



REPUBLIQUE ALGERIENNE POPULAIRE ET DOMOCRATIQUE
MINISTERE DE L'EDUCATION ET DE LA RECHRCHE SCIENTIFIQUE
UNIVERSITE ABOU BAKR BELKAID
FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE



SUJET DU MEMOIRE DE MAGISTERE:

**Proposition d'un Outil d'aide à l'évaluation des Stratégies de
Coordination dans le fonctionnement d'un WEB-DSS : Application à
une Gestion de production dynamique.**

Présenté par :

Mr. GAOUAR ADIL

Pour l'obtention du:
Diplôme de magister

Spécialité : Informatique
Option: Intelligence Artificielle et Aide à la Décision

Membres du jury:

Président	Pr. Feham Mohammed	<i>Prof</i>	<i>Université de Tlemcen</i>
Examineur	Pr. Chikh Mohammed Amine	<i>Prof</i>	<i>Université de Tlemcen</i>
Encadreur	Mme. Taghezout Noria	<i>MCA</i>	<i>Université d'Oran</i>
Invité	Mr. Mohammad Amine Abderrahim	<i>MCB</i>	<i>Université de Tlemcen</i>

Soutenu le : 29 Juin 2013

Résumé:

Étant donné les diverses sources de données et les différents outils d'aide à la décision, la conception, la spécification et la mise en œuvre d'un SIAD à base du Web, souvent dans un environnement distribué, est toujours un domaine de recherche en plein essor.

L'objectif majeur de cette étude est de renforcer le travail des agents de production et de coordination par la mise en place d'un programme de gestion de temps et de priorité. L'outil à développer permettra de donner une analyse statistique des résultats obtenus en matière de temps (temps-improductifs, temps d'analyse, temps de recherche sur le WEB, temps-réels de production etc..).

En effet, notre contribution est matérialisée par le recensement de toutes les stratégies possibles de coordination, en particulier celles qui sont suivies par l'agent coordinateur durant une session de travail, soit une prise de décision globale. Pour satisfaire cet objectif, nous avons enrichi le modèle proposé par une *mémoire collective (MC)*. Cette mémoire permet de mentionner les adaptations apportées aux solutions tirées de la base des cas de l'agent coordinateur et elle permet à ce dernier de se procurer des solutions aux pannes, si elles existent au niveau de la MC.

De plus, l'on évaluera grâce à l'outil développé chaque stratégie de l'agent de coordination par la mesure de certaines valeurs de temps. Six scénarios ont été testés sur l'application et montrent la faisabilité de l'approche.

Summary:

Due to various sources of data and Decision making tools, the design, specification and implementation of a Web based DSS in a distributed environment is a growing field of research.

The main objective of this study is to enhance the performance of the production and coordination agents by developing a time and priorities management program. The tool being developed will make it possible to analyze statistically the results obtained based on time variables (unproductive time, time of analysis, search time on the WEB, real production time etc.).

Indeed, our contribution is materialized by the identification of all possible strategies of coordination, in particular those which are followed by the coordinator agent during a work session, as decision-making.

To satisfy this objective, we augmented the model suggested by a *collective memory (CM)*. This memory makes it possible to mention the adaptations brought to the solutions drawn from the coordinator agent's case-base to get solutions in case of breakdowns from the level of the CM.

Moreover, we will evaluate using the developed tool, each coordination agent strategy by measuring certain characteristic times. Six scenarios have been tested using the application and helped to show the feasibility of this approach.

_____:

نظرا لتنوع مصادر المعلومات و اختلاف وسائل المساعدة على اتخاذ القرار, فان التصميم, المواصفة و تنفيذ الأنظمة التفاعلية لدعم القرار المبنية على شبكة الانترنت, غالبا في بيئات موزعة, لا تزال مواضع بحث هامة و في تطور مستمر.

الهدف الرئيسي لهذه الدراسة, هو تدعيم عمل الوكلاء الإنتاج و التنسيق بوضع برنامج لتسيير الوقت و الأولويات, الوسيلة التي سنصممها ستسمح لإعطاء دراسة إحصائية للنتائج المحصل عليها في ما يخص الوقت (الزمن غير المفيد, زمن التحليل, زمن البحث عبر الانترنت, ,

في الواقع, فان مساهمتنا تتجسد في إحصاء كل الاستراتيجيات التنسيق الممكنة و خاصة تلك المتتبعة من طرف الوكيل المنسق خلال دورة عمل, يعني خلال عملية اتخاذ قرار إجمالية. بغية الوصول إلى هذا الهدف, لقد قمنا بإثراء النموذج المقترح عن طريق إضافة ذاكرة جماعية (ذ.ج). هذه الذاكرة ستسمح لنا بذكر التحسينات المضافة للحلول المستخرجة من قاعدة الحالات الخاصة بالوكيل المنسق و كذلك تسمح له بالحصول على الحلول الملائمة للمشاكل في حالة وجودها في هذه الذاكرة.

بالإضافة إلى ذلك, سنقوم باستعمال هذا البرنامج لتقييم كل إستراتيجية خاصة بالوكيل المنسق و ذلك بحساب بعض الأزمنة المميزة. ستة سيناريوهات اختبرت باستعمال هذا البرنامج و سمحت بتحديد جدوى هذه المناهج.

Remerciements

C'est une habitude saine que de remercier au début d'un tel travail tous ceux qui, plus ou moins directement, ont contribué à le rendre possible. C'est avec mon enthousiasme le plus vif et le plus sincère que je voudrais rendre mérite à tous ceux qui m'ont aidé à mener à bien ce mémoire.

Je commence par remercier Dieu le tout puissant de m'avoir aidé et guider durant les périodes difficiles par lesquelles je suis passé.

*Je souhaite remercier du plus profond du cœur ma directrice de thèse, Mme **Taghezout Noria**, Maitre de Conférences à l'Université d'Oran, pour avoir dirigé ce travail. Je lui suis reconnaissant pour le temps conséquent qu'elle m'a accordé, pour ses conseils avisés et son écoute qui ont été prépondérants pour la bonne réussite de ce mémoire. Son énergie et sa confiance ont été des éléments moteurs pour moi. J'ai beaucoup appris à ses côtés, je lui adresse toute ma gratitude.*

*Mes vifs remerciements vont également à Mr **Feham Mohamed**, Professeur à l'Université de Tlemcen, qui m'a honoré en acceptant de présider le jury.*

*Je désire remercier vivement Mr **Chikh Amine**, Professeur à l'Université de Tlemcen pour ses encouragements à réaliser ce travail, son soutien, sa simplicité, pour ses discussions enrichissantes et pour avoir suivi l'évolution de ma thèse et surtout pour avoir été un enseignant modèle et respectueux durant tout notre cursus universitaire. Merci d'avoir accepté de juger ce travail.*

*Je tiens à remercier également Mr **Abderrahim Mohamed Amine**, Maitre de Conférences à l'Université de Tlemcen, pour le grand honneur qu'il me fait en étant présent dans ce jury, et pour les efforts incommensurables qu'il a produit pour nous offrir toujours le meilleur en terme d'enseignement, et plus encore humainement.*

*Je remercie également Mr **Bouguettaya Walid**, pour son aide et ses précieux conseils qui m'ont permis d'améliorer et de valoriser les résultats obtenus. Je le remercie aussi de son amitié.*

*Je remercie également Melle **Bessedik Imen** pour sa contribution.*

Merci à toutes les personnes de bonne volonté de l'Université de Tlemcen qui m'ont apporté leur contribution de diverses manières au cours de cette formation.

Dédicaces

Je dédie ce modeste travail en signe de reconnaissance, de respect et de dévouement avec plein d'humilité :

A mes très chers parents qui n'ont jamais cessé de m'encourager pour entreprendre mes études et atteindre mes objectifs et qui m'ont apporté leur aide scientifique, matérielle et psychologique pour l'élaboration de ce travail, je les remercie du fond du cœur d'être présents pour moi, que Dieu les garde pour nous le plus longtemps possible.

A ma très chère femme et mon adorable fille pour leur soutien, leur patience et leurs encouragements

A mon très cher frère et ami Imad.

A mes très chères sœurs Manel et Naïla; à leurs maris Amine et Tarik ainsi que leurs très chers enfants.

A ma belle famille

A mes très chers Amis

A tous les membres de ma famille, tantes, oncles, cousins et cousines ;

A tous mes camarades de promotion

A tous ceux qui m'ont aidé et encouragé pour l'élaboration de ce mémoire.

Et une pensée particulière pour mes feux grands parents que Dieu les accueille dans son vaste paradis.

TABLE DES MATIERES

TABLE DES MATIERES

INTRODUCTION GENERALE.....	3
CHAPITRE I : PILOTAGE ET GESTION DE PRODUCTION DYNAMIQUE.....	7
I. Introduction :	8
II. Notions de bases relatives à la production :	9
1. Contexte général :	9
2. Définitions de base :	9
3. Définition de la production :	11
4. Evolution de la production et des règles correspondantes :	11
<i>a. Les principales périodes :</i>	11
<i>b. Les nouvelles règles de la production moderne :</i>	12
III- Gestion de la production dynamique:	12
<i>Définitions</i>	12
<i>Objectifs</i>	13
1. Les décisions stratégiques :	14
2. Les décisions tactiques :	14
3. Les décisions opérationnelles :	14
<i>a. par l'horizon de temps considéré :</i>	14
<i>b. par le niveau d'agrégation :</i>	15
<i>c. par le niveau de responsabilité :</i>	15
IV- Typologie de la production :	15
1. structure du produit :	15
2. Les modes de production:	16
3. Structure des ateliers de production :	18
V- L'ordonnancement : problème et définitions...	18
1. Définition et objectifs :	18
2. Le Problème de l'ordonnancement :	19
3. l'optimisation et les systèmes d'ordonnancement :	20
3.1- <i>Objectifs à long terme :</i>	21
3.2- <i>Objectifs à moyen et court terme :</i>	21
4. Les types d'ordonnancement :	21
<i>a. Ordonnancement prévisionnel :</i>	22
<i>b. Ordonnancement réactif :</i>	22
VI- Pilotage d'un système de production :	22
1- Pilotage des systèmes industriels :	23
2- Fonctions génériques du pilotage de la production :	26

TABLE DES MATIERES

VII- Typologie des structures de pilotage :	27
1. Structure centralisée :	27
2. Structure hiérarchisée :	27
3. Structure coordonnée :	28
4. Structure distribuée :	28
5. Structure distribuée supervisée :	28
6. Structure décentralisée :	28
7. Pilotage prédictif :	29
8. Pilotage proactif :	29
9. Pilotage réactif ou pilotage en temps réel :	30
VIII- Conclusion :	30
CHAPITRE II : SYSTEME D'AIDE A LA DECISION EN MILIEU INDUSTRIEL	32
I. Introduction :	32
II. La prise de décision :	33
1. Définitions :	33
2. Le processus de décision :.....	33
a. Les étapes du processus décisionnel :	34
b. Le processus décisionnel selon la structure de l'entreprise :	34
3. Niveau de structuration des problèmes de décisions :.....	35
III. Les Systèmes Interactifs d'Aide à la Décision (SIAD):	36
1. Introduction:.....	36
2. Définitions :.....	37
3. Les différentes architectures des SIAD :.....	40
a. Architecture en réseau :.....	41
b. Architecture centralisée :.....	41
c. Architecture hiérarchisée :.....	42
4. Classification des SIAD :.....	43
a. Classification au niveau utilisateur :.....	43
b. Classification au niveau conceptuel :.....	44
c. Classification au niveau système et opérationnel:.....	44
d. Classification au niveau connaissance :.....	45
5. Exemples de domaine d'application des SIAD :.....	46
a. Un système d'allocation de wagon :.....	46
b. La gestion de la production :.....	46

TABLE DES MATIERES

c. Un système de surveillance pour les sites industriels à hauts risques :	47
d. Pioneer Natural Resources :	47
IV. Les systèmes interactifs d'aide à la décision à base de web (web-based DSS) :	47
1. Définitions d'un SIAD à base de Web (Web-based DSS) :	48
2. Les travaux de recherche récents dans le SIAD à base de Web (Web-based DSS) :	49
a. Architectures et technologies :	49
b. Applications et réalisations :	50
V. Barrières au succès des SIAD :	51
VI. Conclusion :	52
CHAPITRE III : SYSTEMES MULTI-AGENTS & COORDINATION.....	53
PARTIE I : AGENTS ET SYSTEMES MULTI-AGENTS.....	54
I- Introduction :	56
II- Les notions relatives aux Agents :	57
1. Définitions :	57
a. <i>Agent</i> :	57
b. <i>Déterminant d'un agent</i> :	58
2. Les différents types et architectures d'agents:	59
a. <i>Les agents cognitifs</i> :	59
b. <i>Les agents réactifs</i> :	61
c. <i>Les agents BDI</i> :	64
d. <i>Les agents hybrides</i> :	65
e. <i>Les agents logiciels</i> :	67
f. <i>Les agents mobiles</i> :	67
g. <i>Les agents assistants</i> :	67
III- Les Systèmes Multi-Agents (SMA) :	67
1. Définitions :	67
2. Caractéristiques des SMA :	68
3. Décomposition d'un SMA :	70
4. Interactions entre les agents au sein d'un SMA :	71
a. <i>L'interaction par la communication</i> :	71
b. <i>L'interaction par la coopération d'agent</i> :	73
c. <i>L'interaction par la négociation entre ressources</i> :	74
5. Protocoles d'interactions :	74
a. <i>Protocoles de Coordination</i> :	74
b. <i>Protocoles de Coopération</i> :	75
c. <i>Protocoles de Négociation</i>	75

TABLE DES MATIERES

6. Quelques exemples d'applications des SMA :	75
1. Application des SMA dans les télécommunications :	75
7. Quelques Plates-formes de développement des SMA :	77
IV- Conclusion : partie I.	78
PARTIE II : COORDINATION AU SEIN DES SMA.	80
Coordination entre agents	80
I- Introduction :	80
II- Définitions :	81
III- Problématique de la coordination :	83
IV- Objectifs et motivations de la coordination dans les SMA :	84
1. Pluralisme :	84
2. Recherche de la cohérence globale du système :	84
V- Dimensions de la coordination :	86
1. Structure d'un agent :	86
2. Modèle de spécification du problème de coordination :	87
3. Mécanismes de coordination : modèles orientés tâches Vs modèles orientés agents :	87
a. Modèles de coordination orientés tâches :	87
b. Modèles de coordination orientés agents	88
VI- Approches de coordination d'agents :	88
1. Coordination basée sur la résolution distribuée de problèmes :	88
2. Coordination basée sur les structures organisationnelles	90
a. Organisations statiques :	90
b. Organisations dynamiques :	91
3. Coordination basée sur les protocoles de coopération :	91
4. Modèles de négociation et de prise de décision distribuée	92
5. Modèles de coordination fondés sur la formation de coalitions :	92
VII- Coordination d'actions par planification multi-agent et distribuée :	93
1. La coordination par synchronisation d'actions :	93
2. La planification multi-agent comme spécialisation de la résolution distribuée de problème :	94
3. La coordination par planification multi-agent centralisée ou distribuée :	94
4. Planification multi-agent centralisée et plans distribués :	95
5. Planification multi-agent distribuée	95
a. Planification distribuée pour plan centralisé	95
b. Planification distribuée pour plan distribué	95
6. Processus de coordination d'actions par planification :	96
VIII- Critères d'évaluation des systèmes de coordination :	96

