

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la
Recherche Scientifique



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Ecole Nationale Polytechnique
Département de Génie Industriel

Mémoire du Projet de Fin d'Etudes d'Ingénieur

Thème

Etude d'un problème d'ordonnancement Open Shop par
une approche distribuée :
Application au problème $O4|\tau_j|\bar{F}$

Présenté par :

M^{lle} Amina Sarah BABA AISSA

M^{lle} Rym Meriem MAOUASSA

Dirigé par :

M. BAKALEM

Amina

A la mémoire de mon père

A ma mère, Mon Frère et ma Sœur

A mon tonton

Rym

A ma maman et Mon Frère

Pour leur soutien et leur amour

A toute ma famille

Remerciements :

Nous remercions Monsieur BAKALEM pour son aide, ses précieux conseils et sa disponibilité tout au long de la réalisation de ce mémoire, ainsi que pour l'intérêt qu'il a toujours porté à l'égard de notre travail.

Nous tenons à exprimer notre profonde gratitude envers, Mademoiselle GHAZI Nawel de l'Ecole Nationale Polytechnique d'Alger, Mademoiselle Valérie BELANGER de l'université de LAVAL et Angel RUIZ professeur à l'université de LAVAL pour le temps qu'ils ont consacré à répondre à nos questions.

Nous remercions tous les enseignants du département Génie Industriel pour leur soutien tout au long de notre formation.

ملخص

في هذه الدراسة نتناول مشاكل الجدولة في محيط ورشة صناعية مرنة. الجدولة من النوع المفتوح. الهدف هو تقليص مدة تواجد القطع في الورشة. سنتطرق الى هذه المشكلة وفقا لإستراتيجية تعاونيه وتوزيعية مما يفضي بنا الى وضع بروتوكول للاتصال بين الموارد.

مقاربتنا مقارنة تجريبية محضة مستعملين تقنيات التصنع و ذلك عبر توليد برنامج تجريبي يتضمن عدة قواعد تسيير مقترحة لتحسين مرد و دية الورشة.

المفاتيح

الورشات المرنة، مشاكل الجدولة، الورشات المفتوحة، مقارنة تجريبية، تقنيات التصنع،جدولة توزيعية.

Résumé :

Dans cette étude, on se propose d'étudier un problème d'ordonnement Open Shop dans un environnement flexible d'assemblage. Nous aborderons ce problème selon une stratégie d'ordonnement coopérative et distribuée inspirée des systèmes multi agents nous conduisant à définir un protocole de communication entre les ressources.

Du fait de la complexité NP-difficile au sens fort du problème Open Shop considéré, la nature dynamique des règles d'ordonnement testées et la typologie circulaire de la cellule étudiée (rendant le problème « plus combinatoire » par rapport aux lignes d'assemblage linéaires conventionnelles) nous avons opté pour une approche empirique moyennant la simulation afin d'évaluer la performance de la cellule d'assemblage pour différentes configurations testées. L'instance étudiée est le problème $O4|\tau_j|\bar{F}$.

Mots clés : *Système Flexible de Production, Atelier Flexible, SFP, Système Cyclique d'assemblage, Ordonnement, Open Shop, Simulation, Ordonnement réparti, Système multi agents, Protocole de communication, Ordonnement temps réel.*

Abstract:

In this study, we propose to study a scheduling problem open shop in a flexible manufacturing cell. We will address this problem according to a strategy of cooperative scheduling and distributed based on multi-agent systems, leading us to define a protocol for communication between resources.

Because of the complexity NP-hard in the strong sense of the problem Open Shop in question, the dynamic nature of scheduling rules tested and the typology of the circular cell studied (making the problem "more combinatorial" compared to the linear assembly lines Conventional), we opted for an empirical approach through the simulation to assess the performance of the cell assembly for various configurations tested. The application described in this document is a problem $O4|\tau_j|\bar{F}$.

Keywords: *Flexible manufacturing system, FMS, Cyclic Assembly System, scheduling, Open Shop, Simulation, distributed scheduling, multi agent system, Communication Protocol, Online Scheduling.*

Table des matières :

Introduction et problématique	13
Chapitre I : Généralités et Etat de l'art	18
I. Introduction	19
II. Généralités sur les systèmes flexibles de production SFP	19
1. Les systèmes de production	19
A. Définition	19
B. Décomposition du système de production	20
C. Caractéristiques des systèmes de production	21
2. Les systèmes flexibles de production	22
A. Définition	23
B. Classification des SFP.....	23
C. Eléments constitutifs des SFP	24
III. L'ordonnancement et les systèmes d'ordonnancement	28
1. Positionnement de l'ordonnancement dans la gestion de production	28
2. Généralités sur l'ordonnancement d'atelier manufacturier	29
A. Définition du problème d'ordonnancement	29
B. Eléments de base d'un problème d'ordonnancement.....	30
C. Classification des problèmes d'ordonnancement.....	31
D. Stratégies d'ordonnancement	33
E. Méthodes de résolution des problèmes d'ordonnancement.....	34
F. Complexité des problèmes d'ordonnancement (Open Shop)	35
3. Les systèmes d'ordonnancement	35
IV. Conclusion.....	37
Chapitre II : Modélisation multi agents de la cellule d'assemblage	38
I. Introduction	39
II. Les systèmes d'ordonnancement distribués	39
1. Nécessité d'une approche distribuée	39
2. Présentation des systèmes multi agents	40
A. Notions d'agents	40
B. Définition d'un système multi agents.....	41
C. Les types d'agents.....	42
D. Modèles d'agent.....	43
E. Les formes d'interaction entre agents	45
F. La communication entre agents	47
G. Méthodologie de conception des systèmes multi agents.....	48
III. Modélisation multi agents.....	51
1. Rappel du problème étudié	51
2. Description du système cyclique d'assemblage.....	52
3. Approche de modélisation multi agents.....	53
4. Protocole de coopération inter ressources.....	54

A.	Les primitives de coopération	55
B.	Les étapes du protocole de communication	56
5.	Architecture du centre de décision ordonnanceur.....	58
IV.	Conclusion.....	63
Chapitre III : Modélisation et simulation de la cellule d'assemblage		64
I.	Introduction	65
II.	Simulation, définitions et concepts	65
1.	Définitions	65
2.	Fondement de la simulation.....	66
A.	Modélisation et processus de modélisation.....	66
B.	Types de modèles pour la simulation.....	67
3.	Méthodes analytiques vs simulation.....	68
4.	Apport de la simulation dans les systèmes de production	68
5.	Avantage et limite de la simulation	69
6.	Les étapes du processus de simulation	70
III.	Présentation du logiciel ARENA 7.01	71
1.	Structure d'un programme ARENA	71
2.	Structure hiérarchique d'ARENA.....	72
3.	Transfert des entités dans ARENA.....	73
4.	Utilisation de VBA dans ARENA	73
IV.	Mise en œuvre du processus de simulation.....	75
1.	Formulation du problème	75
2.	Identification et collecte des données	76
3.	Modélisation par l'approche processus.....	76
4.	Construction du modèle ARENA et transcription informatique du modèle	77
A.	Modélisation statique de l'atelier d'assemblage.....	78
B.	Modélisation dynamique de la cellule d'assemblage.....	82
5.	Vérification et validation du modèle	84
V.	Conclusion.....	84
Chapitre IV : Simulation et test de la stratégie d'ordonnancement proposée.....		86
I.	Introduction	87
II.	La planification stratégique et tactique de la simulation.....	87
1.	Le choix d'une configuration.....	88
2.	Le plan d'expérimentation.....	89
3.	Les indicateurs de performance	90
III.	Analyse et interprétation des résultats	93
1.	Analyse des temps d'assemblage d'une commande	93
2.	Analyse des temps de séjour moyen des pièces dans la cellule d'assemblage	95
3.	Analyse des temps de transfert moyen des pièces dans la cellule d'assemblage	98
4.	Analyse des temps d'attente moyen des pièces dans les files d'attente d'entrée	102
5.	Analyse des temps d'attente moyen des pièces dans la cellule d'assemblage	105
6.	Analyse du nombre moyen des pièces en attente dans les files d'attente d'entrée des stations	106

7. Analyse des taux d'utilisation des robots.....	108
IV. Conclusion.....	109
Conclusion et perspectives	111
Références bibliographiques	114
Annexes	117

Liste des figures :

Figure I.1 : Décomposition d'un système de production	20
Figure I.2: L'atelier flexible, compromis entre flexibilité et productivité	23
Figure I.3: classification des SFP.....	24
Figure I.4: Composants de la cellule d'assemblage	25
Figure I.5 : schéma d'une station	26
Figure I.6: aperçu sur les systèmes de contrôle des SFP.....	27
Figure I.7 : Fonctions de pilotage de la production	29
Figure I.8: typologie par les ressources des problèmes d'ordonnancement	30
Figure I.9: Flow shop à 3 machines	32
Figure I.10: Exemple d'organisation Job Shop Simple à 4 machines.....	32
Figure I.11 : classification des problèmes d'ordonnancement	33
Figure I.12 : Les différentes structures des systèmes d'ordonnancement	37
Figure II.1 : Les 3 phases de réalisation d'une tâche par un agent.....	41
Figure II.2: Système multi agents vu selon différents niveaux de détail.....	42
Figure II.3 : architecture d'un agent réactif	42
Figure II.4: architecture d'un agent cognitif	43
Figure II.5 : Une architecture BDI.....	44
Figure II.6 : Architecture AOP	45
Figure II.7 : Les méthodologies orientées agents	48
Figure II.8 : Architecture du système d'ordonnancement basé sur la coopération	52
Figure II.9 : Cellule flexible d'assemblage étudiée.....	53
Figure II.10 : modèle d'agent pour un système d'ordonnancement coopératif.....	54
Figure II.11 : Philosophie générale du protocole de coopération.....	56
Figure II.12 : architecture de l'agent ordonnanceur	59
Figure II.13: <i>Cas 1</i> : Arrivée d'une pièce ne disposant pas d'engagement (EG=0).....	61
Figure II.14 : <i>Cas 2</i> : Arrivée d'une pièce disposant d'un engagement (EG≠0)	62
Figure III.1 : Processus de modélisation	67
Figure III.2: simulation des systèmes de production.....	69
Figure III.3:Les étapes d'un projet de simulation selon Pritsker.....	71
Figure III.4: Structure hiérarchique d' <i>Arena</i>	73
Figure III.5 : Procédure de VBA.....	74
Figure III.6 : Modélisation de l'entrée du système par RockWell Arena 7.01.....	78
Figure III.7 : Modélisation de l'entrée d'une station par RockWell Arena 7.01	79
Figure III.8 : Modélisation Submodel « Entrée station 1 » par RockWell Arena 7.01	80
Figure III.9 : Modélisation de la station d'assemblage par RockWell Arena 7.01.....	81
Figure III.10 : Modélisation de la sortie d'une station par RockWell Arena 7.01	81
Figure III.11 : Modélisation de la sortie du système par RockWell Arena 7.01	82
Figure III.12: La représentation graphique de la cellule par RockWell Arena 7.01.....	82
Figure III.13: a. Station, b. file d'attente d'entrée des stations, c. File d'attente de sortie	82
Figure III.14: La représentation graphique des robots	83
Figure III.15: modélisation des Palette par des transporteurs	83

Figure III.16: modélisation d'un tronçon du circuit par <i>Distances</i>	83
Figure III.17: le système en plein évolution.....	84
Figure IV.1 : Les taux d'amélioration des temps d'assemblage d'une commande par la règle1	93
Figure IV.2 : Les taux d'amélioration des temps d'assemblage d'une commande par la règle 2	93
Figure IV.3 : Moyenne et écart type des temps de séjours d'une commande dans la cellule.....	94
Figure IV.4 : Fréquence des cas d'amélioration du Cmax	94
Figure IV.5 : Taux d'amélioration des temps de séjour moyens des pièces par la règle 1 pour les cas dont la capacité des FAEs est de 3	95
Figure IV.6 : Taux d'amélioration des temps de séjour moyens des pièces par la règle 1 pour les cas dont la capacité des FAEs est de 4	96
Figure IV.7 : Taux d'amélioration des temps de séjour moyens des pièces par la règle 1 pour les cas dont la capacité des FAEs est de 5	96
Figure IV.8 : Moyenne et écart type des taux d'amélioration des temps de séjour moyens des pièces par la règle1	96
Figure IV.9 : Taux d'amélioration des temps de séjour moyens des pièces par la règle 2 pour les cas dont la capacité des FAEs est de 3	97
Figure IV.10 : Taux d'amélioration des temps de séjour moyens des pièces par la règle 2 pour les cas dont la capacité des FAEs est de 4	97
Figure IV.11 : Taux d'amélioration des temps de séjour moyens des pièces par la règle 2 pour les cas dont la capacité des FAEs est de 5	97
Figure IV.12 : Moyenne et écart type des taux d'amélioration des temps de séjour moyens des pièces par la règle 2	98
Figure IV.13 : Moyenne et écart type des taux d'amélioration des temps de séjour moyens des pièces dans la cellule par les règles de coopération	98
Figure IV.14 : Taux d'amélioration des temps de transfert moyens des pièces par la règle 1 dans les cas où la capacité des FAEs est de 3	99
Figure IV.15 : Taux d'amélioration des temps de transfert moyens des pièces par la règle 2 dans les cas où la capacité des FAEs est de 3	99
Figure IV.16 : Moyenne et écart type des taux d'amélioration des temps de transfert moyens des pièces dans la cellule par les règles de coopération dans les cas où la capacité des FAEs est de 3	99
Figure IV.17 : Taux d'amélioration des temps de transfert moyens des pièces par la règle 1 dans les cas où la capacité des FAEs est de 4	100
Figure IV.18 : Taux d'amélioration des temps de transfert moyens des pièces par la règle 2 dans les cas où la capacité des FAEs est de 4	100
Figure IV.19 : Moyenne et écart type des taux d'amélioration des temps de transfert moyens des pièces dans la cellule par les règles de coopération dans les cas où la capacité des FAEs est de 4	100
Figure IV.20 : Taux d'amélioration des temps de transfert moyens des pièces par la règle 1 dans les cas où la capacité des FAEs est de 5	101
Figure IV.21 : Taux d'amélioration des temps de transfert moyens des pièces par la règle 2 dans les cas où la capacité des FAEs est de 5	101
Figure IV.22 : Moyenne et écart type des taux d'amélioration des temps de transfert moyens des pièces dans la cellule par les règles de coopération dans les cas où la capacité des FAEs est de 5	101
Figure IV.23 : Taux d'amélioration des temps d'attente moyens des pièces dans les FAEs dont la capacité est de 3 par la règle 1	102
Figure IV.24 : Taux d'amélioration des temps d'attente moyens des pièces dans les FAEs dont la capacité est de 3 par la règle 2.....	102
Figure IV.25: Taux d'amélioration des temps d'attente des pièces dans les FAEs dont la capacité est de 3.....	102
Figure IV.26 : Taux d'amélioration des temps d'attente moyens des pièces dans les FAEs dont la capacité est de 4 par la règle 1.....	103
Figure IV.27: Taux d'amélioration des temps d'attente moyens des pièces dans les FAEs dont la capacité est de 4 par la règle 2.....	103
Figure IV.28: Moyenne des taux d'amélioration des temps d'attente des pièces dans les FAEs dont la capacité est de 4	103

Figure IV.29 : Taux d'amélioration des temps d'attente moyens des pièces dans les FAEs dont la capacité est de 5 par la règle 1.....	104
Figure IV.30 : Taux d'amélioration des temps d'attente moyens des pièces dans les FAEs dont la capacité est de 5 par la règle 2.....	104
Figure IV.31 : Moyenne des taux d'amélioration des temps d'attente moyens des pièces dans les FAEs dont la capacité est de 5	105
Figure IV.32: Moyenne des temps d'attente des pièces dans les FAEs	105
Figure IV.33 : Taux d'amélioration du temps d'attente moyen des pièces dans la cellule par la règle 1.....	106
Figure IV.34 : Taux d'amélioration du temps d'attente moyen des pièces dans la cellule par la règle 2.....	106
Figure IV.35 : Moyenne et écart type du nombre moyen des pièces en attente dans les FAEs dont la capacité est de 3 par les règles de coopération	107
Figure IV.36 : Moyenne et écart type du nombre moyen des pièces en attente dans les FAEs dont la capacité est de 4 par les règles de coopération	107
Figure IV.37 : Moyenne et écart type du nombre moyen des pièces en attente dans les FAEs dont la capacité est de 5 par les règles de coopération	107
Figure IV.38 : Moyenne et écart type des taux d'amélioration des robots par les règles de coopération dans les cas où la capacité des FAEs est de 3	108
Figure IV.39 : Moyenne et écart type des taux d'amélioration des robots par les règles de coopération dans les cas où la capacité des FAEs est de 4	108
Figure IV.40 : Moyenne et écart type des taux d'amélioration des robots par les règles de coopération dans les cas où la capacité des FAEs est de 5	109

Liste des tableaux :

Tableau II.1: Classification des situations d'interaction	46
Tableau II.2 : Quelques actes de langages élémentaires	48
Tableau IV.1: Les distances entre stations	88
Tableau IV.2 : résumé du plan expérimental	90

Liste des abréviations :

SPT: Shortest Processing Time
LPT: Longest Processing Time
LWKR1: Least Work Remaining First, selon la durée d'exécution totale restante.
LWKR2: Least Work Remaining First, selon le nombre d'étapes d'assemblage restantes.
MWKR1: Most Work Remaining First, selon la durée d'exécution totale restante.
MWKR2: Most Work Remaining First, selon le nombre d'étapes d'assemblage restantes.
SFP : Système flexible de production
FMS: Flexible manufacturing system
MOCN : Machine outil à commande numérique
LAN: local area network
PSE : Procédures par Séparation et Évaluation
PL : Méthodes basées sur la Programmation Linéaire
PD : Méthodes basées sur la Programmation Dynamique
SPT: Shortest Processing Time
EDD: Earliest deadline
NP: Non polynomial
ERP: Entreprise Resource Planning
MES: Manufacturing Execution System
IA : Intelligence Artificielle
IAD : Intelligence Artificielle Distribuée
SMA : Systèmes Multi-Agents
BDI: Beliefs, Desires, Intentions
AOP: Agent Oriented Programming
KQML: Knowledge Query Manipulation Language
FIPA: Foundation for Intelligent Physical Agents
MaSE: Multiagent Systems Engineering
UML: Unified Modeling Language
Sdp : Système de production
SED : Simulation à événements discrets
VBA : Visual Basic for Application
FAEs : Files d'attente d'entrée des différentes stations
FAEi : File d'attente d'entrée de la *station i*
FIFO: First in First out
FASs: File d'attente de sortie des différentes stations
FASi : File d'attente de sortie de la *station i*

Liste des symboles :

<i>Symboles</i>	<i>Description</i>
ri	Date de disponibilité de la tâche
di	La date de fin ou d'achèvement de la tâche i , (Completion time) ;
Fi	La durée de séjour de la tâche i dans l'atelier, $Fi = Ci - ri$ Flow time ;
Li	Le retard algébrique $Li = Ci - di$ (Lateness), il dénote l'écart par rapport au délai souhaité avec avance favorable et retard défavorable ;
Ti	Le retard vrai $Ti = \max(0, Ci - di)$ (Tardiness) ;
Ui	L'indicateur de retard $Ui (Ui = 0 \text{ si } Ci \leq di, Ui = 1 \text{ sinon})$;
Ei	L'avance $Ei = \max(0, di - Ci)$ (Earliness).
di	Date de livraison de la tâche i (Due date)
Pi	Durée opératoire de la tâche i
ak	Capacité de la ressource k
$Cmax$	Durée totale de l'ordonnancement
\bar{F}	Durée moyenne de séjour
$Fmax$	La plus grande durée de séjour
$Tmax$	Le plus grand retard vrai
\bar{T}	Le retard moyen
N_T	Nombre de tâches en retard
$Lmax$	Plus grand retard algébrique
DOi	Durée opératoire de l'entité dans la station i
NumberOut	Nombre de produits finis sortant de la cellule
TPTi	Temps de prochain traitement de la pièce au niveau d'une station engagée i
TimeIn	Moment d'entrée dans la cellule
VarArrayi	Variation i
EG = j	Engagement de la station j
$NQ(FAEi)$	l'état de la file d'attente de la station i

F	temps de séjour total (Total Flow Time)
DO_i	Durée Opératoire de la <i>tâche i</i>
$NFAE_i(t)$	Nombre de pièces dans la file d'attente <i>i</i> à l'instant <i>t</i>
$TNOW$	Run Current Time
T_{ij}	Temps de transfert de la <i>station i</i> à la <i>station j</i>
$\tau_{TSCR\grave{e}gle}$	Taux d'amélioration des temps de séjour d'un lot dans la cellule
T_{SCFIFO}	Temps de séjour d'une commande sous un pilotage FIFO
$T_{SCR\grave{e}gle}$	Temps de séjour d'une commande sous le pilotage de la Règle utilisée
$\tau_{TSR\grave{e}gle}$	Taux d'amélioration des temps de transfert
T_{SFIFO}	Temps de séjour d'une pièce sous le pilotage FIFO
$T_{SR\grave{e}gle}$	Temps de transfert d'une pièce sous le pilotage de la Règle utilisée
$\tau_{TAFR\grave{e}gle}$	Taux d'amélioration des temps d'attente dans les FAEs
T_{AFFIFO}	Temps d'attente d'une pièce dans une FAE sous le pilotage FIFO
$T_{AFR\grave{e}gle}$	Temps d'attente d'une pièce dans une FAE sous le pilotage de la Règle utilisée
$\tau_{TAR\grave{e}gle}$	Les taux d'amélioration des temps d'attente dans la cellule
T_{AFIFO}	Temps d'attente d'une pièce dans la cellule sous le pilotage FIFO
$T_{AR\grave{e}gle}$	Temps d'attente d'une pièce dans la cellule sous le pilotage de la Règle utilisée
$\tau_{NAFR\grave{e}gle}$	Taux d'amélioration du nombre de pièces en attente dans les FAEs
N_{AFFIFO}	nombre de pièces en attente dans une FAE sous le pilotage FIFO
$N_{AFR\grave{e}gle}$	nombre de pièces en attente dans une FAE sous le pilotage de la Règle utilisée
$\tau_{TURR\grave{e}gle}$	Taux d'amélioration des taux d'utilisation des robots
T_{URFIFO}	Taux d'utilisation du <i>robot Ri</i> sous le pilotage FIFO
$T_{URR\grave{e}gle}$	Taux d'utilisation du <i>robot Ri</i> obtenu par la Règle utilisée

Introduction et problématique :

La flexibilité et la réactivité sont devenues des qualités incontournables pour les entreprises, car elles se trouvent confrontées à une demande variée et fluctuante avec des contraintes de qualité et de délais de plus en plus fortes.

Les pressions toujours grandissantes concernant la réduction des délais de livraison et la diminution de la taille des lots de production obligent ces dernières à opter pour de nouveaux systèmes de production, tels que les systèmes flexibles de production avec des systèmes de pilotage capables de les gérer.

Les systèmes flexibles sont des systèmes rapidement reconfigurables en fonction des contraintes de production. Ils constituent une méthode alternative d'agencement des postes de travail quand la production en ligne (à la chaîne) devient trop complexe à organiser du fait de la diversification des produits. Il s'agit à la base d'ateliers de type *job shop* dans lesquels stockages, manipulations et traitements des pièces sont automatisés, le cas extrême étant l'*open shop* dans lequel les opérations à réaliser sur les produits sont permutable (pas de séquence fixée).

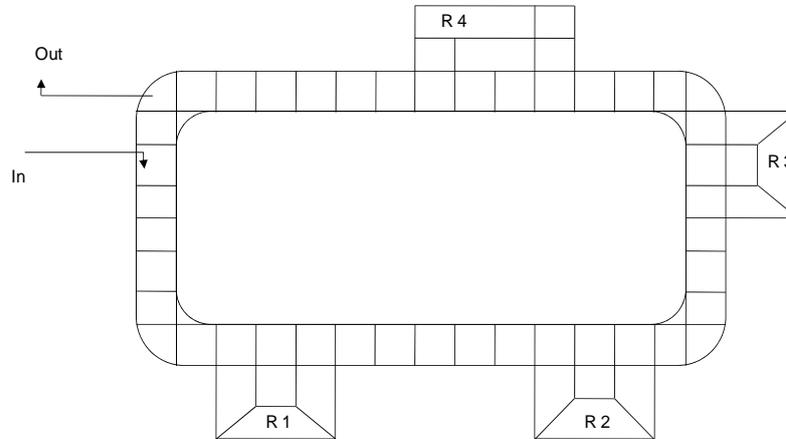
Pour de tels systèmes, l'ordonnancement des tâches de production devient une fonction critique de la gestion opérationnelle et l'un des points clés de la rentabilité du système manufacturier sachant qu'elle permet à la fois l'optimisation de l'utilisation des ressources et la réactivité de l'atelier face aux évolutions du marché.

En effet, le problème d'ordonnancement est relativement simple dans le cas des chaînes de fabrication où les produits suivent en flots continus une suite d'opérations sans grande variation. Mais il se complique pour les systèmes flexibles de production dont la fabrication se fait en petit lots avec des processus de fabrication diversifiés qui admettent des alternatives dans la suite des opérations.

Ces évolutions vers une automatisation flexible ont des répercussions sur la conception et la modélisation des systèmes d'ordonnancement. Elles ont notamment conduit à délaisser les structures centralisées, trop rigides pour essayer de mettre en œuvre des structures **réparties**, où le pilotage est partagé entre plusieurs centres de décisions dotés chacun d'une capacité d'autonomie totale ou partielle. Ces systèmes permettent ainsi d'améliorer la flexibilité et l'adaptabilité des équipements tout en maîtrisant les problèmes d'explosion combinatoire qu'elles peuvent engendrer. Dès lors, la performance du système n'est pas globalement planifiée, elle émerge de la dynamique des interactions en **temps réel** entre les entités.

Dans le but d'évaluer les potentialités offertes par les SFP, Finke et Dupont [FIN93], s'intéresseront à une configuration particulière de ces systèmes qui consisterait à disposer les postes de travail de manière cyclique autour d'un convoyeur en boucle permettant de les relier

entre eux. Cette configuration, particulièrement robuste, présente de nombreux avantages ; elle permet en effet de mieux parer aux perturbations de la production en évitant toute forme de blocage ou d'interruption du flux ainsi que de limiter considérablement les ressources de stockage sachant que les pièces ont la possibilité de circuler sur le convoyeur.



Configuration cyclique de la cellule d'assemblage étudiée

Néanmoins, il s'est avéré qu'une telle organisation de type *job shop* n'apportait aucune amélioration à la performance de la cellule par rapport à une organisation classique linéaire, mais que l'intérêt était dans la manipulation des plans d'assemblage ouverts (*open-shop*) où chaque ordre des opérations d'assemblage est possible. Ils ont montré que l'apport d'une combinaison plans non linéaires-pilotage FIFO à la performance globale du système est insignifiant (inférieur à 5%).

Une deuxième étude, réalisée par [GHA06], a permis de démontrer qu'un ordonnancement Open Shop associé à des règles de priorité (SPT, LPT, LWKR1, LWKR2, MWKR1, MWKR2) au niveau des files d'attente ne permettait toujours pas d'améliorer le temps moyen de séjour des pièces dans la cellule par rapport à la configuration initiale (Open Shop + FIFO) étudiée par Finke et Dupont [FIN93].

L'analyse des résultats a permis de mettre en évidence un certain nombre d'éléments :

- Des réductions considérables sont réalisées sur certains types de pièces mais se trouvent généralement absorbées par des retards acquis par d'autres types de pièces du fait des phénomènes d'interactions;
- Toute amélioration des temps d'attente est accompagnée d'une dégradation des temps de transfert et vice versa ; ce qui explique l'absence de performance globale (C_{max}).

Tous ces éléments justifient l'absence d'une performance stable et régulière. Ceci peut s'expliquer par un manque de collaboration entre les différents centres de décisions ou ressources. En effet, chaque équipement est géré localement indépendamment des autres par l'application de simples règles de priorité au niveau des files d'attente d'entrée. Il est clair que garantir une autonomie au niveau des centres de décision c'est accepter que les décisions

prises s'appuient sur une vision limitée du problème global. Ces décisions ne peuvent être que sous optimales, comparées au cas où le même problème est traité par une coordination avec une vision plus générale.

Pour bien exploiter les propriétés d'un système de pilotage réparti dans un intérêt global, une coordination des actions individuelles doit être assurée au sein de l'organisation de l'ensemble.

Dans cette perspective, on se propose d'analyser le comportement d'un tel système suivant un ordonnancement collaboratif qui permettrait de prendre en compte, en temps réel, l'état des différentes ressources pour déterminer un meilleur routage de la pièce à travers le système afin de compléter son assemblage. Notre démarche consiste à gérer de manière locale les files d'attente qui se forment devant les ressources avec en même temps à un niveau plus global, diverses règles de pilotage qui permettront d'ajuster les comportements individuels afin d'obtenir un comportement global cohérent par rapport aux objectifs fixés et aux tâches à accomplir.

Le champ d'application de cette approche décrit dans ce document est un problème d'ordonnancement *Open shop* à 4 ressources avec pour fonction objectif la minimisation du temps de séjour des pièces $O4|\tau_j|\bar{F}$ dans la cellule.

Du fait de la complexité NP-difficile au sens fort du problème *Open Shop* considéré, la nature dynamique des règles d'ordonnancement testées et la typologie circulaire de la cellule étudiée (rendant le problème « plus combinatoire » par rapport aux lignes d'assemblage linéaires conventionnelles) nous avons opté pour une approche empirique moyennant la simulation.

Notre approche de résolution se compose de 3 étapes successives :

Modélisation multi agents de la cellule d'assemblage : la première phase de résolution passe par la modélisation et la formalisation de la cellule d'assemblage. Les SMA, inspirés de l'intelligence artificielle distribuée, constituent un bon moyen pour spécifier la répartition des tâches entre les différents centres de décision ou agents. Ceci comprend la définition d'une architecture individuelle adaptée pour chaque agent, ainsi qu'un protocole de communication destiné à gérer les interactions en temps réel entre les agents pour assurer leur coopération selon des règles préalablement définies.

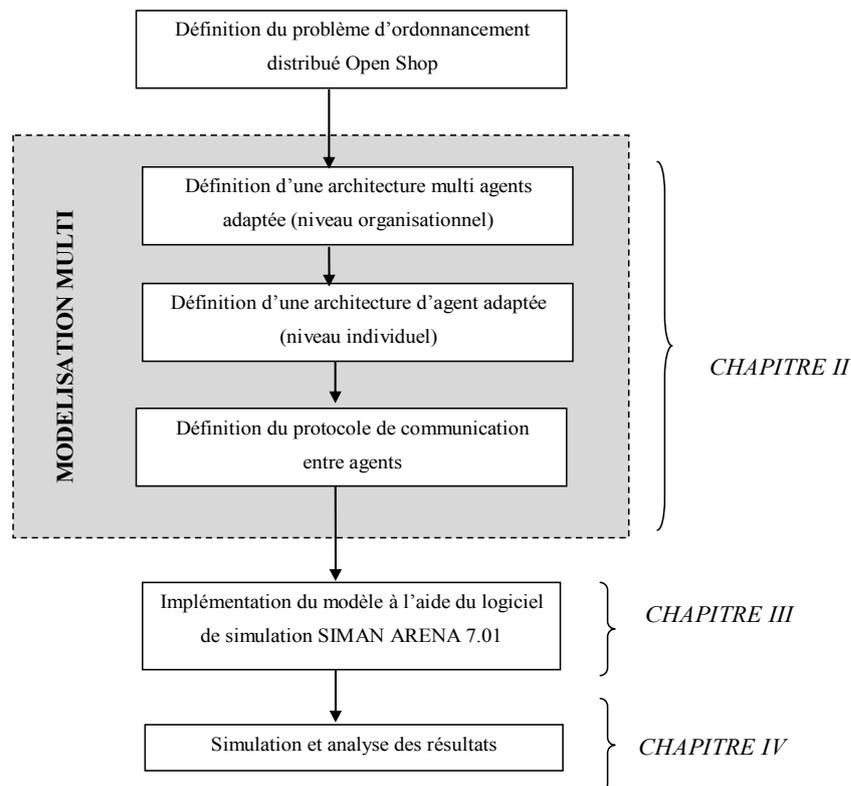
Construction d'un modèle de simulation: cette étape vise à reproduire le comportement du système tel que défini par l'étape 1 dans un programme informatique à l'aide du logiciel de simulation Arena Rockwell 7.01. Le choix de ce logiciel est intéressant dans la mesure où il présente une facilité d'utilisation grâce à une interface graphique adaptée aux systèmes industriels, ainsi que de nombreux outils d'analyse et d'optimisation. De plus il offre plusieurs

niveaux de modélisation ainsi que la possibilité d'introduire des codes en C++, VBA pour un niveau de détails important.

Exécution de la simulation et exploitation des résultats : l'utilisation des techniques de simulation discrète permet dans ce cas de constituer un modèle de l'évolution dynamique du système de production en fonction des décisions prises tant structurelles que décisionnelles. Le modèle du système de production sera alors intégré à un outil d'aide à la décision permettant de tester la stratégie d'ordonnancement proposée.

Organisation du manuscrit :

Ce mémoire est composé de quatre chapitres qui s'articulent selon le plan ci-dessous :



Le premier chapitre a pour objectif de préciser les concepts fondamentaux nécessaires à l'aboutissement de notre étude. La première partie de ce chapitre est consacrée à l'aspect physique des systèmes de production, et principalement aux systèmes flexibles de production. Nous y décrivons les différents composants de ces systèmes et cela dans le but de se familiariser avec l'objet de cette étude à savoir les cellules flexibles.

Nous nous intéresserons, par la suite, à la partie pilotage des systèmes, indispensable à la bonne marche de tout système de production, en prêtant une attention toute particulière au problème d'ordonnancement en tant qu'élément fondamental du pilotage opérationnel des ateliers manufacturiers. L'ordonnancement, objet principal de notre travail, est donc le sujet de la deuxième partie de ce chapitre.

Les chapitres 2, 3, 4 décrivent la mise en œuvre de l'approche proposée pour la résolution du problème étudié. Dans le cadre du chapitre 2, nous développerons une approche de modélisation du système basée sur paradigme agent en tant que nouvelle approche d'analyse et de conception des systèmes de pilotage distribués.

Au chapitre 3, nous présenterons le modèle de simulation conçu à l'aide du logiciel de simulation Arena Rockwel 7.01. C'est le modèle d'action qui nous permettra d'agir sur le système afin de tester la stratégie d'ordonnancement proposée. Enfin, le chapitre 4 décrit le processus d'expérimentation et d'analyse du système selon un plan d'expérimentation préétabli.

Nous clorons ce mémoire par une conclusion générale dans laquelle nous développerons des perspectives selon les résultats obtenus à l'issue de notre étude.

Chapitre I : Généralités et Etat de l'art

- I. Introduction

- II. Généralités sur les systèmes flexibles de production **SFP**
 - 3. Les systèmes de production
 - A. Définition
 - B. Décomposition du système de production
 - C. Caractéristiques des systèmes de production
 - 4. Les systèmes flexibles de production
 - A. Définition
 - B. Classification des SFP
 - C. Eléments constitutifs des SFP

- III. L'ordonnancement et les systèmes d'ordonnancement
 - 4. Positionnement de l'ordonnancement dans la gestion de production
 - 5. Généralités sur l'ordonnancement d'atelier manufacturier :
 - G. Définition du problème d'ordonnancement
 - H. Eléments de base d'un problème d'ordonnancement
 - I. Classification des problèmes d'ordonnancement
 - J. Stratégies d'ordonnancement
 - K. Méthodes de résolution des problèmes d'ordonnancement
 - L. Complexité des problèmes d'ordonnancement (Open Shop)
 - 6. Les systèmes d'ordonnancement

- IV. Conclusion

I. Introduction :

Ce chapitre a pour objet de présenter toutes les notions élémentaires essentielles à notre étude afin de l'entamer sans aucune ambiguïté. La deuxième section de ce chapitre est consacrée à la partie physique du système de production et plus précisément aux systèmes flexibles de production (SFP) ; nous y présenterons une description détaillée des SFP, leurs composants ainsi qu'une classification de ces systèmes, telle que proposée dans [JUB94].

La troisième section est consacrée à l'ordonnement des systèmes manufacturiers. Dans cette section, nous visons à situer la fonction ordonnancement dans le processus de pilotage de la production en respectant la hiérarchie entre les différents niveaux de pilotage conformément à la logique MRPII. Nous nous sommes intéressés dans un premier temps aux éléments fondamentaux d'un problème d'ordonnement, nous en présenterons les différentes classifications et enfin, nous décrirons ses différentes stratégies et méthodes de résolution. Nous abordons également, les différentes organisations d'ateliers afin d'introduire le cas qui nous intéresse : l'Open Shop.

Dans la dernière sous section, nous focaliserons notre attention sur les différentes structures permettant la mise en œuvre de l'ordonnement dans la gestion de production avec le souci de positionner ces systèmes parmi les différents systèmes de pilotage de l'entreprise.

II. Généralités sur les SFP :

1. Le système de production :

A. Définition :

Un système de production est un ensemble de ressources réalisant une activité de production. [GHA06]. Les ressources appartenant au système de production mobilisées pour réaliser l'activité de production peuvent être des machines, des opérateurs, de l'énergie, des informations, des outillages,..., etc.

Une des fonctions importantes du système de production est la fabrication elle-même du produit fini, mais son bon déroulement nécessite la mise en œuvre de fonctions additionnelles telles que les achats de composants et de matières premières, la distribution du produit fini, la gestion de la qualité des composants et du produit où la maintenance des différentes ressources intervient aussi de manière importante.

Dans le cadre d'une entreprise, le système de production, outre sa finalité première qui est de produire un bien économique, cherche à satisfaire d'autres objectifs : [WEB01]

- **Objectif en termes de quantités produites :** La fonction de production doit permettre à l'entreprise de satisfaire la demande qui lui est adressée ce qui suppose que l'entreprise adapte sa capacité de production au volume des ventes ;
- **Objectif en termes de qualité :** Les biens économiques produits doivent être de bonne qualité, c'est-à-dire doivent permettre de satisfaire les besoins de la clientèle ;
- **Objectif en termes de coût :** Le système productif adopté par l'entreprise doit proposer les plus faibles coûts de production possibles de manière à garantir la compétitivité de l'entreprise ;

- **Objectif en termes de délai** : Le système de production doit certes produire, mais dans des délais raisonnables.

B. Décomposition des systèmes de production :

D'un point de vue systémique, il est classique de décomposer le système « entreprise » en trois sous-systèmes qui coopèrent [HAB01], [GIA07], [KIE02] : Le premier correspond à une réalité **tangible**, tandis que les deux autres relèvent d'une modélisation de ce que le processus peut et doit faire :

- Le *sous-système opérant* (ou sous-système physique) correspond à l'ensemble des ressources matérielles (machines, outillage, aires de stockage, composants et matières premières...) et humaines (opérateurs), disponibles dans le système sur la période considérée.
- Le *sous-système d'information* comporte un ensemble d'informations techniques (gammes et nomenclatures), un ensemble d'informations de gestion décrivant l'état du système (utilisation des ressources, commandes à exécuter, avancement des commandes en cours d'exécution...) à des instants donnés ou en continu, et un ensemble de procédures mobilisées dans la prise de décision.
- Le *sous-système de décision* utilise le système d'information pour piloter le système opérant (identifie, analyse et corrige les dérives en proposant des actions correctives ou préventives).

Dans les systèmes de production, les sous-systèmes d'information et de décision n'ont pas d'existence propre, l'un sans l'autre. Ils constituent ensemble ce que nous appelons le *système de pilotage* ou le *système d'information et de décision (SID)* ou encore le *système directeur* (voir figure I.1). Ainsi, il est plus conforme à la réalité de considérer le système de production comme l'association d'un système de fabrication et d'un système de pilotage.

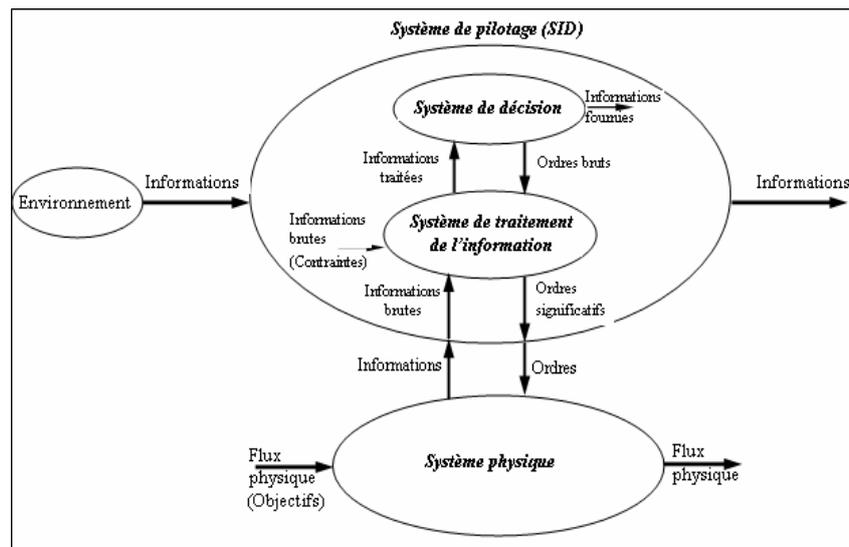


Figure I.1 : Décomposition d'un système de production [GHA06]

C. Caractéristiques des systèmes de production :

Si la valeur d'un produit était autrefois caractérisée par son coût, elle intègre désormais d'autres aspects tels que : la qualité, la possibilité de personnalisation (créer la diversité), les délais de livraison,...etc. Pour répondre à ces nouvelles exigences, les systèmes de production doivent se doter de nouvelles caractéristiques telles que : la flexibilité, la réactivité, la proactivité et la robustesse.

Flexibilité : « *La flexibilité est la capacité ou l'aptitude ou la rapidité d'un système industriel à créer et à gérer la variété, de façon économique et continue, afin de s'adapter aux changements de l'environnement, externe ou interne, tout en maintenant son équilibre (sa stabilité)* » [KIE02].

La définition retenue décrit la flexibilité comme une capacité d'adaptation sous la double contrainte de l'incertitude et de l'urgence. Elle note le coût lié à une telle adaptation, posant le dilemme entre productivité et flexibilité. Elle introduit par ailleurs la notion de rapidité avec laquelle le système réalise cette adaptation. Nous distinguons, d'autre part, deux grandes approches qui peuvent coexister au sein d'une même installation : [JUB94]

- La flexibilité capacitaire (« flexibilité-volume »), elle caractérise la capacité du système à absorber des variations de volumes de production sans pénaliser les délais ; elle est essentiellement obtenue par des équipements en surnombre, mais aussi par la standardisation des équipements ;
- La flexibilité de diversification (« flexibilité- variété »), qui consiste à pouvoir s'adapter à des variantes de produits réalisés (familles de produits), sans modification substantielle du système.

Réactivité : « *La réactivité est la capacité du système à élaborer une solution admissible face à un aléa, dans un délai suffisant par rapport à la dynamique de l'environnement* » [KIE02].

Selon les définitions citées ci-dessus, nous pouvons conclure que les notions de flexibilité et de réactivité sont complémentaires. La réactivité doit permettre de donner une solution à une perturbation, alors que flexibilité donne au système de production un nombre de choix possibles face à cette perturbation. *La flexibilité* est donc un moyen pour une meilleure *réactivité* [KIE02].

Robustesse : F. Chetouane définit la robustesse comme suit : « *La robustesse d'un système de production est sa capacité à maintenir ses performances, face aux perturbations.* » [BES95].

Des aléas peuvent apparaître en cours de production mais le système est conçu et dimensionné afin qu'il n'y ait aucun élément perturbateur pour la production et que les objectifs du plan directeur de production soient réalisés.

Proactivité : « *La proactivité d'un système de production se caractérise par ses capacités d'anticipation (prévoir et/ou provoquer) les changements d'état, d'apprentissage et d'enrichissement des connaissances (pour améliorer sa réactivité), d'adaptation de ses règles de fonctionnement et par sa capacité de réorganisation reposant sur une architecture décentralisée et une délégation de responsabilité.* » [GHA06]

La notion de proactivité est toujours rattachée à celle de gestion des perturbations (aléas), avec comme but d'empêcher leur apparition. Elle met ainsi en œuvre différents points caractérisant un système proactif, par exemple : l'anticipation et la prévention (détecter de manière précoce les dégradations), la nécessité pour les acteurs de disposer de compétences suffisantes pour élaborer des réponses nouvelles et pertinentes...etc.

2. Les systèmes flexibles de production :

Une évolution particulièrement importante des techniques de production a pris place vers la fin des années 60, caractérisée par l'avènement de l'automatisation flexible.

Elle est le fruit de la conjonction entre : [JUB94]

Le raccourcissement de la durée de vie des produits : la concurrence incite en effet à modifier plus souvent les produits existants ou à les remplacer par d'autres plus performants ; par ailleurs, la tendance est de proposer à la clientèle plusieurs variantes de produits de base ou des options. Ces diverses contraintes se traduisent par le fait qu'il est de plus en plus rare d'avoir à fabriquer un seul modèle d'un produit qui reste figé pendant plusieurs années.

Enfin, le souci de réduire les frais financiers liés aux stocks, et l'introduction dans les entreprises de systèmes de gestion de production pour tendre notamment vers une production « **juste à temps** » ont pour conséquence de conduire à fractionner celle-ci, par lancement de **petits lots** répétitifs, en fonction des besoins.

Dans ces conditions, une installation dédiée à la fabrication d'un seul produit est confrontée au dilemme suivant : [AZZ05]

- fonctionner toute l'année, ce qui conduit à de faibles taux d'utilisation des équipements et entraîne des frais de main-d'œuvre élevés ;
- ou fonctionner quelques mois par an : en résultent des volumes plus importants et des frais de main-d'œuvre plus supportables, mais des investissements plus difficiles à amortir ;

Ces contraintes ont donc fait naître le besoin d'**ateliers de fabrication** assez souples, **dits polyvalents ou flexibles**, pour pouvoir prendre en charge, en l'état ou moyennant des modifications mineures, des procédés non prévus lors de la phase de conception.

Ils permettent ainsi de réagir rapidement aux demandes du marché et de satisfaire les attentes des clients selon le concept du « juste à temps ».

Dans l'optique de l'automatisation des techniques de fabrication, cette contrainte de flexibilité condamne *a priori* les solutions du type machine spéciale ou machine automatique à cycle modifiable par des réglages mécaniques généralement longs à réaliser, mais favorise au contraire des **solutions programmables** s'appuyant sur l'**informatique industrielle** et la **commande numérique des machines (MCON)** [JUB94].

