



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université 20 Août 1955 - Skikda

Faculté des Sciences

Département d'Informatique

MEMOIRE

Présenté pour l'obtention du diplôme de Magister en Informatique.

Spécialité : Informatique parallèle et distribuée.

Option : *Techniques Avancées pour les Systèmes Parallèles et Distribués.*

Thème

Formalismes et environnements pour la modélisation et la simulation des systèmes de production

Présenté par :

Bouaita Riad

Soutenu le : / /2015

devant le jury composé de :

Mazouzi Smaine	Président	MC'A'	Université 20 Août 1955- Skikda
Boucheham Bachir	Examineur	MC'A'	Université 20 Août 1955- Skikda
Redjimi Mohammed	Encadreur	Professeur	Université 20 Août 1955- Skikda



REMERCIEMENTS

Toutes mes infinies grâces à Monsieur M. Redjimi Professeur à l'université du 20 Aout 1955 de Skikda pour avoir accepté de diriger ce travail ainsi que pour ses encouragements et son aide précieuse.

Je remercie aussi Monsieur S. Mazouzi maître de conférences classe 'A' à l'université du 20 Aout 1955 de Skikda pour avoir accepté de présider le jury de magister.

Mes chaleureux remerciements aussi à Monsieur B. Boucheham maître de conférences classe 'A' à l'université du 20 Aout 1955 de Skikda pour avoir accepté d'examiner ce travail.

Mes sincères remerciements vont également à mes parents, mon frère et mes sœurs pour leur soutien moral, leur encouragement inconditionnel. Je les remercie également de tout ce qu'ils ont fait pour moi durant toutes ces années.

Mes vifs remerciements vont aussi à tous mes amis et mes collègues de travail. Je tiens à adresser toute ma profonde reconnaissance et ma gratitude à toutes les personnes qui, de près ou de loin, m'ont aidé, conseillé et encouragé.

Résumé

Les systèmes de production deviennent de plus en plus complexes de par le nombre d'entités qui les composent ainsi que vu le nombre très élevé d'interactions entre ces composants. La modélisation et la simulation de tels systèmes nous permettent de représenter leurs structures ainsi que leurs comportements. Aujourd'hui, plusieurs méthodes existent. Nous pouvons considérer les approches mathématiques, mécaniques et informatiques. Les approches formelles telles que les réseaux de Pétri ou les machines à états finis présentent un ensemble d'outils permettant la représentation des modèles et leur évolution dans le temps.

Les méthodes informatiques qui s'appuient sur les concepts de modèles relationnels ou objets ont été appliquées et ont donné de bons résultats. L'approche multi-agent semble donner actuellement lieu à des méthodes et des environnements qui permettent d'avoir de meilleures représentations des entités composant ces systèmes, chacune pouvant être simulée par un système multi-agent. Les interactions entre ces entités (SMA) se faisant grâce à des mécanismes de passage de message ou de tableau noir.

Le travail que nous préconisons consiste, dans un premier temps, à faire un état de l'art très poussé de différentes approches, formalismes, méthodes et environnements touchant à la modélisation. Nous nous intéresserons ensuite plus particulièrement à la modélisation par les systèmes multi-agents.

Un cas pratique de modélisation et de simulation multi-agents d'un système de production de type Job-Shop appliqué au système de pilotage est présenté et détaillé à la fin de ce mémoire.

Mots-clés : *Modélisation et simulation, Systèmes de production, réseaux de Pétri, Systèmes multi-agents, environnements de développements d'applications.*

Abstract

Production systems become more and more complex due to the number of entities which compose them and given the very high number of interactions between these components. The modeling and the simulation of such systems allow us to represent their structures and their behaviors. Today, several methods exist. We can consider the mathematical, mechanical and computing approaches. The formal approaches such as the Petri networks or finite state machines present a set of tools for the representation of models and their evolution over time.

Computational methods that lean on the concepts of relational and object models have been applied and have given good results. The multi-agent approach now seems to give rise to methods and environments that allow for better representation of entities composing these systems, each of which can be simulated by a multi-agent system. The interactions

between these entities (MAS) are doing through message passing mechanisms or blackboard.

The present work consists, at first, in making a very elaborate state of the art of various approaches, formalisms, methods and environments related to modeling. We will then discuss more particularly to modeling multi-agent systems.

A practical case of multi-agent modeling and simulation of a of job-shop production system applied to the piloting system is presented and detailed at the end of this dissertation.

Key-words : *Modeling and Simulation, Production systems, Petri nets, Multi agent systems, Application development environments.*

المخلص

نظم الإنتاج أصبحت أكثر تعقيدا من جهة عدد الكيانات المكونة لها و كذلك بالنظر إلى العدد الكبير جدا من التفاعلات بين هذه المكونات. النمذجة والمحاكاة لأنظمة معينة تسمح لنا بتمثيل بنياتها و كذلك سلوكها. اليوم، توجد عدة طرق . يمكننا النظر في المناهج الرياضية والميكانيكية والمناهج الحاسوبية. المناهج الرسمية مثل شبكات بيترى أو الآلات محدودة الحالات لديها مجموعة من الأدوات لتمثيل النماذج وتطورها على مر الزمن.

الطرق الحاسوبية التي تعتمد على مفاهيم النماذج العلائقية أو الكيان قد نفذت، وقد أعطت نتائج جيدة. يبدو النهج متعدد الوكلاء الآن أنه سمح بوجود طرق وبيئات تسمح بتمثيل أفضل للكيانات المشكلة لهذه الأنظمة، كل منها يمكن أن تحاكي من خلال نظام متعدد الوكلاء. التفاعلات بين هذه الكيانات (SMA) يكون من خلال آليات تمرير الرسائل أو اللوح الأسود.

العمل الذي نشيد به، في البداية، يتمثل في حالة من الفن متطورة من المناهج المختلفة، الشكليات والأساليب والبيئات المتعلقة بالنمذجة. ونهتم بعد ذلك بشكل خاص بالنمذجة بواسطة الأنظمة متعددة الوكلاء.

حالة خاصة للنمذجة والمحاكاة متعددة الوكلاء لنظام إنتاج من نوع Job-Shop مطبقة على نظام التوجيه ستقدم و تفصل في نهاية هذه المذكرة.

كلمات الدلالة : النمذجة والمحاكاة ، نظم الإنتاج ، شبكات بيترى ، الأنظمة متعددة الوكلاء ، بيئات تطوير التطبيقات.

Table des matières :

Introduction générale.....	1
Chapitre 01 : Systèmes de production	
1. Introduction.....	4
2. Généralités sur les systèmes.....	4
2.1. Notion de système	4
2.2. Types de système.....	5
2.3. systèmes complexes.....	6
3. Les systèmes de production.....	7
3.1. Notion de production.....	7
3.2. Notion de système de production.....	8
3.3. Eléments intervenant dans un système de production.....	8
3.3.1. Matière d'œuvre	8
3.3.2. l'information	8
3.3.3. L'énergie.....	8
3.3.4. Les intervenants humains.....	8
3.4. Approche systémique.....	9
3.5. Ressources d'un système de production.....	11
3.6. Flux de production	12
3.7. Pilotage des systèmes de production.....	12
3.8. Classification des systèmes de production.....	14
3.8.1. Classification selon la nature et le volume de flux physique.....	14
3.8.2. Classification selon le mode de pilotage.....	15
3.8.3. Classification selon le mode de cheminement des produits.....	16
3.9. Automatisation des systèmes de production.....	17
3.9.1. Principe.....	17
3.9.2. Structure d'un système automatisé de production.....	18
3.9.3. Fonctionnement d'un système de production.....	20
3.10. Gestion de la production.....	20
3.10.1. Définition.....	20
3.10.2. Fonctions assurées par la gestion de production.....	21

3.10.3. Typologie des décisions prises par la gestion de production.....	21
3.11. Finalités des systèmes de production.....	22
3.12. Caractéristiques des systèmes de production.....	23
4. Conclusion	24

Chapitre 02 : Modélisation et Simulation des systèmes complexes

1. introduction	26
2. Généralité sur les modèles	26
2.1. Notion de modèle	26
2.2. Intérêt d'un modèle	27
2.3. Stratégie de choix d'un modèle	27
2.4. Modèle physique/modèle mathématique	28
2.5. Classification des modèles	29
2.5.1. Classification selon le déterminisme	29
2.5.2. Classification selon la continuité	29
2.5.3. Classification selon le dynamisme	30
2.6. Natures des variables d'un modèle	30
3. Modélisation	30
3.1. Définition	30
3.2. Schéma d'expérimentation	31
3.3. Paradigme de modélisation	31
3.4. Formalisme de modélisation	32
3.5. Modélisation multi-formalismes	32
3.6. Multi-modélisation	32
3.7. Quelques outils, formalismes et approches de modélisation.....	33
3.7.1. Les réseaux de Pétri.....	33
3.7.2. La méthode SADT (Structured Analysis and Design Technic).....	39
3.7.3. La méthode SA-RT (Structured Analysis for Real Time).....	42
3.7.4. Les automates d'état finis.....	43
3.7.5. Les équations différentielles	43
3.7.6. Le formalisme DEVS (<u>D</u> iscret <u>E</u> Vent system <u>S</u> pecification)	44

3.7.7. La modélisation par Agents.....	45
3.7.8. L'outil GRAFCET (GRAphe Fonctionnel de Commande Etape/Transition).....	46
4. Simulation.....	48
4.1. Définition.....	48
4.2. Simulation informatique.....	48
4.3. Processus de conception d'une simulation informatique.....	49
4.4. Catégories de simulation selon la représentation du temps.....	49
4.4.1. Simulation continue.....	49
4.4.2. Simulation discrète.....	50
4.4.3. Simulation à événements discrets	51
5. Les entités constituant la simulation informatique	52
6. La validation dans le processus de simulation.....	53
6.1. Validation du modèle.....	53
6.2. Validation du simulateur.....	54
7. Conclusion.....	54

Chapitre 03 : Modélisation et Simulation des systèmes de production

1. Introduction.....	56
2. Concepts essentiels pour la modélisation d'un système de production.....	56
2.1. L'Entité Circulante (EC)	56
2.1.1. Définition.....	56
2.1.2. Propriétés.....	56
2.2. Le Système de Transformation de Produits (STP)	57
2.3. Le centre de pilotage (CP).....	59
2.3.1. Définition	59
2.3.2. Processus de prise décision.....	61
2.3.3. Typologie des structures de pilotage	62
2.4. Le réseau STP-CP.....	63
3. Les approches hétérarchiques de modélisation des systèmes de production..	65
3.1. Approches Holoniques.....	65

3.2. Approches Bioniques.....	67
3.3. Approches Fractales.....	68
3.4. Approches Multi Agents	72
3.4.1. Agent et Système Multi agent.....	72
3.4.2. La modélisation multi-Agents	74
3.4.3. De la modélisation à la simulation multi-Agents	74
3.4.4. Applications des SMA dans les systèmes de production.....	75
3.4.5. Travaux en relation.....	76
3.4.6. Synthèse.....	83
4. Conclusion.....	85

Chapitre 04 : Modélisation et Simulation multi-agents d'un système de production de type Job-Shop : Application au système de pilotage

1. Introduction.....	87
2. Le système étudié.....	87
3. Objectifs.....	88
4. Structure conceptuelle à base d'agent du système étudié.....	89
4.1. Structure de pilotage considérée.....	89
4.2. Architecture générale multi-agent du système étudié.....	90
5. Fonctionnement de l'approche multi agent.....	94
5.1. Description formelle du système étudié.....	94
5.2. Algorithme d'allocation de ressources.....	95
5.3. Sources d'incertitude.....	97
5.4. Interaction entre agents du système	97
6. Conclusion.....	100
Conclusion Générale et perspectives.....	101
Bibliographie.....	103

Liste des figures

Figure 1.1. Décomposition élémentaire d'un Système de production.....	9
Figure 1.2. Décomposition systémique d'un Système de Production.....	10
Figure 1.3. Décomposition d'un Système de Production selon G.HABCHI et al [HABCHI et al. 2006].....	11
Figure 1.4. Classification des systèmes de production selon Giard [Giard, 2003][Widmer, 1991.....	17
Figure 1.5. Décomposition fonctionnelle d'un système de production automatisé..	18
Figure 1.6. Contraintes et enjeux des systèmes de production.....	22
Figure 2.1. Stratégie de choix d'un modèle [Sperandio, 2005].....	28
Figure 2.2. L'activité de modélisation.....	31
Figure 2.3 Exemple d'un Réseau de Pétri.....	34
Figure 2.4. Réseau de Petri temporisé d'une colonne de séparation de gaz.....	38
Figure 2.5 Actigramme.....	40
Figure 2.6. Datagramme.....	41
Figure 2.7 Hiérarchie des niveaux SADT.....	41
Figure 2.8. GRAFCET correspondant à une barrière automatique.....	47
Figure 2.9. Simulation informatique [Fishwick, 1997].....	49
Figure 2.10. Processus de conception d'une simulation informatique [Shannon, 1998]	50
Figure 2.11. Entités constituant une simulation informatique.....	53
Figure 3.1. Les trois fonctions de base d'un STP [Bakalem et al, 96].....	58
Figure 3.2. Comportement interne d'un Centre de Pilotage.....	60
Figure 3.3. Typologie des structures de pilotage.....	63
Figure 3.4. Exemple d'un réseau STP-CP.....	64
Figure 3.5. Système de production holonique [Van Brussel et al, 1998]	66
Figure 3.6. Entités Fractales d'un système de production [Tharumarajah et al, 1998].....	69

Figure 3.7. Coopération et itération des EFs [Warneke, 1993].....	71
Figure 3.8. Modules fonctionnels et leurs relations d'une Fractale dans un système de production fractal [Ryu et al, 2000]	72
Figure.4.1. Un problème de type job-shop avec 3 Jobs et 3 Ressources.....	88
Figure.4.2. Structure du système de pilotage considérée.....	89
Figure 4.3. Architecture générale multi-agent du système étudié.....	91
Figure 4.4. Architecture modulaire de l'agent Superviseur.....	92
Figure 4.5. Architecture modulaire de l'agent Superviseur.....	92
Figure 4.6. Architecture modulaire de l'agent Tache.....	93
Figure.4.7. Scenario d'allocation de ressources.....	99
Figure.4.8. Diagramme de Gantt pour l'exemple illustratif	100

Liste des tableaux :

Tableau 3.1. Tableau récapitulatif des travaux multi agents pour la modélisation et la simulation des systèmes de production.....	84
Tableau 4.1. Quelques performatives FIPA-ACL.....	98
Tableau 4.2. Exemple illustratif	100

Introduction générale

Avec la mondialisation de l'économie, les entreprises actuelles visent à développer leurs systèmes de production pour faire face à la concurrence et à la diversité des demandes du marché ainsi qu'à la nature incertaine et imprévisible de leurs environnements. Ces facteurs rendent les systèmes de production de plus en plus complexes et posent d'énormes problèmes aux niveaux de leurs analyses, de leurs conceptions, de leurs modélisations et de leurs maintenances. De ce fait, plusieurs outils, méthodes et approches sont utilisés pour permettre aux systèmes de production de s'adapter à ces fluctuations en assurant une production de qualité ayant un rendement optimal et respectant les délais et les coûts.

Les méthodes classiques utilisées dans ce contexte telles que les méthodes analytiques, les méthodes de dimensionnement et d'optimisation ont donné des résultats importants.

Cependant, étant donnée la complexité de ces systèmes engendrée par le nombre important de leurs composants et les fortes interactions entre ces composants, ces méthodes ne peuvent être utilisées. La modélisation et la simulation sont des techniques très puissantes permettant l'analyse, la conception et l'optimisation de ces systèmes. La modélisation peut être utilisée comme un outil puissant pour le contrôle et la prédiction des comportements des systèmes de production. La puissance de la modélisation se justifie par sa flexibilité et par sa capacité à représenter pratiquement n'importe quel système. La simulation informatique est un moyen permettant la maîtrise de la complexité, c'est une technique particulière qui repose sur la modélisation et qui permet l'évaluation des performances des systèmes de production, elle permet une reproduction typique des flux physiques des systèmes de production. Elle permet d'évaluer le modèle en appliquant dessus des entrées externes et en analysant les effets produits.

Les systèmes multi agents offrent de nouvelles capacités de représenter les entités et leurs interactions pour un système de production : l'autonomie de décision, la flexibilité, l'auto-organisation, ...

Dans ce mémoire, nous introduisons un environnement de modélisation et de simulation multi agents d'un système de production de type *Job Shop*. C'est un système capable de produire plusieurs types de produits défini chacun par une gamme de fabrication et utilisant pour cela un ensemble de ressources (machine, robots, énergie,...). L'architecture multi agent proposée est une architecture distribuée supervisée où un ensemble d'agents réagissent d'une façon autonome et coopèrent entre eux afin d'assurer un pilotage en temps réel du système de production.

Le reste de ce mémoire est organisé comme suit :

Dans le premier chapitre, nous présentons d'une manière générale les systèmes de production nous rappelons leurs définitions, leurs types et leurs finalités et caractéristiques.

Dans le second chapitre, la modélisation et la simulation des systèmes complexes sont abordées en commençant d'abord par une présentation des concepts liés au modèle et à la modélisation ainsi qu'une présentation de quelques outils, formalismes et approches de modélisation. Ensuite nous explorons quelques concepts liés à la modélisation et à la simulation.

Dans le troisième chapitre, nous nous intéressons en particulier à la modélisation et à la simulation des systèmes de production. Après avoir exploré les concepts essentiels dans la modélisation et la simulation des systèmes de production, nous présentons un état de l'art sur les approches hétérarchiques utilisées dans ce contexte à savoir les approches holoniques, bioniques, fractales, et multi agents, tout en parcourant quelque travaux de recherche et en se concentrant beaucoup plus sur les approches multi agents.

Dans le quatrième chapitre, nous proposons une architecture multi agents distribuée supervisée pour la modélisation et la simulation d'un système de production de type *Job Shop* appliquée au système de pilotage. Dans ce chapitre, nous présentons l'architecture générale du système multi agent, l'architecture modulaire de chaque agent et un algorithme d'allocation de ressources ainsi qu'un scénario décrivant l'interaction entre les agents du système.

Ce mémoire est ensuite achevé par une conclusion récapitulant notre travail et présentant quelques perspectives de futurs travaux de recherche.

CHAPITRE 1

Les systèmes de production

1. Introduction :

Les systèmes de production jouent un rôle très important dans une entreprise. De ce fait la complexité d'une entreprise est fortement liée à celle de son système de production. Ce dernier devient actuellement plus en plus complexe avec l'introduction des nouvelles technologies, la diversité de ses composants ainsi que des fortes interactions entre ces composants. En effet, les systèmes de production doivent être impérativement conçus de façon à répondre à ces diverses exigences et à s'adapter avec les changements de leurs environnements.

Dans ce chapitre, on présente une vue détaillée sur les systèmes de production (SdP). En commençant par des généralités sur les systèmes puis on essayera de définir, classifier, et présenter les contraintes et les enjeux des systèmes de production.

2. Généralités sur les systèmes :

2.1. Notion de système :

La notion de système est d'un usage très répandu. Plusieurs définitions existent dans la littérature :

Définition 1 [IEEE, 2005] : Un système est un ensemble d'éléments (personnes, produits, processus) qui sont en relation et en interaction dynamique dont la finalité est de répondre au différents besoins.

Définition 2 [Ansi, 1998] : Un système est défini comme un ensemble de produits nécessaires, d'un point de vue fonctionnel (indépendant de la technologie), à la réalisation de la fonction opérationnelle appelés *end product*, et à sa réalisation, sa mise en service, son maintien et son retrait de service d'un point de vue organique (dépendant de la technologie), appelés *enabling products*.

Définition 3 [Rechtin, 2000] : Un système est défini comme « *une construction ou une collection de différents éléments qui produisent ensemble des résultats non accessibles par les éléments seuls. Les éléments ou parties peuvent inclure des personnes, du matériel, du logiciel, des installations, des politiques et des documents, c'est-à-dire tout ce qui est requis pour produire les résultats au niveau système. Les résultats incluent les qualités, les propriétés, les fonctions, le comportement et les performances. La valeur ajoutée par le*

« système pris comme un tout, au delà de celle procurée indépendamment par les parties, est principalement créée par les relations entre parties, c'est-à-dire par la façon dont elles sont interconnectées ».

Définition 4 [Larvet. P, 1994], [Le Moigne, 1990] : Un système est un ensemble d'éléments matériels ou immatériels appelé (selon le domaine) constituants, composants, objets ou agents. Un élément du système peut être lui-même un système composé d'un autre ensemble d'éléments. Ces éléments sont en interaction entre eux, ou avec l'environnement, transformant par un processus, des éléments (les entrées) en d'autres éléments (les sorties).

En effet, un système :

- Possède des frontières : qui permettent de délimiter le système par rapport à son environnement en le séparant de ce dernier.
- Interagit avec l'environnement : à travers des flux matériels set/ou informationnels.
- Fonctionne pour arriver à un but.
- Evolue dans le temps.
- Peut se décomposer en sous-systèmes.
- Peut être vu comme une boîte noire pourvue d'un ensemble d'entrée et d'un ensemble de sorties.

2.2. Types de systèmes : Il existe de nombreuses classifications des systèmes. Nous nous sommes surtout intéressés à celle citée par D. Harel et A. Pnueli [Harel et al. 1985] qui distinguent trois types de systèmes :

- *Systèmes transformationnels* : Ces systèmes transforment des données acquises pour produire des résultats en sortie.
- *Systèmes interactifs* : Ce sont des systèmes qui interagissent - à une vitesse propre à eux - avec leurs environnements.
- *Systèmes réactifs* : Ce sont des systèmes interagissant à une vitesse imposée par l'environnement et d'une façon permanente avec leurs environnements. Ces systèmes ne sont pas totalement maîtrisables ni totalement prévisibles à cause du couplage et les fortes interactions avec l'environnement.

On peut, en outre, classifier les systèmes selon les deux critères : la continuité et le déterminisme.

Classification selon la continuité :

- *Système continu* : un système est dit continu s'il est soumis à des paramètres ou des conditions externes appelées variables qui sont continues (le temps par exemple) et si la fonction de variation de ce système est une fonction continue. C'est-à-dire une information est délivrée à tout instant.
- *Système discret* : un système discret est un système dont le changement des valeurs de ses variables d'état n'est effectué que sur un nombre fini de points, en général on parle d'événements.

Classification selon le déterminisme :

- *Systèmes déterministe* : c'est un système qui se comporte toujours de la même façon s'il est soumis à des conditions identiques, il est possible donc de prédire son évolution au cours du temps.
- *Système stochastique* : appelé aussi aléatoire ou non déterministe. C'est un système qui évolue d'une manière probabiliste dans le temps (par exemple la température quotidienne).

2.3. Systèmes complexes : Le Moigne [Le Moigne, 1990] a défini un système complexe comme étant « *un système que l'on tient pour irréductible à un modèle fini, aussi compliqué, stochastique, sophistiqué, que soit ce modèle, quelle que soit sa taille, le nombre de ses composants, l'intensité de leurs relations,... la notion de complexité implique celle d'imprévisible possible, d'émergence, plausible du nouveau et de sens au sein du phénomène que l'on tient pour complexe* ».

Selon Barwise.J et Etchemendy.J [Barwise et al. 1990], Un système complexe est un système composé d'éléments en forte interaction entre eux et avec l'environnement. Ces éléments sont distribués soit dans le temps soit dans l'espace et impliquent de multiples agents, de multiples représentations et différentes sortes de régularités. C'est un système dont le fonctionnement n'est pas régi par un nombre limité de règles et dont le comportement est imprévisible.

Ces deux définitions permettent de mettre en évidence les éléments fondamentaux de caractérisation de la complexité suivants :

- *L'irréductibilité* : on ne peut pas représenter le système par un modèle unique ou fini. Sa représentation ne se restreint donc pas à une forme unique.

- *Organisation et auto-organisation* : le terme organisation signifie l'agencement ou la structuration des relations inter-composant ou individus composants une unité complexe (système), ce qui assure à ce dernier un état d'équilibre. L'auto-organisation est la réaction autonome d'un système qui est soumis -au cours de son fonctionnement- à des perturbations aléatoires. En effet, des lois ou des règles d'organisation et de fonctionnement sont produites d'une façon permanente en prenant toujours en compte les caractéristiques de complexité du système.
- *Emergence* : Ou propriété émergente, c'est le résultat de l'activité *réursive* d'auto-organisation. C'est un résultat global qui n'est pas observable par les constituants du système complexe.
- *Le non linéarité* : Les variables d'état du système complexe qui sont reliées entre elles ne varient pas proportionnellement les unes aux autres.
- *Le graphe d'interaction est non trivial* : Il n'existe pas forcément un lien d'interaction entre chaque couple d'entités du système. Certains liens sont privilégiés.

La complexité d'un système dépend des trois facteurs suivants [Hardebolle, 2008] :

- *Nombre et nature des éléments* : la complexité est liée au nombre d'éléments qui peut être élevé ou évolutif au cours du temps, ainsi qu'à la diversité de la nature des éléments.
- *La nature de l'organisation interne* : les relations qui existent entre les éléments d'un système peuvent former des réseaux ou des hiérarchies. L'organisation des éléments n'est pas aussi toujours stable au cours du temps à cause des fortes interactions entre les entités constituant le système global.
- *Le couplage avec l'environnement* : la réaction du système avec son environnement implique une sorte d'incertitude et d'imprévisibilité à cause des fortes interactions.

3. Les systèmes de production

3.1. Notion de production :

La production est un ensemble d'opérations constituant un processus. Ce processus permet la création de biens ou de services en exploitant un certain nombre de ressources (équipements, matières d'œuvre, main d'œuvre, informations, ...). La production ne se restreint pas à cette fonction, elle peut également s'étendre à d'autres fonctions : achat, maintenance, conception, ordonnancement des tâches et des ressources, ... [Giard, 2003]

3.2. Notion de système de production :

D'une façon générale ; un système de production peut être considéré comme étant un système composé d'un ensemble d'éléments matériels ou immatériels en interaction dans l'objectif de réaliser une activité de production et en faisant intervenir plusieurs opérations à savoir la transformation, le transfert, l'assemblage et le désassemblage. Ces opérations permettent de transformer la matière première (composants entrants du système) en produits finis ou semi fins (composants sortants du système) [TAMANI, 2008].

On peut définir aussi un système de production comme un système capable de transformer des ressources de différents types (des équipements, des ressources humaines, des matières ou des informations techniques ou procédurales) afin d'obtenir des biens et des services [Giard, 1988].

3.3. Eléments intervenant dans un système de production : Un système de production peut être décomposé structurellement en quatre éléments principaux (figure 1.1) :

3.3.1. Matière d'œuvre : les systèmes de production traitent en général des matière d'œuvre reçues à l'état initial et les transforment en d'autres à état final ou semi-final.

3.3.2. L'information : C'est un ensemble de signaux échangés. Les informations sont essentielles pour le dialogue interne (entre constituants du système) ainsi qu'entre le système et son environnement.

3.3.3. L'énergie : le système de production a besoin d'énergie (électrique, mécanique, hydraulique,...) afin de mettre en œuvre le processus de production.

3.3.4. Les intervenants humains : Pour assurer un bon fonctionnement, le système de production nécessite différents intervenants humains :

- *Personnel d'exploitation* : Chargé de la surveillance, l'approvisionnement, le transport,...
- *Personnel de réglage* : Assure l'adaptation et le réglage du système de production pour assurer les qualités souhaitées.
- *Personnel de maintenance* : interviennent lorsque le système est en état de défaillance ou en cas de perturbation et assurent la maintenance préventive du système.

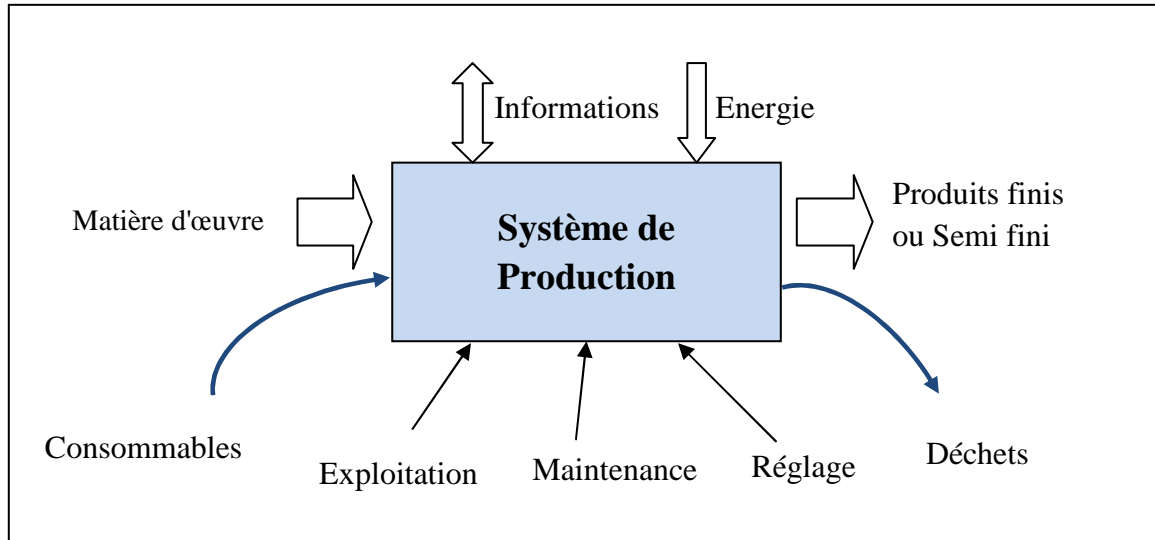


Figure 1.1. Décomposition élémentaire d'un Système de production

3.4. Approche systémique : Les systèmes de production peuvent être décomposés classiquement en trois sous-systèmes coopérants [Le Moigne, 1990] (Figure 1.2) :

- ***Système physique de fabrication*** : Représente la grande partie des infrastructures du système, composé d'un ensemble de *ressources* et *entités*. Les ressources peuvent être actives (machines, robots, moyens de transfert, stocks,...) ou passives (outils, palettes,...). Les entités peuvent être des produits, des matières premières, des pièces, des lots,... En effet un système physique de fabrication est considéré comme un ensemble de processus effectuant des opérations sur des entités.
- ***Système de pilotage*** : Englobe les structures organisationnelles ainsi que les processus de prise de décision du système. Il prend les décisions basées sur les données offertes par le système d'information afin de coordonner et organiser les activités. Il peut être décomposé en :
 - *point de captage (les capteurs)* qui permet la récupération et la collection des informations nécessaires à la prise de décision.
 - *un processus de décision* pour l'analyse, le traitement des informations récupérées et la génération des décisions.

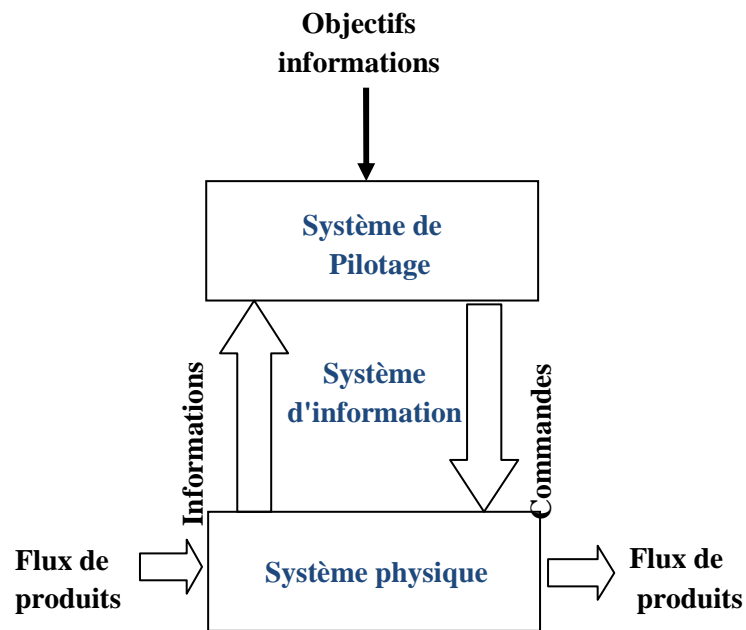


Figure 1.2. Décomposition systémique d'un Système de Production

- *points d'action* permettant le transfert des actions au système physique de fabrication.