TABLE DES MATIERES

IX- Conclusion : Partie II	97
CHAPITRE IV : CONCEPTION	98
I. Introduction :	99
II. Architecture générale du Web-DSS :	99
1. Le rôle des composants du Web-DSS :	101
II. Structure générale du Système Multi-Agents (SMA) :	103
III. Architecture des agents du SMA :	104
1. Agent Production (APr):	104
<i>a- Architecture et composants de l'Agent Production:</i>	104
<i>b- Rôles des composants de l'Agent Production :</i>	106
<i>c- Rôle de l'Agent Production :</i>	110
<i>d- variables temporelles de l'agent production :</i>	111
<i>e- Algorithme du fonctionnement de l'agent production :</i>	111
2. Agent Analyseur (AA) :	113
<i>a- Architecture et Composants de l'Agent Analyseur (AA) :</i>	113
<i>b- Rôles des composants de l'Agent Analyseur (AA) :</i>	113
<i>c- Rôle de l'agent analyseur :</i>	115
<i>d- Variables temporelles de l'agent analyseur :</i>	115
<i>e- Algorithme de fonctionnement de l'agent analyseur :</i>	116
3. Agent Ressources (AR) :	116
<i>a- Architecture de l'agent ressources :</i>	116
<i>b- Rôles des composants de l'agent ressources :</i>	117
<i>c- Variables temporelles de l'agent ressources :</i>	117
<i>d- Algorithme de fonctionnement de l'agent ressource :</i>	117
4. Agent Proposition (AP):	118
<i>a- Architecture de l'agent proposition :</i>	118
<i>b- Rôles des composants de l'agent proposition :</i>	119
<i>c- Rôle de l'agent proposition :</i>	120
<i>d- Variables temporelles de l'agent proposition :</i>	120
<i>e- Algorithme de fonctionnement de l'agent proposition :</i>	120
5. Agent Coordinateur (AC) :	121
<i>a- Architecture de l'agent coordinateur :</i>	121
<i>b- Rôles des composants de l'agent coordinateur :</i>	122
<i>c- Variables temporelles de l'agent coordinateur :</i>	123
<i>d- Algorithme de coordination :</i>	123
IV. Fonctionnement général du Web-DSS:	125

TABLE DES MATIERES

1. Déroulement du système :	125
2. Diagramme des séquences : les échanges entre les agents :	126
V. Les stratégies de coordination :	127
1. Stratégies de coordination :	127
a- La coordination centralisée :	127
b- La coordination décentralisée :	128
2. Techniques de coordination :	128
a- La coordination par structure organisationnelle :	128
b- La coordination par échange de méta-informations :	128
c- La coordination par planification multi-agents :	128
3. Stratégies de coordination adoptées par notre système :	129
a- Stratégie de coordination locale :	129
b- Stratégie de coordination externe:	129
c- Métriques de comparaison entre les deux stratégies :	129
VI. Syntaxe des requêtes échangée entre agents :	130
VII. Langage de communication entre agents :	130
1. Envoi des messages :	130
2. Contenu des messages :	131
3. Syntaxe des messages :	131
VII. Conclusion :	132
CHAPITRE V : IMPLEMENTATION	133
I. Introduction :	133
II. La technologie J2EE :	133
1. les fonctions couvertes par J2EE	135
a. Technologies du niveau client	135
b. Technologies du niveau intermédiaire	135
c. Autres technologies J2EE	136
III. L'environnement de développement	136
1. Choix de langage de programmation	136
2. JBuilder : environnement de développement intégrant Java	136
IV. La plate-forme JADE	137
1. Brève description de JADE	137
2. La norme FIPA	137
3. Architecture logicielle de la plate-forme JADE	138
4. Langage de communication de la plate-forme JADE	138
V. Exemple d'application	139

TABLE DES MATIERES

1. Création d'un nouveau projet	139
2. Affectation des ressources au projet	140
3. Traitement de l'évènement panne	142
a. Exécution des ressources	142
4. Résolution de panne	143
CONCLUSION GENERALE & PERSPECTIVE	150
REFERENCES BIBLIOGRAPHIE	154

INTRODUCTION GENERALE

INTRODUCTION GENERALE

Depuis l'existence de l'humanité, la production a toujours constitué un facteur majeur pour la survie des hommes et ce, quels que soient leurs origines ou leurs modes de vie. L'explosion démographique qu'a connue le 18^{ième} siècle a provoqué un changement majeur dans le mode de production car c'est à cette époque que celle-ci est passée du statut artisanal à celui d'industriel, comprenant des machines, des ouvriers, de la matière première à transformer... réalisant ainsi une production en série et à grande échelle. Pour ce faire, il fallait des idées nouvelles, de la recherche scientifique... mais surtout un excellent sens de gestion, permettant d'établir de nouvelles règles pour améliorer et faciliter la gestion de ce mode de production.

Actuellement, dans le monde dans lequel nous vivons, les entreprises de production font face à un environnement en perpétuels changements, donc elles ont besoin d'être réactives pour rester compétitives et pour conquérir de nouveaux marchés. Pour y arriver, elles sont contraintes d'améliorer leur façon de piloter la production, tant au niveau stratégique, pour s'adapter aux progrès de la technologie ou suivre les évolutions du marché, qu'au niveau opérationnel, pour réagir face aux aléas et aux changements imprévisibles des marchés financiers. Au niveau stratégique, ceci les conduit à modifier et adapter leurs moyens (matériels et logiciels) de production pour faire face à l'arrivée de nouveaux produits et de nouveaux concurrents, pour réduire notamment leurs délais de fabrication et leurs coûts de production, donc améliorer leur productivité quantitativement et qualitativement. L'objectif de la réactivité en production est de pouvoir adapter régulièrement le système de production à la demande. Les industriels doivent aussi maîtriser leur système de production au niveau opérationnel. Ils doivent être capables de réagir sur le très court terme aux événements imprévus et/ou imprévisibles, tels qu'une modification ou une annulation d'un ordre de fabrication, l'arrivée d'une commande urgente, des perturbations aléatoires du système de production et ce, de façon la moins perturbante et la plus rapide possible.

Avec la mondialisation des marchés, l'entreprise doit s'adapter, si possible anticiper et même influencer, en tout cas réagir avec énormément d'agilité et de prudence. Pour y parvenir au mieux et dans de bonnes conditions, les gestionnaires ont besoin d'avoir les informations adéquates au moment opportun pour pouvoir prendre les meilleures décisions possibles. L'importance de l'information n'est plus à l'ordre du jour. En effet, depuis plus de quarante ans les chercheurs tels que **Le Moigne (1973, 1974)** et **Simon (1977)** ont attiré l'attention sur le poids de l'information et sur sa répercussion sur les décisions, au point que certains ont eu du mal à dissocier « information » et « décision » et d'autres pensent que « information » est égale au « pouvoir ».

A partir de là, nous pouvons déduire que gérer une production n'est pas une tâche facile à réaliser, mais surtout nous comprenons que le gérant ou le décideur doit avoir des connaissances et des informations très précises pour pouvoir prendre les bonnes décisions, d'où l'utilité et la nécessité d'avoir les outils d'aide à la décision adéquats.

L'aide à la décision existe depuis la nuit des temps ; elle était basée sur l'expérience individuelle et la connaissance des hommes, ainsi que sur l'analyse historique, sujette à l'opinion et à la subjectivité. En effet, les devins, les oracles et les prêtres avaient un rôle très important auprès des décideurs d'autrefois, mais après le siècle des lumières et la révolution industrielle, des modèles mathématiques ont vu le jour, tels que l'ensemble flou, les probabilités, les possibilités, l'intervalle flou, l'incertitude, l'analyse de la décision, la théorie des graphes, les principes du "plus court chemin", du flot maximal, de l'optimisation linéaire, etc.....

A partir de ces modèles et de leurs algorithmes, rapidement, des systèmes informatiques d'aide à la décision sont apparus et ont pris une place croissante dans les processus décisionnels, au point parfois de remplacer l'Homme par des processus automatiques, donnant ainsi naissance à de nouvelles disciplines dans le domaine informatique, telles que l'intelligence artificielle, la recherche opérationnelle et plus récemment l'informatique décisionnelle que les anglophones désignent par « *Business Intelligence* ».

Si autrefois, le problème de l'aide à la décision, incarnée par les hommes, résidait dans l'accès à l'information et aux connaissances ; aujourd'hui, grâce à l'outil informatique (logiciels, BDD, BC...), les entreprises sont en mesure de recueillir, traiter, filtrer, stocker et diffuser des quantités impressionnantes de toutes sortes d'informations en un espace de temps de plus en plus court et à des coûts très raisonnables ; cela étant, elles ont un problème majeur qui se résume en ce qui suit : « comment stocker, organiser et analyser ces connaissances et ces informations dans une base de données décisionnelle pour qu'elles puissent être exploitées pour la prise de décisions? et de quelle façon présenter une aide à la prise de décision? ». Ces questions méritent réponses, et c'est ce à quoi nous tenterons de répondre tout au long de ce mémoire.

La prise de décision et par conséquent l'aide à la décision font souvent face à des problèmes où il y a beaucoup d'appréhension de la part des décideurs. Souvent ces problèmes ne peuvent pas être exprimés sous forme mathématique, ni avoir une structure bien définie, des difficultés qui se répercutent sur le type d'information qu'elle engendre. Les **SIAD**, acronyme de **S**ystèmes **I**nteractifs d'**A**ide à la **D**écision, semblent être une technique et un moyen très performant pour résoudre et répondre à ce type de problèmes c'est-à-dire les problèmes qui sont peu ou mal structurés. Au cours de ce mémoire, nous allons essayer de présenter les SIAD et les technologies les plus pertinentes qui s'y rattachent tout en mettant l'accent sur une classe particulière des SIAD appelée **SIAD basé sur le Web (Web-Based Decision Support System)** et aux mécanismes de prise de décision. Car à ce niveau déjà, nous pouvons présenter ces derniers comme étant des outils d'aide à la décision, innovants et très performants.

En effet, un SIAD basé sur le Web se compose souvent de données et d'instruments rattachés, provenant de régions multidisciplinaires, et qui ne sont pas à l'origine conçus pour collaborer ensemble. D'une part, les méthodes de conception de SIAD traditionnelles sont incapables d'aider à les organiser dans une vue hiérarchique et de spécifier les architectures de logiciel de SIAD basé sur le Web dans la voie formelle. D'autre part, avec l'assistance du Web

et des technologies de réseau, les données et les instruments d'aide à la décision des régions multidisciplinaires peuvent être localisés sur les ordinateurs qui sont distribués sur un réseau. Dans un tel environnement distribué, un SIAD basé sur le Web a besoin d'un cadre distribué pour diriger et intégrer les données et les instruments d'une façon sans coutures [Taghezout, 2011].

Contribution

Le SIAD basé sur le Web proposé par Bessedik [2010] s'articule principalement autour de cinq composants réalisés par le développement d'un système multi-agents (SMA) : agent analyseur (AA), agent ressource (AR), agent proposition (APr), agent coordinateur (AC) et agent production (AP), à partir desquels nous avons pris la liberté d'apporter quelques modifications pour pouvoir arriver à nos objectifs. En particulier, les agents sont utilisés pour recueillir des informations et des options de solutions qui permettront au décideur (opérateur) de se concentrer sur les solutions jugées significatives et adéquates. L'objectif majeur de notre projet est de définir les stratégies de coordination en temps réel, suivies par l'agent coordinateur durant toutes les phases du processus de prise de décision et de fournir une analyse détaillée de ces stratégies. Cette analyse porte essentiellement sur les différents temps d'exécution, d'attente, de recherche...etc, des agents qui composent le SIAD basé sur le Web. L'intérêt majeur est de choisir la meilleure stratégie de coordination.

La deuxième contribution de taille, est celle de l'intégration d'une mémoire collective au sein de notre Web-DSS, permettant le dépôt, l'enregistrement et la consultation des solutions pour un problème nominal, trouvées et validées par les différents intervenants, c'est-à-dire les agents de production ou même des techniciens à travers le web et donc à travers le monde ; favorisant ainsi les échanges, la discussion, le partage des informations et la coopération entre les différents agents de production et les différents décideurs à travers le réseau, pour arriver à la meilleure solution possible, tout en gardant une trace significative, qui permettrait une éventuelle réutilisation. L'autre intérêt majeur de l'utilisation d'une telle mémoire est celui de proposer une aide à la décision en accéléré, en offrant plusieurs scénarios de correction en ligne, afin de limiter les conséquences d'un événement critique. Cela pourrait permettre aussi un retour d'expérience sur la décision et son applicabilité.

Enfin, nous pensons que l'utilisation d'un SIAD basé sur le web semble être l'une des meilleures solutions et l'un des meilleurs outils pour l'aide à la prise de décision. Pour pouvoir illustrer nos propos et notre approche, nous avons choisi un domaine d'application très particulier qui est celui du traitement des situations nominales de pannes des ressources (machines, pièces,...etc.), dans une unité de production.

Organisation du mémoire

Ce mémoire est organisé en cinq chapitres :

Le premier chapitre nous permet de situer le contexte général de notre travail. Aussi, nous tenterons de présenter tout ce qui se rattache à la production dans son contexte le plus général, en faisant un tour d'horizon sur la gestion de production dynamique et le pilotage dans le milieu industriel.

Le second chapitre sera entièrement consacré au développement de tout ce qui se rattache à la décision, au processus de prise de décision pour conclure finalement par la présentation des systèmes interactifs d'aide à la décision (SIAD), en mettant l'accent sur une variante qui est celle des SIAD basés sur le web, leurs contextes, leurs domaines d'applications et enfin quelques travaux relatifs à ces domaines.

Le troisième chapitre est d'une grande importance, puisqu'il est la clé de voute de tout notre travail car il comporte une description très détaillée de tout ce qui se rattache aux Systèmes Multi-Agents (SMA), y compris les agents, leurs types, leurs architectures et les types d'interactions existant entre eux. Ces interactions sont très importantes car ce sont ces dernières qui font que les agents ne sont pas qu'une collection d'individus dispersés dans un environnement ; mais il s'agit d'une organisation cohérente qui est régie par des règles et dont les individus communiquent pour avoir un comportement global cohérent. L'une des interactions les plus importantes dans un tel système est la coordination.

La coordination sera définie en détail dans la seconde partie de ce chapitre, en passant en revue ses mécanismes et ses techniques ainsi que les types existants tout en citant les travaux de recherches se rapportant à son application dans les systèmes multi-agents.

Notons que les SMA permettent de réaliser et d'implémenter les SIAD, car il s'agit d'un outil très performant, qui s'est inspiré du comportement social humain pour apporter de l'aide.

Nous présenterons dans le chapitre quatre l'architecture de notre modèle, le déroulement de son fonctionnement, la description en détail des algorithmes établis pour la mise à jour de la mémoire collective, servant de support d'archivage à des scénarios de solutions, sans oublier la présentation des structures de chaque agent. En fait, ce chapitre se focalisera sur la conception proprement dite de notre modèle de système d'aide à la décision.

Enfin, le chapitre cinq propose une mise en œuvre informatique de l'ensemble des concepts présentés dans ce mémoire, pour illustrer la faisabilité de l'approche proposée et présenter la structure logicielle que nous avons développée.

La conclusion de ce mémoire nous permettra de dresser le bilan des résultats obtenus et d'énumérer quelques perspectives futures.

CHAPITRE I

PILOTAGE ET GESTION DE

PRODUCTION DYNAMIQUE

I. Introduction :

Dans une entreprise manufacturière, le système de production rassemble tous les moyens permettant de donner une valeur ajoutée aux produits ou aux services. Cette transformation est commandée par un système de gestion qui doit respecter un ensemble de contraintes, en vue d'atteindre des objectifs définis [Artigues, 1997].

Vu la crise économique mondiale que nous traversons et avec la mondialisation des marchés, les entreprises de production ont de plus en plus de mal à écouler leurs productions et à réaliser des bénéfices. Face à ces problèmes, la maîtrise de la gestion de la production devient une arme indispensable, plus que jamais, pour assurer la pérennité de toute entreprise, quelle que soit sa taille ou le type de produit qu'elle commercialise.