Un atelier flexible doit donc être capable de produire, à une cadence relativement élevée (intermédiaire entre **lignes automatisées** et **Ateliers manuels**), des pièces différentes dans un ordre et/ou avec des lots de taille aléatoire.

En résumé, tout l'art de l'automatisation flexible consiste à réaliser le meilleur compromis possible entre la productivité de l'automatisation classique (mais rigide) et la souplesse d'adaptation des tâches manuelles.

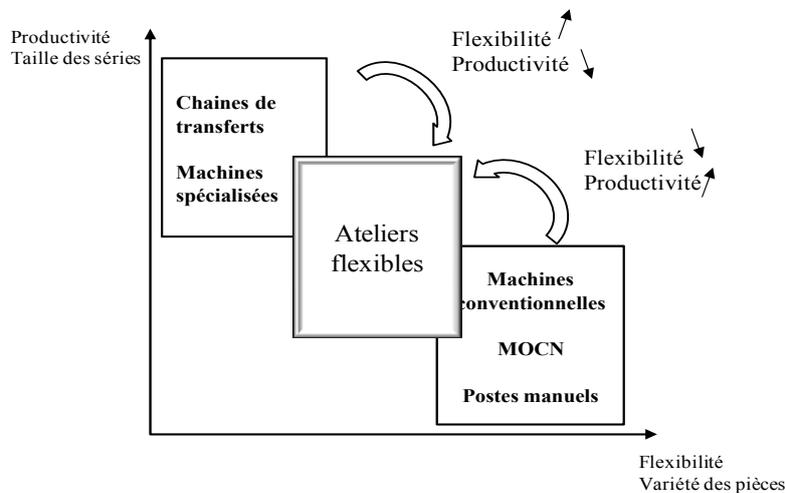


Figure I.2: L'atelier flexible, compromis entre flexibilité et productivité

Dès son apparition, l'automatisation flexible de la production a été étroitement liée à l'usinage. Aujourd'hui encore, ce domaine reste prépondérant parmi les différentes réalisations en place. En effet, d'une part l'usinage est de loin le procédé le plus employé en mécanique ; il est réalisé sur des machines-outils spécifiques vis-à-vis de l'opération, mais universelles, c'est-à-dire non liées à une pièce particulière, telles que : tours, fraiseuses, perceuses...etc.

A. Définition d'un SFP:

« Les ateliers flexibles sont des ateliers rapidement configurables en fonction des contraintes de production. Ils constituent une méthode alternative d'agencement des postes de travail quand la production en ligne, à la chaîne, devient trop complexe à organiser du fait de la diversification des produits » [WEB04].

D'après V. Giard [BES97] un atelier flexible (ou Flexible Manufacturing System, FMS) est « un ensemble de machines reliées par un système de manutention automatique, le tout contrôlé par un système central informatisé ».

Une autre définition est donnée dans [JUB94] « Un atelier flexible permet la production automatique de pièces de types divers et en quantités variables. Les opérateurs n'interviennent pas directement dans le processus de fabrication et limitent essentiellement leurs interventions à l'entretien. L'ordonnancement de la production est géré par un système informatique ».

B. Classification des SFP :

Conformément à la figure I.3, il est possible de classer les systèmes flexibles en trois grandes catégories [JUB94] : les ateliers flexibles spécialisés ou lignes transferts flexibles, les ateliers flexibles complexes et les cellules ou îlots flexibles.

Les ateliers flexibles spécialisés ou lignes transferts flexibles : ces ateliers sont conçus pour la production de variantes dimensionnelles de pièces d'une même famille sur le plan fonctionnel et morphologique. A cette famille correspond une gamme unique (une succession d'opérations bien déterminées) ; les pièces passent **successivement** sur les différentes machines de l'atelier disposées en ligne dans l'ordre ou elles interviennent suivant la logique du « juste à temps ». Le nombre de machine (MOCN) est généralement compris entre 4 et 10 unités.

Les ateliers flexibles complexes : Regroupant plusieurs (5 à 20) MOCN associées à des systèmes de transfert de pièces et de chargement/déchargement totalement ou partiellement automatisés selon les cas. Ces ateliers sont destinés à la production de pièces variées du point de vue de la forme et des dimensions. Aux divers types de pièces correspondent des gammes différentes, ces gammes ont cependant certaines opérations en commun de manière à ce qu'une machine donnée du système puisse intervenir dans la fabrication de plusieurs pièces de la même famille. La progression des pièces dans un atelier flexible de ce type n'est pas prédéterminée. Le logiciel de commande doit assurer une véritable **gestion en temps réel** en fonction des règles de pilotage préalablement établies.

Les cellules ou îlots flexibles : Par rapport aux ateliers flexibles, les cellules sont des systèmes moins complexes, regroupant un petit nombre (2 ou 4) MOCN. A ces machines sont associés, comme pour les ateliers, divers équipements assurant de manière automatique les fonctions de stockage, de transfert, de chargement/déchargement des pièces brutes et finies et de contrôle. Les machines constituant les cellules flexibles sont associées les unes aux autres en fonction de l'ensemble des opérations à réaliser indépendamment de l'appartenance des pièces à une famille bien déterminée.

La classification précédente a seulement pour objectif de situer plusieurs niveaux d'automatisation flexible. Dans la réalité, la situation peut s'avérer un peu plus complexe ; certains systèmes flexibles, exploités dans l'industrie, ont en effet des caractéristiques qui les apparentent à la fois aux cellules et à certaines formes d'ateliers flexibles.

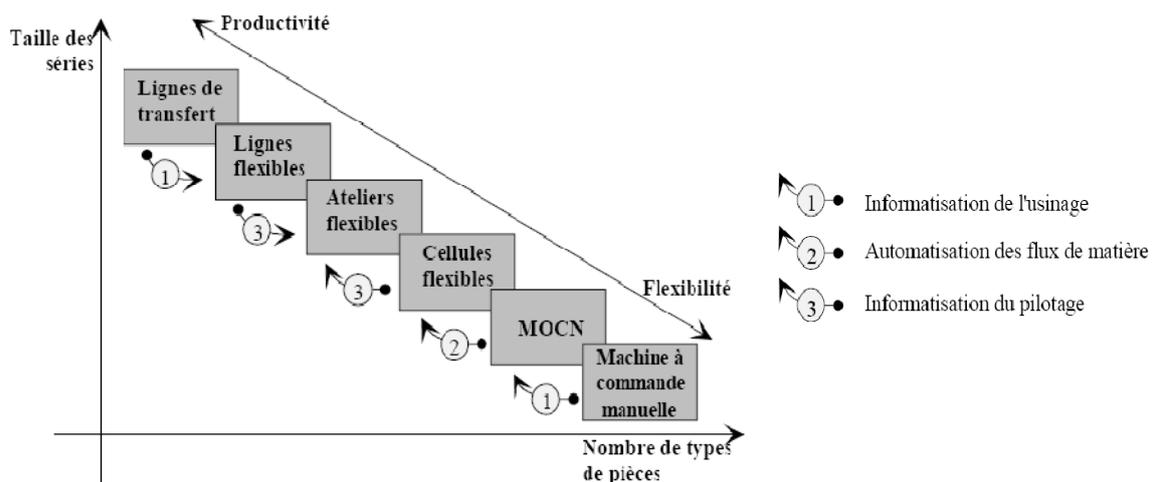


Figure I.3: classification des SFP

C. Eléments constitutifs des SFP :

Les systèmes flexibles de production sont constitués d'un certain nombre d'équipements qui peuvent être divisés en quatre familles : [GHA06]

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

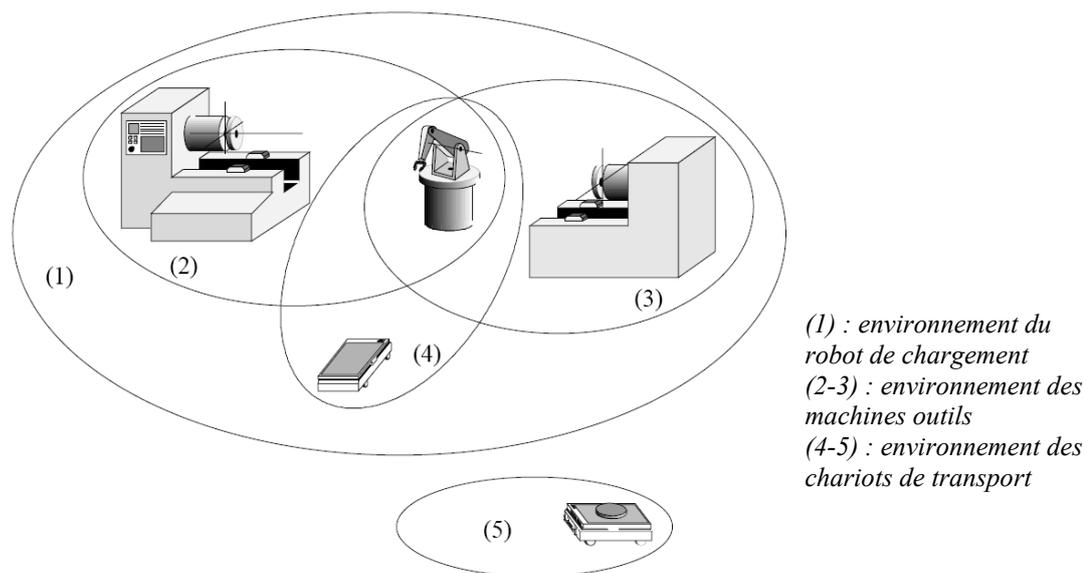


Figure I.4: Composants de la cellule d'assemblage [BES95]

1. Les stations de travail

Une station de travail est composée d'une MOCN, d'un magasin d'outils avec un système automatique de changement d'outils, d'un magasin à palettes et supports, d'un système automatique de chargement/déchargement des palettes dans les machines, d'un système automatique d'inspection et d'un système de contrôle :

- Les machines :

Les machines utilisées dans les SFP, sont, en général, des MOCN. En changeant d'outil et de programme de commande, une machine est en mesure de réaliser une grande variété d'opérations sur des produits de forme, de taille et de matériaux différents.

- Les palettes et supports :

Les produits, solidaires de leurs supports, se déplacent dans le système de production sur des palettes qui servent d'interface entre le produit et la machine.

- Chargeur automatique d'outil :

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

- **Magasin et chargement/déchargement des palettes :**

Lorsque les produits arrivent à la station de travail sur les palettes, ils sont chargés sur la table de travail de la machine à l'aide d'un système automatique de chargement/ déchargement, qui peut être par exemple un robot manipulateur. À l'aide d'un système de détection (code-barres), la machine identifie le produit, charge l'outil et le programme correspondant et procède à la vérification du produit fini.

- **Système automatique d'inspection :**

L'inspection des produits a pour but de s'assurer que ces derniers répondent aux spécifications exigées.

- **Système de contrôle :**

Le système de contrôle a pour mission de coordonner et de superviser le fonctionnement de tous ces équipements.

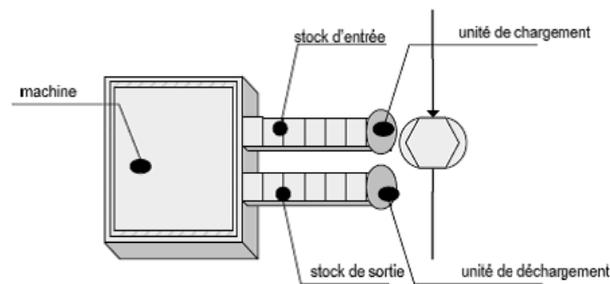


Figure I.5 : schéma d'une station [CAU06]

2. Equipements de transport :

Les équipements de transports servent à déplacer les produits et éventuellement les outils entre les machines suivant les chemins planifiés pendant la conception du SFP. A titre d'exemple, on peut citer : les convoyeurs, les véhicules industriels,...etc.

3. Les systèmes de stockage :

Les aires de stockage peuvent aller du système complètement automatique au stockage manuel.

4. Les systèmes de contrôle :

C'est le système de contrôle qui décide quand et comment les produits doivent se déplacer, être chargés ou déchargés entre les différentes machines. Un système de commande élaboré peut commander et ordonnancer en temps réel le SFP. Les systèmes de contrôle sont composés de trois entités principales :

- Les calculateurs, sont des ordinateurs à base de microprocesseurs ;
- Les logiciels, développés pour chaque type de commande ;
- Les bases de données, les systèmes de transmission d'information, les capteurs etc.

Celui-ci doit être capable de gérer dynamiquement un grand nombre de tâches, d'entités et de ressources. Il doit veiller à la parfaite coordination de tous les composants de l'atelier tout en gérant les aléas apparaissant de façon quasi-permanente au cours de la production (figure I.5).

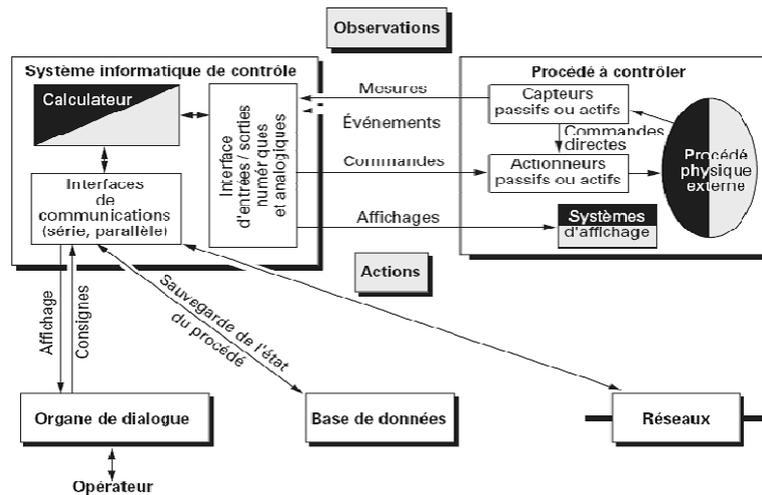


Figure I.6: aperçu sur les systèmes de contrôle des SFP [COT99]

3. Les systèmes de communication :

Pour permettre une coopération entre les différents centres de contrôle, quelque soit la structure, il est nécessaire de prévoir un système de communication entre eux. Pour ce faire, il est nécessaire de concevoir un réseau de communication local (local area network, LAN).

III. L'ordonnancement et les systèmes d'ordonnancement :

Cette section est consacrée à la partie pilotage des systèmes flexibles de production et plus particulièrement à la fonction ordonnancement.

1. Positionnement de l'ordonnancement dans la gestion de production :

Usuellement, le pilotage des systèmes de production est décrit comme étant la superposition de cinq étapes correspondant à des niveaux décisionnels et à des horizons temporels distincts [KIE02]. Comme le montre la figure I.7, le pilotage d'un système de production comprend 02 phases réparties sur 05 niveaux : la *gestion prévisionnelle* et le *pilotage en temps réel* du système de production ; la gestion prévisionnelle concerne les décisions prises hors ligne (avant lancement et exécution), alors que le pilotage en temps réel, concerne les décisions prises en ligne (après le lancement et pendant l'exécution).

Gestion prévisionnelle : nous distinguons trois fonctions de la gestion prévisionnelle : [TRE02]

La planification : Elle détermine le plan directeur de production, compromis entre les objectifs commerciaux, financiers et la production. L'horizon de planification est souvent le moyen ou le long terme. Cette fonction est caractérisée surtout par un horizon temporel étendu et par un degré de finesse des informations traitées très générique.

La programmation : Elle établit un programme prévisionnel de production basé sur le plan directeur de production et sur l'état des stocks. L'horizon de la programmation est souvent le très court terme (la semaine).

L'ordonnancement : Il détermine d'abord les priorités de passage des travaux sur les ateliers (séquencement) et les différentes affectations temporelles des ressources. En fait, cette sous fonction est constituée d'une partie séquencement et d'une autre affectation. L'horizon peut être le court ou le moyen terme.

Pilotage temps réel : nous distinguons deux fonctions pour le pilotage en temps réel : [KIE02]

La conduite, qui est chargée de réaliser la production prévue définie par l'ordonnancement. Elle doit régler tous les problèmes non résolus par le niveau prévisionnel et prendre en compte l'ensemble des contraintes de fabrication en temps réel.

La commande, qui est le dernier niveau décisionnel, directement en relation avec le système physique, car elle a un rôle d'interface et d'interpréteur et traduit un ordre en séquence automatique.

La fonction conduite regroupe les activités décisionnelles qui sont traduites en ordres et transmises au niveau commande. Ce dernier niveau gère la réalisation du processus physique de fabrication et retourne au niveau conduite un ensemble de données de suivi. Cette approche reprend la notion de boucle de rétroaction.

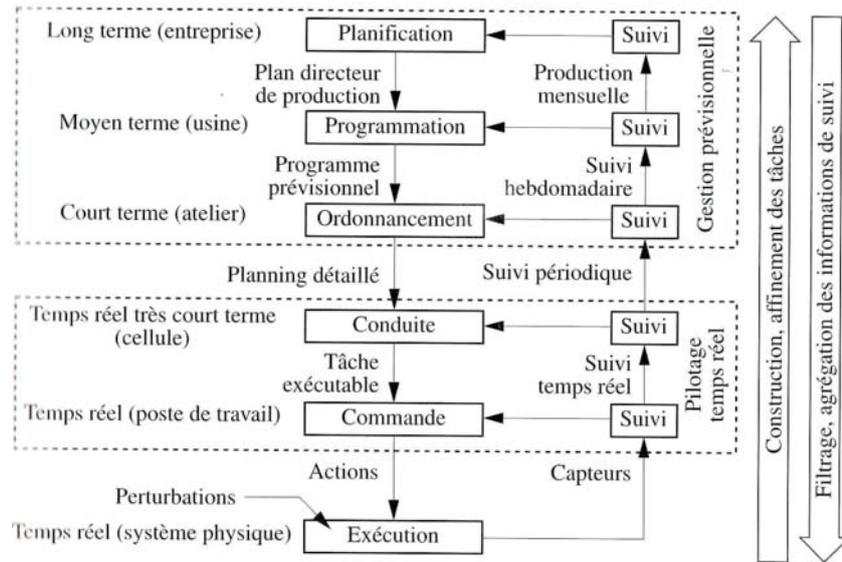


Figure I.7 : Fonctions de pilotage de la production [KIE02]

Cette hiérarchie obéit à un modèle de résolution de problème décomposant les décisions en 03 niveaux, stratégique (portefeuille d'activités, investissement...), tactique (planification) et opérationnel (conduite et suivi quotidien des flux). Dans ce schéma classique, la place de la fonction **ordonnancement** varie entre le niveau tactique (Ordonnancement moyen terme) et le niveau opérationnel (ordonnancement quotidien ou réordonnancement temps réel).

Le rôle de cette fonction est de piloter un ensemble donné de ressources (ateliers, ilots, machines) et de régler les conflits d'utilisation que pose la réalisation d'un ensemble de produits sur un horizon temporel donné [ESQ99].

Le problème est relativement simple dans le cas des chaînes de fabrication où les produits suivent en flots une suite d'opérations continues sans grande variation. Mais il se complique dès que la fabrication se fait en petit lots, que les processus de fabrication sont diversifiés et admettent des alternatives dans la suite des opérations, et/ou que l'on prétend à des modes de production flexibles.

2. Généralités sur l'ordonnancement d'atelier manufacturier :

A. Définition des problèmes d'ordonnancement :

Le problème d'ordonnancement d'ateliers peut se formuler de la façon suivante : « Programmer au mieux l'ensemble des tâches à réaliser, en leur allouant les ressources nécessaires et en fixant leur date de début » [AZZ05].

Le terme « au mieux » donne implicitement à cette définition une fonction d'optimisation, de sorte que le problème ne consiste pas uniquement à déterminer des solutions réalisables, mais comporte aussi la recherche des plus favorables d'entre elles, au regard d'un critère souhaité.

Si l'énoncé de ce problème peut paraître très général et s'appliquer à des domaines très différents comme la mécanique, la construction (suivi de projet) et l'informatique, néanmoins la terminologie qu'ils emploient est essentiellement tournée vers les systèmes industriels de production (machine, job, opération et ressources,...).

Dans le domaine de la gestion de la production, Un problème d'ordonnancement consiste alors à : [GOU03]

- déterminer l'entrée des produits dans le système ;
- trouver un ordre de traitement admissible (qui respecte l'ensemble des contraintes). Le problème peut être surcontraint et donc l'ensemble des solutions admissibles vide ;
- déterminer une planification robuste aux événements aléatoires : dans un contexte avec des événements aléatoires, certaines contraintes peuvent entraîner des blocages ou des conflits d'utilisation d'une ressource critique, on cherche donc à minimiser l'impact des événements aléatoires sur les systèmes.

B. Eléments de base d'un problème d'ordonnancement :

Le problème d'ordonnancement est défini en se basant sur les concepts de tâche, ressource, contrainte et objectif :

(a) **La tâche** : Une tâche i est localisée dans le temps par une date de début r_i et de fin C_i , ou par l'une de ces dates et une durée P_i . Une fabrication décomposable en tâches peut avoir une date de livraison souhaitée di . On note $[r_i, di]$, l'intervalle temporel dans lequel la tâche i doit s'exécuter [ESQ99].

Si les tâches peuvent être interrompues, on parle d'ordonnancement **préemptif** (non préemptif dans le cas contraire). L'ordonnancement d'atelier manufacturier relève en général du cas non préemptif.

(b) **La ressource** : Une ressource k est un moyen technique ou humain requis pour la réalisation d'une tâche, et disponible en quantité limitée : sa **capacité** ak . On distingue plusieurs types de ressources :

- Une ressource est **renouvelable** si, lorsqu'elle est libérée, elle est de nouveau disponible en même quantité. Dans le cas contraire, elle est **consommable**.
- On distingue enfin les ressources **disjonctives** (ou non partageables) qui ne peuvent exécuter qu'une tâche à la fois et les ressources **cumulatives** ou partageables.

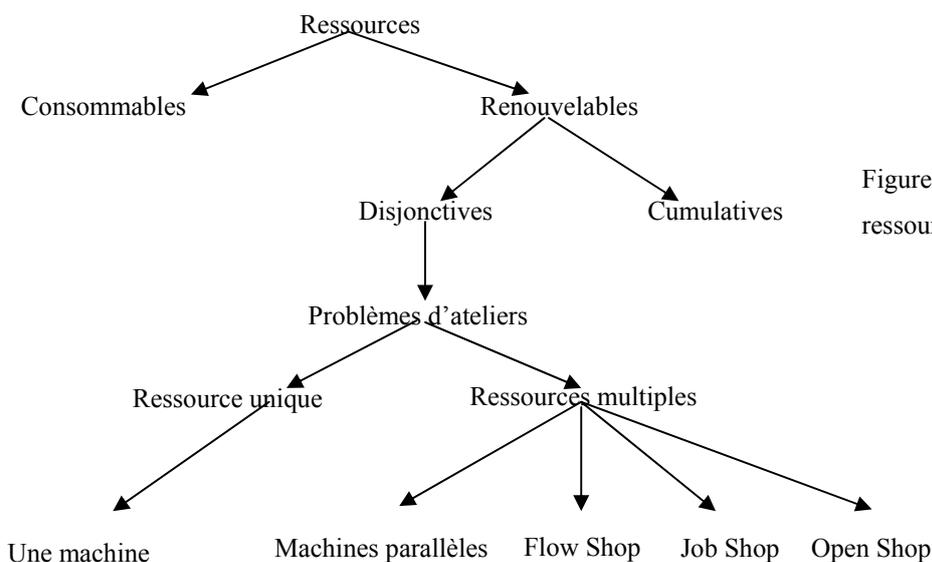


Figure I.8: typologie par les ressources des problèmes

Remarque : Les problèmes non préemptifs à ressources renouvelables disjonctives couvrent une classe importante d'applications : les problèmes d'ateliers qui constituent l'objet de cette étude.

(c) La notion de contrainte : Des contraintes de différents types conditionnent l'admissibilité d'un ordonnancement : [AZZ05]

- contraintes temporelles : respect d'un début au plus tôt, d'une fin au plus tard d'une fabrication...etc. ;
- contraintes d'antériorité (respect de la succession des opérations dans la gamme) ou de cohérence technologique (durée minimale ou maximale entre deux tâches concordance des débuts, des fins de tâches différentes, etc.) ;
- contraintes de ressources :
 - contraintes disjonctives : les tâches doivent être effectuées l'une après l'autre ($i < j$ ou $j < i$),
 - contraintes cumulatives : la somme des besoins en ressources de tâches simultanées doit être inférieure à la capacité des ressources.

(d) Objectifs : Les critères les plus souvent rencontrés dans la littérature pour l'évaluation d'une solution d'un problème d'ordonnancement sont : l'utilisation efficace des ressources (le taux d'utilisation des ressources), le délai global de réalisation des tâches, la minimisation des encours et le respect des dates échues. Les mesures de performance usuelles sont [ESQ99] :

- La durée totale de l'ordonnancement $Cmax = \max \{Ci\} i$ (make-span) ;
- Le plus grand retard algébrique $Lmax = \max \{Li\} i$;
- La plus grande durée de séjour $Fmax = \max \{Fi\} i$;
- La durée moyenne de séjour $\bar{F} = \sum_{i=1}^n F_i$
- Le plus grand retard vrai $Tmax = \max \{Ti\} i$;
- Le retard moyen $\bar{T} = \sum_{i=1}^n T_i$
- Le nombre de tâches en retard $N_T = \sum_{i=1}^n U_i$.

L'objectif de l'ordonnancement vise donc à classer les solutions possibles, voire à guider la méthode d'élaboration de l'ordonnancement utilisée. Il est alors possible de définir un ordonnancement « optimal », c'est-à-dire satisfaisant le mieux l'objectif considéré [AZZ05].

C. Classification des problèmes d'ordonnancement d'atelier :

Dans les problèmes d'ordonnancement d'atelier, il est souvent fait une association entre *tâche* \leftrightarrow *étape* de fabrication d'un produit, et *job* \leftrightarrow *produit* (ou gamme d'un produit) [TRA01]. La gamme décrit une suite d'opérations ou de tâches à effectuer pour réaliser un produit.

[CAU06] présente une typologie de problèmes d'ordonnancement, basés sur la nature des **ressources** et des **gammes** des travaux à réaliser :

- Pour le problème à une machine, les gammes consistent à visiter l'unique machine durant un temps prédéterminé ;

- Pour le problème de flow shop, tous les jobs ont la même gamme. Une gamme consiste à visiter toutes les machines dans un ordre donné ;
- Pour le problème de job shop, chaque job a sa propre gamme. Une gamme consiste à visiter un ensemble de machines dans un ordre donné ;
- Pour le problème d'open shop, chaque job a également sa propre gamme. Une gamme consiste alors à visiter un ensemble de machines dans un ordre à déterminer ;
- Pour le problème à machines parallèles, tous les jobs ont la même gamme. Une gamme consiste à visiter une des machines parallèle, le choix de la machine est à déterminer.

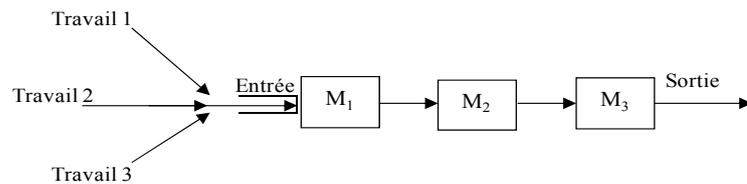


Figure I.9: Flow shop à 3 machines

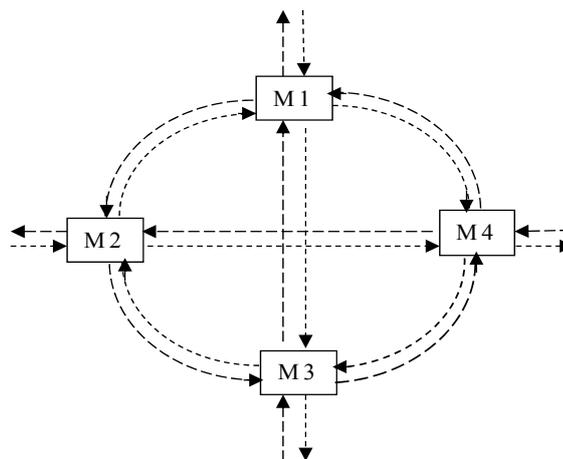


Figure I.10: Exemple d'organisation Job Shop Simple à 4 machines.

La figure ci-dessous (figure I.11) illustre schématiquement les différences et les ressemblances qui existent entre les différentes organisations de production.

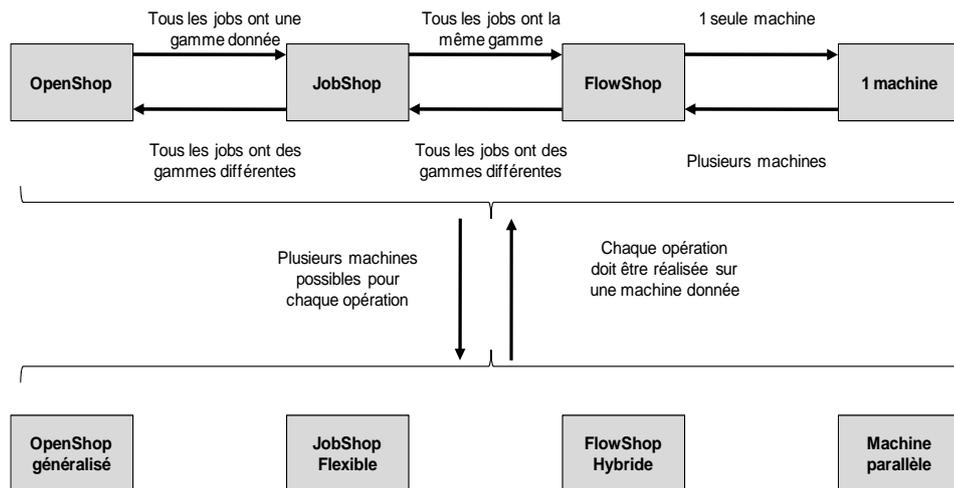


Figure I.11 : classification des problèmes d'ordonnancement [Cau06]

Une autre classification, plus fine, est présentée par Graham et al [DJE07]. Elle se base sur la notation $\alpha/\beta/\gamma$ et permet de caractériser un problème d'ordonnancement de manière précise. Le champ α décrit la structure du problème et se décompose généralement en deux sous champs α_1 et α_2 , le premier indiquant la nature du problème (job shop, flow shop, etc.), le second précisant le nombre de machines. Le champ β décrit les types de contraintes prises en compte. Enfin, le champ γ indique la fonction objectif considérée. A titre d'exemple la notation $J3/\rho_i = 1/Cmax$, indique qu'il s'agit d'un problème d'ordonnancement de type *Job shop*, à 3 machines, dont les tâches ont toutes des durées opératoires unitaires, avec comme objectif la minimisation du *Make span* (*Cmax*).

D. Stratégies d'ordonnancement

Trois stratégies peuvent être adoptées pour l'ordonnancement, à savoir: prédictive, dynamique et réactive [BER91].

Ordonnancement prédictif/statique Sur la base de la description du monde et une prévision des activités à accomplir, on calcule un ordonnancement avant même que son exécution ne soit commencée. L'horizon de temps est fixe, toutes les données sont supposées connues à l'avance et on est dans l'hypothèse que le programme établi subira peu de changements à l'exécution.

L'inconvénient de cette approche réside dans le fait que plus le programme est détaillé, moins il est robuste et à même de supporter des événements imprévus qui risquent de le remettre totalement en considération.

Ordonnancement dynamique L'ordonnancement se fait progressivement en « temps réel », et on choisit, par des règles ou autres moyens, l'activité suivante parmi celles qui se présentent. Cette stratégie repose lourdement sur une connaissance instantanée et fidèle de l'état de l'atelier. Elle permet de tenir compte des imprévus: nouvelles ressources à disposition, nouveaux ordres pressants etc.. Son inconvénient est qu'elle souffre d'effet d'horizon ou myopie.

Ordonnement réactif L'idée est de permettre la révision et la réparation d'un ordonnancement prédictif dans un environnement dynamique suite à une modification (l'arrivée d'un nouveau produit, les pannes de ressources, l'absence de personnel...). Là aussi, il faut que l'état de l'atelier soit accessible pour pouvoir comparer entre le comportement du système réel et celui prévu. Cette comparaison permet de déterminer les changements minimaux qu'il faut opérer sur l'ordonnement prévu en vue de maintenir la stabilité de l'atelier.

Selon les caractéristiques particulières du domaine d'application, on choisira l'une ou l'autre ou encore *une combinaison intelligente des stratégies* précitées. Ainsi un ordonnancement prédictif pourrait être revu lors de son exécution par un ordonnanceur réactif. Le cas extrême étant de tout recalculer.

E. Méthodes de résolution des problèmes d'ordonnement :

Présenter la totalité des méthodes et leurs variantes apparaît difficile au vue du nombre de travaux existants. Aussi nous présentons uniquement les grandes orientations de ces travaux sans toutefois rentrer dans le détail des calculs. Les méthodes de résolution se répartissent en deux catégories principales : les méthodes **d'ordonnement prévisionnel** ou **statique**, **approches d'ordonnement réactif**.

(a) **Ordonnement statique** : En ordonnancement prévisionnel, **les méthodes classiques** ou **optimales** ont pour but de générer un ordonnancement optimal, c'est-à-dire minimisant un des critères présentés (ou une combinaison de plusieurs critères). Les problèmes d'ordonnement ainsi définis sont des problèmes d'optimisation combinatoire. Parmi ces méthodes, on peut citer :

- **Les Procédures par Séparation et Évaluation (PSE)** qui énumèrent par une recherche arborescente un ensemble de solutions, en éliminant les branches de l'arbre de recherche non optimales afin d'éviter la numération exhaustive des solutions (utilisation de borne inférieure et supérieure du critère).
- **Les méthodes basées sur la Programmation Linéaire (PL)**, modélisant les critères et les contraintes comme des fonctions linéaires des variables, et utilisant pour la résolution la méthode du Simplexe.
- **Les méthodes basées sur la Programmation Dynamique (PD)**, elles consistent en une décomposition du problème principal de dimension n , en sous problèmes de dimension $n-1$, que l'on résout en tenant compte à chaque étape des informations issues de la résolution du sous problème précédent.

Dans des problèmes de taille réelle, ces procédures requièrent des temps d'exécution prohibitifs car la plupart de ces problèmes appartiennent à la classe des problèmes NP-difficiles, qui ne peuvent être résolus au moyen d'un algorithme polynomial. Ainsi à côté des algorithmes classiques de l'optimisation combinatoire ont été développés un grand nombre d'**heuristiques** :

- **Les algorithmes gloutons** dans lesquels les décisions d'ordonnement sont prises progressivement, à temps croissant, au fur et à mesure que les ressources se libèrent, grâce à des règles de priorité simples de type SPT (Shortest Processing Time), EDD (Earliest deadline) ;

- **Les méthodes de recherche locale** (tabou, recuit simulé, algorithmes génétiques, etc.) qui, partant d'une solution initiale, définissent un voisinage, qui est ensuite exploré pour trouver des solutions meilleures ;
- **Les méthodes de recherche arborescente tronquée**, proches des PSE, excepté que l'arbre de recherche est volontairement restreint, quitte à perdre des solutions optimales, afin de gagner en temps de calcul.

(b) Ordonnement réactif : Un système d'ordonnement temps-réel est un système qui inclut une méthode qui permet à chaque fois qu'une décision d'affectation doit être prise de proposer une solution en temps réel tout en tenant compte de l'état présent du système et des objectifs de production. Nous distinguons plusieurs approches d'ordonnement réactif issues de l'intelligence artificielle :

- Approches par contraintes ;
- Systèmes expert ;
- Réseaux de neurones ;
- Systèmes multi agents.

La performance de telles méthodes peut être évaluée à l'aide de la simulation.

F. Complexité des problèmes d'ordonnements Open shop :

Les difficultés des problèmes d'ordonnement des systèmes de production sont multiples. On peut les séparer en deux catégories distinctes : les difficultés de la phase de modélisation et les difficultés de la phase d'optimisation. En effet, les systèmes de production complexes présentent en général une double complexité : [GOU03]

- *Une complexité algorithmique* qui se traduit par la recherche d'une solution qui minimise un ou plusieurs critères de performance.
- *Une complexité structurelle et fonctionnelle du système* qui se traduit par la difficulté d'évaluer de manière simple le ou les critères de performance.

Pour les problèmes de type *open shop* ou à cheminement libre, la détermination des gammes, c'est-à-dire fixer un ordre de passage sur les machines pour chaque travail, est un problème supplémentaire à celui qui consiste à trouver une séquence des travaux sur chacune des machines. Dans le cas non préemptif, le $O2 \parallel C_{max}$ (minimisation de la durée totale dans un problème à deux machines) est un des rares problèmes de type Open shop dont la résolution reste polynomiale. Pour la plupart des autres critères ou dès la considération de trois machines, les problèmes deviennent NP-difficiles.

3. Les systèmes d'ordonnement :

Il est intéressant de noter la différence d'approche entre les méthodes de résolution énumérées précédemment et leur mise en œuvre dans des systèmes d'ordonnement. Cette différence provient de la différence des points de vue : les méthodes étudient essentiellement comment résoudre un problème d'ordonnement, alors que les systèmes d'ordonnement se préoccupent aussi du cadre d'application dans lequel le problème se situe [TRA01].

Un système d'ordonnement (*Scheduling System*) est un programme informatique dont la fonction est de produire des solutions efficaces pour l'allocation (ou la re-allocation) de tâches

sur les ressources tout en respectant certains objectifs et contraintes. Au-delà de l'aspect logiciel, ces systèmes prennent en compte des contraintes pratiques spécifiques ignorées des méthodes plus théoriques [TRA01].

Les systèmes d'ordonnancement, souvent, se préoccupent essentiellement de la planification à *court terme* mais peuvent englober aussi les effets de *la conduite des processus* qui, quant à elle, se fait en temps réel. Leur tâche est de décider à quel moment on exécutera une activité parmi celles en compétition ainsi que de déterminer la séquence des opérations. Ceci se fait en connaissance des *ressources* (en personnel et en centres de machinage), des *procédés et plans* d'opérations ainsi que des *contraintes* impliquées.

Nous pouvons recenser plusieurs approches concernant les différentes structururations possibles pour un système d'ordonnancement. Elles se positionnent par rapport à la décentralisation de la capacité de décision et d'information :

Structure centralisée : Cette approche est très classique. Elle se caractérise par une conduite localisée au sein d'une ressource unique qui gère seule, en temps réel, les événements qui surviennent tout au long de la production. Les avantages sont ceux qui proviennent de la centralisation des informations et des capacités de décision. Les inconvénients se situent au niveau de l'accroissement de la complexité des informations et des décisions qui peuvent de plus en plus difficilement être prises en compte dans ce type d'approche.

Structure hiérarchisée : Dans ce type d'approche, chaque niveau coordonne les unités de pilotage du niveau inférieur, et ce jusqu'au niveau le plus bas. Donc, chaque niveau a des relations de dépendance vis à vis du niveau supérieur et de dominance vis à vis du niveau inférieur. Chaque décision est élaborée au niveau où un problème est détecté. Les niveaux inférieurs traitent cette décision comme une contrainte et transmettent en retour une information de suivi au niveau supérieur.

Structure distribuée : Elle est fondée sur une distribution complète de la décision sur l'ensemble des ressources pilotant un centre de production. Le contrôle d'une telle structure est beaucoup plus complexe en raison de sa modularité, par rapport à une structure hiérarchique qui est plus rigide mais plus facilement maîtrisable. Cette structure présente un niveau élevé de réactivité et de flexibilité.

Structure distribuée supervisée : Un pilotage distribué supervisé se caractérise par un ensemble d'entités coopérant sous le contrôle d'une entité superviseur dont le rôle est d'imposer, de conseiller ou de modifier une décision afin de respecter un objectif plus global. Un superviseur, qui possède une vision plus globale du processus de production est rajouté à la méthode distribuée.

Cette approche est un compromis entre les structures distribuée et centralisée. Elle permet une gestion globale et efficace par la centralisation de contrôle d'une part, et une meilleure réaction aux perturbations par la distribution des capacités de décision d'autre part.

Structure coordonnées : L'approche coordonnée correspond à un ensemble de structures hiérarchisées où une coopération est possible au sein d'un même niveau. Ces structures accroissent l'autonomie et la capacité de décision au sein de chacun de ces niveaux en proposant des mécanismes de coopération qui vont aider à la résolution locale des problèmes décisionnels sans systématiquement se référer aux niveaux supérieurs, ce qui conduit à un gain de réactivité.

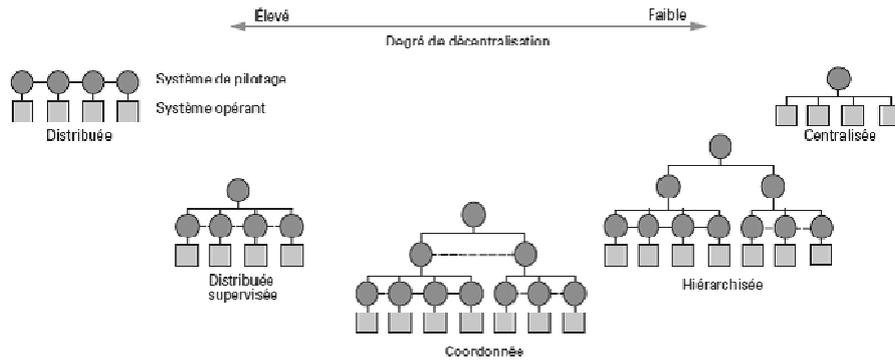


Figure I.12 : Les différentes structures des systèmes d'ordonnancement [TRE02]

L'ordonnancement est donc centralisé s'il s'exécute sur une architecture centralisée ou sur un site privilégié de l'architecture distribuée qui contient l'ensemble des paramètres des tâches. Il est réparti lorsque des décisions d'ordonnancement sont prises sur chaque site par un ordonnancement local après une éventuelle coopération pour effectuer un ordonnancement global [COT99].

IV. Conclusion :

La maîtrise de l'ordonnancement est d'un intérêt capital pour les entreprises, sans cesse confrontées à des impératifs de productivité, de flexibilité et de réactivité. L'étude des problèmes d'ordonnancement est également d'un intérêt théorique toujours renouvelé pour les chercheurs, car il n'existe pas encore de méthode de résolution à la fois générale et de faible complexité algorithmique en raison de la nature fortement combinatoire de ces problèmes.

Dans le chapitre suivant, nous focaliserons notre attention sur les méthodes issues de l'intelligence artificielle distribuées, basées sur le paradigme d'agent en tant que nouvelle approche de modélisation et de résolution des problèmes d'ordonnancement.

Chapitre II : Modélisation multi agents de la cellule d'assemblage

- I. Introduction

- II. Les systèmes d'ordonnancement distribués
 - 3. Nécessité d'une approche distribuée
 - 4. Présentation des systèmes multi agents
 - H. Notions d'agents
 - I. Définition d'un système multi agents
 - J. Les types d'agents
 - K. Modèles d'agent
 - L. Les formes d'interaction entre agents
 - M. La communication entre agents
 - N. Méthodologie de conception des systèmes multi agent

- III. Modélisation multi agents
 - 6. Rappel du problème étudié
 - 7. Description du système cyclique d'assemblage
 - 8. Approche de modélisation multi agents
 - 9. Protocole de coopération inter ressources :
 - C. Les primitives de coopération (schéma philosophie générale)
 - D. Les étapes du protocole de communication (schéma étapes)
 - 10. Architecture du centre de décision ordonnanceur (schéma PC)

- IV. Conclusion

I. Introduction :

Actuellement, les applications industrielles temps réel apparaissent dans tous les secteurs économiques : systèmes de communication, systèmes de contrôle et de commande des procédés industriels,...etc., ce qui implique la nécessité d'une méthodologie dans la conception de tels systèmes.

Les approches à base d'objets, très utilisées dans le domaine du Génie logiciel, représentent l'application sous une forme abstraite. Le passage d'une approche orienté objets à une approche orienté agents est caractérisé par l'ajout de nouvelles propriétés telles que la propriété décisionnelle des entités et les processus de communication inter agents.

Le domaine des systèmes multi agents est issu initialement des recherches entreprises en intelligence artificielle. Ils fournissent une nouvelle méthode pour analyser, désigner et implémenter des applications sophistiquées. Ils font partie de l'IAD (intelligence artificielle distribuée) en bénéficient aussi d'autres disciplines : sciences cognitives, sociologie et psychologie sociale.

La première partie de ce chapitre vise à présenter un panorama simplifié des systèmes multi agents. Cette présentation décrit tout d'abord un système multi agents, puis détaille les propriétés minimales que doivent présenter de tels systèmes. Cette même approche sera adoptée pour l'élaboration du model conceptuel décrivant le système de pilotage de la cellule d'assemblage étudiée. Ce modèle sera décrit dans le cadre de la deuxième partie de ce chapitre.

II. Les systèmes distribués :

Nous aborderons dans cette section les concepts relatifs à l'intelligence artificielle distribuée et plus particulièrement aux systèmes multi agents en tant qu'approche de modélisation permettant l'implémentation distribuée des problèmes complexes.

1. Nécessité d'une approche distribuée :

Initialement, le domaine de l'Intelligence Artificielle (IA) cherchait à décrire et à résoudre des problèmes complexes identifiés par des experts. Dans ce domaine, il est possible de construire des programmes informatiques, en particulier des systèmes à base de connaissances, capables d'exécuter un nombre important de tâches en centralisant « l'intelligence » au sein d'un système unique. Cependant, l'IA a été confrontée à de nombreux problèmes théoriques, en particulier celui lié à l'explosion combinatoire.

L'apport de l'Intelligence Artificielle Distribuée (IAD) permet d'éviter de manipuler de nombreuses connaissances dans une seule entité en répartissant ces dernières sur plusieurs entités intelligentes. Comme le soulignent Bond et Gasser, l'intelligence Artificielle Distribuée présente les avantages suivants : [BAR03]

- L'IAD est bien adaptée à la distribution de problèmes spatiaux, logiques...etc.
- Les processus distribués entre différents ordinateurs augmentent la vitesse de calcul et de raisonnement.

- Dans certains cas, les systèmes confèrent aux agents individuels des ressources limitées pour résoudre les problèmes, la coopération et la coordination sont essentielles à la résolution de ces problèmes.

Une des branches issue de ce domaine, les Systèmes Multi-Agents (SMA), permet d'introduire dans un système, un ensemble d'individus (ou agents) dotés de connaissances, d'intentions et de capacités d'évolution différentes. Ces agents sont capables d'interagir entre eux, l'intelligence de tels systèmes se révèle être une expression de l'**activité collective**.