La réactivité d'un système de production dépend d'abord de la capacité de réaction du système de pilotage qui dépend à son tour, d'une part de la qualité et la quantité des points d'actions et des points de captage et de l'efficacité du processus de décision, d'autre part.

- **Système d'information** : représente une passerelle entre le système physique de fabrication et le système de pilotage. Il collecte, stocke, traite et transmet les informations nécessaires à la prise de décision.

En outre, il existe d'autres décompositions d'un système de production. On peut citer entre autres celle de Georges HABCHI *et al* [HABCHI et al. 2006], qui considère un système de production comme l'association dynamique de trois parties : Partie Pilotage, Partie flux et Partie Physique (Figure 1.3).

La partie pilotage : Représente la partie intelligente du système. Cette partie est munie des mécanismes de captage et d'acquisition d'informations, ainsi que des processus décisionnels, en lui-permettant d'évaluer et de traiter des informations et générer des actions transférées aux autres parties du système.

La partie flux : Constituée des flux de matières, entités à assembler ou à désassembler et entités à transformer. Le flux formé par ces éléments peut être sous différentes formes : pièces, matières premières, lots, séries, ...

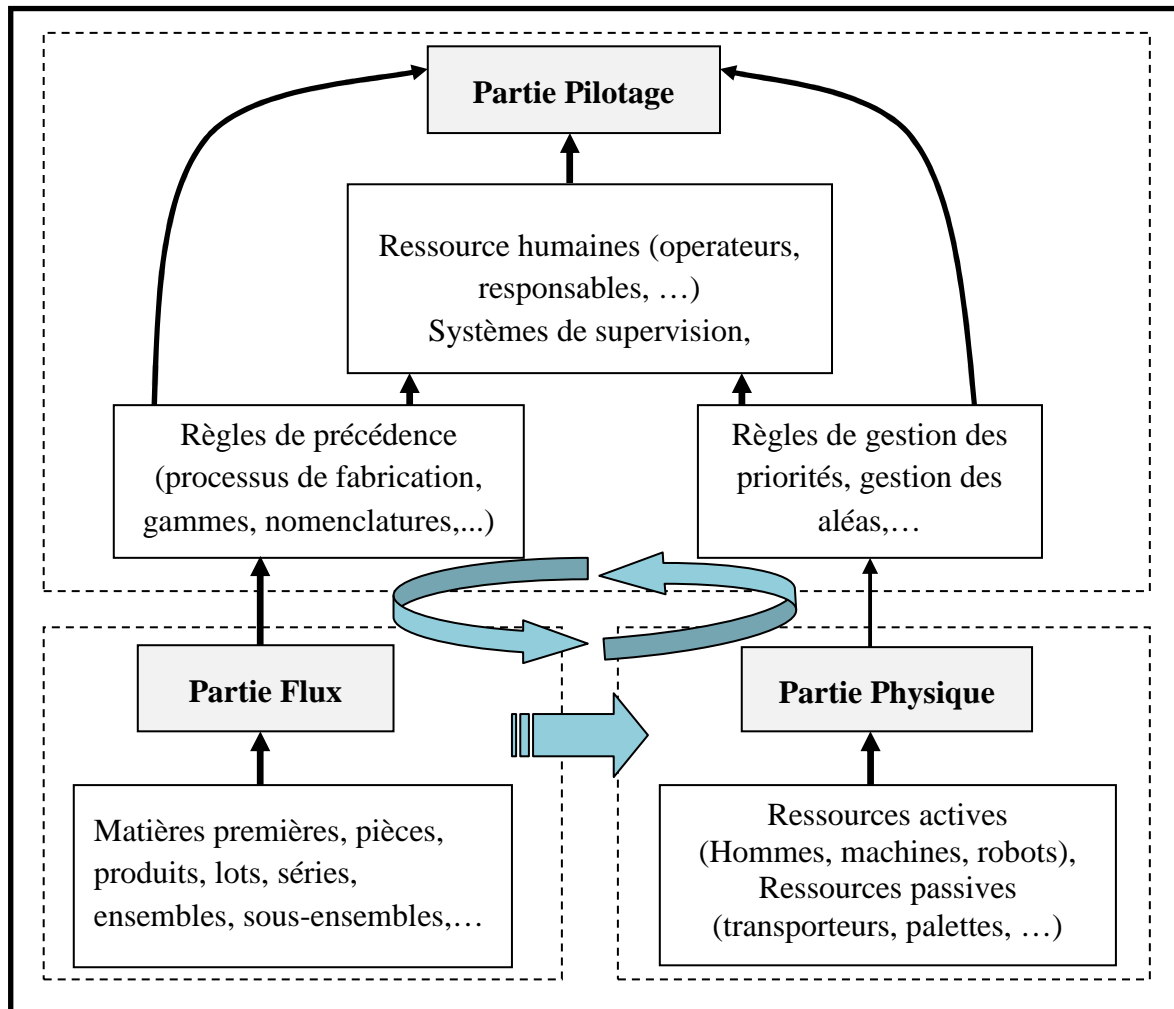


Figure 1.3. Décomposition d'un Système de Production selon G.HABCHI et al [HABCHI et al. 2006].

La partie physique : Représente la partie opérative du système, elle est chargée de réaliser les actions (générées par la partie pilotage) sur les flux de matières. Elle est constituée généralement des machines, des robots, des espaces de stockage, des ressources humaines....

3.5. Ressources d'un système de production : Pour réaliser ses différentes tâches, un système de production a besoin d'un ensemble de ressources. Ces ressources sont nécessaires

pour la fabrication, le transport et le stockage. Elles peuvent être de deux types [GRIMAUD, 1996] :

- **Ressources Actives** : Ces ressources peuvent agir, pour réaliser une activité, d'une façon autonome, c'est-à-dire que son état peut évoluer indépendamment des autres entités appartenant au même système (hommes, machines, robots, ...).
- **Ressources passives** : Ce sont des ressources qui sont utilisées par les ressources actives pour réaliser ses activités.

Dans un système de production toute ressource nécessite un dimensionnement (quantité, capacité, durée d'allocation,...) et une gestion (alimentation, acheminement des flux, transports de matières d'œuvre,...).

3.6. Flux de production : Le flux de production représente toutes sortes d'entités passives matérielles ou immatérielles circulant dans le système de production. On distingue deux types de flux [Tounsi, 2009] :

- **Flux physique** : représente la circulation des entités matérielles (palpables) de l'amont vers l'aval (matières premières, composants d'assemblages, produits finis ou semi-finis,...)
- **Flux de données** : englobe toutes les données qui circulent entre les différents acteurs du système en leur permettant de coordonner les différentes tâches et activités. Ces données sont classifiées en trois catégories :
 - *Données informationnelles* : Ce sont des données de gestion (capacités, temps, localisation,...) et des données informatiques (statiques, dynamiques et historiques).
 - *Données décisionnelles* : nécessaires à la prise de décisions à court et à long terme par les différents acteurs du système.
 - *Données métriques* : Ce sont des indicateurs et de mesures aidant, d'une part, le système de pilotage dans sa prise de décisions stratégiques, tactiques ou opérationnelles (règles et conditions de pilotage des activités...) et de mesurer les performances du système d'une autre part.

3.7. Pilotage des systèmes de production: Il existe de nombreuses définitions concernant le concept pilotage des systèmes de production. Trentesaux et al [Trentesaux et al. 2000] ont défini le pilotage par: « *Le pilotage consiste à décider dynamiquement des consignes*

pertinentes à donner à un système soumis à perturbation pour atteindre un objectif donné décrit en terme de maîtrise de performances. La notion de maîtrise intègre non seulement celle de maintien d'un niveau de performance donné, mais également celle de progrès (évolution vers un niveau de performance souhaité ou avec une amélioration continue) ».

Un système de pilotage est donc un ensemble de processus de prise de décision permettant d'atteindre un objectif global. Cet objectif ne se réduit pas à maintenir un niveau de performances mais aussi à une amélioration continue.

On peut définir aussi le pilotage des SdP comme un ensemble de techniques, méthodes et principes permettant de conduire, ordonnancer, mesurer et évaluer les performances du SdP en utilisant des données en provenance de différents acteurs du SdP. Ces données permettent à ce dernier de communiquer et mettre à jour la situation des ordres de fabrication et de contrôler les mouvements des entités circulantes dans le système.

Le système de pilotage est donc un ensemble de fonctions génériques [Mirdamadi, 2009] [Aissani, 2010] :

- **Planification** : Consiste à établir un plan prévisionnel à long et à court terme. Ce plan recouvre les trois niveaux de décision : stratégique, tactique et opérationnel (voir §3.10.3). La planification est établie en fonction des objectifs de l'entreprise et de l'état de son environnement en prenant en considération les circonstances éventuelles exigeant une adaptabilité du système (évolution des technologies, changement des demandes des clients,...).
- **Programmation** : Cette fonction part de la planification à court terme. Elle permet précisément d'interpréter les objectifs de la planification en ordres partiels de fabrication. Elle sert donc à établir un programme prévisionnel de production avec prise en compte des contraintes, des délais, des capacités, ...
- **Ordonnancement** : Permet de fournir un calendrier d'exécution des tâches à réaliser sur les ressources en fixant les dates de début et de fin d'exécution des tâches avec prise en considération des contraintes de précédences et des gammes de production dans le but d'optimiser l'utilisation des moyens de production.

L'ordonnancement ne se restreint pas à la dimension prévisionnelle (suivre la planification en off-line) mais elle présente aussi une dimension dynamique (temps

réel) pour s'adapter aux incertitudes et données perçues en temps réel (machines en pannes, matières premières livrées en retard, perturbation,...).

- **Conduite** : Correspond à la réalisation des tâches ordonnancées. La phase d'ordonnancement doit avoir un degré de souplesse et d'adaptabilité afin de permettre aux solutions du niveau conduite d'être réalisables. La conduite doit assurer les performances établies au niveau prévisionnel et doit respecter les contraintes de fabrication (qualité, maintenance préventive,...). Elle doit aussi prendre en compte les aléas possibles ainsi que les problèmes posés par les opérateurs (compétences, qualifications,...).
- **Commande** : le rôle de cette fonction est de traduire les ordres de fabrication en un ensemble d'instructions compréhensibles par le système physique. Cette fonction interagit directement avec le système physique de fabrication : les ordres interprétés (décodés) sont envoyés au système physique à l'aide des actionneurs, d'autre part le suivi et la perception de l'état d'exécution des tâches sont effectués par des capteurs.

3.8. Classification des systèmes de production :

Dans la littérature, plusieurs classifications des systèmes de productions existent. On se restreint aux deux classifications : celle définie par Giard.V [Giard, 2003] qui a classifié les SdP selon la nature et le volume de flux physique puis selon le mode de pilotage, et celle définie par Widmer [Widmer, 1991] qui a classifié les systèmes de production selon le mode de cheminement des produits (figure 1.4) :

3.8.1. Classification selon la nature et le volume de flux physique : Cette classification concerne le système physique de fabrication et fondée sur sa nature ainsi que le volume des produits fabriqués par celui-ci. On distingue trois classes de systèmes :

1. Systèmes à Flux Continu (SFC) : Dans ces systèmes, la matière circule d'une façon continue (24h/24h et 7jours/7), il n'y a pas donc un attente entre deux ressources. A cet effet l'arrêt d'une machine serait trop couteux. En général dans ces systèmes, l'automatisation est très importante, le stockage est réduit, les opérateurs sont spécialisés (avec un petit nombre). Ils concernent précisément les systèmes de production dits "*process*" où la production manipule des matières liquides ou gazeuses, par exemple le cas de l'industrie des hydrocarbures.

2. Systèmes à Flux Discret (SFD) : Les produits sont fabriqués en discontinu et peuvent être donc séparés individuellement par des stocks intermédiaires. Dans ce cas ces systèmes utilisent des zones de stockage temporaires entre deux postes de travail successifs. L'automatisation n'est pas importante et les machines sont organisées par catégories. Dans ce type de systèmes, deux classes de production peuvent être distingués :

- **Production en série :** appelée aussi production de masse. Dans ce type de cas, la production est standardisée avec un grand volume (en séries) en passant régulièrement par les mêmes ressources (machines, stocks,...). La taille de la série dépend de la demande du client et le coût (permettant de satisfaire la demande du client avec un coût minimum). Ce type de production permet au système de production de réduire les coûts de production.
- **Production unitaire :** Appelée souvent HVLV (*High-Variety, Low-Volume*). Dans ce type de système ; les clients exigent des produits différenciés (non standardisés) répondant à des besoins spécifiques et avec un taux de renouvellement rapide. La production est en petits volumes avec une grande diversité ce qui nécessite un changement fréquent des postes de travail ainsi que des réglages des machines, des stocks,...

3. Systèmes à flux Hybride (SFH) : Encore appelés systèmes à flux discontinu. Ces systèmes se situent entre les systèmes à flux continu (SFC) et les systèmes à flux discret (SFD). Deux configurations possibles peuvent être distinguées :

- Les deux types de systèmes (SFC et SFD) sont couplés dans le sens où la production est continue mais il y a conditionnement discret des produits.
- Le traitement des produits est continu mais effectué par lots. C'est-à-dire une production continue où le produit passe d'une machine à une autre et avec des quantités finies (lots).

3.8.2. Classification selon le mode de pilotage : Cette classification est liée au mode de déclenchement de la production utilisé, c'est-à-dire la stratégie de supervision du processus de production. On distingue :

1. Systèmes avec pilotage à flux tiré : Dans ces systèmes, la production n'est pas anticipée mais elle est déclenchée par la consommation de produits finis. Dans ce type deux cas de figure peuvent être distingués :

- **Production sur stock :** Dans ce cas, la production est déclenchée lorsque le produit arrive à un niveau déterminé, c'est le stock minimum maintenu ou encore appelé stock de sécurité.
- **Production sur commande :** Dans ce cas, le déclenchement de la production est appelé lorsqu'une commande est passée. Ce type est aussi appelé *zéro stock*, où le stockage des produits finis n'est pas nécessaire.

2. Systèmes avec pilotage à flux poussé : Dans ce type de systèmes, le déclenchement de la production est basé sur une prévision préalable des commandes et permettant de déterminer un planning de production, ce qui nécessite un dimensionnement afin d'absorber les incertitudes pouvant perturber la production. Dans ce type de système, c'est la disponibilité des produits qui permet de déclencher l'étape suivante de fabrication, ce qui peut engendrer des ruptures éventuelles de stock, ainsi, le produit fini est stocké avant sa livraison.

3. Systèmes avec pilotage à flux hybride : Dans certains cas et pour profiter des avantages des deux types de pilotage (à flux tiré et à flux poussé), il convient de les combiner dans un même système de production. Par exemple ; on pilote la partie amont du système par un pilotage à flux poussé et la partie aval par un pilotage à flux tiré.

3.8.3. Classification selon le mode de cheminement des produits :

1. système de production Flow Shop : Dans ce type de système, les produits circulent d'une façon unique et bien déterminée d'une machine à l'autre. C'est le cas par exemple des chaînes de production.

2. système de production Job Shop : Dans ce type de système, les produits suivent différents chemins selon la gamme de production de chaque produit. Ce type de système est appelé aussi système de production à cheminement multiple.

2. système de production Open Shop : Dans ces systèmes, aucun ordre de cheminement des produits n'est déterminé. Les produits passent d'une machine à l'autre avec une liberté.

3.9. Automatisation des systèmes de production :

3.9.1. Principe :

Les systèmes de production sont en développement rapide et continu. En effet et afin d'atteindre les objectifs visés par les entreprises (assurer la qualité des produits avec des coûts réduits, la suppression des travaux dangereux et pénibles, la réalisation des opérations qui sont impossibles à réaliser manuellement, améliorer les conditions de travail), ces systèmes ont été automatisés de façon très poussée.

En plus de ces objectifs, l'automatisation d'un système de production permet de [ROHÉE, 2008] :

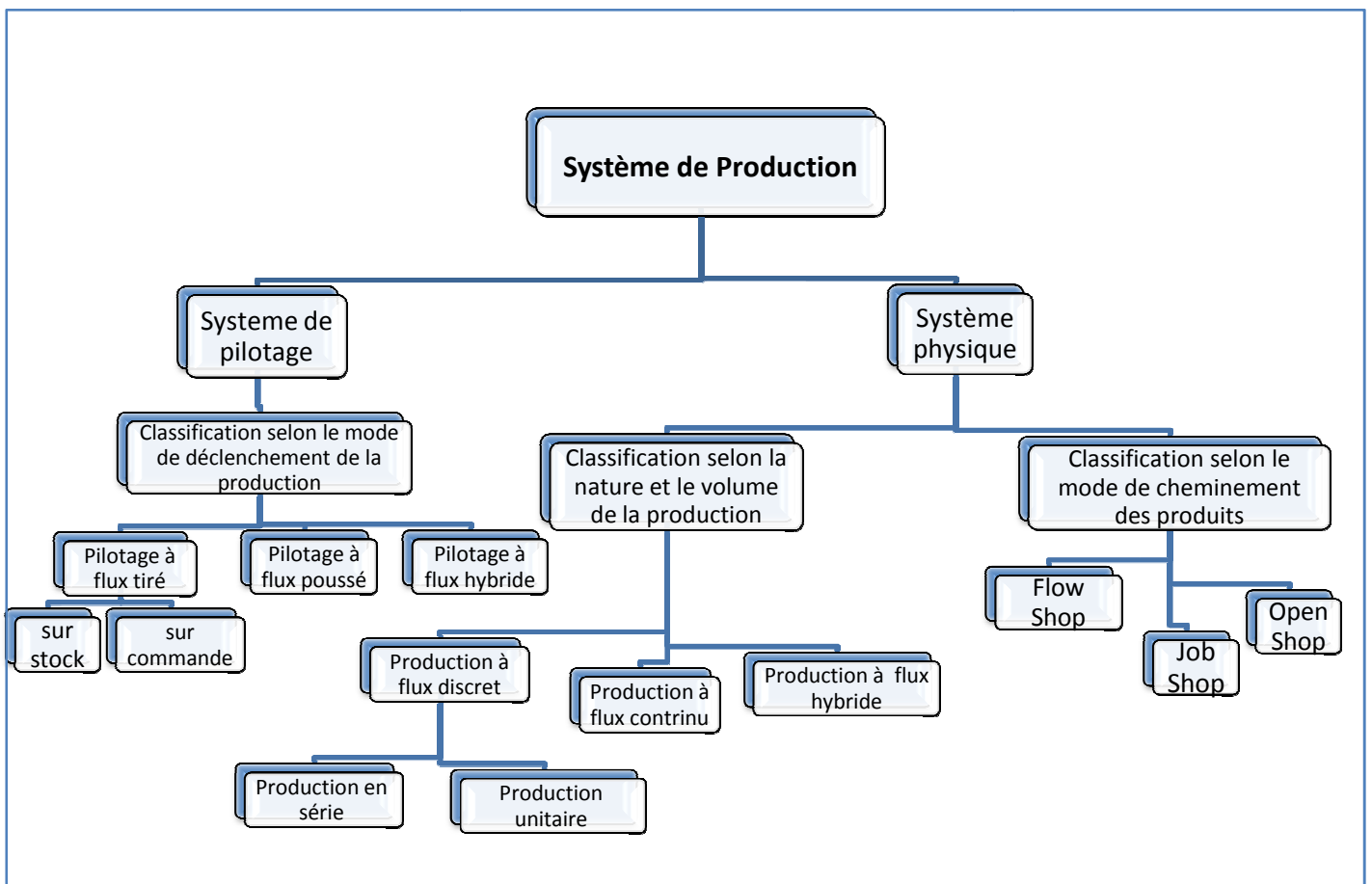


Figure 1.4. Classification des systèmes de production selon Giard [Giard, 2003][Widmer, 1991]

- Rendre son fonctionnement autonome avec un comportement systémique.
- Réduire les interventions humaines dans l'utilisation continue du système.

- Piloter la totalité du système avec un ensemble de matériel dédié.
- Améliorer la productivité et la disponibilité des produits.
- Remettre en cause le processus de production, ce qui permet de l'améliorer.

3.9.2. Structure d'un système automatisé de production : L'automatisation consiste essentiellement à séparer le système de production en deux parties fortement interactives : *Partie Commande* et *Partie Opérative* (Figure 1.5) [ROHÉE, 2008].

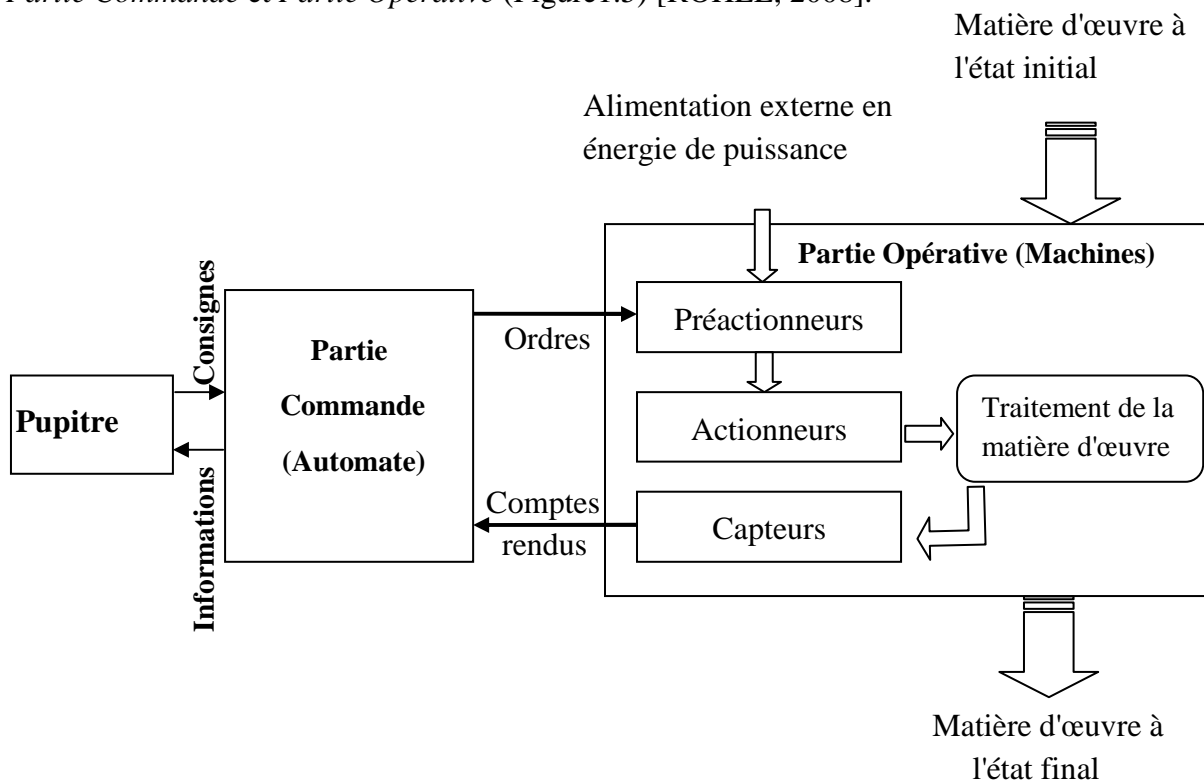


Figure 1.5. Décomposition fonctionnelle d'un système de production automatisé.

- **La partie commande :** sert à coordonner et contrôler les actions élémentaires effectuées par la partie opérative et à piloter tout le système avec un matériel dédié à la commande. D'autre part, il ya une dissociation entre la technologie utilisée pour la commande et l'énergie utilisée pour générer les actions sur la partie opérative.

Deux évolutions technologiques de base sont apparues à la commande [ROHÉE, 2008]:

- **Les composants électroniques:**

La miniaturisation des composants électroniques a permis de résoudre les problèmes liés à la complexité des systèmes. Les composants électroniques (les portes logiques, les bascules, les circuits intégrés,...) ont permis de réaliser des systèmes de commande moins coûteux.

- L'informatique industrielle :

L'informatique industrielle est une discipline qui s'appuie sur les notions de base de la logique fonctionnelle. Elle regroupe l'informatique de calcul, l'automatique et les circuits intégrés. Les systèmes automatisés de production sont pilotés actuellement par des automates programmables industriels (API) qui permettent une adaptation et une flexibilité dans la modification de comportement de ces systèmes.

Un API est composé de quatre éléments principaux :

- Une mémoire programmable permettant de recevoir et stocker les programmes écrits par un programmeur à travers une console ou un PC dans un langage spécifique à l'automate. Ce programme contrôle le déroulement du processus à automatiser par l'envoi des ordres à la partie opérative.
- Un processeur (ou plusieurs) permettant d'exécuter des opérations élémentaires et complexes. Il permet aussi d'organiser les différentes relations entre la mémoire et les interfaces d'E/S.
- Des interfaces d'E/S : Les interfaces d'entrée comportent des adresses d'entrée des capteurs de la partie opérative. Ces interfaces permettent de mesurer des grandeurs logiques, numériques et analogiques. Les interfaces de sortie comportent des adresses de sortie pour les préactionneurs de la partie opérative.
- Un ou plusieurs blocs d'alimentation.

Chaque automate possède son propre langage. Mais la plupart des constructeurs proposent actuellement des langages normalisés selon la norme CEI (Commission Electronique Internationale) [Jargot, 1999]. Cette dernière a défini cinq langages : GRAFCET, FBD, LD, ST et IL.

- La partie opérative :

La partie opérative correspond à la partie mécanisée du système automatisé. Cette partie est constituée de trois unités principales : les capteurs, les actionneurs et préactionneurs.

Les capteurs : Servent à sonder l'état du système et permettent d'observer son évolution. Ils convertissent les grandeurs physiques mesurées (vitesse, température,...) en un signal électrique de basse tension adapté à la partie commande suivant le type d'énergie utilisée.

Les actionneurs : Transforment l'énergie d'entrée (d'alimentation) en énergie de puissance utilisable par les mécanismes (constituants l'actionneur) afin d'exécuter les actions. Les actionneurs peuvent être des moteurs, des vérins,...

Les préactionneurs : Reçoivent des ordres de la partie commande. Le rôle principale d'un préactionneur est de transformer l'énergie utile (énergie de puissance) aux actionneurs sous contrôle de la partie commande.

3.9.3. Fonctionnement d'un système automatisé de production :

Pour effectuer le traitement du produit, le système effectue une boucle constituée de deux phases :

- **Chaîne d'actions** : Constituée d'un sous-ensemble de la partie opérative exécutant les actions élémentaires sur la matière d'œuvre par ordre de la partie commande. Cette chaîne est constituée des actionneurs, préactionneurs, des effecteurs et des transmetteurs de puissance.
- **Chaîne d'acquisition** : Regroupe les composants chargés de mesurer les grandeurs physiques et de l'envoi des comptes rendus sur la partie opérative à la partie commande (signaux compréhensibles par celle-ci). Cette chaîne est constituée des capteurs et des détecteurs.

3.10. Gestion de la production :

3.10.1. Définition :

La gestion de production est la fonction qui permet la réalisation des opérations de production en procédant à une organisation efficace de celle-ci [Giard, 2003]. Cette organisation permet également d'assurer une production respectant les contraintes et les objectifs visés par le système de production : coûts, délais, qualité. Elle assure un équilibre entre [Blondel, 1997] :

- Le taux d'occupation des ressources.
- Le niveau des encours (les stocks intermédiaires entre deux ressources) et des stocks.
- Les délais.

3.10.2. Fonctions assurées par la gestion de production :

En plus de l'équilibre assuré par la gestion de production cité précédemment. Celle-ci assure trois fonctions principales [Hax et Candea, 1984] : La conception, la planification et le contrôle.

- *La conception* : Cette fonction sert à prédire les caractéristiques techniques et commerciales soit pour le système de production, soit pour les produits, ce qui permet d'établir un schéma représentant le système production (capacités, localisations, moyens utilisés, ...) et les produits (les gammes, nomenclatures, ...).
- *La planification* : Cette fonction permet de coordonner entre les demandes de production et les capacités du système de production en termes de capacités, couts, délais,...
- *Le contrôle* : Le contrôle permet de comparer les résultats de production obtenus avec les objectifs attendus.

3.10.3. Typologie des décisions prises par la gestion de production :

La gestion de production s'appuie sur un ensemble d'outils d'analyse et de conception pour résoudre les problèmes de la production. Pour cela trois catégories de décisions sont prises : Stratégiques, Tactiques et Opérationnelles [Giard, 2003] :

- **Décisions stratégiques** : Correspondent aux décisions à long terme. Elles permettent d'établir une politique globale et durable du système de production. Ces décisions visent à installer des procédures de gestion de ressources. Ces ressources peuvent être :
 - *des machines* : les décisions dans ce cas sont des investissements, des radiations, des modifications de disponibilité des machines par des maintenances,...
 - *des moyens humains* : Des plans de formation du personnel sont établis, les retraites, les embauches sont aussi planifiées.
 - *des informations* : Bases de données techniques, des procédures de gestion,...
 - *des produits finis ou semi finis,...*

- **Décisions tactiques** : Correspondent aux décisions à moyen terme. Ces décisions concernent en générale la mise en œuvre des décisions prises au niveau stratégiques sur des périodes plus courtes. Ces décisions traitent souvent la planification de la production et/ou de transport.
- **Décisions opérationnelles** : Correspondent aux décisions à plus court terme, directement exécutables par le système en respectant le plus possible les plans établis au niveau tactique. Elles sont applicables pratiquement en temps réel et permettent le suivi quotidien du système de production ce qui nécessite une réactivité et une flexibilité face aux perturbations du système.

3.11. Finalités des systèmes de production :

Les systèmes de production actuels se mettent face à des contraintes et des enjeux majeurs conduisant la fonction de production vers une évolution importante [Dhouibi, 2005] (figure 1.6).

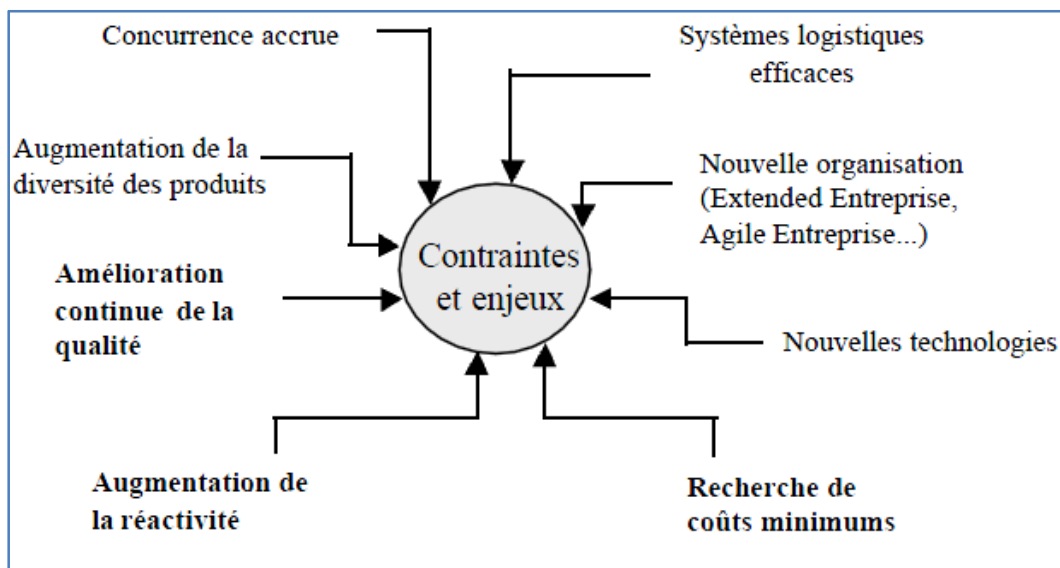


Figure 1.6. Contraintes et enjeux des systèmes de production [Dhouibi, 2005].

Avec la diversité des produits, l'accroissement de la concurrence, la mutation rapide des besoins du marché, les systèmes de production visent principalement à [M'HALLA, 2010] :

- Minimiser les coûts et les délais de production afin, d'une part, d'augmenter les bénéfices financiers, et de garantir une compétitivité permettant à l'entreprise d'accompagner ses concurrents.
- Améliorer les conditions de travail des employés en automatisant les tâches dangereuses et pénibles.
- Répondre aux demandes des clients en assurant la qualité souhaitée avec respect des délais tout au long de la production.
- Augmenter la flexibilité en s'adaptant aux changements des besoins des clients qui exigent des produits de plus en plus diversifiés, ainsi qu'aux actualités de production (évolution de la technologie, nouvelles règles d'organisation,...).
- Augmenter leur disponibilité par la diminution des pannes et des temps de maintenance, la réaction aux aléas et perturbation pouvant bloquer le processus de production.