Dans le contexte industriel actuel, l'offre est largement excédentaire par rapport à la demande, par conséquent, la clientèle est de plus en plus exigeante, donc l'entreprise doit:

- **Etre Réactive** : être capable de s'adapter très vite et en permanence aux besoins en produit qui sont de plus en plus variés, à un marché mondial fortement concurrentiel ;
- **Etre Proactive** : avoir la capacité d'influencer l'évolution du marché en introduisant par exemple de nouveaux produits avant les concurrents ;
- Passer d'une logique de charges à une logique de flux ;
- Respecter le temps qui a une importance fondamentale.

Ceci implique pour l'entreprise:

- la maîtrise des coûts grâce à un suivi précis de la production;
- la réduction des coûts par la réorganisation ou l'élimination des procédures coûteuses n'apportant que peu de valeur ajoutée, par la minimisation des en-cours et des stocks;
- des délais de livraisons courts et fiables;
- une qualité constante et irréprochable;
- de petites séries de produits personnalisés et fréquemment renouvelés¹;
- une grande adaptabilité face aux évolutions de plus en plus rapides de la demande et à l'émergence de nouvelles technologies.

Certaines de ces exigences sont contradictoires ; raison pour laquelle la gestion de la production a pour but d'assurer une cohérence globale en effectuant les arbitrages nécessaires.

Ce chapitre a pour objectif de situer le contexte de notre travail, à savoir la gestion et le pilotage de la production dynamique. Loin de prétendre faire une étude approfondie et détaillée de ce domaine très vaste, nous allons essayer de donner une idée générale assez complète des préceptes et des bases qui se rapportent à la gestion et au pilotage de la production.

¹La durée de vie des produits devient de plus en plus inférieure à la durée de vie des équipements de production. Il est donc indispensable de concevoir des équipements de telle sorte de pouvoir les réutiliser pour d'autres productions; leur flexibilité et leur programmabilité prend, dans ce cadre, toute son importance stratégique.

II. Notions de bases relatives à la production :

1. Contexte général :

La fabrication relève d'une alchimie subtile entre les clients, l'entreprise de fabrication et le produit proprement dit, tout en respectant l'environnement, afin d'aboutir à ce qu'attend chaque client.

En effet, la production industrielle comporte trois éléments en interaction :

Le client éprouve un *besoin* immédiat ou différé pour un *produit* réalisé par *l'entreprise*.

2. Définitions de base :

- **Entreprise** : La norme [AFNOR 90] définit l'entreprise comme un "Système dirigé et organisé en services dont la finalité est de générer de la valeur ajoutée". Pour être compétitive, l'entreprise doit être **réactive** et **proactive**. **Réactive**: être capable de s'adapter très vite et en permanence aux besoins des produits de plus en plus variés, dans un marché mondial fortement concurrentiel. **Proactive** : avoir la capacité d'influencer l'évolution du marché en introduisant par exemple un nouveau produit avant les autres. Ensuite, elle doit passer d'une logique de charges à une logique de flux. Enfin et surtout, le temps a une importance fondamentale.

- **Client** : Dans la même source, le client est : "La personne ou l'entité pour qui le produit a été conçu".

- **Produit** : L'Analyse de la Valeur présente le produit comme "ce qui est (ou sera) fourni à un client pour répondre à son besoin" [AFNOR 90]. Cette définition correspond donc au produit final commercialisé par l'entreprise.

De façon plus concrète, dans le domaine industriel, ce produit final peut être un :

- **Ensemble** (ou produit composé ou appareil) ou un **Composant** indivisible (ou pièce) correspondant à un bien d'équipement, à un bien de consommation durable ou à un composant destiné à être intégré dans un bien d'équipement ou de consommation.

Ces définitions, nous amènent à expliquer de nouveaux termes :

- **Ensemble** : **Blondel [1997]** définit un ensemble comme "la réunion et l'assemblage de plusieurs pièces élémentaires, constituant un élément à nouveau stockable dans cet état". Cette définition s'applique aussi à la notion de **Sous-Ensemble**, dont la principale différence avec l'ENSEMBLE est qu'il ne correspond pas à un produit fini.

- **Pièce** détachée ou **Composant** indivisible : on appelle pièce, tout constituant simple du produit composé de matière et de **Valeur Ajoutée**.

- **Valeur Ajoutée** : C'est la différence entre la valeur de la production et les consommations de biens et de services fournis par des tiers pour réaliser cette production [**Blondel, 1997**]. Elle peut donc s'exprimer simplement par l'opération suivante :

VENTES (produits, pièces détachées)- ACHATS (machines, matière première, salaires)= VALEUR AJOUTEE

- **Ordre de Fabrication (O.F)** : C'est la commande à l'atelier d'un produit donné, pour une quantité donnée et généralement dans un délai à respecter.

- **Flux** : est une notion fondamentale de la production entre les différentes ressources. Un flux correspond à une circulation d'entités physiques ou informationnelles au travers d'un processus, caractérisé par un débit. En production, on pourra donc distinguer essentiellement deux types de flux : **Les flux de matières** et **les flux d'informations** et parfois on peut même

trouver un troisième type qui est celui *des flux financiers*. Une autre distinction des flux concerne leur localisation dans la chaîne logistique qui comporte trois maillons : *approvisionnement, production et distribution*.

- *Les flux EXTERNES* concernent uniquement l'approvisionnement et la distribution.
- *Les flux INTERNES* correspondent à l'ensemble de la production.

La Gestion de Production s'intéresse essentiellement aux *flux internes* de l'entreprise (Figure1).

- **Stock et Encours** : [AFNOR 91] et [AFGI 91] distinguent précisément *les stocks des encours* :

- *Le Stock* correspond, à un moment donné et à un endroit donné, à la quantité d'un article non encore utilisé soit pour l'encours, soit pour la consommation, mesurée dans une unité de stockage appropriée. Le stock se situe donc à l'interface des flux externes de l'entreprise.

- *L'encours* a pour effet ou pour fonction de constituer un "tampon" de régulation des flux internes. De façon plus générale, un encours correspond à l'accumulation d'une différence de flux entre deux ressources. Au point de vue comptable, tous les articles (pièces, sous-ensembles et ensembles) en cours de fabrication sont considérés comme étant sortis du stock fournisseurs ou matières premières, et non encore rentrés dans le stock clients ou produits finis. Donc la totalité des articles présents dans le flux interne du processus, qu'ils soient en attente devant une ressource, en transfert d'une ressource vers une autre, ou en transformation sur une ressource, est considérée comme de l'encours.

Nous avons défini précédemment, le produit par rapport aux besoins du client. Ces besoins peuvent être de deux types :

- **Besoins immédiats** : ceux dont le délai de livraison acceptable par le client est quasiment nul. Ce qui signifie que le produit en question est en stock. Donc l'entreprise doit produire en fonction de ses prévisions de vente en constituant un stock de telle sorte que celui-ci ne soit ni trop important (augmentation du coût de stockage), ni trop faible (incapacité à satisfaire les livraisons urgentes).
- **Besoins différés** : le client accepte un délai de livraison non nul. Ce délai doit toutefois être le plus court possible, car l'entreprise évolue dans un marché où la concurrence est sévère et le délai peut être un critère de décision du client par rapport à des produits similaires d'entreprises concurrentes.

Après avoir décrit quelques notions de bases, nous allons à présent définir ce que nous entendons par « **production** ».

3. Définition de la production :

Giard [1988] définit la production comme étant une transformation de ressources appartenant à un système productif et conduisant à la création de *biens* et/ou de *services*.

Les ressources peuvent être de quatre types :

- des équipements (machines, ...),
- des hommes (opérateurs, ...),
- des matières (matières premières et composants),

- des informations techniques ou procédurales (gammes, nomenclatures, fiches opératoires, ...).

La production d'un bien s'effectue par une succession d'opérations consommant des ressources et transformant les caractéristiques morphologiques ou spatiales de "matières".

La production d'un service s'effectue par une succession d'opérations consommant des ressources pour produire une prestation, quel que soit son type, sans qu'il y ait nécessairement transformation de matière.

4. Evolution de la production et des règles correspondantes :

d.1- Les principales périodes :

Pour **Courtois [1995]** et **Blondel [1997]**, on distingue trois à quatre périodes qui marquent chacune une évolution de la production industrielle :

Première période : aux débuts de l'industrie (à partir de la fin du 19ème siècle), la production est alors proche de l'artisanat : faibles quantités, grande diversité, personnel très qualifié (compagnons).

Deuxième période : de la première guerre mondiale à 1975, période incluant les fameuses "Trente glorieuses" de l'industrie de 1945 à 1975, pendant laquelle la demande est très importante, et même supérieure à l'offre. Les marges sont confortables et les principales caractéristiques de la production sont les suivantes : fabrication en très grandes séries, faible diversité (on connaît la phrase d'Henry Ford " Le client qui désire une Ford T peut demander n'importe quelle couleur, pourvu qu'elle soit noire"), personnel peu qualifié, travail découpé en tâches élémentaires simplifiées et rapides pour garantir un enchaînement rapide des opérations. L'entreprise n'avait qu'à **PRODUIRE PUIS VENDRE**.

Troisième période : Durant cette période transitoire, de 1975 à 1985, l'offre et la demande s'équilibrent, le client a le choix du fournisseur. C'est l'après-choc pétrolier, il est nécessaire de faire des prévisions commerciales, d'organiser les approvisionnements, de réguler les stocks. Il faut alors **PRODUIRE CE QUI SERA VENDU**.

Quatrième période : Depuis la fin des années 70, les marchés sont fortement concurrentiels et surtout se mondialisent. L'offre est supérieure à la demande et de nouvelles contraintes apparaissent : maîtrise des coûts, qualité, délais de livraisons courts et fiables, produits personnalisables et à faible durée de vie, etc...Les séries sont toujours importantes, mais très diversifiées : par exemple Renault a produit plus de 60000 variantes pour la R18 à la fin des années 70, plus de 200 000 variantes sur la Peugeot 306 au début des années 90 (environ 250 variantes de moteurs et 1000 variantes de caisses habillées). Toujours dans l'industrie automobile, la taille moyenne annuelle d'une série d'un même véhicule est de 40 spécimens ; le véhicule le plus vendu atteint moins de 8000 exemplaires.

Des pays comme le Japon sont les précurseurs de l'utilisation de nouvelles méthodes dont le mot clé est le "**JUSTE A TEMPS**". L'entreprise doit tendre à **PRODUIRE CE QUI EST DEJA VENDU**.

d.2- Les nouvelles règles de la production moderne :

Dans le contexte de la quatrième période définie précédemment, l'entreprise est face à plusieurs difficultés :

- Evoluer dans un marché volatil, mal maîtrisé, où les clients sont imprévisibles, infidèles à une marque, mais sensibles au délai ou à son respect, à la qualité, au service après-vente.

- Trouver des compromis entre stocks minimums, délais minimums et aléas minimums.
- Réduire les coûts de production, limiter les investissements, disposer de ressources flexibles, etc...

III- Gestion de la production dynamique:

Définitions...

Blondel [1997] définit la gestion de production comme étant la fonction qui permet de réaliser les opérations de production en respectant les conditions de qualité, de délai et de coûts qui résultent des objectifs de l'entreprise et dont le but est d'assurer l'équilibre très fragile entre :

- le taux d'emploi des ressources ;
- le niveau des encours et des stocks ;
- les délais.

Molet [1997] remarque à juste titre que cet équilibre est très difficile à obtenir puisque "l'on veut réduire en même temps stocks, délais, pannes, tout en accroissant la flexibilité, la variabilité des produits,... autant d'objectifs complexes, multiples mais souvent contradictoires et dont l'importance relative peut varier à chaque moment". Il en conclut que "la gestion de production reste, malgré les apports des outils informatiques, la gestion des compromis".

Selon **Courtois [1995]**, l'objectif principal de la gestion de production est de gérer les flux de matières et d'informations par rapport aux objectifs prioritaires définis par la direction générale de l'entreprise. Le schéma suivant montre l'ensemble des flux que gère, totalement ou partiellement, la gestion de production.

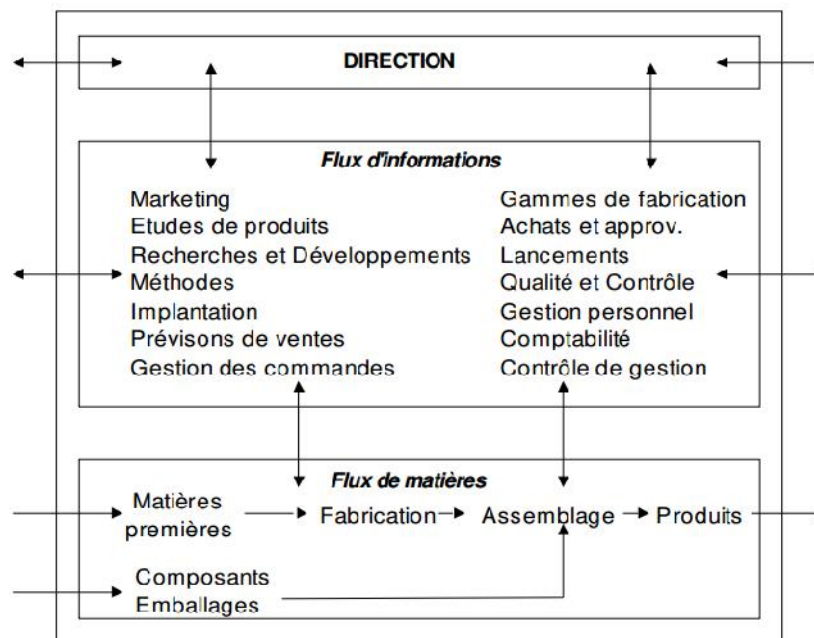


Figure 1 : Les flux informationnels et physiques

Objectifs...

L'objectif majeur de la gestion de production est de pouvoir livrer un produit avec la **qualité** adéquate dans les meilleurs **délais** tout en garantissant des **bénéfices**. A première vue, cela nous semble simple mais surtout facilement réalisable ; mais réellement ce n'est pas du

tout le cas. En effet, pour pouvoir satisfaire l'objectif général précédemment cité, il faut veiller à la rigueur et au respect des procédures définies, garantir la cohérence des décisions et la rapidité de circulation des informations.

L'objectif général étant défini, alors on se pose la question suivante : « comment y arriver ? », la réponse à cette question se traduit par la satisfaction des objectifs partiels suivants :

- **Objectifs techniques** : il faut :
 - réduire le cycle de production (délai commercial) ;
 - améliorer la qualité des produits;
 - veiller à-ce-que les ressources soient en quantité suffisante.
- **Objectifs Économiques** :
 - Diminuer les coûts de production pour avoir une marge bénéficiaire assez confortable: $\text{Marge} = \text{Prix de vente} - \text{Coût de revient}$
 - Diminuer (Optimiser) les stocks et les en-cours :

Stocks nuls : ils permettent de dégager l'argent immobilisé, libérer l'espace, mettre en évidence les dysfonctionnements (assurer la capacité de production optimale), diminuer le risque de détérioration, de vol, ...etc.

Mais : nécessitent de prévoir les achats, avoir des délais de livraison fiables et faibles, ne pas pouvoir satisfaire une demande aléatoire...

Coût des en-cours = Coût matière + Salaires + Frais de fonctionnement

- **Objectifs Humains** : Amélioration des conditions de travail et de sécurité.
- **Objectifs de pilotage** :
 - Maîtriser les stocks et les en-cours,
 - Réguler la charge de travail par poste,
 - Assurer la réactivité en cas d'aléas et de perturbations.

Il s'agira donc de répondre aux questions :

- Quels produits va-t-on fabriquer ou acheter ?
- Où seront-ils fabriqués ?
- Quand seront-ils fabriqués ?
- Qui les fabriquera ?
- Combien de temps sera-t-il nécessaire ?

Une telle gestion, réalisant ces objectifs, a pour conséquence :

- Augmenter la productivité : réduire le cycle de fabrication (temps de manipulation et de transport) ;
- Optimiser les achats (prix, qualité des achats, délais des commandes et de livraisons...);
- Diminuer les frais généraux ;
- Augmenter les gains.