2. Présentation des systèmes multi agents :

A. Notion d'agent :

L'une des premières définitions du concept d'agent est donnée par J. Ferber :

« *Un agent est une entité réelle ou abstraite qui est capable d'agir sur elle-même et sur son environnement, qui dispose d'une représentation partielle de cet environnement, qui, dans un univers multi-agents, peut communiquer avec d'autres agents, et dont le comportement est la conséquence de ses observations, de sa connaissance, et des interactions avec les autres agents.* » [FER95]

En d'autres termes, un agent est une entité physique ou virtuelle qui : [DRA98]

- est capable d'agir dans un environnement ;
- peut communiquer directement avec d'autres agents ;
- est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction qu'elle cherche à optimiser) ;
- possède des ressources propres ;
- est capable de percevoir (mais de manière limitée) son environnement ;
- possède des compétences et offre des services ;
- 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.

Les propriétés d'un agent qui le distinguent d'un objet sont, au nombre de cinq: [GIA07]

- *Pro-activité* : Les objets sont réactifs tandis que les agents peuvent être autonomes.
- *Langage de communication* : Les objets utilisent une communication restreinte à de simples messages tandis que les agents peuvent utiliser un langage de communication entier.
- *Adaptation* : Les objets sont stimulés tandis que les agents peuvent apprendre tout seul.
- *Concepts mentaux* : Un agent, contrairement à un objet, est capable d'utiliser des concepts mentaux comme les croyances, les désirs et les intentions.
- *Notion de localisation* : Contrairement à un objet, les agents peuvent être affectés à des lieux.

Une approche orientée agent ne se limite pas à un ensemble de nouveaux concepts; elle propose plutôt une nouvelle manière de penser qui conduit à un nouveau processus de

décomposition des problèmes. Avec cette approche, les problèmes abordés peuvent être modélisés par une collection d'agents qui prennent chacun en charge une tâche spécifique. La résolution du problème est conduite par la manière dont interagissent les différents agents.

Les capacités de l'agent sont de trois ordres qui répondent aux trois phases générales de réalisation d'une tâche :

- Une phase de perception : l'agent perçoit son environnement et met à jour ses représentations internes de l'environnement et des autres agents ;
- Une phase de cognition : l'agent détermine ce qui est à faire (sa tâche) et décide quand et comment le faire. Une large palette de types de raisonnement est utilisée et implémentée, les extrêmes étant les raisonnements réactifs (réflexes) et cognitifs (de haut niveau) ;
- Une phase d'action : la réalisation effective des actions qui ont été décidées.

Ces trois phases forment un cycle par le bouclage (voir figure II.1) de la phase action sur la phase perception. En effet, l'environnement évolue continuellement et offre donc, à tout moment, au système de nouvelles données à percevoir et de nouvelles tâches à réaliser.

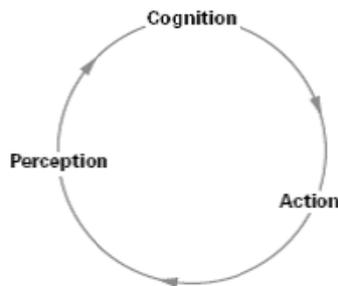


Figure II.1 : Les 3 phases de réalisation d'une tâche par un agent [MAN02]

B. Définition d'un système multi agents (SMA) :

Un agent est destiné à interagir avec d'autres agents au sein d'un même groupe, dénommé système multi agents (SMA) (voir figure II.2).

Un système multi agents est « *un ensemble d'entités qui coordonnent leurs connaissances, buts, expériences et plans pour agir ou résoudre des problèmes, incluant le problème de la coordination inter agent lui-même* » ou encore « *Un monde artificiel peuplé de processus interagissant est appelé système multi agents* » [MAN02].

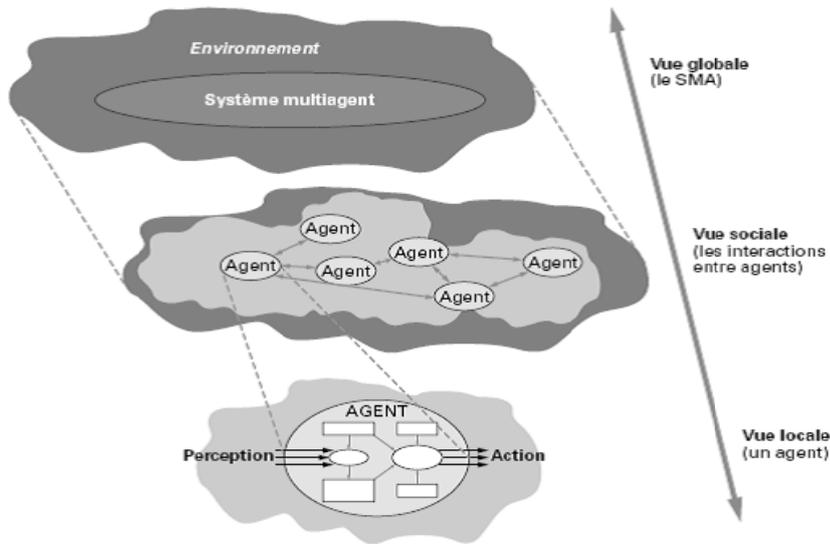


Figure II.2: Système multi agents vu selon différents niveaux de détail [MAN02]

L'ensemble des agents forme un groupe parce qu'ils interagissent. Selon les objectifs qu'ils poursuivent et leurs capacités à les atteindre, l'interaction entre les agents peut être de type *coopératif*, *conflictuel*, ou tout autre type de relation intermédiaire entre ces deux extrêmes.

Alors que la description d'un agent donne une vision locale du système, la description du SMA offre une vue globale et externe sur le système. Pour obtenir une vision complète, il faut ajouter la description intermédiaire qui concerne l'aspect « social », interne au système, des interactions entre les agents. C'est par l'intermédiaire de cette couche d'interactions que les comportements individuels interférents et composent le comportement global du système.

C. Les types d'agents :

Il existe un grand nombre de typologies d'agents, nous retiendrons celle donnée par Ferber qui distingue deux grands types d'agents : réactifs et cognitifs.

- **Agent réactif** : Ces agents se basent sur des raisonnements réflexes ou réactifs. Ces raisonnements caractérisent l'activité de l'agent dans laquelle les actions sont exécutées rapidement. Un agent réactif met en œuvre un raisonnement simple composé de quelques règles de comportement décrites par des paires « conditions-actions » (figure II.3).

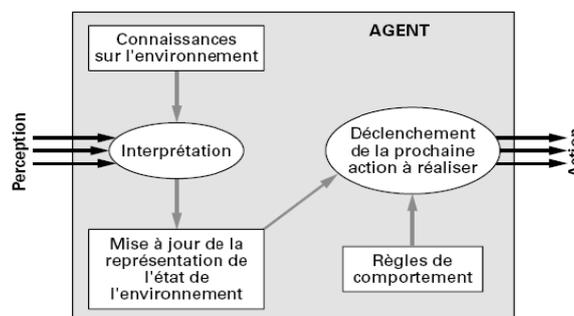


Figure II.3 : architecture d'un agent réactif [MAN02]

La structure possède, indistinctement, à la fois un processus de perception et un processus de raisonnement. Si les informations directement perçues de l'environnement correspondent aux conditions d'une capacité, alors les actions résultant de cette capacité peuvent être sélectionnées.

- **Agent cognitif** : Les agents (à comportements) cognitifs basent leur raisonnement sur un ensemble de connaissances sur leur environnement, sur les autres agents et sur eux-mêmes.

Le comportement de ces agents s'avère en un sens plus « intelligent », mais également plus lent ; ce raisonnement complexe limite également le nombre d'agents capables d'interagir efficacement au sein d'un groupe donné. Dans ce type d'approche, la structure d'un agent est basée sur une décomposition fonctionnelle ou modulaire (voir figure II.4). Ces modules réalisent des fonctions particulières. Ces différents modules interagissent en effectuant des échanges d'informations de manière explicite.

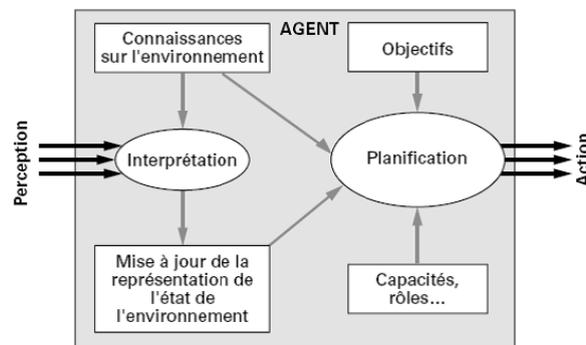


Figure II.4: architecture d'un agent cognitif [MAN02]

Les agents cognitifs peuvent donc travailler de manière relativement indépendante. Les tâches qu'ils accomplissent sont complexes au regard des facultés plus élémentaires des agents réactifs.

D. Modèles d'agent :

Le terme comportement est central dans la définition et la modélisation d'un agent. Tous les modèles existants dans la littérature s'attachent à prouver les bien-fondés de leurs choix conceptuels sur la base desquels ces comportements sont élaborés. Nous nous limitons ici à deux types de modèles qui sont : le modèle agent BDI et le modèle agent AOP.

- **Agent BDI** : L'architecture BDI s'articule sur 3 concepts [BOU07] :

Croyances (Beliefs) : ce que l'agent connaît (ou croit connaître) de l'environnement. Les croyances de l'agent sont des informations qui peuvent s'avérer fausses par la suite. Lorsque les informations considérées par l'agent sont nécessairement vraies, on parle alors de connaissances. Les croyances peuvent changer au fur et à mesure que l'agent, par sa capacité de perception ou par l'interaction avec d'autres agents, recueille plus d'informations.

Désirs (Desires) : ce que l'agent aimerait atteindre ; but à long terme. Notons que ces buts peuvent être propre à l'agent (et donc conflictuels avec d'autres agents) ou commun à un groupe d'agents (ce sont des « **buts collectifs** »).

Intentions : L'agent adopte un ensemble d'intentions pour réaliser un désir. Cet ensemble d'intentions conduit l'agent à exécuter des actions afin de réaliser ses objectifs, en fonction de ses propres croyances.

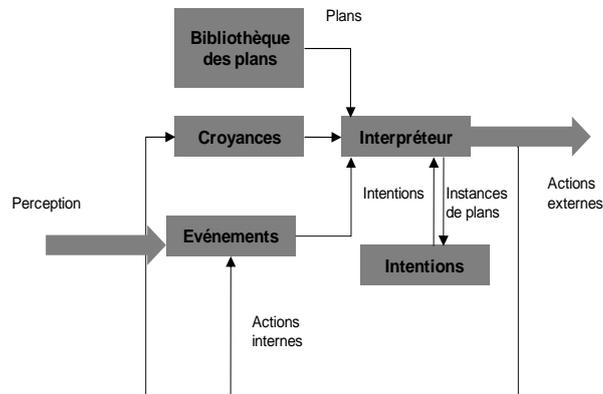


Figure II.5 : Une architecture BDI [BOU07]

Les architectures BDI expliquent le comportement d'un agent (cognitif) par la propension de ce dernier à satisfaire des désirs ou buts, se traduisant par des intentions (plans d'action intermédiaires) à partir de ses croyances (connaissances supposées vraies d'un agent sur lui-même et son environnement) [TRA01].

Un filtre permet ensuite de choisir entre les intentions qui sont en accord avec les croyances et les désirs. Un raisonnement est établi pour déterminer comment arriver à un état donné. Enfin une bibliothèque de plans permet à l'agent de construire les étapes nécessaires à la réalisation du but.

Le modèle BDI est intéressant non seulement car il est intuitif, mais aussi car il fournit une bonne décomposition fonctionnelle, qui identifie clairement les sous-systèmes nécessaires à la conception d'un agent. Néanmoins, la principale difficulté reste toujours l'implémentation efficace de ces sous-systèmes.

- **Approche AOP** : La programmation orientée agents a été proposée par Yoav Shoham en 1993 comme un nouveau paradigme de programmation. Dans cette approche, un agent est perçu comme étant « une entité dont l'état est considéré comme constitué de composantes mentales telles que les croyances, compétences, choix et engagements » [TRA01].

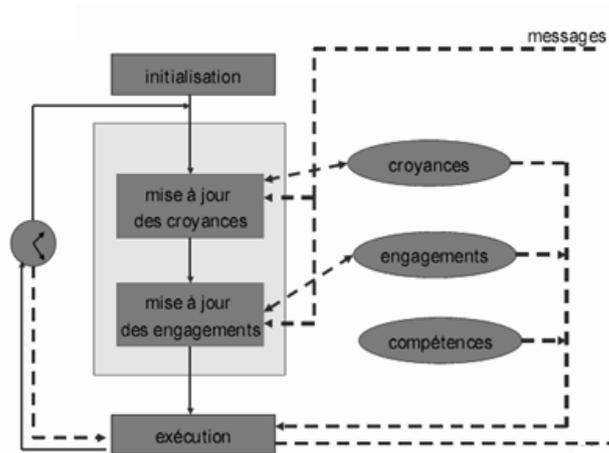


Figure II.6 : Architecture AOP [GAN03]

Les **engagements** sont les actions qu'un agent promet à un autre de réaliser et sont à la base de toute coopération. Les croyances définissent, comme pour les architectures BDI, les connaissances d'un agent.

La plupart des SMA proposent des architectures d'agents particulières empruntant des concepts à l'une ou plusieurs de ces approches. Les architectures d'agents reflètent les besoins particuliers d'un domaine d'application donné : il est en effet très difficile voire impossible d'élaborer une architecture générale et universelle [TRA01].

E. Les formes d'interaction entre agents :

Ferber donne la définition suivante de l'interaction :

« Une interaction est une mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques. Les interactions s'expriment ainsi à partir d'une série d'actions dont les conséquences exercent en retour une influence sur le comportement futur des agents [...] Les interactions sont non seulement la conséquence d'actions effectuées par plusieurs agents en même temps, mais aussi l'élément nécessaire à la constitution d'organisations sociales. C'est par les échanges qu'ils entretiennent, par les engagements qui les lient, par l'influence qu'ils exercent les uns sur les autres que les agents sont des entités sociales et que de nouvelles fonctionnalités peuvent émerger de ces systèmes d'actions mutuelles. » [FER95].

Les situations d'interactions sont nombreuses et diverses: l'échange de données entre serveurs informatiques, la mise en commun des compétences de plusieurs spécialistes pour porter un diagnostic, l'utilisation d'une imprimante par deux programmes simultanément, la répartition des charges sur plusieurs processeurs sont autant d'exemples de situations d'interactions [FER95].

Les principales situations d'interaction peuvent être classées par rapport à trois critères (voir tableau II.1): les objectifs ou intentions des agents, les relations que les agents entretiennent

envers les ressources qu'ils possèdent ainsi que les moyens (ou compétences) dont ils disposent pour parvenir à leurs fins.

Buts	Ressources	Compétences	Types de situations	Catégorie
Compatible	Suffisantes	Suffisantes	Indépendance	Indifférence
Compatible	Suffisantes	Insuffisantes	Collaboration simple	coopération
Compatible	Insuffisantes	Suffisantes	Encombrement	
Compatible	Insuffisantes	Insuffisantes	Collaboration coordonnée	
Incompatible	Suffisantes	Suffisantes	Compétition individuelle pure	Antagonisme
Incompatible	Suffisantes	Insuffisantes	Compétition collective pure	
Incompatible	Insuffisantes	Suffisantes	Conflits individuels pour des ressources	
Incompatible	Insuffisantes	Insuffisantes	Conflits collectifs pour des ressources	

Tableau II.1: Classification des situations d'interaction [Fer95]

Les interactions entre les agents ont lieu soit parce qu'il y a conflit, soit parce qu'il y a coopération pour réaliser une tâche commune. La coopération, étant l'objet principal de notre étude, elle sera détaillée dans cette section.

Les méthodes de coopération : selon [FER95] il existe trois mécanismes de coopération

- **La collaboration par partage de tâches et de ressources :** La collaboration consiste à travailler à plusieurs sur un projet, une tâche commune. Nous appellerons collaboration l'ensemble des techniques permettant aux agents de se répartir des tâches, des informations et des ressources de manière à réaliser une œuvre commune. Résoudre un problème de collaboration consiste donc à répondre à la question « qui fait quoi? » par rapport à un travail donné.
- **La coordination :** La phase de coordination d'actions est directement impliquée dans la définition de l'ordre des actions à effectuer. Elle vise à synchroniser les actions dans le temps en exploitant un référentiel temporel commun, et à gérer la cohérence des actions individuelles par rapport à l'ensemble des activités.
- **La résolution de conflits par arbitrage et négociation :** l'arbitrage et la négociation sont deux des moyens utilisés par les systèmes multi agents pour résoudre les conflits et empêcher que des désaccords entre individus ne se transforment en luttes ouvertes et que le système dans son entier ne dégrade ses performances. L'arbitrage conduit à la définition de règles de comportement qui agissent comme des contraintes sur l'ensemble des agents, mais dont le résultat global a pour effet de limiter les conflits et de préserver à la fois les individus mais surtout les sociétés d'agents.

Les fonctions et les méthodes de coopération sont bien entendu liées entre elles. L'augmentation des performances du système multi agents nécessite le regroupement de leurs ressources et de leurs compétences. Mais en retour ceci peut poser des problèmes tels que : la répartition des tâches, l'augmentation du nombre de conflits ainsi que le manque de coordination. Les méthodes d'allocation des tâches, de coordination d'action et de résolution de conflits servent à résoudre ces problèmes et participent de ce fait à l'augmentation des performances du groupe. Enfin, la réalisation effective de ces méthodes de coopération

suppose l'existence de mécanismes de communication ainsi que la définition des structures internes nécessaires aux agents pour qu'ils interagissent.

F. La communication entre agents :

La communication est la forme d'interaction consciente la plus élémentaire dans un SMA et constitue l'ossature sur laquelle des modes d'interaction plus évolués s'appuient (coordination, collaboration, négociation). En effet, de nombreux travaux considèrent la communication comme un vecteur nécessaire à la coopération [KIE02]. La communication est importante car elle permet à des sociétés d'agents d'effectuer des tâches qu'un seul agent ne pourrait accomplir seul : en communiquant, les agents peuvent échanger des informations et coordonner leurs activités.

La communication dans un SMA suppose l'existence d'un *médium physique* (un réseau informatique par exemple), d'un *médium linguistique* (langage d'expression ou code) et un *mode de diffusion* [FER95]. Le médium physique dépendant davantage de choix techniques d'implémentation (architecture du réseau, types de connections, protocoles de communication) nous insistons sur les deux derniers points.

Le médium linguistique nécessite la spécification d'un langage commun à un ensemble d'agents leur permettant d'interpréter un message. Les recherches en communication inter-agents ont mis en valeur l'apport de la théorie des actes de langages. Le langage KQML (*Knowledge Query Manipulation Language*), largement utilisé dans les SMA [FER95], ainsi que le langage ACL (*Agent Communication Language*) développé par la FIPA (*Foundation for Intelligent Physical Agents* [WEB03]), reposent sur cette théorie. Brièvement, chaque acte de discours (i.e. un envoi de message dans un SMA) est considéré comme une action convoyant des informations sur l'intention de son émetteur et le résultat attendu. Il existe plusieurs types d'actes de langage. D'après Searle (Searle 1979) et Vanderveken (Vanderveken 1988) on peut distinguer les principaux types d'actes suivants: [FER95]

1. **Les assertifs** servent à donner une information sur le monde en affirmant quelque chose.
2. **Les directifs** sont utilisés pour donner des directives au destinataire.
3. **Les promissifs** engagent le locuteur à accomplir certains actes dans l'avenir.
4. **Les expressifs** servent à donner au destinataire des indications concernant l'état mental du locuteur.
5. **Les déclaratifs** accomplissent un acte par le fait même de prononcer l'énoncé.

Ces actes de langages sont introduits par un verbe appelé **performatif** (exemple : *affirmer, questionner, demander de faire, promettre, ordonner, prévenir,...*). La notion de performatif est utile pour les systèmes multi-agents, puisque ce sont les performatifs qui indiqueront les différents types d'actes de langage que peuvent émettre et interpréter les agents (voir tableau II.2). Ces actes de langage ne sont pas isolés mais constituent une séquence d'intervention établie entre agents. Les actes de langages constituent les unités élémentaires permettant d'analyser les conversations (ou protocoles de communication) [FER95].

Type de performatif	Exemple de performatif	Catégorie
Demande	DemanderFaire	Directif
Interrogation	Questionner	Interrogatif
Affirmation	Affirmer	Assertif
Offre de service	OffrirService	Promissif
Indication de compétences	SaitFaire	Expressif
Proposition d'hypothèse	ProposerHypothèse	Expressif

Tableau II.2 : Quelques actes de langages élémentaires [Fer95]

Les actes de langages ainsi que les structures de message propres aux langages KQML et ACL sont décrits dans l'annexe 1 et l'annexe 2.

Les modes de diffusion se divisent en 2 grandes catégories [TRA01] : les modes anonymes et les modes nominatifs.

Le mode de diffusion anonyme peut s'effectuer au travers d'un tableau noir (ou Blackboard System). A l'opposé, dans le mode de diffusion nominatif ou par message, un agent envoie personnellement un message à chaque agent identifié dans sa base d'accointances (les agents connus par celui-ci). Aussi est-il nécessaire de prévoir un mécanisme permettant à chaque agent de savoir comment contacter un agent spécifique (i.e. de connaître son « adresse »).

G. Méthodologie de conception des systèmes multi agents :

De nombreuses méthodologies d'analyse et de conception de systèmes multi-agents s'inspirent de méthodologies de modélisation existantes (issues de l'ingénierie des connaissances ou Orientées Objets). Parallèlement, la maturité du paradigme agent a favorisé le développement de méthodologies directement centrées sur les agents. Celles-ci offrent une prise en compte explicite des rôles, des comportements autonomes, des interactions et de l'organisation sociale [LAB06].

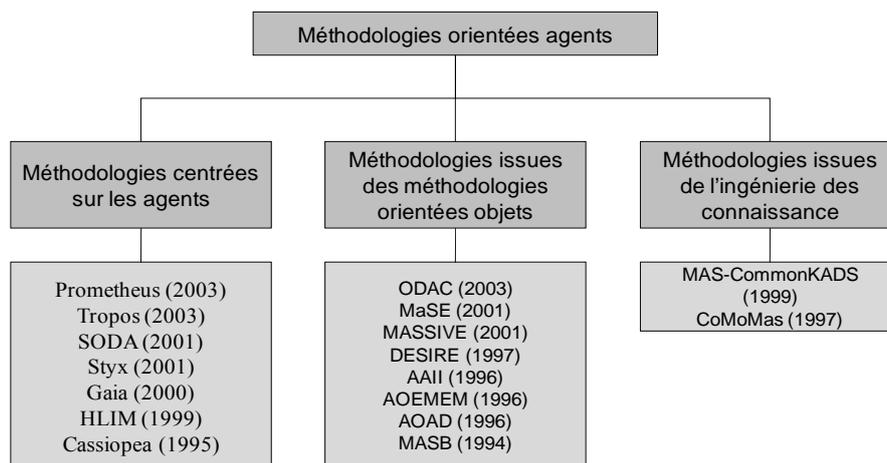


Figure II.7 : Les méthodologies orientées agents [LAB06]

Méthodologies issues de l'ingénierie des connaissances : Les méthodologies orientées agents basées sur l'ingénierie des connaissances s'intéressent à la manipulation et la modélisation de la connaissance dans les modèles d'agents et dans le système multi-agents. Ces méthodologies s'inspirent généralement de la méthodologie CommonKADS issue de l'ingénierie des systèmes à base de connaissances [LAB06]. Les phases de conception du

système multi-agents à partir de cette méthodologie reposent sur la conception des modèles suivants : [LAB06]

- **Modèle d'Agent** (Agent model): décrit les capacités et les caractéristiques des agents (raisonnements, connaissances, buts, services, etc.).
- **Modèle de tâche** (Task model): décrit les tâches pouvant être réalisées par les agents (les buts) ainsi que la décomposition des tâches et les méthodes de résolution de problèmes.
- **Modèle d'Expertise** (Expertise model): décrit les connaissances des agents pour la réalisation de tâches.
- **Modèle de Coordination** (Coordination model): décrit les interactions entre agents et les capacités conversationnelles.
- **Modèle d'Organisation** (Organisation model): décrit l'organisation du système multi-agents.
- **Modèle de Communication** (Communication model): décrit les interactions entre les utilisateurs et les agents pour le développement d'interfaces adaptées.
- **Modèle de Conception** (Design model): décrit le système multi-agents.

Méthodologies issues des méthodologies orientées Objets : Parmi les méthodologies orientées objets, nous présenterons la méthode : MaSE (*Multiagent Systems Engineering*).

La méthodologie MaSE est composée de deux phases principales : analyse (identifier les buts, appliquer les cas d'utilisation et perfectionner les rôles) et conception (créer les classes d'agent, construire les conversations, assembler les classes d'agent et Concevoir le système). Les étapes associées à cette méthodologie sont les suivantes : [LAB06]

- **Identifier les buts** : identifie les buts du système afin de les structurer et de les représenter sous la forme d'une hiérarchie.
- **Appliquer les cas d'utilisation** : identifie les rôles et leurs interactions selon deux sous-étapes : cas d'utilisation et diagrammes de séquence. Les cas d'utilisation définissent le comportement général du système, ses fonctionnalités, son environnement (utilisateurs et acteurs) et les rôles du système. Les diagrammes de séquence représentent les messages échangés entre les rôles.
- **Perfectionner les rôles**: identifie la décomposition fonctionnelle du système.
- **Créer les classes d'agent**: identifie le système multi agents par la description des architectures d'agents et de leurs liens conversationnels. Une classe d'agents précise les rôles qui lui sont assignés.
- **Construire les conversations**: permet de définir les protocoles de coordination entre deux agents.
- **Assembler les classes d'agent** : spécifier l'architecture interne des agents.
- **Concevoir le système**: spécifier la distribution des agents selon l'architecture physique du système.

Cette méthodologie s'inspire fortement des diagrammes proposés dans UML (Unified Modeling Language).

Méthodologies centrées sur les Agents : Les méthodologies centrées sur les agents sont caractérisées par la prise en compte des concepts du paradigme agent. Les propriétés telles que l'autonomie, la réactivité, la proactivité et la sociabilité sont explicitement considérées

lors des phases de conception du système étudié [LAB06]. Il s'agit pour ces méthodologies d'intégrer, au niveau de la phase d'analyse et de conception, des notions qui ne sont pas explicitement définies dans les approches basées sur les méthodologies orientées objets. Voici quelques exemples de ces méthodologies :

Cassiopée : la méthode Cassiopée est présentée comme « une façon d'appréhender un type de résolution de problèmes qui suppose la mise en œuvre de comportements collectifs par un ensemble d'agents logiciels » [BAR03]. Cette méthodologie est modélisée en trois étapes : [BAR03]

- **Définition des Agents** : le concepteur répartit les connaissances et les compétences, nécessaires au système, entre les agents.
- **Définition des Interactions** : le concepteur définit le mode de communication entre les agents, la nature de leurs échanges.
- **Définition de l'Organisation** : le concepteur définit les liens entre les agents qu'il a conçus, comment travailleront-ils en commun afin d'atteindre l'objectif global que le système dans son ensemble doit atteindre ?

Gaia : Elle s'appuie sur une description au niveau micro (structure de l'agent) et au niveau macro (société d'agents). Les agents sont identifiés à partir des notions de rôles et de responsabilités. Chaque rôle est composé de quatre attributs : responsabilités, permissions, activités et protocoles [BAR03]. Cette méthodologie repose sur la conception des six modèles suivants : [LAB06]

- **Modèle de Rôle (Role Model)**: identifie les différents rôles que joueront les différents agents présents dans le système.
- **Modèle d'Interaction (Interaction Model)**: identifie les protocoles de communication entre les agents.
- **Modèle d'Agent (Agent Model)**: identifie les agents du système en les associant aux différents rôles.
- **Modèle de Service (Service Model)**: identifie les services associés avec chaque rôle et spécifie les propriétés de ces services (exemple : les pré-conditions).
- **Modèle d'Accointance (Acquaintance Model)**: identifie la structure de l'organisation multi-agents en définissant les liens de communication entre les agents.

En résumé, les méthodologies orientées agents s'intéressent à des problèmes de conception de systèmes multi-agents relatifs à : l'identification des agents, la spécification des capacités de raisonnement, l'organisation du système multi-agents et la représentation des interactions entre agents. Elles ont pour objectif de guider le concepteur dans les phases d'analyse et de conception de systèmes multi-agents [LAB06].

III. Modélisation multi agents du système étudié :

Cette section vise à présenter la première étape du développement d'une approche de simulation : la modélisation conceptuelle. Ayant comparé dans la section précédente les différentes méthodes de conception des SMA et leur spécificité, nous introduisons maintenant notre méthode de modélisation selon le paradigme agent basé sur la coopération entre les ressources. Dans un premier temps, nous commencerons par rappeler la nature complexe du problème d'ordonnancement étudié et présenter une description détaillée de la cellule d'assemblage concernée par ce type d'ordonnancement. Nous justifierons, par la suite, de l'approche de modélisation adoptée et enfin nous expliciterons notre modèle selon cette même approche.

1. Rappel du problème étudié :

Dans ce présent travail, nous traitons un problème d'ordonnancement du type Open Shop dans un environnement d'atelier flexible. Le problème étudié est $O4|\tau_j|\bar{F}$. Notre objectif est la réduction du temps total de séjour des pièces dans un système cyclique conçu autour d'un convoyeur en boucle.

Les problèmes Open Shop sont déjà NP-difficiles dans le cas général et l'ajout des ressources de transport, en plus des ressources de transformation (les machines), rend le problème plus difficile. Dans notre cas, la préemption est non autorisée (ressources disjonctives). La typologie circulaire ainsi que le contexte dynamique de l'ordonnancement, rendent le problème « plus combinatoire » par rapport aux lignes d'assemblage linéaires conventionnelles. De plus, les temps de transfert sont considérés (considération des ressources de transport) ce qui fait de notre problème un problème NP-difficile au sens fort.

En effet :

- la cellule d'assemblage étudiée est conçue autour d'un convoyeur en boucle ;
- les pièces nécessaires à l'assemblage d'un produit fini sont contenues sur des palettes, dont le nombre est limité ;
- les pièces sont stockées dans une file d'attente à l'entrée du convoyeur jusqu'à ce que l'une des palettes du convoyeur central se libère et soit réquisitionnée pour transporter cette pièce ;
- les pièces ne sont pas toutes disponibles au même moment ($r_i \neq 0$), les différents types de pièces arrivent progressivement à des intervalles réguliers.

Dans ce contexte, il serait difficile de mettre en œuvre un ordonnancement prévisionnel selon l'une des approches optimales présentées au chapitre I. Nous opterons donc pour un ordonnancement temps réel où le routage des opérations d'un travail est décidé en temps réel. Pour plus de flexibilité et de réactivité face aux aléas, nous dotons notre système d'une architecture distribuée inspirée des SMA.

Ce système permettra de prendre en compte simultanément l'état des différentes ressources pour déterminer un meilleur routage de la pièce à travers le système afin de compléter son assemblage. Au fur et à mesure, la prochaine destination de la palette n'est plus la station la plus proche disposant de place dans sa file d'attente d'entrée mais celle dont la file d'attente est le **moins saturée** et la **plus proche** sur le convoyeur. Celle-ci sera déterminée à l'issue du **processus de coopération** entre les différentes ressources ou centres de décisions.

L'ordonnancement des opérations au niveau des files d'attente d'entrée est effectué par application de la règle de priorité FIFO.

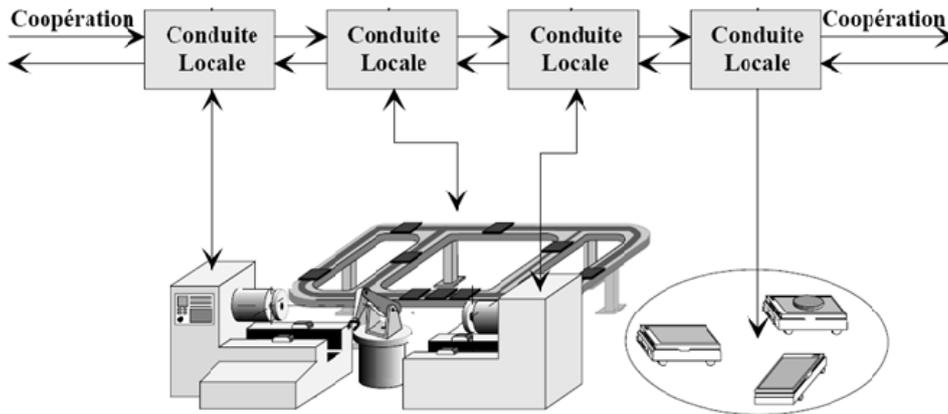


Figure II.8 : Architecture du système d'ordonnancement basé sur la coopération

2. Description du système cyclique d'assemblage :

La cellule d'assemblage cyclique étudiée est détaillée dans la figure II.9, elle est constituée de:

- Un convoyeur en boucle dont la dimension est choisie de façon à ce qu'elle soit un multiple de la dimension des palettes (soit l la longueur de la palette et L la longueur du convoyeur).
- Un nombre fini de palettes (les temps de chargement ou de déchargement sont négligés) (Soit N_p le nombre de palettes dans le système).
- Les points d'entrée et de sortie du système :
 - entrée du système ;
 - sortie du système ;
 - les M entrées des M stations ;
 - les M sorties des M stations.
- Les stations sont disposées en boucles attachées au convoyeur, chacune d'elles est définie par:
 - Le type de robot qui exécute l'opération élémentaire et son temps d'exécution;
 - Le temps opératoire total (selon les gammes opératoires des produits);
 - Positionnement des entrées et sorties des stations : entrée de la station vers la file d'attente, entrée du robot à partir de la file d'attente d'entrée, sortie du robot vers la file d'attente de sortie, sortie de la station à partir de la file d'attente de sortie.
 - La capacité NFAE Files d'Attente d'Entrée (FAEs) devant chacun des robots.
 - La capacité NFAS Files d'Attente de Sortie (FASs) de chacun des robots.
 - Points de branchement avec le convoyeur : à l'entrée de chacune des stations (test d'entrée) et à la sortie de chacune des stations (priorité au convoyeur).

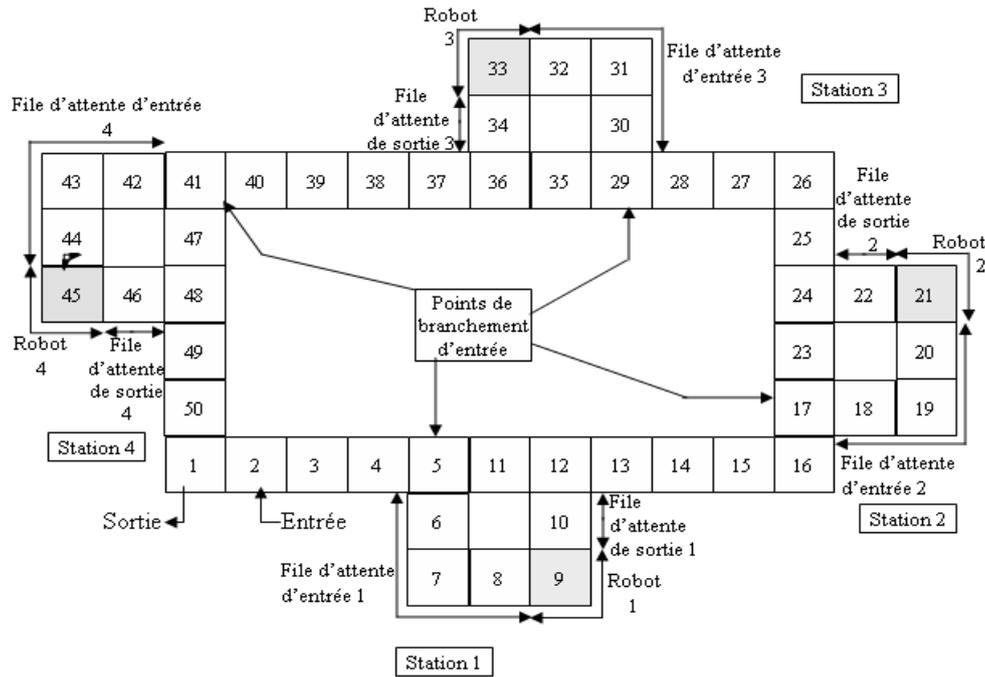


Figure II.9 : Cellule flexible d'assemblage étudiée

3. Approche de modélisation multi agents :

Les systèmes multi-agents ont été développés dans le cadre de l'intelligence artificielle distribuée. L'intérêt qu'ils suscitent est lié à leur capacité d'aborder les problèmes complexes d'une manière distribuée et de proposer des solutions réactives et robustes. La résolution distribuée par l'approche multi-agents s'adapte bien avec la nature des systèmes de production généralement composés de plusieurs équipements, où il faut intégrer plusieurs fonctions (conduite, gestion de production etc.) et où les décisions doivent se prendre à plusieurs niveaux. L'architecture distribuée d'un système multi-agents permet également aux systèmes de production, qui sont modélisés à l'aide de cette approche, d'acquérir la propriété de proactivité [DRA98].

L'approche multi-agents fait partie des méthodes de conception des systèmes d'ordonnancement répartis. Dans cette méthode, chaque agent, qui représente une ressource, dispose d'une capacité de perception de son environnement et d'une capacité de contrôle locale, il décide lui-même des actions qu'il doit réaliser en fonction de son objectif local. La performance du système n'est pas globalement planifiée, elle émerge de la dynamique des interactions en temps réel entre les agents. Ainsi, le système n'est pas obligé de toujours alterner entre une phase de planification et une phase d'exécution, mais son comportement est élaboré à partir des décisions concourantes des agents locaux. Dès lors, la coopération entre les agents est indispensable pour rechercher des solutions dans un espace global à partir d'informations locales.

En résumé, nous avons choisi l'approche multi agents pour modéliser la cellule d'assemblage étudiée car celle-ci est la plus appropriée pour modéliser les systèmes de production complexes, elle ajoute à la localité des comportements, présente dans l'approche

objet, l'autonomie et la répartition des prises de décisions présentée par les systèmes flexibles de production.

Il existe dans la littérature plusieurs approches de conception d'un système multi agents, nous avons pu en avoir un aperçu au chapitre 2. L'approche retenue pour modéliser la cellule d'assemblage étudiée s'inspire des méthodes précédentes pour mieux tenir compte des spécificités du problème étudié. Elle a déjà été appliquée à un problème similaire de réordonnancement dans [TRA01].

Cette approche est essentiellement basée sur la distinction entre les comportements indépendants d'un agent, de ceux résultant d'interaction entre agents. Cette distinction se traduit par la décomposition du modèle d'un agent en **modèle individuel** et **modèle social**.

- Le modèle individuel contient les caractéristiques indépendantes de tout contexte social de l'agent. Il établit les comportements individuels de l'agent ainsi que ses compétences, ses critères de décision et ses objectifs individuels.
- Le modèle social établit les comportements sociaux de l'agent et définit le rôle de chacun des agents durant leurs interactions au travers d'échanges de messages formant ainsi une conversation suivant un protocole de communication.

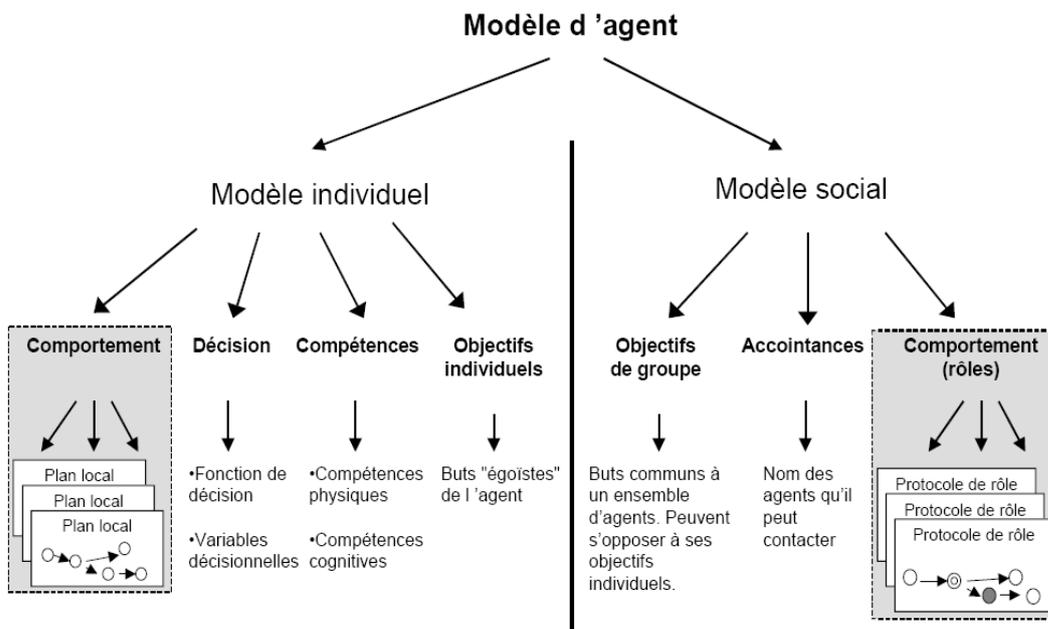


Figure II.10 : modèle d'agent pour un système d'ordonnancement coopératif [TRA01]

4. Protocole de coopération inter ressources (modèle social):

Dans notre approche, la coopération est assimilée à une action de prise de décision collective distribuée en vue de synchroniser les actions réparties sur les différentes ressources. La mise en place d'un système d'ordonnancement coopératif au niveau de la cellule d'assemblage étudiée nécessite donc la définition d'un protocole de communication.

Le protocole de communication peut être défini comme l'ensemble des actes de langage utilisés pour le déroulement du processus de coopération ainsi que leur séquençement. Cette

définition souligne deux aspects importants à savoir : les messages que les différentes entités peuvent s'envoyer et la dynamique opérationnelle associée. Chaque forme de coopération possède son propre protocole.

En effet, la communication est nécessaire à toute interaction entre agents. L'envoi, la réception et l'interprétation des messages nécessitent le recours à des langages de communication. Ces langages vont permettre de supporter la communication entre les agents. Dans le paragraphe suivant, nous proposons les différentes primitives de coopération retenues dans l'élaboration de notre protocole de coopération. Dans ce qui suit, le centre de décision initiateur est celui qui entame la conversation avec un autre centre de décision qui sera appelé participant.

A. Primitives de coopération :

Les différentes primitives que nous proposons sont issues du langage de communication ACL (*Agent Communication Language*) développé par *Foundation for Intelligent Physical Agents* (FIPA).

- Primitives de l'initiateur : L'initiateur possède trois primitives de coopération :

Query : requête ou acte d'interroger un autre agent. L'envoi de cette requête invite l'agent récepteur à exécuter un acte de type « informer ».

Request : L'initiateur demande au destinataire de s'engager à exécuter certaines actions.

Confirm : Ce message indique aux participants que l'engagement en cours est confirmé.

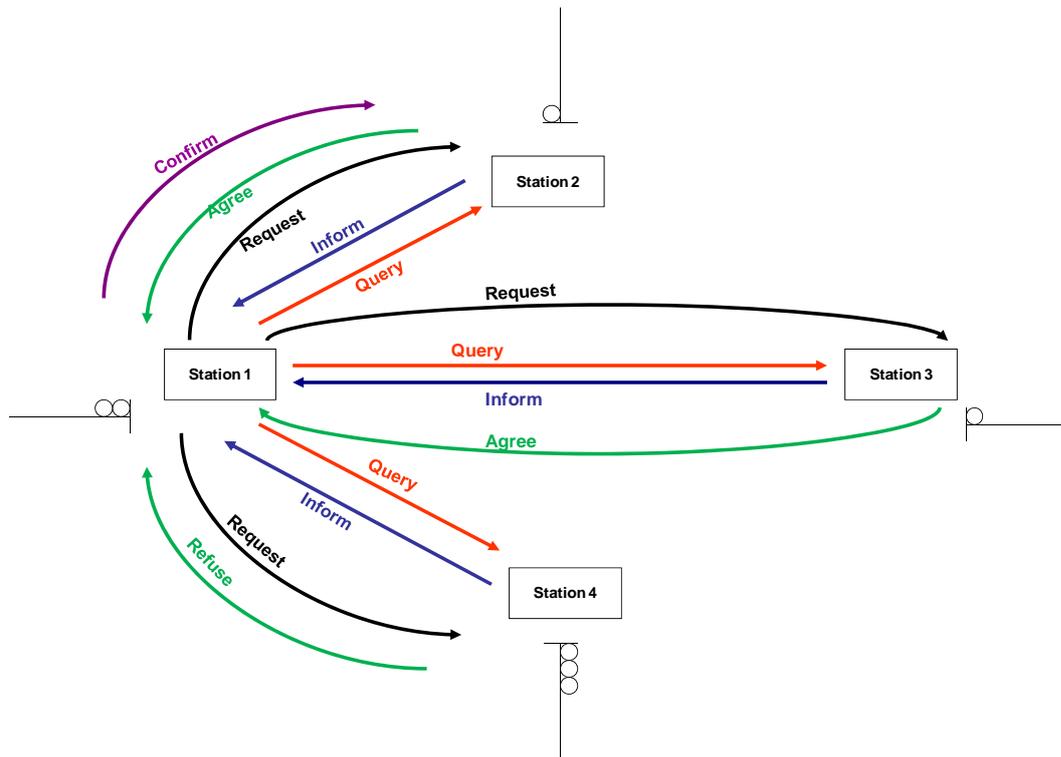
- Primitives du participant : Les messages envoyés par le participant sont destinés à l'initiateur ou à l'ensemble des participants.

Inform : L'émetteur informe le récepteur suite à la requête initiée par l'initiateur (Query).

Agree : Ce message répond à la demande (Request) faite par l'initiateur. Le participant indique par ce message à l'initiateur qu'il s'engage à exécuter certaines actions.

Refuse : L'action de refuser d'exécuter une action donnée.

Cancel : L'action d'un agent d'informer un autre agent qu'il n'a plus l'intention d'exécuter l'action pour laquelle il s'est engagé.



II.11 : Philosophie générale du protocole de coopération

Figure

L'arrivée d'une pièce au niveau d'une *station i* déclenche le protocole de coopération par l'envoi d'une demande d'information (**Query**) à l'ensemble des autres stations non encore visitées par la pièce. Cette requête porte sur l'état des files d'attente des stations interrogées. A partir des renseignements obtenus, s'il s'avère que la station initiatrice n'est pas la plus **adaptée** pour le traitement de la pièce selon un critère préalablement établi, celle-ci envoie des demandes d'engagements (**Request**) à l'ensemble des stations non visitées. La station initiatrice reçoit alors au moins un message d'acceptation (**Agree**).