3.12. Caractéristiques des systèmes de production :

Afin d'atteindre les objectifs cités précédemment, les systèmes de production doivent avoir des nouvelles caractéristiques à savoir [M'HALLA, 2010] : la flexibilité, la réactivité, la proactivité et la robustesse :

- **La flexibilité** : Dans la littérature, il existe plusieurs définitions de la flexibilité. Nous pouvons exprimer la flexibilité par la capacité d'adaptation du système de production à la production de nouveaux produits pour lesquels le système n'était pas étudié. Le système de production peut donc passer d'une configuration à une autre soit d'un point de vue décisionnel (adapter le système de pilotage aux différentes configurations), soit d'un point de vue physique qui correspond à la possibilité de changement des caractéristiques fonctionnelles du système pour permettre aux ressources d'effectuer des tâches différentes avec une possibilité de reprogrammation [Berruet, 1998].
- **La réactivité** : C'est l'aptitude du système de réagir dans un temps requis aux différents changements internes ou externes de son environnement [Dhouibi, 2005] par rapport au régime de fonctionnement. Ces changements peuvent provenir soit du système lui-même, soit de son environnement.

- **La proactivité** : Avec la complexité croissante actuelle des processus de production, les incertitudes de leurs états, les aléas non prévus dans leurs environnements, il est nécessaire d'avoir des systèmes de production proactifs. La proactivité est la capacité d'anticiper les différents aléas et changements d'états lors de l'élaboration de l'ordonnancement initial et de déterminer à priori les réponses adéquates.
- **La robustesse** : C'est la capacité du système de préserver certaines propriétés face à des perturbations [Collart et al., 2007]. Le système doit être donc insensible et peut garder un état d'équilibre auprès de ces perturbations.

4. Conclusion :

Les entreprises actuelles doivent faire face à une concurrence féroce et différents défis étant donné la concurrence tant locale que nationale et internationale sur le marché. Les outils et méthodes de production sont de plus en plus élaborés et mettent en œuvre des procédés et des techniques de pointe. Au niveau international, les entreprises sont pourvues de services recherche et développement (R&D) qui recrutent parmi les chercheurs les plus compétents pour mettre au point des programmes stratégiques assurant une production de grande qualité à des coûts compétitifs. Les systèmes de production ont, ainsi beaucoup évolués durant ces dernières décennies.

CHAPITRE 2

Modélisation et Simulation des systèmes complexes

1. Introduction :

Comme vu dans le chapitre précédent, un système complexe est un système composé d'un ensemble d'éléments en fortes interactions, distribués dans le temps et dans l'espace. Cette complexité pose d'énormes problèmes durant toutes les étapes du cycle de vie du système. En effet, pour analyser et pour comprendre les phénomènes mis en jeu par ces systèmes, les méthodes analytiques traditionnelles ne peuvent être utilisées. Dans ce contexte, la modélisation et la simulation offrent un moyen puissant qui permet d'appréhender un système dont la complexité est importante. Elles prennent en compte la structure et le comportement d'un système complexe ainsi que ses interactions avec l'environnement.

2. Généralités sur les modèles :

2.1. Notion de modèle :

- Pour Pavé [Pavé, 1994] : *«Un modèle est une représentation symbolique de certains aspects d'un objet ou d'un phénomène du monde réel, c'est-à-dire une expression ou une formule écrite suivant les règles du système symbolique d'où est issue cette représentation».*

- Pour Ferrier [Ferrier, 2003] : *« Un modèle est une approximation, une vue partielle plus ou moins abstraite de la réalité afin d'appréhender plus simplement, selon un point de vue et il est établi pour un objectif donné ».*

- Pour Fishwick [Fishwick, 1995] : *« Modéliser c'est décrire la réalité sous la forme d'un système dynamique à l'aide d'un langage de description à un certain niveau d'abstraction ».*

- Pour Minsky [Minsky, 1968]: *« To an observer B, an object A* is a model of an object A to the extent that B can use A* to answer questions that interest him about A».*

En guise de conclusion sur ces définitions, un modèle est une représentation simplifiée du monde réel qui permet de comprendre le comportement et le fonctionnement d'un système selon un objectif précis. Un système est donc représenté par un modèle exprimé dans un langage ou formalisme bien spécifié et permettant de

comprendre son comportement. Un modèle peut être aussi considéré comme une représentation rationnelle du comportement d'un système réel avec une précision satisfaisante. C'est-à-dire, d'une part, il reflète fidèlement le comportement du système représenté, et il est capable de prédire le comportement futur du système s'il est subit à un certain stimuli d'autre part.

2.2. Intérêt d'un modèle :

Un modèle est une image simplifiée utile représentant un aspect bien défini du système. Il permet de décrire la structure et le comportement d'un système dans le but, soit pour simuler ce comportement pour comparer les différents scénarios, soit pour évaluer et améliorer ses performances [Vernadat, 1999]. Pour un système réel, un modèle permet de :

- Décrire et comprendre son comportement.
- Construire des théories ou des hypothèses sur le système observé.
- Evaluer ses performances (coûts, délai, capacités, vitesse, ...).
- Analyser le système et optimiser ses performances.
- Prévoir ses évolutions (prédire le futur comportement).

2.3. Stratégie de choix d'un modèle :

Afin de modéliser un système, ceci doit être appréhendé selon une vue externe et une vue interne. La vue externe représente les frontières du système ainsi que ses finalités dans son environnement. La vue interne représente les composants du système ainsi que les interactions entre eux [Sperandio, 2005].

La vue interne d'un système complexe est modélisée selon trois pôles [Le Moigne, 1994] :

- *Un pôle fonctionnel* : Décrit les objectifs et les fonctions attendus du système (ce que le système fait).
- *Un pôle ontologique* : Décrit la structure du système : ses propriétés, ses composants et les interactions entre ces composants (ce que le système est).
- *Un pôle génétique* : qui s'occupe de l'évolution dans le temps du système et de sa prédiction lorsqu'il est soumis à des changements internes et externes (ce que le système devient).

La modélisation d'un système est un processus assez complexe qui s'articule sur les points suivants [Sperandio, 2005]. (Fig 2.1) :

- Déterminer les limites du système. Il y'a lieu de choisir un point de vue (ou d'observation) et délimiter, ainsi le système réel avec son environnement.
- Déterminer les objectifs et buts du système dans le cadre de l'observation définie.
- Déterminer les interactions du système avec son environnement ceci a pour conséquence de déterminer les relations et les influences entre l'environnement et le système.
- Définir la vue interne fonctionnelle qui représente les processus mis en œuvre par le système.
- Déterminer la vue interne organique qui permet de mettre en relief les différents constituants du système ainsi que leurs interactions.
- Déterminer les comportements et évolution du système dans son environnement. On définira, ainsi, la dynamique du système.

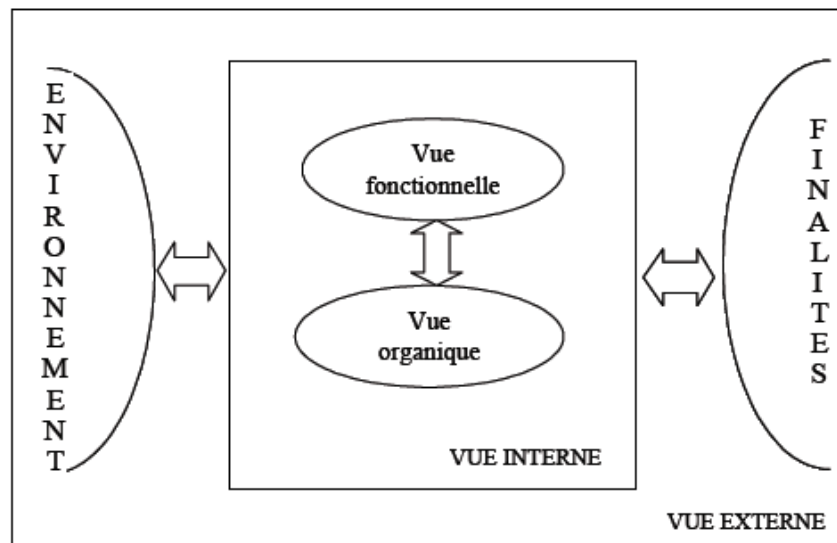


Figure 2.1. Stratégie de choix d'un modèle [Sperandio, 2005].

2.4. Modèle physique/modèle mathématique :

Modèle physique : C'est une représentation du système réel par une forme de maquette sur laquelle on effectue les expérimentations. Ces modèles sont coûteux et ils ne sont utilisés que pour certains phénomènes tels que les expériences physiques où les phénomènes étudiés sont approchés à une structure plus simple.

Modèle mathématique : Appelé aussi modèle abstrait. Ces modèles sont des modèles numériques utilisant des outils mathématiques (équations, variables,...) utilisés depuis longtemps. Ils peuvent être implémentés sur machines, de ce fait ils ont connu leur grand succès avec l'apparition des ordinateurs qui sont caractérisés par une grande capacité de représentation de l'information et de calcul.

2.5. Classification des modèles :

La classification des modèles dépend toujours de la nature du système étudié. Plusieurs classifications peuvent être envisagées :

2.5.1. Classification selon le déterminisme :

- **Modèle déterministe** : C'est un modèle dont l'état futur du système est déterminé en fonction de l'état actuel d'une façon unique (régulièrement).
- **Modèle stochastique** : C'est un modèle contenant une ou plusieurs variables aléatoires. C'est-à-dire le phénomène représenté par ce modèle fait intervenir la loi probabiliste et donc prend en compte la probabilité dans l'apparition de ses résultats. En d'autre terme, le système passe d'un état à un autre selon la probabilité de transition entre états.

2.5.2. Classification selon la continuité:

- **Modèle continu** : Un modèle continu utilise en générale des équations pour déterminer l'état du système qui est délivré chaque instant. Le changement d'états se fait donc d'une façon continue au cours du temps. Les valeurs des variables d'état du système sont déterminées régulièrement et infiniment par un pas d'horloge spécifié par ces équations.
- **Modèle discret** : Dans ces modèles, le changement d'état du système n'est effectué que pendant des intervalles de temps appelés pas de temps et symbolisés généralement par Δt . Dans ce cas, les variables d'état du système modélisé ne peuvent pas être déterminées à chaque instant, mais à des moments bien déterminés. Ce type de modèle implique une fonction permettant de calculer l'état du système à l'instant $t + \Delta t$ à partir de son état à l'instant t .

- **Modèle à événement discret** : C'est la plus grande classe des modèles. Les variables d'états du système modélisé changent d'une manière discrète. Dans ce type de modèle, le changement d'état du système ne survient qu'à l'arrivée d'un événement tels que le début ou la fin d'une opération, la libération d'une ressource, la mise en attente d'une pièce dans un stock,...

2.5.3. Classification selon le dynamisme :

- **Modèle statique** : C'est un modèle qui ne prend pas en compte les conséquences d'évolution de l'état du système dans le temps.
- **Modèle dynamique** : C'est un modèle qui représente l'évolution de l'état du système au cours du temps.

2.6. Natures des variables d'un modèle :

L'état du système est représenté par les valeurs prises par un ensemble de variables. Ces variables peuvent être de deux types [Cellier, 1991] :

- **Variables quantitatives** : Ce sont des variables dont la valeur est représentée par une grandeur physique exprimée numériquement, par exemple la distance, l'âge, ... Ces variables sont exprimées en général par des nombres réels.
- **Variables qualitatives** : Ce sont des variables dont les valeurs sont représentées par des modalités ou des noms qui n'ont pas un sens quantitatif. Dans ce cas, l'ordre n'est pas précis et le domaine de valeurs qu'appartiennent ces variables est fini, par exemple : $D = \{\text{'froid', 'chaud', 'très froid', 'très chaud', 'tiède'}\}$. Le comportement d'un système dynamique dans ce cas est une description des valeurs de ses variables au cours du temps.

3. Modélisation :

3.1. Définition : Le Moigne [Le Moigne, 1990] a défini la modélisation comme *«une élaboration et construction intentionnelle, par composition de symboles, de modèles susceptibles de rendre intelligible un phénomène perçu complexe et d'amplifier le raisonnement de l'acteur projetant une intention délibérée au sein de phénomène ; raisonnement visant notamment à anticiper les conséquences de ces projets d'actions possibles»*.

La modélisation peut être définie aussi par la construction d'un modèle à partir d'un système réel pour en expliquer la structure ou le fonctionnement ou pour analyser son comportement [Vernadat, 1999]. Donc, modéliser consiste à construire un modèle à partir d'un système réel et tester ensuite sa validité par rapport à ce système.

L'activité de modélisation implique : un schéma d'expérimentation, un ou plusieurs paradigmes et une méthodologie (Figure 2.2) [Quesnel, 2006].

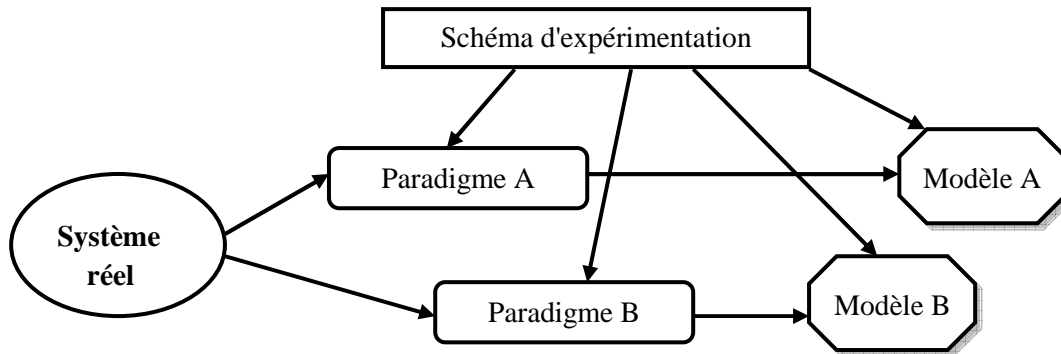


Figure 2.2. L'activité de modélisation [Quesnel, 2006].

3.2. Schéma d'expérimentation :

« L'expérimentation est un processus par lequel on récolte des données sur un système en agissant sur ses entrées » [Cellier, 1991]. Un schéma d'expérimentation est l'ensemble des conditions ou situations sous lesquelles l'expérimentation sur un système se déroule (des variables observées, des conditions d'initialisation, des conditions d'arrêt,...).

3.3. Paradigme de modélisation :

C'est une manière de penser, une manière de voir les choses, une façon de représenter le monde réel par un ensemble de concepts, de techniques, d'outils, de méthodes ainsi que des hypothèses caractérisant une activité de modélisation [Kuhn, 1972]. Par exemple le paradigme Objet considère toute entité du monde réel comme un objet. Ce dernier est caractérisé par les concepts d'encapsulation, d'héritage, de polymorphisme, de surcharge, ...

3.4. Formalisme de modélisation :

Un formalisme de modélisation est un *outil* permettant de représenter le changement d'état d'un système en fonction de son état antérieur. Il permet entre autre d'exprimer un paradigme de modélisation. Par exemple : on peut exprimer le paradigme des systèmes multi agents par le formalisme UML (Unified Modeling Language).

3.5. Modélisation multi-formalismes : Dans le processus de modélisation, le choix d'un formalisme donné dépend de plusieurs facteurs : Le niveau d'abstraction du modèle en fonction de son application et du niveau de connaissances, la disponibilité des données et des analyseurs concernant ce modèle ainsi que des objectifs de modélisation [Kerkouche, 2011].

A cet effet, la modélisation multi-formalismes (qui consiste à utiliser plusieurs formalismes et donc plusieurs modèles pour un système) permet de décrire les différents aspects et composants du système en utilisant : plusieurs niveaux d'explication, différentes théories et différents langages et formalismes dédiés [Milner, 1993].

La modélisation multi-formalismes permet, donc, de combiner et d'intégrer les différents formalismes pour générer d'autres modèles [Kerkouche, 2011], ce qui permet donc d'exploiter les apports des formalismes intégrés et de pallier aux problèmes posés par les limites de chaque formalisme utilisé individuellement.

3.6. Multi-modélisation : Ce concept a été développé par Orën T.I [Orën, 1991] et ensuite étendu par Fishwick [Fishwick, 1995]. La multi-modélisation consiste en une composition récursive de différents modèles qui peuvent être de différents formalismes et/ou paradigmes. Elle rassemble un ensemble de concepts, outils et techniques pour construire un multi-modèle. Dans ce cas, un problème principal se pose : comment coupler les modèles hétérogènes (hétérogénéité de formalismes et hétérogénéité de paradigmes).

La multi-modélisation permet principalement d'augmenter la puissance descriptive du modèle. Elle présente [Filippi, 2003] :

- *plusieurs niveaux de description* en généralisant/spécialisant un système et ses parties dans chaque niveau. Le plus haut niveau représente un seul composant

(boîte noire). Le plus bas niveau constitue plus de composants, et plus le niveau est élevé, moins il y a de composants.



- *plusieurs niveaux d'abstraction* en généralisant/spécialisant un système et ses éléments à chaque niveau. Le multi-modèle constitue donc une hiérarchie d'abstraction où les plus hauts niveaux constituent des entités inhibant les propriétés élémentaires et présentant donc moins d'interactions entre les composants du système. Les bas niveaux disposent de plus de propriétés élémentaires. Le passage du plus haut niveau d'abstraction vers le plus bas niveau se fait par des raffinements itératifs des propriétés du système et les informations concernant ces propriétés (l'état et le comportement du système) sont passées à travers les niveaux.


3.7. Quelques outils, formalismes et approches de modélisation :

3.7.1. Les réseaux de Pétri :

Les Réseaux de Pétri définis pour la première fois par le mathématicien Allemand *Carl Adam Pétri* en 1962 sont des moyens de modélisation des comportements des systèmes dynamiques à événement discret. Ils permettent de décrire les relations entre des conditions et des événements.

Définition d'un Réseau de Pétri : Un Réseau de Pétri est un graphe orienté composé de deux types de sommets (figure 2.3) :

- Les places : 
- Les transitions : 

Ces deux types de sommets sont reliés par des arcs orientés . Un arc relie une place à une transition ou une transition à une place. Les places et les transitions peuvent être nommées, un exemple d'un réseau de pétri est représenté dans la figure 2.1, les places sont nommées (P1, P2, P3, P4), les transitions sont nommées (T1, T2).

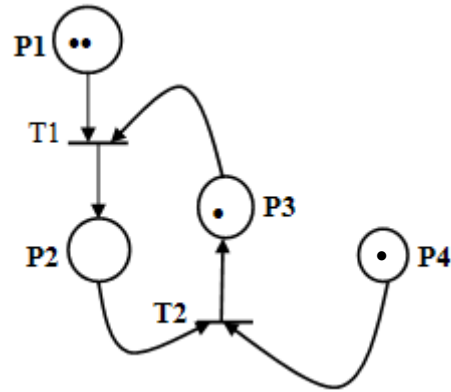


Figure 2.3 Exemple d'un Réseau de Pétri

D'une façon formelle un réseau de Pétri est un quintuplet $\langle P, T, I, O, M_0 \rangle$, où

- P est un ensemble fini non vide de places.
- T est un ensemble fini non vide de transitions.
- $P \cap T = \emptyset$.
- $I(T_i)$ est l'ensemble de places en entrée de la transition i .
- $O(T_i)$ est l'ensemble de places en sortie de la transition i .
- M_0 est le marquage initial.

Notions de bases :

- *Notion de marquage*: à tout instant, chaque place peut contenir un nombre entier positif de marques ou jetons qui représentent la valeur de la variable d'état de chaque place. On appelle ce nombre **Marquage**. Le marquage peut être représenté par un vecteur du nombre de marques de chaque place.
- *Marquage initial* : C'est le marquage à l'instant t_0 .
- *Franchissement ou Tir d'une transition* : Consiste à retirer (si la transition est franchissable) une marque de chaque place de l'entrée de cette transition et d'ajouter une marque à chaque place de sortie.
- *Séquence de Franchissement* : Une suite de franchissements successives permettant de faire passer le système d'un état à un autre (d'un marquage à un autre).

Propriétés des Réseaux de Pétri :

- *Réseau de Pétri avec Conflit* : Un réseau de Pétri est avec conflit si et seulement s'il existe au moins une place avec deux transitions en sortie.

- *Réseau de Pétri à choix libre* : C'est un réseau de Pétri où les transitions de sortie de tout conflit n'admettent qu'une seule place en sortie.
- *Réseau de Pétri à choix libre étendu* : C'est un réseau de Pétri dont les transitions de sortie de chaque conflit admettent les mêmes places en entrée.
- *Réseau de Pétri simple* : Un réseau de Pétri est dit simple si toutes ses transitions interviennent au maximum dans un seul conflit.
- *Réseau de Pétri pur* : Un réseau de Pétri est dit pur si et seulement s'il n'existe pas de transition ayant une place d'entrée qui est en même temps une place de sortie.
- *Réseau de Pétri vivant* : C'est un réseau de Pétri où, à tout instant il existe une transition franchissable.
- *Réseau de Pétri propre* : C'est un réseau de Pétri où, à tout instant il existe une séquence de franchissements conduisant au marquage initial M_0 .
- *Réseau de Pétri borné* : C'est un réseau de Pétri où, à tout instant la somme des marques est limitée par un nombre maximal.
- *Réseau de Pétri conforme* : c'est un Réseau de Pétri borné avec le nombre maximal = 1.

Éléments de modélisation par réseaux de Pétri : Dans n'importe quel système les réseaux de Pétri permettent de modéliser les éléments suivants :

- *Parallélisme* : C'est le cas où plusieurs processus évoluent simultanément dans le système.
- *Synchronisation mutuelle* : ou rendez-vous, permet de synchroniser l'exécution de plusieurs opérations de différents processus.
- *Sémaphore* : étant donné deux processus, les opérations du deuxième processus ne peuvent se poursuivre que si le premier a atteint un certain niveau dans la succession des opérations.

Les réseaux de Pétri peuvent aussi modéliser : le partage de ressources, la mémorisation, la lecture, les ressources à capacités limitées,...

Extension des réseaux de Pétri :

Les réseaux de Pétri classiques ont marqué plusieurs faiblesses en terme de puissance de spécification telle que la distinction entre jetons (types de jetons). Les conditions exprimées par ceux-ci ne sont pas riches, l'aspect temporel n'est pas exprimé,... De ce fait plusieurs extensions ont été apportées. Les extensions les plus

connues sont les réseaux de Pétri colorés, les réseaux de Pétri temporisés, les réseaux de Pétri temporels, les réseaux de Pétri stochastiques, les réseaux de Pétri algébriques, les réseaux de Pétri Prédicat/transition,... etc. Dans cette partie, nous allons donner un aperçu sur les réseaux de Pétri colorés et les réseaux de Pétri temporisés.

- **Les réseaux de Pétri colorés :**

L'objectif des réseaux de Pétri colorés [Jensen, 1992] est de distinguer dans une même place deux marques de différents types. A chaque place et à chaque transition, on associe un domaine de couleurs.

Pour les places : chaque jeton dans une place, est coloré par une des couleurs du domaine de cette place.

Pour les transitions : chaque couleur d'une transition représente une manière différente de franchissement de la transition.

La valuation d'un arc est une fonction dont l'ensemble de départ est le domaine de la transition adjacente et l'ensemble d'arrivé est le multi-ensemble associé au domaine de la place adjacente.

Plus formellement un Réseau de Pétri coloré est un sextuple :

CPN = (P, T, Pré, Post, C_T, C_P) où :

- P : Ensemble fini de places.
- T : Ensemble fini de transitions ($P \cap T = \emptyset$).
- C : Ensemble fini de couleurs.
- Pré : Fonction de pré-condition, $\text{Pré} : P \times T \times C \longrightarrow \mathcal{M}(C)$.
- Post : Fonction de post-condition, $\text{Post} : T \times P \times C \longrightarrow \mathcal{M}(C)$.
- C_T : $T \longrightarrow \mathcal{P}(C)$ associe à une transition un ensemble de couleurs.
- C_P : $P \longrightarrow \mathcal{P}(C)$ associe à une place un ensemble de couleurs.

$$\begin{aligned} \text{Marquage : } M : P &\longrightarrow \mathcal{M}(C) \\ P &\longrightarrow M(p) \in \mathcal{M}(C_P(p)) \end{aligned}$$

Franchissement d'une transition : Une transition T est franchie pour un couleur $c(c \in C_T)$ à partir du marquage initial M_0 si et seulement si :

$$\forall p \in \bullet T, \forall c' \in C_P(p), M(p)(c') \geq \text{Pré}(p, T, c)(c')$$

Autrement dit : une transition T est franchie pour un couleur $c(c \in C_T)$ si et seulement si pour toute place P en entrée de T , pour toute couleur de la fonction Pré de la couleur c (on note c'), le marquage de c' dans la place P est supérieur ou égal au marquage de c' de la fonction Pré de c .

Après franchissement de T pour la couleur c , on atteint le marquage M_1 tel que :

$$\forall p \in T^\bullet, \forall c' \in C_P(p), M_1(p)(c') = M_0(p)(c') - \text{Pré}(p, T, c)(c') + \text{Post}(T, p, c)(c')$$

Autrement dit on retranche un nombre égal à Pré de la couleur c pour toute place en entrée de T et on ajoute un nombre égal à Post à toute place en sortie de T .

- **Les réseaux de Pétri temporisés :**

Les réseaux de Pétri classiques encore appelés Réseaux de Pétri Autonomes n'expriment pas l'aspect temporel. Cependant, l'évolution de certains systèmes (essentiellement les systèmes dynamiques à événements discrets) dépend du temps. Par exemple dans le cas d'évaluer le temps d'exécution d'une série d'opérations dans un atelier de production en fonction de la capacité des machines [David et Alla, 1992], les réseaux de Pétri autonomes ne peuvent être utilisés, ce qui a donné naissance aux Réseaux de Pétri Temporisés.

Définition formelle d'un réseau de Pétri temporisé :

Un réseau de Pétri temporisé est un couple $\langle R, \text{Tempo} \rangle$. Tel que :

R : réseau de Pétri autonome.

Tempo : une application $P \longrightarrow Q^+$ (P est l'ensemble de places. Q^+ est l'ensemble des nombres rationnels positifs ou nuls) avec :

$\text{Tempo}(P_i) = d_i$: durée d'indisponibilité de la marque pour la validation de transition. C'est-à-dire dans la place P_i , les marques doivent rester au moins la durée d_i pour que les transitions soient validées.

Exemple : Soit une unité de production pétrochimique. Cette unité reçoit en entrée un mélange **GPL** (Gaz de Pétrole Liquéfié). Le GPL est ensuite soumis à une température de 118° et une pression de 12.7 Kg/cm^2 pour une durée de 55 minutes. Après cette durée le GPL sera séparé en deux gaz : C_3 (Propane) et C_4 (Butane). La figure 2.4 représente le réseau de Pétri associé.

P1 : Conditions appliquées au GPL ($T = 118^\circ$, $P = 12.7 \text{ Kg/cm}^2$).

P2 : Le GPL entré

P3 : Le GPL soumis aux conditions (température et pression).

P4 et P5 : représentent respectivement C_3 et C_4

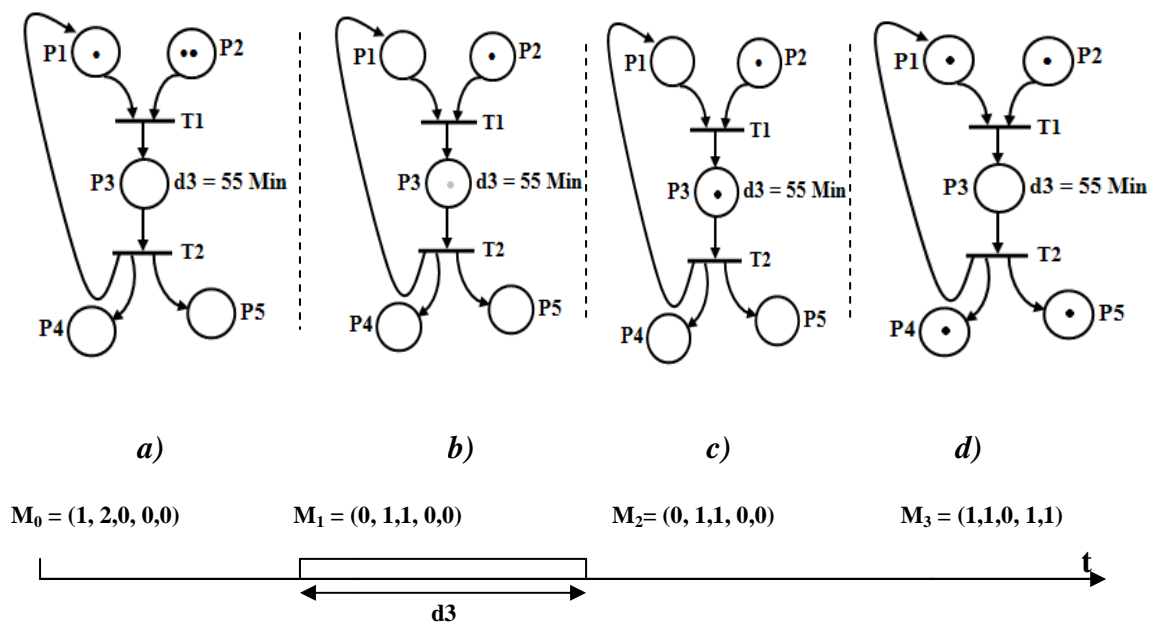


Figure 2.4. Réseau de Pétri temporisé d'une colonne de séparation de gaz

A l'instant $t = 0$, la transition T1 est validée (fig2.4.a). Elle est alors immédiatement franchie (figure2.4.b), la marque dans la place P3 représentée par une couleur grise signifie qu'elle est indisponible. Après écoulement de la durée $d_3=55 \text{ Min}$ (figure 2.4.c), la transition est validée et donc franchie (figure 2.4.d).

Dans cet exemple, on a considéré que si une transition est validée, elle est immédiatement franchie. On parle dans ce cas de *fonctionnement à vitesse maximale*.

Pour un réseau de Pétri temporisé borné, et pour tout marquage initial, le fonctionnement à vitesse maximale conduit - au bout d'un temps fini - à un fonctionnement périodique. Lorsque le fonctionnement périodique est établi, on appelle *fréquence de franchissement* d'une transition T_i le temps moyen de franchissement de cette transition par unité de temps.

3.7.2. La méthode SADT (Structured Analysis and Design Technic) :

La méthode SADT est une méthode d'analyse descendante hiérarchique développée aux USA par Doug Ross en 1977 et introduite en Europe à partir de 1982 par Michel Galiner.

En plus de la description des tâches d'un projet et leurs interactions, la méthode SADT permet de décrire le système que le projet vise à étudier. C'est une méthode fondée sur le formalisme graphique et textuel permettant d'une part de modéliser le problème (informatique, automatique,...) avant de chercher à en extraire une solution, et d'autre part d'assurer une communication efficace entre les différents intervenants du système.

Principe de la méthode :

Son principe est de représenter une image d'un système en utilisant une technique d'analyse structurée permettant d'identifier la fonction du système, les éléments qui le constituent (composants, sous-systèmes,..) ainsi que les données d'entrée ou de contrôle qui provoquent la mise en œuvre de la fonction. Elle se base sur les deux principes suivants :

- *Procéder par une analyse descendante (par niveaux)* : le premier niveau étant en général très abstrait, et ensuite on affine successivement les niveaux.
- *Délimiter le cadre d'analyse* : préciser le contexte d'analyse (limite du système) en décrivant un point de vue objectif.

La méthode SADT repose sur les concepts fondamentaux suivants :

- SADT construit un modèle pour analyser le système dans un contexte particulier afin d'en exprimer une compréhension complète.

- Analyse descendante, hiérarchique, modulaire, structurée, d'un système.
- SADT décrit seulement le quoi (et non le comment).
- SADT modélise d'une part les contrôles (événements, activités,...) et d'autre part les données, objets,... Le lien entre ces deux modèles est effectué par le modèle complet.
- SADT utilise un langage semi formelle (graphique + texte).
- SADT favorise le travail en équipe.
- SADT garde trace de tous les choix effectués pendant l'analyse, les documents sont ensuite accessibles et peuvent être revus ou corrigés.