Réaliser les objectifs partiels et arriver aux conséquences précédemment citées, nécessite obligatoirement la prise de décisions, mais quelles sont ces décisions et de quel type sont-elles ?

Les outils de gestion de la production sont un ensemble de techniques d'analyse et de résolution des problèmes, visant à produire au moindre coût. Pour situer ces différents problèmes entre eux, on classifie souvent les décisions de gestion en trois catégories:

1. **Les décisions stratégiques** : il s'agit de la formulation de la politique à long terme pour l'entreprise (c'est-à-dire à un horizon de plus de deux ans). Entrent dans ces décisions:

- la définition du portefeuille d'activités;
- la définition des ressources stables : aussi bien humaines (engagement, licenciement, préretraite, . . .) que matérielles (décisions d'investissement, de cession, de fermeture,..etc.);

2. **Les décisions tactiques** : il s'agit des décisions à moyen terme parmi lesquelles on trouve la planification de la production à 18 mois. Il s'agit de produire au moindre coût pour satisfaire la demande prévisible en s'inscrivant dans le cadre fixé dans le plan stratégique de l'entreprise (donc à ressources matérielles et humaines connues).

3. **Les décisions opérationnelles** : il s'agit des décisions de gestion quotidienne pour faire face à la demande journalière, dans le respect des décisions tactiques. Parmi ces décisions, on trouve :

- la gestion de stocks;
- la gestion de la main d'œuvre;
- la gestion des équipements.

Ces trois classes de décisions de gestion de production se différencient par un minimum de trois éléments :

a. par l'horizon de temps considéré :

- les décisions opérationnelles se prennent au quotidien;
- les décisions tactiques concernent la planification s'étalant sur 18 mois;
- les décisions stratégiques concernent la planification à long terme.

b. par le niveau d'agrégation :

- les décisions opérationnelles se prennent au niveau d'un atelier;
- les décisions tactiques se prennent au niveau d'une usine;
- les décisions stratégiques se prennent au niveau de l'ensemble de l'entreprise.

c. par le niveau de responsabilité :

- les décisions opérationnelles sont prises par les agents de maîtrise;
- les décisions tactiques sont prises par les cadres;
- les décisions stratégiques sont prises par la direction générale.

IV- Typologie de la production :

1. structure du produit :

Le produit, ainsi que son flux de production, peuvent avoir plusieurs structures [Courtois,1995]:

- **Convergente** : c'est le cas des produits réalisés à partir de l'assemblage de composants. Cette structure est caractérisée par une arborescence présentant plusieurs niveaux qui correspondent à des sous-ensembles du produit final. Cette arborescence met en évidence l'autre appellation que l'on donne à ce type de production "structure en A" [Marris,1996].

- **divergente ou "structure en V"**: cette structure est celle des produits réalisés à partir de la transformation d'une matière première unique : pétrole, lait, acier, etc....

- **parallèle** : les produits sont réalisés à partir de quelques matières premières faiblement transformées : industries de l'emballage, du pneumatique, ...

- **à points de regroupement ou "structure en T"**: cette structure dérivée de la structure convergente, est typique des produits réalisés à partir de l'assemblage de composants soit spécifiques (structure convergente) soit standards (points de regroupement) : cas de l'industrie automobile où, un même moteur peut être implanté dans plusieurs véhicules différents. Une structure en T cherche à concilier la production de masse et la personnalisation des produits. Il s'agit en fait d'un produit de conception modulaire, destiné à permettre la réalisation de produits sur mesure, en combinant de différentes façons, des sous-ensembles dont certains sont standardisés. Une telle conception conduit ainsi à définir un produit complexe au moyen d'une nomenclature arborescente à plusieurs niveaux, chaque niveau correspondant à un stade de la décomposition.

Les structures *divergentes* et *parallèles* correspondent à des usines fabriquant des produits peu variés, en très grandes séries, sur des périodes très longues avec un marché peu fluctuant.

Les structures *convergentes* et *à points de regroupements*, sont les plus fréquentes, correspondant aux industries manufacturières de type automobile, électronique, électroménager, ameublement, aéronautique, etc...

Cependant, il peut exister d'autres configurations intermédiaires. La figure ci-dessous résume un peu ces structures :

- **Divergente (débit), linéaire** : transformation de pièces,
- **Convergente** : assemblage de pièces pour réaliser des sous-ensembles et des produits.

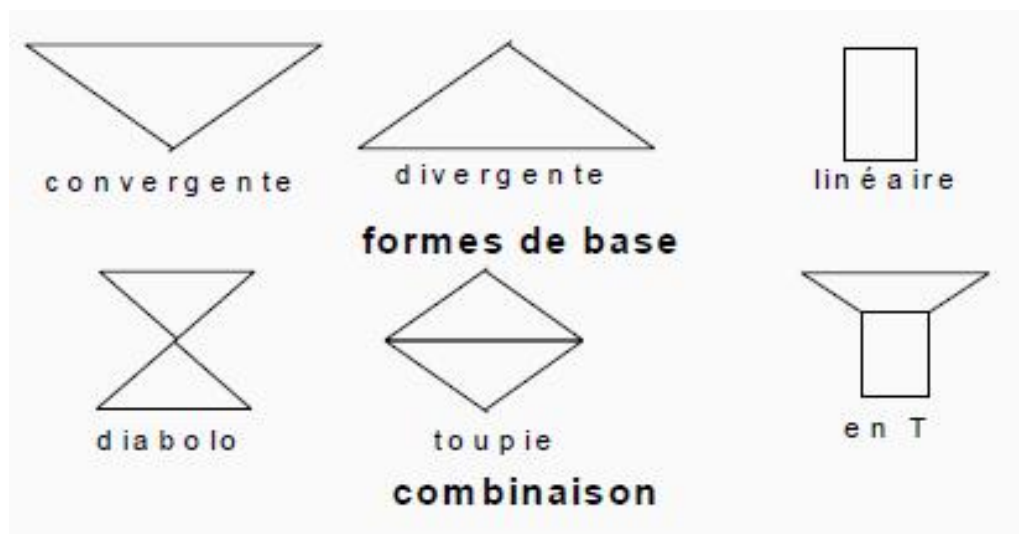


Figure 2 : typologie de production vue Produit /process

2. Les modes de production:

Les entreprises industrielles qui réalisent des produits implantent et organisent leurs ressources de production en fonction non seulement de la structure du produit, mais aussidu volume de production ainsi que par rapport à la circulation des flux de production. On peut ainsi établir une classification des modes de production [Woodward, 1965] :

- **Fabrication continue : Process Shop** : ce type de production est caractérisé par :
 - Quantités importantes de produits peu différenciés ;
 - utilise des lignes de production constituées de machines à vocation particulières, très fortement automatisée ;

- les produits circulent très vite grâce à des systèmes de manutention automatiques (tuyaux, pompes, convoyeurs...) ;
- les stocks d'encours sont très faibles ou nuls ;
- on cherche à équilibrer au maximum les postes ;
- on produit sur stocks établis de manière prévisionnelle ;
- il y a peu d'ordres de fabrication ;
- les opérateurs sont peu qualifiés et assurent la surveillance ;
- l'entretien et la maintenance sont préventifs ;
- la phase d'étude, conception de l'installation, choix du process est très longue et importante.
- **Fabrication linéaire en grande série : Flow Shop**
 - On fabrique en grande série des pièces appartenant à une même famille sur un ensemble de machines reliées par un système de manutention automatique ;
 - la production se fait sur stock, il existe un stock important de matière première et peu d'encours ;
 - flux poussé : les produits sont fabriqués à partir de prévision de vente, ou des commandes fermes ;
 - flux tiré : les produits sont fabriqués pour remplacer les produits vendus (stock) ;
 - les systèmes de production sont souvent des lignes de transfert dédiés.
- **Production discontinue : Job Shop**
 - Il s'agit d'ateliers à vocation polyvalente qui produisent des lots en petite série ;
 - les charges ne sont pas équilibrées et les en-cours sont nombreux ;
 - on a une surcapacité de production pour les machines courantes et quelques postes « goulets d'étranglement » ;
 - les ordres de fabrication sont très nombreux ;
 - l'objectif est d'assurer les délais et le plein emploi des personnels (gestion par la charge) ;
 - les personnels sont polyvalents et assurent une partie de la préparation du travail ;
 - dans certains cas, la politique de production est de réaliser les pièces standards sur stock et d'assurer la personnalisation des produits à la commande.
- **Fabrication à la commande, par projet**
 - L'entreprise travaille sur cahier des charges ou spécifications définies par le client ;
 - elle est maîtresse de son savoir-faire et de sa capacité de production ;
 - les ateliers comportent des machines à vocation polyvalentes, regroupées par nature ;
 - les ouvriers sont très qualifiés, la manutention est importante ;
 - la gestion devra raccourcir les délais d'étude et de réalisation, les coûts de réalisation ;
 - elle s'appuiera sur des méthodes de type PERT.

L'Association Française de la Gestion Industrielle (AFGI) a réalisé un schéma récapitulatif intégrant les paramètres les plus importants :

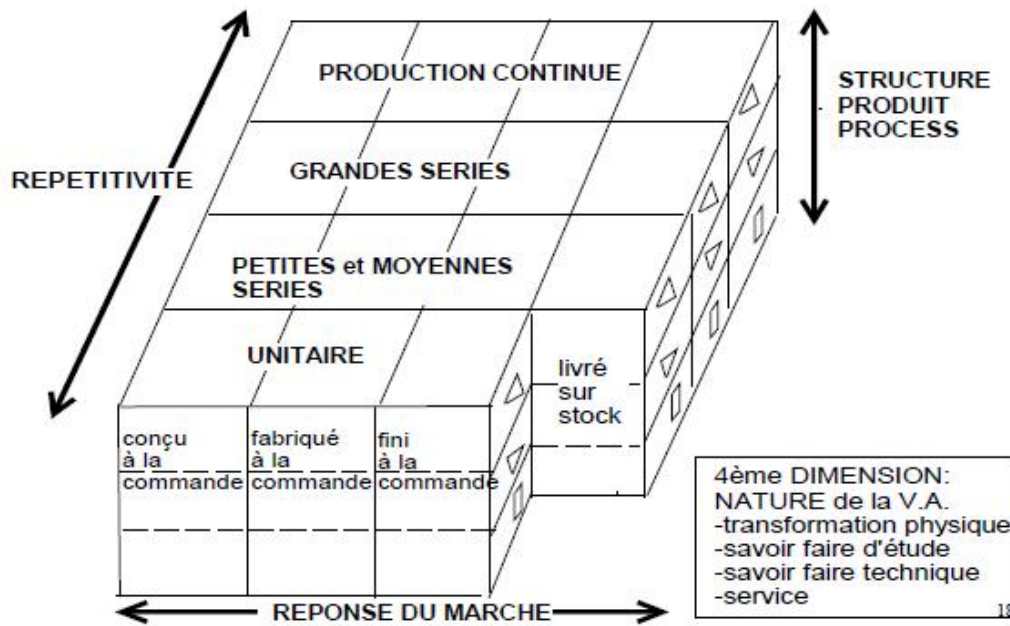


Figure 3: Typologies de production AFGI

3. Structure des ateliers de production :

Cette typologie est très liée à la typologie du produit. On distingue les classes les plus importantes dans [Widmer, 1991] :

- **Machines uniques** : dans ce cas toutes les tâches à réaliser se font sur une seule machine lors d'une seule opération.
- **Machines parallèles** : cette configuration permet d'avoir un ensemble de machines identiques disposées parallèlement. Chaque machine exécute un travail en une seule opération.
- **Ateliers à cheminement unique (Flow-shop)** : il s'agit d'un atelier regroupant un certain nombre de machines différentes, exécutant des tâches différentes, mais dédiées c'est-à-dire que chaque machine a un travail précis à réaliser lors du processus de transformation. Les machines sont implantées en fonction de leurs séquences d'intervention dans la gamme de production et sont alignées sur une ligne droite. Le produit final passe par toutes les machines, suivant un ordre de transformation très précis.
- **Ateliers à cheminements multiples (Job-shop)** : comme son nom l'indique, il s'agit d'un atelier composé de différentes machines disposées sur plusieurs lignes de transformation. Ces lignes sont reliées entre elles, de telle sorte qu'un produit peut démarrer sa transformation sur une ligne et la terminer sur une autre. Les produits fabriqués sont très variés et peuvent empreinter des cheminements différents les uns des autres ; pour cela il faut que l'atelier soit très flexible.
- **Autres configurations (Open-shop et autres)** : il existe une multitude de configurations intermédiaires ou mixtes.

Cette classification est très répandue en milieu industriel, elle a ouvert un champ très vaste de recherche, notamment concernant le problème de l'ordonnancement des tâches au sein d'une unité de production.

V- L'ordonnancement : problème et définitions...

1. Définition et objectifs :

Dans le système de production, le problème d'ordonnancement consiste à organiser dans le temps, l'exécution d'opérations interdépendantes à l'aide de ressources disponibles en quantités limitées, pour réaliser un plan de production [Erschler et al., 1992].

En se basant sur les concepts de tâche, ressources, contraintes et objectifs, l'ordonnancement peut être défini par :

« Ordonnancer un ensemble de tâches, c'est programmer leur exécution en leur allouant les ressources requises et en fixant leur date de début » [Carlier et al., 1993].

Les tâches sont soumises à des contraintes et l'ordonnancement, c'est à dire une solution au problème ainsi défini, est évalué vis-à-vis d'un ou de plusieurs objectifs à atteindre. L'objectif le plus souvent rencontré dans la littérature est l'obtention d'un ordonnancement optimisant un certain nombre de critères [Mesghouni, 1999].

La détermination de ces objectifs s'avère souvent extrêmement délicate. En effet, on se trouve généralement en présence d'un ensemble d'objectifs plus ou moins contradictoires et dont l'importance relative est difficile à apprécier. Parmi ces critères, on peut citer :

- Le coût et la durée de réalisation ;
- le respect du délai d'exécution ;
- la quantité de moyens nécessaires ;
- la quantité de travail en attente ;
- le temps d'immobilisation des moyens.

A partir de là, on peut dire que la fonction ordonnancement gère le temps, et rythme la vie de l'usine. La définition classique dit que cette fonction :

- est responsable des délais de fabrication ;
- assure la mise à la disposition, en temps opportun, des approvisionnements nécessaires à la fabrication ;
- prévoit et affecte les moyens humains et machines nécessaires au respect du planning.

On définit également le problème de l'ordonnancement, de la programmation ou de la planification d'un projet, comme étant la coordination d'opérations distinctes entre lesquelles il existe ou non, un nombre de contraintes temporelles d'un projet. Cette coordination sera telle qu'elle devra satisfaire à un objectif tendant, soit à minimiser la durée totale du projet ou son coût, soit à permettre une affectation optimale des moyens mis en œuvre, soit à trouver un ordonnancement optimisant une combinaison de ces trois facteurs.

Cette définition générale montre l'importance de la prévision pour les responsables de cette fonction. Dans un domaine aussi aléatoire que la prévision commerciale, qu'elle soit à court, à moyen ou à long terme, l'efficacité exige avant tout que l'ordonnancement soit en étroite liaison à la fois :

- avec le service commercial ;
- avec les méthodes et éventuellement les études ;
- avec les achats et le contrôle de réception ;

- avec les ateliers eux-mêmes ;
- avec le contrôle de fabrication et de livraison.

2. Le Problème de l'ordonnancement :

Dans cette section, nous allons essayer de décrire le problème de l'ordonnancement par rapport aux tâches qui lui sont confiées.

La tâche primordiale de l'ordonnancement consiste à définir ce qu'il faut fabriquer et pour quelles échéances. Cette détermination dépend à la fois du programme de vente et de la politique de fabrication choisie par l'entreprise.