Néanmoins, si plusieurs stations acceptent de s'engager, la station initiatrice déclenche alors une procédure de recherche de la station la plus adaptée pour l'exécution de la tâche c'est-à-dire la station ayant la file d'attente la moins saturée et la plus proche sur le convoyeur de l'emplacement actuel de la pièce. La station sélectionnée reçoit alors un message de confirmation (**Confirm**) de la part de la station initiatrice et réactualise son échéancier d'exécution. En cas de désengagement de la station sélectionnée, un message d'annulation (**Cancel**) est envoyé par celle-ci à l'ensemble des participants.

B. Les étapes du protocole de coopération :

Dans ce qui suit, nous décrivons plus en détail les différentes étapes du processus de coopération mis en œuvre dans le cadre du système d'ordonnancement de la cellule d'assemblage. Ce protocole permettra à la fois de spécifier la répartition des capacités décisionnelles entre les différentes ressources et de fixer les règles d'interaction afin de réaliser une coopération entre elles.

Etape 1 : Envoi de demandes d'information

A l'arrivée d'une pièce à l'entrée d'une *station i*, celle-ci envoie une demande d'information (**Query**) à l'ensemble des stations non encore visitées par la pièce. Cette demande d'information porte sur l'état des files d'attente de ces stations.

Lorsqu'une *station j* reçoit une demande d'information, elle exécute aussitôt une **procédure de calcul** pour évaluer le nombre de pièces dans sa file d'attente d'entrée $NFAE_j$ à l'instant $TNOW + T_{ij}$, tel que T_{ij} représente le temps de transfert de la *station i* vers la *station j*. Ce calcul est donc effectué afin de déterminer l'état de la FAE de la *station j* au moment de l'arrivée de la pièce dans l'éventualité qu'elle serait plus adaptée que la *station i* pour exécuter cette tâche.

La **procédure de calcul** est décrite ci-dessous :

Pas 0 : Soit $NQ = NFAE_j(TNOW)$, aller à **Pas 1**

Pas 1 : Calcul du nombre d'engagement NG de la *station j* pour la période $[TNOW, TNOW + T_{ij}]$, aller à **Pas 2**

Pas 2 : Calcul du nombre de pièces traitées NT durant l'intervalle $[TNOW, TNOW + T_{ij}]$, aller à **pas 3**

Pas 3 : Calcul du nombre de pièces dans la *file d'attente j* à $TNOW + T_{ij}$:
 $NFAE_j(TNOW + T_{ij}) = NQ + NG - NT$

Pour plus de détails concernant la procédure de calcul, veuillez vous référer à l'annexe 5.

Etape 2 : Réception des messages d'information

Dès que les données sont disponibles au niveau de la *station i*, elle teste alors la condition suivante :

$$NFAE_i(TNOW) = \text{Min} \{ \{NFAE_j(TNOW + T_{ij})\}_{j \in I}, NFAE_i(TNOW) \} \dots \dots (I)$$

tel que I est l'ensemble des stations non visitées.

- Si cette condition est vérifiée alors la *station i* se charge de traiter la pièce et l'envoie à la fin du traitement vers la station la plus proche non encore visitée.
- Sinon : Elle envoie une requête de type demande d'engagement aux autres stations participantes (**Request**).

En résumé, l'objectif de cette deuxième étape est de déterminer si la *station i* est la plus adaptée pour traiter la pièce c'est-à-dire la station ayant la file d'attente la moins saturée et donc celle qui engendrera un temps d'attente minimal pour la pièce.

Remarque : La relation (I) représente le critère retenu pour déterminer la station la plus adaptée pour le traitement de la pièce en cours. Ce critère est un critère générique auquel on additionnera d'autres conditions au chapitre IV afin de tester différentes stratégies de pilotage possibles.

Etape 3 : Collaboration des stations pour minimiser le temps d'attente

Les *stations j*, capables de prendre en charge la pièce, envoient un message d'acceptation à la *station i* (**Agree**).

- Si la *station i* reçoit plusieurs messages d'acceptation, elle fait appel à une procédure de recherche de la station la plus adaptée c'est-à-dire la moins saturée et la plus proche sur le convoyeur de l'emplacement actuel de la pièce (la pièce est toujours à l'entrée de la *station i*). La pièce est alors transférée vers la station sélectionnée.
- Sinon la pièce est envoyée vers l'unique *station j* ayant accepté la requête.

La **procédure de recherche** est décrite ci-dessous :

Pas 0 : Soit $F = \{station\ j \setminus tel\ que\ réponse\ j = Agree\}$

Pas 1 : Choisir la *station j* la plus proche de *i* sur le convoyeur (T_{ij} minimal)

Si : $NFAE_j(TNOW + T_{ij}) = Min \{ \{NFAE_j(TNOW + T_{ij})\} \mid j \in F \}$ **Alors** aller à **Pas 2**

Sinon : $F = F - \{Station\ j\}$, aller à **Pas 1**

Pas 2 : Engagement de la *station j* et envoie du message de confirmation (**Confirm**) à la *station j*

En résumé, l'objectif de cette étape est d'assurer la meilleure affectation possible des pièces du point de vue du temps d'attente ; en choisissant la station dont la file d'attente d'entrée est le moins saturée, et du temps de transfert ; en sélectionnant la station la plus proche sur le convoyeur qui satisfait au critère précédent.

5. Architecture du centre de décision ordonnanceur (modèle individuel) :

L'architecture d'un agent ordonnanceur se décompose en quatre modules: Communication, Connaissances, Décision et Expertise. La figure II.12 illustre comment ces quatre Modules interagissent pour dynamiser le comportement de cet agent ordonnanceur.

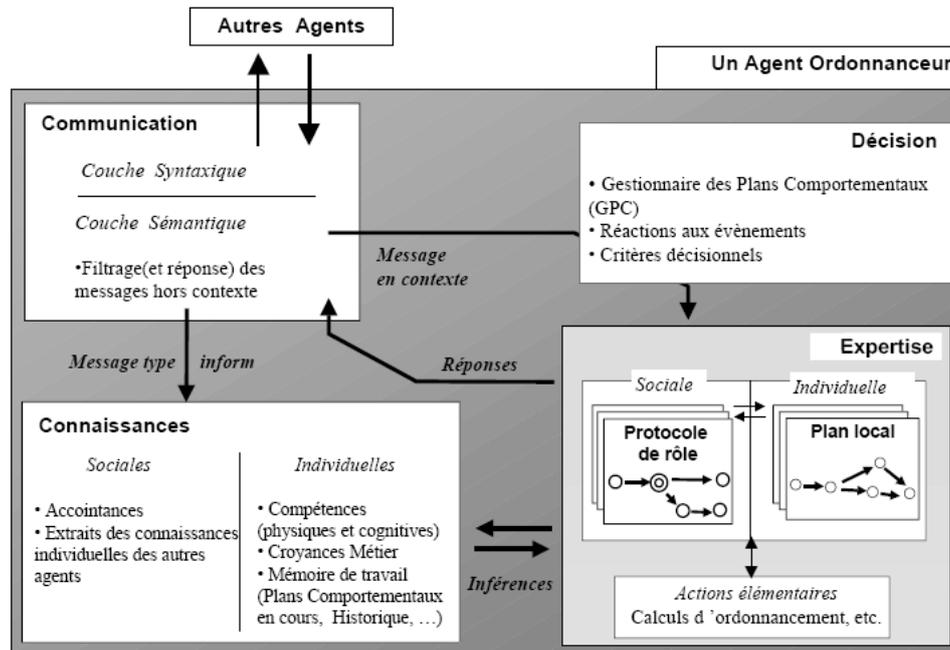


Figure II.12 : architecture de l'agent ordonnanceur [TRA01]

Cette architecture s'inspire à la fois d'une approche AOP (Croyances, Compétences, Choix, Engagements) et d'une approche BDI (Croyance, Décision, Intention) [TRA01]. La notion de Plan Comportemental (PC) est utilisée pour expliquer le comportement des agents et aider à la conception du système multi agents.

Module Communication :

Le module « Communication » gère l'envoi et la réception des messages à destination ou en provenance des autres acteurs. Cette gestion concerne le médium linguistique de la transmission des messages, c'est-à-dire l'expression et l'interprétation d'un message.

Pour qu'un message soit compréhensible par un centre de décision, il faut que ce message soit exprimé correctement dans un langage connu du centre de décision et qu'il porte sur des informations interprétables par cet acteur. En d'autres termes, le message doit être syntaxiquement et sémantiquement correct. Pour s'assurer que ces deux contraintes soient respectées, le module communication est organisé en deux couches: l'une dite syntaxique et l'autre sémantique. La couche syntaxique vérifie simplement que le message respecte les règles grammaticales du langage ACL. La couche sémantique s'assure de la bonne compréhension du message, en vérifiant que le contexte du message est connu. Le contexte est formé du nom du protocole, de l'identifiant de la conversation. Lorsqu'un message est identifié comme étant hors contexte, un message d'erreur est renvoyé indiquant si le protocole ou la conversation sont inconnus, ou encore si la conversation spécifiée est déjà terminée.

Module Connaissance

Le module « Connaissance » contient l'ensemble des informations détenues par un centre de décision sur lui-même (connaissances individuelles), ou sur les autres (connaissances sociales). Les connaissances individuelles d'un centre de décision ordonnanceur, forgent son identité, c'est-à-dire qu'elles identifient ce qu'il sait faire (compétences physiques et

cognitives) et ce qu'il fait ou va faire (croyances métier et mémoire de travail). Les connaissances sociales caractérisent son implication dans le groupe. Elles portent tout d'abord sur les accointances, c'est -à-dire l'ensemble des agents dont il a connaissance de l'existence. Cette connaissance peut être complétée par des extraits des connaissances individuelles de certains agents. Lorsqu'un agent recherche une coopération avec d'autres agents il se réfère à ses connaissances sociales pour ne s'adresser qu'à un ensemble restreint et pertinent d'agents.

Module Expertise

Alors que les compétences dans le module « Connaissance » décrivent ce que sait faire le centre de décision, le module « Expertise » détaille le comment de ces compétences. En effet, ce module contient l'ensemble des Plans Comportementaux (PC) ; à savoir les plans locaux, et l'ensemble des actions élémentaires exécutables par un centre de décision. Retenons qu'un plan spécifie une séquence d'actions élémentaires tels que l'envoi de message, les calculs divers (dont l'ordonnancement), la mise à jour des connaissances,...etc.

Module Décision

Le module « Décision » contrôle l'exécution des actions entreprises au niveau du module « Expertise ». Ce contrôle peut être réfléchi ; c'est le rôle du gestionnaire des Plans Comportementaux, ou réflexe lorsque des événements spécifiques sont détectés. Le gestionnaire des Plans Comportementaux active ou désactive les Plans Comportementaux après avoir vérifié leurs conditions de déclenchement. Il maintient également à jour la mémoire de travail du module « connaissance ».

Chaque station dispose de deux plans comportementaux qui sont représentés dans les schémas suivants :

Cas 1 : Arrivée d'une pièce ne disposant pas d'engagement (EG=0)

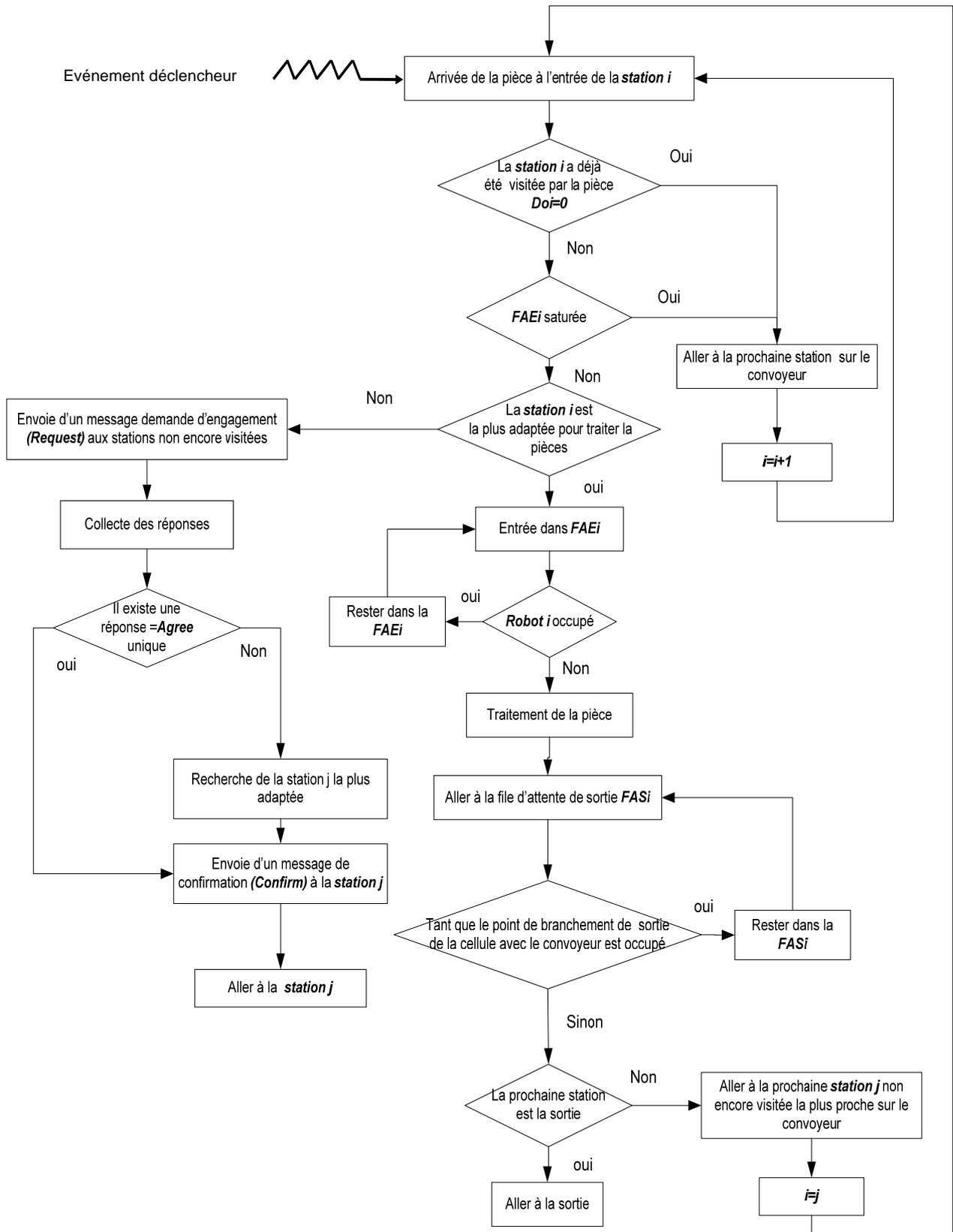


Figure II.13: **Cas 1** : Arrivée d'une pièce ne disposant pas d'engagement (EG=0)

Cas 2 : Arrivée d'une pièce disposant d'un engagement ($EG \neq 0$)

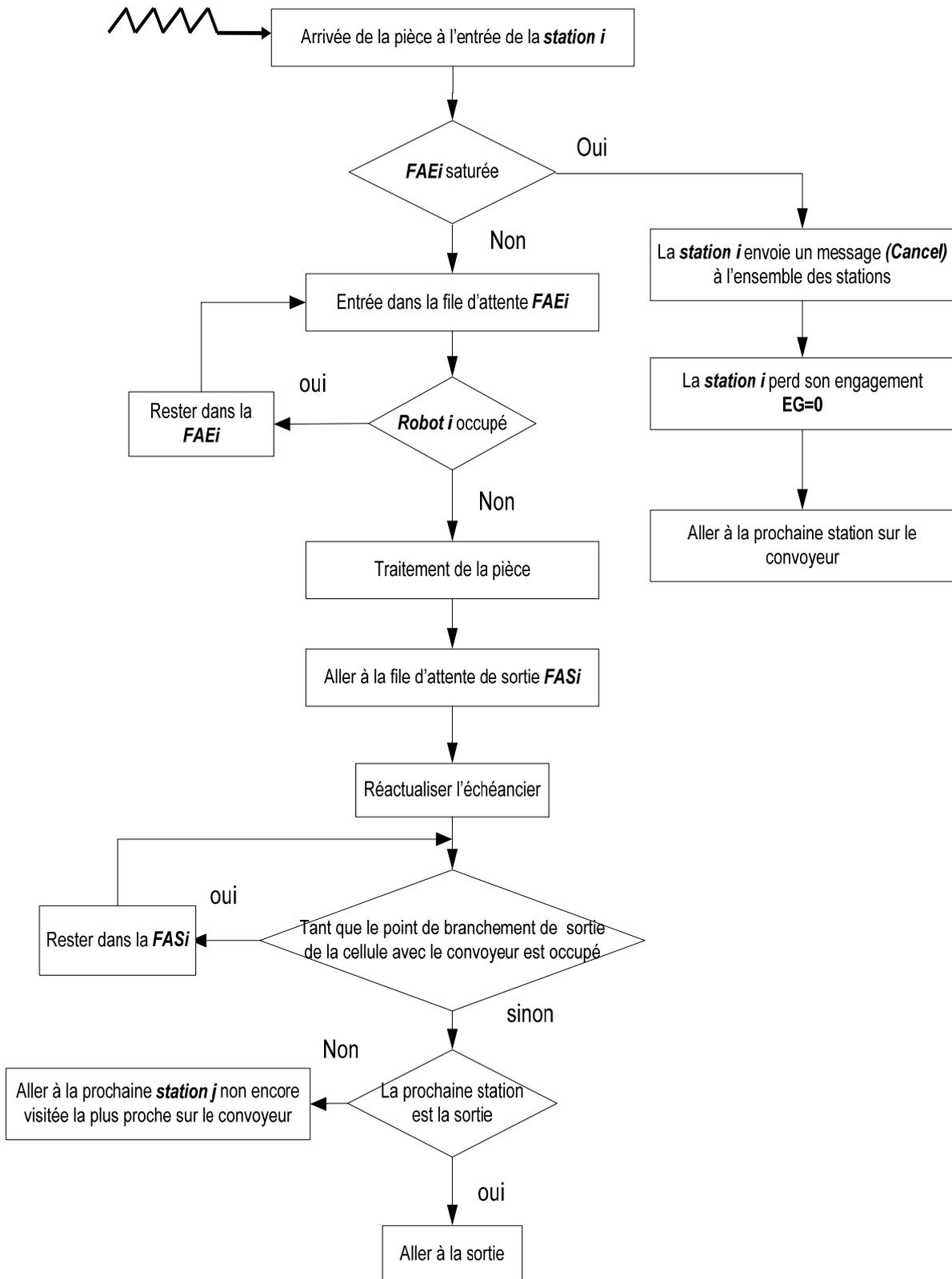


Figure II.14 : **Cas 2** : Arrivée d'une pièce disposant d'un engagement ($EG \neq 0$)

IV. Conclusion :

Aujourd'hui, la plupart des applications temps réel nécessitent de distribuer des tâches entre des " entités " autonomes (ou semi-autonomes) afin d'atteindre leurs objectifs d'une manière optimale.

L'application des systèmes **multi-agents** dans un environnement **temps réel** ouvre de nombreuses perspectives quant à l'élaboration et le développement de systèmes, réputés complexes et restreints, que sont les systèmes temps réel. De tels systèmes trouvent leur utilité dans de nombreux secteurs de l'industrie comme : l'automobile, l'aéronautique, les systèmes dits de contrôle-commande, la robotique, etc. Il est donc tout naturel qu'on ait cherché à utiliser les agents pour l'élaboration notre système d'ordonnancement.

Dans ce chapitre, nous avons proposé une approche par coopération inter machines pour la résolution du problème Open Shop. Nous avons donc établi un protocole de coopération permettant de résoudre le problème d'une manière distribuée. La modélisation du système selon l'approche SMA a permis de définir les comportements individuels des agents à savoir les ressources, de sorte que ces comportements soient globalement cohérents avec notre objectif « *minimisation du C_{max}* ».

Chapitre III : Modélisation et simulation de la cellule d'assemblage

- I. Introduction

- II. Simulation, définitions et concepts
 - 7. Définitions
 - 8. Fondement de la simulation
 - C. Modélisation et processus de modélisation
 - D. Types de modèles pour la simulation
 - 9. Méthodes analytiques vs simulation
 - 10. Apport de la simulation dans les systèmes de production
 - 11. Avantage et limite de la simulation
 - 12. Les étapes du processus de simulation

- III. Présentation du logiciel ARENA 7.01
 - 5. Structure d'un programme ARENA
 - 6. Structure hiérarchique d'ARENA
 - 7. Transfert des entités dans ARENA
 - 8. Utilisation de VBA dans ARENA

- IV. Mise en œuvre du processus de simulation
 - 1. Formulation du problème
 - 2. Identification et collecte des données
 - 3. Modélisation par l'approche processus
 - 4. Construction du modèle ARENA et transcription informatique du modèle
 - 5. Vérification et validation du modèle

- V. Conclusion

I. Introduction :

Etant donné la complexité du système étudié, tel que défini au chapitre 2, il n'est pas imaginable de l'analyser de manière analytique en raison de la complexité des processus. Reste l'approche simulateur qui permet l'analyse de scénarios alternatifs en terme de structures, ressources et de procédures afin d'en évaluer la pertinence et l'intérêt.

La simulation informatique révèle toute son utilité pour évaluer les performances prévisionnelles des systèmes complexes. Aujourd'hui, elle est devenue indispensable pour aider à prendre en charge les problèmes d'optimisation des flux physiques ou des flux d'informations dans les systèmes de production manufacturiers.

Pour toutes ces raisons, nous avons retenu, dans le cadre de ce travail, la simulation comme support principal de mise en œuvre de l'approche empirique adoptée pour la résolution du problème posé.

La première étape du développement de l'approche de simulation pour l'évaluation des performances et l'analyse de la cellule flexible d'assemblage, la modélisation conceptuelle, a été présentée au chapitre précédent. Le modèle conceptuel, présenté selon une approche agent, doit maintenant être implémenté afin de créer le modèle de simulation à proprement dit. Le présent chapitre vise donc à présenter l'implémentation du modèle conceptuel grâce au logiciel de simulation général ARENA.

Dans le souci de garder un enchaînement logique à ce chapitre, et avant d'aborder le cœur du sujet (élaboration du modèle de simulation), nous nous intéresserons d'abord à la simulation des systèmes de production afin d'y introduire tous les concepts nécessaires à l'aboutissement de ce travail.

Nous abordons, dans la deuxième section, la définition et les fondements de la simulation, son application au domaine de la production, ses objectifs avantages et limites, ses approches de modélisation ainsi que les étapes du processus de simulation. La troisième section est consacrée à la présentation du logiciel Arena. Enfin, la quatrième section de ce chapitre est consacrée à la description de la mise en œuvre des cinq premières étapes du processus de simulation de la cellule d'assemblage.

II. Simulation, définitions et concepts :

La simulation est un puissant outil d'aide à la décision pour l'industriel. Faute de pouvoir mettre l'usine en équation (les contraintes sont multiples et les acteurs-décideurs sont de plus en plus nombreux), la simulation permet de reproduire son fonctionnement sur ordinateur et comparer des scénarios d'exploitation. On peut de cette manière, tester l'impact d'un investissement, d'une modification de paramètre, du lancement d'un nouveau produit sur une ligne de production, d'une autre règle d'ordonnancement sur la performance du système étudié. Cette technique peut aider à mieux cerner les conséquences des choix potentiels d'actions, afin de mieux les maîtriser et d'améliorer ainsi le pilotage de l'atelier.

1. Définitions :

Une synthèse de différentes définitions de la simulation est proposée dans [BAK96] : « *la simulation est une méthode de mesure et d'étude consistant à remplacer un phénomène ou un système à étudier par un modèle informatique plus simple mais ayant un comportement analogue* ».

2. Fondement de la simulation :

Simuler un système revient à imiter son comportement afin de mesurer sa réponse (output) à différents intrants (inputs) en fonction du temps. La simulation requiert donc une première étape de **modélisation** du système réel. Ce modèle abstrait est ensuite traduit ou interprété afin de générer un code informatique respectant la logique du modèle. Le logiciel résultant est soumis aux différentes expériences ou scénarios préparés par le concepteur et permet, après une phase d'analyse des résultats, d'inférer des connaissances quant au comportement du système et d'évaluer les opportunités de pilotage les plus profitables.

La modélisation constitue donc le fondement même de la simulation. La réciproque n'est d'ailleurs pas forcément vraie. Effectivement, la modélisation n'a pas forcément pour finalité la simulation [BER00]. Il existe des techniques de modélisation qui ont pour objectif l'amélioration de la compréhension du système étudié, l'analyse de son organisation, sans pour autant effectuer de simulation. Mais la modélisation est la première phase essentielle dans la simulation, c'est pourquoi, nous proposons le rappel suivant sur la notion de modèle.

A. Modélisation et processus de modélisation :

« *La modélisation est un processus de représentation qui permet d'obtenir une image approchée du système réel, suite à une phase d'abstraction, exprimée dans un langage de représentation* » [BER00]. Ce langage peut être formel (c'est-à-dire ayant une syntaxe et une sémantique bien définies comme un langage informatique ou mathématique), semi-formel (notation graphique normalisée) ou informel (description en langage naturel).

Les modèles sont largement utilisés pour l'étude des systèmes industriels car ils sont moins coûteux que l'utilisation du système réel et rendent possible la conduite d'expériences multiples. En effet, Dans certains cas, il peut s'avérer trop difficile, trop cher, voire impossible de manipuler directement le système réel et cela pour l'une des raisons présentées ci-dessous :

- **Le coût économique** : dans le cas d'un processus de production, des essais industriels impliqueraient l'arrêt de la production tout au long de la phase d'expérimentation ;
- **Risque écologique** : les conséquences d'une expérimentation peuvent, dans certains cas, devenir indésirables ou intolérables ;
- **Impossibilité d'opérer des expérimentations** : Le système n'existe pas (phase de conception) ou impossibilité de réaliser des expérimentations sur le système (sismologie, géologie,..).

Le processus de modélisation se compose de 04 phases. La première est la phase de spécification, et les trois suivantes forment la phase d'exploitation.

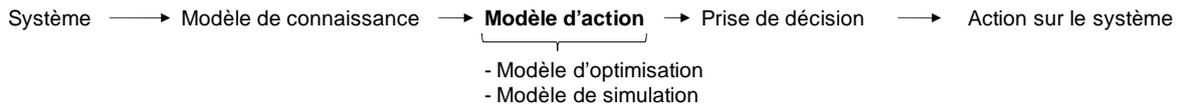


Figure III.1 : Processus de modélisation [CAU06]

Dans ce processus, la connaissance que l'on a du système est rassemblée dans un modèle appelé « modèle de connaissance ». Ce modèle contient la description des entités et de leurs interactions, mais ne contient aucun élément de la solution mise en œuvre. Tant que le système n'est pas modifié le modèle de connaissance doit rester valide. L'obtention de ce premier modèle marque la fin de la phase de spécification.

La phase d'exploitation du modèle peut alors commencer avec l'élaboration d'un ou plusieurs modèles d'action. Le modèle d'action est le modèle mis en œuvre pour avoir une action sur le système. Suivant le type de modèle d'action, on parle de « modèle d'optimisation » si le modèle fait appel aux techniques d'optimisation (heuristiques, programmation linéaire,...), de « modèle de simulation » s'il fait appel aux techniques de simulation (simulation à événements discrets, simulation continue...).

Le modèle d'action produit des résultats dont l'exploitation aide les experts à prendre une décision. Une décision étant prise, une action peut être initiée sur le système.

B. Types de modèles pour la simulation :

On distingue plusieurs types de modèles de simulation en ce qui concerne les systèmes de production : [BER00]

Les modèles discrets (ou discontinus), déroulent l'historique des événements qui surviennent lors d'une fabrication. Les changements d'état ne surviennent que lors d'événements tels le début ou la fin d'une opération, la mise en attente d'une pièce dans une file, la libération d'une ressource, l'occurrence d'une panne, ...etc. Quatre approches de modélisation des systèmes de production discrets pour la simulation ont été identifiées :

- Approche orientée événements (event scheduling) ;
- Approche orientée activités (activity scanning) ;
- Approche orientée processus (process interaction)
- Approche orientée objets.

Pour plus de détails concernant ces approches se référer à l'annexe 3.

Les modèles continus, plus adaptés aux industries dites de process, utilisent des équations mathématiques pour prendre en compte les changements d'état qui s'effectuent de façon continue au cours du temps. Les valeurs des variables d'état sont recalculées régulièrement selon un pas d'horloge d'après ces équations.

Les modèles combinés sont capables d'intégrer les deux aspects.

3. Méthodes analytiques vs simulation

Contrairement aux méthodes analytiques qui recherchent des solutions optimales, la simulation n'est pas une méthode de résolution de problème. Elle permet de répondre à la question : « *Qu'obtiendra-t-on si l'on fait ceci ?* » mais ne permet pas de répondre à la question : « *Que faut-il faire pour obtenir cela ?* ».

Pour aboutir à une solution intéressante, il est nécessaire de tester un nombre suffisant de scénarios afin de les comparer et de retenir le plus intéressant. Ainsi, la simulation est une démarche par induction (étude de cas particuliers afin d'aboutir à une conclusion, la plus générale possible) et non par déduction (solution obtenue par un raisonnement, un algorithme).

Cependant, de nombreux auteurs notamment [HAB01], insistent sur l'absence d'opposition entre les méthodes analytiques et la simulation. Selon eux, ces approches ne sont pas concurrentes mais le plus souvent complémentaires, pouvant intervenir, du fait de leurs caractéristiques, à des étapes différentes du cycle de vie d'un SdP.

Ainsi, les méthodes analytiques semblent bien adaptées aux premières phases de la conception d'un système. En effet, à ce niveau, les données sont peu précises et un grand nombre de configurations doivent être testées. De ce fait, il est plus intéressant d'utiliser les méthodes analytiques pour déterminer une configuration initiale qui servira de base de départ à des configurations beaucoup plus précises à l'aide de la simulation. Ceci s'explique bien par les spécificités de chacune des deux techniques : [HAB01]

Les **méthodes analytiques ou d'optimisation** se distinguent par les points suivants : elles imposent en général des hypothèses simplificatrices sur le modèle, elles présentent le risque d'explosion combinatoire, elles ne prennent en compte que les phénomènes les plus importants (d'éventuelles perturbations : blocage, pannes, etc., sont négligées), elles ne fournissent que des performances moyennes sur une longue période de temps, les résultats sont exacts pour des modèles simples et de petites tailles, les résultats sont approximatifs pour des modèles complexes.

La **simulation**, quant à elle, se caractérise par les points suivants : elle introduit des changements dans le temps (technique temporelle), elle possède un processus de conception du modèle logique et d'expérimentation, elle est utilisée pour la modélisation de systèmes complexes (généralement stochastiques), le modèle peut être détaillé sans contraintes jusqu'au niveau désiré, elle nécessite beaucoup de données, la durée de traitement d'une simulation est souvent plus longue.

4. Apport de la simulation dans les systèmes de production :

L'analyse par simulation poursuit l'objectif de toute technique d'aide à la décision, à savoir d'évaluer les décisions possibles dans un certain contexte et de choisir la meilleure stratégie à adopter par le décideur.

Un simulateur par événements discrets peut conduire à diverses utilisations : [AZZ05]

Outil prédictif d'aide à la production : un SED peut être utilisé pour fixer des délais de production dans un contexte donné. On peut en général visualiser le déroulement d'une campagne et détecter les mauvais fonctionnements, les lacunes de l'organisation de production (manque de matières premières, retards excessifs...), ou les goulets d'étranglement du système. Il peut également servir à la détermination de la capacité maximale de production d'un atelier donné ;

Réaction aux aléas de production : l'incidence d'un aléa sur la campagne, et sur les délais de production en particulier, peut néanmoins être évaluée et différentes stratégies testées pour en limiter l'impact ;

Aide à la conception : elle permet d'évaluer les performances d'un système de configuration donnée aussi bien au niveau de la structure physique (conception de nouveaux systèmes), qu'au niveau du système de décision (exploitation, réorganisation), afin de déterminer la plus favorable, au regard d'un critère de performance donné ;

Optimisation interactive : sous cette appellation un peu abusive, on désigne l'aptitude d'un simulateur à contribuer à la mise au point de paramètres de fonctionnement d'un système. L'utilisation du simulateur permet d'évaluer l'impact de modifications des paramètres de fonctionnement sur les indicateurs de performance de production du système. On peut par exemple tester l'influence de l'ajout d'un équipement supplémentaire à un endroit donné de l'atelier, ou l'augmentation des capacités des stockages...etc.

Fonction d'évaluation des performances : on peut signaler l'aptitude du modèle de simulation à être une fonction d'évaluation d'une performance de production choisie du système étudié. Le modèle de simulation peut en effet être utilisé pour évaluer l'efficacité d'une configuration de production choisie, avec un critère (ou « fonction d'évaluation ») de nature technique.

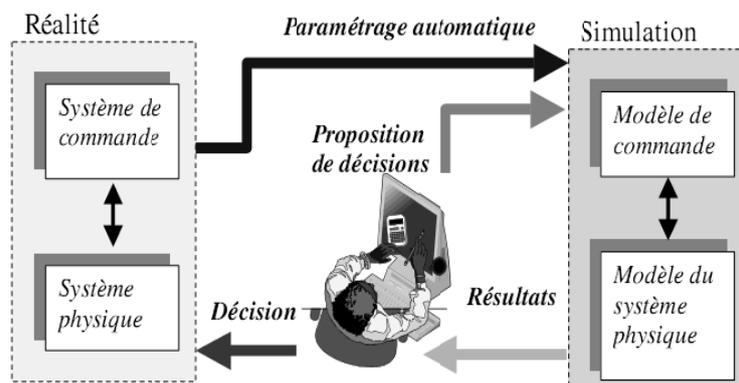


Figure III.2: simulation des systèmes de production [GHA06]

5. Avantage et limite de la simulation :

Avantages : Cette grande popularité de la simulation, s'explique par les nombreux avantages offerts par cette technologie. On peut citer, entre autres, les suivants : [AIT05]

- La simulation permet d'intégrer un grand nombre de paramètres et permet aussi de tester aisément des hypothèses sur le fonctionnement du système et de mieux le comprendre.
- Elle permet une étude plus réaliste des systèmes (événement aléatoires, interaction complexe entre entités, distribution de probabilités quelconques...)
- Par sa nature expérimentale et grâce aux possibilités de visualisation graphique (animation des pièces, évolution temps réel de courbes de performance), la simulation présente des avantages déterminants en terme de compréhension et de communication du modèle, de vérification et de validation du modèle, de présentation et de vérification des résultats, de promotion d'approches globales des problèmes, de stimulation d'attitudes créatives chez les managers ainsi que d'utilisation pour la formation du personnel.

Limites : Certaines limites sont dues à la technique elle-même, d'autres sont dues aux produits actuellement disponibles sur le marché (notons qu'une formation sur les logiciels utilisés est souvent nécessaire) : [AIT05]

- La programmation demande un certain niveau d'expertise. La qualité des résultats fournis lors de l'analyse des solutions est liée à la qualité de la modélisation et au savoir-faire du « *modélisateur* » (la modélisation est un métier).
- La simulation n'est pas une technique d'optimisation au sens propre. Elle ne peut qu'établir les performances d'une solution conçue et imaginée par l'utilisateur. C'est une *technique* entièrement *itérative* qui ne propose pas de solution finale mais qui permet seulement à l'utilisateur d'envisager des choix possibles. En tout état de cause, c'est lui qui devra décider de ce qui répond le mieux aux problèmes posés.

La difficulté liée à la simulation est double :

- Les résultats de simulation sont souvent complexes à interpréter. On étudie des phénomènes aléatoires et les techniques d'analyse demandent de la rigueur ; il est souvent difficile de faire la part du crucial et de l'anecdotique (le modèle doit être ni trop grossier, ni trop précis).
- Souvent pour des raisons financières, on doit aller au plus vite vers une solution finale (sans passer trop de temps à explorer d'autres solutions intermédiaires).

6. Les étapes du processus de simulation :

La modélisation est une tâche qui fait appel à des compétences scientifiques et informatiques, mais c'est aussi un art. Délimiter le système à étudier et trouver un modèle adéquat sont deux tâches difficiles et dont le résultat dépend des compétences et de l'expérience de l'analyste concepteur du modèle. Un modèle n'est pas bon dans l'absolu, mais relativement aux objectifs de l'étude [AIT05]. En effet, le degré d'abstraction, la nature et le nombre des entités à inclure dans les modèles dépendent, entre autres, de ces objectifs.

Enfin, du point de vue du management, une étude de simulation est un projet dont le succès nécessite une gestion comme tout autre projet [AIT05]. Toutes ces considérations impliquent une décomposition du projet de simulation en plusieurs phases. En effet, Passer du problème à la simulation suppose de franchir un certain nombre d'étapes. Pour Pritsker, celles-ci sont au nombre de dix (figure III.3) :

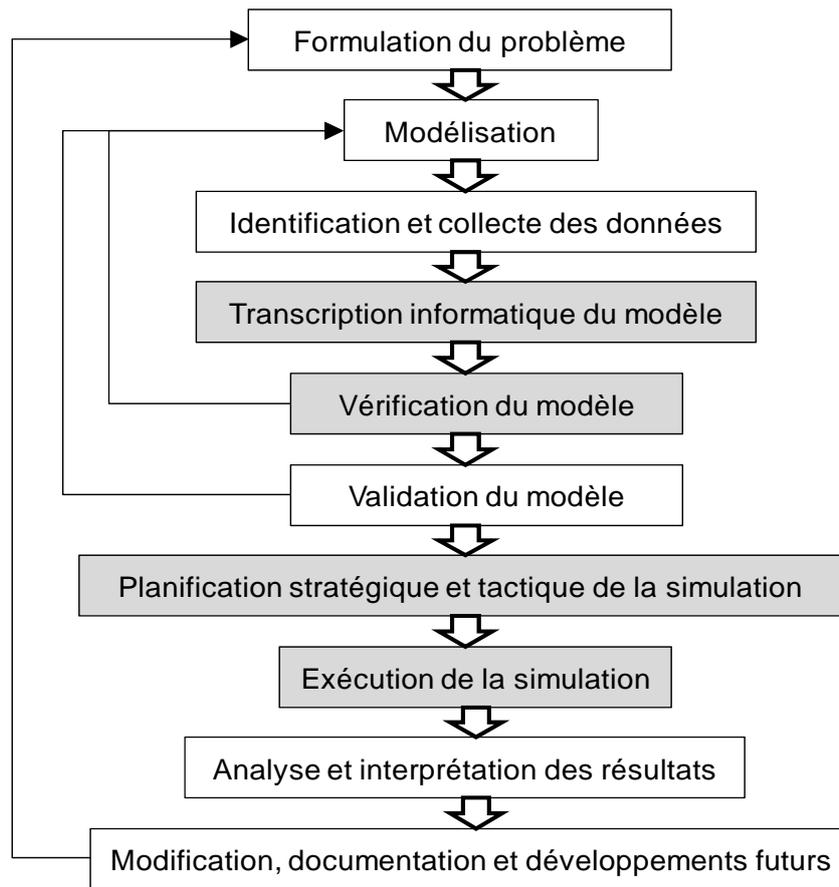


Figure III.3:Les étapes d'un projet de simulation selon Pritsker [HAB01]

Le cycle de cette démarche d'utilisation de la simulation n'est donc pas linéaire. Ces étapes constituent un processus itératif qui, très souvent, n'est pas rigoureusement séquentiel. Le schéma de la figure III.3 illustre la façon dont ces différentes étapes s'articulent dans la pratique classique (se référer à l'annexe 4).

III. Présentation du logiciel Arena 7.01 :

ARENA est un outil de simulation général qui permet de créer une simulation à événements discrets en connectant entre eux différents blocs aux fonctions variées. ARENA peut être utilisé facilement pour représenter des systèmes manufacturiers avec différents niveaux de complexité.

1. Structure d'un programme Arena :

Un programme (ou modèle logiciel) élaboré avec ARENA est sauvegardé dans un fichier ayant pour extension *.doe* et est constitué :

- d'une partie *modèle*, qui représente l'algorithme décrivant les caractéristiques statiques et dynamiques des différents blocs fonctionnels composant le modèle ;
- du *cadre expérimental*, qui regroupe les données précisant les paramètres spécifiques à une simulation donnée (conditions initiales, durée de la simulation, capacité des files d'attente, nombre de ressources,...).

En fait, les entités traversent uniquement les blocs fonctionnels de la partie *modèle*

2. Structure hiérarchique d'ARENA :

Tel qu'il a été mentionné précédemment, ARENA offre plusieurs niveaux de modélisation (figure III.4). Le niveau le plus élevé propose l'utilisation de *templates* pour faciliter la création du modèle tandis que le niveau le plus bas, et par le fait même le plus flexible, propose l'utilisation de SIMAN et de la programmation VBA ou C/C++. SIMAN est un puissant langage de simulation général pour la modélisation de systèmes discrets, continus et/ou combinés.

Tout comme ARENA, SIMAN utilise différents blocs pour générer la séquence dans laquelle vont circuler les entités afin de simuler le système modélisé. Un code SIMAN propre à chaque modèle est ensuite généré, au moment de l'exécution du modèle, selon la séquence créée. De la même façon, lorsqu'une simulation est effectuée avec ARENA, le logiciel examine la configuration et la paramétrisation de chaque module, puis génère un code SIMAN. De plus, tous les blocs ARENA sont créés à partir des blocs SIMAN, également disponibles dans ARENA. Les blocs SIMAN permettent plus de flexibilité que les modules proposés par ARENA. Ils permettent l'entrée de variables ou d'équations algébriques là où les blocs ARENA équivalents ne permettent que l'entrée de valeurs numériques. Toutefois, l'utilisation de blocs SIMAN nécessite plus de rigueur puisque tous les attributs, variables, stations ou ensembles utilisés par les blocs SIMAN ne sont pas générés automatiquement comme c'est le cas pour les blocs ARENA. Ils devront donc être créés à priori à l'aide des éléments SIMAN correspondants.

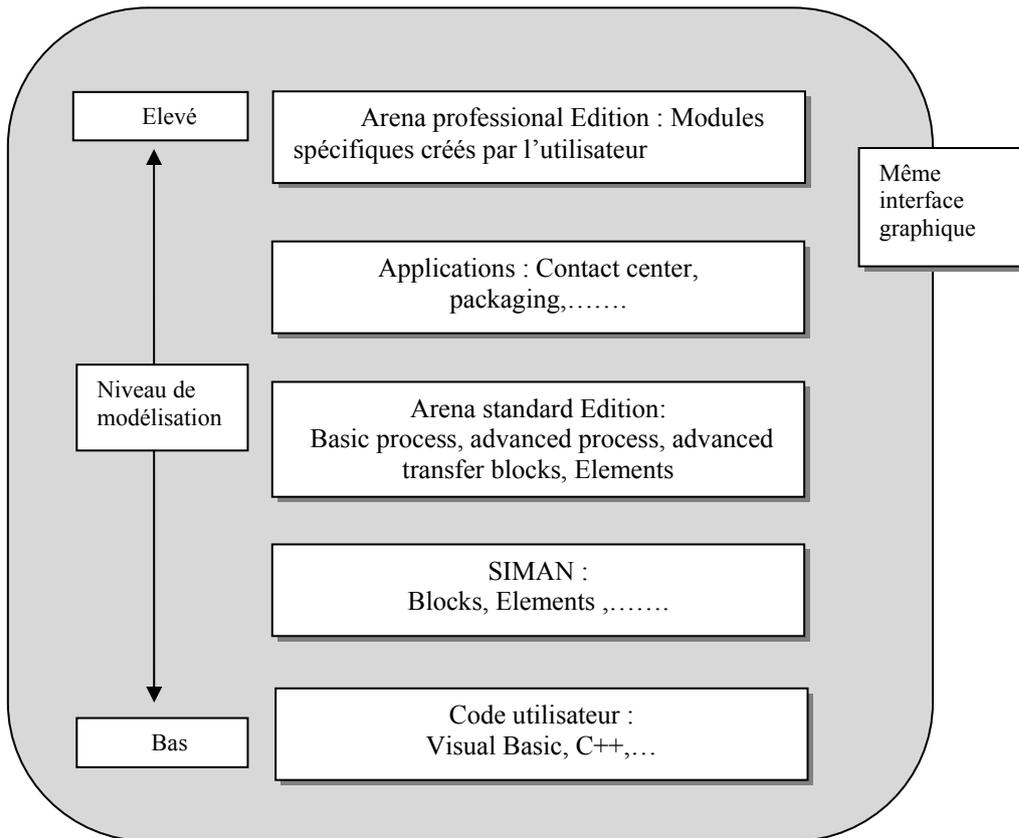


Figure III.4: Structure hiérarchique d'Arena

3. Transfert des entités dans Arena :

Le transfert des entités, dans un modèle de simulation, peut être effectué de différentes façons. Tout d'abord, les entités peuvent circuler d'un bloc à l'autre par une connexion directe entre les blocs. Cette méthode est très utilisée en pratique. Toutefois, pour les systèmes complexes, il est préférable de créer des sous-modèles et d'utiliser des blocs particuliers *Station*, *transfert* et *Route* pour assurer le transfert des entités. Les entités entrent ainsi dans le sous-modèle par un bloc *Station* puis le quittent par un bloc *Route* qui assure le transfert des entités vers la destination déterminée. Un sous-modèle peut, par exemple, représenter un poste de travail.

4. Utilisation de VBA dans Arena :

Le logiciel de simulation ARENA intègre également la programmation VBA et C/C++. L'utilisation de la programmation permet d'améliorer la flexibilité d'ARENA. Toutefois, elle requiert de bonnes connaissances et des habilités supplémentaires au niveau de la programmation. Puisque le simulateur développé n'utilise que la programmation VBA, seule l'utilisation de ce langage de programmation est traitée dans la présente section.

Deux (2) approches sont disponibles pour l'utilisation de VBA dans ARENA: l'utilisation de *blocs VBA* et la *programmation directe via l'éditeur VBA*. La première approche consiste à insérer un bloc VBA directement dans la séquence créée. Ainsi, pendant la simulation, à chaque fois qu'une entité passe par le bloc VBA, la procédure programmée est effectuée. En cliquant directement sur le bloc VBA en question, l'éditeur VBA s'ouvre automatiquement afin de permettre la génération du code. Les procédures de calcul complexes ainsi que les changements particuliers de variables peuvent ainsi être facilement programmés en utilisant le langage VBA.