Le modèle SADT : un modèle SADT est un ensemble de diagrammes ordonnés hiérarchiquement, le premier diagramme (le plus général) décrit le contexte du système à analyser, ce dernier est représenté par une boîte noire avec les sources et les destinataires des informations qui décrivent les interfaces avec l'extérieur.

Un diagramme SADT peut être un *Actigramme* ou un *Datagramme*.

- **Actigramme :** Manipule les données par des directives de contrôle, il est identifié par des *verbes d'action*. Il génère des données en sortie en manipulant des données en entrée (Figure 2.5). Les données de contrôle permettent de déclencher ou modifier la réalisation d'une fonction. Les supports d'activité sont des éléments physiques ou technologiques qui réalisent la fonction.

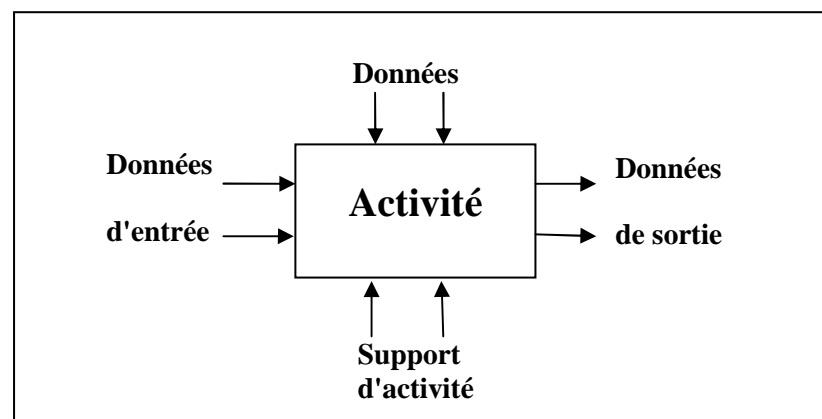


Figure 2.5 Actigramme

- **Datagramme** : C'est une boîte de données (Figure 2.6). Les entrées sont des actions produisant les données de la boîte. Les sorties représentent les actions qui utilisent les données.

- **Liste hiérarchique et numérotation des diagrammes :**

Au niveau inférieur, on représente seulement la boîte principale appelé *boite 0* par un diagramme "-0". Cette boîte est décomposée elle-même en un diagramme de niveau inférieur appelé "0". Cette boîte est composée aussi en un diagramme de niveaux inférieur "0" contenant les boîtes 1, 2, 3,... Chaque boîte *i* est décomposée en un diagramme *i1*, *i2*, *i3*,... la figure 2.7 représente la hiérarchie des niveaux.

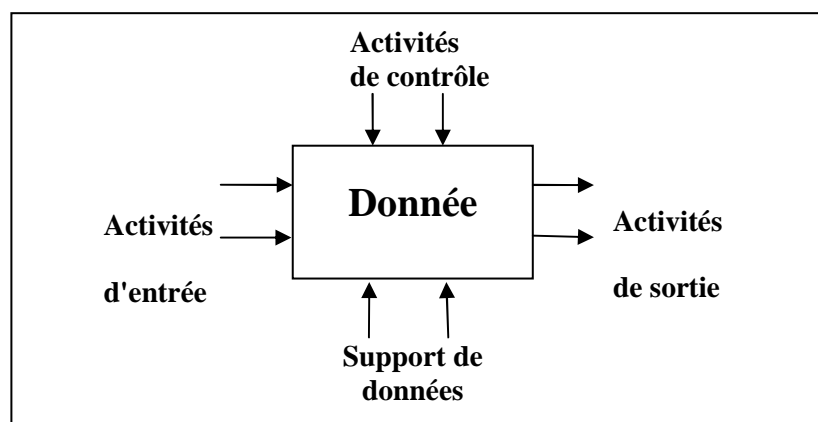


Figure 2.6. Datagramme

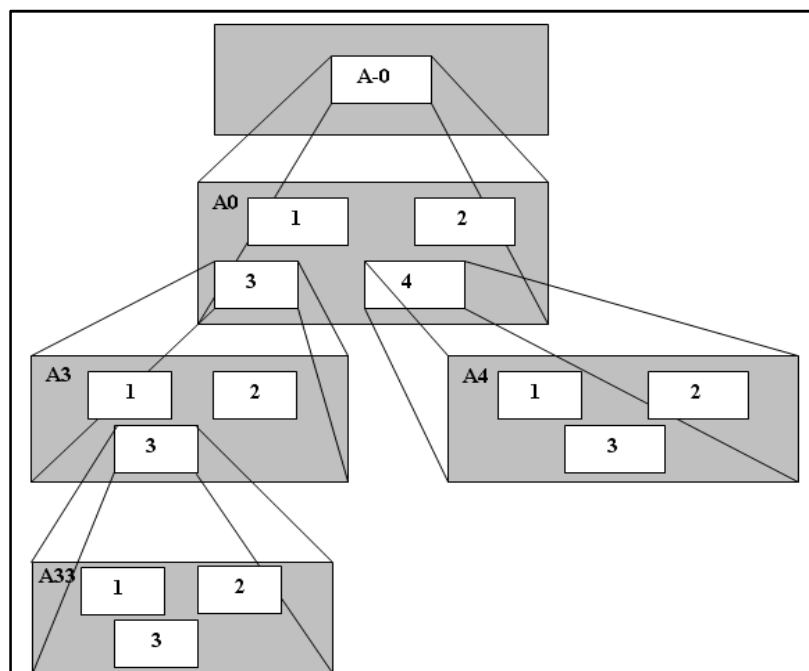


Figure 2.7 Hiérarchie des niveaux SADT

3.7.3. La méthode SA-RT (Structured Analysis for Real Time) :

C'est une stratégie de spécification des systèmes temps réels introduite par D.J. Hatley, I.A. Pirbhai. Masson à la fin des années 80.

Systèmes temps réel : Un système temps réel est un système informatique connecté à un procédé extérieur, l'évolution dynamique de ce dernier permet de contrôler le fonctionnement et le comportement du système. La correction d'un système temps réel dépend principalement du temps auquel les résultats de calcul sont produits, il doit donc satisfaire des contraintes temporelles.

Parmi les caractéristiques d'un système temps réel on peut citer :

- *Taille et complexité* : l'environnement extérieur avec lequel le système temps réel interagit est en général complexe et en évolution.
- *Certification* : le fonctionnement doit être conforme aux spécifications.
- *Simulation et prototypage* : vérification à priori du système.

Principe de la méthode SA-RT : Cette méthode intègre trois aspects essentiels :

- *L'aspect fonctionnel* : les données transférées par le système ainsi que le processus de transformation sont représentés.

- *L'aspect événementiel* : l'évolution du système conditionné par des événements est représentée, on représente aussi la logique de contrôle produisant des événements et des actions en fonction des événements en entrée qui changent l'état du système.

- *L'aspect informationnel (données)* : on spécifie les données sur les flots (cheminement des données) ou dans le stockage (mémorisation des données).

Organisation de la méthode SA-RT : la méthode est organisée en une structure descendante hiérarchique mettant en œuvre les aspects fonctionnels et comportementaux. Elle est représentée par les composants suivants :

- *Diagramme de contexte* : Décrit l'environnement ainsi que le flot de données.

- *Diagramme préliminaire* : C'est le premier diagramme de données (premier niveau).

- *Diagramme de décomposition* : Diagramme de flot de données décrivant les processus fonctionnels non primitifs.
- *Spécifications des processus fonctionnels primitifs* : description textuelle des fonctions réalisées par les processus fonctionnels non primitifs.
- *Dictionnaire des données*: liste exhaustive des données et des évènements utilisés dans la spécification.

3.7.4. Les automates d'état finis :

Un automate d'états finis est une machine ayant un état courant et d'autres états peuvent être passés avec l'évolution du système (à l'arrivée des signaux ou évènements). Le changement d'état est appelé "transition".

Un état est une situation particulière du système, c'est une abstraction qui permet de cacher les détails d'une modélisation. Les transitions sont étiquetées, chaque étiquette peut exprimer :

- Un évènement (un signal) : permet de franchir une transition
- Une garde : une condition nécessaire selon laquelle une transition peut être franchie.
- Des actions : effectuées par l'automate avant de changer son état.

Automate déterministe: pour chaque couple (état, étiquette), le choix de la transition à franchir est unique.

Automate non déterministe: s'il existe au moins un couple (état, étiquette), mène à deux états différents.

3.7.5. Les équations différentielles :

Les équations différentielles mettent en jeu une relation entre une fonction et ses dérivées successives, c'est-à-dire la variation d'une fonction à l'aide de sa dérivée. Elles permettent de décrire l'évolution de certains phénomènes et sont largement utilisées dans le domaine scientifique par exemple la chute d'un corps (mécanique), la radioactivité (physique nucléaire), la croissance végétale (systèmes naturels), les lois de Kirchhoff (électricité).

Les équations différentielles présentent des avantages principaux à savoir leur formalisation qui leur permet d'être compréhensibles, ainsi un système d'équation différentielle permet de décrire plusieurs types d'interactions entre plusieurs ensembles d'éléments.

3.7.6. Le formalisme DEVS (Discret Event system Specification) :

Ce formalisme a été mis au point par le professeur B.P. Zeigler en 1976 [Zeigler, 1976]. C'est un formalisme abstrait utilisé pour modéliser les systèmes à événements discrets. Il définit deux types de modèles : les modèles atomiques et les modèle couplés.

Les modèles atomiques décrivent les entités simples (composants de base d'un système). Ils sont représentés sous la forme :

Atomic_DEVS = (X, Y, S, δ_{ext} , δ_{int} , δ_{con} , λ , ta) où :

- X : l'ensemble des événements d'entrée.
- Y : l'ensemble des événements de sortie.
- S : l'ensemble des états du système.
- δ_{ext} : la fonction de transition externe.
- δ_{int} : la fonction de transition interne.
- δ_{con} : la fonction de transition conflit.
- λ : la fonction de sortie.
- ta : la fonction d'avancement du temps (durée de vie dans un état).

Le modèle atomique permet de décrire le comportement basique d'une entité simple. Le principe de base de ce modèle est le suivant :

- A un instant donné le système est dans un état, il doit rester dans cet état pour une durée déterminée si aucun événement extérieur ne le perturbe.
- Lorsque la durée est écoulée sans qu'aucun événement externe ne survienne le système calcule l'ensemble des événements de sortie en fonction de l'état initial et effectue une transition vers un autre état dit interne.
- Si un événement externe survient avant le découlement de la durée, le système effectue une transition vers un état dit externe.

- Dans les deux cas le système devient dans un autre état et y reste pour une autre durée définie, et le même algorithme recommence.

Pour modéliser les entités complexes on doit passer à un modèle couplé qui une interconnexion des composants. Ces derniers peuvent être des modèles atomiques comme ils peuvent être des modèles couplés. Un modèle couplé est représenté sous la forme :

Coupled_DEVS = $\langle X, Y, D, \{M_d \in D\}, EIC, EOC, IC \rangle$ où :

- X : L'ensemble des ports d'entrée du modèle couplé.
- Y : L'ensemble des ports de sortie du modèle couplé.
- D : L'ensemble des identifiants des sous-modèles (couplés ou atomiques).
- M_d : Le modèle DEVS pour chaque $d \in D$.
- EIC : L'ensemble des connections d'entrée.
- EOC : L'ensemble des connections de sortie.
- IC : L'ensemble des connections internes qui connectent les composants internes du modèle couplé.

Le formalisme DEVS présente plusieurs avantages, on peut citer :

- Il sert, une fois établi, à être utilisé pour la simulation.
- Il permet d'exprimer la communication et les échanges entre plusieurs modèles provenant de différentes disciplines.

3.7.7. La modélisation par Agents :

Les systèmes multi agents sont actuellement très largement utilisés dans les applications complexes et distribuées caractérisées principalement par les interactions entre plusieurs entités. Ces systèmes sont développés avec l'apparition de l'intelligence artificielle distribuée où les systèmes deviennent de plus en plus complexes, évolutifs, repartis sur plusieurs sites et l'accroissement de l'hétérogénéité et donc la nécessité d'entités 'intelligentes' interagissant entre eux afin d'atteindre un but commun.

Un agent [Zeigler, 1976] est un système informatique encapsulé situé dans un environnement dans lequel il est capable d'effectuer des actions flexibles et autonomes. L'agent est en interaction avec cet environnement ou avec d'autres environnements et vise à réaliser des buts de conception.

Les systèmes multi agent sont utilisés dans plusieurs domaines d'application. Ces derniers sont capables de distribuer un problème entre plusieurs entités intelligentes coopérant et partageant des connaissances afin de réaliser un but commun.

3.7.8. L'outil GRAFCET (GRaphe Fonctionnel de Commande Etape/Transition) :

Le Grafcet est une méthode de spécification graphique des automates logiques qui a été développée pour la première fois par le groupe AFCET (Association Française pour la Cybernétique Economique et Technique) en 1977 et puis normalisé sous le nom de *Sequential Function Chart*. Il permet de décrire le comportement séquentiel à partir d'un système automatisé. La figure 2.8 montre un exemple d'une barrière automatique.

Le GRAFCET est composé de trois types d'éléments [Vareille et Arnoux, 1998] :

- *Un ensemble d'étapes* : qui modélisent le comportement du système. Les étapes peuvent être actives ou non actives. L'ensemble des étapes actives représente la situation du système ou du sous-système.
- *Un ensemble de Transitions* : ces transitions permettent de contrôler le passage du système d'une étape à une autre par l'intermédiaire des réceptivités qui leur sont associées. La réceptivité d'une transition est une grandeur logique (vraie ou fausse).
- *Des liens* : reliant les étapes aux transitions et les transitions aux étapes. Ils représentent la structure du système.

Le GRAFCET repose sur deux postulats et cinq règles d'évolution [Vareille et Arnoux, 1998]:

Postulat 1 : Tous les événements sont pris en compte dès leur occurrence et pour toutes leurs incidences.

Postulat 2 : la causalité est considérée à temps nul.

Règle 1 (Situation initiale) : Correspond à la situation du système au début du fonctionnement. Elle est représentée par l'ensemble des étapes actives initialement. Elle caractérise le comportement initial de la partie commande vis-à-vis de la partie opérative et/ou des éléments extérieurs.

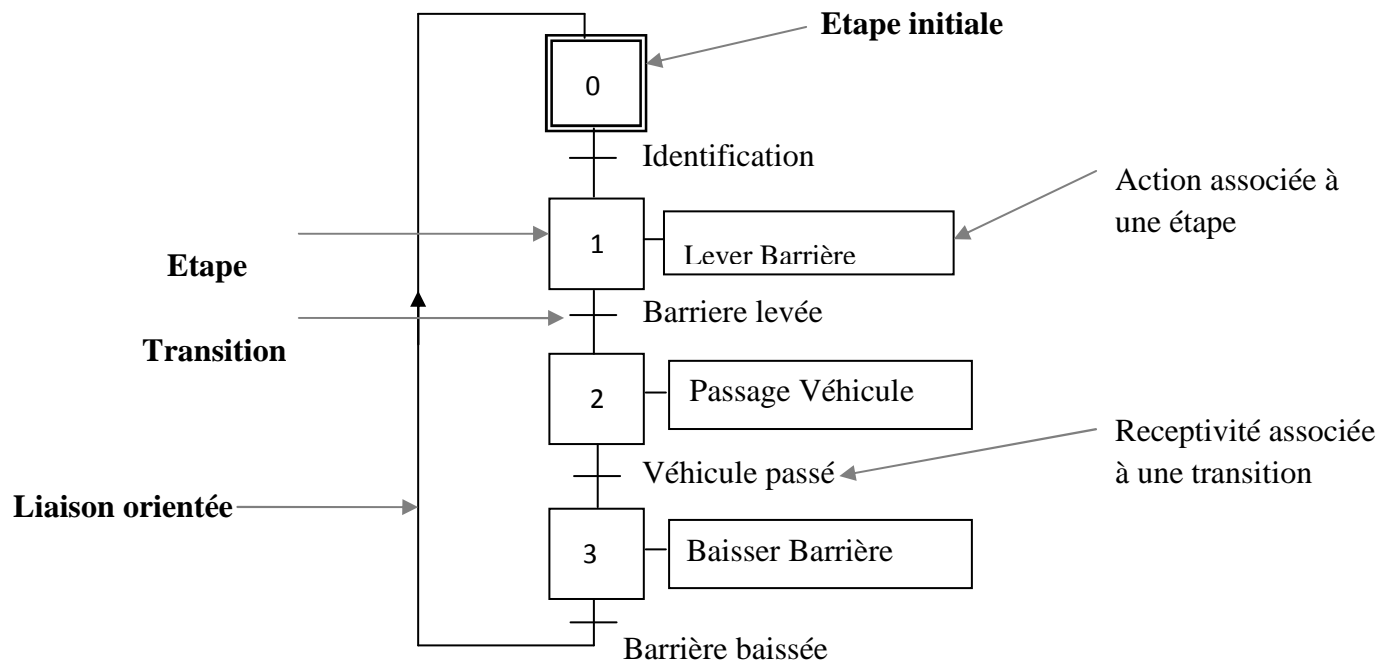


Figure 2.8. GRAFCET correspondant à une barrière automatique

Règle 2 (Franchissement d'une transition) : L'évolution du GRAFCET est caractérisée par le franchissement d'une transition. Deux conditions sont nécessaires au franchissement de celle-ci :

- La transition est validée, cela nécessite que toutes les étapes immédiatement précédentes soient actives
- La réceptivité associée à cette transition est vraie.

Si ces deux conditions sont vérifiées la transition est obligatoirement franchie.

Règle 3 (Evolution des étapes actives) : le franchissement d'une transition entraîne simultanément l'activation de toutes les étapes immédiatement suivantes et la désactivation de toutes les étapes immédiatement précédentes.

Règle 04 (Evolution simultanée) : si plusieurs transitions sont simultanément franchissables, celle-ci sont simultanément franchies.

Règle 05 (Activation et désactivation d'une étape) : Si au cours du fonctionnement, une étape est simultanément activée et désactivée, elle reste active.

4. Simulation :

4.1. Définition :

Plusieurs définitions dans la littérature existent concernant la notion de simulation. Shannon [Shannon, 1976] a défini la simulation par : « *The process of designing a model of a real system and conducting experiments with this model for the purpose either of understanding the behavior of the system or of evaluating various strategies (within the limits imposed by a criterion or a set of criteria) for the operation of the system* ».

La simulation est donc un ensemble d'activités commençant par la conception d'un modèle à partir d'un système réel. Le comportement du système est ensuite étudié et analysé à travers ce modèle en l'exécutant par un simulateur ce qui permet de comprendre son fonctionnement et de prévoir son évolution sous certaines conditions.

La simulation peut être aussi définie comme l'évolution d'un modèle du système réel au cours du temps en modifiant certains paramètres ce qui permet d'aider à comprendre le comportement et le fonctionnement de ce système en mettant l'accent sur certaines caractéristiques dans l'objectif d'évaluer les différentes décisions [Hill, 1996].

4.2. Simulation informatique :

La simulation informatique appelée encore *simulation par ordinateur* est un outil permettant de reproduire un phénomène réel à travers l'exécution d'un programme informatique.

Selon Fishwick [Fishwick, 1997] la simulation informatique est une itération de trois phases fortement couplées et interdépendantes à savoir : la conception du modèle, l'exécution du modèle et l'analyse de l'exécution (Figure 2.9).

- **Elaboration du modèle :** Cette phase consiste à extraire les aspects nécessaires devant être pris en compte d'un système (réel ou théorique) pour élaborer ensuite une représentation abstraite du système (le modèle).
- **Exécution du modèle :** Dans cette phase le modèle établi (modèle abstrait) est traduit en un langage informatique compréhensible par une plateforme de

simulation. Le modèle traduit est exécuté par un simulateur en produisant des données.

- **Analyse de l'exécution** : Cette phase consiste à analyser les résultats de l'exécution du modèle. Les résultats de simulation sont comparés avec les observations pour évaluer les différentes décisions.

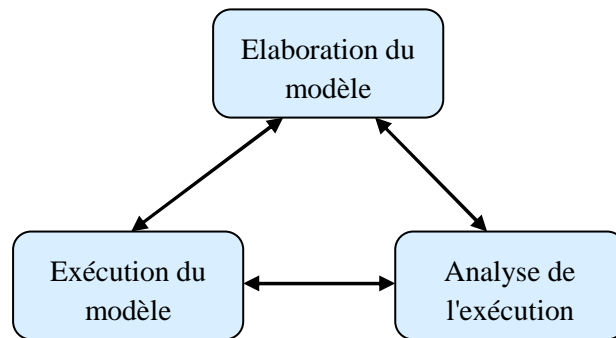


Figure. 2.9. Simulation informatique [Fishwick, 1997]

4.3. Processus de conception d'une simulation informatique :

Fishwick a décomposé la simulation informatique en trois grandes phases fondamentales, Cependant, Shannon [Shannon, 1998] a considéré la simulation comme un processus d'expérimentation composé d'un ensemble d'étapes plus détaillées. Ces étapes sont illustrées dans la figure 2.10.

4.4. Catégories de simulation selon la représentation du temps :

On peut distinguer trois catégories de simulation informatique :

4.4.1. Simulation continue : Dans ce type de simulation les variables d'état du système changent d'une façon continue au cours de la simulation où le temps est représenté par un réel ce qui permet de définir l'état du système à chaque instant.

Ce type de simulation est bien adapté aux systèmes dynamiques qui changent en permanence au cours du temps. Les équations différentielles présentent un outil adéquat pour ce type de simulation.

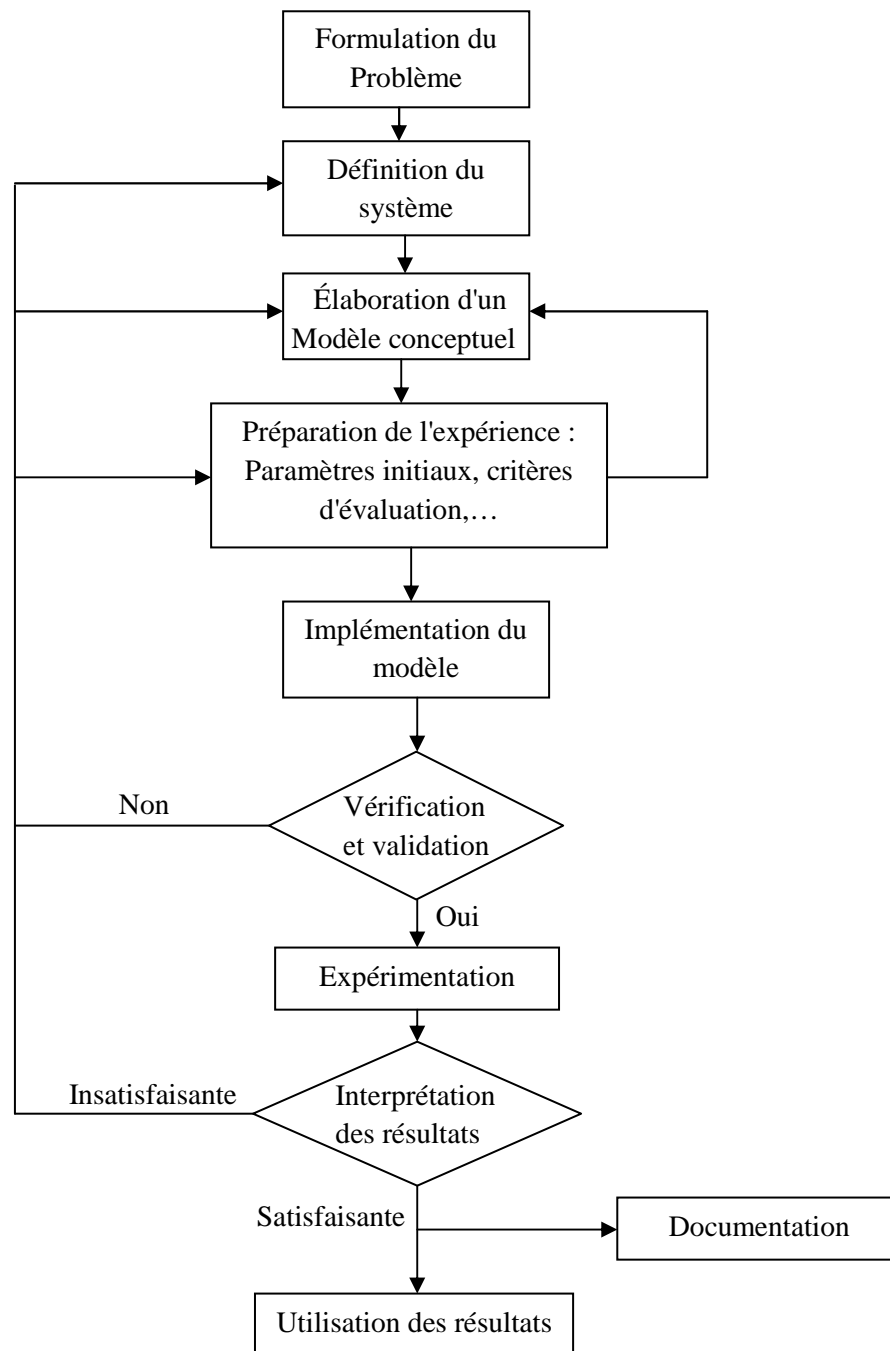


Figure. 2.10. Processus de conception d'une simulation informatique [Shannon, 1998].

4.4.2. Simulation discrète :

Contrairement à la simulation continue, dans la simulation discrète, l'état du système n'est défini qu'à des périodes (unités) de temps. Au cours de cette simulation

le temps est représenté par un entier et l'évolution du système est représentée par une fonction escalier.

4.4.3. Simulation à événements discrets :

Dans ce type de simulation, le changement d'état du système correspond à l'arrivée d'un événement plutôt que l'avancement du temps. Chaque événement est couplé avec son temps d'arrivée. Dans ce cas, on utilise un ordonnanceur d'évènements qui permet de gérer la liste des évènements.

La simulation à événements discrets peut être vue selon trois approches [Miller, 2004] :

- **Simulation orientée événement (Event Scheduling) :**

Dans cette approche, on répertorie tous les événements avec leurs dates d'occurrence. Lorsque le temps est avancé, l'événement le plus tôt dans la liste d'évènements est choisi et le traitement adéquat est effectué (ce traitement est souvent appelé *routine d'événement*) [Cota et Sergent, 1992]. Ce traitement permet de changer l'état du système instantanément et/ou prévoir le futur événement ce qui permet de définir le comportement futur du système.

Un exemple représentant bien cette approche est le graphe d'évènements où les nœuds représentent les événements et les arcs orientés (pouvant être étiquetés par des temps de retard et des conditions) représentent les causalités [Schruben, 1998].

- **Simulation orientée activité (Activity Scanning) :** Dans cette approche l'accent est mis sur les activités et leurs conditions de déclenchement (règle d'activation). Le modélisateur définit les types d'évènements pouvant se produire ainsi que les relations de causalité entre eux [Cota et Sergent, 1992]. Une activité correspond au début et à la fin d'un événement. La condition de déclenchement doit être satisfaite pour que l'activité soit schedulée et effectuée. En plus de la condition de déclenchement d'une activité, celle-ci possède des actions à exécuter au début (lors de la satisfaction de la condition), une durée de l'activité et quelques actions exécutées à sa fin. Les Réseaux de Pétri et les diagrammes de cycle d'activité sont deux exemples utilisant cette approche.

- **Simulation orientée processus (Process Interaction) :** Dans cette approche l'accent est mis sur les processus et leurs interactions. Un processus est une séquence d'événements ou des activités reliés logiquement, ordonnés temporellement et décrivant le comportement d'une entité. Les processus sont représentés typiquement par des diagrammes de contrôle de flux décrivant les séquences procédées par une entité durant son cycle de vie. Le flux de ressources pour chaque entité du système doit être défini par le modélisateur afin de décrire l'interaction entre processus [Cota et Sergent, 1992]. Un mécanisme de gestion des conflits et de synchronisation est nécessaire : les règles d'interruption et de reprise sont utilisées par exemple.

Cette approche est utilisée largement dans la simulation à événements discrets impliquant une contention de ressources. Durant la simulation les entités subissent des séquences d'activités séparées par des intervalles de temps. Ces entités soit reçoivent des ressources soit attendent d'autres [Cassandras et Lafortune, 1999].

Les diagrammes d'activités ont des outils bien adéquats pour cette approche. Ces derniers sont composés d'un ensemble de nœuds fonctionnels : le démarrage, la terminaison, les retards, l'occupation et la libération des ressources [Birtwistle, 1979].

5. Les entités constituant la simulation informatique:

Dans la théorie de Modélisation et Simulation, plusieurs entités interviennent pour constituer une expérience de simulation. Dans ce contexte, Zeigler et al [Zeigler et al, 2000] ont défini les entités suivant (fig. 2.11) :

- **Le système source :** représente le système (réel ou virtuel) à modéliser. Ce dernier est vu comme une source ou une base de données regroupant les différentes informations sur le système. Ces informations sont fortement relatives aux objectifs spécifiés par le cadre expérimental.
- **Le cadre expérimental :** englobe l'ensemble des circonstances et des conditions d'observations du système ainsi que les objectifs qu'on cherche à atteindre : paramètres à prendre en compte, initialisation des paramètres, paramètres de l'expérimentation,...Le cadre expérimental est une entité très

importante durant le processus de simulation : il influence sur la validation du modèle, définit la manière d'évaluer l'efficacité qualitative et quantitative de la simulation [Michel, 2004].

- **Le modèle** : Représente une abstraction du phénomène ou système à simuler. C'est une description abstraite du comportement du système réel ou virtuel. Cette description comporte l'ensemble des instructions, des lois, des équations et des contraintes. Elle permet entre autre de spécifier les mécanismes de changement d'état du système [Michel, 2004].

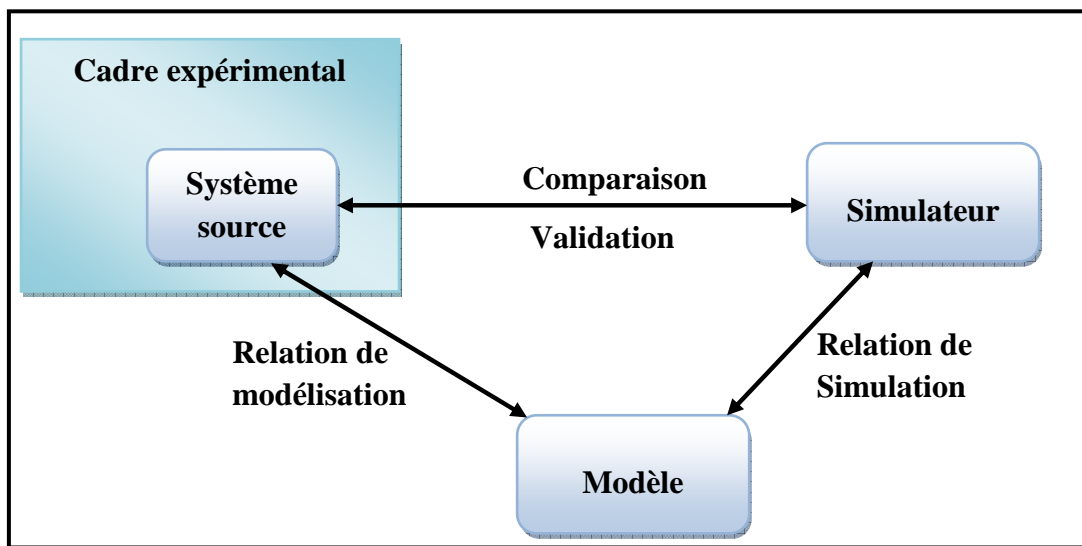


Figure. 2.11. Entités constituant une simulation informatique [Zeigler et al, 2000]

- **Le simulateur** : C'est l'entité capable d'exécuter les spécifications établies par le modèle. Il permet de reproduire ou générer le comportement du système sous certaines conditions. C'est un programme ou une plateforme informatique capable d'interpréter les modèles dynamiques en produisant les perturbations désirées [Treuil et al, 2008], tout en respectant les spécifications.