La détermination du plan de production demande une connaissance exacte des moyens de l'entreprise, c'est-à-dire de sa capacité de production en fonction du temps, ainsi que du délai probable demandé par la fabrication, suivant le procédé choisi par les services techniques, majoré des divers temps d'attente et de manutention. Pour un lot défini de produits, ce temps total est appelé cycle de fabrication. L'ordonnancement doit posséder une connaissance précise des cycles de fabrication réels des principales familles de produits de l'entreprise, très différents, la plupart du temps, du cycle de fabrication théorique.

La deuxième tâche de l'ordonnancement est de réunir à temps les ressources : approvisionnements, pièces sous-traitées et matières premières nécessaires. Parallèlement, l'ordonnancement désigne les ateliers et sections, groupes de machines sur lesquels les fabrications seront exécutées. Il indique les besoins en opérateurs et provoque, si nécessaire, les affectations indispensables. Il peut aussi recourir à la sous-traitance. Mais il faut préciser qu'il s'agit, sauf urgence de dépannage, d'une décision fondamentale de la direction concernant la politique de fabrication.

La troisième tâche de l'ordonnancement consiste, dès que les moyens sont réunis dans les magasins et ateliers, à déterminer les périodes et calendriers d'exécution et à distribuer les documents nécessaires : c'est ce qu'on appelle le lancement.

Enfin, la quatrième tâche essentielle consiste à suivre l'exécution et à remédier aux aléas inhérents à toute organisation humaine. Suivre l'exécution réclame une capacité à saisir le plus rapidement possible au niveau des machines, le début et la fin des différentes phases. D'assez nombreux systèmes ont été utilisés, la plupart fondés sur la circulation d'une fiche suiveuse et le pointage des bons de travail.

Ces objectifs peuvent correspondre à des exigences quantitatives (valeurs à atteindre ou à ne pas dépasser) et se présentent sous forme de contraintes à respecter, ou bien à des exigences qualitatives s'exprimant sous forme de critères à optimiser. Parmi les critères les plus utilisés pour évaluer la qualité d'un ordonnancement obtenu, on distingue [**Carlier et Chrétienne, 1988**]:

- **Durée totale (Makespan) :** La durée totale de l'ordonnancement est égale à la différence entre la date d'achèvement de la tâche, la plus tardive, et la date de départ de la première tâche. La minimisation de cette durée est le critère le plus souvent rencontré puisque cela conduit inévitablement à une utilisation efficace des ressources.

- **Respect des dates au plus tard :** Dans beaucoup de problèmes réels, le respect au mieux des délais s'obtient en minimisant soit le plus grand retard soit la somme des retards.

- **Minimisation des coûts :** Ce genre de critère peut s'exprimer sous des formes très variées telles que, par exemple, la minimisation des stocks d'encours.

3. l'optimisation et les systèmes d'ordonnancement :

L'ordonnancement doit satisfaire au mieux des contraintes contradictoires deux à deux :

- Respect des délais ;
- meilleur emploi des moyens en hommes et en matériel ;
- réduction de l'appel aux moyens financiers de l'entreprise par diminution des stocks et des en-cours dont la masse est, trop souvent, incontrôlée.

Ce triangle de contraintes doit être optimisé. Mais le nombre des paramètres en jeu et l'incertitude des quantifications rendent sans effet une démarche logique : dans ce domaine, il n'existe que des approximations successives, conduisant à des propositions de décisions soumises à la subjectivité de jugements humains.

Il doit être clair que l'introduction de systèmes d'ordonnancement automatisés, plus ou moins intégrés, si elle fournit généralement une amélioration incontestable de la gestion de la production, elle ne conduit pas à une solution du problème de l'optimisation.

Ces systèmes d'ordonnancement automatisés apportent cependant, à la fois une aide importante à la prise de décision, et contribuent très efficacement à améliorer la gestion des nomenclatures, la gestion des stocks, le respect des délais, les divers approvisionnements. Ils conduisent à une simplification et à une mise à jour des gammes.

L'optimisation constitue un des objectifs essentiels de l'ordonnancement ; cependant, pour y arriver, il faut définir la portée temporelle de l'ordonnancement c'est-à-dire : à long terme, à moyen terme, à court terme, ou à très court terme, car l'objectif recherché n'est pas forcément le même.

3.1- Objectifs à long terme :

A long terme, il convient d'apprécier si l'ensemble des prévisions commerciales (commandes attendues et alimentation du stock), permet la meilleure utilisation du potentiel industriel de l'entreprise. On est donc amené à faire un bilan Charges/Capacités, poste par poste, au moins pour les principaux ateliers ou sections et, en particulier, celles concernant les machines les plus coûteuses.

En cas de sous charge, il faut que l'ordonnancement signale les capacités disponibles aux services commerciaux, de façon à faire un effort adéquat. Au besoin, l'entreprise pourra se placer en position de sous-traitance pour certains ateliers. Si le bilan montre une surcharge pour tout ou partie de ses ateliers, l'ordonnancement s'efforce d'appliquer la politique définie par la direction générale dans cette hypothèse, soit :

- recours à la sous-traitance ;
- stockage, s'il s'agit de surcharge saisonnière ;
- heures supplémentaires ;
- recours à du personnel intérimaire ;
- création d'équipes supplémentaires avec embauche ;
- construction de nouveaux ateliers.

Il sera toujours utile de rechercher, par un calcul, quelle est la solution la plus favorable sur le plan économique, mais il sera souvent nécessaire d'évaluer le coût social des solutions envisagées.

3.2- Objectifs à moyen et court terme :

• **A moyen terme** : on répartira plus exactement les charges entre les établissements ou ateliers, en s'efforçant de déterminer l'échéancier normal des commandes.

Parallèlement, on fixera le dispositif de réalisation des différents approvisionnements, compte tenu des délais moyens demandés par les fournisseurs. Si l'échéancier s'écarte de

façon plus ou moins importante des dates de livraison demandées par les clients, on fera intervenir les ordres de priorité permettant de jouer sur les différents délais d'attente ou de manutention, puis on envisagera les moyens disponibles de fractionnement et de chevauchement des opérations. L'intervention de ces différentes mesures qui, dans le cas de l'Informatique, seront provoquées par l'introduction de paramètres adaptés aux caractéristiques de l'entreprise et aux décisions des responsables à leur niveau, permettra de fixer à court terme l'ordre desancements, par période d'ordonnancement, tel que la semaine de 5 jours ouvrables, la période de 2 jours ou la journée, ainsi que le jalonnement des opérations concernant chaque commande. On devra, parallèlement, disposer d'un plan de travail par poste, qui permettra, soit de vérifier que chaque poste est judicieusement employé, soit de prendre les mesures complémentaires qui permettraient d'améliorer la charge.

- **A très court terme :** l'atelier s'efforcera d'assurer le meilleur déroulement des opérations et d'obtenir à ce niveau une optimisation de l'emploi des moyens et du respect des délais.

4. Les types d'ordonnancement :

Généralement on distingue deux types d'ordonnancement :

a. Ordonnancement prévisionnel :

Il a pour but de générer un ordonnancement optimal, minimisant un critère donné ou une combinaison de plusieurs critères. Les problèmes d'ordonnancement ainsi définis sont des problèmes d'optimisation combinatoire, difficiles à résoudre au moyen d'un algorithme polynomial [Lawler et al., 1989]. La solution recherchée peut être obtenue soit par des méthodes exactes (Programmation linéaire [Portmann et Bolzoni, 1994] ou dynamiques [Shrage et Baker, 1978 ; Bellman et al., 1982]), utilisées pour traiter des problèmes de très petite taille (exemple: problème dû à une machine), soit par des méthodes heuristiques [Lévy, 1996], se basant sur des algorithmes approchés, souvent polynomiaux, qui permettent de résoudre des problèmes complexes. Les heuristiques ne garantissent pas l'obtention de la solution optimale, mais fournissent, dans un laps de temps raisonnable et à un coût acceptable, une solution dont les performances sont en général assez bonnes et un ordonnancement pour lequel la valeur trouvée est proche de la valeur optimale.

b. Ordonnancement réactif :

C'est un système d'ordonnancement qui inclue une méthode pour réagir en temps réel face aux aléas [Lamothe, 1996]. Ces aléas peuvent être internes, survenant à l'intérieur de l'atelier (pannes de ressources, absence de personnel, ...) ou externes, provenant de son environnement (retard d'approvisionnement, arrivée imprévue d'un ordre de fabrication, ...).

VI- Pilotage d'un système de production :

Le pilotage des systèmes de production a fait l'objet de nombreux travaux et de publications depuis deux ou trois décennies, que ce soit dans le milieu industriel ou dans le milieu académique. Cependant, force est de constater qu'il n'y a pas d'accord complet sur la définition de ce vocable, ni même sur ce qu'il recouvre. En effet, selon que l'on a à faire à des automaticiens, des gestionnaires, des chercheurs en génie industriel ou des éditeurs de logiciels, les définitions changent et parfois se complètent.

Pour les dictionnaires, piloter c'est tout à la fois conduire, diriger, et gouverner, implicitement, le pilotage fait référence à un système complexe. Ceci se retrouve dans le langage commun : on conduit sa voiture, mais on pilote une formule 1 ou un avion.

Les organisations industrielles ont subi de nombreuses mutations ces dernières décennies. Comme tout autre système, les organisations industrielles sont contraintes d'évoluer sous la pression de leur environnement. Or, on est obligé de constater que le monde, sous l'influence de l'homme, a connu plus de changements ces dernières décennies qu'il n'en avait connu au cours des millénaires précédents. Ceci est tout aussi valable pour l'environnement industriel. Si l'on se restreint aux entreprises, leur évolution est principalement due :

- au renversement du rapport entre l'offre et la demande ;
- à la mondialisation et la globalisation de l'économie ;
- aux mutations techniques et technologiques.

Les entreprises ont dû se remettre en question et modifier leur appréhension de l'environnement. Ces évolutions vont aussi impacter les modes de pilotage des systèmes industriels.

(i) D'une part, prises dans une économie de marché de plus en plus contraignante, les entreprises de production sont confrontées à des demandes clients de plus en plus volatiles, mais aussi à une concurrence de plus en plus forte au fur et à mesure que disparaissent les barrières douanières (marché commun ou accords commerciaux internationaux). Face à cette double contrainte, elles ont été conduites à adapter leur finalité. Par la suite, les priorités sur les attendus du système de production ont été changées. On parle ainsi de pilotages par les coûts, les délais, la qualité, etc.

(ii) D'autre part, les appareils de production ont fortement évolué, tant au niveau matériel qu'au niveau organisationnel. L'utilisation de machines-outils à commande numérique, de lignes flexibles et l'automatisation permettent par exemple de réaliser un compromis entre la productivité des lignes de production fordienne et la flexibilité/réactivité des machines servies par l'opérateur humain. Les nouvelles formes d'organisation du travail, la polyvalence, permettent de reporter sur les opérateurs une partie des décisions.

Les équipements et les machines sont de plus en plus automatisés et les ressources productives sont devenues de plus en plus coûteuses. L'arrivée des systèmes informatiques permet de remonter plus facilement les informations sur l'état réel du système et de fournir des systèmes d'aide à la décision plus performants.

Les trois composantes essentielles d'un système à piloter :

1. *le système physique*, qui comprend tout à la fois l'engin à piloter, les équipements (capteurs) permettant de surveiller la trajectoire et les gouvernes (actionneurs) permettant de la modifier,
2. *le système d'information*, qui assure la collecte et la mémorisation des informations provenant de l'engin et de l'environnement,
3. *le système de décision* qui, en fonction des informations reçues et des indicateurs de performance retenus, décide des actions à mener pour atteindre les objectifs.

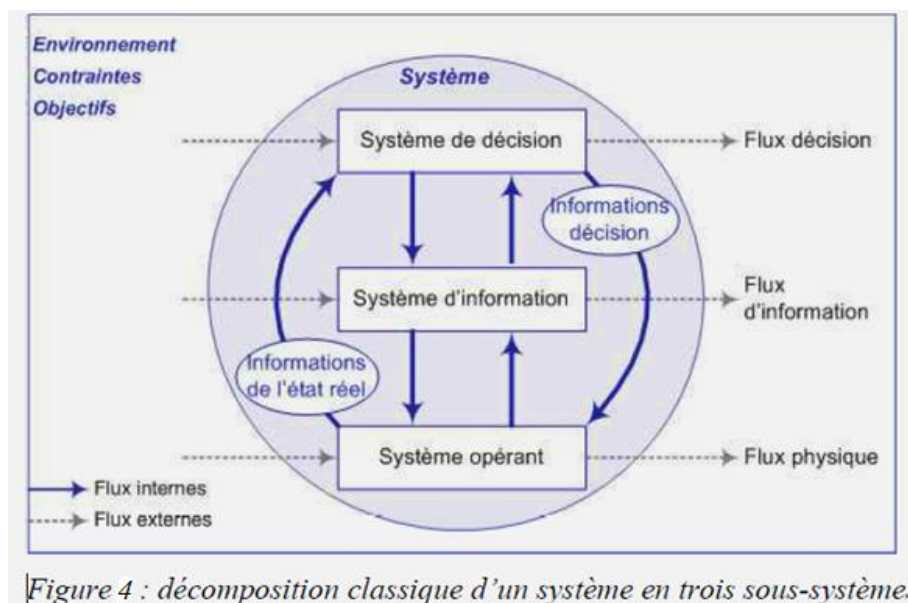
1- Pilotage des systèmes industriels :

Parmi les nombreuses définitions du pilotage industriel, la première que nous retiendrons est celle de **Mélèze [1972]**, qui fait ressortir le concept de « module de pilotage ».

Cette entité, base de l'analyse modulaire des systèmes, constitue le premier formalisme spécifique, dédié à la compréhension du mécanisme de prise de décision, en particulier dans les systèmes de production. Pour lui, ce module de pilotage comprend un décideur ainsi que des sous-systèmes décisionnels et d'évolution.

Le Moigne [1974] introduit la séparation entre système physique, système d'information et système de décision. Pour lui, le pilotage concerne la définition et l'organisation des relations entre sous-système physique et sous-système de décision, ainsi que l'organisation des activités liées à la prise de décision. Dans les systèmes de production, ceci induit une notion de boucle fermée : une activité n'est réellement décisionnelle que s'il existe une boucle de rétroaction entre le système physique et le système décisionnel. Ce qui va permettre au système décisionnel de contrôler que les résultats obtenus sur le système physique sont conformes à l'attendu, et dans le cas contraire, va lui permettre de rectifier les ordres.

Le schéma désormais classique de Le Moigne fait ressortir le rôle du système d'information, qui fait l'intermédiaire entre les sous-systèmes physiques et de décision (Figure 2). Ceci met en exergue le fait que, la plupart du temps, les décisions sont prises non pas en se basant sur la réalité du terrain, mais en se basant sur les informations que l'on a de ce terrain.



Ceci implique deux niveaux de risques dans la prise de décision ; une décision peut s'avérer inadaptée :

- soit parce que le système de décision est mal conçu et/ou mauvais ;
- soit parce que les informations sur lesquelles il se fonde sont erronées.

Considérons un exemple très simple. Dans une gestion de stock à point de commande, on passe une nouvelle commande lorsque le stock restant atteint un seuil s donné. Si l'on tombe quand même dans une situation de pénurie :

- (i) soit la valeur du seuil s a été mal calculée (erreur du système de décision) ;
- (ii) soit l'information que le stock réel a atteint le seuil s n'a pas été transmise au moment opportun et la commande a été déclenchée trop tard par le système de décision.

Avec l'arrivée de l'automatique, des technologies informatiques et des logiciels industriels à tous les niveaux des systèmes de production, on dispose actuellement de moyens de plus en plus performants pour remonter les informations du terrain vers les systèmes d'informations. Paradoxalement, les risques d'erreurs croissent aussi. Arriver à faire coïncider système réel et système d'information est d'autant plus crucial que l'on travaille en temps réel.