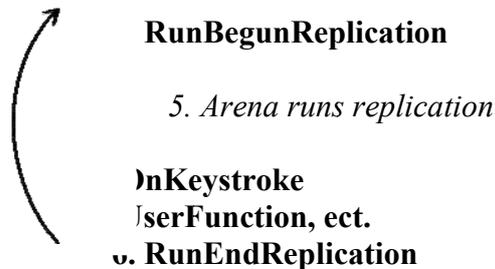
La deuxième approche consiste à coder directement les différentes procédures à partir de l'éditeur de VBA. L'éditeur d'ARENA permet l'utilisation de toutes les méthodes et outils standards de VBA: procédures de programmation (*Sub, Function, Class, If, EndIf, Elseif, While, Wend, Do, On error et Select Case*), la création de formulaires et de boîtes de dialogue en plus des outils de débogage. Le code général disponible dans l'éditeur VBA est divisé en plusieurs parties qui correspondent aux différentes étapes de la simulation ; de l'initialisation à l'exécution de chaque réplication. Il est donc important d'inscrire le code à exécuter à l'endroit approprié en fonction des objectifs à atteindre de façon à ce que la procédure programmée soit activée au moment voulu.

Par exemple, le code inscrit sous *RunBeginSimulation* sera exécuté une seule fois, au tout début de la simulation tandis que le code inscrit sous *RunBeginReplication* sera exécuté au début de chaque réplication. [KEL03] fournit quelques informations sur l'utilisation de VBA avec ARENA.

1. RunBegin

2. Arena checks and initialized the model

3. RunBegin simulation



7. RunEndSimulation

8. Arena terminates the simulation run

9. RunEnd

Figure III.5: Procédure de VBA

La figure ci-dessus présente une liste exhaustive des alternatives de localisation d'un code VBA dans un modèle d'Arena. Arena fait automatiquement appel à chacune de ces procédures lors de la phase de simulation. Si aucun code n'est spécifié pour un événement donné, aucune action n'est exécutée ; Arena se comporte comme si l'événement en question n'existait pas [KEL03].

L'utilisation de VBA permet donc d'atteindre un niveau de flexibilité accru et de faciliter les procédures de calcul complexes. Ainsi, il est possible de générer un modèle de simulation générique et de limiter la manipulation des paramètres et variables par l'utilisateur en automatisant la plupart des procédures avec VBA.

IV. Mise en œuvre du processus de simulation :

Cette section décrit la mise en œuvre du processus de simulation pour la cellule flexible d'assemblage étudiée.

1. Formulation du problème :

Nous procédons à la caractérisation de la cellule selon les éléments suivants :

- Le convoyeur en boucle où les palettes circulent à la même vitesse et sans arrêt ;
- Les stations annexées au convoyeur qui assurent l'entrée des palettes (solidaires aux pièces à assembler) ;
- Les pièces et les palettes présentes sur le convoyeur ;
- Les règles de coopération choisies entre les stations.
- Les files d'attente d'entrée et de sortie à capacité limitée dans les stations.
- Les conditions d'entrée dans les stations suivant les règles de coopération.
- Les conditions de sortie de stations afin de donner la priorité aux palettes circulantes sur le convoyeur ;

Dans le cadre de cette formulation nous avons retenu les hypothèses ci-après :

- Les stations supports des robots sont:
 - indépendantes les unes des autres et disponibles en un seul exemplaire chacune ;
 - disponibles pendant toute la période de l'ordonnancement : il n'y a pas d'arrêt ou de panne ;
 - mono-opération: elles exécutent une seule opération à un instant donné.
- Les opérations sont :
 - non-préemptives : une opération en cours d'exécution ne peut être interrompue ;
 - les temps de préparation des machines sont négligeables, associées à des temps opératoires déterministes.
 - les durées de transport entre deux opérations sont prises en considération.
- Le contexte de l'ordonnancement est de type dynamique : l'ensemble des jobs à ordonnancer est connu à l'avance : il n'y a pas d'aléas, ni d'occurrences de nouveaux jobs au cours de cet ordonnancement. Les paramètres considérés à chaque instant sont :
 - Le nombre de tâches sur le convoyeur ainsi que le nombre des palettes correspondant ;
 - Les états des postes de travail : occupé ou non ;
 - Le nombre d'opérations en attente dans les files d'attente des postes de travail ainsi que les durées opératoires correspondantes ;

- Chaque produit possède une gamme opératoire non linéaire qui se compose de l'ensemble des postes de travail associés à des durées opératoires non nulles.
- Le routage des opérations d'une tâche (produit) donnée, se fait selon un ordonnancement dynamique distribué basé sur des règles de coopération inter machines définies au chapitre précédent.

2. Identification et collecte des données :

Les données du problème sont essentiellement issues des hypothèses de la formulation et des configurations arrêtées et testées lors de la planification stratégique et tactique de la simulation (cf. chapitre 4).

3. Modélisation par l'approche processus :

De nombreux logiciels, et c'est le cas de SIMAN/ARENA, privilégient l'approche processus en tant qu'approche de modélisation. Un processus est une séquence de transformation que subit une entité pendant son séjour dans le système. Une entité est un objet dont l'état est modifié par le processus qu'elle traverse. Les changements d'états nécessitent souvent l'utilisation de ressources. Un modèle est constitué par un ensemble de processus qui interagissent entre eux.

La construction du modèle logique de la cellule d'assemblage à l'aide du logiciel Arena peut dès lors s'articuler en plusieurs étapes successives :

a) Identification des entités

L'entité considérée dans ce système est l'association des entités (représentant les pièces à assembler) solidaires à leurs palettes.

b) Identification des processus

Nous avons recensé six processus :

- Un processus d'entrée ;
- Un premier processus de transformation : la première station ;
- Un second processus de transformation : la seconde station ;
- Un troisième processus de transformation : la troisième station ;
- Un quatrième processus de transformation : la quatrième station ;
- Un processus de sortie.

c) Identification des ressources

- **Les ressources de transformation** : quatre robots, chacun dans une station portant le même indice.
- **Les ressources de stockage** : neuf files d'attente, une à l'entrée du système (Leave 1.Queue), deux files d'attentes propres à chaque station : une d'entrée (FAE1, FAE2, FAE3 et FAE4) et une autre de sortie (FAS1, FAS2, FAS3 et FAS4).

- **Les ressources de transport** : un convoyeur en boucle reliant toutes les stations, quatre petits convoyeurs reliant l'entrée de chaque station à son robot et enfin quatre autres petits convoyeurs reliant le robot de chaque station à sa sortie.

d) Identification des attributs et des variables :

Ils caractérisent le système dans son évolution au cours du temps :

Les attributs : ce sont des variables spécifiques à une entité donnée qu'elle transporte avec elle pendant sa vie dans le système (type d'entité, gamme opératoire, date d'entrée dans le système, etc.). Ces variables sont locales et interviennent dans le pilotage du système (exemple : la gamme de fabrication d'une pièce va déterminer la décision d'envoi de cette pièce sur les machines de l'atelier).

Les variables d'état : celles-ci sont au contraire connues de tous les processus (ce sont des variables globales). Elles concernent en général les durées de séjour des entités dans le système, les taux d'utilisation des ressources et le niveau des encours.

Pour le système étudié, nous avons défini les attributs et les variables suivants :

- Pièce Index : cet attribut est propre au type de l'entité c'est-à-dire au travail qu'elle représente (Job1, Job2, Job3 et Job4).
- D01 : durée opératoire de l'entité par rapport au premier robot.
- D02 : durée opératoire de l'entité par rapport au deuxième robot.
- D03 : durée opératoire de l'entité par rapport au troisième robot.
- D04 : durée opératoire de l'entité par rapport au quatrième robot.
- NumberOut : cette variable représente le nombre de produits finis sortant de la cellule. Il est initialisé à zéro et s'incrémente à chaque sortie d'un produit fini. Cet attribut est utilisé comme un seuil du nombre de pièces à traiter, de telle sorte que ce seuil représente le nombre de pièces constituant une commande. Le but est d'arrêter la simulation dès que l'assemblage d'une commande (ensemble de pièces) est achevé.
- $EG = j \in \{1,2,3,4\}$: Cet attribut signifie que la *station j* est la prochaine station visitée par la pièce. (la *station j* s'est engagée à prendre en charge la pièce à l'issue du processus coopératif)
- TPT_i : représente le temps du prochain traitement de la pièce au niveau d'une station engagée EG.
- VarArray_i : variable qui permet de transmettre les résultats des tests de faisabilités, effectués au niveau des blocks VBA au programme Arena.
- TimeIn : Cet attribut représente le moment d'entrée dans la cellule.

4. Construction du modèle ARENA et transcription informatique du modèle :

Nous avons conçu un programme sous Rockwell ARENA 7.01. Cette étape du processus de simulation est la plus délicate de toutes, car en plus de la création d'un modèle informatique fidèle au système étudié et de la reproduction exacte de son comportement ainsi que de ses composantes, la version du logiciel utilisé est une version académique. Une version académique Rockwell ARENA 7.01 est limitée par :

- le nombre de modules utilisés, pas plus de 150 blocks sachant qu'un module est conçu d'une association de blocks.
- Nombres d'objets SIMAN, pas plus de 250 des composants du types Blocks ou Elements.
- Le nombre des objets cinéma (objets d'animation), pas plus de 150 objets.

Dans ce qui suit, nous décrivons les différentes parties du modèle Arena pour la cellule d'assemblage étudiée. Ce modèle se compose de deux parties : une partie **statique** qui décrit les différents processus identifiés et une partie **dynamique** qui permet d'animer les différentes composantes du système.

A. Modélisation statique de l'atelier d'assemblage : (voir programme annexe 5)

a) Modélisation de l'entrée du système :

L'entrée de la cellule est conçue tel que montré dans la figure III.6 :

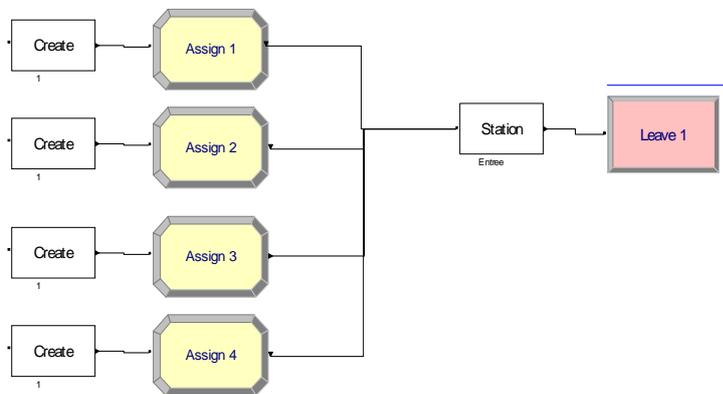


Figure III.6 : Modélisation de l'entrée du système par RockWell Arena 7.01

Nous avons utilisé quatre combinaisons des modules *Create-Assign* afin de créer quatre différents types de pièces. Ils représentent nos quatre travaux (Jobs) considérés dans cette étude : *Pièces 1*, *Pièces 2*, *Pièces 3* et *Pièces 4*.

Grâce au module *Assign*, nous avons affecté différents attributs aux entités créées :

- o Le premier attribut *EG* représente l'engagement d'une station pour une pièce. Il est initialisé à zéro (aucun engagement au début de la simulation).
- o Le second attribut *PieceIndex* représente l'index de l'entité créée.
- o Le troisième attribut *TimeIn* représente le moment d'entrée dans la cellule.

- Les quatre attributs suivants représentent : D01, D02 , D03, D04, les différentes durées opératoires assignées à l'entité. Ces durées sont associées aux robots des stations portant le même indice.
- Un huitième attribut DO représente la somme des durées opératoires associées à une entité donnée et qui sera mis à jour à chaque étape d'assemblage (soustraction de la durée effectuée).
- Le dernier attribut assigne une image propre à l'entité.

Une fois l'entité créée, elle est stockée dans une file d'attente d'entrée nommée *Leave1.Queue* à l'aide de la fonctionnalité *Queue* du module *Leave* où elle reste jusqu'à ce qu'une palette se libère. Dès qu'une palette est libérée, elle est réquisitionnée pour transporter la pièce à l'aide de la fonctionnalité *Request transporter* du module *Leave*. L'entité est alors directement mise en mouvement vers la première station à l'aide de la fonctionnalité *transport* du module *Leave*.

Remarque : Une fois qu'une entité est associée à une palette (transporteur), ARENA leur attribue un seul et unique indice c'est-à-dire qu'ils sont confondus.

b) Modélisation d'une station

La cellule est composée de quatre stations, chacune contenant une file d'attente d'entrée FAE, un robot et une file d'attente de sortie FAS. Une station est modélisée en trois étapes :

- L'entrée de la station (figure III.7);
- Le Robot ainsi que les deux files d'attente d'entrée et de sortie (figure III.9) ;
- La sortie de la station (figure III.10).

○ Modélisation de l'entrée d'une station:

Le submodèle *entrée station 1* est composé d'un block VBA, qui permet au passage d'une entité d'exécuter un algorithme préalablement défini à l'aide de l'éditeur visual basic disponible sur Arena.

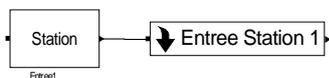


Figure III.7 : Modélisation de l'entrée d'une station par RockWell Arena 7.01

Structure du programme VBA : 

Le programme correspondant au bloc VBA est défini à l'aide de l'éditeur Visual Basic fourni par Arena (voir annexe 6). Ce programme s'exécute à chaque passage de l'entité par ce bloc. Dans notre cas ce programme permet d'effectuer :

- **La récupération des données à partir d'Arena** : à chaque passage d'une entité par le bloc VBA, un ensemble d'information sur l'état du système est automatiquement recueilli à partir du programme principal dont :
 - La valeur des attributs DO_i de l'entité active ;
 - $TNOW$: le moment d'entrée de la pièce dans le block VBA ;
 - $NQ(FAE_i)$: l'état des files d'attente $i=1,4$;
 - EG : la valeur représentant l'engagement des stations ;
- **L'appel à la fonction *FileAttente*** : qui permet d'évaluer le nombre de pièces dans chaque file d'attente j à $TNOW + T_{ij}$ $j \neq i$.
- **Test des conditions d'entrée** : les conditions d'entrée dans une station donnée sont testées au passage de la pièce par le bloc VBA de cette station à l'issu du processus de coopération mis en place entre les ressources afin de connaître le routage de la pièce. Les résultats de l'évaluation sont stockés dans des variables nommées *VarArray*i** et transmis au programme principal via un block *Branch*.
- **Réactualisation de la base de données de la station *i*** (échancier) : qui se serait éventuellement engagée pour une pièce.

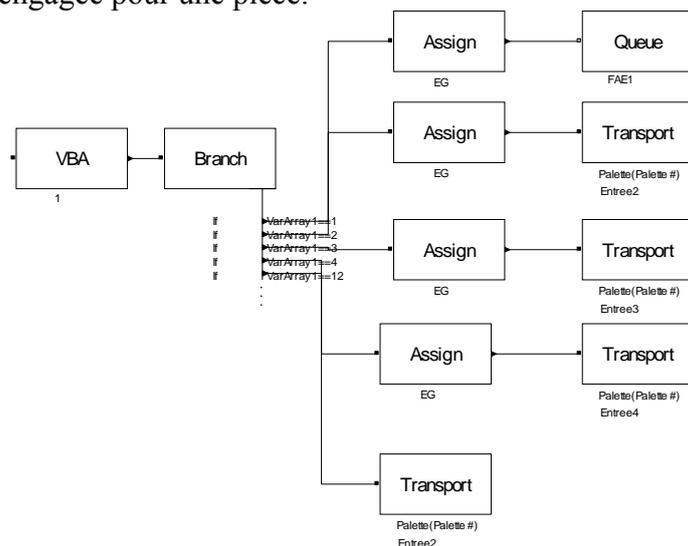


Figure III.8 : Modélisation Submodel « Entrée station 1 » par RockWell Arena 7.01

Les conditions d'entrée, définies par les règles de coopération, sont testées au niveau du block VBA, les résultats sont ensuite enregistrés dans une variable nommée *VarArray*i** préalablement définie dans le spreadsheet *variable*. La valeur de ces variables est vérifiée au niveau du block branchement *Branch* et les actions en conséquence sont alors exécutées. En résumé, les variables *VarArray*i** permettent de coder chaque issue possible du processus de coopération; chaque scénario possible (station choisie pour le traitement de la pièce) correspond à une valeur bien déterminé cette variable.

○ **Modélisation de la station d'assemblage:**

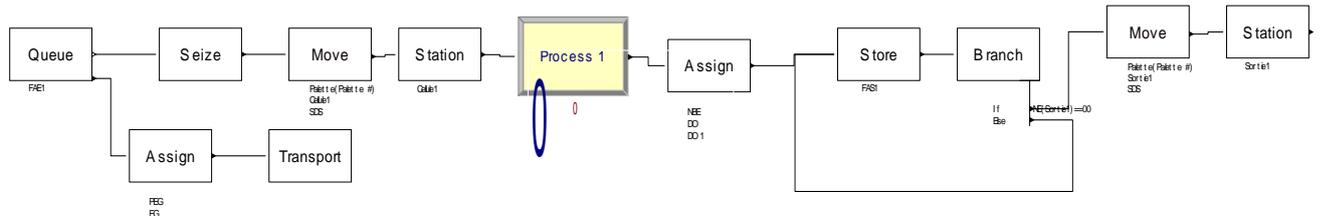


Figure III.9 : Modélisation de la station d'assemblage par RockWell Arena 7.01

Une fois l'entité admise dans la file d'attente, elle y reste jusqu'à ce que le robot soit libéré (La règle de priorité appliquée au niveau de la file d'attente est la règle FIFO). Dès que le tour de la pièce arrive, elle est traitée (*Queue, Seize, Move, Station, Process*). Quand le traitement de la pièce est fini, elle est mise dans une file d'attente de sortie FASi (*Store*) et la durée opératoire effectuée est assignée à zéro pour attester le passage de la pièce dans cette station (*Assign :DOi = 0*).

o **Modélisation de la sortie d'une station:**

Une fois que l'entité est dans la file d'attente de sortie, elle n'en sort que si le point de branchement entre la sortie de la station et le convoyeur n'est pas occupé sachant que la priorité revient aux pièces sur convoyeur (*Branch*). Si ce point de branchement est libre alors la pièce est tout de suite remise sur le convoyeur central sinon elle reste dans la file d'attente de sortie de la station (*Move, Station*). La prochaine station est choisie selon les postes déjà visités (*Branch* : les trois dernières conditions), si c'est la dernière alors la prochaine station est la sortie du système (*Branch* : la première condition).

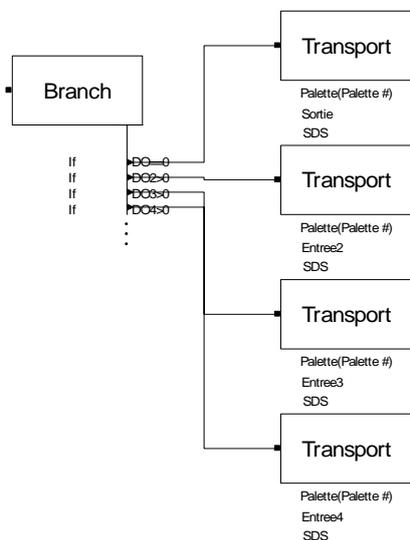


Figure III.10 : Modélisation de la sortie d'une station par RockWell Arena 7.01

c) **Modélisation de la sortie du système:**

A la fin de l'assemblage d'une pièce, elle est tout de suite orientée vers la sortie du système. La sortie du produit est modélisée par la destruction de l'entité représentative (*Dispose*). A la sortie de chaque entité, la variable *NumberOut* est incrémentée de 1 par le block *Assign* (*NumberOut == NumberOut + 1*) ainsi la simulation est arrêtée dès que cette variable atteint la taille d'une commande. Dans le cas d'une commande de 120 pièces, la

condition d'arrêt de la simulation, portée au niveau *Run/Setup/Replication Parameters/Terminating Condition*, est $\text{NumberOut} == 120$.

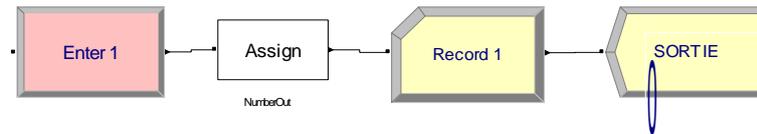


Figure III.11 : Modélisation de la sortie du système par RockWell Arena 7.01

B. Modélisation Dynamique de la cellule d'assemblage :

Le schéma graphique de l'atelier par ARENA est présenté dans la figure III.12 :

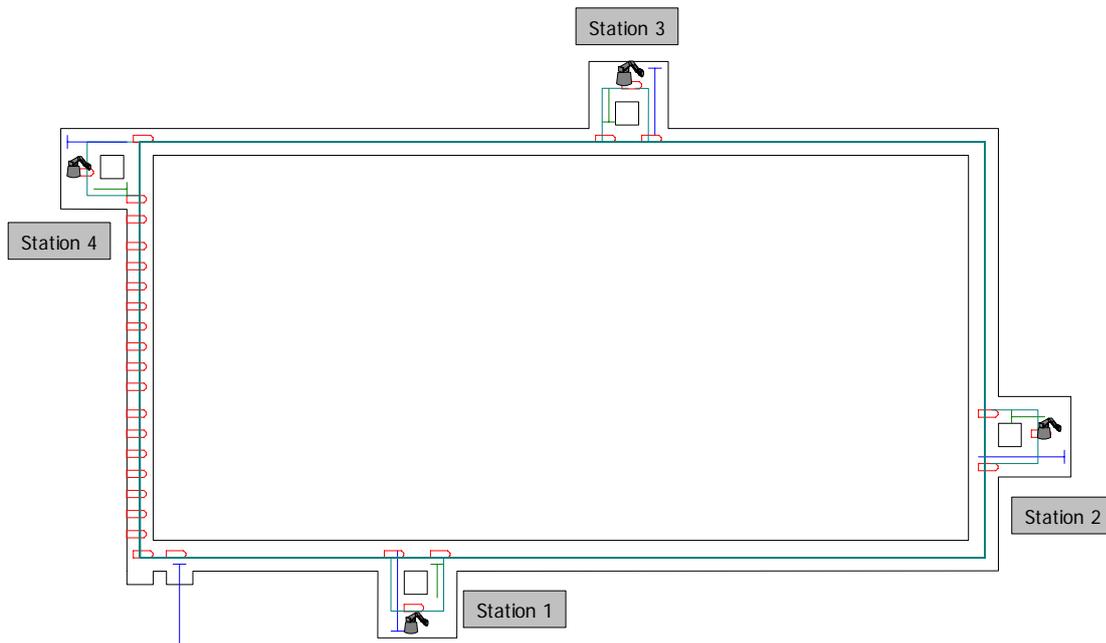


Figure III.12: La représentation graphique de la cellule par RockWell Arena 7.01

Nous avons représenté les différents points d'entrée et de sortie des stations à l'aide de la fonctionnalité *AnimateTransfer/Station* du logiciel Arena, comme montré dans la figure III.13 (a).

Nous avons représenté les différentes files d'attente d'entrée (FAEi) des stations et les différentes files d'attente de sortie des stations (FASi) par *Animate/Queue* (figure III.13 (b), (c))



Figure III.13: a. Station, b. file d'attente d'entrée des stations, c. File d'attente de sortie

Les ressources sont générées automatiquement par *Animate/Ressource* et nous les avons représentées par des figures de robot ramenées de la bibliothèque d'images *Machine.plb* (figure III.14).

Dans notre cas de figure, le robot n'admet que deux états possibles : occupé (a) ou inoccupé (b).

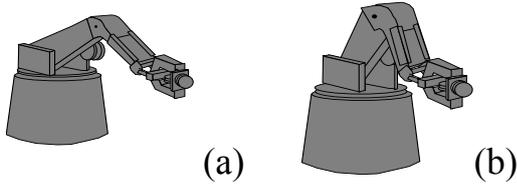


Figure III.14: La représentation graphique des robots

Les palettes sont représentées par des transporteurs générés dans le module *Transporter* (Figure III.15).

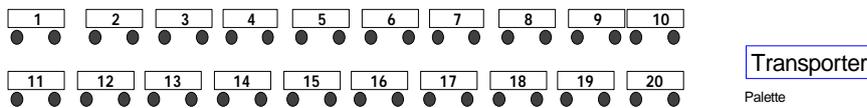


Figure III.15: modélisation des Palette par des transporteurs

Le convoyeur est modélisé par le circuit associé au module *Transporter* du type à cheminement libre composé de tronçons nommés *Distances* (figure III.16).



Figure III.16: modélisation d'un tronçon du circuit par *Distances*

La figure III.17 représente une image du système en pleine évolution. On y voit des palettes sur le convoyeur, des pièces dans les files d'attente ainsi que des robots en pleine exécution.

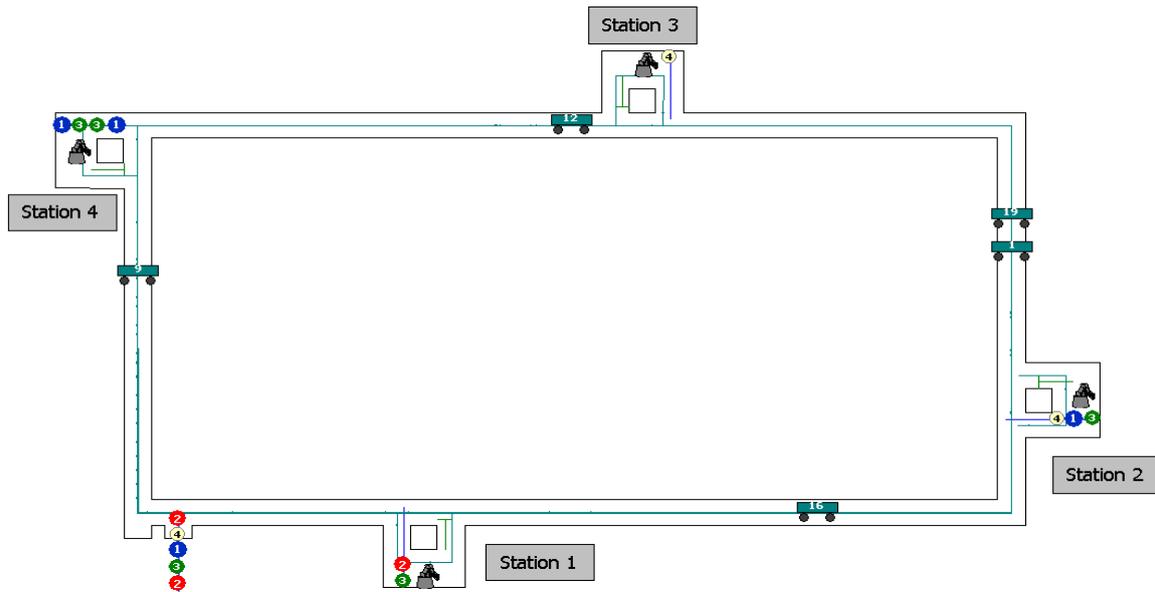


Figure III.17: le système en plein évolution

5. Vérification et validation du modèle :

Afin de s'assurer du bon fonctionnement du modèle et de sa robustesse, nous lui avons fait subir des tests extrêmes. Parmi ces tests nous citons :

- La capacité d'une file d'attente infinie,
- La capacité d'une file d'attente à un, durées opératoires unitaires avec une seule palette pour s'assurer de sa trajectoire sur le convoyeur.
- L'évolution du modèle en créant uniquement 5 pièces
- Utilisation de fichiers sorties pour analyser les données recueillies au cours de la simulation

Par ailleurs, Visual Basic nous permet d'afficher en temps réel durant la simulation, la valeur des attributs calculés au niveau des blocks VBA, ce qui nous permet de garder une traçabilité du fonctionnement du système et de le comparer au fonctionnement souhaité. Nous avons donc pu suivre le fonctionnement du système pour 5, 6, 7 pièces afin de s'assurer que les conditions définies dans les blocks VBA étaient correctement appliquées.

V. Conclusion :

Dans le cadre de ce chapitre, nous avons réalisé les cinq premières étapes du processus de simulation telles que présentées par Pritsker.

La création d'un modèle fidèle au système étudié et sa transcription informatique sont les deux étapes les plus délicates d'un processus de simulation et c'est sur elles qu'une grande partie du temps a été consacrée.

Nous avons donc conçu un modèle de la cellule étudiée à l'aide de l'éditeur graphique d'Arena, ce qui était d'autant plus difficile à cause des limitations et des contraintes imposées par la version académique du logiciel utilisé.

Une fois le modèle validé, nous procédons à l'application des cinq étapes restantes du processus de simulation dans le cadre du chapitre suivant. La prochaine étape est la planification stratégique et tactique de la simulation.

Chapitre IV : Simulation et test de la stratégie d'ordonnancement proposée

- I. Introduction

- II. La planification stratégique et tactique de la simulation
 - 4. Le choix d'une configuration
 - 5. Le plan d'expérimentation
 - 6. Les indicateurs de performance

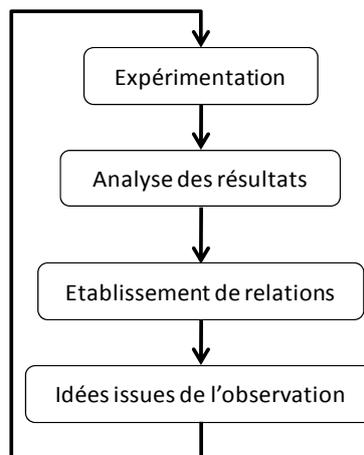
- III. Analyse et interprétation des résultats
 - 8. Analyse des temps d'assemblage d'une commande
 - 9. Analyse des temps de séjour moyens des pièces dans la cellule d'assemblage
 - 10. Analyse des temps de transfert moyens des pièces dans la cellule d'assemblage
 - 11. Analyse des temps d'attente moyens des pièces dans les files d'attente d'entrée
 - 12. Analyse des temps d'attente moyens des pièces dans la cellule d'assemblage
 - 13. Analyse du nombre moyen des pièces en attente dans les files d'attente d'entrée des stations
 - 14. Analyse des taux d'utilisation des robots

- IV. Conclusion

I. Introduction :

La simulation consiste à faire évoluer une abstraction du système au cours du temps afin d'aider à en comprendre le fonctionnement et le comportement pour appréhender certaines de ses caractéristiques dynamiques dans l'objectif d'évaluer différentes décisions [GIA07]. Un modèle de simulation permet de prévoir, à travers les différentes expériences, les performances du futur système de production en fonction des choix qui sont faits tant en ce qui concerne le dimensionnement du système physique que de la conception du système de pilotage.

L'intérêt de cette démarche réside dans le bouclage entre les expérimentations, l'analyse des résultats, l'établissement de relations et la réalisation de nouvelles expérimentations à partir d'idées issues de l'observation, comme le montre le schéma ci-dessous :



Dans ce chapitre, nous mettrons en œuvre les étapes restantes du processus de simulation, telles que définies par Pritsker (cf. Chapitre 3), c'est-à-dire la planification stratégique et tactique de la simulation, son exécution selon l'étape précédente et enfin l'analyse et l'interprétation des résultats. Ainsi, la deuxième section du chapitre est consacrée à l'élaboration du plan d'expérimentation pour la simulation et à la définition des indicateurs de performance.

Dans la troisième section, nous examinerons les résultats obtenus après l'exécution de la simulation. En fonction de l'analyse des données obtenues, les différentes expérimentations planifiées dans la section 2 pourront donner lieu à des expérimentations complémentaires non programmées initialement ou à d'éventuelles modifications au niveau de la configuration initiale de la cellule.

II. La planification stratégique et tactique de la simulation :

Le processus de simulation aboutit, après la phase de conception et d'implémentation informatique du modèle, à la phase d'expérimentation et d'interprétation des résultats. Selon l'objectif de l'étude, les paramètres à considérer lors des expérimentations peuvent être très nombreux (la longueur du convoyeur central, le nombre de stations, la capacité des files d'attente d'entrée, le nombre de palettes, le nombre de tâches, la taille des commandes, la fréquence d'entrée des pièces dans la cellule, ...etc.), chacun pouvant avoir une influence sur la performance globale du système étudié et pouvant faire l'objet de variations diverses [GHA06].

Ceci rend nécessaire le passage par une phase de planification judicieuse des expériences qui permet de rationaliser le nombre d'expériences à mener tout en assurant une couverture suffisante du problème étudié.

Pour ce faire, nous commencerons par arrêter une configuration de base de la cellule en définissant : la dimension du convoyeur, le nombre de stations, le nombre de jobs et le nombre de palettes conformément aux hypothèses retenues lors de la construction du modèle conceptuel (cf. chapitre 3). Nous identifierons dans un second temps les paramètres physiques ainsi que les règles de pilotage à faire varier en définissant un plan d'expérimentation (Annexe 6). Enfin, nous présenterons les différents indicateurs retenus pour analyser la performance du système étudié en cohérence avec l'objectif de notre étude.

Les objectifs de ces expérimentations sont multiples :

- Recueillir des données pour pouvoir analyser le comportement dynamique du système en fonction des paramètres de pilotage.
- Proposer une méthodologie permettant, à partir de l'analyse des données recueillies, de déterminer les valeurs optimales des paramètres de pilotage pour minimiser une variable de sortie (temps de séjour d'une commande, cycle moyen d'une pièce,...)
- Trouver des relations permettant de dimensionner certains paramètres physiques (nombre de palettes, capacité des ressources de stockage,...)

1. Le choix d'une configuration :

La configuration arrêtée pour l'expérimentation est une cellule composée de quatre stations, dont le nombre de palettes est de 20 et la longueur du convoyeur est de 500 palettes. Cette configuration est la même que celle retenue dans [GHA06]. Les distances entre les différentes stations sont représentées dans le tableau IV.1.

	Sortie	Entrée	Entrée1	Sortie1	Entrée2	Sortie2	Entrée3	Sortie3	Entrée4	Sortie4
Sortie	500	1	49	51	199	201	319	321	449	451
Entrée	499	500	48	50	198	200	318	320	448	450
Entrée1	451	452	500	2	150	152	270	272	400	402
Sortie1	449	450	498	500	148	150	268	270	398	400
Entrée2	301	302	350	352	500	2	120	122	250	252
Sortie2	299	300	348	350	498	500	118	120	248	250
Entrée3	181	182	230	232	380	382	500	2	130	132
Sortie3	179	180	228	230	378	380	498	500	128	130
Entrée4	51	52	100	102	250	252	370	372	500	2
Sortie4	49	50	98	100	248	250	368	370	498	500

Tableau IV.1 : Les distances entre stations

Le nombre de jobs (types de pièces à assembler) considérés est de 4, chacun étant composé de 4 tâches qui sont exécutés au niveau des 4 stations. La règle de priorité appliquée au niveau des FAEs est FIFO.

Nous rappelons que, le problème considéré est $O4|\tau_j|\bar{F}$, tel que τ_j représente les temps de transport entre les stations et F le temps de séjour total (Total Flow Time).

2. Le plan d'expérience :

La simulation est une méthode numérique, expérimentale, d'étude de situations complexes. A ce titre elle requiert la construction de plans d'expérience pouvant comporter de nombreux facteurs [COH89]. Dans notre cas, nous nous sommes intéressés aux paramètres influençant directement le temps de séjours des pièces dans la cellule. Nous ferons varier les paramètres suivants :

- Le nombre de gammes non-linéaires testées est de 50 : Dans un souci de maintien de l'équilibre entre les stations (Line balancing), les durées opératoires ont été choisies de sorte que les différences entre les durées des travaux (jobs) ne dépassent pas les 25 %. Ces durées sont données dans le tableau 2 de l'annexe 6.
- La capacité des FAEs des stations : les capacités testées sont 3, 4, 5. Les choix des capacités des FAEs sont faits de sorte que le nombre d'emplacements d'attente global dans les stations de la cellule soit proche du nombre de palettes en circulation sur le convoyeur central ($3 \times 4 = 12$, $4 \times 4 = 16$, $4 \times 5 = 20$) ;
- Les règles de pilotage : nous testerons deux variantes (règle1, règle2) de l'approche d'ordonnancement basée sur la coopération inter machines décrite au chapitre 2 ainsi qu'une approche basée uniquement sur les règles de priorité de type FIFO au niveau des FAEs. Notre objectif étant de comparer la performance des deux approches par rapport aux temps de séjours.

Sur la base de ce qui précède, le nombre d'expérience à exécuter est 450. En effet, pour chaque configuration arrêtée (capacité 3, 4, 5), nous avons fait varier les durées opératoires (contenues dans le tableau 2 de l'annexe 6) afin d'obtenir les différentes variantes de produits ce qui nous donne 50 cas possibles pour chacune des configurations. Enfin, nous avons fait varier les règles de pilotage qui nous donnent 3 expériences pour chaque combinaison produit/capacité de file d'attente. Ainsi pour chacune des expériences nous obtenons 150 tests pour une capacité donnée de la FAE, ce qui nous ramène aux 450 expériences recensées plus haut.

Le tableau IV.2 ci après reprend clairement le plan d'expérimentation :

N° de l'expérience	Capacité des FAEs	Durées opératoires (Dos)	Règles de pilotage
1	3	50 combinaisons de 4 produits à assembler, chacun d'eux est représenté par une combinaison de 4 durées opératoires	Test de 3 règles de pilotage : règle1, règle2, FIFO (fractionnés en 150 cas)
2	4	50 combinaisons de 4 produits à assembler, chacun d'eux est représenté par une combinaison de 4 durées opératoires	Test de 3 règles de pilotage : règle1, règle2, FIFO (fractionnés en 150 cas)
3	5	50 combinaisons de 4 produits à assembler, chacun d'eux est représenté par une combinaison de 4 durées opératoires	Test de 3 règles de pilotage : règle1, règle2, FIFO (fractionnés en 150 cas)
Le but de l'étape	Influence des capacités des files d'attente sur le temps de séjour des pièces dans la cellule.	Influence des durées opératoires des produits à assembler sur la performance de la cellule (la performance de la cellule est-elle stable ? dépend elle des durées opératoires ?).	Influence des règles de coopération sur la performance de la cellule : minimisation du temps moyen d'attente dans les FAEs, minimisation du temps moyen de transfert des pièces sur le convoyeur (Recherche d'un ordonnancement plus judicieux entre les stations).

Tableau IV.2 : résumé du plan expérimental

Règle1 (*règle générique*) : coopération inter machines afin de déterminer la station la plus adaptée pour le traitement de la pièce : Choix de la station ayant **la file d'attente**

- **la moins saturée** ;
- **non encore visitée** ;
- **et la plus proche** de l'emplacement de la pièce sur le convoyeur.

En résumé, les machines coopèrent pour déterminer la station ayant la file d'attente telle que :

$$NFAE_i(TNOW + T_{ij}) = \min \{NFAE_j(TNOW + T_{ij})\} j = 1,4$$

Règle2 (*modification1*) : règle1+ **prise en compte de l'état de la ressource** : on distingue entre l'état « FAE non saturée, robot inoccupé » et « FAE non saturée, robot occupé ».

3. Les indicateurs de performance à analyser :

Pendant la phase de l'analyse des résultats (section 3), nous mesurerons la performance de chacune des expériences exécutées par rapport à FIFO à l'aide d'une batterie d'indicateurs de performance décrite ci-dessous :

a) *Les taux d'amélioration des temps de séjours d'un lot dans la cellule* $\tau_{TSCRègle}$:

Dans la première partie, les temps étudiés sont les temps de séjour d'une commande (120 pièces). L'amélioration correspond à une réduction du temps de séjour d'une commande ou

Cmax (sous ARENA, ce temps est nommé Stop Time). Les taux d'amélioration des temps de séjour d'un lot dans la cellule $\tau_{TSCR\grave{e}gle}$ sont calculés comme suit :

$$\tau_{TSCR\grave{e}gle} = \frac{(T_{SCFIFO} - T_{SCR\grave{e}gle})}{T_{SCFIFO}} * 100 \quad \text{tel que :}$$

- T_{SCFIFO} est le temps de séjour d'une commande sous un pilotage FIFO
- $T_{SCR\grave{e}gle}$ est le temps de séjour d'une commande sous le pilotage de la Règle utilisée.

b) Les taux d'amélioration des temps de séjour $\tau_{TSR\grave{e}gle}$:

Dans la seconde partie, les temps étudiés sont les temps de séjour moyens des pièces dans la cellule (sous ARENA, ces temps sont nommés Total Time (Entity)). Les taux d'amélioration des temps de séjour $\tau_{TSR\grave{e}gle}$ sont calculés comme suit :

$$\tau_{TSR\grave{e}gle} = \frac{(T_{SFIFO} - T_{SR\grave{e}gle})}{T_{SFIFO}} * 100 \quad \text{tel que :}$$

- T_{SFIFO} est le temps de séjour d'une pièce sous un pilotage FIFO ;
- $T_{SR\grave{e}gle}$ est le temps de séjour d'une pièce sous le pilotage de la Règle utilisée.

c) Les taux d'amélioration des temps de transfert $\tau_{TTR\grave{e}gle}$:

Dans la troisième partie, les temps étudiés sont les temps de transfert moyens des pièces dans la cellule (sous ARENA, ces temps sont nommés Transfert Time (Entity)). Les taux d'amélioration des temps de transfert $\tau_{TTR\grave{e}gle}$ sont calculés comme suit :

$$\tau_{TTR\grave{e}gle} = \frac{(T_{TFIFO} - T_{TR\grave{e}gle})}{T_{TFIFO}} * 100 \quad \text{tel que :}$$

- T_{TFIFO} est le temps de transfert d'une pièce sous un pilotage FIFO ;
- $T_{TR\grave{e}gle}$ est le temps de transfert d'une pièce sous le pilotage de la Règle utilisée.

d) Les taux d'amélioration des temps d'attente dans les FAEs $\tau_{TAFR\grave{e}gle}$:

Dans la quatrième partie, les temps étudiés sont les temps d'attente moyens des pièces dans les files d'attente d'entrée des stations de la cellule (sous ARENA, ces temps sont nommés Waiting Time (Queue)). Les taux d'amélioration des temps d'attente dans les FAEs $\tau_{TAFR\grave{e}gle}$ sont calculés comme suit :

$$\tau_{TAFR\grave{e}gle} = \frac{(T_{AFFIFO} - T_{AFR\grave{e}gle})}{T_{AFFIFO}} * 100 \quad \text{tel que :}$$

- T_{AFFIFO} est le temps d'attente d'une pièce dans une FAE sous un pilotage FIFO ;
- $T_{AFR\grave{e}gle}$ est le temps d'attente d'une pièce dans une FAE sous le pilotage de la Règle utilisée.

e) Les taux d'amélioration des temps d'attente moyens dans la cellule $\tau_{TAR\grave{e}gle}$:

Dans la cinquième partie, les temps étudiés sont les temps d'attente moyens des pièces dans la cellule ce qui correspond à la somme des temps d'attente de la pièce dans les différentes

FAEs. Les taux d'amélioration des temps d'attente dans les FAEs $\tau_{TAR\grave{e}gle}$ sont calculés comme suit :

$$\tau_{TAR\grave{e}gle} = \frac{(T_{AFIFO} - T_{AR\grave{e}gle})}{T_{AFIFO}} * 100 \quad \text{tel que :}$$

- T_{AFIFO} est le temps d'attente d'une pièce dans la cellule sous un pilotage FIFO ;
- $T_{AR\grave{e}gle}$ est le temps d'attente d'une pièce dans la cellule sous un pilotage de la Règle utilisée.

f) Les taux d'amélioration du nombre de pièces en attente dans les FAEs $\tau_{NAFR\grave{e}gle}$:

Dans la sixième partie, les données étudiées sont le nombre moyen de pièces en attente dans les files d'attente d'entrée des stations de la cellule (sous ARENA, ces temps sont nommés Number Waiting (Queue)). Les taux d'amélioration du nombre de pièces en attente dans les FAEs $\tau_{NAFR\grave{e}gle}$ sont calculés comme suit :

$$\tau_{NAFR\grave{e}gle} = \frac{(N_{AFFIFO} - N_{AFR\grave{e}gle})}{N_{AFFIFO}} * 100 \quad \text{tel que :}$$

- N_{AFFIFO} est le nombre de pièces en attente dans une FAE sous un pilotage FIFO ;
- $N_{AFR\grave{e}gle}$ est le nombre de pièces en attente dans une FAE sous le pilotage de la Règle utilisée.

g) Les taux d'amélioration des taux d'utilisation des robots $\tau_{TURR\grave{e}gle}$:

Dans la septième partie, les données étudiées sont les taux moyens d'utilisation des robots de la cellule (sous ARENA, ces données sont nommées Instantaneous Utilization (Resource)). Une amélioration correspond à une augmentation de ces taux. Les taux d'amélioration des taux d'utilisation des robots $\tau_{TURR\grave{e}gle}$ sont calculés comme suit :

$$\tau_{TURR\grave{e}gle} = \frac{(T_{URR\grave{e}gle} - T_{URFIFO})}{T_{URFIFO}} * 100 \quad \text{tel que :}$$

- T_{URFIFO} est le taux d'utilisation du robot Ri sous un pilotage FIFO ;
- $T_{URR\grave{e}gle}$ est le taux d'utilisation du robot Ri obtenu par la Règle utilisée.

III. Analyse et interprétation des résultats :

1. Analyse des temps d'assemblage d'une commande (Cmax) :

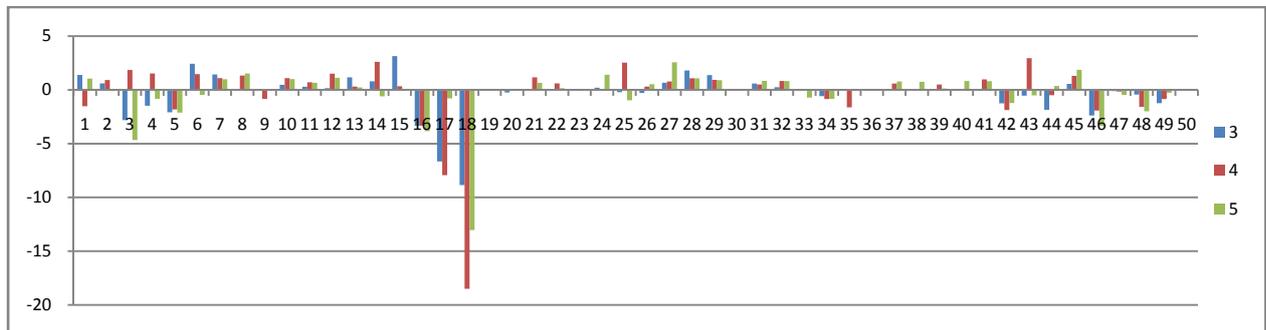


Figure IV.1 : Les taux d'amélioration des temps d'assemblage d'une commande par la règle 1

Par l'application de la règle 1 (Figure IV.1), on remarque que la distribution des taux d'amélioration est concentrée vers la partie supérieure de l'axe des abscisses à l'exception des expériences 16, 17 et 18, qui correspondent aux cas dont les durées opératoires sont les plus faibles relativement au temps de transfert sur le convoyeur. Néanmoins, l'amélioration des temps d'assemblage ne dépasse pas les 3.14%. Ce taux est obtenu par la configuration dont la capacité des files d'attente est de 3 (expérience N°15).

