufacturing Control*. Conference on Intelligent Systems in Design and Manufacturing II, SPIE, Boston, MA. 2000.

[Usher et Wang, 2000b] J.M. Usher et Y-C. Wang, *Intelligent Agents Architectures for Manufacturing Control*. Proc. of the 9<sup>th</sup> Industrial Engineering Research Conference, Cleveland, Ohio.2000.

[Usher, 2001] J.M. Usher, "Collaborative Agent Performance in Job-Shop Routing," 5th International Engineering Design and Automation Conference, Las Vegas, Nevada, 2001.

[Valckenaers et al., 1994a] P. Valckenaers, H.V. Brussel, F. Bonneville, L. Bongaerts, J. Wyns, « *IMS Test Case 5 : Holonic Manufacturing Systems* », Preprints of IMS'94, IFAC Workshop, Vienne, 13-15 juin 1994.

[Valckenaers et al., 1994b] P. Valckenaers, H.V. Brussel, F. Bonneville, L. Bongaerts, J. Wyns, « *Results of the Holonic Control System Benchmark at KU Leuven* », 4<sup>th</sup> International Conference on Computer Integrated Manufacturing and Automation Technology (CIMAT), New York, 1994.

[Valckenaers et al., 1999] Valckenaers P., Heikkilä T., Baumgaertel H., McFarlane D., Courtois J.P., "Towards a novel manufacturing control principle", *Proceedings of the Second International Workshop on Intelligent Manufacturing Systems*, H. Van Brussel and P. Valckenaers (Ed.), Katholieke Universiteit Leuven, ISBN 90-73802-69-5, p. 871-875, 1999.

[Wisner, 1993] J. D. Wisner, «*How Does Your Shop Stack Up?* », *Modern Machine Shop*, vol. 66, no. 2, September 1993, pp. 86-90.

[Wooldridge et Jennings, 1995] M. Wooldridge et N. R. Jennings. «*Intelligent agents: Theory and practice* ». *The Knowledge Engineering Review*, vol. 10(2) pp. 115–152, 1995.

[Wright, 1983] A. B. Wright, «*Basic Scheduling Techniques* », in *American Production & Inventory Control Society 1983 Conference Proceedings*, pp. 200-204. 1983.

[Wyns, 1999] J. Wyns, «*Reference architecture for holonic manufacturing systems: the key to support evolution and reconfiguration*,“ PhD thesis, Katholieke Universiteit Leuven, Leuven, 1999.

[Zeghal, 1992] K. Zeghal. «*Un modèle de coordination d'actions réactive appliqué au trafic aérien* ». JFIADSMA'93 : Actes des 1<sup>ère</sup> Journée Francophones d'Intelligence Artificielle et Systèmes Multi-Agents, 1992.

[Zweben et Fox, 1994] M. Zweben et M.S. Fox, M.S. «*Intelligent Scheduling* ». Morgan Kaufman Publishers, San Francisco, CA. 1994.

## Glossaire

PN :	Protocole de Négociations
PR :	Production Reservation
SSPR :	Single-Step Production Reservation
SD :	Session de Demande
SMA :	Système Multi-Agent
AP :	Agent-Produit
AR :	Agent-Ressource
ASD :	Agent de la Session de Demande
APZ :	Agent-Produit déclenchant le moment de la décision dans la Session de Demande.
LUS :	LookUp Service
CMU :	Co-operative Manufacturing Units
O-P :	Orienté-Produit
OR :	Orienté-Ressource
BD :	Bi-Directionnelle
FIFO :	First-In-First-Out
FCFS :	First-Come-First-Serve
SPT :	Shortest Processing Time
DD :	Due Date
TF :	Temps de Fabrication
TA :	Temps d'Attente
TR :	Temps de Retard
TUS :	Taux d'Utilisation du Système
LFA :	Longueur des Files d'Attente
NPR :	Nombre des Produits en Retard
ACP :	Analyse en Composantes Principales



## **Annexes**

### **Annexe A : 35 méthodes mathématiques utilisées pour le pilotage des Systèmes de Production ( Teti, 1997) :**

1. Expert System (Knowledge Based System)
2. Pattern Recognition
3. Graph Theory
4. Similarity Theory
5. Optimization Theory
6. Game Theory
7. Time Series Analysis
8. Wavelet Analysis
9. Computer Vision
10. Natural Language Processing
11. Knowledge Representation
12. Heuristic Search
13. Constraint Based Search
14. Confidence Theory
15. Qualitative Reasoning
16. Reasoning Technologies
17. Machine Learning
18. Machine Proving
19. Multiple-Valued Logic
20. Fuzzy Logic
21. Artificial Neural Network
22. Petri Networks
23. Immune Networks
24. Genetic Algorithms
25. Artificial Life
26. Associative Memory
27. Blackboard Architecture
28. Multi-Agent Systems
29. Non-Classical Control Theory
30. Operations Research
31. System Engineering
32. Simulated Annealing
33. Combinatorial Mathematics
34. Fractal Theory
35. Chaos Theory

## Annexe B : résultats d'évaluation des plate-forme d'agent

Name	Voyager	Grasshopper	Aglets	Lana	Odyssey
<b>Availability</b>	Commercial	Free software	Open source	Open source	Free software
<b>Complexity</b>	free software	Sufficient	Small	Small	Small
<b>Mobility</b>	Support	Support	Not Sufficient	No	Sufficient
Validity of Object references	Support (proxies keep valid)	Support(proxies keep valid)	Not Support (lost references of migrating objects)	Support (message boards)	Not Support (references become invalid)
Migration Control	Agent itself, other agents, application	Agent itself, other agents, application, user(GUI), agent system	Agent itself, other agents, agent system(GUI)	Agent system, application	Agent itself, other agents, application
Stationary Agent Concept	No	Yes	No	No	No
<b>Standards</b> <sup>3</sup>	HTTP, CORBA,	MASIF, FIPA, XML, CORBA,	HTTP		
<b>Security</b>	Not Sufficient	Complex	Not Sufficient	Sufficient	Not Sufficient
Internal Security mechanisms	Java Security Manager	Grasshopper Security Manager	Aglet Security Manager	Lana Security Mechanisms	Java Security Manager
External security mechanisms	No	X.509 certificates, SSL (RSA/DES)	No	Lana Security Mechanisms	No
<b>Communication</b>	Method invocation	Method invocation	Message passing	Method invocation	Method invocation
Remote communication	Yes	Yes	Yes	Yes	No
Multicast	Yes	Yes	Yes	No	No
Event service	Yes	Yes	Yes	Yes	No
Message forwarding	Yes	Yes	No	No	No
Communication Language	Java	Java, KQML	Java	Java	Java
<b>GUI</b>	No	Yes	Yes	No	No

---

<sup>3</sup> all platforms support RMI