6. La validation dans le processus de simulation :

6.1. Validation du modèle :

La validité du modèle consiste en général à vérifier si le modèle établi est une simplification (abstraction) acceptable du système source en fonction des objectifs exprimés dans le cadre expérimental. Selon le niveau de la hiérarchie de spécification

du système, Zeigler et al, ont défini trois types de validité différents [Zeigler et al, 2000] :

- *Validité réplivative* : Vérification si le comportement du modèle est satisfaisant à celui du système source pour toutes les expériences menées par le cadre expérimental avec un degré acceptable de tolérance.
- *Validité prédictive* : Implique la validité réplivative et offre en plus une capacité de prédire le comportement invisible du système.
- *Validité structurelle* : Prend sa place au niveau hiérarchique correspondant aux mécanismes de changement d'état du système [Michel, 2004]. Elle implique validité réplivative et en plus, elle permet de reproduire d'une façon plus détaillée (composant par composant, étape par étape) le mécanisme de changement d'état du système en imitant son fonctionnement principal.

6.2. Validation du simulateur :

Consiste à vérifier la validité du passage du modèle abstrait vers l'implémentation (le simulateur). On doit vérifier si le simulateur reproduit fidèlement le comportement du système.

Le simulateur n'est pas jugé valide a priori même si les spécifications du modèle sont faites sans ambiguïté. On doit prendre en considération d'autres facteurs qui viennent à posteriori : par exemple la qualité des générateurs des nombres aléatoires dans une simulation stochastique, l'environnement d'exécution du simulateur,... [Michel, 2004].

7. Conclusion

Il existe actuellement une grande panoplie d'outils, de méthodes et de plateformes informatiques qui fournissent un ensemble de cadres formels et opérationnels favorables à la mise en œuvre de modèles de systèmes et de leurs simulations. Des contraintes additionnelles se posent lorsqu'il s'agit de multi-modélisation où différents formalismes et paradigmes sont utilisés pour la représentation des systèmes. Ceci pose un problème de conception pour cette classe de systèmes.

CHAPITRE 3

Modélisation et Simulation des systèmes de production

1. Introduction :

Les systèmes de production sont en évolution permanente avec l'automatisation des différentes opérations, la concurrence du marché, la diversité et la qualité des produits. Ces facteurs rendent ces systèmes de plus en plus complexes au niveau de leurs structures, de leurs comportements et de l'interdépendance de leurs fonctions, leur analyse et leur conception ne peuvent se réaliser que selon une approche globale.

La modélisation et la simulation sont des techniques adéquates pour analyser, concevoir, optimiser et comprendre les phénomènes mis en jeu par ces systèmes. Dans ce chapitre nous traitons les différentes approches de modélisation et de simulation des systèmes de production, en passant d'abord par la définition de quelques concepts et des notions de base de ce domaine.

2. Concepts essentiels pour la modélisation d'un système de production :

Comme vu dans le chapitre précédent, la décomposition d'un système de production peut mener à extraire trois parties fondamentales : une partie pilotage, une partie flux et une partie physique. A partir de ces trois parties on peut identifier trois concepts pour modéliser un système de production [HABCHI *et al*, 2006] : l'Entité Circulante (EC), le Système de Transformation de Produit (STP) et le Centre de Pilotage (CP).

2.1. L'Entité Circulante (EC) :

2.1.1. Définition :

Ces entités constituent le flux physique dans un SdP. Une entité circulante représente donc le produit ainsi que son flux à travers les différentes ressources du système physique. Ce flux n'est pas aléatoire : il possède un objectif et suit une trajectoire. Cette dernière est définie à priori par un procédé de fabrication préétabli (son changement est rare : en cas de changement de gammes ou de nomenclature). L'EC est une entité passive (elle n'exécute pas d'actions), mais elle permet d'activer les CP et les STP par son passage en déclenchant leurs comportements.

2.1.2. Propriétés :

Une EC est caractérisé principalement par les attributs suivants [Berchet 2000] [Azeiez 2007] :

- **Un identifiant** : qui est unique et représente le nom du produit.
- **La gamme** : définie formellement par le quadruplet:

$G = \langle I, S, STP, t_{op} \rangle$, tel que:

- **I** : identificateur de la gamme.
- **S** : Numéro d'opération (entier naturel) effectué sur l'EC.
- **STP** : Identificateur du STP représentant la ressource nécessaire pour réaliser l'opération.
- **t_{op}** : temps d'une opération (un nombre réel).

Une EC est défini aussi par :

- *une liste de fonction*: Créer, grouper, dissocier, supprimer, ...
- *une liste d'indicateurs* : Date de début et date de fin de l'opération sur le STP concerné, temps moyen de production par type d'entité ainsi que le temps de production pour une entité.
- *une liste d'événements* : Principalement correspondant au lancement des ECs dans le processus de fabrication. Un événement est caractérisé par un identificateur et une date d'occurrence.

Le comportement d'une EC est défini par un processus via lequel un ensemble d'opérations est effectué. Chaque opération est caractérisée par un identifiant et un temps opératoire.

2.2. Le Système de Transformation de Produits (STP) :

C'est un concept regroupant toutes les caractéristiques fonctionnelles et structurelles d'une ressource de production (machine robot, stock, convoyeur,...). Le STP exécute en cascade trois opérations de base : réception des ECs, transformation, et fourniture (Figure 3.1) [Azeiez 2007] [Berchet, 2000] :

- **Réception**: Cette fonction consiste à obtenir une ou plusieurs ECs en provenance d'une source principale d'alimentation du SdP ou d'un autre STP. Cette fonction nécessite la disponibilité des l'ECs ainsi que l'habilité de la ressource invoquée à réceptionner ces ECs.

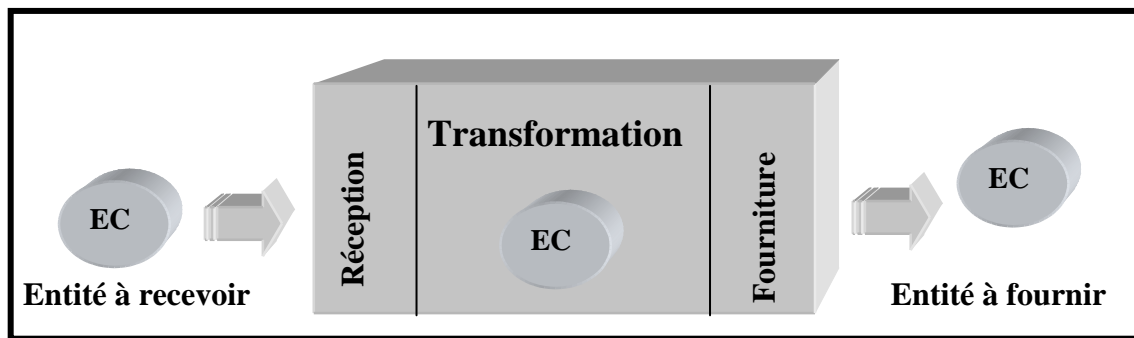


Figure 3.1. Les trois fonctions de base d'un STP [Bakalem et al, 96]

- **Transformation** : Consiste à transformer ou assembler/désassembler une ou plusieurs ECs en prenant une certaine durée de temps définie par la gamme de production. Pour une durée supérieure ou égale à cette durée, l'EC et le STP sont occupés.
- **Fourniture** : Après l'écoulement la durée de transformation, et si l'EC est prête à fournir, celle-ci est transférée vers une destination qui peut être un autre STP. Le STP courant est donc libéré.

Le STP peut être une entité active (machine, robot,...) permettant d'ajouter la valeur au produit. Il peut être aussi une entité passive (Stocks, moyens de transfert). En outre, plusieurs STPs différents (en performances, en réglages, en pannes...) fonctionnellement homogènes, peuvent être représentées par le même STP. Ce dernier est appelé Centre de Charge (CC) [Azeiez 2007].

Le STP est une structure récursive et hiérarchique constituant un réseau et permettant de modéliser une ressource d'un SdP à des niveaux hiérarchiques et d'abstraction différents. Formellement, un STP est caractérisé par un ensemble de caractéristiques qu'on peut citer [Habchi, 2001] :

- **Un identificateur** : qui permet une désignation unique dans le système.
- **Une capacité** : concernant un ensemble de ressources identiques.
- **Un calendrier d'ouverture et de fermeture** : défini par une combinaison de la durée d'ouverture/fermeture avec la capacité du STP pendant cette durée.

- **Une loi des pannes** : en plus des arrêts programmés (pour la maintenance préventive), le STP peut subir des pannes aléatoires. Ces dernières sont générées à l'aide d'une analyse de l'historique des pannes afin d'en bien déterminer la distribution aléatoire adéquate (Loi de poisson, loi normal, loi exponentiel). Dans ce contexte, il est nécessaire aussi de connaître la durée de panne ainsi que la durée entre deux pannes successives. Ces deux renseignements sont générés par une distribution aléatoire.
- **Le temps de changement de série** : Correspond au changement du type du produit fini ou semi finis et est défini par une distribution aléatoire.
- **Un taux de Rebut** : Qui est un pourcentage accepté de rebut de production pour un STP. L'évolution dans le temps du rebut d'un STP suivant une loi de dégradation doit être aussi pris en compte.
- **Un rendement** : qui est un rapport représentant l'évolution de la productivité du STP en se référant au temps opératoire de la gamme.

2.3. Le Centre de Pilotage (CP) :

2.3.1. Définition :

Le Centre de pilotage est le processeur d'un système de production représentant une entité complexe et intelligente. Berchet [Berchet, 2000] a défini un CP par :

«Une structure organisée et autonome (mais dépendante de la stratégie de l'entreprise) ayant un pouvoir décisionnel, associée à une entité à piloter et disposant d'un ensemble de ressources nécessaires à la mise en place d'actions pour atteindre un ou plusieurs objectifs définis dans le cadre global de l'entreprise ».

- **Structure organisée** : D'un point de vue interne, le comportement d'un CP est structuré par un processus contenant une séquence d'étapes (figure 3.2). D'un point de vue externe, le CP est défini selon trois axes : *verticale* correspondant à l'hierarchie du système de pilotage, *horizontale* correspondant au type d'organisation du système physique et *transversal* correspondant aux conditions spatio-temporelles pouvant orienter la décision.

- **Structure autonome** : Cette autonomie réside dans le fait qu'un CP possède un pouvoir de choisir entre plusieurs solutions possibles afin de contrôler le processus d'action dans le cadre des objectifs assignés.

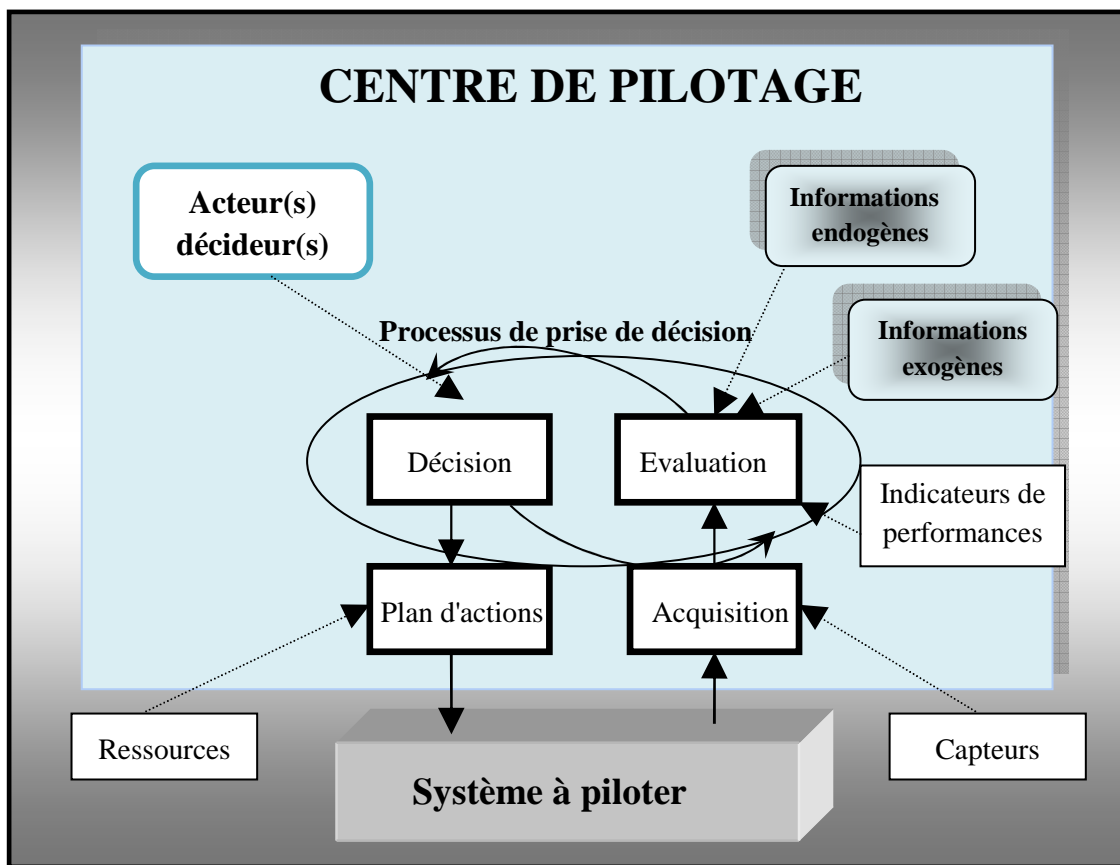


Figure 3.2. Comportement interne d'un Centre de Pilotage [Berchet, 2000].

- **Structure dépendante de la stratégie globale** : La structure est dépendante de la stratégie globale choisie pendant le développement de l'entreprise (hiérarchique, hétéroarchique, coordonnées,...). Toute décision prise par un CP doit être en cohérence avec le cadre général de l'entreprise.
- **Structure ayant un pouvoir décisionnel** : Les acteurs décideurs sont les composants les plus importants du CP. Ces derniers ont des capacités de raisonnement et d'apprentissage à l'aide des moyens et outils (variables d'actions, outils d'aide à la décision, des ressources, ...) ce qui permet d'établir des plans d'actions et des prises de décisions afin d'atteindre leurs propres objectifs.

- **Structure associée à une entité** : Selon le niveau (position) hiérarchique dans le réseau STP-CP (voir §2.4) d'un CP, ceci pilote un CP inférieur ou un STP.
- **Structure disposant de ressources** : un CP dispose de ressources matériels (machines, stocks...) et des ressources humaines internes ou externes.
- **Structure ayant des objectifs définis dans le cadre globale de l'entreprise** : La stratégie globale de l'entreprise devient opérationnelle. Ceci est réalisé par des objectifs globaux qui sont déployés en objectifs locaux propres aux CPs.

D'après cette définition on peut déduire qu'un CP est une entité possédant un certain niveau d'autonomie et capable d'acquérir des mesures d'un STP. Ces mesures permettent au CP dans son processus de prise de décision de délivrer un plan d'actions en faisant mobiliser diverses ressources afin d'atteindre son propre objectif qui doit être adapté avec l'objectif global de l'entreprise. Cet objectif permet de réaliser la stratégie globale de l'entreprise.

2.3.2. Processus de prise décision :

Comme illustré dans la figure 3.2, le processus de prise de décision est une séquence de quatre phases :

- **Mesures (Acquisitions)** : qui peuvent être des mesures événementielles ou périodiques. Les mesures événementielles permettent d'acquérir des informations concernant l'évolution de l'état du système piloté à l'aide des capteurs (marches, arrêts, ...). Ces mesures peuvent être déterminées à priori ou accidentellement. Les mesures périodiques permettent d'offrir à la hiérarchie des informations tirées du système piloté (taux d'activité, quantité produite par unité de temps, ...). Ces informations peuvent être considérées comme des comptes rendus ayant une influence sur la prise de décision.
- **Evaluation** : Cette phase consiste à analyser, générer et développer les différentes alternatives possibles tout en sélectionnant les critères d'évaluation. Cette phase contient deux étapes :

Calcul de la performance : consiste à comparer les mesures avec les seuils définis dans le cadre de la stratégie globale du système.

Evaluation de la performance : consiste à vérifier si chaque inducteur atteint son objectif et ensuite prévoir les actions correspondantes.

- **Décision** : Cette phase est fortement liée à la précédente. Elle consiste à décider le plan d'action à appliquer et d'analyser les différentes conséquences.
- **Application du plan d'action** : Dans cette dernière phase, le plan d'action choisi est envoyé au système piloté afin de l'appliquer en utilisant les ressources nécessaires.

2.3.3. Typologie des structures de pilotage :

Les systèmes de pilotage peuvent s'organiser selon différentes formes (figure 3.3), on parle précisément de la façon de prise de décision concernant la gestion des ressources. On distingue [Trentesaux 1996] :

- **Pilotage centralisé** : Ce type est basé sur l'ordonnancement des tâches où un centre de pilotage unique comprenant un mécanisme de prise de décision centralisé, prend en charge la supervision et la coordination de production et la prévision des différentes tâches. Ce type de pilotage est bien adapté dans les petites entreprises.
- **Pilotage hiérarchisé** : Ce type de pilotage est bien adapté dans le cas des systèmes complexes où chacun de ces derniers est décomposé en sous-systèmes. Dans cette structure, une seule entité de pilotage de niveau supérieur est chargée de piloter la production, les niveaux inférieurs peuvent avoir un degré d'autonomie de décision (gestion des aléas et perturbations).
- **Pilotage coordonné** : C'est un ensemble de structures hiérarchisées avec une coopération entre les entités du même niveau ce qui permet à un niveau de décision d'avoir une réactivité et une capacité de résoudre les problèmes localement.
- **Pilotage décentralisé** : constitué de plusieurs entités décisionnelles de même niveau fonctionnel. Ces entités ne sont pas donc subordonnées par un centre hiérarchique supérieur. Ce type de pilotage nécessite un mécanisme d'auto-organisation des centres de décision ainsi qu'un mécanisme de synchronisation entre eux.

- **Pilotage distribué** : Dans ce cas le pilotage est reparti sur un sous ensemble d'entités décisionnelles ce qui permet aux informations de circuler facilement entre les entités de pilotage.

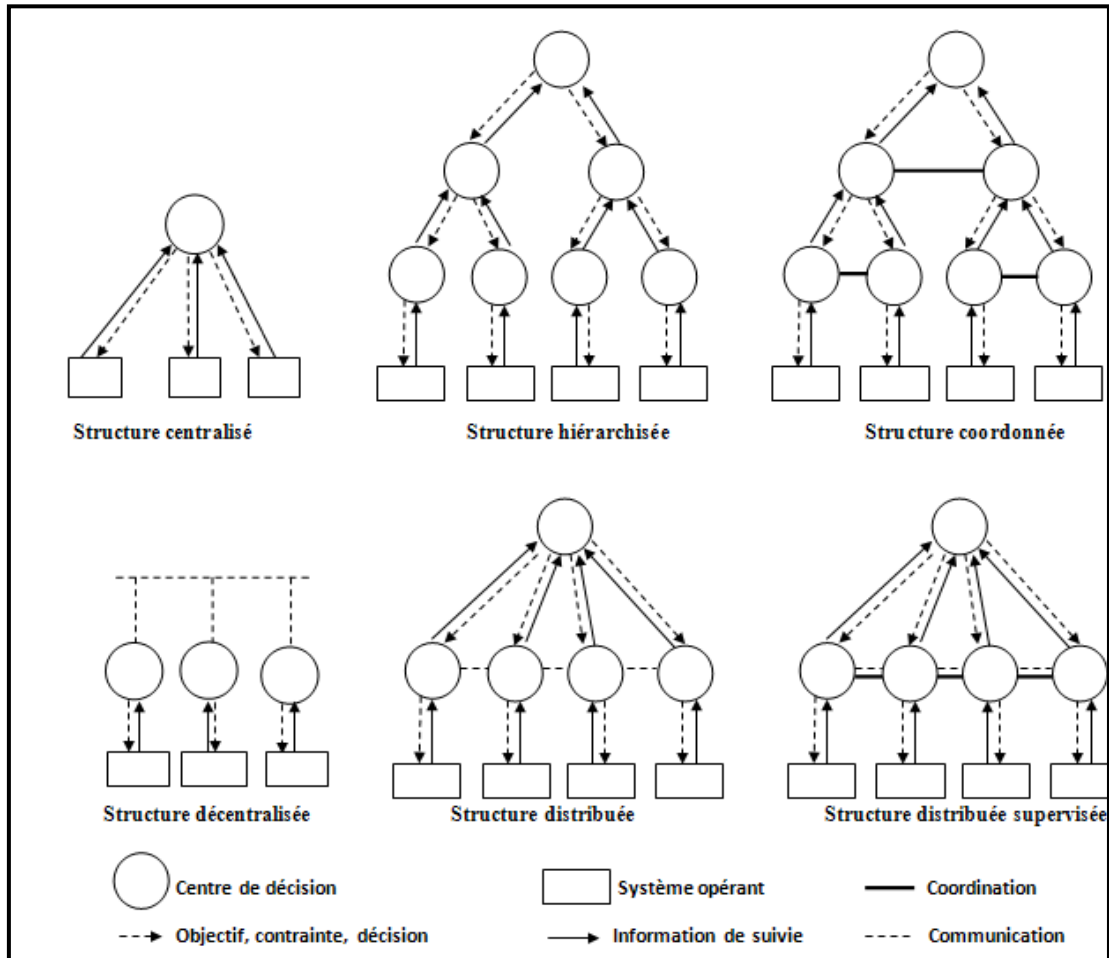


Figure 3.3. Typologie des structures de pilotage

- **Pilotage distribué supervisé** : Ce type de pilotage consiste à distribuer le pilotage à un ensemble d'entités décisionnelles. Ces entités sont interconnectées (par un mécanisme d'intercommunication) afin de permettre l'échange d'informations et les décisions prises. Ces entités sont supervisées par un centre de décision supérieur hiérarchiquement.

2.4. Le réseau STP-CP :

Comme vu dans la section précédente, le CP constitue la partie décisionnelle dans un système de production et modélise donc le système de pilotage. Le système physique de fabrication ainsi que le flux de production sont modélisés à l'aide du STP et l'EC.

Pour lier ces deux systèmes le concept de réseau STP-CP [Berchet, 2000] est introduit. Un réseau STP-CP est une structure reliant les CPs et les STPs par des liens hiérarchiques à plusieurs niveaux (figure 3.4) avec représentation des flux entrants et sortants (modélisés par des ECs). L'agencement de cette structure dépend toujours de l'organisation du système de production réel et la structure de pilotage adaptée (hiérarchique, coordonnée, centralisée,...). En effet, un CP peut piloter un ou plusieurs CPs de niveaux inférieurs

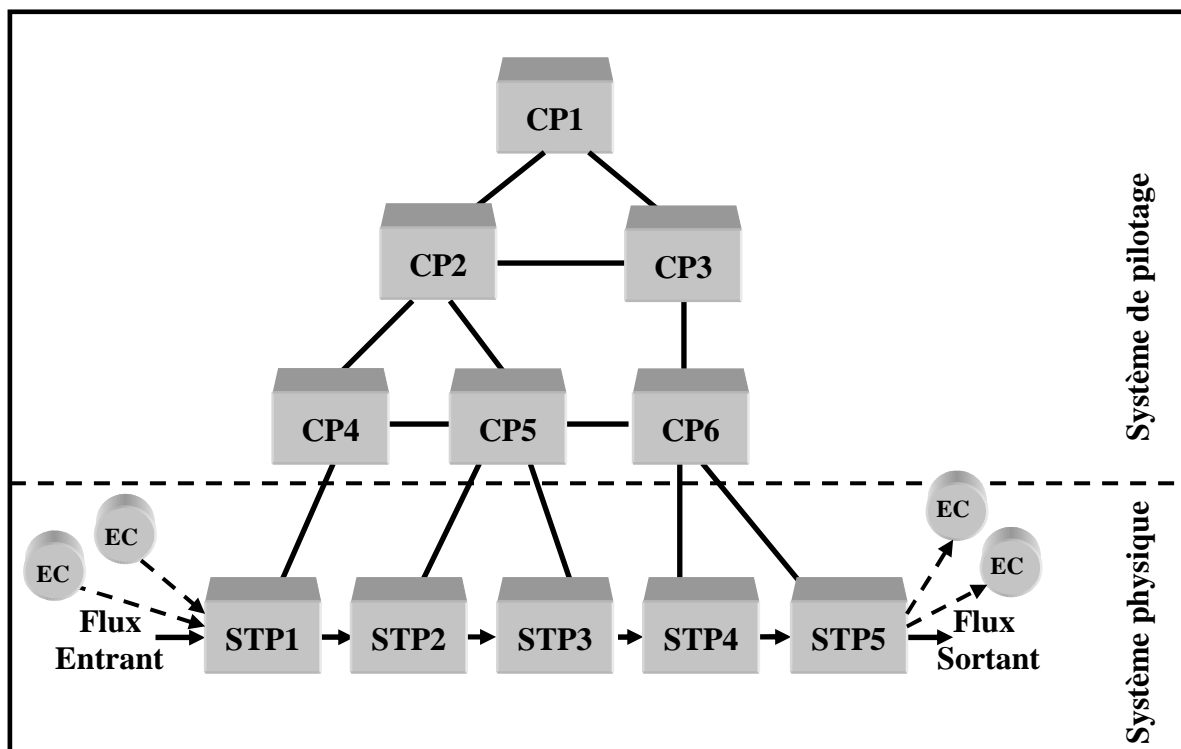


Figure 3.4. Exemple d'un réseau STP-CP

comme il peut piloter des STPs dans le cas où ce CP est relié directement avec le système physique.

Pour modéliser le système de pilotage, on a vu précédemment que le concept « CP » est utilisé, toutefois, il existe d'autres structures pour modéliser ce système, on notera par exemple le concept SIP (Station Intégrée de Pilotage) introduit par Trentesaux [Trentesaux 1996]. Une SIP est une entité permettant de piloter des SIP de niveaux inférieurs ou un ensemble de ressources appartenant au système physique. Une SIP est composée de cinq sous-systèmes : sous-système de décision, sous-système de contrôle/commande, sous-système d'information interne, sous-système de communication et un sous système d'interface. Une SIP est une entité autonome et permet donc d'assurer une gestion locale des actions ainsi

qu'une gestion des processus d'allocation dynamiques des tâches aux ressources et en plus elle gère la disponibilité des ressources physiques (reprise des ressources après pannes,...).

Un autre concept est le CD (Centre de **D**écision). Ce concept a été défini par Bitton comme « *ensemble d'activités d'une fonction (type d'activités similaires à l'intérieur d'un système de décision donné) donnée, à un niveau temporel donné* » [Bitton 1990]. On peut déduire de cette définition que l'ensemble d'activités d'un CD doit se réaliser dans le même intervalle de temps (à un niveau temporel donné) pour atteindre un objectif commun (type d'activités similaires...).

3. Les approches hétérarchiques de modélisation des systèmes de production :

La littérature fait référence à plusieurs approches de modélisation des entités du système de production ainsi que leurs interactions. Dans cette section on examine les approches hétérarchiques et autonomes- *appelés approches émergentes*- les plus connues, à savoir les approches *Holoniques, Bioniques, Fractales* et *Multi Agents*.

3.1. Approches Holoniques :

L'approche holonique a été introduite par Arthur Koestler [Koestler, 1967] pour modéliser les systèmes sociaux complexes. Dans cet ouvrage le discours original était: "*The term I would propose is 'holon', from the Greek "holos = whole", with the suffix "on" which, as in proton or neutron, suggests a particle or part*".

Le principe de base de cette approche est que chaque entité de la vie réelle est considérée à la fois comme un ensemble constitué d'autres entités et en tant que partie d'un ensemble. Cette idée est appliquée dans différents domaines, particulièrement dans les systèmes HMS (*Holonics Manufacturing Systems*). Les éléments d'un HMS (les holons) constituent donc un système autonome et coopératif dédié à la réalisation des tâches de transformation, de transport, de stockage et/ou de validation des informations ou des objets matériels. Un Holon est constitué d'une partie de traitement d'informations et parfois de flux physique [Van Brussel et al, 1998].

Une société d'holons coopérants pour atteindre un objectif global est appelée *Holarchie*. Pour atteindre cet objectif, les holons - bien qu'ils possèdent une autonomie - doivent avoir un certain niveau de connaissances du système entier.

De ce point de vue, les éléments clés d'un HMS peuvent être des matières premières, des machines, des produits, des pièces, des AGV, etc., disposant toujours des deux propriétés [Van Brussel et al, 1998] :

- **L'autonomie** : Un holon possède une autonomie décisionnelle. Il peut créer ses propres plans et/ou stratégies et contrôler leur exécution.
- **La coopération** : Les plans créés par un ensemble d'holons peuvent être exécutés par d'autres ensembles et vice-versa.

De nombreux travaux de modélisation des systèmes de production se basant sur cette approche ont été réalisés. Le plus réputé, on trouve l'architecture de référence PROSA (**P**roduct **R**essource **O**rdre **S**taff **A**gent) proposée par Van Brussel et al [Van Brussel et al, 1998] fondée sur trois types d'holons de base : les ordres, les produits et les ressources (figure 3.5) structurés selon les concepts d'agrégation et de spécialisation du paradigme orienté-objet. Un quatrième type d'holon peut être ajouté : le Staff Agent qui assure leur coordination avec ses connaissances expertes.

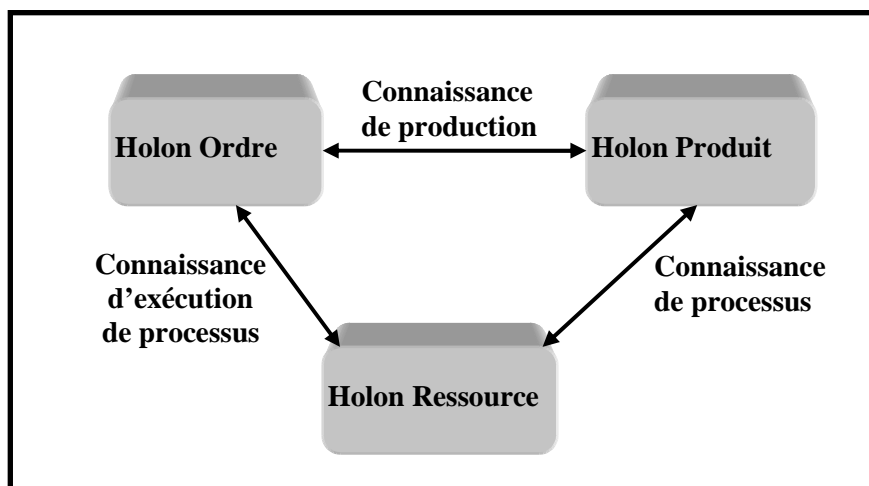


Figure 3.5. *Système de production holonique [Van Brussel et al, 1998]*

Holon Ressource : intègre la partie contrôle à la partie physique du système de production. C'est une abstraction d'un moyen de production (Atelier, machine, convoyeur, pipeline, matière première, etc.). Elle tient les méthodes d'allocation de ressources ainsi que les

connaissances et les procédures permettant d'organiser et de contrôler les ressources de production afin de piloter la production.

Holon Produit : considéré comme un serveur d'autres holons du HMS en offrant des informations consistantes et à jours sur le cycle de vie du produit, les exigences de l'utilisateur, les procédures d'assurance de qualité, etc.

Holon Ordre : Représente une tâche dans le système (ordre de client, ordre d'approvisionnement, ordre de maintenance) et gère le produit physiques, le modèle d'état de produit ainsi que tous les traitements des informations logistiques concernées. Ce holon sert aussi à suivre l'état des ressources en liées pendant l'accomplissement de la tâche.

Afin de maîtriser la complexité engendrée par les interactions entre un nombre important d'agents de bas niveaux, des relations d'agrégation et de spécialisation au niveau de chacun de ces trois holons sont mises en place.

Un autre modèle se basant sur l'approche holonique ADACOR [Leitao et Restivo, 2008] propose un holon superviseur pour la coordination et l'optimisation ainsi que des holons opérationnels. C'est une approche distribuée pour la résolution du problème d'ordonnancement. Cette structure permet de basculer dynamiquement vers une organisation hiérarchique si on cherche une optimisation, et vers une organisation distribuée (sans intervention de l'agent superviseur) dans le cas où le système est subi à des événements aléatoires.

[Ounnar et Pujo, 2012] ont proposé une approche pour le contrôle des systèmes de production de type job-shop en utilisant le paradigme holonique basée sur une architecture isoarchique et une capacité de prise de décision reposant sur l'analyse multicritère.