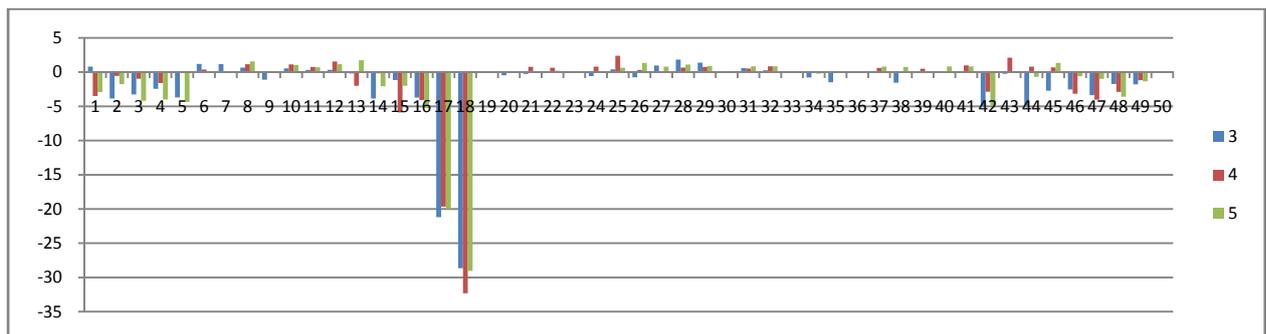


Figure IV.2 : Les taux d'amélioration des temps d'assemblage d'une commande par la règle 2

Par l'application de la règle 2 (Figure IV.2), on remarque très peu de cas d'amélioration. De même que pour la règle 1, les cas de dégradation les plus importants sont obtenus pour les expériences 16, 17 et 18, qui correspondent aux cas dont les durées opératoires sont les plus faibles relativement au temps de transfert sur le convoyeur. L'amélioration des temps d'assemblage ne dépasse pas les 2.4%. Ce taux est obtenu par la configuration dont la capacité des files d'attente est de 4 (expérience N°25).

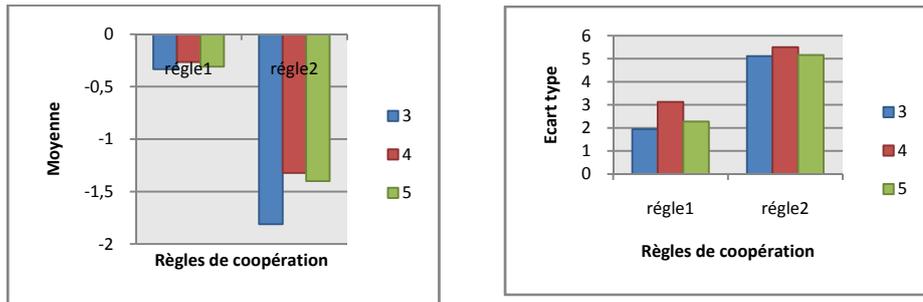


Figure IV.3 : Moyenne et écart type des temps de séjours d'une commande dans la cellule

Pour les 3 configurations (capacité 3, 4, 5), les taux d'amélioration moyens sont négatifs atteignant $-1.8\% \pm 5.11$ dans le cas où la capacité des FAEs est de 3 sous la règle 2 (Figure IV.3). Pour les 2 règles de coopération, on remarque que le taux de dégradation moyen le plus important est obtenu dans le cas où la capacité des FAEs est de 3. D'autre part, le taux de dégradation moyen le plus faible est obtenu dans le cas où la capacité des FAEs est de 4.

Cependant, en recalculant les moyennes des taux d'amélioration sans prendre en considération les cas 16, 17 et 18, nous obtenons des taux d'amélioration moyens positifs pour la règle 1 ne dépassant pas $0.35\% \pm 1.15$ dans le cas où la capacité des FAEs est de 4.

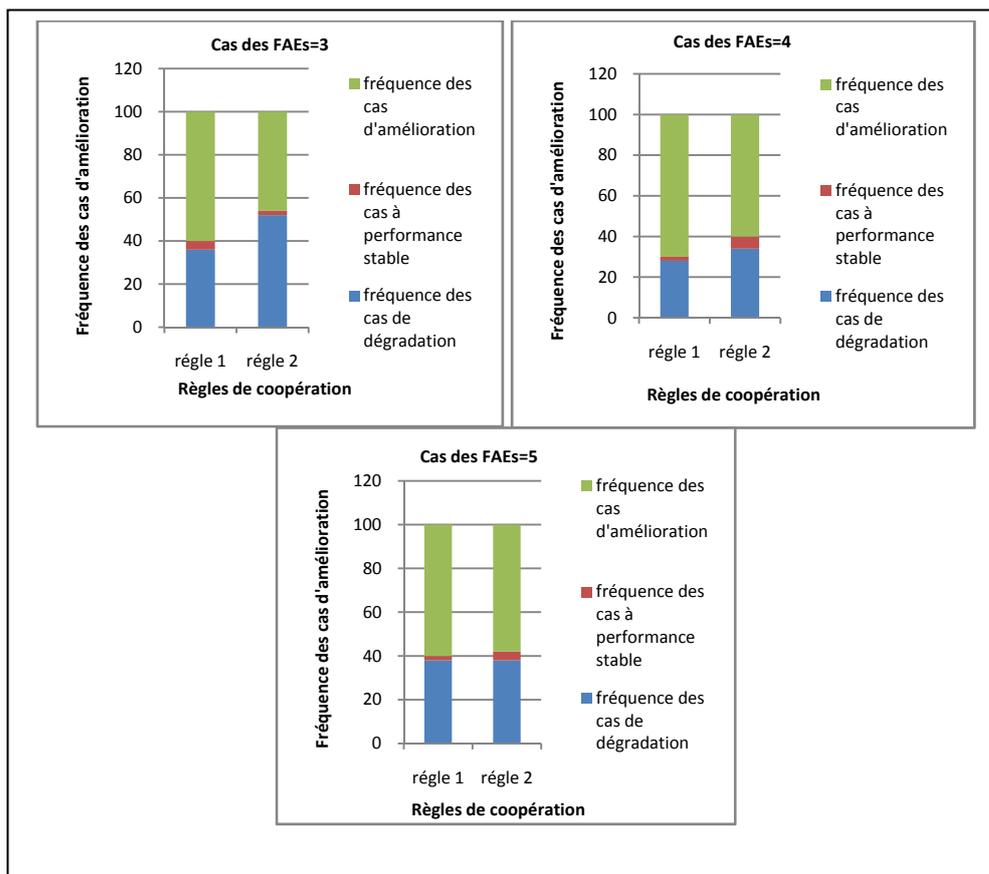


Figure IV.4 : Fréquence des cas d'amélioration du Cmax

Conformément à la figure IV.4, nous pouvons constater que, pour les 3 configurations (capacité 3, 4, 5), la fréquence des cas d'amélioration est supérieure à 50 %, à l'exception du

cas dont la capacité des FAEs est de 3 sous la règle 2. Le pourcentage maximal d'amélioration est de 70% pour le cas dont la capacité des FAEs est de 4 sous la règle 1.

L'écart de performance entre la règle 1 et la règle 2 se réduit avec l'augmentation des capacités des FAEs. En effet, dans le cas où la capacité des FAEs est de 5, le nombre des cas d'amélioration est sensiblement identique pour les 2 règles de coopération.

Lors de l'analyse des taux d'amélioration des temps de séjours d'une commande dans la cellule, nous avons constaté que les taux d'amélioration moyens sont négatifs pour toutes les configurations testées. Bien que le nombre de cas d'amélioration soit supérieur au nombre de cas de dégradation, les taux de dégradation restent plus importants, principalement pour les durées opératoires faibles, et atteignent -32.23% dans le cas où la capacité des FAEs est de 4 sous la règle 2. Nous déduisons que les règles de coopération n'apportent pas d'amélioration significative au temps d'assemblage d'une commande (ne dépasse pas 3.14%), mais présentent une performance stable et régulière.

2. Analyse des temps de séjour moyens des pièces dans la cellule d'assemblage :

a) Taux d'amélioration des temps de séjour moyens des pièces dans la cellule d'assemblage par la règle 1 :

Par l'application de la règle 1, nous remarquons que la distribution des taux d'amélioration est globalement concentrée sur la partie supérieure de l'axe des abscisses. Néanmoins, l'amélioration des temps de séjours ne dépasse pas les 5%, 7%, 4.76%, pendant que la détérioration atteint les 15%, 20%, 22,5%, dans l'ordre des figures IV.5, IV.6, IV.7. De plus, il est possible de constater que les taux de dégradation les plus importants correspondent aux durées opératoires les plus faibles du plan expérimental et plus précisément au cas 18.

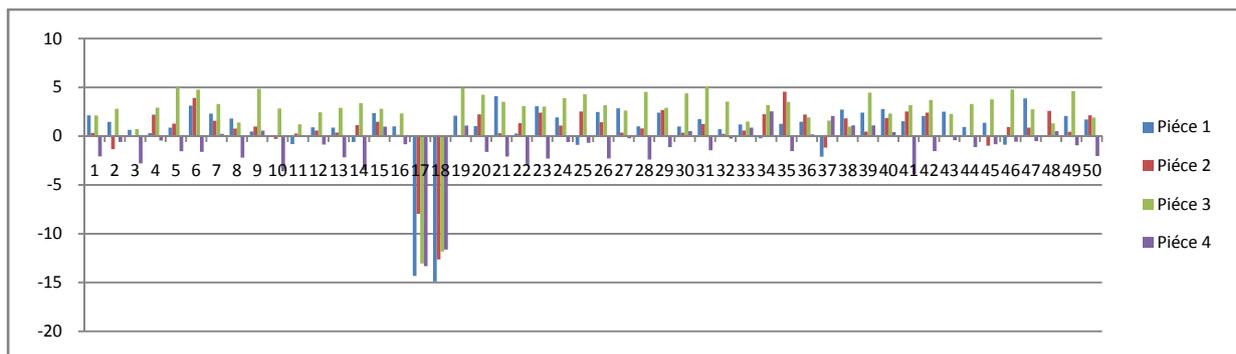


Figure IV.5 : Taux d'amélioration des temps de séjour moyens des pièces par la règle 1 pour les cas dont la capacité des FAEs est de 3

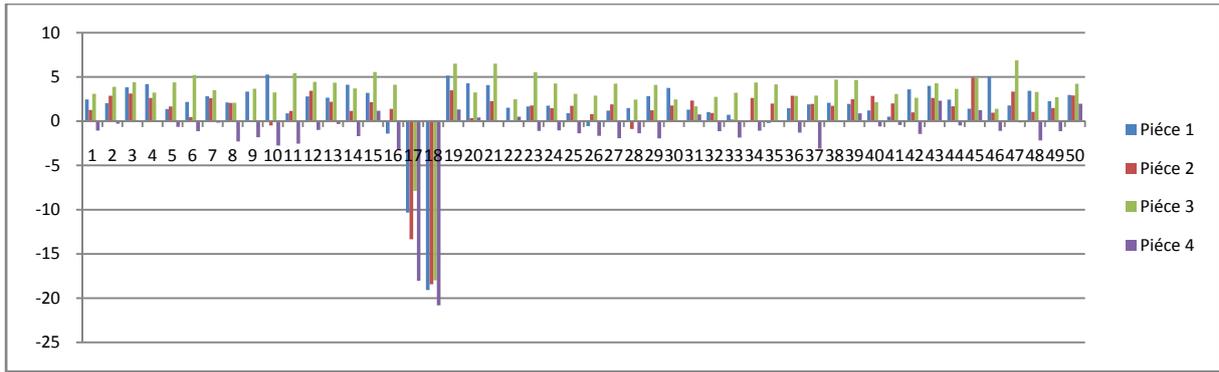


Figure IV.6 : Taux d'amélioration des temps de séjour moyens des pièces par la règle 1 pour les cas dont la capacité des FAEs est de 4

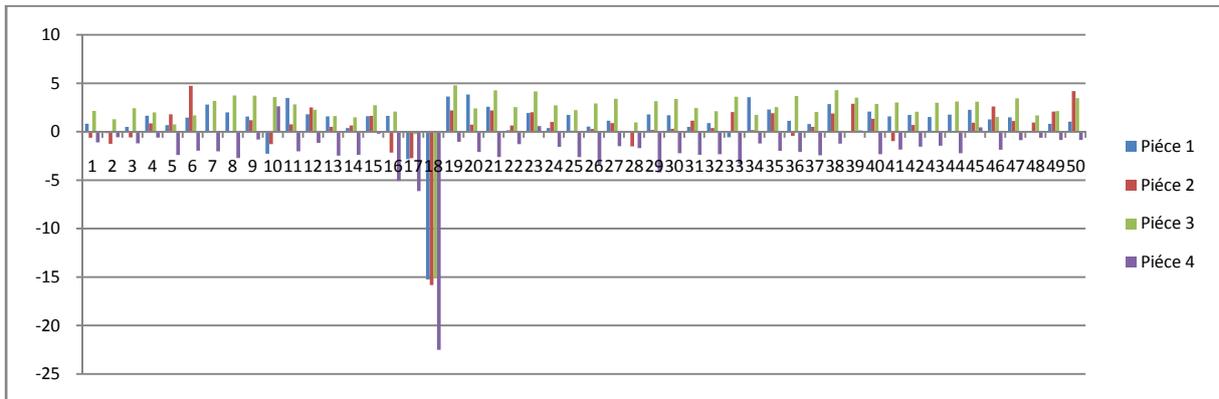


Figure IV.7 : Taux d'amélioration des temps de séjour moyens des pièces par la règle 1 pour les cas dont la capacité des FAEs est de 5

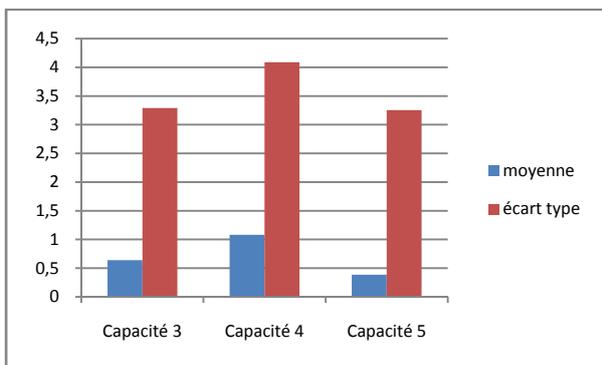


Figure IV.8 : Moyenne et écart type des taux d'amélioration des temps de séjours moyens des pièces par la règle 1

En résumé (Figure IV.8), la règle 1 réalise une amélioration pour les 3 configurations testées. Néanmoins, le taux d'amélioration moyen ne dépasse pas les $(1.08\% \pm 4)$ pour le cas dont la capacité des FAEs est de 4. Par ailleurs, on peut constater que la règle 1 présente une performance stable et régulière à l'exception des cas 16, 17, 18 qui présentent une forte dégradation. En effet, en recalculant les moyennes des taux d'amélioration sans prendre en considération les cas 16, 17 et 18, nous obtenons des moyennes positives

pour la règle 1 de l'ordre de 1,8% dans le cas où la capacité des FAEs est de 4, avec un écart type de 2%, ce qui traduit une concentration des taux d'amélioration autour de la moyenne et par conséquent une stabilité au niveau de la performance réalisée par la règle 1.

b) Taux d'amélioration des temps de séjour moyens des pièces dans la cellule d'assemblage par la règle 2 :

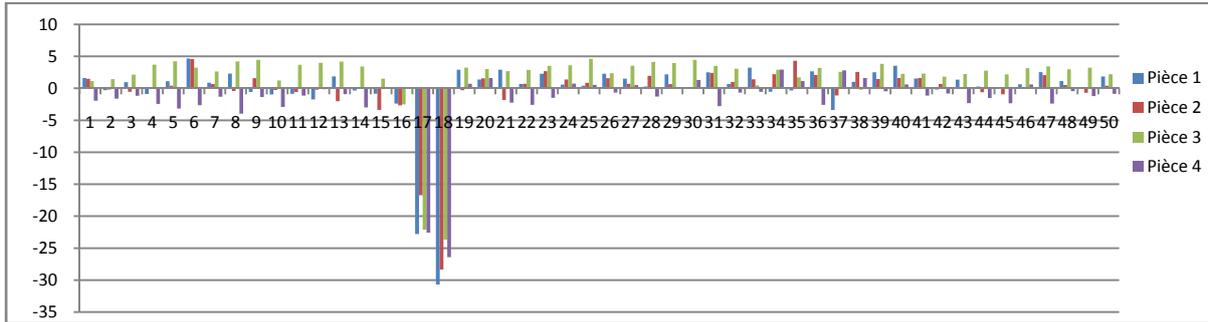


Figure IV.9 : Taux d'amélioration des temps de séjour moyens des pièces par la règle 2 pour les cas dont la capacité des FAEs est de 3

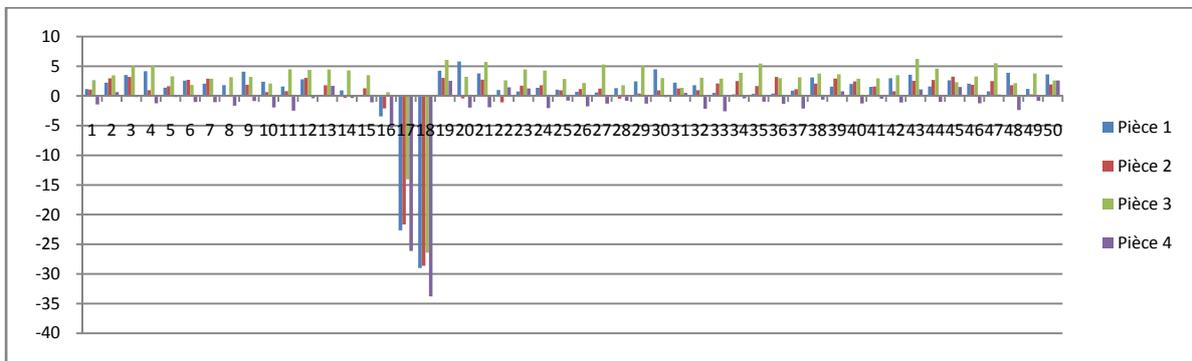


Figure IV.10 : Taux d'amélioration des temps de séjour moyens des pièces par la règle 2 pour les cas dont la capacité des FAEs est de 4

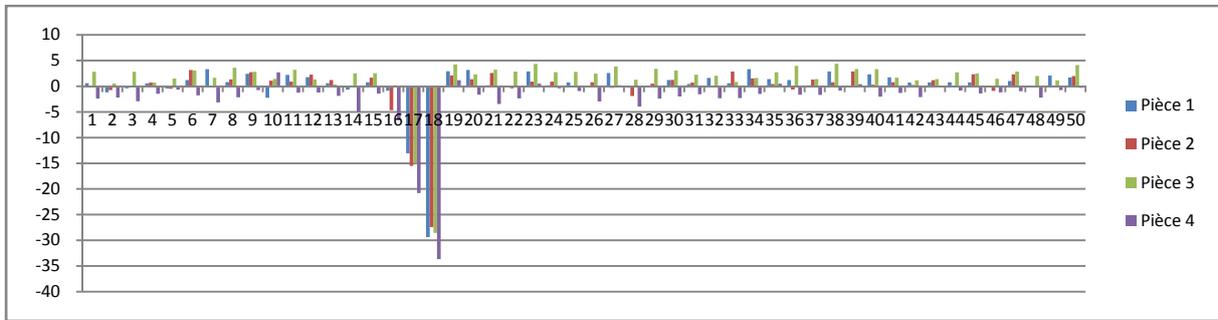


Figure IV.11 : Taux d'amélioration des temps de séjour moyens des pièces par la règle 2 pour les cas dont la capacité des FAEs est de 5

Par l'application de la règle 2, on remarque pour les 3 configurations que la distribution des taux d'amélioration est plus dense vers le haut (Figures IV.9, IV.10, IV.11), mais les taux sont plus importants vers le bas de l'axe (expériences 16, 17 et 18). L'amélioration des temps de séjour ne dépasse pas les 4.65%, 6.23%, 4.38% dans l'ordre des figures IV.9, IV.10, IV.11.

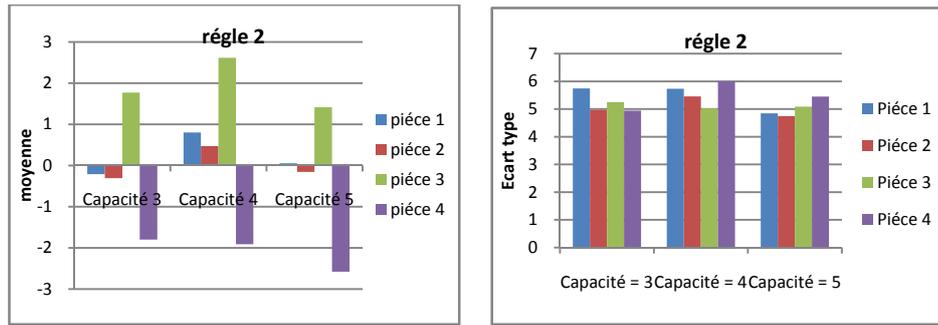


Figure IV.12 : Moyenne et écart type des taux d'amélioration des temps de séjour moyens des pièces par la règle 2

Le taux d'amélioration moyen maximal ($2.61\% \pm 5.02$) est obtenu dans le cas où la capacité des FAEs est de 4 pour les pièces de type « pièce3 » (Figure IV.12). Pour les 3 configurations, la moyenne des taux d'amélioration des pièces de type « pièce4 » est négative atteignant (-2.57 ± 5.45) dans le cas où la capacité des FAEs est de 5 (Figure IV.12).

La figure IV.13 présente une synthèse générale des résultats obtenus par les règle 1 et règle 2. Il est ainsi possible de constater que la règle 1 présente une amélioration moyenne plus importante que la règle 2 qui présente une dégradation moyenne pour les cas dont la capacité des FAEs est de 3 ou 5. Par ailleurs, l'analyse des taux d'amélioration moyens des temps de séjour démontre que l'impact de l'ordonnement coopératif sur la performance du temps de cycle reste insignifiant mais que la performance obtenue est plutôt stable et régulière. L'absence d'amélioration significative s'explique par l'effet des interactions entre les pièces circulant en même temps sur le convoyeur. En effet, les améliorations gagnées sur certains types de pièces (essentiellement pour les types « pièce1 », « pièce2 », « pièce3 ») sont absorbées par les retards acquis par d'autres (celles de type « pièce4 »)

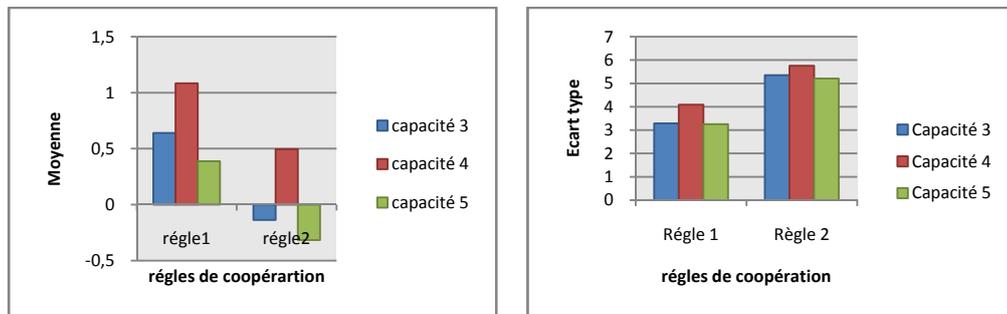


Figure IV.13 : Moyenne et écart type des taux d'amélioration des temps de séjour moyens des pièces dans la cellule par les règles de coopération

3. Analyse des temps de transfert moyens des pièces dans la cellule d'assemblage :

a) Cas dont la capacité des FAEs est de 3 :

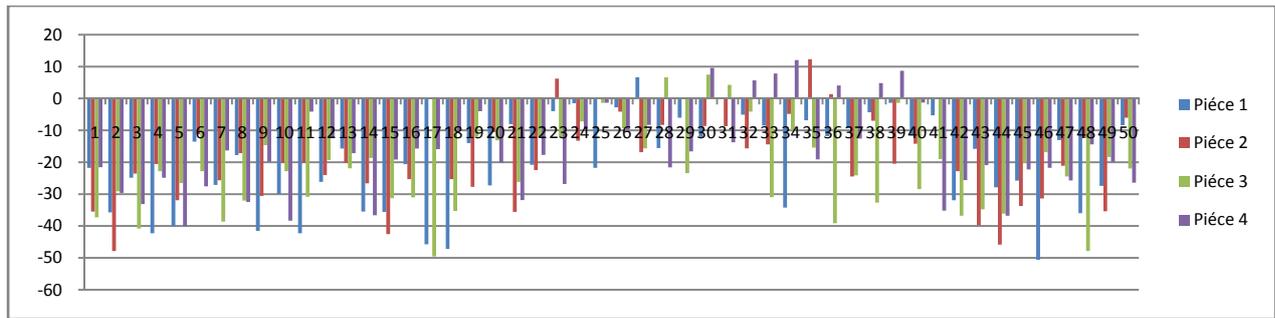


Figure IV.14 : Taux d'amélioration des temps de transfert moyens des pièces par la règle 1 dans les cas où la capacité des FAEs est de 3

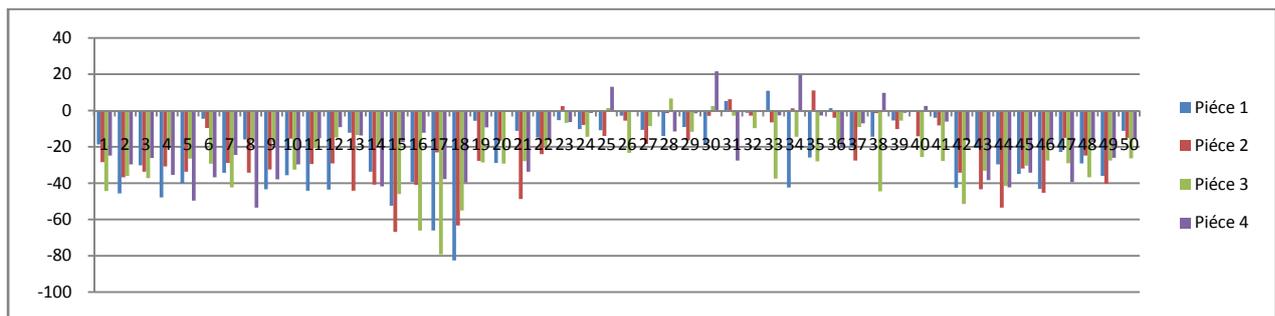


Figure IV.15 : Taux d'amélioration des temps de transfert moyens des pièces par la règle 2 dans les cas où la capacité des FAEs est de 3

Pour les 2 règles de coopération, on remarque que la distribution des taux d'amélioration est concentrée vers la partie inférieure de l'axe des abscisses. Le taux de dégradation atteint 50.58% (expérience 46) sous la règle 1 (Figure IV.14), alors que le taux d'amélioration maximal obtenu est 12.28% (expérience 35). D'autre part, sous la règle 2 (Figure IV.15), le taux de dégradation atteint 82.58%, alors que le taux d'amélioration maximal obtenu est 21.58%.

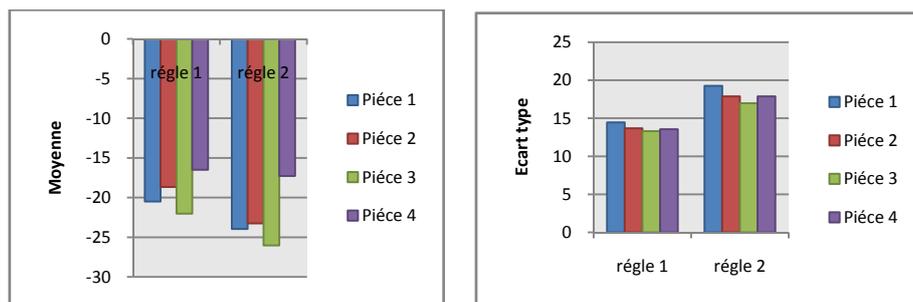


Figure IV.16 : Moyenne et écart type des taux d'amélioration des temps de transfert moyens des pièces dans la cellule par les règles de coopération dans les cas où la capacité des FAEs est de 3

Pour les 4 types de pièces (Figure IV.16), les moyennes des taux d'amélioration sont négatives sous les 2 règles de coopération, atteignant la valeur de $-26.04\% \pm 16.97$ pour les pièces du type « pièces3 » sous la règle 2.

b) Cas dont la capacité des FAEs est de 4 :

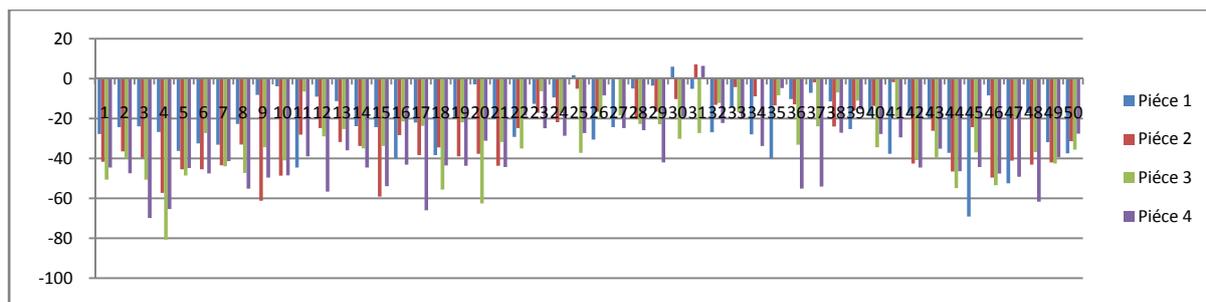


Figure IV.17 : Taux d'amélioration des temps de transfert moyens des pièces par la règle 1 dans les cas où la capacité des FAEs est de 4

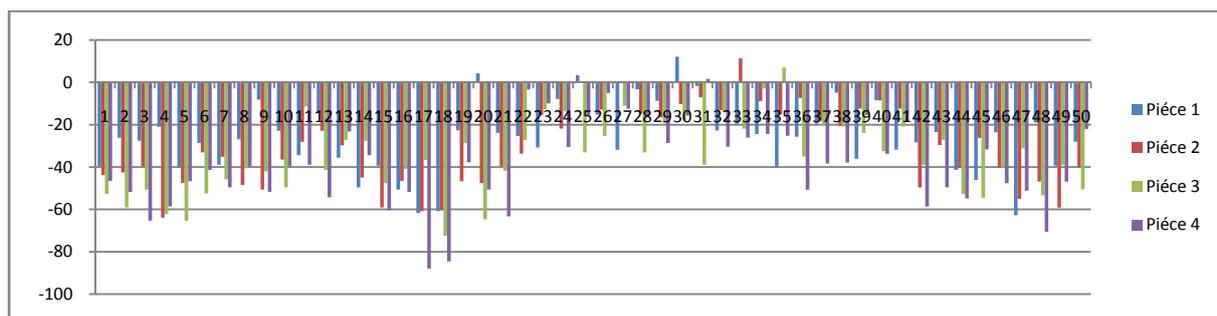


Figure IV.18 : Taux d'amélioration des temps de transfert moyens des pièces par la règle 2 dans les cas où la capacité des FAEs est de 4

De la même façon que pour le cas précédent, les taux d'amélioration restent concentrés vers la partie inférieure de l'axe (Figures IV.17, IV.18). Le taux de dégradation maximal atteint 80.68% (expérience 4) sous la règle 1 et 87.96% par l'application de la règle 2. Les cas d'amélioration sont rares, ne dépassant pas 12.05% pour la règle 2.

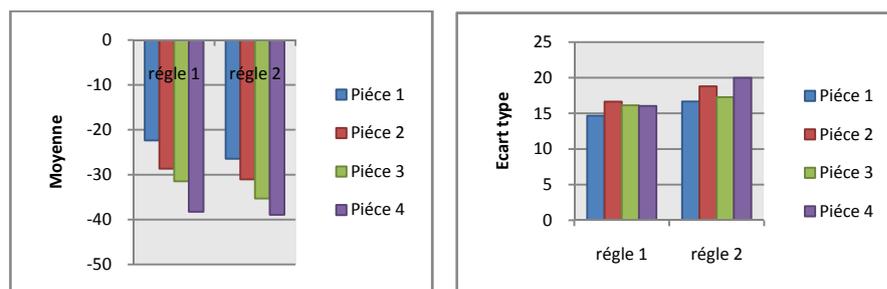


Figure IV.19 : Moyenne et écart type des taux d'amélioration des temps de transfert moyens des pièces dans la cellule par les règles de coopération dans les cas où la capacité des FAEs est de 4

Conformément à la figure IV.19, les moyennes des taux d'amélioration pour les 4 types de pièces sont négatives sous les 2 règles de coopération, atteignant la valeur $-38.95\% \pm 19.97$ pour les pièces du type « pièces4 ».

c) Cas dont la capacité des FAEs est de 5 :

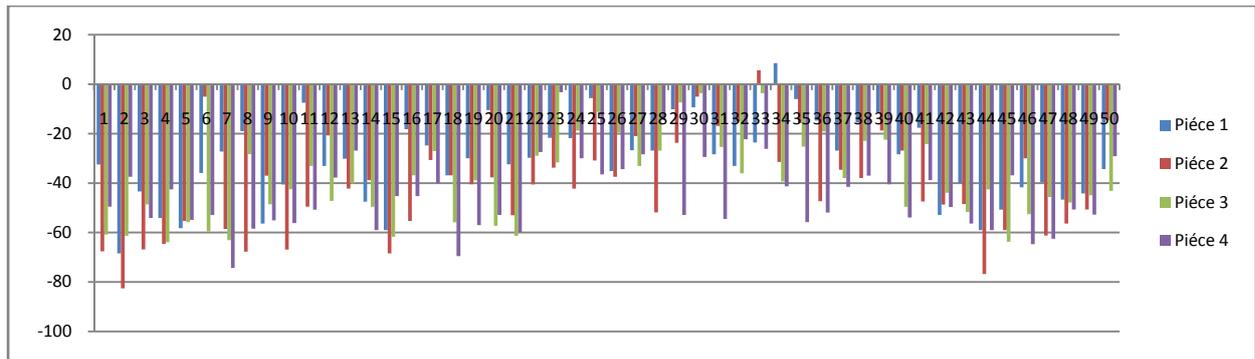


Figure IV.20 : Taux d'amélioration des temps de transfert moyens des pièces par la règle 1 dans les cas où la capacité des FAEs est de 5

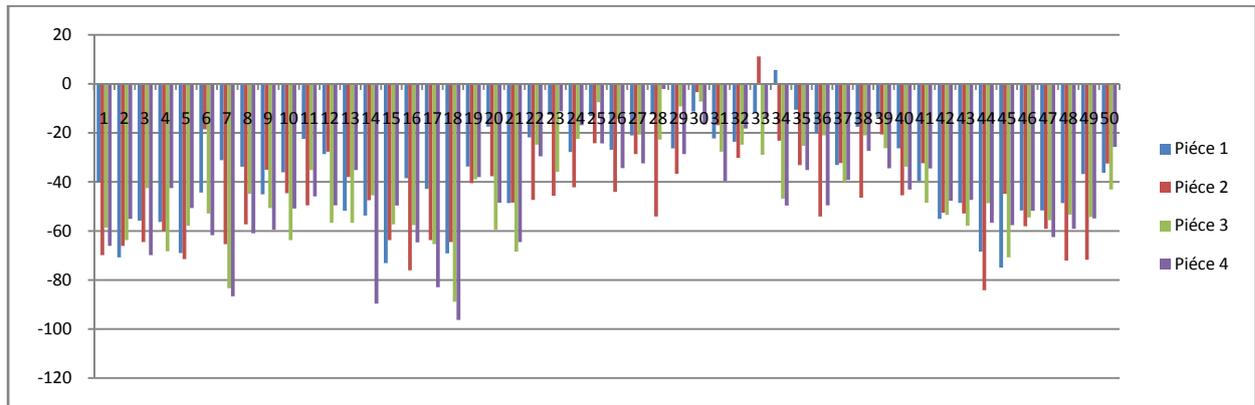


Figure IV.21 : Taux d'amélioration des temps de transfert moyens des pièces par la règle 2 dans les cas où la capacité des FAEs est de 5

Dans ce cas (figures IV.20, IV.21), le taux de dégradation atteint 82.58% (expérience 2) sous la règle1 et 96.30% sous la règle 2.

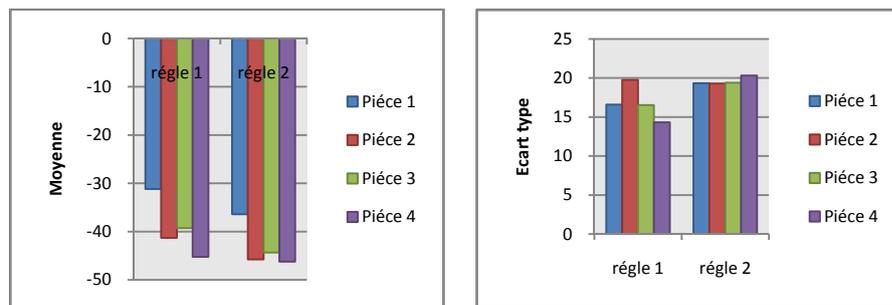


Figure IV.22 : Moyenne et écart type des taux d'amélioration des temps de transfert moyens des pièces dans la cellule par les règles de coopération dans les cas où la capacité des FAEs est de 5

Conformément à la figure IV.22, une forte dégradation des temps de transfert est obtenue pour les 2 règles de coopération, atteignant en moyenne $46.24\% \pm 20.31$ pour les pièces du type « pièces4 » sous la règle 2.

Les taux d'amélioration des temps de transfert sont tous négatifs pour les 2 règles de coopération appliquées. Le taux de dégradation le plus important est obtenu dans le cas où la capacité des FAEs est de 5 sous la règle 2 (96.30%). Nous avons constaté que la moyenne de dégradation devient plus importante avec l'augmentation des capacités FAEs

passant, pour la règle 1, de 22.03% capacité 3 à 45.23% dans le cas où la capacité des FAEs est de 5.

Cette détérioration du temps de transfert vient du fait que le déplacement de la pièce augmente sur le convoyeur pour atteindre la station la plus adaptée pour son traitement.

4. Analyse des temps d'attente moyens des pièces dans les files d'attente d'entrée :

a) Cas dont la capacité des files d'attente d'entrée est de 3 :

Conformément aux figures IV.23 et IV.24, les 2 règles de coopération réalisent une réduction importante du temps d'attente dans les FAEs qui se traduit par une forte concentration de la distribution des temps d'attente vers le haut de l'axe. Ces réductions sont plus importantes dans le cas de la FAE 1 et varient de 11% à 68% sous la règle 1 et de 8 à 82% sous la règle 2.

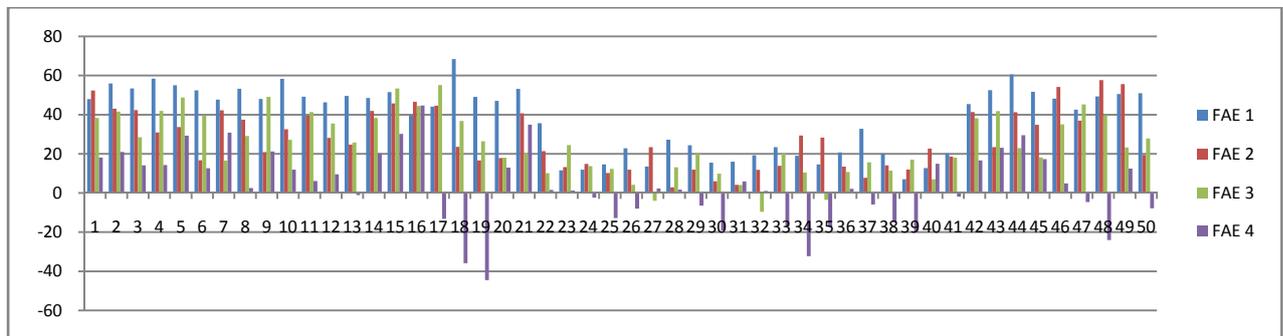


Figure IV.23 : Taux d'amélioration des temps d'attente moyens des pièces dans les FAEs dont la capacité est de 3 par la règle 1

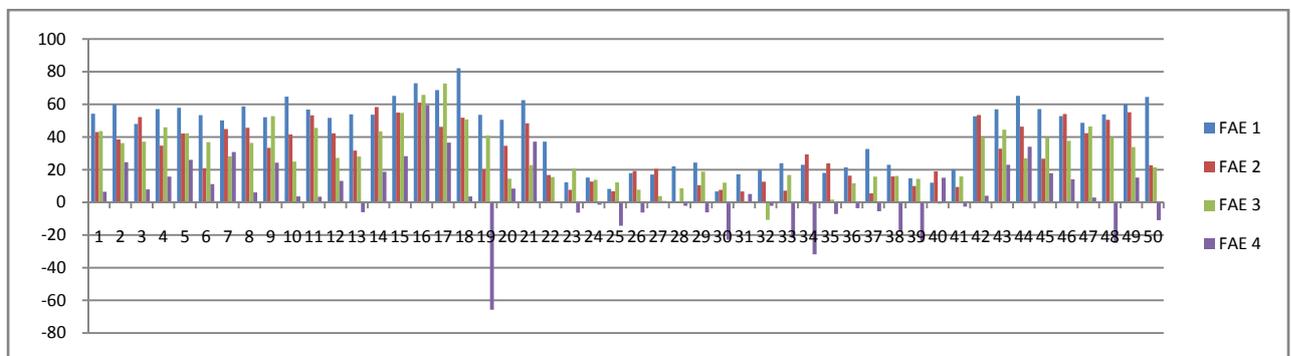


Figure IV.24 : Taux d'amélioration des temps d'attente moyens des pièces dans les FAEs dont la capacité est de 3 par la règle 2

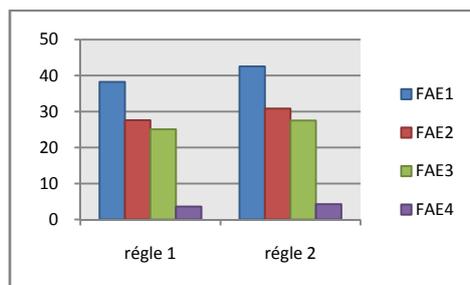


Figure IV.25: Taux d'amélioration des temps d'attente des pièces dans les FAEs dont la capacité est de 3

Conformément à la figure IV.25, le temps moyen d'attente est réduit sous les deux règles de pilotage dans l'ensemble des FAEs. Les taux d'amélioration moyens augmentent dans le sens décroissant de l'ordre de positionnement des FAEs pour atteindre une amélioration moyenne de 42% dans le cas de la FAE 1 sous la règle 2.

b) Cas dont la capacité des files d'attente d'entrée est de 4 :

Conformément aux figures IV.23 et IV.24, nous constatons une forte amélioration des temps d'attente dans les FAEs. Les meilleurs taux d'amélioration correspondent aux durées opératoires les plus faibles relativement aux temps de transfert et sont essentiellement obtenus dans le cas de la FAE 1. Ces réductions atteignent, au point 18, 71% pour la règle 1 et 84% pour la règle 2.

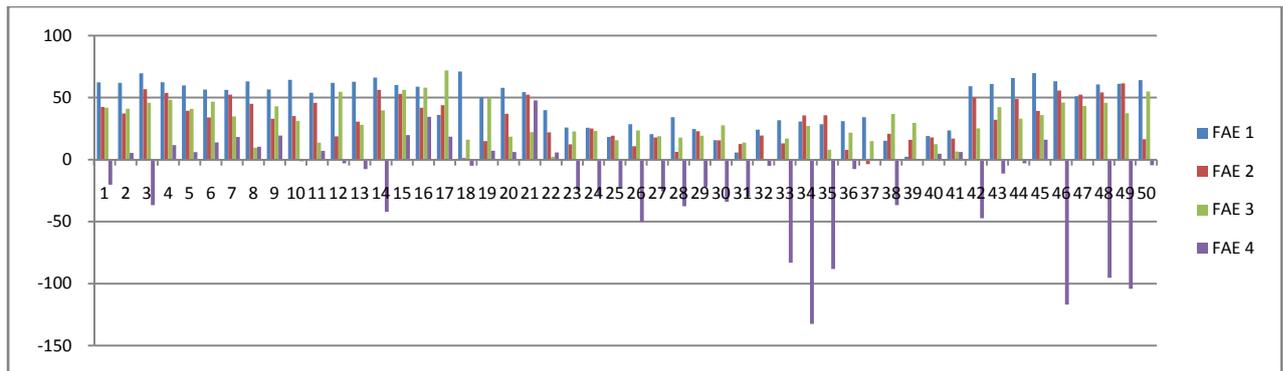


Figure IV.26 : Taux d'amélioration des temps d'attente moyens des pièces dans les FAEs dont la capacité est de 4 par la règle 1

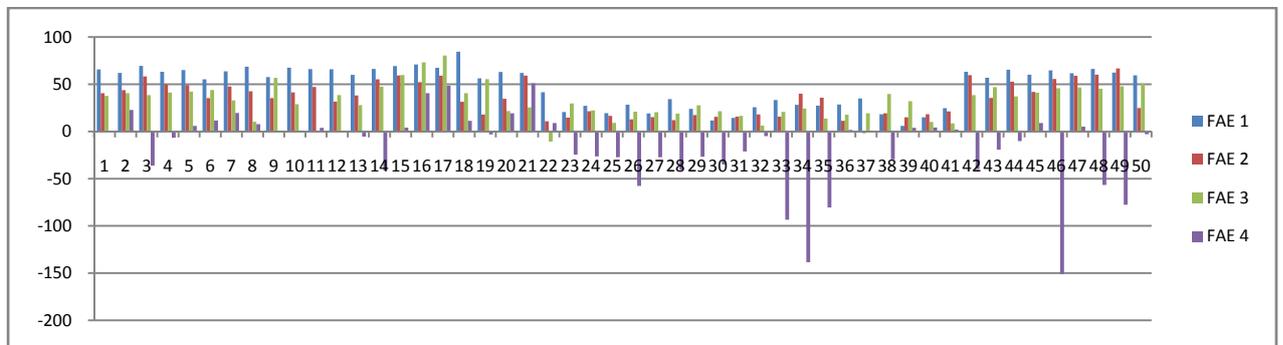


Figure IV.27: Taux d'amélioration des temps d'attente moyens des pièces dans les FAEs dont la capacité est de 4 par la règle 2

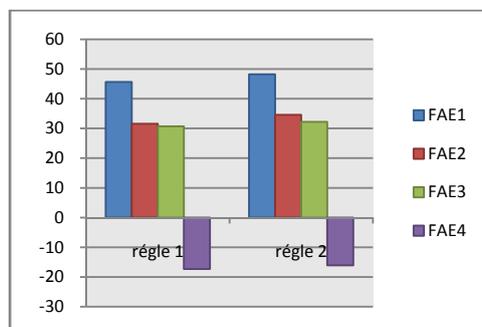


Figure IV.28: Moyenne des taux d'amélioration des temps d'attente des pièces dans les FAEs dont la capacité est de 4

Conformément à la figure IV.28, les règles de coopération réalisent une performance positive dans le cas des FAE 1, FAE 2 et FAE 3 avec un taux d'amélioration moyen de 45%, 31.5%, 30.7% respectivement sous la règle 1 et de 48%, 34%, 32% respectivement sous la règle 2. Par ailleurs, on enregistre une performance négative pour la FAE 4 de l'ordre de -16%.

c) Cas dont la capacité des FAEs est de 5 :

Conformément aux figures IV.23 et IV.24, Les meilleurs taux d'amélioration sont obtenus dans le cas de la FAE 1. Ces réductions atteignent, au point 18, 79% pour la règle 1 et 87% pour la règle 2. Cependant, on enregistre une forte dégradation des temps moyens d'attente dans la FAE 4 atteignant -621% au point 35.