Selon l'approche automatique [Breuil, 1984], on retrouve derrière le concept de pilotage un « système physique » ou partie opérative, qui effectue les opérations réelles et un « système de commande ». La partie opérative effectue les opérations ordonnées par la partie commande. La partie commande élabore les ordres à partir des informations et des retours venant de la partie opérative et des objectifs assignés.

Pour Doumeingts [1984], piloter un système c'est « assigner à chaque partie du système, un ou plusieurs objectifs à atteindre ». Par nature, le système est donc finalisé. Les activités de pilotage ont pour rôle de guider chacune des parties du système vers ses objectifs.

Le système doit, là aussi, posséder des dispositifs de retour d'information et de « feed back » pour assurer la concordance entre les objectifs assignés et les résultats obtenus. Cette définition laisse ouvert le problème de la cohérence entre les objectifs des sous-systèmes et du système complet.

Lorino [1992] place l'objectif global de l'entreprise au centre du processus de pilotage. Pour lui, piloter, c'est définir des règles de comportement cohérentes avec cet objectif global. En termes actuels, on retrouve la notion de processus. L'objectif global de l'entreprise est l'aboutissement d'un ensemble d'activités liées, autrement dit d'un processus. Piloter c'est donc :

- déterminer le processus supérieur de l'entreprise ;
- décliner ce processus en sous processus ;
- s'assurer que chaque activité participe en cohérence avec les autres pour atteindre de l'objectif global.

Par la suite, Lorino [1997] assimile la performance de l'entreprise au surplus que dégage l'entreprise (différence entre la valeur fournie et les ressources consommées). En termes de processus on parlerait maintenant de création de valeur ajoutée monétaire. Nous verrons par la suite que, dans le contexte actuel, la valeur créée par une activité doit s'estimer en termes plus larges que le simple flux financier (qualité, réactivité, délais). *In fine*, au niveau de l'entreprise, la performance se traduira par des rentrées monétaires. Dans ce cadre, le pilotage tente de répondre à la question suivante : quelles sont les règles d'actions et de décisions à mettre en œuvre au niveau de chaque activité pour assurer à l'entreprise une création de valeur suffisante.

Le fait que le pilotage doive se décliner au niveau de l'activité pose de nombreux problèmes, en particulier :

(i) comment s'assurer que les objectifs particuliers et parfois contradictoires de chaque activité convergent vers l'objectif global ;

(ii) comment gérer les interdépendances entre les activités : une action menée sur une activité peut influencer en cascade la performance des activités qui la suivent ;

(iii) comment synchroniser des activités en parallèle, qui concourent à une même réalisation et gérer les décalages temporels entre activités en série.

De nombreux auteurs restreignent le pilotage de la production au pilotage de l'atelier proprement dit et en temps réel. **Trentesaux [1996]** définit le pilotage comme « la structure de décision et d'information associée à la gestion en temps réel ». Ce pilotage en temps réel comprend deux sous-fonctions liées par une boucle de rétro action :

- (i) la conduite, qui est la fonction décisionnelle qui donne les ordres ;
- (ii) la commande, qui exécute les ordres décidés par la conduite et lui retourne les données de suivi.

Pour **Grabot et Huguet [1996]**, le pilotage permet la production à court terme de l'atelier, en accord avec les objectifs établis par la gestion de production et en adaptant la production aux perturbations pouvant survenir au niveau de l'atelier et de son environnement.

Pour terminer ce tour volontairement restreint de la littérature, nous présentons la définition de l'APICS. Ce terme de pilotage d'atelier au sens général, ou « Production Activity Control » [**Apics, 2005**], est la fonction qui consiste à faire cheminer et à répartir le travail à exécuter dans les unités de production ainsi qu'à piloter les fournisseurs. Le pilotage d'atelier recouvre les principes, méthodes et techniques nécessaires pour ordonnancer, piloter, mesurer et évaluer l'efficacité des opérations de production. Dans le cas particulier d'un atelier manufacturier ou « Shop Floor », la gestion et le pilotage utilisent les données en provenance du terrain pour mettre à jour et communiquer la situation des ordres de fabrication et des centres de charge, ainsi que pour contrôler les mouvements de matières dans l'usine. Les sous-fonctions principales de la gestion d'atelier sont les suivantes : affecter une priorité à chaque ordre de fabrication, gérer le niveau des en-cours, donner la situation des ordres de fabrication, communiquer les sorties réelles pour gérer la capacité, fournir la quantité par emplacement et par ordre de fabrication pour gérer et comptabiliser les en-cours et mesurer l'efficacité, l'utilisation et la productivité de la main-d'œuvre et des machines.

Les sous-fonctions principales du pilotage des flux sont basées d'abord sur les cadences de production, sur un chargement d'atelier correspondant, puis sur le contrôle et la gestion de la production.

2- Fonctions génériques du pilotage de la production :

D'après **Le Moigne [1974]**, « décider » c'est identifier et résoudre les problèmes que rencontre toute organisation. La conduite des systèmes de production peut se décrire selon les trois niveaux hiérarchiques de décision : stratégique, tactique et opérationnel qui correspondent respectivement à des décisions à long terme, moyen terme et court terme.

(i) Les décisions stratégiques sont les décisions qui introduisent des orientations durables de l'entreprise sur un horizon à long terme. On rangera dans cette catégorie, par exemple, les décisions concernant l'implantation de nouvelles unités de production, la conception d'une nouvelle ligne de production, les décisions portant sur de nouvelles orientations en termes de produits ou de marchés. Si une réflexion stratégique est nécessaire tout au long de la vie de l'entreprise, les questions stratégiques se posent plus particulièrement quand l'entreprise affronte des changements importants tels qu'une hausse ou une baisse significative des demandes ou encore une fusion ou une cession d'une partie de l'activité.

(ii) Les décisions tactiques concernent la mise en œuvre et l'adaptation des décisions prises au niveau stratégique sur une période plus courte. Elles concernent particulièrement la planification mensuelle de la production afin de répondre à la demande, des flux matières tout au long de la chaîne logistique (approvisionnements, production, stockage, distribution) et la gestion des ressources en hommes et équipements.

(iii) Les décisions opérationnelles nous intéressent plus particulièrement. La décision à ce niveau met en application le plan d'actions prévu par le niveau tactique. Ce sont les décisions qui concernent les actions à court terme, comme par exemple l'ordonnancement de la production. La procédure de prise de décision intervient durant la production de deux façons : de façon périodique ou événementielle. Dans le premier cas, le déclenchement de la procédure de prise de décision est régulier et il est fait à des intervalles de temps réguliers. Dans l'autre cas, le déclenchement repose sur l'apparition d'événements dans le système. Par la suite, nous détaillerons ces deux cas en les appliquant sur une ligne expérimentale.

Si l'on s'intéresse plus spécifiquement aux activités de gestion de production manufacturière, on va retrouver classiquement cinq fonctions essentielles qui sont :

- *La planification ;*
- *la programmation ;*
- *l'ordonnancement ;*
- *la conduite ;*
- *la commande.*
-

VII- Typologie des structures de pilotage :

Les systèmes de pilotage peuvent s'organiser selon diverses architectures. Nous établissons une description et une comparaison de chacune d'entre elles.

1. Structure centralisée :

Il s'agit de l'approche la plus classique et la plus ancienne. Ici, toutes les ressources sont pilotées par un centre de décision unique, qui supervise la production, synchronise et coordonne les différentes ressources et gère en temps réel les imprévus qui surviennent. Le pilotage se fait essentiellement sur la base d'un ordonnancement prévisionnel des différentes tâches (intégrant donc l'ensemble des ressources et des entités). Cet ordonnancement est calculé sur des valeurs moyennes des charges machines et des durées d'exécution. Ceci permet de conserver une certaine souplesse d'exécution (maintenance, réparation...) et de garder une petite marge de sécurité. Les perturbations limitées peuvent être absorbées. Les perturbations trop importantes entraînent le recalcul d'un nouvel ordonnancement. Cette approche est adaptée aux petits systèmes de production, elle se rencontre fréquemment dans les petites entreprises où l'essentiel du pilotage consiste bien souvent à réordonner manuellement les tâches.

2. Structure hiérarchisée :

Dans le cas de systèmes de production plus complexes, on est amené à décomposer le système en sous-systèmes. Cette décomposition est effectuée :

- par fonctions (approvisionnement, ateliers, logistique aval) ;
- par horizons (court, moyen et long termes).

Dans un schéma de décision hiérarchisée, un centre de niveau supérieur coordonne et fixe un cadre de décision pour les niveaux inférieurs. Ce cadre de décision va leur fixer les objectifs et les contraintes à respecter. Les niveaux inférieurs se voient ainsi déléguer un certain degré d'autonomie de décision (dans le cadre fixé). En retour, ils doivent remonter au niveau supérieur des informations sur les résultats atteints. Chacune des structures est donc dotée de deux fonctions principales :

- une fonction de planification : les ordres reçus des niveaux supérieurs sont traités et décomposés en sous-ordres transmis au niveau inférieur,
- une fonction d'observation : chaque structure de niveau inférieur envoie un compte rendu à la structure de niveau supérieur. Celle-ci collecte les différents comptes rendus, les traite et émet à son tour un compte-rendu global vers le niveau supérieur.

Chaque structure va donc pouvoir gérer un certain nombre d'aléas à son niveau. L'apparition de perturbations de plus grande ampleur entraîne cependant la remise en cause des décisions des niveaux supérieurs. Par rapport au modèle centralisé, ce modèle permet de limiter l'impact des perturbations, mais nous constatons l'absence de liaisons transversales entre les différents sous-systèmes.

3. Structure coordonnée :

Les liaisons entre les différents sous-systèmes de pilotage d'un même niveau (ex : production, maintenance) sont ajoutées dans la structure dite «coordonnée ». Cette architecture augmente la capacité de décision dans chacun de ces niveaux pour avoir une résolution locale des problèmes. La coopération entre les différents sous-systèmes de pilotage cherche à optimiser la réponse et à mieux réagir localement en cas d'apparition de perturbations.

De nombreux travaux de recherche s'intéressent à l'application de cette approche comme par exemple CODECO [Pellet, 1985], qui peut être considérée comme une extension, par l'intégration de la coopération intra niveau du modèle ORABAID. Chaque sous-système de pilotage du niveau inférieur organise ses fonctions localement, selon ses contraintes et en collaboration avec les autres sous-systèmes de même niveau. La difficulté se situe dans la détermination du degré de liberté de décision pour chaque poste de conduite locale.

Citons également le modèle de conduite PCS [Archimède, 1991], qui considère le développement d'une fonction de réactivité basée sur la définition et l'exploitation d'une typologie de perturbation.

4. Structure distribuée :

Cette structure résout la difficulté rencontrée dans la structure coordonnée pour la communication et la circulation d'information entre différents niveaux. Elle est basée sur une répartition des capacités de décisions sur un ensemble de centres de pilotage. Cette approche est particulièrement adaptée dans les productions avec des flux simples, des demandes stables ou à faible variation et l'apparition d'aléas réguliers ou connus.

5. Structure distribuée supervisée :

En général, cette structure offre des possibilités d'intercommunication des systèmes de pilotage de différents niveaux, permettant l'échange d'informations et la transmission des décisions prises. Cette structure est caractérisée par la coopération entre les sous-systèmes dépendant d'un même système supérieur. Ce système supérieur sert à commander ou corriger une décision prise afin de satisfaire les objectifs plus globaux, car il a une vision plus large du processus de production dans l'entreprise. Nous pouvons citer certaines expériences, comme les travaux de [Tacquard *et al.*, 1995 ; Trentesaux, 1996] qui présentent une structure avec une performance plus élevée au niveau de la réactivité et de la flexibilité. En effet, chaque élément a pour fonction de planifier sur ses sous-objectifs propres, de réagir face à des aléas et d'analyser les résultats. Un système superviseur contrôle le processus de production de façon

plus générale pour établir les ratios de production sur plusieurs périodes, proposer différents choix lorsque les objectifs risquent de ne pas être atteints et synthétiser les résultats.

6. Structure décentralisée :

Dans cette structure, tous les centres de pilotage sont au même niveau fonctionnel, ce qui fait la différence de la structure distribuée. Il n'existe pas de centre de pilotage de niveau hiérarchique supérieur. Les centres de décision doivent s'auto-organiser pour assurer une gestion cohérente. Ces systèmes de pilotage totalement décentralisés sont restés marginaux, principalement en raison des problèmes de synchronisation entre les contrôleurs locaux.

La plupart du temps, on conserve une certaine structure hiérarchique. Les tâches de suivi de la production ou de lancement des opérations apparaissent comme des entités de haut niveau, distribuant le travail aux autres modules. Les centres des différents niveaux sont interconnectés via des bus de communication, permettant l'échange d'informations et la transmission des décisions. Ceci n'est rendu possible que par l'arrivée des nouvelles techniques informatiques (système d'information, bases de données partagées, logiciel de type MES) dans les ateliers de production.

En plus de ces six classes de pilotages, il existe d'autres types de pilotage en fonction des aléas, de l'origine de perturbation et de la période de lancement de la production.

L'industrie manufacturière est contrainte d'améliorer son pilotage d'atelier au niveau opérationnel pour réagir face aux aléas. Nous proposons de distinguer quatre types de pilotage selon que la décision est prise avant le lancement de la production ou en cours d'exécution et selon la nature des perturbations prises en compte (perturbation prévisionnelle ou réelle).

7. Pilotage prédictif :

Le pilotage prédictif est destiné à préparer le lancement d'un système existant. Il se fait donc *à priori*. Il permet de définir l'objectif de la production et la meilleure trajectoire pour y parvenir et de prendre des décisions pour assurer le fonctionnement courant. Ce pilotage est basé sur des paramètres (comme le temps opératoire d'une machine, le taux d'arrivée, le taux de panne etc.) estimés de manière déterministe (ex : le temps opératoire moyen est de 15 minutes) ou probabiliste (ex : le temps opératoire suit une loi uniforme entre deux valeurs minimum et maximum). Les paramètres de pilotage sont déterminés avant l'exécution sur le système réel. Un expert ou un outil est chargé de trouver la meilleure décision (ex. règle de priorité). Le pilotage prédictif fournit en quelque sorte le scénario optimal dans un environnement non perturbé.

Les outils utilisés sont l'ordonnancement prévisionnel, les méthodes de modélisation, la GPAO, les ERP mais aussi les outils de simulation hors ligne. Les meilleurs paramètres trouvés vont souvent être directement implantés dans des automates programmables, pour permettre l'exécution de la production. Mais ce pilotage pose généralement des difficultés après le lancement de la production, car ces paramètres choisis sont obtenus à partir d'un modèle et ne correspondent pas toujours à la réalité du terrain.

8. Pilotage proactif :

Le pilotage proactif est utilisé également dans la phase de préparation avant que le processus de production ne soit lancé. L'objectif ici est d'anticiper un certain nombre d'aléas et de déterminer *à priori* la meilleure réponse possible à apporter, si l'aléa survient. On va donc ainsi pouvoir répondre très vite et être beaucoup plus réactif. Ce pilotage est basé sur les mêmes principes que le pilotage prédictif, mais :

- il consiste d'abord à anticiper les perturbations les plus certaines et le plus grand nombre possible de perturbations éventuelles. Ces données peuvent être remontées grâce aux indicateurs de performance d'un système déjà existant ou bien extraites d'une base de données mémorisant les événements passés ;

- ensuite, il cherche à identifier les risques de ne pas atteindre les objectifs initialement fixés, notamment de ne pas respecter le délai prévisionnel face aux perturbations.

La simulation hors ligne est aussi un des outils utilisés pour ce type de pilotage afin d'anticiper les perturbations et calculer les risques éventuels de ne pas respecter l'objectif visé.