3.2. Approches Bioniques :

Appelées **BPS** pour ***Bionic Production Systems***. Ces approches sont proposées par Okino [Okino, 1993] et sont inspirées des phénomènes naturels issus du monde vivant. Ces approches exploitent les propriétés et les comportements biologiques à savoir : l'autonomie, l'auto-organisation, l'apprentissage, l'adaptation, l'évolutivité pour les intégrer dans la conception des systèmes de production [Ueda, 2001].

Les entités d'un système de production bionique peuvent être vues comme des organismes d'un système vivant. Dans ce contexte, diverses analogies sont possibles : par

exemple, on peut comparer une unité de production (constituée des produits et des ressources) à une cellule, et dans ce cas l'organisation des entités est naturellement hiérarchique (cellule, organe, être vivant, société) et coordonnée (à l'aide des enzymes) [Okino, 1993]. Autre analogie compare une colonie de fourmis à un système de production et dans ce cas une entité intelligente décisionnelle est comparée à une fourmi. Cette organisation est totalement hétérarchique et utilise un mécanisme de communication entre entités par phénomènes.

L'approche bionique est bien utilisée dans le cas des systèmes nécessitant une adaptabilité par rapport aux changements de l'environnement.

Plusieurs travaux relèvent de cette approche, on trouve par exemple dans le domaine de la robotique les travaux d'Arkin et al. [Arkin et al., 1997]. Ces derniers ont développé leur propre approche nommée *Colonies de Robot (Robot colonies)*. Cette approche exploite les mécanismes d'interaction biologiques et les utilise dans les systèmes d'information distribués pour résoudre les problèmes d'interaction entre robots intelligents. Pour concevoir un système de contrôle de robots, cette approche se base sur certaines méthodes d'interaction biologique tels que : la perception de l'environnement (détection d'autres robots, détection des obstacles, ...), la vision active, l'apprentissage par renforcement.

Peeters et al [Peeters et al, 2001] ont proposé un système de contrôle basé sur le mécanisme de coordination des colonies des insectes offrant une capacité de réponse face aux perturbations et de reconfiguration avec le changement de l'environnement. Ce système permet le contrôle d'un atelier de production de type Flow Shop Flexible et utilise l'approche multi agents pour représenter les entités de prise de décision. Ces agents fonctionnent dans un environnement d'information distribué où l'information est disponible sous forme de phéromones (substances chimiques secrétés par des espèces naturelles pour guider leurs semblables) artificielles. Cette approche exploite les comportements des insectes (comme les fourmis) dans leurs environnements et construit une analogie avec des agents logiciels.

3.3. Approches Fractales :

Appelés **FrMS** pour *Fractal Manufacturing Systems*. L'approche fractale a été introduite par Warneke [Warneke, 1993] pour répondre aux besoins et exigences des nouvelles organisations: la flexibilité, l'adaptabilité, la réactivité, la stabilité vis-à-vis aux perturbations ainsi que les changements continus. L'origine de cette approche était la géométrie fractale, de ce fait, un système de production est un ensemble d'entités fractales

(EFs) organisés à plusieurs niveaux hiérarchiques. Ces entités coopèrent pour assurer une cohérence entre les objectifs locaux et globaux (figure 3.6).

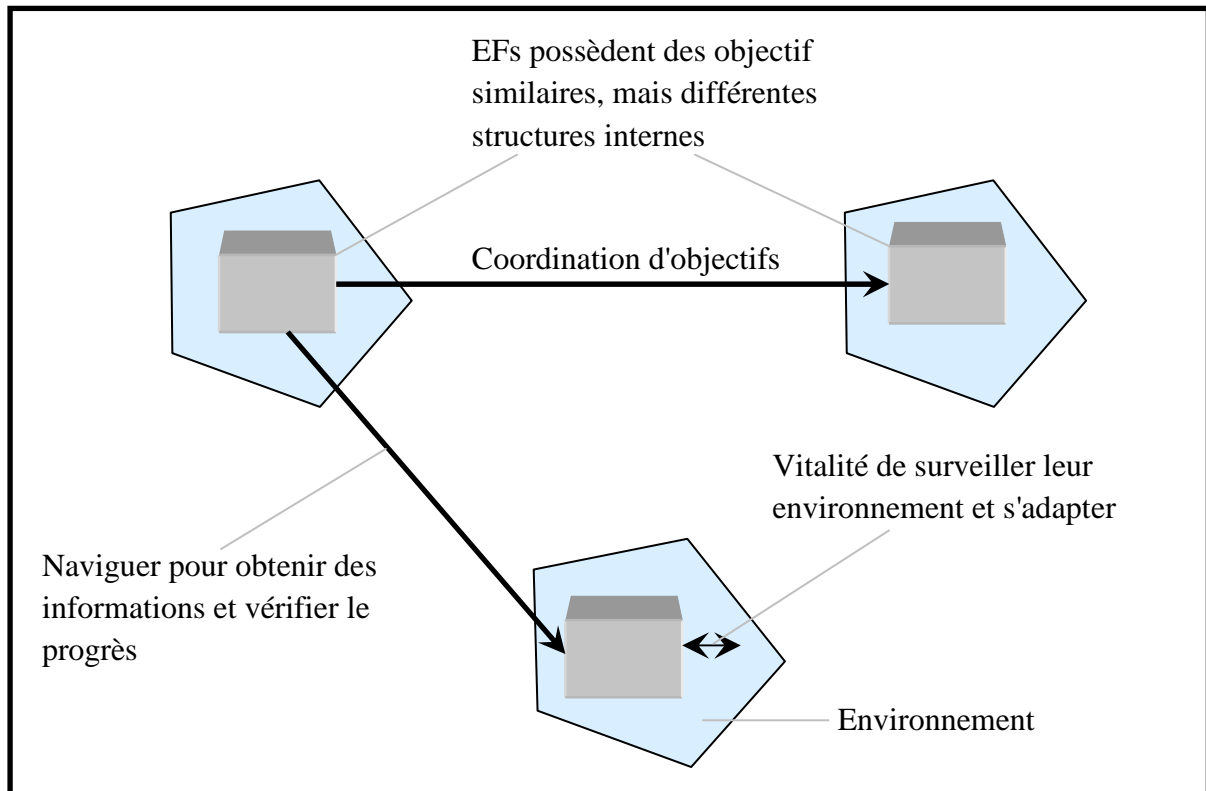


Figure.3.6. Entités Fractales d'un système de production [Tharumarajah et al, 1998]

Les caractéristiques principales des fractales sont [Warneke, 1993][Tharumarajah et al, 1998][Ryu et al, 2003] :

- **L'auto-similarité** : Les fractales ont des objectifs similaires et conformes bien que leurs structures internes sont différentes. En d'autre terme, les fractals sont capables d'introduire les mêmes sorties en utilisant les mêmes entrées sans considérer les structures et les procédures internes de celle-ci. La similarité d'objectifs est assurée par les coordinateurs du système et un processus de négociation entre les EFs du même niveau hiérarchique.
- **L'auto-organisation** : Chaque unité est capable d'agencer et restructurer sa structure interne en fonction de sa configuration initiale et ce face aux événements inattendus (pannes, changement des gammes, changements des besoins des clients,...). Pour cela les entités fractales ont une liberté pour choisir leurs propres méthodes de résolution de problèmes durant le processus de production.

- **L'auto-optimisation :** Les entités fractales sont continuellement soumises à une évaluation de leurs performances ce qui permet d'améliorer les processus de production. Ceci est réalisé grâce à la capacité des EFs de se restructurer dynamiquement.
- **L'orientation-but :** Bien que chaque fractal possède son propre objectif, la consistance d'un objectif global doit être maintenue. Pour cela la formation d'objectif émergent qui est un processus de négociation entre les fractals, est nécessaire. Ce processus doit résoudre les conflits entre les objectifs concurrents des différents fractals. Le développement itératif des objectifs individuels et par un feedback à chaque réalisation d'un objectif mène à cet objectif global.
- **Dynamiques et vitalité :** Les fractals doivent être dynamiques et capables de s'adapter aux changements dynamiques de l'environnement. En plus les fractals doivent être – durant leurs vie – vitales et donc capable de coopérer et de négocier ce qui conduit à des éventuelles corrections en terme d'objectifs et des relations avec les autres fractals.

Cette approche a connu plusieurs travaux. On peut citer entre autres, ceux de Warneke [Warneke, 1993] qui a représenté les unités de fabrication par des entités fractales considérée chacune comme un fournisseur de services et possédant un ensemble de buts. Les buts individuels sont dirigés vers l'atteinte d'un but global et corporatif. Ceci est réalisé par la coopération et l'itération de processus de développement de buts (figure 3.7).

Ryu et al [Ryu et al, 2000] ont proposé une architecture d'une Entité Fractale composée de cinq modules fonctionnels (figure 3.8) : un Observateur, un Analyseur, un Résolveur, un Organisateur et un Rapporteur. Les fractals sont distribués aux différents niveaux hiérarchiques du système et disposant - à un niveau donné- les mêmes modules fonctionnels. Ces fractals sont en interaction (coopération et négociation) avec les fractals de niveaux adjacents. Cette architecture dispose d'un très haut niveau d'auto-similarité (l'une des caractéristiques principales d'un système fractal) :

- **L'observateur :** Le rôle de ce module est de contrôler l'état de l'entité fractale, et d'échanger des informations avec les autres fractals. Les fractales de niveau hiérarchique plus bas détectent les signaux envoyés par le système physique à l'aide de senseurs.

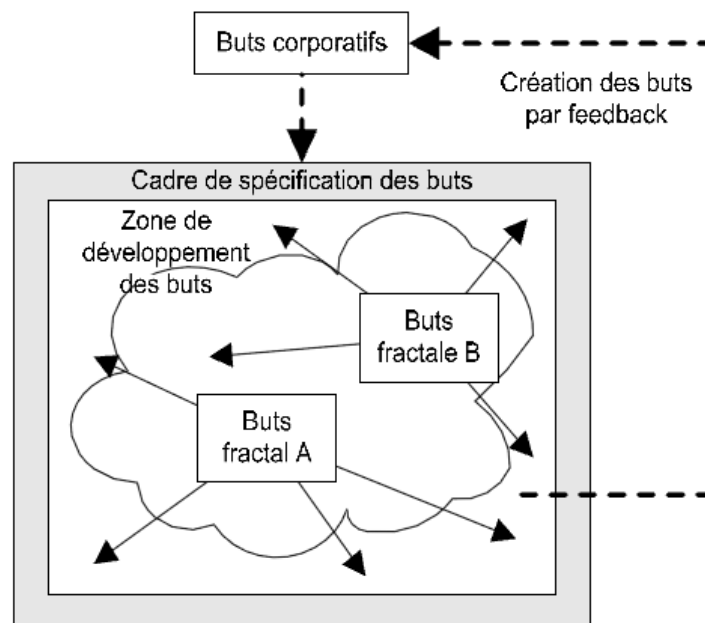


Figure. 3.7. *Coopération et itération des EFs [Warneke, 1993]*

- **L'Analyseur** : Son rôle est d'analyser et évaluer les alternatives de profils de travail (les plans) générées par le Résolveur. Il choisit la meilleure règle de priorité pour atteindre ses objectifs (par exemple le meilleur délai), simule les profils analysés en temps réel afin de les évaluer, et envoie les résultats au Résolveur.
- **Le résolveur** : C'est le module le plus important dans une entité fractale. Il génère les profils de travail, forme un objectif optimal en utilisant des techniques heuristiques ou numériques d'optimisation, prend les décisions en exploitant une base de connaissances et si nécessaire, initie des processus de négociation, de coopération et de coordination entre les fractales.
- **L'organisateur** : sert à gérer l'état du fractale en cherchant dynamiquement à une reconfiguration ou une restructuration optimale en utilisant différentes techniques d'optimisation. Par exemple pour préserver la productivité, les charges des fractales sont dynamiquement équilibrées.
- **Le rapporteur** : Sert à rapporter les résultats des processus d'une fractale aux autres fractales. Les résultats sont des messages qui peuvent être de différents types : des sous-objectifs des sous-fractales, des réponses de négociation sur l'état courant de la super-fractale. Dans le cas où le Rapporteur se situe au plus bas niveau hiérarchique, la fractale est considérée comme un contrôleur d'équipement simple.

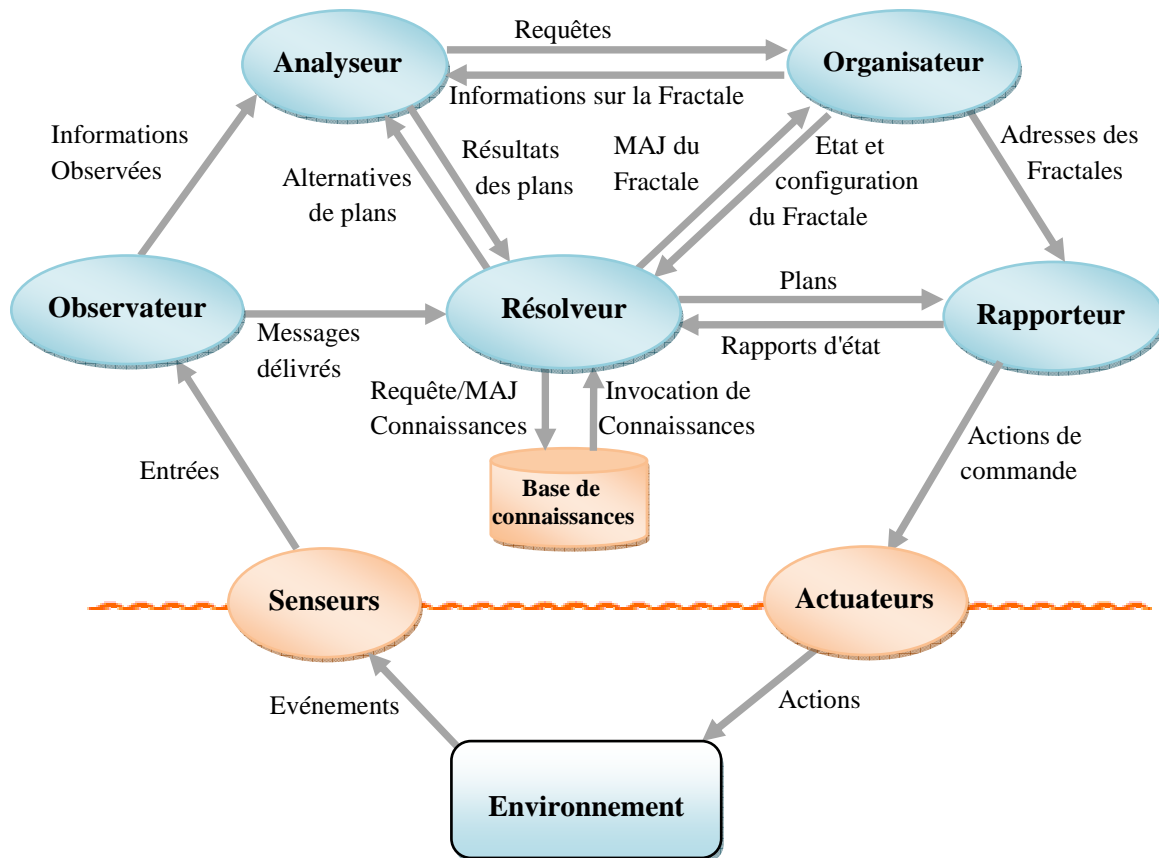


Figure. 3.8. Modules fonctionnels et leurs relations d'une Fractale dans un système de production fractal [Ryu et al, 2000]

3.4. Approches Multi-Agents :

3.4.1. Agent et Système Multi agent :

- **Agent :**

Il n'existe pas encore un consensus pour la définition d'un agent, car ce paradigme a été utilisé dans diverses disciplines et par diverses communautés scientifiques où chacune considère ce terme de son point de vue. En effet, plusieurs définitions peuvent être envisagées pour un agent. On se restreint à la définition classique de Ferber [Ferber, 1995] qui a considéré un agent comme une entité physique ou virtuelle possédante une autonomie de décision, capable d'agir sur elle-même ou sur son environnement, son comportement est influencé par ses observation et ses interactions avec d'autres agents, cherche à satisfaire ses objectifs en fonction de ses ressources et compétences.

- **Architectures d'un agent :**

On distingue trois grandes catégories d'agents :

Agents réactifs : Ces agents sont classifiés non intelligents. Un agent réactif ne possède pas une représentation de l'environnement, il réagit seulement à un stimuli externe sans compréhension de son environnement.

Agents cognitifs : Ces agents sont dotés d'une représentation symbolique ou logique de leurs environnements. De ce fait, un agent cognitif possède une rationalité lui permettant de prendre des décisions en fonction de ses buts explicites.

Agents hybrides : Ces agents combinent entre les deux architectures précédentes pour modéliser un système complexe.

- **Systèmes multi agents (SMA) :**

Un système multi agents (SMA) est un système composé de plusieurs agents évoluant dans un même environnement et dont l'interaction entre ces agents est de plusieurs modes (coopération, coordination, négociation, ...).

Un système multi agents peut être considéré selon quatre points de vue essentiels [Demazeau, 2001] :

- *Une vue Agent* : Architecture interne ainsi que les connaissances de l'agent.
- *Une vue Environnement* : l'espace partagé par l'ensemble d'agents dans le système.
- *Une vue Interaction* : Structure des relations dynamiques entre un ensemble d'agents, par exemple le langage de communication entre agent.
- *Une vue Organisation* : Le schéma de coopération entre les agents : groupe, rôle,...

Un système multi agent est composé des éléments suivants [Ferber, 1995] :

Un Environnement : L'espace commun partagé par les agents.

Un ensemble d'objets passifs et situés : Ces objets peuvent être perçues, créés, modifiés ou supprimés par les agents.

Un ensemble d'agents actifs : Ce sont des objets particuliers représentant les éléments essentiels du système multi agents.

Un ensemble de relations : unissant les objets (agents et objets) entre eux.

Un ensemble d'opérations : appliquées par les agents actifs aux objets passifs.

Un ensemble d'opérateurs : représentant l'application des opérations et la réaction du monde face à cette application.

3.4.2. La modélisation multi-Agents :

La modélisation multi agent *ABS (Agent Based Modeling)* consiste à concevoir un modèle représenté par un ensemble de types d'agents. Dans un système complexe, les systèmes multi agents permettent de modéliser des situations complexes où la structure globale d'un tel système émerge des interactions des entités constituant ce système [Ferber, 1995]. En plus, un système complexe nécessite parfois une décentralisation de décision, peut avoir des fortes interactions entre individus et entre individus et leurs environnement, possède un comportement non déterministe, ainsi les composants d'un système complexe peuvent être divers et hétérogènes.

En effet les systèmes multi agents - appliqués dans divers domaines et disciplines - offrent un moyen puissant pour modéliser les entités d'un système complexe ainsi que leurs interactions en exploitant leurs atouts (la modularité, l'autonomie, la flexibilité,...). L'application de la modélisation multi agents dans le domaine des systèmes de production sera abordée dans la section 3.4.4 ci-dessous.

3.4.3. De la modélisation à la simulation multi-Agents :

La simulation multi-agent correspond à l'exécution dynamique des modèles définis à base de systèmes multi-agents en les soumettant à des ensembles d'entrées et en analysant les sorties. On peut, de ce fait, raisonner sur les solutions globales qui émergent des comportements des agents locaux. Pour construire un modèle de simulation à base d'agents, Macal et North [Macal et North, 2006] ont défini les étapes suivantes :

- 1- Identifier les types d'agents désignés par le modèle.
- 2- Définir l'environnement dans lequel les agents vivent, évoluent et interagissent.
- 3- Spécifier les méthodes pouvant mettre à jour l'agent face à une interaction avec un autre agent ou avec l'environnement.
- 4- Définir les procédures d'interaction : quels agents interagissent et dans quel cadre spatio-temporel les agents interagissent pendant la simulation.
- 5- Implémenter le modèle de simulation dans une plateforme logicielle.

3.4.4. Applications des SMA dans les systèmes de production:

Les systèmes de production deviennent de plus en plus complexes, leur contrôle est décentralisé, les décisions prises doivent prendre en considération leurs contraintes et leurs objectifs (planification, contrôle, ordonnancement,...), les changements dynamiques et inattendus (pannes de machines, changement des gammes, arrivée des tâches urgents,...) [Kouis et al, 1997]. De ce fait, les propriétés caractérisant les SMA sont bien adaptées à cette classe de systèmes : La distribution des informations et des tâches de prise décision, la coopération et la coordination pour produire un comportement globale et collectif (propriété émergente), l'auto-organisation vis-à-vis des changements dynamiques de l'environnement. Dans un tel système, ces agents sont parfois dédiés aux différentes tâches de contrôle et de pilotage, et parfois sont associées à des ressources physiques (machines, équipements, matières premières,...).

L'adéquation des SMA vis-à-vis les systèmes de production et plus précisément les systèmes de production industriels est résumée par Parunak dans son article *What can Agents do in Industry, and Why?* [Parunak, 1998] en cinq caractéristiques communes principales :

- **Modulaire** : les agents sont des objets proactifs. Ils sont bien adaptés aux applications naturellement modulaires, ils partagent donc les intérêts de la modularité. Un système de production est décomposé en sous systèmes (modules), chacun avec ses propres variables d'état, son identité organisationnel ainsi que ses procédures qui l'interfacent avec son environnement.
- **Décentralisé** : Un agent possède une autonomie, et ne nécessite pas une invocation externe. Les agents son adaptés aux applications décomposables en processus spécifiques. Chaque processus peut s'exécuter sans aucune direction continue d'un autre processus. Ainsi, l'augmentation de la variation des produits a permet d'accroître le nombre et la complexité des outils de production. Dans ce cas les problèmes de contrôle et d'ordonnancement même dans une entreprise unique sont naturellement considérés d'un point de vue décentralisé.
- **Changeable** : La modularité permet de changer un seul module à la fois. La combinaison de cette propriété avec la décentralisation conduit à un couplage faible entre modules ce qui permet de réduire l'impact de changement d'un module sur les modules invoqués.

Avec un système de production fréquemment changeable où la variété de produits est toujours en augmentation et la minimisation du coût et de temps est fortement exigée, l'architecture basée-agent assure une auto-configuration et une réutilisation des composants existants du système en assurant une réduction du coût et de temps.

- **Mal-structuré :** Les systèmes de production modernes sont dynamiques, subissant des changements à tout moment durant tout le cycle de vie du système d'information. Si la structure d'un tel système qui est conçu pour un domaine particulier change, l'idée traditionnelle nécessite d'invoquer le concepteur initial, mais la technologie agent est capable de découvrir ces changements et peut s'adapter à cette structure sans changer la structure elle-même (on la réutilise) avec réduction de coût de maintenance et de reconfiguration.
- **Complexe :** Lorsque la complexité des produits augmente, la taille de l'espace de recherche pour le concepteur augmente, le nombre de machines à ordonnancer et à contrôler aussi augmente. Cette complexité rend donc la mise en correspondance entre la conception et le modèle de simulation plus difficile. Cette complexité est aussi engendrée par le nombre élevé d'interactions entre les éléments du système plutôt que le nombre d'éléments, ce qui la rend de nature combinatoire. En faisant correspondance ces éléments avec des agents individuels, le nombre très élevé des interactions se diminue en ne générant ces interactions qu'au moment de l'exécution ce qui permet de réduire la quantité de code de logiciel générée et donc le coût du système à construire.

3.4.5. Quelques travaux en relation :

La technologie agent est utilisée depuis longtemps pour la modélisation et la simulation des systèmes de production. Plusieurs recherches ont appliqué cette technologie vue son importance. De ce contexte, on peut citer quelques projets et travaux de recherches importants :

❖ **Les travaux de Kouis et al [Kouis et al, 1997] :**

Ces auteurs proposent une stratégie d'ordonnancement dynamique et distribuée dans un système de production flexible en utilisant l'architecture Multi-Agents. Dans cette architecture, chaque agent représente un centre de travail et sélectionne dynamiquement les

règles de priorité des tâches les plus adaptées. Une nouvelle règle de priorité est appliquée à chaque fois qu'un événement apparaît ce qui permet de s'adapter aux changements de l'état du système.

La sélection des règles de priorité par chaque agent se compose de deux étapes exploitant chacune une base de connaissances : la première étape permet de détecter un symptôme actif. Il s'agit d'un seuil prédéfini qui peut être dépassé (concernant le comportement global du système) et comme effet les objectifs du système peuvent dévier. La deuxième étape sert à choisir les règles de priorité à appliquer parmi un sous ensemble de règles présélectionnées.

La sélection des règles de priorité se fait localement par chaque agent, et pour maintenir l'objectif global, un agent superviseur est mise en place. Son rôle est de contrôler l'état global du système en détectant les symptômes globaux. Il impose parfois des règles de priorité particulières pour satisfaire les objectifs globaux.

Les agents affectés aux centres de travail sont chargés de l'ordonnancement des tâches existant dans ses propres files d'attente. Pour cela, les deux étapes citées précédemment sont appliquées tout en prenant en considération l'état du système, les ordres reçus de l'agent superviseur et les objectifs globaux du système de production.

Dans cette architecture chaque agent est constitué de trois couches :

- *Couche de connaissances statiques et sociales* : contient des connaissances sur l'agent lui-même et sur les agents de son environnement. Par exemple : des valeurs des seuils activant les symptômes prédéfinis, des données concernant les règles de priorité pouvant être sélectionnées, des données sur les capacités des autres agents.
- *Couche d'expertise* : C'est la couche intelligente de l'agent. Elle est constituée des connaissances sous forme de règles de production et permet à l'agent de détecter les symptômes prédéfinis et d'appliquer des règles de priorité sélectionnées en utilisant les données de la couche précédentes et ensuite envoyer des ordres aux agents concernés (agent superviseur ou agents affectées aux centres de travail).
- *Couche communication* : Regroupe l'ensemble des protocoles de communication entre les agents (par échange de message : KQML, ACL,...) et entre l'agent et les ressources physiques du système. Ces communications permettent à l'agent de contrôler des machines, d'extraire

des informations de la base de données du système, des requêtes d'application des règles de priorité.

❖ **Les travaux de Babiceanu et Chen [Babiceanu et Chen, 2005] :**

Ces travaux consistent à la proposition d'un système décentralisé d'ordonnancement en temps réel des opérations de la manutention des matériaux basé sur l'architecture multi-agents dans un environnement de production cellulaire. Cette architecture est composée de trois types d'agents possédant une autonomie de prise décision grâce aux algorithmes d'évaluation y intégrés : Agent de Manutention de Matériel (MHA : *Material Handling Agent*), Agent Ordre (OA : *Order Agent*), Agent Vue Globale (GVA : *Global View Agent*) ainsi qu'une entité de contrôle disposant d'une base de données pour le contrôle du système (SMD : *System Monitoring Database*).

- *Le MHA* : Chaque ressource est contrôlée par son propre MHA. Ce dernier doit suivre l'état de cette ressource et répondre en temps réel en cas d'arrivée d'un événement inattendu (par exemple panne de la ressource). Il communique avec les entités du système par l'envoi des différents types de messages : réception des requêtes, diffusions des informations,...

- *Le OA* : Cet agent est attribué à chaque nouvel ordre entrant dans le système. Chaque OA utilise des algorithmes d'évaluation, d'allocation, et d'exécution pour générer des ordonnancements individuels (*IS : Individual Schedule*). Les IS générés par les OAs actifs ainsi que les états inattendus des ressources sont transférés vers le GVA.

- *Le GVA* : C'est un agent possédant une vue globale du système. Il reçoit les ordonnancements individuels issus des OAs actifs. Ensuite il arrange ces ISs en utilisant son perspective globale de telle sorte de satisfaire l'objectif global du système. Il obtient en résultat un ordonnancement émergent global du niveau système appelé SLS pour *System Level Schedule* qui est diffusé vers toutes les entités du système.

- *Le SMD* : Reçoit - en temps réel - toutes les informations relatives aux changements d'état des ressources issues des différentes entités du système tel que les pannes, l'ajout, la suppression des ressources. Ces informations sont récupérées en temps réel par les entités concernées afin de réagir par les actions appropriées.

L'objectif principal de cette approche est de minimiser le temps total d'exécution (appelé le *makespan*) des opérations de manutention selon l'ordonnancement obtenu par cette architecture. Pour évaluer ses performances, cette approche est comparée par l'approche conventionnelle centralisée où le GVA s'occupe de la prise de toutes les décisions concernant l'allocation des tâches de manutention aux ressources. Pour cela, un intervalle de confiance est construit pour comparer le makespan obtenu par les des deux approches. Le temps inter arrivées de deux tâches, le temps où une ressource tombe en panne, le temps de reprise après panne, ainsi que le temps de traitement d'une tâches, sont générés aléatoirement et suivant des différentes lois de distribution. Des simulations sont effectuées utilisant un algorithme permettant de calculer le makespan après une série d'exécutions et utilisant toujours la génération aléatoire des valeurs des paramètres cités précédemment. Le makespan est calculée autant de fois pour les deux approches (pour construire l'intervalle de confiance).

Avec l'approche à base d'agents, la tolérance aux pannes est bien illustrée lorsque les tâches affectées à une ressource en panne sont réordonnées vers d'autres ressources disponibles dans le système sans impact sur la qualité des résultats. L'approche est comparée avec l'algorithme optimal d'ordonnancement (embarqué dans le GVA) en terme de temps CPU nécessaire pour obtenir une solution faisable et les résultats de simulations montrent que la solution délivrée est en très proche. En effectuant une comparaison, cette approche est un peu mieux que les algorithmes heuristiques développés pour l'approche globale en terme de temps CPU nécessaire pour donner une solution faisable. Cette différence est un résultat de la décentralisation des décisions prises par l'approche à base d'agents qui requiert des algorithmes simples sans explosions combinatoires.

❖ **MANPro : Mobile Agent-based Negotiation Process for distributed intelligent manufacturing :**

Shin et Jung [Shin et Jung, 2004] ont proposé un mécanisme de négociation orienté-ressource basé sur les agents mobiles dans un système de contrôle d'atelier de production distribué. Ce mécanisme décrit deux architectures pour la négociation entre agents : une architecture de communication et une architecture d'information.

L'architecture de communication : exploite les fonctionnalités de communication de base du protocole *contract net* [Smith, 1980] pour résoudre les problèmes de distribution. Cette architecture introduit des agents représentant les tâches et les ressources et un agent de

coordination. Elle est composée de quatre modules fonctionnels : 1) *Le Gestionnaire des offres (BM pour Bid Manager)* : permet de contrôler et de coordonner le processus global de négociation entre agents. Ce processus englobe un ensemble de processus. Le BM initialise ce processus, crée les agents Tache et sélectionne quel processus de négociation à initialiser dans le temps approprié entre l'agent Tache et l'agent Ressource. Avant la création d'un agent Tache, le BM doit contrôler les conflits qui apparaissent lorsque par exemple un agent Ressource est entrain de négocier avec plusieurs agents Tache qui peuvent sélectionner la même offre proposée dans le même temps. 2) *L'agent Tache (T-agent)* : correspond à une tache unique créée par le BM et dispose de toutes les informations concernant le processus de négociation. Il crée les agents Négociation qui parcourent les agents Ressource invoqués dans le processus de négociation en collectant les offres. Ensuite, il évalue ces offres, sélectionne une et envoie l'attribution de l'offre à l'agent Ressource concerné. 3) *L'agent Négociation (N-agent)* : C'est un agent mobile capable de parcourir le réseau des contrôleurs de ressources selon un ordonnancement de routage, surveille l'état des ressources, génère les offres et ensuite rapporte la liste des offres à l'agent Tache après une pré-évaluation en libérant quelques ressources proposant des offres jugées non performantes. 4) *L'agent ressource R-Agent* : attaché à une ressource et sert à contrôler celle-ci. Il offre au N-agent un accès contrôlé à la base de données d'une ressource et reçoit l'attribution de l'offre provenant du T-agent et envoie dans le sens inverse l'acceptation.

L'architecture d'information : offre des mécanismes d'échange entre des modèles d'information hétérogènes. Pour cela, MANPro propose un modèle d'information à base d'ontologies incluant diverses ontologies (description des tâches, messages de négociations,...) afin de représenter des connaissances spécifiques. Ce modèle est basé sur le langage DAML + OIL [Harmelen et al. 2001].