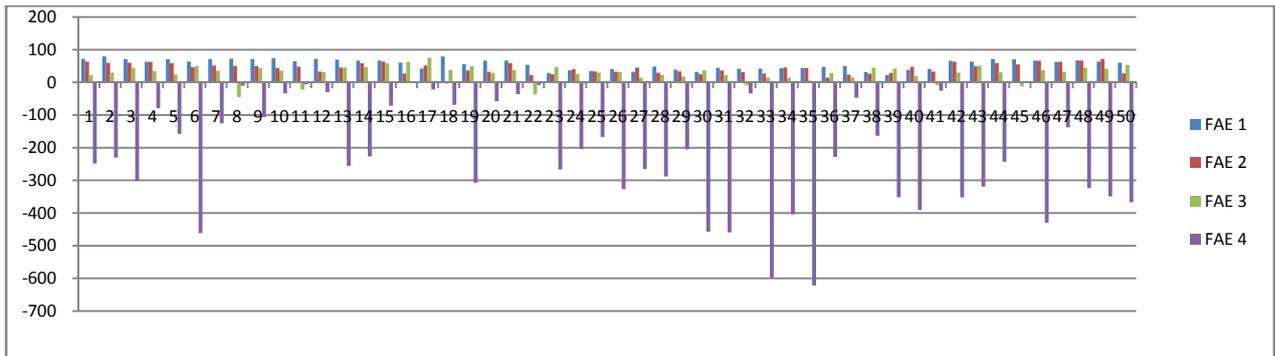


Figure IV.29 : Taux d'amélioration des temps d'attente moyens des pièces dans les FAEs dont la capacité est de 5 par la règle 1

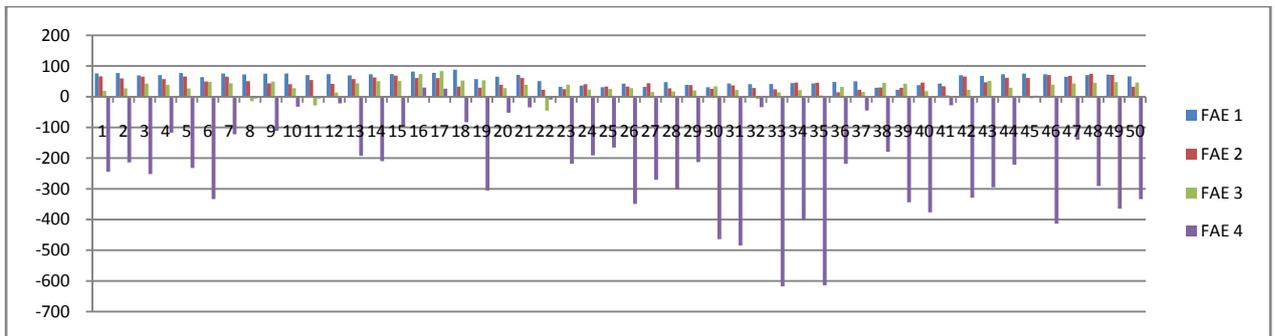


Figure IV.30 : Taux d'amélioration des temps d'attente moyens des pièces dans les FAEs dont la capacité est de 5 par la règle 2

Conformément à la figure IV.31, nous pouvons constater qu'une forte amélioration est apportée aux temps d'attente dans les FAE 1, FAE 2 et FAE 3 avec un taux d'amélioration moyen de l'ordre de 56%, 43% et 28% respectivement pour la règle 1 ainsi qu'une amélioration de 59%, 46% et 29% respectivement sous la règle 2, alors qu'on enregistre une dégradation supérieure à 200% dans le cas de la FAE 4.

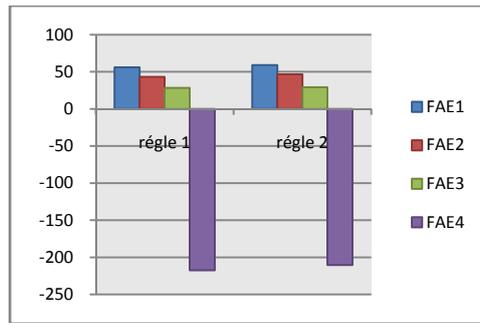


Figure IV.31 : Moyenne des taux d'amélioration des temps d'attente moyens des pièces dans les FAEs dont la capacité est de 5

La figure IV.32 présente une synthèse des temps moyens d'attente obtenus pour toutes les configurations testées sous les deux règles de coopération retenues. Dès lors, nous pouvons constater que sous les 2 règles de pilotage et pour toutes les configurations testées, les temps d'attente moyens dans les files d'attente sont croissants suivant l'ordre de positionnement des files d'attente. Ils sont plus importants dans la dernière file d'attente (FAE4) et moins importants dans la première (FAE1). Nous déduisons que la station 4 est un poste goulet sur le convoyeur.

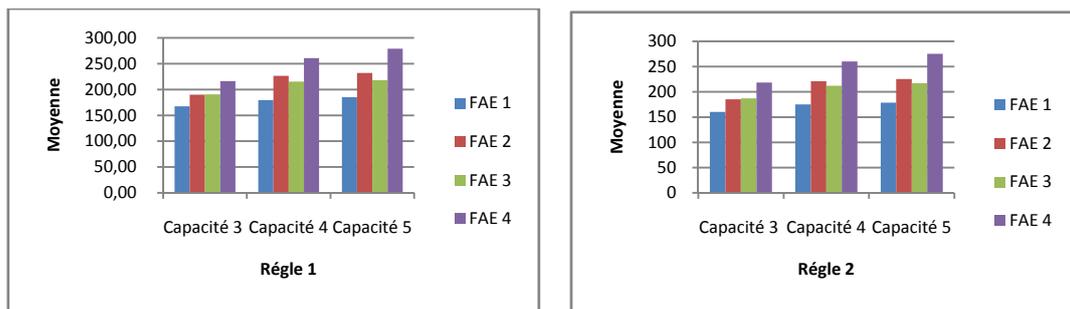


Figure IV.32: Moyenne des temps d'attente des pièces dans les FAEs

5. Analyse du temps d'attente moyen des pièces dans la cellule :

L'analyse du temps d'attente moyen des pièces dans les FAEs a révélé une forte amélioration qui décroît suivant l'ordre croissant de l'ordre de positionnement des stations pour atteindre une performance négative dans la FAE 4 pour les deux dernières configurations testées (capacité 4 et capacité 5). Notre objectif à travers l'analyse des temps d'attente moyens des pièces dans la cellule c'est-à-dire la somme des temps d'attente dans les différentes FAEs, est de mesurer l'effet de compensation entre les performances des différentes FAEs et son impact sur le temps global d'attente dans la cellule.

a) Taux d'amélioration du temps d'attente moyen des pièces dans la cellule d'assemblage par la règle 1 :

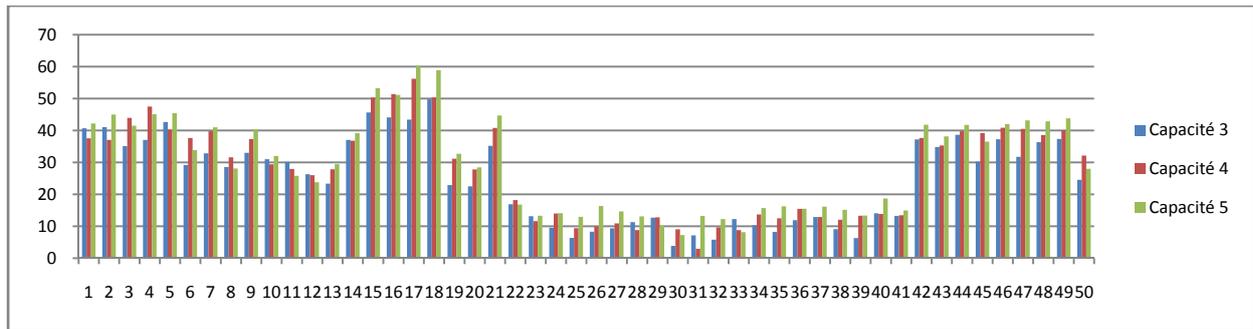


Figure IV.33 : Taux d'amélioration du temps d'attente moyen des pièces dans la cellule par la règle 1

b) Taux d'amélioration du temps d'attente moyen des pièces dans la cellule d'assemblage par la règle 2 :

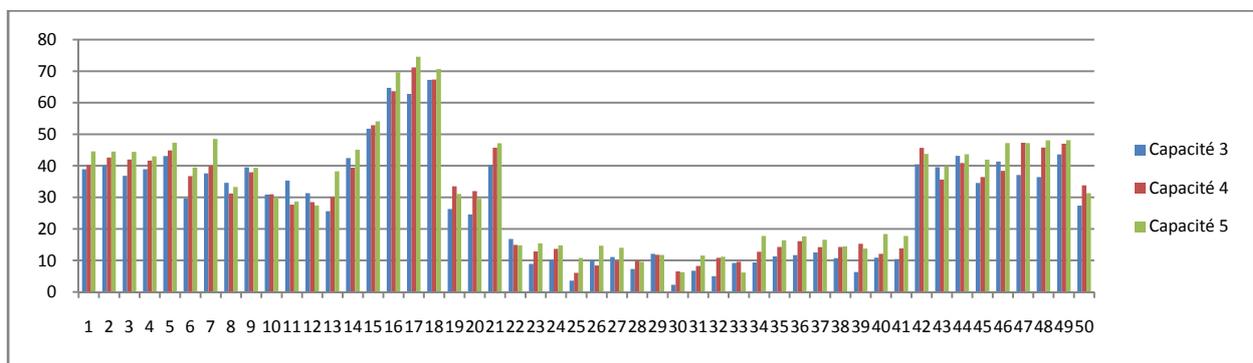


Figure IV.34 : Taux d'amélioration du temps d'attente moyen des pièces dans la cellule par la règle 2

On remarque conformément aux figures IV.33 et IV.34 que le temps d'attente moyen des pièces dans la cellule est réduit pour toutes les configurations testées malgré la dégradation du temps d'attente dans la FAE 4 qui atteint -17% pour les cas dont la capacité des FAEs est de 4 et - 200% pour les cas dont la capacité des FAEs est de 5. D'autre part, on remarque que les meilleurs taux d'amélioration sont enregistrés pour les cas dont la capacité des FAEs est de 5 et atteignent 60% et 74% sous les règle 1 et règle 2 respectivement.

6. Analyse du nombre moyens des pièces en attente dans les files d'attentes d'entrée:

a) Cas dont la capacité FAEs est de 3 :

Dans les cas dont la capacité des FAEs est de 3, le nombre moyen des pièces en attente est réduit dans l'ensemble des FAEs ce qui confirme les résultats obtenus dans la partie 4 (Figure IV.25). Ce nombre est réduit en moyenne de 38% pour la FAEs sous la règle 1 et de 42% pour la règle 2.

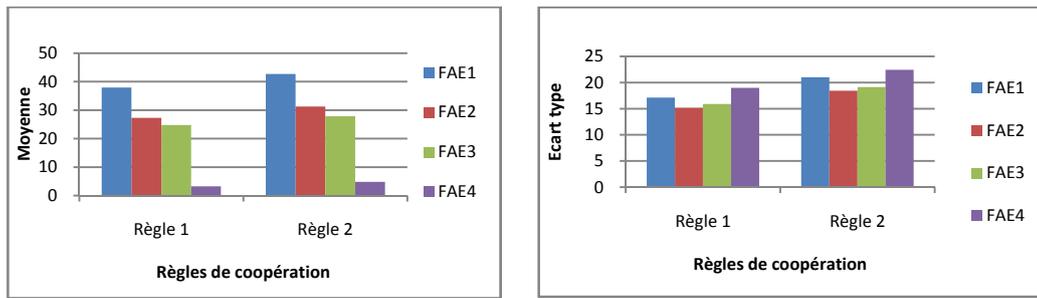


Figure IV.35 : Moyenne et écart type du nombre moyen des pièces en attente dans les FAEs dont la capacité est de 3 par les règles de coopération

b) Cas dont la capacité FAEs est de 4 :

Pour les cas dont la capacité des FAEs est de 4 (figure IV.36), on enregistre une performance négative dans la FAE 4 de la même façon que pour le temps d'attente moyen (figure IV.28). Le taux d'amélioration moyen maximal est obtenu par l'application de la règle 1 dans le cas de FAE 1 (de l'ordre de $52\% \pm 24$).

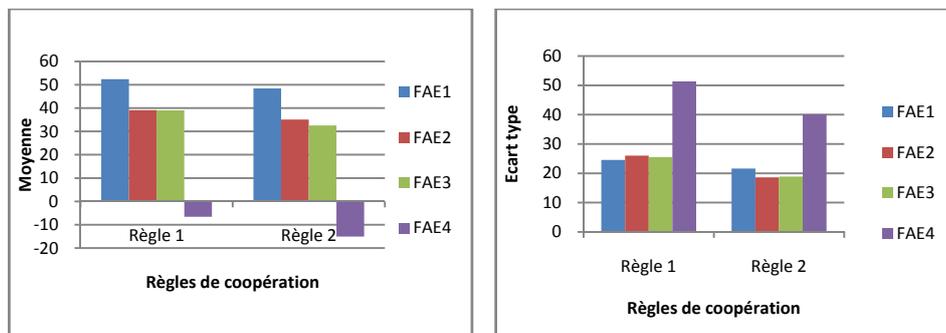


Figure IV.36 : Moyenne et écart type du nombre moyen des pièces en attente dans les FAEs dont la capacité est de 4 par les règles de coopération

c) Cas dont la capacité FAEs est de 5:

Pour les cas dont la capacité des FAEs est de 5 (Figure IV.37), on enregistre une performance négative dans la FAE 4 de la même façon que pour le temps d'attente moyen (Figure IV.31). Le taux d'amélioration moyen maximal est obtenu par l'application de la règle 1 dans le cas de FAE 1 (de l'ordre de $56\% \pm 24$).

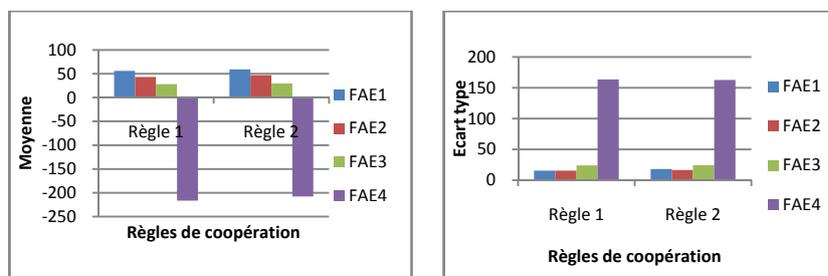


Figure IV.37 : Moyenne et écart type du nombre moyen des pièces en attente dans les FAEs dont la capacité est de 5 par les règles de coopération

A travers l'analyse du nombre moyen en attente dans les FAEs, nous avons constaté que les taux d'amélioration suivaient sensiblement la même évolution que le temps moyen d'attente.

7. Analyse des taux d'amélioration des robots :

a) Cas dont la capacité FAEs est de 3 :

Conformément à la figure IV.38, les taux d'amélioration des taux d'utilisation moyens des robots sont insignifiants sous la règle 1 ne dépassant pas $0.48\% \pm 5.72$. D'autre part, on enregistre une détérioration des taux d'utilisation sous la règle 2.

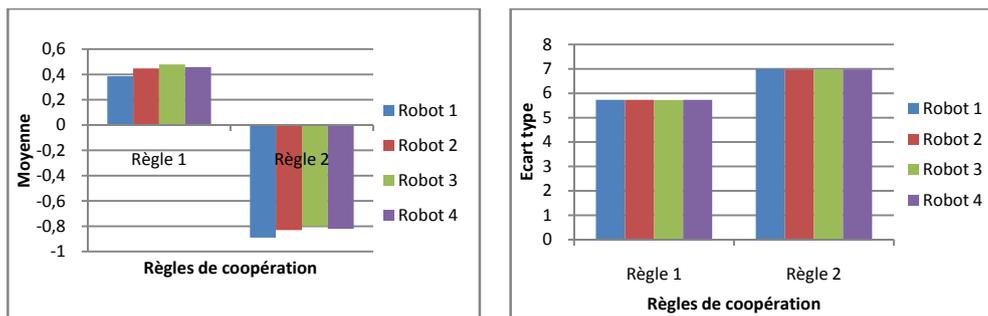


Figure IV.38 : Moyenne et écart type des taux d'amélioration des robots par les règles de coopération dans les cas où la capacité des FAEs est de 3

b) Cas dont la capacité FAEs est de 4 :

Pour les cas dont la capacité des FAEs est de 4 (Figure IV.39), aucune amélioration n'est apportée aux taux d'utilisation des robots par l'application d'un ordonnancement coopératif.

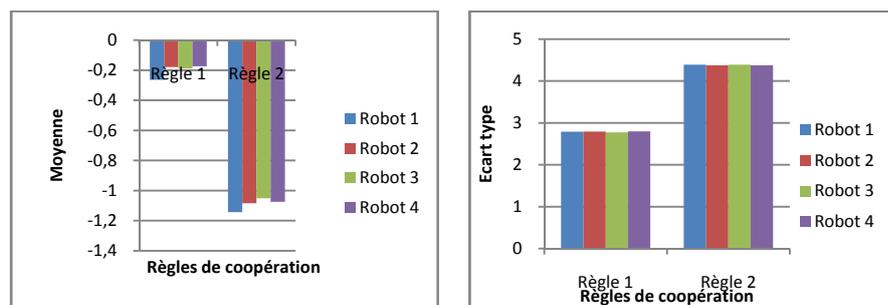


Figure IV.39 : Moyenne et écart type des taux d'amélioration des robots par les règles de coopération dans les cas où la capacité des FAEs est de 4

c) Cas dont la capacité FAEs est de 5 :

Dans cette configuration (Figure IV.40), aucune amélioration n'est apportée aux taux d'utilisation des robots par l'application d'un ordonnancement coopératif.

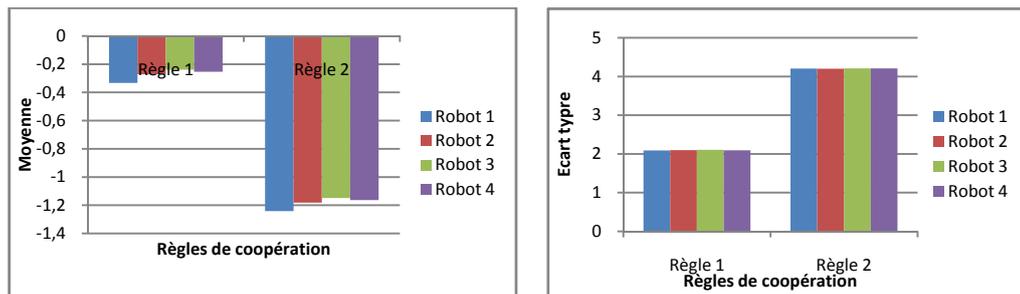


Figure IV.40 : Moyenne et écart type des taux d'amélioration des robots par les règles de coopération dans les cas où la capacité des FAEs est de 5

IV. Conclusion :

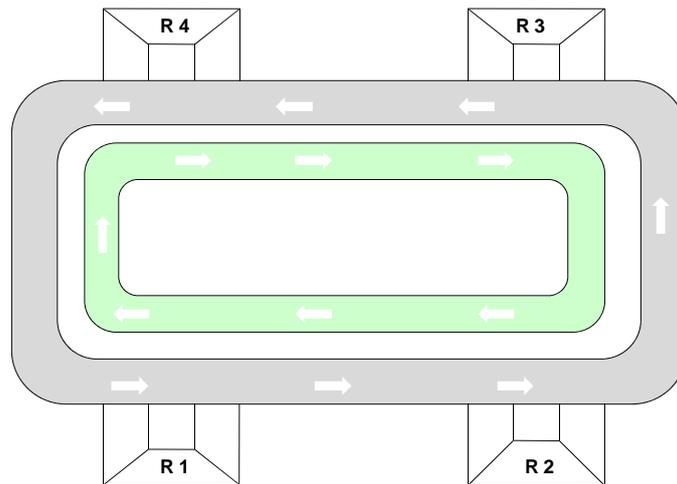
Après l'analyse des résultats obtenus à l'issue du processus de simulation, il en est ressorti les éléments suivants :

- Bien que le nombre des cas d'amélioration du temps de séjour d'une commande (C_{max}) soit supérieur au nombre de cas de dégradation (plus de 70% de cas d'amélioration dans le cas où la capacité des FAEs est de 4), les taux de dégradation restent plus importants, principalement pour les durées opératoires faibles.
- L'amélioration réalisée par l'application de la *règle 1* est plus importante que celle réalisée par la *règle 2*. Néanmoins, cette amélioration ne dépasse pas 3.14% pour le temps de séjour d'une commande et 7% pour le temps de séjour moyen d'une pièce.
- Pour les deux règles de coopération appliquées, on enregistre une forte diminution des temps d'attente dans les FAE1, FAE2 et FAE3 (amélioration moyenne de 59% pour FAE1 capacité 5) et du nombre de pièces dans la cellule (amélioration moyenne de 56%). Malgré une dégradation du temps d'attente moyen dans la FAE4, on obtient une amélioration importante du temps d'attente total des pièces dans la cellule (atteignant pour certains cas 75%).
- A l'opposé du temps d'attente, on constate une dégradation générale des temps de transfert des pièces sur le convoyeur atteignant en moyenne -45% pour les cas dont la capacité des FAEs est de 5.
- En général, nous avons constaté que les meilleures performances sont obtenues pour les configurations dont la capacité des FAEs est de 4 sous la *règle 1*.

A partir de ces résultats, nous pouvons proposer un certain nombre de solutions susceptibles d'améliorer la performance du système d'assemblage :

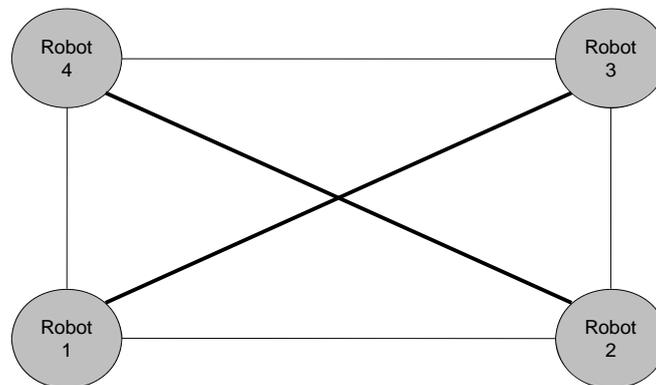
- Une première alternative serait de conférer aux différents objets à produire l'aptitude à prendre des rendez vous auprès des différentes ressources participant à leur production, la pièce sera dès lors en mesure de choisir la station la plus adaptée pour son traitement.

- Etant donné que le nombre moyen de pièces en attente au niveau de la station 4 est plus important comparé aux autres stations, les pièces sont donc contraintes d'attendre plus longtemps ou d'effectuer des tours supplémentaires sur le convoyeur, nous déduisons donc qu'il serait plus judicieux d'augmenter la capacité du robot 4 (poste goulet) qui limite la performance du système.
- Enfin, la dernière alternative envisagée serait de modifier la configuration du convoyeur pour minimiser le temps de transfert des pièces. Nous avons retenu deux configurations possibles :



Configuration (a) : Convoyeur à double sens

Dans cette configuration, la pièce serait autorisée à circuler dans deux sens opposés, cela passe par l'ajout d'un deuxième convoyeur rattaché aux stations.



Configuration (b) : interconnexions diagonales

Cette configuration consiste à ajouter des interconnexions diagonales avec plusieurs alternatives :

Relier : entrée 1-entrée 3, sortie 1-entrée 3, entrée 2-entrée 4, sortie 2-entrée 4, entrée 3-entrée 1, sortie 3-entrée 1, entrée 4-entrée 2, sortie 4-entrée 2.

Pour déterminer la plus avantageuse du point de vue du temps de transfert et du temps de séjour global, il sera indispensable de définir un plan expérimental contenant toutes les alternatives.

Conclusion et perspectives:

Les SFP open-shop peuvent constituer une opportunité d'amélioration des performances des plus intéressantes. Toutefois, cette amélioration ne peut être obtenue que via un pilotage efficace. La complexité croissante de ces systèmes de production associée à de nouveaux besoins en termes de flexibilité, de réactivité et de travail collaboratif ont conduit à délaisser le pilotage centralisé pour explorer de nouvelles formes de pilotage distribué. L'utilisation du concept d'agent est apparue comme un bon outil pour formaliser les connaissances, les expertises, les procédures, les interactions et les communications à mettre en œuvre pour résoudre le problème de répartition des décisions.

Dans le cadre de cette étude, nous avons conçu dans un premier temps, selon une approche orientée agent, un système d'ordonnancement distribué basé sur la coopération entre les ressources destinées à piloter en temps réel une cellule de production de type Open Shop composée de 04 stations disposées autour d'un convoyeur en boucle. La coopération entre les ressources est rendue possible par la définition d'un protocole de communication permettant de conserver une cohérence forte au niveau des objectifs individuels (locaux) de chaque agent par rapport à l'objectif global poursuivi de minimisation du temps de séjour des pièces dans la cellule.

Par la suite, nous avons simulé la stratégie d'ordonnancement proposée sous Arena 7.01, ce qui nous a permis de tester plusieurs règles de pilotage que nous avons retenues pour l'élaboration de notre système d'ordonnancement. Pour ce faire, nous avons conçu un plan d'expérimentations basé sur la capacité des files d'attente, des durées opératoires ainsi que les règles de pilotage pour un total de 450 expériences effectuées.

L'analyse des résultats obtenus à l'issue du processus de simulation a permis de mettre en évidence deux éléments fondamentaux :

- Une forte amélioration des temps d'attente moyens au niveau des FAEs et du temps d'attente des pièces dans la cellule pour toutes les configurations testées, ainsi qu'une diminution considérable du nombre moyen de pièces en attente tout en maintenant un niveau élevé d'utilisation des robots. Cette performance est réalisée grâce à une étroite coopération entre les ressources qui permet de définir en temps réel la station la plus adaptée pour le traitement des pièces.
- Une performance stable et régulière des temps de séjour moyens d'une commande et du temps de cycle d'une pièce et ce quelque soit les configurations testées (capacité des FAEs et les durées opératoires) à l'exception des cas dont les durées opératoires sont faibles comparés au temps de transfert sur le convoyeur.
- Une dégradation globale des temps de transfert qui s'explique par la nécessité pour la pièce, d'augmenter ses déplacements sur le convoyeur afin d'atteindre la station la plus adaptée.

Ces premiers résultats sont encourageants car ils démontrent la capacité de l'ordonnancement coopératif à assurer la régularité et la cohérence du comportement du système et à maîtriser le temps d'attente qui constitue une partie conséquente du temps de séjour pour toutes les configurations testées. Cependant, ces résultats s'ils sont prometteurs, nécessitent d'effectuer de nouvelles recherches pour réduire le temps de transfert qui constitue également une partie non négligeable du temps de séjour.

Le travail de recherche que nous avons effectué nous a permis d'approfondir nos connaissances dans deux domaines, d'une part l'ordonnancement d'atelier et d'autre part l'Intelligence Artificielle Distribuée au travers de la résolution distribuée et coopérative de problèmes et des systèmes multi agents SMA.

De plus, nous nous sommes initiés à la conduite de projets de simulation des systèmes de production, voire maîtriser le processus de simulation. En effet, la simulation est la méthode la plus adaptée à l'étude du comportement de systèmes complexes qui permet de considérer toutes les contraintes, sans imposition de restrictions, tel que c'est le cas lors de l'utilisation des méthodes analytiques.

Cette étude s'est voulue l'occasion de l'apprentissage du logiciel Arena Rockwel 7.01 très utilisé dans les milieux académiques et industriels. Nous avons eu recours, pour la construction de notre modèle, à tous les niveaux de modélisation proposés par Arena; allant d'une modélisation générale combinant les blocs et les éléments de Siman, les blocs plus élaborés d'Arena jusqu'au niveau le plus bas fournissant la possibilité d'intégrer des programmes en VBA (langage visuel basic).

En définitive, ce travail nous a permis de mieux cerner le fonctionnement de la cellule d'assemblage et d'en améliorer la compréhension pour nous et pour les autres) ce qui nous autorise à dégager un certain nombre de travaux en perspectives susceptibles d'améliorer la performance du système d'assemblage. Ces perspectives impliquent des configurations particulières à deux niveaux : Niveau organisationnel (protocole de communication) et niveau structurel (physique) :

Niveau organisationnel :

- Une première alternative serait de conférer aux différents objets à produire l'aptitude à prendre des rendez vous auprès des différentes ressources participant à leur production, la pièce sera dès lors en mesure de choisir la station la plus adaptée pour son traitement en évitant d'effectuer des déplacements inutiles sur le convoyeur. Ceci revient à considérer les pièces comme des agents à part entière pouvant interagir avec les ressources par le biais d'un protocole de communication adapté.

Niveau physique :

- Enfin, la dernière alternative envisagée serait de modifier la configuration du convoyeur pour minimiser le temps de transfert des pièces sur le convoyeur.

Par ailleurs, l'approche collaborative semble être une approche intéressante de pilotage des systèmes de production complexes. Ainsi, nous proposons qu'une étude relative à son application pour d'autres problèmes avec d'autres critères (contraintes de délais, minimisation de taches en retard,.. etc.) pourrait mettre en évidence l'apport de cette approche.

Ouvrage, thèses, mémoires et revues

- [AIT05] Ait Hssain A., 2005, « *Optimisation des flux de production* », 2ème Edition, L'usine nouvelle, DUNOD, Paris.
- [AZZ05] Azzaro-Pantel C. 2005, « *Ordonnancement et conception d'ateliers discontinus* », Edition technique de l'ingénieur, AG 3010.
- [BAK96] Bakalem M., 1996, « *Modélisation et simulation orientées objet des systèmes manufacturiers* », Thèse de Doctorat en Electronique-Electrotechnique-Automatique, Université de Savoie, France.
- [BAR03] Le Bars M., 2003, « *Un Simulateur Multi-Agent pour l'Aide à la Décision d'un Collectif : Application à la Gestion d'une Ressource Limitée Agro-environnementale* ». Thèse de doctorat, Université Paris IX-DAUPHINE, France.
- [BER91] Berthe Y., 1991, « *Ordonnancement: Stratégies, Débats et Intelligence Artificielle* », No. 24, pp 38-47, Suisse.
- [BER00] Berchet C., 2000, « *Modélisation pour la simulation d'un système d'aide au pilotage industriel* », Thèse de doctorat en génie industriel, Institut National Polytechnique de Grenoble.
- [BER07] Berruet P., Petin J.P., Rigaud F., Toguyeni A., Zemai E., 2007, « *Architectures de pilotage de procédés industriels* », Edition technique de l'ingénieur, AG 3510.
- [BES95] BESLON G., 1995, « *approche pour le pilotage réactif en atelier flexible* », Thèse de doctorat : Ingénierie Informatique, l'institut national des sciences appliquées de Lyon, France.
- [BOU07] Boutin F., 2007, « *Dialogue intégrant les émotions et la personnalité* », Rapport de stage : Spécialité Intelligence Artificielle et Décisions, Université Pierre et Marie Curie, France.
- [CAU06] Caumon A., 2006, « *le problème de jobshop avec contraintes : modélisation et optimisation* », Thèse de doctorat, Université Blaise Pascal, France.
- [CHA03] Chahed S., 2003, Projet bibliographique : « *La simulation en génie industriel* », Laboratoire Génie Industriel, Ecole Central Paris, France.
- [COH89] Cohen V. 1989, « *Introduction aux plans expérimentaux* », Revue de statistique appliquée, tome 37, Volume n°2.
- [COT99] Cottet F., Delacroix J., Kaiser C., Mammeri Z., 1999, « *Ordonnement temps réel : Ordonnancement centralisé* », Edition technique de l'ingénieur, S 8055.
- [DJE07] Djellouli S., 2007, « *Approche par coopération inter machine pour un*

- ordonnancement distribué* », mémoire de magister, Ecole nationale polytechnique, Alger.
- [DRA98] Draghici G., Brinzei N., Filipas I., 1998, « *La modélisation et la simulation en vue de la conduite des systèmes de production* », Les Cahiers des Enseignements Francophones en Roumanie, universitatea politehnica Din Timisoara.
- [ESQ99] Esquirol P., Lopez P., 1999, « *L'ordonnancement* », édition Economica, Paris.
- [FIN93] Finke G., Dupont L., 1993 IEEE Systems, « *Combinatorics of Cyclic Assembly Systems* », Man and Cybernetics Conference Proceedings; Grenoble, France.
- [FER95] Ferber J., 1995, « *Les systèmes multi agents : Vers une intelligence collective* », Inter Edition, France.
- [GAN03] Gançarski P., 2003, « *Cours intelligence artificielle distribuée systèmes multi-agents* », université Louis Pasteur, Strasbourg, premier semestre universitaire.
- [GHA06] Ghazi N., 2006, « *Problème d'ordonnancement Open Shop dans un environnement d'atelier d'assemblage cyclique* », mémoire de magister, Spécialité : automatique, département Génie électrique, Laboratoire de Commande des Processus, ENP, Alger.
- [GIA07] Giard V., Balin S., 2007, « *Problèmes méthodologiques posés par la simulation de processus de production de services* », Université Paris-Dauphine, RS – JESA, France.
- [GOU03] Gourgand M., Grangeon N. et S. Norre, 2003, « *Problèmes d'ordonnancement dans les systèmes de production de type Flow-Shop Hybride en contexte déterministe* », Journal sur l'enseignement des sciences et technologies de l'information et des systèmes, Volume 2, Université Blaise Pascal (Clermont-Ferrand II).
- [HAB01] Habchi G., 2001, « *Conceptualisation et modélisation pour la simulation des systèmes de production* », Document de Synthèse, Université de Savoie, France.
- [JUB94] Jubin M., 1994, « *Ateliers flexibles d'usinage* », Edition Technique de l'Ingénieur, B 7030.
- [KEL03] Kelton W. D., 2003, « *Simulation with Arena* », 3ème Edition, International Edition, New York.
- [KIE02] Kieffer J., Pujo P., 2002, « *Fondements du pilotage des systèmes de production* », Edition Lavoisier, Paris.
- [LAB06] Labarthe O., 2006, « *Modélisation et simulation orientées agents de chaînes logistiques dans un contexte de personnalisation de masse : modèles et cadre méthodologique* », Thèse de doctorat, Université Laval, Canada.

- [MAN02] Mandiau R., Strugeon E. G., 2002, « *Systèmes multi agents* », Edition Technique de l'Ingénieur, S 7216.
- [SAN05] Sansonnet J. P., 2005, « *présentation : Introduction aux formalismes des Langages de Communication Agent* », LIMSI-CNRS.
- [TRA01] Tranvouez E., 2001, « *IAD et ordonnancement : une approche coopérative du réordonnancement* », Thèse de doctorat, Université d'Aix-Marseille III, France.
- [TRE02] Trentesaux D., Sénéchal O., 2002, « *Conduite des systèmes de production manufacturière* », Edition technique de l'ingénieur, S 7598.

Webographie

- [WEB1] http://geronim.free.fr/ecoent/cours/la_notion_de_fonction_de_production.htm: La notion de fonction de production.
- [WEB2] <http://turing.cs.pub.ro>: Agents intelligents cours WAB interactif.
- [WEB3] www.fipa.org: foundation for intelligent physical agents, FIPA Communicative Act Library Specification.
- [WEB4] http://fr.wikipedia.org/wiki/Atelier_flexible : Atelier flexible

Listes des annexes :

Annexe 1 : Langage KQML

Annexe 2 : Langage ACL

Annexe 3 : Approches de modélisation pour la simulation

Annexe 4 : Etapes du processus de simulation

Annexe 5 : Programme Visual Basic

Annexe 6 : Plan expérimental

Annexe 7 : Glossaire

Le langage KQML :

Pour pouvoir communiquer, tous les agents considèrent que leurs partenaires de dialogue possèdent une représentation de leurs connaissances. Les connaissances communiquées par chaque agent aux autres agents sont vues en KQML comme une base virtuelle de connaissances BC (*Virtual Knowledge Base*). Dans ce qui suit, nous allons utiliser les abréviations suivantes : A : l'agent émetteur, B : l'agent récepteur, **p** : **le contenu du message** BC : la base virtuelle de connaissances.

Liste des performatifs KQML : [FER95]

Performatif	Description
<i>Achieve</i>	Veut que Y accomplisse une action
<i>Advertise</i>	A indique qu'il peut faire P
<i>Ask-about</i>	A veut toutes les expressions sur p de la BC de B
<i>Ask-all</i>	A veut toutes les réponses de B à une question
<i>Ask-one</i>	A veut une réponse de B à une question
<i>Break</i>	A veut supprimer une communication
<i>Broadcast</i>	A veut que B envoie un performatif à toutes ses accointances
<i>Broker-all</i>	A veut que B récupère toutes les réponses à un performatif
<i>Broker-one</i>	A veut de l'aide pour récupérer la réponse à un performatif
<i>Deny</i>	A indique à B que le performatif ne s'applique plus à B
<i>Delete</i>	A veut que B supprime une expression de sa BC
<i>Delete-all</i>	A veut que B supprime toutes les p correspondantes de sa BC
<i>Delete-one</i>	A veut que B supprime une expression p de sa BC
<i>Discard</i>	A ne veut pas les réponses suivantes de B
<i>Eos</i>	Fin d'une suite de réponses à une requête préalable
<i>Error</i>	A considère que le message de B est mal formé
<i>Evaluate</i>	A veut que B simplifie l'expression associée
<i>Forward</i>	A veut que B délègue un performatif
<i>Generator</i>	Identique à « Standby » pour un « Stream-all »
<i>Insert</i>	A veut que B insère une expression dans sa BC
<i>Monitor</i>	A veut que B mette à jour ses réponses à un « Stream-all »
<i>Next</i>	A veut la réponse suivante à une requête préalable
<i>Pipe</i>	A veut que B redirige tous les performatifs suivants à un agent
<i>Ready</i>	A est prêt à répondre à un performatif préalable de B
<i>Recommend-all</i>	A veut tous les agents qui répondent à un performatif donné
<i>Recommend-one</i>	A veut un agent qui réponde à un performatif donné
<i>Recuit-all</i>	A veut que tous les agents capables répondent à un performatif
<i>Recuit-one</i>	A un agent qui répond à un performatif
<i>Register</i>	A indique à B qu'il peut accomplir une action
<i>Reply</i>	A répond à une demande attendue
<i>Rest</i>	A veut toutes les réponses suivantes
<i>Sorry</i>	A ne veut plus fournir une réponse plus informative
<i>Standby</i>	A veut que B soit prêt à répondre à un performatif
<i>Stream-about</i>	Version à réponses multiples de « Ask-about »
<i>Stream-all</i>	Version à réponses multiples de « Ask-all »
<i>Suscripbe</i>	A veut que B mette à jour ses réponses à un performatif
<i>Tell</i>	Indique qu'une expression fait partie de la Bc de A
<i>Transport-adress</i>	A associe un nom symbolique à une adresse transport
<i>Unregister</i>	Un « Deny » associé à un performatif « Register »
<i>Untell</i>	Indique qu'une expression ne fait pas partie de la BC de A

Langage ACL :

Au cours des années 90, KQML est devenu un standard de fait dans les SMA industriels. De nombreuses applications à base d'agents cognitifs ont pu être déployées sur la base de KQML mais les chercheurs ont fait plusieurs critiques sévères : [SAN05]

- La sémantique des performatifs et des attributs des messages de KQML n'a pas été définie de manière rigoureuse ce qui a conduit à des interprétations divergentes.
- Le langage ne possède pas de performatifs concernant les « commissifs » ou « promissifs » (promettre de faire quelque chose pour un autre agent et gérer de telles promesses) ce qui fut critiqué par Cohen et Levesque.
- L'ensemble des performatifs est vaste et surtout redondant (40 en tout).

Sur cette base, en 1999, la FIPA (Foundation for Intelligent Physical Agent) a développé un nouvel langage ACL qui est sensé mieux répondre à ces critiques :

- Extérieurement, ACL-FIPA ressemble beaucoup à KQML.
- Il possède 22 performatifs seulement.
- Il possède une sémantique bien définie

Description détaillée des performatifs ACL : [WEB03]

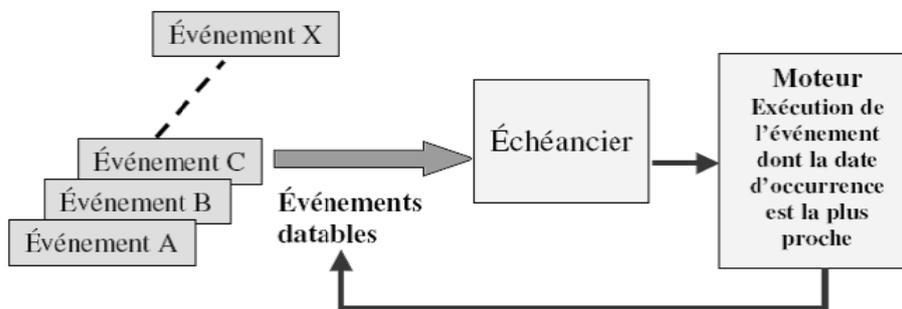
Performatif	Description
<i>accept-proposal</i>	The action of accepting a previously submitted proposal to perform an action.
<i>agree</i>	The action of agreeing to perform some action, possibly in the future.
<i>cancel</i>	The action of cancelling some previously <i>request'ed</i> action which has temporal extent (i.e. is not instantaneous).
<i>cfp</i>	The action of calling for proposals to perform a given action.
<i>confirm</i>	The sender informs the receiver that a given proposition is true, where the receiver is known to be uncertain about the proposition.
<i>disconfirm</i>	The sender informs the receiver that a given proposition is false, where the receiver is known to believe, or believe it likely that, the proposition is true.
<i>failure</i>	The action of telling another agent that an action was attempted but the attempt failed.
<i>inform</i>	The sender informs the receiver that a given proposition is true.
<i>inform-if</i>	A macro action for the agent of the action to inform the recipient whether or not a proposition is true.
<i>inform-ref</i>	A macro action for sender to inform the receiver that an object corresponds to a definite descriptor (e.g. a name).
<i>not-understood</i>	The sender of the act (e.g. <i>i</i>) informs the receiver (e.g. <i>j</i>) that it perceived that <i>j</i> performed some action, but that <i>i</i> did not understand what <i>j</i> just did.
<i>propose</i>	The action of submitting a proposal to perform a certain action, given certain preconditions.
<i>query-if</i>	The action of asking another agent whether or not a given proposition is true.
<i>query-ref</i>	The action of asking another agent for the object referred to by an expression.
<i>refuse</i>	The action of refusing to perform a given action, and explaining the reason for the refusal.

<i>reject-proposal</i>	The action of rejecting a proposal to perform some action during a negotiation.
<i>request</i>	The sender requests the receiver to perform some action. One important class of uses of the request act is to request the receiver to perform another communicative act.
<i>request-when</i>	The sender wants the receiver to perform some action when some given proposition becomes true.
<i>request-whenever</i>	The sender wants the receiver to perform some action as soon as some proposition becomes true and thereafter each time the proposition becomes true again.
<i>request-whomever</i>	The sender wants an action performed by some agent other than itself. The receiving agent should either perform the action or pass it on to some other agent.
<i>subscribe</i>	The act of requesting a persistent intention to notify the sender of the value of a reference, and to notify again whenever the object identified by the reference changes.

Les différentes approches de modélisation pour la simulation : [HAB01]

La dynamique du modèle est obtenue grâce à la logique de changement d'état qui traduit temporellement et spatialement le comportement des objets, c'est à dire l'évolution des objets du modèle dans le temps et dans l'espace. La simulation pilote le comportement du modèle en examinant et en mettant à jour le modèle (à travers ses variables et ses attributs) quand il est connu qu'un changement dans l'état du système doit arriver. De ce fait, la simulation comporte : des changements instantanés à des points différents du temps et de l'espace, une série d'événements, une série d'activités et des processus. De ce fait il existe traditionnellement trois logiques de changement d'état : la logique orientée événements, la logique orientée activités et la logique orientée processus. A ces trois logiques, nous pouvons rajouter l'approche orientée objet utilisée pour décrire les objets physiques à travers des objets informatiques.

Approche orientée événements : La logique orientée événements discrets consiste à répertorier les différents types d'événements (généralement inconditionnels) et à décrire la procédure de changement d'état correspondante au déclenchement de chaque type d'événement. Aucune modification n'est censée avoir lieu entre l'occurrence de deux événements successifs. Les événements sont ordonnés dans un échéancier selon l'ordre chronologique de leur date d'occurrence.