9. Pilotage réactif ou pilotage en temps réel :

Le pilotage réactif intervient pendant l'exécution de la production, une fois le lancement effectué. Il a pour but de corriger les valeurs des variables de décision lors de l'apparition d'un événement imprévu. Ce pilotage doit réagir en temps réel. C'est la raison pour laquelle on l'appelle aussi pilotage en temps réel. Dans le meilleur des cas, l'événement a été anticipé dans la phase de pilotage proactif et la réponse est connue. Souvent, l'événement n'a pu être anticipé. On peut distinguer deux situations nécessitant une réaction :

- Des événements imprévisibles peuvent survenir, sans que l'on puisse les anticiper par un pilotage prédictif ou proactif. Il peut s'agir par exemple de l'arrivée d'une commande urgente, de la modification d'une commande ou encore d'une panne sur une machine. Le pilotage réactif devient nécessaire pour analyser les conséquences de cet événement imprévu par rapport à l'objectif de production et, le cas échéant, pour déterminer les paramètres de pilotage à corriger pour minimiser l'impact de cette perturbation [Berchet, 2000],

- Des dérives sont détectées (par exemple, le temps de cycle d'une machine augmente dans le temps). Ces dérives peuvent avoir pour conséquence le non-respect des objectifs ou la survenue ultérieure d'événements plus perturbants. Par exemple, l'augmentation de la durée d'une opération peut entraîner un retard généralisé ou l'impossibilité de maintenir l'ordonnancement prévisionnel. Ceci permet au décideur d'appliquer, de façon préventive, des modifications aux paramètres de pilotage. La réactivité de ce pilotage dépend donc de l'évolution de l'état du système par rapport à ce qui avait été prévu par le pilotage prédictif.

VIII- Conclusion :

Cette partie du mémoire avait pour objectif de présenter le contexte général de la production. Nous avons expliqué les notions de base se rattachant à la production. Ensuite nous avons abordé de façon assez approfondie tout ce qui concerne la gestion de production, tout en soulignant son importance au sein d'un système de production. Aussi, nous avons abordé les notions relatives à l'ordonnancement, y compris le problème de l'ordonnancement et les problèmes à résoudre dans le but de trouver le meilleur ordonnancement pour atteindre nos objectifs.

Enfin, les différentes fonctions génériques du pilotage de la production ont été présentées selon le type d'atelier de production et selon les objectifs des entreprises. Nous avons expliqué aussi d'une part, les évolutions des systèmes de production et des modes de pilotage et d'autre part, les évolutions des ateliers de production en comparant leur architecture. Enfin, nous avons distingué les différents types de pilotage selon l'origine des

perturbations survenues et la période temporelle dans la production. Nous avons expliqué notamment un pilotage en temps réel ou pilotage réactif durant la fabrication afin de pouvoir réagir face aux événements critiques.

Le chapitre suivant permet de présenter l'aide à la décision en milieu industriel, en analysant les avantages et les difficultés d'intégration de ces outils, tout en expliquant le processus décisionnel et les Systèmes Interactifs d'Aide à la Décision (SIAD).

CHAPITRE II

SYSTÈME D'AIDE À LA

DECISION EN MILIEU

INDUSTRIEL

I- Introduction :

Les dirigeants d'entreprise d'aujourd'hui savent que les outils informatiques sont un support essentiel à la prise de décision. Le monde industriel, les administrations et les entreprises de services les utilisent quotidiennement pour apporter des réponses à tous les niveaux du processus décisionnel : pour la planification des investissements, l'optimisation de la conception des nouveaux produits ou des installations, pour la gestion financière, l'affectation des ressources, pour l'organisation au jour le jour des opérations, pour la prévision des demandes, pour l'analyse de données, etc.

Il existe une panoplie de produits informatiques et de techniques pour proposer une aide à la décision aux gestionnaires. Dans ce mémoire, nous nous intéressons tout particulièrement à une technique, qui est celle des *Systèmes Interactifs d'Aide à la Décision (SIAD)*.

Le domaine des SIAD est très vaste. Quelques auteurs proposent des états de l'art relativement complets [Davis *et al.*, 1986; Lévine et Pomerol, 1990]. Ce chapitre introduit en premier lieu la notion de prise de décision ainsi que quelques définitions liées à celle-ci. Cette présentation ne se prétend pas exhaustive, mais a uniquement pour but d'introduire un minimum d'éléments nécessaires à la présentation des SIAD. Par la suite, les différents concepts liés aux SIAD seront présentés ainsi que les principales barrières ayant freiné leur succès.

II- La prise de décision :

1. Définitions :

Pour Holtzman [1989], prendre une décision signifie concevoir et s'engager dans une stratégie d'allocation irrévocable de ressources. Le processus de prise de décision n'inclut pas l'allocation de ces ressources qui est appelée une *action*. De manière plus générale, on peut dire que tout individu placé devant plusieurs alternatives mutuellement exclusives choisit l'une d'entre elles à la suite d'un processus mental appelé *décision*.

On peut associer à toute décision un *domaine* qui correspond à son champ d'application. Par exemple, distinguer les décisions médicales, des décisions militaires ou industrielles. Ces domaines peuvent être eux-mêmes divisés en sous-domaines. Le domaine d'une décision est par définition générique, c'est-à-dire que plusieurs décisions partagent le même domaine. Par contre, certains facteurs d'une décision sont uniques et dépendent de la décision et du décideur. L'ensemble de ces facteurs est appelé la *situation* d'une décision. Cette situation se compose du *contexte* et des *préférences* du décideur. Chaque décision a un contexte qui lui est associé et qui affecte fortement l'intérêt et la disponibilité des différents choix pour le décideur. Ce contexte inclut l'état des informations détenues par le décideur et constitue sa perception des conséquences possibles de ses actions. Chaque décideur a des désirs particuliers qui sont exprimés par ses préférences sur les résultats possibles de sa décision. Par exemple, pour l'achat d'une voiture, le contexte d'un décideur pourrait inclure le fait d'avoir une famille, d'avoir l'habitude d'effectuer de longs trajets pour son travail et d'habiter dans une région désertique. Ses préférences peuvent inclure son enthousiasme pour les voitures 4X4, les embrayages automatiques ou la transmission intégrale avec ABS. À toute décision est toujours associée des *alternatives* parmi lesquelles le décideur doit choisir. Sans alternatives, il n'y a pas de décision. À chaque alternative est associé un *résultat* ou bénéfice espéré (au sens large du terme) qui peut guider le choix entre les alternatives.

Dans la section suivante, nous allons décrire les différentes phases d'un processus de décision ainsi que les différents niveaux de structuration d'un problème de décision.

2. Le processus de décision :

Le processus de décision est un ensemble d'actions déclenché par un événement, aboutissant à un engagement spécifique à l'action. Donc on peut dire que la décision est l'aboutissement d'un processus de réflexion. En 1950, Herbert SIMON a proposé un schéma explicatif de ce processus sous le nom de **IMC (Intelligence, Modélisation et Choix)**.

a. Les étapes du processus décisionnel :

Pour **SIMON [1977]**, la prise de décision se décompose nécessairement en trois phases distinctes :

Phase intelligence, au cours de cette phase le décideur analyse le problème posé, détermine les enjeux et recense les contraintes.

Phase Modélisation, elle permet la conception des solutions. A ce moment le décideur doit formuler et évaluer les solutions qui s'offrent à lui.

Phase Choix, après la conception des solutions possibles, le décideur doit choisir la meilleure en se basant sur des faits concrets ou bien, comme c'est le cas très souvent, le manager va s'appuyer sur son intuition bien que cela pourrait être très risqué.

Le processus décisionnel va être plus ou moins long selon qu'il s'agisse d'une décision stratégique ou opérationnelle.

b. Le processus décisionnel selon la structure de l'entreprise :

Une structure centralisée favorise la cohérence des décisions car elles sont prises par un petit nombre de décideurs qui détiennent la majorité des informations. Cependant, le processus de décision est ralenti par le temps nécessaire qu'il faut à l'information pour arriver jusqu'au sommet de la hiérarchie. L'efficacité des solutions est réduite du fait que les décisions ne soient pas toujours bien comprises, exécutées par une base écartée du processus.

Une structure décentralisée permet une meilleure pertinence des décisions prises par des managers de terrain, proche des problèmes à résoudre. La solution est mise en œuvre par des salariés plus impliqués.

En revanche, comme il y a décentralisation, il est nécessaire de prévoir des procédures de contrôle afin que les subordonnés respectent bien les solutions trouvées.

La prise de décision peut être soumise à six types de contraintes :

- budget ;
- temps de recherche d'information et de réflexion ;
- l'environnement (micro, macro, contraintes, opportunité) ;
- l'information (disponibilité et coûts) ;
- les connaissances, les expériences ;
- la personnalité : la perception des contraintes et des risques peut être différente d'un décideur à l'autre en fonction de sa personnalité

Jusqu'ici, un processus de décision comprenait les trois phases précédemment citées, mais on remarque qu'on ne peut pas voir ni la pertinence ni la justesse de la décision prise avant sa mise en pratique et les conséquences d'une mauvaise décision peuvent être très lourdes. Pour cela, dans le cadre défini par la gestion des organisations, on a proposé un processus décisionnel qui se compose de quatre phases : une phase d'information, une phase de conception, une phase de choix et une phase d'évaluation du choix [**Lévine et Pomerol, 1990**].

Lors de la phase d'information, il s'agit d'identifier les objectifs ou buts du décideur, c'est-à-dire de définir le problème à résoudre. Pour cela, il est nécessaire de rechercher les informations pertinentes en fonction des questions que se pose le décideur. Ensuite, la décision est classée parmi les différentes catégories connues. Pendant cette phase, l'acquisition d'informations pertinentes peut se poser elle-même en terme de décision. Pour certains problèmes, il est parfois difficile de trouver des informations pertinentes. Or, ce sont elles qui sont à l'origine du processus de décision et leur choix est crucial, car elles influencent fortement les autres phases, puisque tous les choix suivants en découlent.

La phase de conception comprend la génération, le développement et l'analyse des différentes suites possibles d'actions. Il va donc être nécessaire de choisir un ou plusieurs modèles de décision en fonction de la complexité du problème à traiter. Pour le ou les modèles choisis, il faut également déterminer les variables de décision, les variables incontrôlables, les variables résultats ainsi que les relations mathématiques, symboliques ou qualitatives [Grassland et Wynne, 1995] entre ces variables et construire les différentes alternatives.

Lors de la phase de choix, le décideur choisit entre les différentes suites d'actions qu'il a été capable de construire et d'identifier pendant la phase précédente. Cette phase inclut la recherche, l'évaluation et la recommandation d'une solution appropriée au modèle.

Finalement, la phase d'évaluation du choix correspond à une évaluation *a posteriori* du choix du décideur. Cette évaluation permettra éventuellement de corriger les petites erreurs. Comme le montre la Figure 1, ce processus n'est pas obligatoirement séquentiel, il peut y avoir des retours en arrière. Par exemple, pendant la seconde ou la troisième phase, le décideur peut être amené à générer une nouvelle alternative ou encore à rechercher de nouvelles informations, ensuite à modifier le ou les modèles choisis. La présence de ces retours en arrière pendant le processus de décision dépend du niveau de structuration du problème de décision.

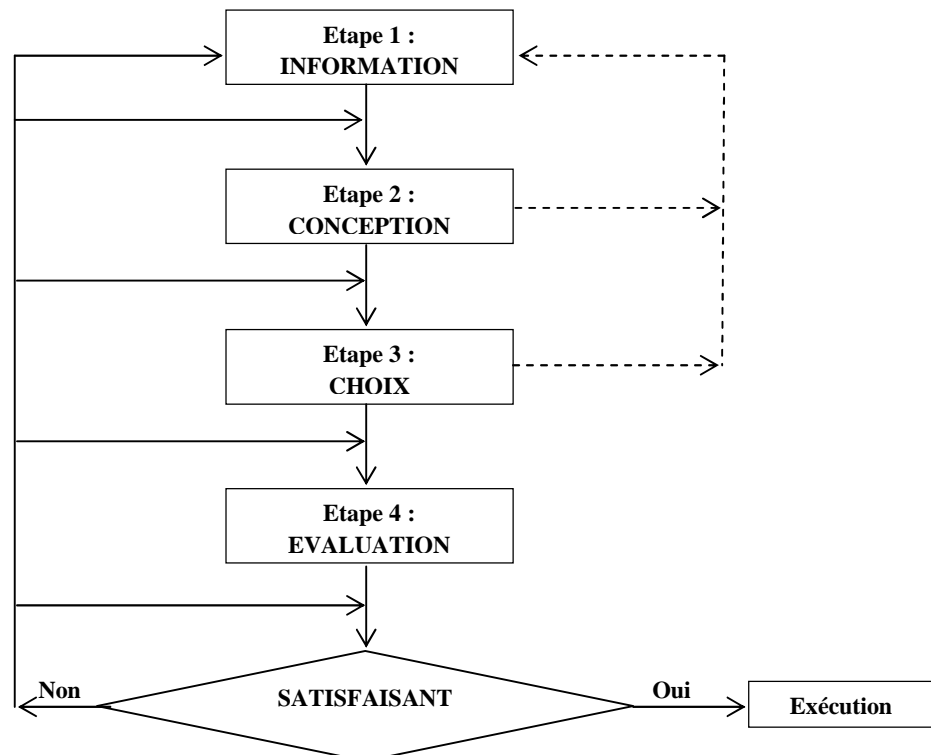


Figure 1 : Modèle du processus décisionnel [Reix, 2000].

3. Niveau de structuration des problèmes de décisions :

On distingue généralement *les décisions programmables* (correspondant aux problèmes structurés) *des décisions non programmables* (correspondant aux problèmes peu ou mal structurés). En pratique, il n'existe toutefois pas de dichotomie entre les problèmes structurés et les problèmes non structurés mais un continuum du degré de structuration allant des moins structurés aux plus structurés [Garlatti, 1996].

Un problème de décision est dit structuré s'il peut être exprimé par un modèle calculable, stable et s'il est accompagné d'une règle de choix invariante. Cette dernière fait en sorte qu'une décision structurée présente souvent un caractère répétitif. Le processus de prise de décision se réduit dans ce cas à l'exécution d'un calcul. Il s'agit alors de pseudo-décision de type algorithmique [Reix, 2000].

À l'opposé, un problème de décision peu ou mal structuré est un problème qui va nécessiter un effort important pour être formalisé. Ce genre de problèmes de décision possède en général trois principales caractéristiques [Klein et Tixier, 1971] :

- Premièrement, leur résolution est fortement dépendante des préférences, des jugements, de l'intuition et de l'expérience du décideur.
- Deuxièmement, les objectifs poursuivis lors de la prise de décision sont nombreux, en conflits et fortement dépendants de la perception de l'utilisateur. La recherche d'une solution pour ce genre de problème implique un mélange de recherche d'informations, de formulation du problème, de calcul et de manipulation de données.
- Finalement, ce sont des problèmes qui évoluent rapidement et dont la solution doit être obtenue dans un temps limité. Un des aspects les plus importants est que, pour cette classe de problème de décision, l'homme prend l'avantage sur la machine, contrairement aux problèmes structurés. Résoudre le problème exige de faire appel à l'intuition et au savoir-faire du décideur qui devient l'élément prépondérant du couple Homme/Machine. Dans ce cas, le contrôle de la recherche de solutions doit être laissé, en totalité ou en partie, au décideur.

Les SIAD, présentés dans la section suivante, ont été conçus pour résoudre la seconde catégorie de problèmes de décision [Eierman et Niederman, 1995].

III- Les Systèmes Interactifs d'Aide à la Décision (SIAD):

1. Introduction:

De nos jours, l'environnement des organisations est de plus en plus complexe, évolue rapidement et la tendance est plutôt à l'accroissement de cette complexité. Cet accroissement est dû à plusieurs facteurs comme la technologie, une complexité structurelle des décisions et une plus forte compétition nationale comme internationale causée par la mondialisation des économies. Par ailleurs, les conséquences financières et humaines d'une erreur de décision peuvent s'avérer dramatiques à cause des réactions en chaîne qu'elles pourraient engendrer sur les différentes parties de l'organisation. Il importe donc pour le décideur de pouvoir prévoir les conséquences éventuelles des décisions qu'il compte prendre.