❖ **MetaMorph I : An Adaptive Agent-Based Architecture for Intelligent Manufacturing :**

MetaMorph [Maturana et al, 1999] est un projet développé à l'université de Calgary. Maturana et al ont proposé une architecture adaptative multi-agents à base de médiateurs pour améliorer la coopération et la coordination entre les agents intelligents dans un système de production intelligent et distribué. Cette architecture regroupe deux sortes d'agents hybrides (réactifs et délibératifs) : des agents Ressource et des agents Médiateurs (appelés Médiateurs). Les agents Ressources sont des contrôleurs des ressources physiques (machines,

robots,...) et des opérations effectuées par celles-ci. Ils utilisent un mécanisme de clonage - quand il est nécessaire - pour créer d'autres agents et peuvent donc créer des groupes dynamiques. Les Médiateurs servent à aider les agents Ressource pour trouver d'autres agents et coordonnent leurs interactions. La coordination se fait à haut niveau (entre groupes) et à bas niveau (dans un même groupe) et utilise deux mécanismes : la décomposition des tâches et la création des groupes virtuels. Les Médiateurs sont modélisés par sept domaines d'activités : Entreprise, Spécification et Conception de Produits, des Organisations Virtuelles, Planification et Ordonnancement, Exécution, Communication et Apprentissage. L'apprentissage est un élément important dans MetaMorph I ; ce dernier permet d'améliorer les performances du système et pour cela deux mécanismes sont implémentés : le premier basé sur l'approche d'apprentissage à *base de cas* distribuée permet aux Médiateurs d'apprendre à partir de l'historique des comportements et des interactions entre agents. Le deuxième est basé sur la simulation du futur comportement du système en faisant subir le système à des perturbations inattendues et des changements des priorités de production.

❖ **MetaMorph II : An Agent-Based Architecture for Distributed Intelligent Design and Manufacturing :**

Ce projet était l'extension du projet MetaMorph I. Il utilise une architecture hybride à base d'agents médiateurs hiérarchiques afin d'intégrer dynamiquement les activités de l'entreprise avec celles : des clients, des fournisseurs et des partenaires dans un environnement ouvert et distribué [Shen et al, 2000]. Dans cette architecture, le système de production est décomposé hiérarchiquement en sous systèmes. Un sous-système peut communiquer avec le système global, avec un autre sous système ou avec un agent d'autre sous-système par le biais des médiateurs spéciaux. Les agents dans MetaMorph II peuvent représenter soit des ressources de production (outils, machines,...) soit des coordinateurs de systèmes/sous-systèmes (médiateurs). Les Médiateurs sont de différents types : Médiateurs d'Entreprise, Médiateur de Conception, Médiateur de Ressources (machines, outils, ouvriers,...), Médiateur de Marketing, Médiateur de Simulation et Médiateurs d'exécution. Un Médiateur est un agent constituant composé de différents types de connaissances, une interface réseau, une interface de communication, des mécanismes d'apprentissage et de raisonnement ainsi que des mécanismes de contrôle.

❖ Les travaux de Aissani et al :

Aissani et al. [Aissani et al., 2009] ont proposé une approche réactive multi-agents pour l'ordonnancement dynamique des tâches de maintenance dans un système de production pétrolier (la raffinerie d'Arzew - Algérie). L'architecture proposée est organisée hiérarchiquement en un ensemble d'agents capables de réagir en temps approprié sans affecter les performances globales du système. Pour cela, chaque agent de cette architecture est doté d'un module d'apprentissage par renforcement utilisant l'algorithme SARSA. Cette approche considère l'ordonnancement des tâches de production comme donné et ensuite les tâches de maintenance (préventive et curative) sont intégrées dynamiquement afin d'avoir un ordonnancement global production/maintenance. L'architecture générale du système est composée de trois types d'agents : Agent ressource, Agents Produit et Agent Observateur. Ces agents utilisant la technique d'apprentissage par renforcement sont capables de s'adapter et d'apprendre les meilleurs comportements de leurs rôles sans avoir affecté ni les performances du système, ni sa réactivité.

❖ Travaux de Kouider et Bouzouia :

Les auteurs [Kouider et Bouzouia, 2012] proposent une architecture multi agents distribuée basée sur la coopération entre agents dans un système de production de type job shop. L'objectif de cette architecture était de minimiser le temps d'inoccupation des ressources du système et en assurant un temps minimal du makespan. Cette architecture est composée de deux agents : Agent Superviseur et Agent Ressource. L'Agent Superviseur permet de décomposer le problème d'ordonnancement de l'atelier en sous problèmes. Chaque sous-problème est envoyé à l'Agent Ressource concerné. La technique de décomposition qui est le cœur de cette architecture consiste à construire – en premier lieu – un graphe disjonctif. Ce graphe est décomposé en sous graphes représentant chacun un sous problème. Pour résoudre un tel sous problème, un agent Ressource doit communiquer avec les agents Ressources voisins (pour satisfaire les contraintes de précédence : échange des temps de fin des opérations) et en assignant une certaine valeur de priorité pour chaque opération. Les ordonnancements locaux établis par les agents Ressource sont envoyés vers l'agent Superviseur. Pour valider cette approche, celle-ci est évaluée en terme de temps CPU pour établir un ordonnancement optimal dans le cas d'ordonnancement statique, et est comparée avec certaines règles de priorité dans le cas d'ordonnancement dynamique.

3.4.6. Synthèse :

La littérature fait référence à un nombre croissant des travaux utilisant l'approche agent pour la modélisation et la simulation des systèmes de production. Les travaux de recherche cités précédemment peuvent être comparés dans le tableau 3.1. Chaque projet ou travail de recherche utilise cette approche d'une façon spécifique pour modéliser/simuler divers types de système de production et tendant à atteindre différents objectifs.

Dans ces travaux, les acteurs composant un système de production sont représentés par des agents. La structure, l'architecture des agents et les modes d'interactions entre agents se diversifient d'un projet/travail à l'autre. La plus part de ces travaux se base sur une structure décentralisée et exploitant la capacité des agents (autonomie de décision, l'émergence, l'apprentissage...). Chacun de ces projets/travaux dispose de ses propres caractéristiques qu'on peut résumer la totalité comme suit :

- La prise de décision décentralisée et dynamique (en temps réel).
- La flexibilité et l'adaptation avec l'environnement.
- La réactivité face aux éventuelles perturbations et événements inattendus.
- L'apprentissage des entités à partir de l'état du système des interactions entre ces entités.
- L'interopérabilité entre les environnements hétérogènes.

Projets/travaux	Type de système de production à modéliser	Objectifs/problématiques	Modélisation (types d'agents)	Caractéristiques de base
Kuiss et al	Système de production flexible	Ordonnancement dynamique des règles de priorité	- Un Agent superviseur - Des agents locaux pour la sélection des règles de priorités	Sélection dynamique des règles de priorité
Babiceanu et Chen	Système de production cellulaire	Minimiser le temps total d'accomplissement des tâches (le makespan) de manutention des matériaux.	- Agent vue globale - n agents ordre - m agents manutention des matériaux. - une système possédant une BDD pour le contrôle du système.	- Décisions décentralisées d'allocation des tâches de manutention aux ressources disponibles. - Flexibilité : l'ajout et la suppression des ressources, reprise après pannes.
MANPro	Atelier de production distribué	Résoudre le problème de négociation entre des environnements hétérogènes, minimiser le flux d'informations dans le processus de négociation	- Une architecture de communication contenant (Un Gestionnaire des offres, un agent Tache, un agent négociation et un agent Ressource). - Une architecture d'information : un modèle d'information à base d'ontologies.	Un agent mobile parcourt le réseau pour collecter l'information au lieu que chaque agent possède une vue globale du système.
MethaMorph I	Système de production intelligent et distribué (multi-ateliers)	- Minimiser l'overhead dans la communication inter-agents. - faciliter la communication, la coordination et la coopération entre agents. - Améliorer les performances du système.	- Agent Ressource. - Agent Médiateur : <i>Médiateurs à haut niveau, Médiateur à bas niveau.</i> - Gestionnaires de données des Agents.	- Architecture à base de médiateurs-centrique. - décomposition des tâches - groupage et clonage dynamique - Adaptation avec les tâches urgentes et le changement d'environnement. - Apprentissage
MethaMorph II	Chaîne d'approvisionnement distribuée	- Communication des agents d'un sous-système avec d'autres sous-systèmes ou avec des agents d'un autre sous-système. - Résoudre le problème de goulot et l'atténuer par la communication directe entre agents.	- Agent ressource. - Agents médiateurs : <i>Des médiateurs de plusieurs types : d'entreprise, de conception,...</i>	- Architecture à base de médiateurs. - Intégration des activités de l'entreprise avec celles des clients, des fournisseurs et des partenaires. - Apprentissage
Aissani et al	Système de production à flux continu	- Réduire l'effet des pannes des ressources. - Minimiser le temps de maintenance. - Garder la capacité des stocks non nulles.	- Agent Produit. - Agent Ressource. - Agent Observateur.	- Architecture adaptative et réactive. - Apprentissage par renforcement.
Kouider et Bouzouia	Système de production job shop distribué	- Minimiser le temps d'occupation des ressources. - Minimiser le makespan	- Agent Superviseur - Agent Ressource	- Décomposition du problème d'ordonnancement en sous problème. - Solution coopérative

Tableau 3.1. Tableau récapitulatif des travaux multi agents pour la modélisation et la simulation des systèmes de production.

4. Conclusion :

Au cours de ce chapitre nous avons présenté l'utilisation de la modélisation et la simulation dans les systèmes de production en commençant par une description des concepts utilisés dans un tel modèle. Nous avons ensuite présenté les approches hétérarchiques les plus utilisées dans la littérature pour la modélisation/simulation des systèmes de production à savoir les approches holoniques, bioniques, fractales et multi agents. L'approche multi agent qui constitue notre choix pour modéliser/simuler les systèmes de production est illustrée par quelques projets et travaux de recherche. La plus part de ces travaux se concentre sur le contrôle et le pilotage des SdP et présentent peu d'interaction avec l'utilisateur. Ainsi ces travaux ne prennent pas en considération d'une part l'optimisation dans l'utilisation des ressources d'un système de production, et l'analyse des indicateurs de performance d'autre part. Dans le chapitre suivant on présentera un système multi agent pour le pilotage en temps réel d'un système de production de type job-shop en proposant un algorithme d'allocation dynamique des ressources et en offrant une interaction avec l'utilisateur et en essayant - dans une certaine mesure – d'offrir un moyen d'analyser et d'évaluer les performances de ce type de système.

CHAPITRE 4

Modélisation et Simulation multi-agents d'un système de production de type Job-Shop : Application au système de pilotage

1. Introduction :

Dans ce chapitre nous présentons une approche multi-agent distribuée supervisée pour le pilotage en temps réel (ou encore appelé pilotage à ordonnancement dynamique) d'un système de production de type Job-Shop.

Olivier-Maget a défini le pilotage en temps réel par : « *Le pilotage en temps réel consiste à exécuter la séquence opératoire de chaque recette et assure en temps réel la gestion des ressources nécessaires à cette exécution. Pour cela, il met en œuvre en temps réel les ordres de fabrication (consigne) provenant de la fonction ordonnancement en traduisant cette information sous forme de tâche de commande et en la distribuant aux automates et contrôleurs locaux* » [Olivier-Maget, 2007].

L'approche hétérarchique proposée se base sur les activités d'ordonnancement, d'allocation dynamique de ressources et de réaction en cas de panne. Le système peut être subit à des événements inattendus : l'arrivée des commandes urgentes, les pannes des ressources de production, rupture de stock, ... Dans ces cas, le système doit réagir en temps réel en arrangeant de l'ordonnancement. Ce réordonnancement doit assurer toujours les objectifs principaux du système.

2. Le système étudié :

Le système étudié est un système de production de Job-Shop (figure 4.1) composé d'un ensemble de ressources (machines, robots, stocks, ..) et exécutant un ensemble de Jobs. Dans chaque Job les produits passent d'une ressource à l'autre pour une durée déterminée et selon un certain ordre déterminé par la gamme de production. A un instant donné, chaque Job ne peut être traité que par une seule ressource, ainsi, chaque ressource ne peut exécuter qu'un seul Job à la fois. A chaque Job est associée une date de fin souhaitée (*due date*). Une fois une opération commence, elle n'est plus interrompue.

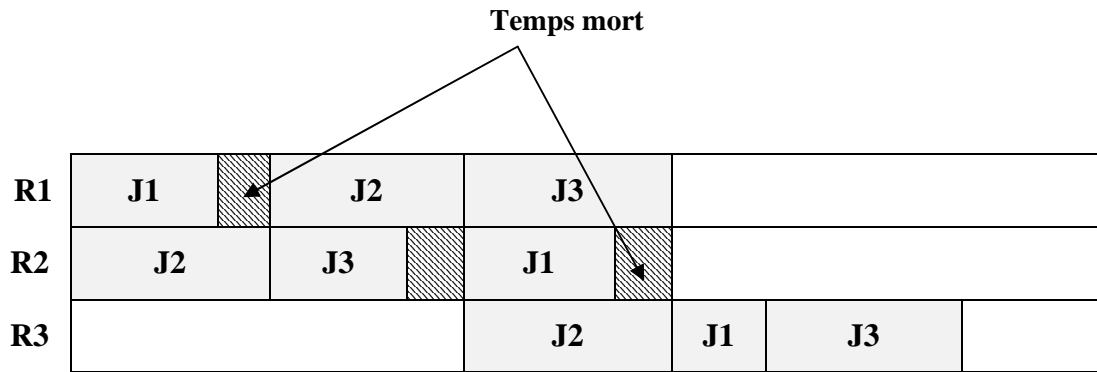


Figure.4.1. Un problème de type job-shop avec 3 Jobs et 3 Ressources.

3. Objectifs :

Dans ce travail, on introduit un système distribué composé d'un ensemble d'agents autonomes assurant l'ordonnancement dynamique, l'allocation de ressources et le contrôle d'un système de production job-shop.

Les agents du système réagissent de façon autonome en essayant de satisfaire ses propres objectifs. Le système multi-agent proposé prend en considération les tâches de maintenance curative, c'est-à-dire les événements liés aux pannes inattendues de ressources. Les tâches de maintenance préventives ne sont pas considérées.

À cet effet, Le système de pilotage proposé doit assurer les objectifs suivants :

- Tolérance aux pannes.
- Réagir face aux événements pouvant perturber le système de production : (pannes des ressources, remplacement des ressources après leur réparation, arrivée des jobs urgents).
- Capacité du système de s'organiser et de s'adapter dans le cas de changement de l'environnement.
- Emergence : les agents autonomes doivent toujours satisfaire l'objectif global du système.

En plus de ses objectifs, l'utilisateur a la possibilité de cibler un objectif, par exemple :

- Minimiser le temps total d'accomplissement des jobs (le Makespan).
- Minimiser la somme pondéré des retards.

- Minimiser le nombre de jobs en retard (dépassant leurs *due date*).

Pour cela, le système utilise des règles de priorité (*dispatching rules*) définies dans la littérature. Parmi ces règles, on peut citer :

- *SPT* : La priorité élevée est donnée à l'opération ayant le plus court temps opératoire.
- *LPT* : La priorité élevée est donnée à l'opération ayant le plus long temps opératoire.
- *EDD* : La priorité élevée est donnée à l'opération appartenant au job ayant le plus tôt date de fin souhaitée (*Due Date*).

Une règle ou une combinaison de règles de priorité adéquate (satisfaisant l'objectif de l'utilisateur) est ensuite choisie dynamiquement selon une expérimentation.

4. Structure conceptuelle à base d'agent du système étudié :

4.1. Structure de pilotage considérée :

Comme vue dans le chapitre III, plusieurs structures de pilotage sont envisageables. Dans ce mémoire on considère une structure de pilotage distribuée supervisée (figure 4.2) où un ensemble d'entités de décisions sont distribuées et ayant le même niveau hiérarchique, avec une entité qui supervise le système.

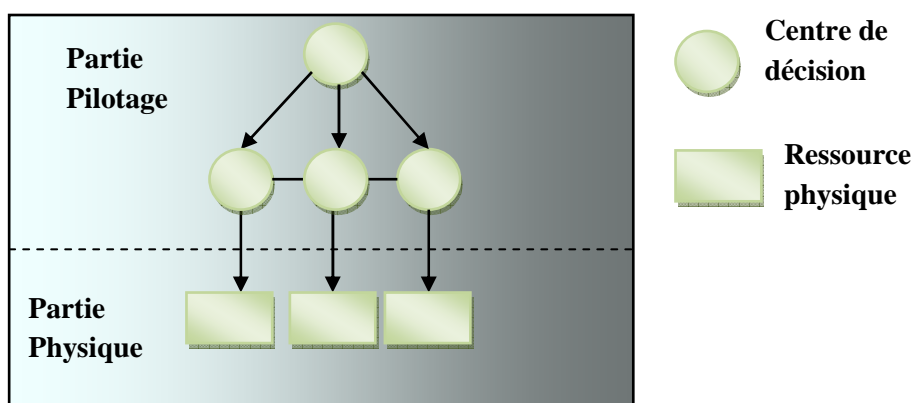


Figure.4.2. Structure du système de pilotage considérée.

Cette structure est caractérisée par :

- Chaque entité dispose une autonomie de prise de décision et doit en même temps satisfaire l'objectif global du système.
- Les entités peuvent communiquer entre elles par l'échange de messages d'intercommunication.
- Chaque entité dispose de ses propres compétences, données et connaissances.

4.2. Architecture générale multi-agent du système étudié :

Dans cette section on introduit une architecture multi-agents pour le pilotage en temps réel d'un système de production de type job-shop. Cette architecture est constituée d'un ensemble d'agents (réactifs et cognitifs) autonomes et coopératifs ainsi qu'une base de données centrale.

Le système est constitué d'un ensemble d'agents et une base de données centrale (figure 4.3) :

- Agent Superviseur :

Contient des configurations et des consignes initiales du système définies par les décideurs. Il inscrit les jobs arrivés et suit en temps réel l'état global du système (état d'accomplissement de ces jobs, pannes des ressources, changement de gammes, fin de réparation des ressources,...) et envoie des comptes rendus aux décideurs.

L'agent superviseur est composé des modules suivants (figure 4.4) :

Module Communication : Permet à l'agent Superviseur de communiquer avec les autres agents du système par envoi de message.

Module Evaluation de performances : Ce module assure que les objectifs du système soient atteints à travers un ensemble d'indicateurs de performance déterminés.

Module Décisionnel : Permet de prendre des décisions (application des plans d'actions) en fonction de l'état actuel du système et des informations reçues à travers le module communication. Il gère aussi la base de données centrale : inscriptions/désinscription

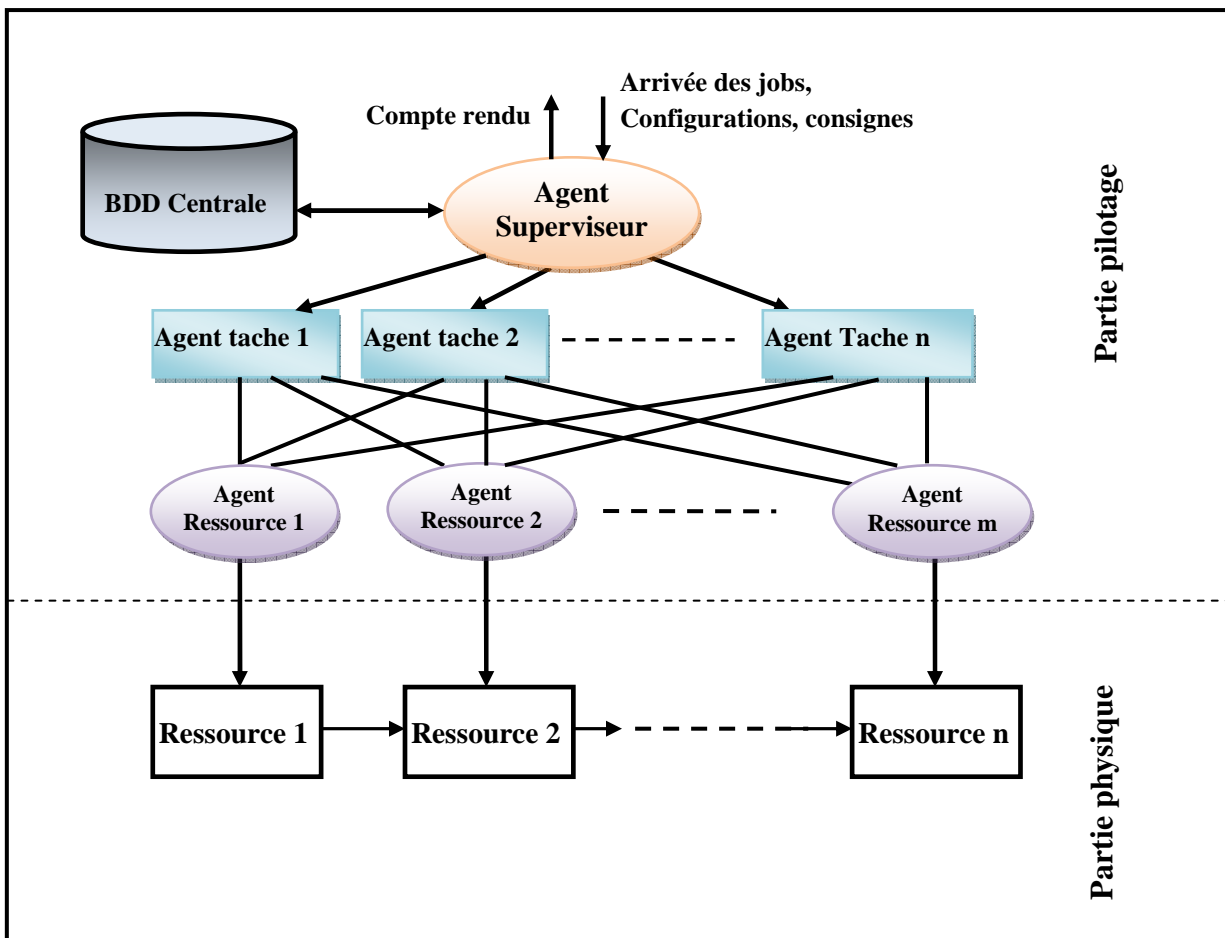


Figure 4.3. Architecture générale multi-agent du système étudié

des ressources avec leurs états, mise à jour de la lise de jobs.

- Agent Ressource :

Chaque ressource est contrôlée par un Agent Ressource. Ce dernier contrôle l'état de la ressource, lance les opérations planifiées par l'agent Tache, gère la base de données locale et envoie des compte rendus à l'Agent Tache (signale la fin d'une opération, l'état d'accomplissement d'une opération, panne de la ressource, ...). Un agent Ressource est composé des modules suivants (figure 4.5) :

Module Perception : Permet d'obtenir l'état de la ressource (active, en panne, libre) ainsi que l'état d'avancement d'achèvement des opérations.

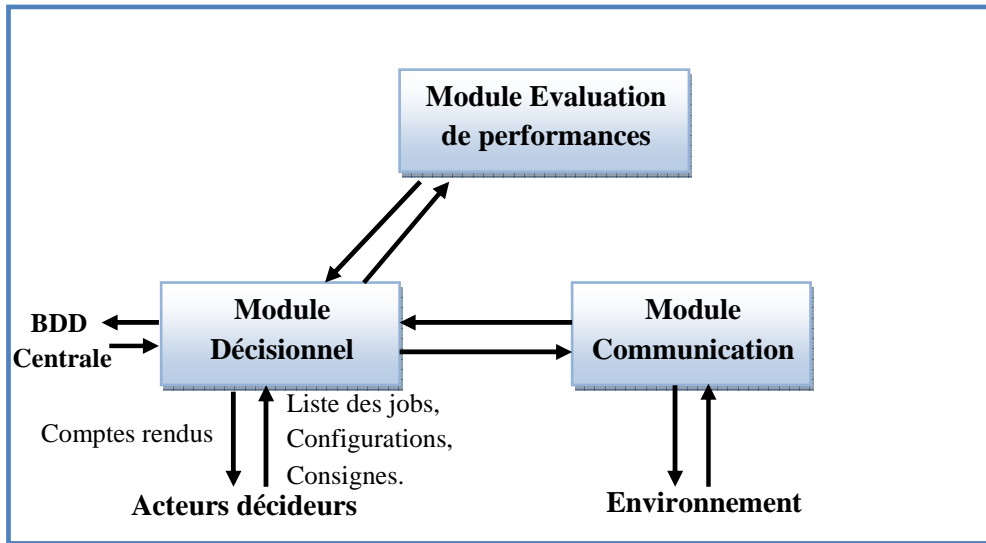


Figure 4.4. Architecture modulaire de l'agent Superviseur

Module Communication : Permet à l'agent Ressource de communiquer avec les autres agents du système par envoi de message.

Module Décisionnel : Sert à réagir en fonction des informations obtenues à travers le module communication et des événements reçus.

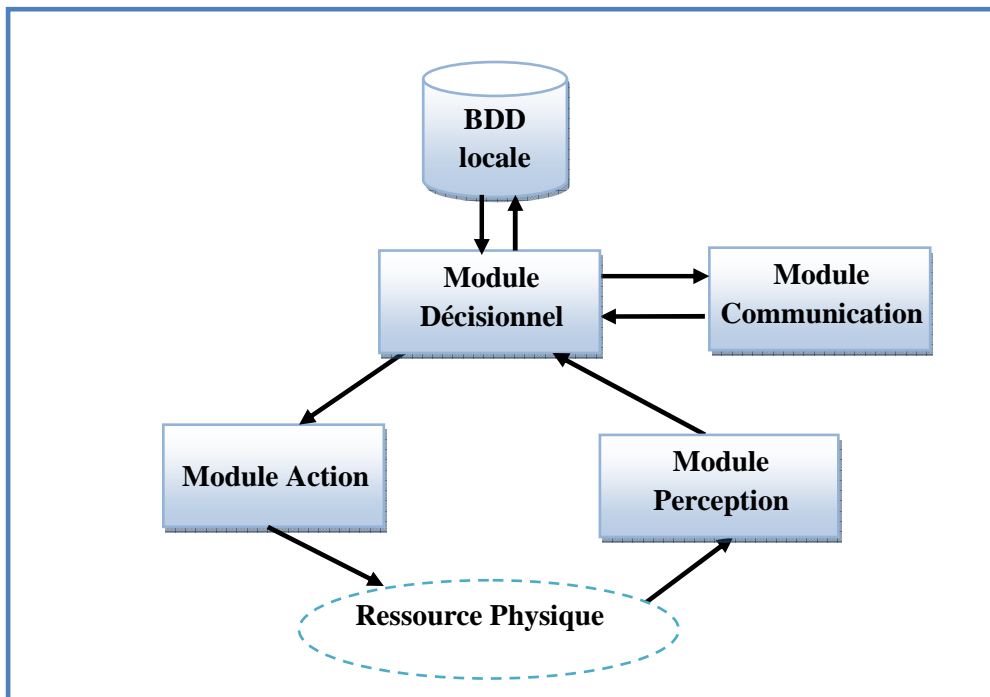


Figure 4.5. Architecture modulaire de l'agent Ressource

Module Action : Ce module lance l'exécution des opérations sur la ressource concernée.

- Agent Tache :

Un agent tache est créé à chaque arrivée d'un job et détruit après l'allocation de toutes les opérations du job et l'envoi des informations à l'agent Superviseur. L'agent Tache est responsable au suivi de l'ordonnancement des taches et à leur exécution. Sa mission principale est l'allocation de ces taches aux ressources concernées et le suivi de leur exécution. Cet agent assure la bonne exécution des taches (respect des gammes, des délais) et envoie à l'agent superviseur des informations liées à l'état d'accomplissement des taches (les dates de début, les dates de fin, ...). Il permet aussi de résoudre les problèmes liés à l'exécution des taches par exemple, il réalloue une tache à une autre ressource si celle-ci tombe en panne, il change l'ordonnancement en cas de changement de priorité. L'architecture modulaire de l'agent tache est définie comme suit (figure 4.6) :

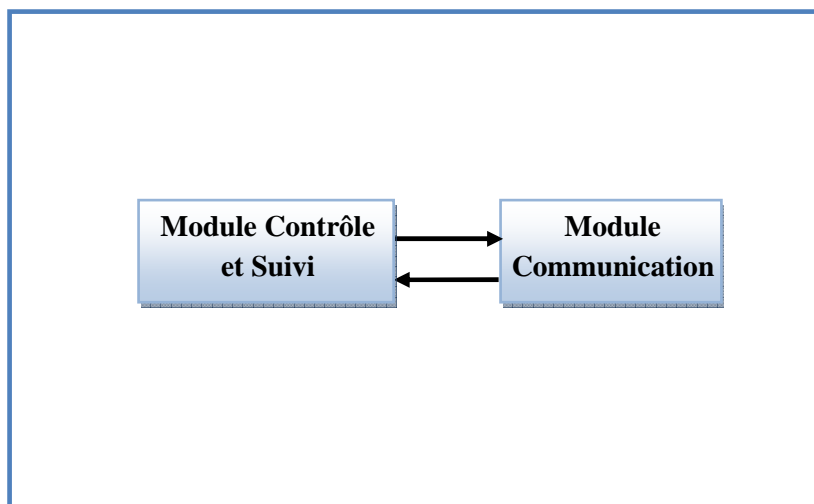


Figure 4.6. Architecture modulaire de l'agent Tache

Module Communication : Permet à l'agent Tache de communiquer avec les autres agents du système par envoi de message.

Module Contrôle et Suivi : Ce module enregistre les Jobs et leurs opérations (avec les durées opératoires), alloue et réalloue les opérations aux ressources, assure et suit en temps réel et la bonne exécution des opérations en communiquant avec l'agent Ressource et (en lui envoyant

par exemple des ordres de fabrication élémentaires), et envoie toutes les informations liées à l'exécution des tâches (par exemple la date fin d'une tâche) à l'agent superviseur.

- Base de Données Centrale :

Contient toutes les informations relatives aux Jobs, leurs durées d'exécution, la liste des ressources, les gammes. Cette base de données est mise à jour en temps réel, par exemple si une ressource tombe en panne, celle-ci est désinscrite, si une autre devient disponible, elle sera inscrite...

5. Fonctionnement de l'approche multi agent :

5.1. Description formelle du système étudié : Comme cité dans la section 1, le système étudié est un système de production de type job-shop. Le système est constitué de m Ressources (Machines, Robots, Stocks, Chariots) notées par $\{R_1, R_2, \dots, R_m\}$ et de n Job notés par $\{J_1, J_2, \dots, J_n\}$. Chaque Job -constitué de m d'opérations- est exécuté sur l'ensemble des m ressources selon un ordre bien défini (selon la gamme). On notera :

- Chaque opération est un septuplet $\langle J, R, T_d, T_f, T_{exec}, V, Ordre \rangle$, tel que :

J : l'indice (identifiant) du job que l'opération appartient.

R : la ressource où l'opération est exécutée. Celle-ci est un paire $\langle Id, Etat \rangle$ tel que :
 Id : numéro de la ressource

$Etat$: état de la ressource {libre, occupée, panne}

T_d : date de début de l'opération.

T_f : date de fin de l'opération.

T_{exec} : temps d'exécution de l'opération.

V : une valeur entière déterminant la priorité de l'opération par une règle de priorité.

$Ordre$: Ordre de l'opération de la gamme (Job).

- Chaque Job J_i ($i=1$ à n) est un triplet $\langle Op_courante, Op_succ, DD \rangle$ tel que :

$Op_courante$: l'opération actuelle attendant l'allocation à une ressource.

Op_succ : l'opération suivante de l'opération courante dans le job. Tel que

$Op_succ = null$ si l'opération est la dernière opération dans le job.

DD : date de fin souhaitée (*Due Date*)

On cherche une allocation des jobs aux différentes ressources avec prise en compte des événements inattendus (par exemple les pannes de ressources) avec résolution des conflits par les règles de priorité. L'utilisateur peut choisir ses critères de performances en appliquant la règle adéquate.