Simulation selon l'approche par événements

La simulation évolue dans le temps en se déplaçant d'un événement à un autre en exécutant la logique concernée. Parmi les langages de simulation orienté événement nous pouvons citer : *SIMSCRIPT* et *GASP IV*. Cette logique convient bien aux systèmes dont les composants sont relativement indépendants, c'est à dire qu'il existe peu de conditions à tester pour déterminer si un événement doit avoir lieu.

Approche orientée activité : C'est une logique duale de la précédente. Elle consiste à recenser les différents types d'activités et à décrire les procédures de changement d'état correspondant aux conditions de début et de fin de chaque type d'activité. Pendant la simulation, les procédures décrivant les conditions de début et de fin sont vérifiées par intervalle de temps fixe ou variable. Si cette technique est plus simple que la précédente, elle

s'avère moins efficace, car à un instant donné, une procédure peut être testée sans que les conditions de début ou de fin de l'activité ne soient vérifiées.

Approche orientée processus : C'est une logique largement utilisée dans les langages de simulation. Elle cadre naturellement les systèmes manufacturiers qui comprennent plusieurs parties devant interagir, qui peuvent être décomposés en objets passifs et actifs (entités et ressources), ou qui ont souvent des processus identiques. Elle consiste à décrire le fonctionnement d'un système comme une interaction de plusieurs processus, à travers des primitives, des blocs ou des fonctions, conçus au préalable. Chaque type de primitive (bloc ou fonction) décrit un processus élémentaire et générique du système à modéliser. Souvent, on considère dans ce type de logique (modélisation) la notion de file d'attente qui introduit certains problèmes dans la modélisation des systèmes sans stocks.

Quand un processus (de type entité) ne dispose pas de ressources suffisantes, il se met en attente dans une file jusqu'à ce qu'un autre processus le réveille. Cette approche combine la simplicité de description de l'approche orientée activité et l'efficacité de l'approche orientée événement. Parmi les langages utilisant cette approche, citons SLAM et ARENA.

Approche orientée objet : L'approche orientée objets est de plus en plus répandue dans le développement des systèmes. Le système est décrit par un ensemble d'objets qui communiquent par envoi de messages. La modélisation orientée objets se fait en deux étapes : la première concerne la définition de la structure du système par la spécification des objets (attributs) et la seconde concerne la définition du comportement dynamique du système en précisant les relations entre objets ainsi que leur fonctionnement propre (méthodes). L'un des avantages de l'approche orientée objets est la combinaison des avantages des langages généraux (flexibilité et champ d'application larges) avec les avantages des outils spécialisés (objets paramétrables et construction simple des modèles). Parmi les langages orientés objets, citons SIMULA, CLOS, MODSIM.

Les étapes du processus de simulation :

La modélisation est une tâche qui fait appel à des compétences scientifiques et informatiques, mais c'est aussi un art. Délimiter le système à étudier et trouver un modèle adéquat, sont deux tâches difficiles et dont le résultat dépend des compétences et de l'expérience de l'analyste concepteur du modèle. En particulier, un modèle n'est pas bon dans l'absolu, mais relativement aux objectifs de l'étude [AIT05]. En effet, le degré d'abstraction, la nature et le nombre des entités à inclure dans les modèles dépendent, entre autre, de ces objectifs.

Enfin, du point de vue du management, une étude de simulation est un projet dont le succès nécessite une gestion (affectation des ressources, planification) comme tout autre projet [AIT05].

Toutes ces considérations impliquent une décomposition du projet de simulation en plusieurs phases. En effet, passer du problème à la simulation suppose de franchir un certain nombre d'étapes. Pour Pritsker [GHA06], celles-ci sont au nombre de dix :

1. Analyse du système à étudier et formulation du problème

Le but est de définir les objectifs réels de l'étude (problèmes, réponses attendues), par opposition aux objectifs préalablement exprimés par le demandeur qui peuvent être des symptômes, parfois vagues et contradictoires.

Elle implique de :

- Bien comprendre le système de production cible,
- fixer les objectifs de l'étude,
- Déterminer les facteurs à faire varier et les critères de performance pertinents à observer et à optimiser,
- S'assurer de l'engagement des managers et/ou des décideurs,
- Planifier les ressources (personnel, matériel, logiciel),
- Spécifier les frontières du modèle et le niveau de détail,
- Déterminer les données à collecter et éventuellement à analyser,
- Rédiger le cahier des charges.

2. Modélisation conceptuelle :

C'est une description du fonctionnement essentiel du système dans un formalisme graphique (organigramme, diagramme, schéma,...). Le résultat de modélisation dépend du concepteur. Des hypothèses sont souvent émises. Il faut chercher un compromis entre la simplicité et l'exactitude du modèle.

Il est très important de valider le modèle conceptuel avant d'entamer la phase de programmation informatique. En effet, une présentation pas à pas du modèle conceptuel avant programmation permettra de valider la cohérence des informations recueillies localement et/ou des hypothèses émises.

- 3. Collecte des données nécessaires à la simulation :** On distingue (03) catégories de données : [Ber00]
 - **les données physiques :** il faut cerner l'organisation du système, avec le nombre de machines, d'opérateurs, de chariots, les temps d'ouverture, le partage des ressources, le nombre de produits, leur type, leur gamme,...;
 - **les données de gestion :** le carnet de commandes, les quantités commandées, les tailles des séries, les dates, les types de flux (poussé, tiré, tendu, kanban, ...), ... ;
 - **les données de pilotage :** les règles d'ordonnancement et de priorité, les aléas de fonctionnement (pannes, absentéisme, déréglage, règles de facteurs maîtrisables, ...)

4. Transcription informatique du modèle :

Il s'agit de traduire le modèle conceptuel dans un environnement informatique permettant l'expérimentation par ordinateur [AIT05]. Souvent, la modélisation conceptuelle est négligée et on passe rapidement à la saisie du programme grâce aux éditeurs graphiques offerts par les langages de simulation. Ceux-ci offrent la possibilité de réduire la distance entre le modèle conceptuel et le programme. Le passage du système graphique au langage est maintenant souvent automatisé (c'est le cas par exemple pour les langages SLAM et SIMAN). Le programme de simulation est souvent désigné par « modèle » ou encore « modèle/programme ».

5. Vérification du modèle :

Cette phase consiste à s'assurer que le modèle conceptuel est traduit correctement en un programme informatique : pas d'erreurs involontaires de logique et/ou de saisie des données.

6. Validation du modèle :

La validation du modèle, c'est avoir « LE bon » modèle, c'est-à-dire reflétant au mieux la réalité. Cette phase doit permettre de s'assurer de la cohérence entre le comportement du modèle de simulation et le système réel qu'il est censé représenter. Si ce système existe déjà, cette phase pourra se faire en comparant une trace du comportement du système réel avec celui du modèle de simulation. Lorsque, et c'est souvent le cas, le système étudié n'existe pas encore, cette validation est beaucoup plus délicate.

7. Définition des expériences à effectuer

On définit les conditions d'utilisation du modèle, la stratégie (le nombre d'expériences, la durée de simulation, plans d'expérience, ...) et la tactique (conditions initiales, types de résultats, ...).

8. Simulations et enregistrement des résultats

Après chaque simulation, les résultats sont enregistrés avant leur analyse.

9. Evaluation des performances, interprétation des résultats, et recommandations

L'analyse des résultats est possible à l'aide des indicateurs de performance sous forme de tableaux, courbes, graphiques, histogrammes, ...etc. Le recours à des outils d'analyse statistiques est parfois nécessaire (analyse de la variance). Cette interprétation permet soit : [Ber00]

- **dans le cas de résultats non satisfaisants**, d'émettre des recommandations, comme la remise en question des paramètres ou le cas échéant, des objectifs. Dans ce cas, ces recommandations (hypothèses de plan d'action) sont encore à tester et donc à réintroduire dans le modèle de simulation. Il faut ensuite à nouveau effectuer une simulation avec ce nouveau paramétrage, et cela, jusqu'à satisfaction des résultats.
- **dans le cas de résultats satisfaisants**, de mettre en œuvre les plans d'action sur le système réel (nous passons alors à l'étape suivante).

10. Mise en œuvre des plans d'action sur le système réel

Les plans d'action, suite aux divers tests sont à appliquer sur le système réel en fin de simulation, lorsque les différents acteurs-décideurs sont d'accord sur les résultats et sur le choix d'application.

Programme Visual Basic**'plan expérimental**

```
Const MDO1 = 144, MDO2 = 140, MDO3 = 141, MDO4
= 150
Const C = 4
Const t12 = 150, t13 = 270, t14 = 400, t23 = 120, t24 =
250, t21 = 350, t34 = 130, t31 = 230, t32 = 380, t41 =
100, t42 = 250, t43 = 370
```

'Variables globales

```
Dim TPT1(120), TPT2(120), TPT3(120), TPT4(120),
DO1(120), DO2(120), DO3(120), DO4(120)
Dim dime1 As Integer
Dim dime2 As Integer
Dim dime3 As Integer
Dim dime4 As Integer
Dim oSIMAN As Arena.SIMAN
```

Function FileAttente (NQ As Integer, DOG, TPTG,
MDOG, dime As Integer, TNOW, temps, NOG)

```
Dim i, k, NG, u, NT, n, j, p, w As Integer
Dim DOmax, pmax, y(120), x(20), v, V1(120), V2(120),
Q2(120), R
NG = 0
NT = 0
p = dime - 1
```

'si le vecteur est vide

```
If dime = 0 Then
NG = 0
GoTo 2
End If
```

'algorithme de tri

```
For i = 0 To p
V1(i) = DOG(i)
V2(i) = TPTG(i)
Next i
For j = 0 To p
pmax = V2(j)
DOmax = V1(j)
v = -1
For n = j + 1 To p
If V2(n) < pmax Then
pmax = V2(n)
DOmax = V1(n)
v = n
End If
Next n
If v > -1 Then
V2(v) = V2(j)
V1(v) = V1(j)
V2(j) = pmax
V1(j) = DOmax
End If
Next j
```

'Calcul du nombre de pièces engagées pour la période [TNOW, temps]

```
w = 1
For i = 0 To p
If (TNOW <= V2(i)) And (V2(i) <= temps) Then
Q2(w) = V1(i)
NG = NG + 1
w = w + 1
End If
```

```
Next i
```

2 'Calcul du nombre de pièces traitées pendant [TNOW, temps]

```
x(0) = temps - TNOW
If NQ = 0 Then
GoTo 31
Else: GoTo 32
End If
```

```
31 If NOG = -2 Then
If (x(0) - MDOG) > 0 Then
NT = 1
x(1) = x(0) - MDOG
GoTo 100
Else: NT = 0
GoTo 1
End If
Else: x(1) = x(0)
GoTo 100
End If
```

```
100 If NG = 0 Then
GoTo 1
Else: GoTo 110
End If
```

```
110 For k = 1 To NG
If (x(k) - Q2(k)) > 0 Then
x(k + 1) = x(k) - Q2(k)
NT = NT + 1
Else: GoTo 1
End If
Next k
GoTo 1
```

```
32 If (x(0) - MDOG) > 0 Then
x(1) = (x(0) - MDOG)
NT = 1
GoTo 33
Else: NT = 0
GoTo 1
End If
```

```
33 For u = 1 To NQ
If (x(u) - MDOG) > 0 Then
x(u + 1) = x(u) - MDOG
NT = NT + 1
Else: GoTo 1
End If
Next u
```

```
12 If NG = 0 Then
GoTo 1
End If
y(0) = x(u)
For k = 1 To NG
If (y(k) - Q2(k)) > 0 Then
y(k + 1) = y(k) - Q2(k)
NT = NT + 1
Else: GoTo 1
End If
Next k
1 R = NQ + NG - NT
If R = -1 Then
FileAttente = 0
Else: FileAttente = R
End If
```

End Function

Function min(x1, x2, x3, x4)

Dim B(5), pmax, v

Dim j As Integer

Dim n As Integer

B(1) = x1

B(2) = x2

B(3) = x3

B(4) = x4

For j = 1 To 4

pmax = B(j)

v = -1

For n = j + 1 To 4

If B(n) < pmax Then

pmax = B(n)

v = n

End If

Next n

If v > -1 Then

B(v) = B(j)

B(j) = pmax

End If

Next j

min = B(1)

End Function

Private Sub ModelLogic_RunBeginSimulation()

dime1 = 0

dime2 = 0

dime3 = 0

dime4 = 0

Set oSIMAN = ThisDocument.Model.SIMAN

End Sub

Private Sub VBA_Block_1_Fire()

'station 1

Dim NQ As Integer

Dim NQ1 As Integer

Dim TNOW, D1, xEG, FA2, FA3, D2, D3, D4, FA4,

NO4, NO2, NO3, temps3, temps2, temps4

TNOW = oSIMAN.RunCurrentTime

D1 = oSIMAN.EntityAttribute(oSIMAN.ActiveEntity,

oSIMAN.SymbolNumber("DO1"))

xEG = oSIMAN.EntityAttribute(oSIMAN.ActiveEntity,

oSIMAN.SymbolNumber("EG"))

NQ1 =

oSIMAN.QueueNumberOfEntities(oSIMAN.SymbolNumber("FAE1"))

D2 = oSIMAN.EntityAttribute(oSIMAN.ActiveEntity,

oSIMAN.SymbolNumber("DO2"))

D3 = oSIMAN.EntityAttribute(oSIMAN.ActiveEntity,

oSIMAN.SymbolNumber("DO3"))

D4 = oSIMAN.EntityAttribute(oSIMAN.ActiveEntity,

oSIMAN.SymbolNumber("DO4"))

NO2 =

oSIMAN.ResourceState(oSIMAN.SymbolNumber("Robot2"))

NO3 =

oSIMAN.ResourceState(oSIMAN.SymbolNumber("Robot3"))

NO4 =

oSIMAN.ResourceState(oSIMAN.SymbolNumber("Robot4"))

If D1 > 0 Then

GoTo 1

Else:

oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("VarArray1")) = 12

GoTo 4

End If

1 If xEG = 1 Then

oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("VarArray1")) = 1

GoTo 4

Else: GoTo 3

End If

3 If NQ1 < C Then

GoTo 5

Else:

oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("VarArray1")) = 12

GoTo 4

End If

5 'est ce que la station 1 est la plus adaptée pour prendre en charge cette pièce?'

temps2 = TNOW + t12

temps3 = TNOW + t13

temps4 = TNOW + t14

FA2 = 1000

FA3 = 1000

FA4 = 1000

If D2 > 0 Then

NQ =

oSIMAN.QueueNumberOfEntities(oSIMAN.SymbolNumber("FAE2"))

FA2 = FileAttente(NQ, DO2, TPT2, MDO2, dime2,

TNOW, temps2, NO2)

End If

If D3 > 0 Then

NQ =

oSIMAN.QueueNumberOfEntities(oSIMAN.SymbolNumber("FAE3"))

FA3 = FileAttente(NQ, DO3, TPT3, MDO3, dime3,

TNOW, temps3, NO3)

End If

If D4 > 0 Then

NQ =

oSIMAN.QueueNumberOfEntities(oSIMAN.SymbolNumber("FAE4"))

FA4 = FileAttente(NQ, DO4, TPT4, MDO4, dime4,

TNOW, temps4, NO4)

End If

If NQ1 = min(NQ1, FA2, FA3, FA4) Then

oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("VarArray1")) = 1

GoTo 4

Else: GoTo 13

End If

13 'Recherche de la station la plus adaptée'

```
If FA2 = min(NQ1, FA2, FA3, FA4) Then
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray1")) = 2
DO2(dime2) = D2
TPT2(dime2) = temps2
dime2 = dime2 + 1
GoTo 4
Else: GoTo 14
End If
```

```
14 If FA3 = min(NQ1, FA2, FA3, FA4) Then
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray1")) = 3
DO3(dime3) = D3
TPT3(dime3) = temps3
dime3 = dime3 + 1
GoTo 4
Else: GoTo 15
End If
```

```
15 If FA4 = min(NQ1, FA2, FA3, FA4) Then
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray1")) = 4
DO4(dime4) = D4
TPT4(dime4) = temps4
dime4 = dime4 + 1
GoTo 4
End If
MsgBox "error"
4 End Sub
```

Private Sub VBA_Block_5_Fire()**'station2**

```
Dim NQ As Integer
Dim NQ2 As Integer
Dim TNOW, D1, xEG, FA1, FA3, D2, D3, D4, FA4,
NO1, NO3, NO4, temps1, temps3, temps4
```

```
TNOW = oSIMAN.RunCurrentTime
D2 = oSIMAN.EntityAttribute(oSIMAN.ActiveEntity,
oSIMAN.SymbolNumber("DO2"))
xEG = oSIMAN.EntityAttribute(oSIMAN.ActiveEntity,
oSIMAN.SymbolNumber("EG"))
NQ2 =
oSIMAN.QueueNumberOfEntities(oSIMAN.SymbolNu
mber("FAE2"))
D1 = oSIMAN.EntityAttribute(oSIMAN.ActiveEntity,
oSIMAN.SymbolNumber("DO1"))
D3 = oSIMAN.EntityAttribute(oSIMAN.ActiveEntity,
oSIMAN.SymbolNumber("DO3"))
D4 = oSIMAN.EntityAttribute(oSIMAN.ActiveEntity,
oSIMAN.SymbolNumber("DO4"))
NO1 =
oSIMAN.ResourceState(oSIMAN.SymbolNumber("Rob
ot1"))
NO3 =
oSIMAN.ResourceState(oSIMAN.SymbolNumber("Rob
ot3"))
NO4 =
oSIMAN.ResourceState(oSIMAN.SymbolNumber("Rob
ot4"))
```

```
If D2 > 0 Then
GoTo 1
Else:
```

```
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray2")) = 23
GoTo 4
End If
```

```
1 If xEG = 2 Then
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray2")) = 2
GoTo 4
Else: GoTo 3
End If
```

```
3 If NQ2 < C Then
GoTo 5
Else:
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray2")) = 23
GoTo 4
End If
```

5 'est ce que la station 2 est la plus adaptée pour prendre en charge cette pièce?'

```
temps1 = TNOW + t21
temps3 = TNOW + t23
temps4 = TNOW + t24
FA1 = 1000
FA3 = 1000
FA4 = 1000
```

```
If D3 > 0 Then
NQ =
oSIMAN.QueueNumberOfEntities(oSIMAN.SymbolNu
mber("FAE3"))
FAE3 = FileAttente(NQ, DO3, TPT3, MDO3, dime3,
TNOW, temps3, NO3)
End If
```

```
If D4 > 0 Then
NQ =
oSIMAN.QueueNumberOfEntities(oSIMAN.SymbolNu
mber("FAE4"))
FA4 = FileAttente(NQ, DO4, TPT4, MDO4, dime4,
TNOW, temps4, NO4)
End If
```

```
If D1 > 0 Then
NQ =
oSIMAN.QueueNumberOfEntities(oSIMAN.SymbolNu
mber("FAE1"))
FA1 = FileAttente(NQ, DO1, TPT1, MDO1, dime1,
TNOW, temps1, NO1)
End If
```

```
If NQ2 = min(NQ2, FA1, FA3, FA4) Then
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray2")) = 2
GoTo 4
Else: GoTo 13
End If
```

13 'Recherche de la station la plus adaptée'

```
If FA3 = min(NQ2, FA1, FA3, FA4) Then
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray2")) = 3
DO3(dime3) = D3
```

```
TPT3(dime3) = temps3
dime3 = dime3 + 1
GoTo 4
Else: GoTo 14
End If

14 If FA4 = min(NQ2, FA1, FA3, FA4) Then
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray2")) = 4
DO4(dime4) = D4
TPT4(dime4) = temps4
dime4 = dime4 + 1
GoTo 4
Else: GoTo 15
End If
```

```
15 If FA1 = min(NQ2, FA1, FA3, FA4) Then
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray2")) = 1
DO1(dime1) = D1
TPT1(dime1) = temps1
dime1 = dime1 + 1
GoTo 4
End If
MsgBox "error"
4 End Sub
```

Private Sub VBA_Block_6_Fire()

'station 3

```
Dim NQ As Integer
Dim NQ3 As Integer
Dim TNOW, D1, xEG, FA2, FA1, D2, D3, D4, FA4,
NO1, NO2, NO4, temps1, temps2, temps4
```

```
TNOW = oSIMAN.RunCurrentTime
D3 = oSIMAN.EntityAttribute(oSIMAN.ActiveEntity,
oSIMAN.SymbolNumber("DO3"))
xEG = oSIMAN.EntityAttribute(oSIMAN.ActiveEntity,
oSIMAN.SymbolNumber("EG"))
NQ3 =
oSIMAN.QueueNumberOfEntities(oSIMAN.SymbolNu
mber("FAE3"))
D1 = oSIMAN.EntityAttribute(oSIMAN.ActiveEntity,
oSIMAN.SymbolNumber("DO1"))
D2 = oSIMAN.EntityAttribute(oSIMAN.ActiveEntity,
oSIMAN.SymbolNumber("DO2"))
D4 = oSIMAN.EntityAttribute(oSIMAN.ActiveEntity,
oSIMAN.SymbolNumber("DO4"))
NO2 =
oSIMAN.ResourceState(oSIMAN.SymbolNumber("Rob
ot2"))
NO1 =
oSIMAN.ResourceState(oSIMAN.SymbolNumber("Rob
ot1"))
NO4 =
oSIMAN.ResourceState(oSIMAN.SymbolNumber("Rob
ot4"))
```

```
If D3 > 0 Then
GoTo 1
Else:
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray3")) = 34
GoTo 4
```

```
End If

1 If xEG = 3 Then
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray3")) = 3
GoTo 4
Else: GoTo 3
End If

3 If NQ3 < C Then
GoTo 5
Else:
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray3")) = 34
GoTo 4
End If
```

5 'est ce que la station 3 est la plus adaptée pour prendre en charge cette pièce?'

```
temps1 = TNOW + t31
temps2 = TNOW + t32
temps4 = TNOW + t34
FA1 = 1000
FA2 = 1000
FA4 = 1000
```

```
If D4 > 0 Then
NQ =
oSIMAN.QueueNumberOfEntities(oSIMAN.SymbolNu
mber("FAE4"))
FA4 = FileAttente(NQ, DO4, TPT4, MDO4, dime4,
TNOW, temps4, NO4)
End If
```

```
If D1 > 0 Then
NQ =
oSIMAN.QueueNumberOfEntities(oSIMAN.SymbolNu
mber("FAE1"))
FA1 = FileAttente(NQ, DO1, TPT1, MDO1, dime1,
TNOW, temps1, NO1)
End If
```

```
If D2 > 0 Then
NQ =
oSIMAN.QueueNumberOfEntities(oSIMAN.SymbolNu
mber("FAE2"))
FAE2 = FileAttente(NQ, DO2, TPT2, MDO2, dime2,
TNOW, temps2, NO2)
End If
```

```
If NQ3 = min(NQ3, FA1, FA2, FA4) Then
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray3")) = 3
GoTo 4
Else: GoTo 13
End If
```

13 'Recherche de la station la plus adaptée'

```
If FA4 = min(NQ3, FA1, FA2, FA4) Then
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray3")) = 4
DO4(dime4) = D4
TPT4(dime4) = temps4
dime4 = dime4 + 1
GoTo 4
```

```
Else: GoTo 14
End If
```

```
14 If FA1 = min(NQ3, FA1, FA2, FA4) Then
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray3")) = 1
DO1(dime1) = D1
TPT1(dime1) = temps1
dime1 = dime1 + 1
GoTo 4
Else: GoTo 15
End If
```

```
15 If FA2 = min(NQ3, FA1, FA2, FA4) Then
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray3")) = 2
DO2(dime2) = D2
TPT2(dime2) = temps2
dime2 = dime2 + 1
GoTo 4
End If
MsgBox "error"
4 End Sub
```

Private Sub VBA_Block_7_Fire()

'station4

```
Dim NQ As Integer
Dim NQ4 As Integer
Dim TNOW, D1, xEG, FA2, FA3, D2, D3, D4, FA1,
NO1, NO2, NO3, temps1, temps2, temps3
```

```
TNOW = oSIMAN.RunCurrentTime
D2 = oSIMAN.EntityAttribute(oSIMAN.ActiveEntity,
oSIMAN.SymbolNumber("DO2"))
xEG = oSIMAN.EntityAttribute(oSIMAN.ActiveEntity,
oSIMAN.SymbolNumber("EG"))
NQ4 =
oSIMAN.QueueNumberOfEntities(oSIMAN.SymbolNu
mber("FAE4"))
D1 = oSIMAN.EntityAttribute(oSIMAN.ActiveEntity,
oSIMAN.SymbolNumber("DO1"))
D3 = oSIMAN.EntityAttribute(oSIMAN.ActiveEntity,
oSIMAN.SymbolNumber("DO3"))
D4 = oSIMAN.EntityAttribute(oSIMAN.ActiveEntity,
oSIMAN.SymbolNumber("DO4"))
NO2 =
oSIMAN.ResourceState(oSIMAN.SymbolNumber("Rob
ot2"))
NO3 =
oSIMAN.ResourceState(oSIMAN.SymbolNumber("Rob
ot3"))
NO1 =
oSIMAN.ResourceState(oSIMAN.SymbolNumber("Rob
ot1"))
```

```
If D4 > 0 Then
GoTo 1
Else:
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray4")) = 41
GoTo 4
End If
```

```
1 If xEG = 4 Then
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray4")) = 4
```

```
GoTo 4
Else: GoTo 3
End If
```

```
3 If NQ4 < C Then
GoTo 5
Else:
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray4")) = 41
GoTo 4
End If
```

5 'est ce que la station 4 est la plus adaptée pour prendre en charge cette pièce?'

```
temps1 = TNOW + t41
temps3 = TNOW + t43
temps2 = TNOW + t42
FA1 = 1000
FA3 = 1000
FA2 = 1000
```

```
If D1 > 0 Then
NQ =
oSIMAN.QueueNumberOfEntities(oSIMAN.SymbolNu
mber("FAE1"))
FA1 = FileAttente(NQ, DO1, TPT1, MDO1, dime1,
TNOW, temps1, NO1)
End If
```

```
If D2 > 0 Then
NQ =
oSIMAN.QueueNumberOfEntities(oSIMAN.SymbolNu
mber("FAE2"))
FA2 = FileAttente(NQ, DO2, TPT2, MDO2, dime2,
TNOW, temps2, NO2)
End If
```

```
If D3 > 0 Then
NQ =
oSIMAN.QueueNumberOfEntities(oSIMAN.SymbolNu
mber("FAE3"))
FAE3 = FileAttente(NQ, DO3, TPT3, MDO3, dime3,
TNOW, temps3, NO3)
End If
```

```
If NQ4 = min(NQ4, FA1, FA3, FA2) Then
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray4")) = 4
GoTo 4
Else: GoTo 13
End If
```

13 'Recherche de la station la plus adaptée'

```
If FA1 = min(NQ4, FA1, FA3, FA2) Then
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray4")) = 1
DO1(dime1) = D1
TPT1(dime1) = temps1
dime1 = dime1 + 1
GoTo 4
Else: GoTo 14
End If
```

```
14 If FA2 = min(NQ4, FA1, FA3, FA2) Then
```

```
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray4")) = 2
DO2(dime2) = D2
TPT2(dime2) = temps2
dime2 = dime2 + 1
GoTo 4
Else: GoTo 15
End If
```

```
15 If FA3 = min(NQ4, FA1, FA3, FA2) Then
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber(
"VarArray4")) = 3
DO3(dime3) = D3
TPT3(dime3) = temps3
dime3 = dime3 + 1
GoTo 4
End If
MsgBox "error"
4 End Sub
```

Annexe 6 : plan d'expérimentation

Capacité des FAEs	Cas	Règles d'ordonnement	N°
3	1	FIFO	1
		Règle 1	2
		Règle 2	3
	2	FIFO	4
		Règle 1	5
		Règle 2	6
	3	FIFO	7
		Règle 1	8
		Règle 2	9
	4	FIFO	10
		Règle 1	11
		Règle 2	12
	5	FIFO	13
		Règle 1	14
		Règle 2	15
	6	FIFO	16
		Règle 1	17
		Règle 2	18
	7	FIFO	19
		Règle 1	20
		Règle 2	21
	8	FIFO	22
		Règle 1	23
		Règle 2	24
	9	FIFO	25
		Règle 1	26
		Règle 2	27
	10	FIFO	28

		Règle 1	29
		Règle 2	30
11	FIFO	31	
	Règle 1	32	
	Règle 2	33	
12	FIFO	34	
	Règle 1	35	
	Règle 2	36	
13	FIFO	37	
	Règle 1	38	
	Règle 2	39	
14	FIFO	40	
	Règle 1	41	
	Règle 2	42	
15	FIFO	43	
	Règle 1	44	
	Règle 2	45	
16	FIFO	46	
	Règle 1	47	
	Règle 2	48	
17	FIFO	49	
	Règle 1	50	
	Règle 2	51	
18	FIFO	52	
	Règle 1	53	
	Règle 2	54	
19	FIFO	55	
	Règle 1	56	
	Règle 2	57	
20	FIFO	58	
	Règle 1	59	
	Règle 2	60	

21	FIFO	61
	Règle 1	62
	Règle 2	63
22	FIFO	64
	Règle 1	65
	Règle 2	66
23	FIFO	67
	Règle 1	68
	Règle 2	69
24	FIFO	70
	Règle 1	71
	Règle 2	72
25	FIFO	73
	Règle 1	74
	Règle 2	75
26	FIFO	76
	Règle 1	77
	Règle 2	78
27	FIFO	79
	Règle 1	80
	Règle 2	81
28	FIFO	82
	Règle 1	83
	Règle 2	84
29	FIFO	85
	Règle 1	86
	Règle 2	87
30	FIFO	88
	Règle 1	89
	Règle 2	90
31	FIFO	91
	Règle 1	92

		Règle 2	93
32	FIFO	94	
	Règle 1	95	
	Règle 2	96	
33	FIFO	97	
	Règle 1	98	
	Règle 2	99	
34	FIFO	100	
	Règle 1	101	
	Règle 2	102	
35	FIFO	103	
	Règle 1	104	
	Règle 2	105	
36	FIFO	106	
	Règle 1	107	
	Règle 2	108	
37	FIFO	109	
	Règle 1	110	
	Règle 2	111	
38	FIFO	112	
	Règle 1	113	
	Règle 2	114	
39	FIFO	115	
	Règle 1	116	
	Règle 2	117	
40	FIFO	118	
	Règle 1	119	
	Règle 2	120	
41	FIFO	121	
	Règle 1	122	
	Règle 2	123	
42	FIFO	124	

		Règle 1	125	
		Règle 2	126	
	43	FIFO	127	
		Règle 1	128	
		Règle 2	129	
	44	FIFO	130	
		Règle 1	131	
		Règle 2	132	
	45	FIFO	133	
		Règle 1	134	
		Règle 2	135	
	46	FIFO	136	
		Règle 1	137	
		Règle 2	138	
	47	FIFO	139	
		Règle 1	140	
		Règle 2	141	
	48	FIFO	142	
		Règle 1	143	
		Règle 2	144	
	49	FIFO	145	
		Règle 1	146	
		Règle 2	147	
	50	FIFO	148	
		Règle 1	149	
		Règle 2	150	
	4	1	FIFO	151
			Règle 1	152
Règle 2			153	
2		FIFO	154	
		Règle 1	155	
		Règle 2	156	

	3	FIFO	157
		Règle 1	158
			Règle 2
	4	FIFO	160
		Règle 1	161
		Règle 2	162
	5	FIFO	163
		Règle 1	164
		Règle 2	165
	6	FIFO	166
		Règle 1	167
		Règle 2	168
	7	FIFO	169
		Règle 1	170
		Règle 2	171
	8	FIFO	172
		Règle 1	173
		Règle 2	174
	9	FIFO	175
		Règle 1	176
		Règle 2	177
	10	FIFO	178
		Règle 1	179
		Règle 2	180
	11	FIFO	181
		Règle 1	182
		Règle 2	183
	12	FIFO	184
Règle 1		185	
	Règle 2	186	
13	FIFO	187	
	Règle 1	188	

		Règle 2	189
		FIFO	190
	14	Règle 1	191
		Règle 2	192
	15	FIFO	193
		Règle 1	194
		Règle 2	195
	16	FIFO	196
		Règle 1	197
		Règle 2	198
	17	FIFO	199
		Règle 1	200
		Règle 2	201
	18	FIFO	202
		Règle 1	203
		Règle 2	204
	19	FIFO	205
		Règle 1	206
		Règle 2	207
	20	FIFO	208
		Règle 1	209
		Règle 2	210
	21	FIFO	211
		Règle 1	212
		Règle 2	213
	22	FIFO	214
		Règle 1	215
		Règle 2	216
23	FIFO	217	
	Règle 1	218	
	Règle 2	219	
24	FIFO	220	

		Règle 1	221
		Règle 2	222
	25	FIFO	223
		Règle 1	224
		Règle 2	225
	26	FIFO	226
		Règle 1	227
		Règle 2	228
	27	FIFO	229
		Règle 1	230
		Règle 2	231
	28	FIFO	232
		Règle 1	233
		Règle 2	234
	29	FIFO	235
		Règle 1	236
		Règle 2	237
	30	FIFO	238
		Règle 1	239
		Règle 2	240
	31	FIFO	241
		Règle 1	242
		Règle 2	243
	32	FIFO	244
		Règle 1	245
		Règle 2	246
	33	FIFO	247
		Règle 1	248
	Règle 2	249	
34	FIFO	250	
	Règle 1	251	
	Règle 2	252	

35	FIFO	253
	Règle 1	254
	Règle 2	255
36	FIFO	256
	Règle 1	257
	Règle 2	258
37	FIFO	259
	Règle 1	260
	Règle 2	261
38	FIFO	262
	Règle 1	263
	Règle 2	264
39	FIFO	265
	Règle 1	266
	Règle 2	267
40	FIFO	268
	Règle 1	269
	Règle 2	270
41	FIFO	271
	Règle 1	272
	Règle 2	273
42	FIFO	274
	Règle 1	275
	Règle 2	276
43	FIFO	277
	Règle 1	278
	Règle 2	279
44	FIFO	280
	Règle 1	281
	Règle 2	282
45	FIFO	283
	Règle 1	284

		Règle 2	285
	46	FIFO	286
		Règle 1	287
		Règle 2	288
		47	FIFO
	Règle 1		290
		Règle 2	291
		48	FIFO
	Règle 1		293
		Règle 2	294
49		FIFO	295
	Règle 1	296	
	Règle 2	297	
	50	FIFO	298
Règle 1		299	
	Règle 2	300	
	5	1	FIFO
Règle 1			302
Règle 2			303
	2	FIFO	304
		Règle 1	305
		Règle 2	306
	3	FIFO	307
		Règle 1	308
		Règle 2	309
	4	FIFO	310
		Règle 1	311
		Règle 2	312
	5	FIFO	313
		Règle 1	314
		Règle 2	315
6	FIFO	316	

		Règle 1	317
		Règle 2	318
7	FIFO	319	
	Règle 1	320	
	Règle 2	321	
	8	FIFO	322
Règle 1		323	
	Règle 2	324	
	9	FIFO	325
Règle 1		326	
	Règle 2	327	
	10	FIFO	328
Règle 1		329	
	Règle 2	330	
	11	FIFO	331
Règle 1		332	
	Règle 2	333	
	12	FIFO	334
Règle 1		335	
	Règle 2	336	
	13	FIFO	337
Règle 1		338	
	Règle 2	339	
	14	FIFO	340
Règle 1		341	
	Règle 2	342	
	15	FIFO	343
Règle 1		344	
	Règle 2	345	
	16	FIFO	346
Règle 1		347	
	Règle 2	348	

17	FIFO	349
	Règle 1	350
	Règle 2	351
	18	FIFO
Règle 1		353
	Règle 2	354
	19	FIFO
Règle 1		356
	Règle 2	357
	20	FIFO
Règle 1		359
	Règle 2	360
	21	FIFO
Règle 1		362
	Règle 2	363
	22	FIFO
Règle 1		365
	Règle 2	366
	23	FIFO
Règle 1		368
	Règle 2	369
	24	FIFO
Règle 1		371
	Règle 2	372
	25	FIFO
Règle 1		374
	Règle 2	375
	26	FIFO
Règle 1		377
	Règle 2	378
	27	FIFO
Règle 1		380

	Règle 2	381
28	FIFO	382
	Règle 1	383
	Règle 2	384
29	FIFO	385
	Règle 1	386
	Règle 2	387
30	FIFO	388
	Règle 1	389
	Règle 2	390
31	FIFO	391
	Règle 1	392
	Règle 2	393
32	FIFO	394
	Règle 1	395
	Règle 2	396
33	FIFO	397
	Règle 1	398
	Règle 2	399
34	FIFO	400
	Règle 1	401
	Règle 2	402
35	FIFO	403
	Règle 1	404
	Règle 2	405
36	FIFO	406
	Règle 1	407
	Règle 2	408
37	FIFO	409
	Règle 1	410
	Règle 2	411
38	FIFO	412

	Règle 1	413
	Règle 2	414
39	FIFO	415
	Règle 1	416
	Règle 2	417
40	FIFO	418
	Règle 1	419
	Règle 2	420
41	FIFO	421
	Règle 1	422
	Règle 2	423
42	FIFO	424
	Règle 1	425
	Règle 2	426
43	FIFO	427
	Règle 1	428
	Règle 2	429
44	FIFO	430
	Règle 1	431
	Règle 2	432
45	FIFO	433
	Règle 1	434
	Règle 2	435
46	FIFO	436
	Règle 1	437
	Règle 2	438
47	FIFO	439
	Règle 1	440
	Règle 2	441
48	FIFO	442
	Règle 1	443
	Règle 2	444

49	FIFO	445
	Règle 1	446
	Règle 2	447
50	FIFO	448
	Règle 1	449
	Règle 2	450

Tableau 1 : détail du plan expérimental

Annexe 6 : plan d'expérimentation

N° expérience	JOB	DO1	DO2	DO3	DO4
1	J1	100	100	100	100
	J2	100	100	100	100
	J3	100	100	100	100
	J4	100	100	100	100
2	J1	95	100	100	100
	J2	90	100	100	100
	J3	98	100	100	100
	J4	96	100	100	100
3	J1	100	95	100	100
	J2	100	90	100	100
	J3	100	98	100	100
	J4	100	96	100	100
4	J1	100	100	95	100
	J2	100	100	90	100
	J3	100	100	98	100
	J4	100	100	96	100
5	J1	100	100	100	95
	J2	100	100	100	90
	J3	100	100	100	98
	J4	100	100	100	96
6	J1	95	102	100	100
	J2	90	110	100	100
	J3	98	105	100	100
	J4	96	108	100	100
7	J1	100	95	102	100
	J2	100	90	110	100
	J3	100	98	105	100
	J4	100	96	108	100
8	J1	95	100	100	102
	J2	90	100	100	110
	J3	98	100	100	105
	J4	96	100	100	108
9	J1	95	102	92	100
	J2	90	110	95	100
	J3	98	105	94	100
	J4	96	108	97	100
10	J1	100	95	102	105
	J2	100	90	110	115
	J3	100	98	105	114
	J4	100	96	108	112

11	J1	95	100	92	105
	J2	90	100	95	115
	J3	98	100	94	114
	J4	96	100	99	112
12	J1	95	102	100	105
	J2	90	110	100	115
	J3	98	105	100	114
	J4	96	108	100	112
13	J1	92	100	100	100
	J2	88	100	100	100
	J3	95	100	100	100
	J4	85	100	100	100
14	J1	100	90	100	100
	J2	100	85	100	100
	J3	100	80	100	100
	J4	100	95	100	100
15	J1	81	88	89	87
	J2	90	78	86	88
	J3	85	92	78	86
	J4	90	92	78	84
16	J1	58	60	75	70
	J2	68	62	72	59
	J3	63	69	78	60
	J4	67	57	76	64
17	J1	35	46	50	47
	J2	39	32	58	45
	J3	40	42	48	39
	J4	42	48	56	38
18	J1	42	35	38	30
	J2	52	30	45	35
	J3	48	39	40	33
	J4	50	39	42	32
19	J1	105	110	100	100
	J2	110	118	100	100
	J3	112	115	100	100
	J4	108	125	100	100
20	J1	100	108	122	105
	J2	114	117	125	119
	J3	106	112	124	118
	J4	102	100	120	100
21	J1	87	83	100	80

22	J2	82	90	98	88
	J3	85	92	100	81
	J4	86	89	95	91
	J1	144	138	135	150
23	J2	139	135	128	148
	J3	135	125	141	146
	J4	132	140	136	145
	J1	148	160	147	135
24	J2	136	156	149	142
	J3	150	158	146	145
	J4	140	152	138	148
	J1	150	150	146	150
25	J2	150	150	142	150
	J3	150	150	145	150
	J4	150	150	148	150
	J1	146	150	150	150
26	J2	142	150	150	150
	J3	145	150	150	150
	J4	148	150	150	150
	J1	150	146	150	150
27	J2	150	142	150	150
	J3	150	145	150	150
	J4	150	148	150	150
	J1	150	150	150	146
28	J2	150	150	150	142
	J3	150	150	150	145
	J4	150	150	150	148
	J1	146	143	150	150
29	J2	142	147	150	150
	J3	145	149	150	150
	J4	148	142	150	150
	J1	146	150	143	150
30	J2	142	150	147	150
	J3	145	150	149	150
	J4	148	150	142	150
	J1	146	158	150	150
31	J2	142	155	150	150
	J3	145	156	150	150
	J4	148	160	150	150
	J1	146	150	158	150
J2	142	150	155	150	

Annexe 6 : plan d'expérimentation

32	J3	142	150	156	150
	J4	148	150	160	150
32	J1	146	150	150	158
	J2	142	150	150	155
	J3	142	150	150	156
	J4	148	150	150	160
33	J1	158	146	150	150
	J2	155	142	150	150
	J3	156	142	150	150
	J4	160	148	150	150
34	J1	158	150	146	150
	J2	155	150	142	150
	J3	156	150	145	150
	J4	160	150	148	150
35	J1	158	150	150	146
	J2	155	150	150	142
	J3	156	150	150	145
	J4	160	150	150	148
36	J1	150	146	158	150
	J2	150	142	155	150
	J3	150	145	156	150
	J4	150	148	160	150
37	J1	150	146	150	158
	J2	150	142	150	155
	J3	150	145	150	156
	J4	150	148	150	160
38	J1	150	158	146	150
	J2	150	155	142	150
	J3	150	156	145	150
	J4	150	160	148	150
39	J1	150	158	150	146
	J2	150	155	150	142
	J3	150	156	150	145
	J4	150	160	150	148
40	J1	150	150	158	146
	J2	150	150	155	142
	J3	150	150	156	145
	J4	150	150	160	148
41	J1	150	150	146	158
	J2	150	150	142	155
	J3	150	150	145	156

42	J4	150	150	148	160
	J1	110	100	100	100
	J2	105	100	100	100
	J3	103	100	100	100
43	J4	108	100	100	100
	J1	100	110	100	100
	J2	100	105	100	100
	J3	100	103	100	100
44	J4	100	108	100	100
	J1	100	100	110	100
	J2	100	100	105	100
	J3	100	100	103	100
45	J4	100	100	108	100
	J1	100	100	100	110
	J2	100	100	100	105
	J3	100	100	100	103
46	J4	100	100	100	108
	J1	110	92	100	100
	J2	105	95	100	100
	J3	103	90	100	100
47	J4	108	98	100	100
	J1	110	100	92	100
	J2	105	100	95	100
	J3	103	100	90	100
48	J4	108	100	98	100
	J1	110	92	100	105
	J2	105	95	100	99
	J3	103	90	100	97
49	J4	108	98	100	100
	J1	110	92	105	100
	J2	105	95	99	100
	J3	103	90	97	100
50	J4	108	98	100	100
	J1	100	110	100	100
	J2	100	105	100	100
	J3	100	115	100	100
50	J4	100	125	100	100

Tableau 2 : Les durées opératoires

Activité	transformation d'éléments d'entrée en éléments de sortie (matière, produit, information, flux) selon une règle reproductible. Cette transformation est supportée par un ensemble de ressources (humaines, techniques, etc.).
Commande	niveau de traduction de l'ordre en une séquence d'instructions exécutables par la machine.
Conduite	niveau de la gestion de production responsable de la pile de tâches, de l'intégration des contraintes locales, du lancement, du suivi des produits et de chaque ressource. Ce niveau est responsable de la réalisation des actions de production planifiées.
Contrôle	composante de la conduite assurant la définition des décisions possibles, la prise de décision à proprement parler et l'application de celle-ci de manière cohérente avec les objectifs de production.
Décision	restriction d'un domaine de valeur d'une variable, tri sur plusieurs domaines ou valeurs ou classement sur plusieurs domaines ou valeurs.
Évaluation des performances	mise en évidence des impacts d'une décision, passée ou future, sur le positionnement relatif de la finalité, des objectifs, des résultats et des moyens constitutifs de ces systèmes, sur la globalité du cycle de vie de ces impacts, et sous chacun des points de vue concernés.
Gestion de production	fonction générale consistant à gérer prévisionnellement et à conduire les produits (leur stockage, leur transformation) de la commande des matières premières à la livraison des produits finis.
Gestion prévisionnelle	prise en compte prévisionnelle des besoins, niveaux de stock et charge, regroupe les fonctions de planification, de programmation et d'ordonnancement.
Indicateur de performance	donnée quantifiée qui mesure l'efficacité de tout ou partie d'un processus ou d'un système par rapport à une norme, un plan ou un objectif déterminé et accepté dans le cadre d'une stratégie d'entreprise.
Lancement	au niveau de la conduite, envoi d'une demande d'exécution d'une tâche à la commande de la ressource

Lot	quantité de transformation d'un produit, désigne l'ensemble physique des éléments subissant la transformation.
Modèle	vue « logique » (c'est-à-dire traitant des informations, par opposition à la vue « physique » traitant de l'énergie ou de la matière) et réductrice d'une réalité afin d'en améliorer la compréhension.
Ordonnancement	planning des opérations complété par les affectations des ressources qui vont réaliser ces opérations. Organisation fine du travail, équilibrage de la charge.
Planification	planification à long terme des ventes et de la production.
Processus	un processus est constitué d'un ensemble d'activités identifiées en nombre fini et concourant chacune à la réalisation d'un objectif commun.
Programmation	définition des besoins nets, jalonnement, création du programme prévisionnel de production.
Supervision de production	composante de la conduite assurant le suivi de l'état du système de production (productivité, stocks, etc.) et la détection des anomalies.
Temps réel	un système en temps réel est un système qui est capable de réagir dans un horizon temporel fixé par son environnement.