5.2. Algorithme d'allocation de ressources :

Etape 1 : { Initialisation }

Pour chaque Job i ($i = 1$ à n) **faire**

$J_i.op_courante := null$

$J_i.op_succ :=$ première opération

Fin pour

Pour chaque ressource R_i ($i=1$ à m) **Faire**

$R_i.etat :=$ libre

Fin pour

Initialiser la BDD centrale // liste des ressources, leurs états

Etape 2 : { Allocation des premières opérations de chaque job }

Pour chaque Job i ($i = 1$ à n) **faire**

$Alloc(J_i.op_succ)$

Fin pour

Etape 3 :

Attendre jusqu'à réception d'un message de l'agent Ressource indiquant la fin d'une opération dans une ressource, soit R_k

Pour chaque job J_i ($i = 1$ à n) **Faire**

Si ($J_i.op_succ.R = R_k$) et ($J_i.op_courante.T_{fin} \leq$ temps courant) **alors**

$Alloc(J_i.op_succ)$

Finsi

S'il existe un job J_i *tel que* $J_i.op_courante.T_{fin} =$ temps courant **alors**

$Alloc(J_i.op_succ)$

Fin pour

Etape 4 :

Pour tous les jobs J_i ($i=1$ à n) **Faire**

S'il existe des jobs tel que $J_i.Succ \neq null$ alors

Aller à **Etape 3**

Sinon

Aller à **Etape 5**

Finsi

Fin pour

Etape 5 :

Enregistrer les *dates début* et les *dates fin* et les envoyer à l'agent

Superviseur

MAJ de la BDD centrale

Calcul de performances

Fin.

// Procédure d'allocation

Procédure Alloc (op : opération)

Si op.R.etat := libre **alors**

Résolution de conflit par la règle de priorité utilisée

Allouer la ressource

op.R.etat := occupée

Op.T_d := max (temps courant, J_{op.J}.Op_courante.T_f)

Op.T_f := Op.T_d + Op.T_{exec}

J_{op.J}.Op_courante := J_{op.J}.op_succ

Enregistrer T_d, T_f.

Sinon

Si op.R.etat = panne **alors**

Refuser la demande d'allocation

Résoudre le problème de panne

Attendre jusqu'à réparation de la ressource.

Finsi

Finsi

Obtention des résultats : On peut obtenir plusieurs résultats par exemple : Le makespan est la date de fin d'exécution de la dernière opération, la somme pondérée des retards est la somme des temps d'inoccupation de chaque ressource multipliée par des coefficients.

5.3. Sources d'incertitude : Dans cette architecture, le système de production utilise certaines variables d'environnement stochastiques. Ces variables peuvent suivre différentes lois de distribution (normale, exponentielle, uniforme,...). Ces variables sont :

MTBA : temps moyen entre deux arrivées de jobs

MTBF : temps moyen entre deux pannes :

TTO : temps opératoire (d'exécution) d'une opération :

MTTR : temps moyen de réparation d'une ressource :

5.4. Interaction entre agents du système :

Les algorithmes proposés n'exhibent pas bien l'interaction entre les différents agents. De ce fait, l'interaction entre les agents de notre système se fait par envoi de message en utilisant le langage de communication FIPA-ACL [FIPA, 2002]. Un message FIPA-ACL est composé d'un ensemble de paramètres :

Performative : Le type de l'acte communicative (le type de message).

Sender : Le nom de l'agent émetteur du message.

Receiver : Le nom de l'agent récepteur du message

Content : Exprime le contenu du message.

Reply-with : Introduit une expression aidant le récepteur pour bien comprendre le message original.

In-reply-to : Indique une expression qui fait référence à une action précédente auquel ce message répond.

Protocol : indique le protocole utilisé par l'agent émetteur du message.

Langage : le langage dans lequel le contenu de l'action est exprimé.

Ontology : indique l'ontologie donnant expliquant les symboles utilisé dans l'expression du content.

conversation-id : une expression utilisé pour identifier la séquence en cours d'actes de communication et qui constituent ensemble une conversation.

Le seul paramètre obligatoire dans un message FIPA-ACL est *Performative*, bien que tout message doit comprendre au moins les deux paramètres *Sender* et *Receiver*. Le tableau 4.1 illustre quelques performatives :

Performative	Description
Request	Demande de l'exécution d'une action
Query-If	Demande si une proposition donnée est vraie
Inform	Informé l'agent qu'une proposition est vraie
Query-Ref	Demande à un agent une information sur un objet référencé
Confirm	Confirmer qu'une proposition est vraie
Disconfirm	Confirmer qu'une proposition est fausse
Agree	Accepter d'effectuer une action
Refuse	Refuser d'effectuer une action
Request-When	Demande d'effectuer une action si certaine proposition devient vraie
CFP	Appel d'offre pour effectuer une action donnée
Accept-Proposal	Accepter une proposition
Reject-Proposal	Refuser une proposition

Tableau 4.1. *Quelques performatives FIPA-ACL*

La figure 4.7 illustre -en résumé- un scénario d'allocation des ressources sans panne.

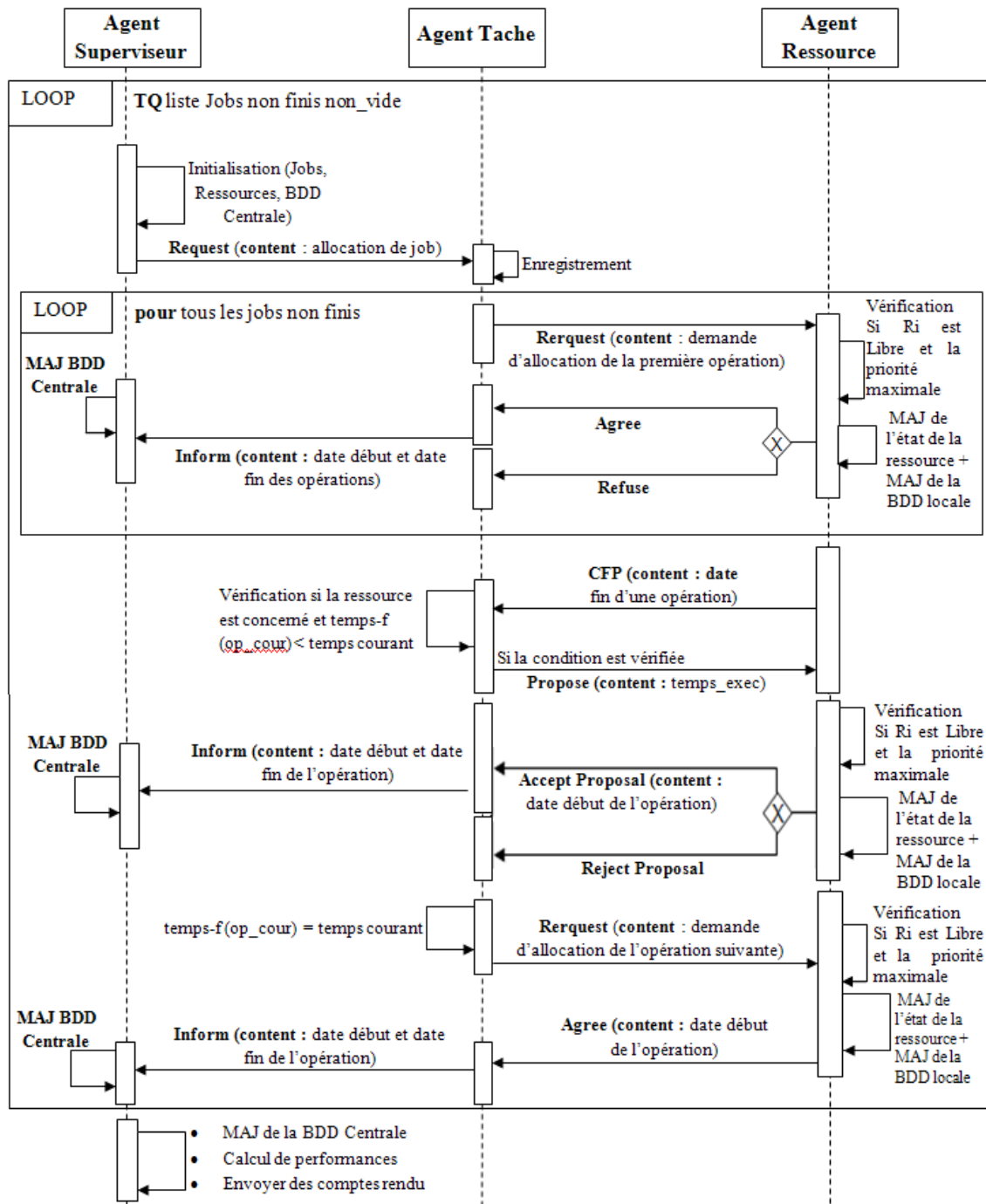


Figure.4.7. Scenario d'allocation de ressources sans panne

Exemple illustratif :

Considérant l'exemple suivant contenant trois ressources (machines) : M1, M2 et M3 et quarts jobs : J1, J2, J3 et J4, arrivant chacun en temps $t=0$. Les ordres de passage des opérations entre les ressources ainsi que les temps d'exécution sont mentionnés dans le tableau 4.2.

Jobs	Ordres de passage des opérations sur les machines avec le temps d'exécution		
J1	M2(5)	M3(2)	M1(7)
J2	M2(4)	M1(8)	M3(11)
J3	M3(10)	M2(5)	M1(6)
J4	M1(3)	M2(4)	M3(2)

Tableau 4.2. Exemple illustratif

L'application de l'algorithme d'allocation de ressources avec la règle de priorité *SPT* nous donne le diagramme de Gantt illustré dans la figure 4.8 suivante.

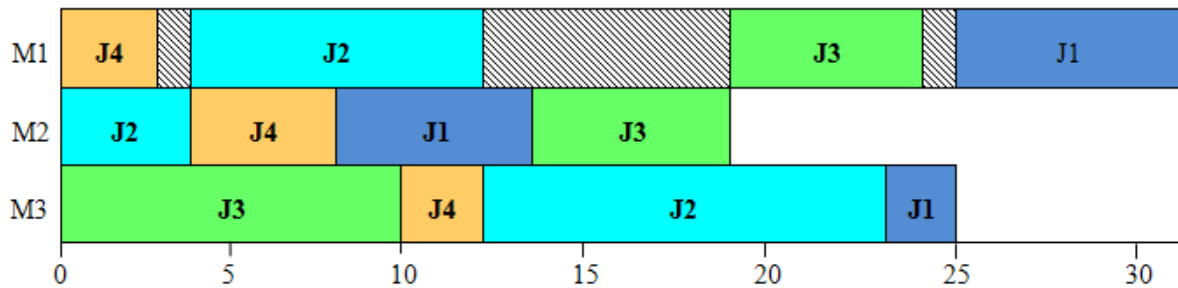


figure 4.8. Diagramme de Gantt pour l'exemple illustratif

6. Conclusion :

Dans ce chapitre on a introduit une approche multi-agent distribué supervisé pour le pilotage en temps réel composé de trois types d'agents qui coopèrent pour réaliser un ordonnancement, une allocation dynamique de ressources et de contrôle d'un système de production de type job-shop. L'architecture proposée prend en considération les contraintes liées à ce type de systèmes de production ainsi que celles liées à la maintenance curative. L'architecture proposée peut aussi aider l'utilisateur à choisir ses propres critères de performances.

Pour atteindre ces objectifs, les agents du système réagissent en temps réel de façon autonome et coopèrent entre eux en utilisant le protocole FIPA-ACL. Pour résoudre les conflits sur les ressources du système, un ensemble de règles de priorité définies dans beaucoup de travaux de recherche est utilisé.

Conclusion générale et perspectives

L'objectif de ce mémoire était La modélisation et la simulation d'un système de production de type job shop en proposant une architecture multi agents distribuée supervisée pour son pilotage en temps réel.

Dans ce mémoire, nous avons présenté tout d'abord une revue détaillée et un état de l'art sur les systèmes de production en abordant les notions générales concernant ces systèmes ainsi que les concepts essentiels qui leurs sont liés : leurs classifications, leurs contraintes et enjeu et leurs finalités ceci afin de mettre en évidence les composantes principales de ces systèmes.

Dans un deuxième temps, nous nous sommes penchés sur l'étude de la modélisation et de la simulation des systèmes complexes vu que les systèmes de production peuvent être classés dans cette catégorie de systèmes étant donné le nombre importants de composantes et d'interactions entre ces dernières qui les caractérisent. Nous avons alors abordé les notions des systèmes complexes ainsi que les concepts de *modèle* et de *modélisation*. Puis nous avons exposé quelques outils, formalismes et approches qui ont prouvé leur importante réussite pour la modélisation des systèmes complexes. La simulation informatique, le processus de conception d'une simulation informatique ainsi que la validation du modèle et du simulateur sont des notions clés pour notre démarche.

Ainsi, nous avons été amenés à étudier de façon assez approfondie la modélisation et la simulation des systèmes de production. Nous avons commencé par une présentation des concepts liés à ce domaine, puis nous avons détaillé les quatre approches de modélisation et de simulation des systèmes de production en nous concentrant sur les approches multi agents avec une introduction de quelques travaux de recherches et une étude comparative entre eux.

Notre contribution a consisté dans la proposition d'une approche distribuée supervisée pour la modélisation et la simulation du pilotage en temps réel d'un exemple typique de système de production : le Job Shop. L'architecture générale du système proposé ainsi que celle de chaque entité constituant ce système a été détaillée. Un algorithme d'allocation de ressources qui fait intervenir l'ensemble des entités du système est proposé. L'interaction entre ces

entités est déterminée par un diagramme de séquences indiquant une coopération entre ces entités utilisant le langage de communication FIPA-ACL.

Ce travail nous a permis de découvrir le monde des systèmes de production qui est un domaine où les notions d'efficacité, de robustesse, de fiabilité, de qualité et des délais sont des contraintes vitales pour les entreprises. Gérer et contrôler ces éléments revient à gérer et à contrôler les ressources humaines et matérielles, les outils de production ainsi que les méthodes et l'organisation du travail ainsi que son ordonnancement.

Comme perspectives, nous envisageons de généraliser la modélisation et la simulation pour d'autres types de systèmes de production tels que les flow shop, les Open Shop ou les ateliers flexibles de production.

Nous visons aussi à doter les agents de notre système de la capacité d'apprentissage surtout dans le cas des perturbations du système et des pannes de ses ressources. Dans ce contexte, nous nous intéressons à l'apprentissage par renforcement qui est utilisé largement dans ce domaine.

Bibliographie

[Aissani et al., 2009] Aissani, N., Beldjilali, B., & Trentesaux, D. *Dynamic scheduling of maintenance tasks in the petroleum industry: A reinforcement approach*. Engineering Applications of Artificial Intelligence, 22(7), 1089–1103. 2009.

[Aissani, 2010] Aissani, N., *Pilotage adaptatif et réactif pour un système de production à flux continu: application à un système de production pétrochimique*. Thèse de doctorat, université de Valenciennes et du Hainaut-Cambrésis et l'Université D'Oran, Décembre 2010.

[ANSI, 1998] ANSI/EIA-632-1998, *Processes for engineering a system*, Approuvé 7 Janvier 1999, Révisé 2003.

[Arkin et al., 1997] Arkin, R., Bekey, C., & George, A. *Robot Colonies*, ISBN: 978-0-7923-9904-9, 160 p., Hardcover. 1997.

[Azeiez 2007] Selma Azaiez. *Approche dirigée par les modèles pour le développement de systèmes multi-agents*. Thèse de doctorat, Université de Savoie, France. Décembre 2007.

[Babiceanu et Chen, 2005] Radu F. Babiceanu, F. Frank Chen, *Performance evaluation of agent-based material handling systems using simulation techniques*. Proceedings of the 2005 Winter Simulation Conference : p 1022-1028, Décembre 2005.

[Bakalem et al, 96] M. Bakalem, G. Habchi, A. Courtois, *Conceptual Frames for Physical and Control Systems Modeling in Manufacturing Simulation*, 8th European Simulation Symposium, Simulation in Industry, vol. 1, pp. 319-323, Genoa, Italy, octobre 24-26, 1996.

[Barwise et al. 1990] Barwise J. et Etchemendy J., “*Information, Infons, and Inference*”, dans *Situation Theory and its applications*, vol. 1, Lecture Notes no. 22, CSLI, Stanford, 1990.

[Berchet, 2000] Claire Berchet. *Modélisation pour la simulation d'un système d'aide au pilotage industriel*. Thèse de doctorat, Institut National Polytechnique de Grenoble, France. 2000.

[Berger et al., 2002] Berger T., Sallez Y., Trentesaux D., Tahon C., *Two heterarchical multi-agent approaches for FMS dynamic scheduling*, *Systems Analysis – Modelling – Simulation*, ISSN 0232-9298, Taylor & Francis, vol. 42, n° 5, pp. 757-768. 2002.

- [**Berruet, 1998**] Berruet. P. *Contribution au recouvrement des systèmes flexibles de production manufacturière : analyse de la tolérance et reconfiguration*, thèse de doctorat, Université des sciences et techniques de Lille, France, Décembre 1998.
- [**Birtwistle, 1979**] G. M. Birtwistle, *Discrete event modeling in SIMULA*. Mac Millian. Londres, Angleterre. 1979.
- [**Bitton 1990**] Moïse Bitton, *Ecograi : Méthode de conception et d'implantation de systèmes de mesure de performances pour organisations industrielles* , Thèse de Doctorat en Automatique, Université de Bordeaux 1, septembre 1990.
- [**Blondel, 1997**] BLONDEL.F. *Gestion de la production*, Dunod, 1997.
- [**Cassandras et Lafortune, 1999**] Cassandras, C.G., La fortune, S. *Introduction to Discrete Event Systems*. Kluwer. 1999.
- [**Cellier, 1991**] Cellier, Francois E. *Qualitative modeling and simulation: promise or illusion*. Pages 1086–1090 de : Proceedings of the 23rd conference on Winter simulation. IEEE Computer Society. 1991.
- [**Collart et al., 2007**] S. Collart Dutilleul, N. Jerbi, E. Craye et M. Benrejeb, *Robust Dynamic Control of Multi-product Job-shops*, 4th IFAC Conference on Management and Control of Production and Logistics (MCPL'07), Sibiu, Vol. 2, pp. 265– 270, September, 2007.
- [**Cota et Sargent, 1992**] Cota, B. A., & Sargent, R.E. *A modification of the process interaction world view*. ACM Transactions on Modeling and Computer Simulation (TOMACS), 2 (2): 109-129. 1992.
- [**David et Alla, 1992**] R. David and H. Alla. *Du Grafcet aux réseaux de Petri*. Hermes, 1992.
- [**Demazeau, 2001**] Y. Demazeau. *Voyelles*. Mémoire d'habilitation à diriger des recherches, INP Grenoble, 2001.
- [**Dhouibi, 2005**] H. Dhouibi. *Utilisation des réseaux de Petri à intervalles pour la régulation d'une qualité : Application à une manufacture de tabac*, Thèse de Doctorat, Université des Sciences et Technologies de Lille, Ecole Centrale de Lille, Décembre 2005.
- [**Ferber, 1995**] Ferber, J. *Les Systèmes Multi-Agents : vers une intelligence collective*. InterEditions. 1995.

[Ferrier, 2003] Ferrier J.L., *Modèle et système : un tour d'horizon*, Notes de cours de DEA automatique et informatique industrielle, Université d'Angers-Ecole centrale de Nantes, disponible sur : www.istia.univangers.fr/~ferrier, 2003.

[Filippi, 2003] Jean Baptiste FILIPPI, *Une architecture logicielle pour la multi-modélisation et la simulation à évènements discrets de systèmes naturels complexes*, Thèse de Doctorat, Université de CORSE. PASQUALE PAOLI, Décembre 2003.

[FIPA, 2002] *Foundation for Intelligent Physical Agents FIPA. FIPA ACL Message Structure Specification*, 2002.

[Fishwick, 1995] Fishwick P.A., *Simulation Model Desing and Execution*, Prentice Hall, 1995.

[Fishwick, 1997] Fishwick, Paul A. *Computer simulation : growth through extension*. Transactions of the Society for Computer Simulation International, 14(1), 13–23.1997.

[Gautier et al. 2000] Gautier F. et Giard V., *Vers une meilleure maîtrise des coûts engagés sur le cycle de vie lors de la conception de produits nouveaux*, GREGOR, Janv.2000.

[Giard, 1988] GIARD, V. *Gestion de la production*. Paris : Economica, 1988. 1068 p.

[Giard, 2003] V. Giard, *Gestion de la production et des flux*, Economica, Paris, 2003.

[GRIMAUD, 1996] F. GRIMAUD, *Conception D'une Base de Composants Logiciels Pour L'évaluation Des Performances Des Entreprises Manufacturières*. Thèse de doctorat, Université Blaise Pascal, Clermont-Ferrand. 1996.

[HABCHI et al. 2006] G.HABCHI, M-P.HUGET, M.PRALUS,*D'une approche composant vers une approche Agent pour un pilotage optimisé des systèmes de production*, 6^e Conférence Francophone de MOdélisation et SIMulation – MOSIM'06 – , Avril 2006 , Rabat , Maroc.

[Habchi, 2001] G. Habchi, *Conceptualisation et modélisation pour la simulation des systèmes de production*, Habilitation à Diriger des Recherches, Université de SAVOIE, 2001.

[Hardebolle, 2008] Cécile Hardebolle, *Composition de modèles pour la modélisation multi-paradigme du comportement des systèmes*. Thèse de doctorat, université de Paris-Sud XI Orsay. Novembre 2008.

- [Harel et al. 1985] David Harel and Amir Pnueli, *Logics and models of concurrent systems*, volume 13 of NatoAsi Series F: Computer And Systems Sciences, chapter On the development of reactive systems, pages 477–498. Springer-Verlag New York, ISBN 0-387-15181-8, New York, NY, USA, 1985.
- [Harmelen et al. 2001] HARMELEN, F., PATEL-SCHNEIDER, P. et HORROCKS, I., *Reference description of the DAML + OIL ontology markup language*.2001. <http://www.daml.org/2001/03/reference>.
- [Hax et Candea, 1984] Hax.A.C, D. Candea. *Production and Inventory Management*. Prentice Hall. 1984.
- [Hill, 1996] D. R.C. Hill. *Object-Oriented Analysis and Simulation*. Addison-Wesley, 1996.
- [IEEE 1220-05] IEEE 1220-05IEEE, *Standard for Application and Management of the Systems Engineering Process*. Approuvé le 8 December 1998, révisé en 2005.
- [Jargot, 1999] P. Jargot. *Langage de programmation pour api. Norme CIE 1131–3.Techniques de l'Ingénieur*, S 8030, 1999.
- [Jensen, 1992] Kurt Jensen, *coloured Petri nets Basic concepts, Analysis methods and practical use*, volume1, second Edition, Springer, 1992.
- [Kerkouche, 2011] El-hillali Kerkouche, *Modélisation Multi-Paradigme : Une Approche Basée sur la Transformation de Graphes*, Thèse de Doctorat en Sciences en Informatique. Université de Mentouri Constantine, Algérie, 2011.
- [Koestler, 1967] A. Koestler, *The Ghost in the Machine*, Arkana Books, Londres, 1967.
- [Kouider et Bouzouia, 2012] Ahmed Kouider, Brahim Bouzouia, *Multi-agent job shop scheduling system based on cooperative approach of idle time minimisation*, International Journal of Production Research, 50:2, 409-424, DOI: 10.1080/00207543.2010.539276, Janvier 2012.
- [Kouis et al, 1997] Kouiss. K, H. Pierreval et N. Mebarki. *Using multi-agent architecture in FMS for dynamic scheduling*. Journal of Intelligent Manufacturing 8, No.1, 41-47, 1997.
- [Kuhn, 1972] KUHN. T. (1972). *La structure des révolutions scientifiques*. Flammarion. Nouvelle traduction par Laure Meyer 1983.
- [Larvet, P, 1994] Larvet, P. (1994) '*Analyse des systèmes : de l'approche fonctionnelle à l'approche objet*' InterEditions – Paris.

- [Leitao et Restivo, 2008] Leitao. P, Restivo. F, *A holonic approach to dynamic manufacturing scheduling*, Robotics and Computer-Integrated Manufacturing”, No 24, pp 625–634, 2008.
- [Le Moigne, 1990] Le Moigne, J.-L. *La modélisation des systèmes complexes*. Dunod. Paris 1990.
- [Le Moigne, 1994] J.L. Le Moigne, *La théorie du système général - théorie de la modélisation*, Editions Presses Universitaires de France, 4ème éd. 1994.
- [M’HALLA, 2010] ANIS M’HALLA. *Contribution à la Gestion des Perturbations dans les Systèmes Manufacturiers à Contraintes de Temps*. Thèse de doctorat, université de Lille Nord de France 2010.
- [Macal et North, 2006] Charles M. Macal et Michael J. North, *Tutorial on agent-based modeling and simulation part 2: how to model with agents. Proceedings of the 2006 Winter Simulation Conference*. ISBN:1-4244-0501-7, Monterey, California, USA, December, 2006.
- [Maturana et al, 1999] F. Maturana, W. Shen, D.H. Norrie, *MetaMorph : an adaptive agent-based architecture for intelligent manufacturing*, International Journal of Production Research, 1999, 37(10) : 2159–2174.
- [Michel, 2004] Fabien Michel. *Formalisme, outils et éléments méthodologiques pour la modélisation et la simulation multi-agents*. Thèse de Doctorat, Université de Montpellier II. Décembre 2004.
- [Miller, 2004] Miller, J. A., & Fishwick, P. A. *Investigating Ontologies for Simulation Modeling*. Papier présenté dans "2004 Simulation Symposium". 2004.
- [Milner, 1993] Robin Milner, *Elements of interaction: Turing Award Lecture*, Communications of the ACM, 36(1):78–89, CODEN CACMA2. ISSN 0001-0782, Janvier 1993.
- [Minsky, 1968] Minsky M., “*Matter, minds and models*”, Marvin Minsky, Ed., 1968.
- [Mirdamadi, 2009] S. Mirdamadi, *Modélisation du processus de pilotage d’un atelier en temps réel à l’aide de la simulation en ligne couplée à l’exécution*, Thèse de Doctorat, Université de TOULOUSE, Juin 2009.
- [Okino, 1993] N. Okino, *Bionic Manufacturing System*, dans “Flexible Manufacturing System : past – present – future”, J. Peklenik (ed), CIRP, Paris, 1993, pp. 73-95.

- [**Olivier-Maget, 2007**] Nelly. Olivier-Maget, *Surveillance des systèmes dynamiques hybrides : Application aux Procédés*. Thèse de Doctorat, Université de Toulouse, Spécialité : Systèmes Industriels. Décembre 2007.
- [**Orën, 1991**] Orën, T. I. *Dynamic templates and semantic rules for advisors and certifiers*. Dans : Paul A. Fishwick, Richard B. Modjeski, *Knowledge Based Simulation : Methodology and application.*, ISBN: 978-0-387-97374-6 (Print) 978-1-4612-3040-3 pages 53.76. Springer Verlag. 1991.
- [**Ounnar et Pujo, 2012**] F. Ounnar, P. Pujo, *Pull control for job shop: holonic manufacturing system approach using multicriteria decision-making*, J Intell Manuf 23:141–153, 2012.
- [**Parunak, 1998**] H. Van Dyke Parunak. *What can agents do in industry and why?* « Proceedings of the 2nd international conference on Cooperative Information Agents (CIA'98) ». Springer-Verlag, Lecture Notes in Computer Science. Paris, France, 3-8 Juillet 1998.
- [**Pavé, 1994**] Pavé A., *Modélisation en Biologie et en Ecologie*, Aléas, 1994.
- [**Peeters et al, 2001**] P. Peeters, H. Van Brussel, P. Valckenaers, J. Wyns, L. Bongaerts, M. Kollingbaum et T. Heikkila, *Pheromone-based emergent shop floor control system for flexible flow shops*, Artificial Intelligence in Engineering, vol. 15, 2001, pp. 343-352.
- [**Quesnel, 2006**] Gauthier Quesnel, *Approche formelle et opérationnelle de la multi-modélisation et de la simulation des systèmes complexes : Apports pour la simulation de Systèmes Multi-Agents*, Thèse de doctorat, École doctorale de l'Université du Littoral - Côte d'Opale, Décembre 2006.
- [**Rechtin, 2000**] Rechtin, E., *Systems architecting of organizations: why eagles can't swim*. CRC Press, Florida, 2000.
- [**ROHÉE, 2008**] Benoît ROHÉE, *Contribution à la conception d'applications de pilotage des systèmes manufacturiers*, Thèse de doctorat, Université de Reims Champagne-Ardenne, 2008.
- [**Ryu et al, 2000**] RYU, K., SHIN, M., KIM, K. JUNG, M., 2000, *Intelligent control architecture for fractal manufacturing system*. Proceedings of 3rd Asia-Pacific Conference on Industrial Engineering and Management Systems, Hong Kong, Décembre 2000, pp. 594–598.

- [Ryu et al, 2003] Kwangyeol Ryu, Mooyoung Jung, *Agent-based fractal architecture and modeling for developing distributed manufacturing systems*, int. j. prod. res., 2003, vol. 41, no. 17, 4233–4255.
- [Schruben, 1998] L. Schruben, *Simulation Modeling with Event Graphs*. Papier présenté dans ACM, vol. 26. 1983.
- [Shannon, 1976] Shannon, Robert E. *Simulation modeling and methodology*. Pages 9–15 of : Proceedings of the 76 Bicentennial conference on Winter simulation.1976.
- [Shannon, 1998] Shannon, Robert E. *Introduction to the art and science of simulation*. Proceedings of the 30th conference on Winter simulation. IEEE Computer Society Press.1998.
- [Shen et al, 2000] Shen, W., Maturana, F. et Norrie, D. H., *MetaMorph II : an agent-based architecture for distributed intelligent design and manufacturing*. Journal of Intelligent Manufacturing, 2000, 11(3): 237-251.
- [Shin et Jung, 2004] Shin Moonsoo et Jung Mooyoung. *MANPro: mobile agent-based negotiation process for distributed intelligent manufacturing*. vol. 42, n°2, pp. 303-320.2004.
- [Smith, 1980] Smith, R., *The Contract Net Protocol: high-level communication and control in a distributed problem solver*. IEEE Transactions on Computers, 1980, 29(12), 1104–1113.
- [Sperandio, 2005] Séverine SPERANDIO, *Usage de la modélisation multi-vue d'entreprise pour la conduite des systèmes de production*, Thèse de doctorat, Université de Bordeaux 1, Décembre 2005.
- [TAMANI, 2008] Karim TAMANI, *Développement d'une méthodologie de pilotage intelligent par régulation de flux adaptée aux systèmes de production*, thèse de doctorat, université de SAVOIE, 2008.
- [Tharumarajah et al, 1998] A. Tharumarajah, A. j. Wells and L. Nemes, *Comparison of Emerging Manufacturing Concepts*, Proc. of the IEEE Int. Conference on Systems, Man and Cybernetics, pp. 325-331, 11-14 Octobre 1998.
- [Tounsi, 2009] Jihène Tounsi, *Modélisation pour la simulation de la chaîne logistique globale dans un environnement de production PME mécatroniques*, Thèse de Doctorat, Université de SAVOIE, Décembre 2009.

- [**Trentesaux 1996**] Damien Trentesaux, *Conception d'un système de pilotage distribué, supervisé et multicritère pour les systèmes automatisés de production*, Thèse de Doctorat, Institut National Polytechnique de Grenoble, janvier 1996.
- [**Treuil et al, 2008**] J.-P. Treuil, A. Drogoul, and J.-D. Zucker. *Modélisation et Simulation à base d'agents*. Dunod, 2008.
- [**Ueda, 2001**] Ueda ., *Synthesis and emergence - research overview*. Artificial Intelligence in Engineering, volume15, pages 321- 327, 2001.
- [**Van Brussel et al. 1998**] Van Brussel H., Wyns J., Valckenaers P., Bongaerts L. et Peeters P., *Reference architecture for holonic manufacturing systems : PROSA*, Computers in industry, Vol. 37, pp. 255-274, 1998.
- [**Vareille et Arnoux, 1998**] J.Vareille, M.Arnoux. *L'enseignement du langage Grafcet et de ses interprétations*. Revue de l'E.P.I., n°90, juin 1998.
- [**Vernadat, 1999**] Vernadat F. *Techniques de Modélisation en Entreprise : Application aux Processus Opérationnels*. Édition Économica. 1999.
- [**Warneke, 1993**] H. J. Warneke, *The fractal company*, Springer Verlag, 1993.
- [**Widmer, 1991**] M. Widmer, *Modèles mathématiques pour une gestion efficace des ateliers flexibles*. Edition Presses Polytechniques Romandes, 1991.
- [**Zeigler, 1976**] Zeigler, B.P. *Theory of Modeling and Simulation.. New York, Wiley*.1976.
- [**Zeigler et al, 2000**] Zeigler, Bernard P., Kim, Tag Gon, & Praehofer, Herbert. 2000. *Theory of Modeling and Simulation*. Academic Press, Inc. 2000.