Le concept de SIAD s'est formalisé dans la littérature dans les années 70 [Little, 1970; Gorry et Scott-Morton, 1971]. Ce sont des outils spécifiquement développés pour supporter la prise de décision. La conception de tels systèmes implique l'utilisation de techniques issues de divers domaines comme l'informatique, la recherche opérationnelle, l'intelligence

artificielle, l'ingénierie logicielle, l'interaction homme-machine et les télécommunications. Les SIAD s'avèrent particulièrement utiles pour aider à trouver une solution appropriée à des problèmes complexes, de grande dimension et ayant des objectifs fortement dépendants des préférences de l'utilisateur. En réalité, la plupart des problèmes d'optimisation font partie de cette catégorie de problèmes.

Résoudre un problème d'optimisation consiste souvent à déterminer la ou les meilleures solutions, vérifiant un ensemble de contraintes et d'objectifs définis par l'utilisateur. Pour déterminer si une solution est meilleure qu'une autre, il est nécessaire que le problème introduise au moins une mesure de performance permettant d'effectuer une comparaison.

Cette mesure de performance correspond souvent à un des objectifs du problème. Ainsi, la meilleure solution, appelée aussi solution optimale, est la solution ayant obtenu la meilleure évaluation par rapport à l'objectif défini. Les problèmes d'optimisation sont utilisés pour modéliser de nombreux problèmes dans différents secteurs de l'industrie: mécanique, chimie, télécommunications, environnement, transport et autres. Lorsqu'un seul objectif est spécifié, par exemple un objectif de minimisation de la distance totale, la solution optimale est clairement définie, c'est celle qui a la plus petite distance. Cependant, comme on l'a déjà vu dans de nombreuses situations, il y a souvent plusieurs objectifs contradictoires à satisfaire simultanément. En fait, les problèmes d'optimisation rencontrés en pratique sont rarement uni-objectifs. Pour ce genre de problèmes, le concept de solution optimale devient alors plus difficile à définir. Dans ce cas, la solution recherchée n'est plus un point unique mais un ensemble de solutions dites de compromis encore appelé ensemble de Pareto. Résoudre un problème comprenant plusieurs objectifs, appelé communément problème multi-objectifs, consiste donc à déterminer le meilleur ensemble de solutions de compromis.

Qu'ils comportent un seul ou plusieurs objectifs, les problèmes d'optimisation sont en général difficiles à résoudre. Nombre d'entre eux sont dits NP-Difficiles et ne peuvent être résolus de façon optimale par des algorithmes exacts. La nécessité de trouver rapidement des solutions acceptables à plusieurs de ces problèmes a entraîné le développement de plusieurs techniques d'approximation dont font partie les systèmes multi-agents (SMA) que nous développerons au cours du prochain chapitre.

2. Définitions :

Selon **Power [1999]**, le concept de SIAD a évolué à partir de deux domaines de recherche : les études sur la prise de décision dans les organisations du "*Carnegie Institute of Technology*" à la fin des années 1950 et les travaux sur les systèmes informatiques interactifs du "*Massachusetts Institute of Technology*" dans les années 1960. Le concept de SIAD est devenu un domaine de recherche en soit dans le milieu des années 1970, avant de prendre plus d'ampleur au cours des années 1980 [**Hättenschwiler, 1999**]. Il apparaît clairement que les SIAD reposent sur des bases multidisciplinaires, incluant notamment l'informatique, la recherche opérationnelle, l'intelligence artificielle, l'ingénierie logicielle, l'interaction homme-machine et de plus en plus les télécommunications [**Gachet, 2001**].

Le concept de SIAD est extrêmement vaste et sa définition varie selon le point de vue de l'auteur [**Druzdel et Flynn, 1999**]. De nombreuses définitions des SIAD ont été proposées comme celle de Peaucelle [**Davis et al., 1986**]: « *L'expression systèmes interactifs d'aide à la décision désigne les systèmes qui servent dans le processus de la prise de décision. Ces systèmes aident mais ne remplacent pas le décideur. Dans cette perspective, l'aide généralement automatisée permet au décideur d'avoir accès aux données et de tester*

différents choix possibles pour la résolution du problème à traiter L'efficacité du processus de résolution du problème décisionnel est amplifiée par l'interaction entre l'humain et la machine, chacun d'eux étant utilisé dans leurs champs distinctifs de compétence ».

Finlay [1994] définit simplement les SIAD comme «des systèmes informatiques supportant la prise de décision». De son côté, **Probst [1984]** pense qu'un SIAD est : « *Un logiciel conçu pour faciliter la préparation d'informations pertinentes sur la base desquelles une décision motivée peut être prise... donc un ensemble de moyens informatiques organisés pour améliorer le processus décisionnel. Ces systèmes fourniraient un cadre normatif à la démarche devant aboutir à une prise de décision ».*

Turban [1993] définit pour sa part un SIAD comme: « *un système d'information interactif, flexible, adaptable et spécifiquement développé pour aider la résolution d'un problème de décision en améliorant la prise de décision. Il utilise des données, fournit une interface utilisateur simple et autorise l'utilisateur à développer ses propres idées ou points de vue ».*

Finalement, **Schroff [1998]** et **Keen [1981]** estiment qu'il est impossible de donner une définition précise qui inclurait toutes les facettes d'un SIAD. S'il n'existe pas une définition universelle des SIAD, les différents auteurs s'accordent sur les différentes caractéristiques ou fonctionnalités recherchées dans ce type de système. Parmi celles-ci, mentionnons les suivantes [**Garlatti, 1996**] :

- a. ils doivent apporter principalement une aide pour les problèmes peu ou mal structurés en connectant ensemble des jugements humains et des informations calculées;
- b. ils doivent posséder une interface simple et conviviale afin d'éviter que l'utilisateur ne soit perdu devant la complexité du système;
- c. ils doivent fournir une aide pour différentes catégories d'utilisateurs ou différents groupes d'utilisateurs;
- d. ils doivent supporter des processus interdépendants ou séquentiels;
- e. ils doivent être adaptatifs dans le temps. Le décideur peut être réactif, être capable de confronter des conditions changeant rapidement et d'adapter le SIAD pour faire face aux nouvelles conditions. Un SIAD doit être suffisamment flexible pour que le décideur puisse ajouter, détruire, combiner, changer et réarranger les variables du processus de décision ainsi que les différents calculs fournissant ainsi une réponse rapide à des situations inattendues;
- f. ils doivent laisser le contrôle de toutes les étapes du processus de décision au décideur pour que celui-ci puisse remettre en cause à tout moment les recommandations faites par le SIAD. Un SIAD doit aider le décideur et non se substituer à lui;
- g. ils doivent utiliser des modèles. La modélisation permet d'expérimenter différentes stratégies sous différentes conditions. Ces expériences peuvent apporter de nouvelles vues sur le problème et un apprentissage;
- h. les SIAD les plus avancés utilisent un système à base de connaissances qui apporte notamment une aide efficace et effective dans des problèmes nécessitant une expertise;
- i. ils doivent permettre la recherche heuristique ; et
- j. ils ne doivent pas être des outils de type *boite noire*. Le fonctionnement d'un SIAD doit être fait de manière à ce que le décideur le comprenne et l'accepte.

Pour satisfaire tous ces critères, un SIAD se compose d'au moins trois composants : un module de dialogue ou interface homme-machine (IHM), un module contenant les données (base d'informations) et un module contenant les procédures de calcul ou modèles (base de modèle) [Sage, 1991]. À ces trois modules peut éventuellement s'ajouter une base de connaissances [Garlatti, 1996]. Comme l'indique la Figure 2, le module de dialogue est interconnecté avec les autres modules, il constitue l'interface entre l'utilisateur et le reste du système. Les modules représentés en pointillés constituent les modules optionnels d'un SIAD.

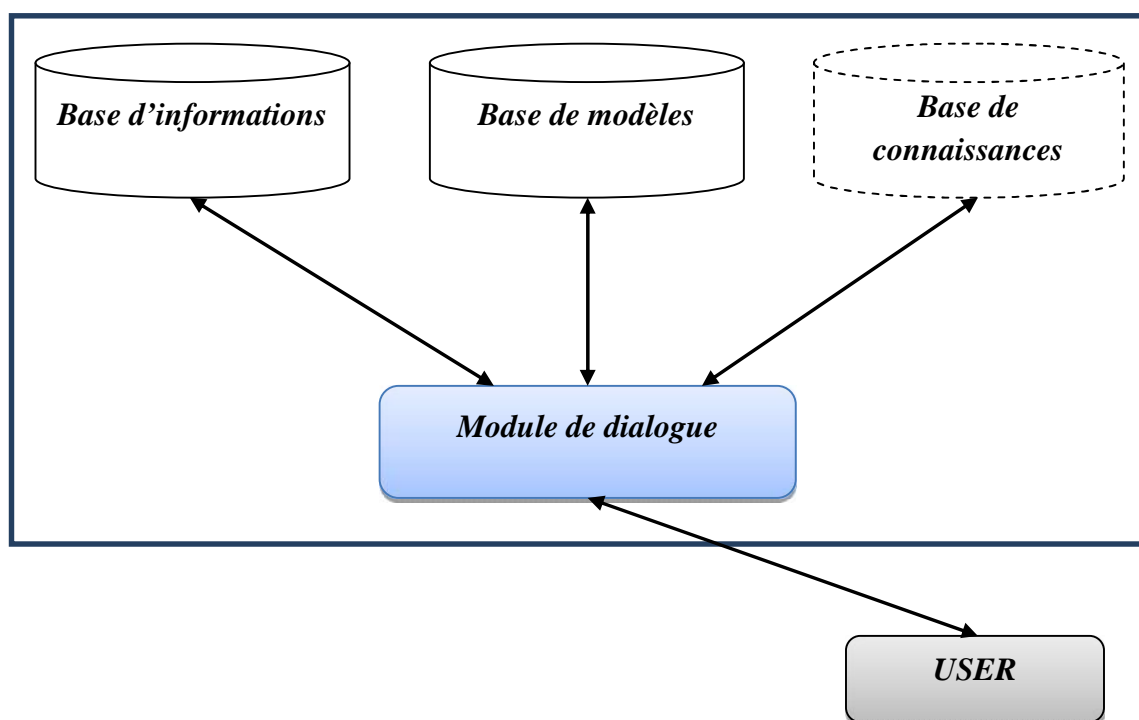


Figure 2 : Composition d'un SIAD

Le module de dialogue représente le module charnière du système permettant d'établir une collaboration entre le décideur et la machine. Pour Courbon et Stabell [1986], le module de dialogue est au centre du SIAD et sa réalisation est primordiale. En fait, une étude [Myers, 1995] a montré qu'au moins 50% du code des applications interactives correspond à l'IHM et 50 % du temps du développement est dépensé pour son implantation. C'est par l'intermédiaire des interfaces gérées par ce module que le décideur accède aux données et aux fonctions de calcul du SIAD. Une fois les manipulations demandées par le décideur effectuées, le système lui renvoie le résultat via les interfaces du module de dialogue. Les échanges sont d'autant plus favorisés que les représentations des résultats, tout comme le mode de questionnement du système, correspondent aux représentations mentales du décideur. Ainsi, le décideur peut exercer son contrôle et effectuer sa recherche heuristique dans de bonnes conditions. Un bon module de dialogue doit permettre d'afficher les informations sous différentes formes (graphiques 2D ou 3D, textes, vidéo ou autres). Il doit aussi fournir une aide à l'utilisateur pour que ce dernier mène à bien sa tâche et il doit le guider à l'aide d'exemples précis tout en étant suffisamment flexible pour s'adapter aux besoins des différents usagers. Un autre point à souligner est que l'interface homme-machine doit permettre le choix entre différents modes ou styles de fonctionnement [Lévine et Pomerol, 1990]. Parmi ces différents modes de

fonctionnement, **Courbon [1986]** distingue notamment le mode *assisté* (toutes les possibilités du dialogue sont disponibles), le mode *expert* (l'assistance est très limitée, ce mode s'applique dans le cas d'un décideur expérimenté), le mode *automatique* (le dialogue se déroule tout seul, suppression de la plus grande partie de l'interactivité), le mode *procédure* (des séquences entières sont exécutées à partir d'une instruction) et le mode *apprentissage* (l'accès au dialogue suit une progression pédagogique).

La base d'informations assure la fonction de mémoire, elle stocke non seulement les données, de façon permanente ou passagère, mais elle gère aussi l'enregistrement de données volatiles ainsi que l'effacement de ces mêmes données selon le souhait de l'utilisateur. Ces données volatiles correspondent aux résultats obtenus lors de traitements de données. Les données que nous avons qualifiées de permanentes sont les informations statistiques ou autres données qui décrivent les situations courantes et passées. Parmi ces données, il peut aussi y avoir des estimations concernant l'évolution de certains paramètres environnementaux.

La base de modèle se compose d'un ensemble de modèles et d'un système de gestion de ceux-ci. Les modèles peuvent être: des outils de recherche opérationnelle, des modèles statistiques ou autres. Pour avoir davantage de flexibilité, un SIAD doit posséder plusieurs modèles [**Chabbat, 1997**]. Dans cette optique, le SIAD organise les liens et le passage de paramètres entre les différents modèles, de même qu'il gère le module de dialogue [**Lévine et Pomerol, 1990**].

La base de connaissance regroupe pour sa part un ensemble de connaissances sur le domaine du problème, sur les modèles et sur les stratégies de constructions des modèles. Elle permet d'apporter une aide active à la résolution du problème de décision pendant toutes les phases du processus [**Klein, 1988**]. Elle introduit la notion d'apprentissage dans le SIAD. La base de connaissance peut aussi jouer dans certains cas le rôle de base de modèles [**Hansen et al., 1995**].

Un SIAD peut assister le décideur lors des trois premières étapes du processus décisionnel [**Davis et al, 1986**] décrit précédemment. À la phase d'information, le système permet l'exploration de l'environnement afin d'identifier les conditions et les situations exigeant une prise de décision. Dans la phase de conception, le système permet de préciser la situation avec les diverses hypothèses, de générer les solutions possibles et de tester leur faisabilité. Finalement, lors de la phase de choix, le SIAD peut suggérer certaines solutions au décideur parmi les différentes alternatives développées au cours de la phase précédente ; cependant, la décision finale revient toujours au décideur.

Jusqu'à maintenant, les SIAD ont été présentés pour être utilisés par un seul usager à la fois. Ils peuvent également être utilisés pour la prise de décision de groupe. En effet, de nombreuses études comme celle de **Keen [1981]**, **Liberatore et Titus [1983]** ou, plus récemment, **Totton et Flavin [1991]** montrent que la plupart des décisions dans une organisation sont prises par un groupe d'individus plutôt que par un décideur isolé. Plus une organisation devient complexe, moins les décisions sont prises par une seule personne [**Gannon, 1979**]. Les SIAD de groupe constituent une spécialisation des SIAD qui reflète cet aspect de la prise de décision.

3. Les différentes architectures des SIAD :

Dans la littérature, on trouve une multitude d'architectures pour les SIAD. Cela dit on peut distinguer trois grandes familles d'architecture :

- *Architecture en réseau ;*
- *Architecture centralisée ;*
- *Architecture hiérarchisée.*

a- Architecture en réseau :

Chaque modèle possède sa base de données, son modèle et son module de dialogue ainsi que des modules d'intégration.

Le principal avantage de cette architecture est une grande modularité mais la simplicité apparente se paye au niveau de l'intégration.

Mais c'est une architecture ouverte et adaptable. Les modifications à l'intérieur d'un complexe ne retentissent pas sur les autres, ce qui est un gros avantage.

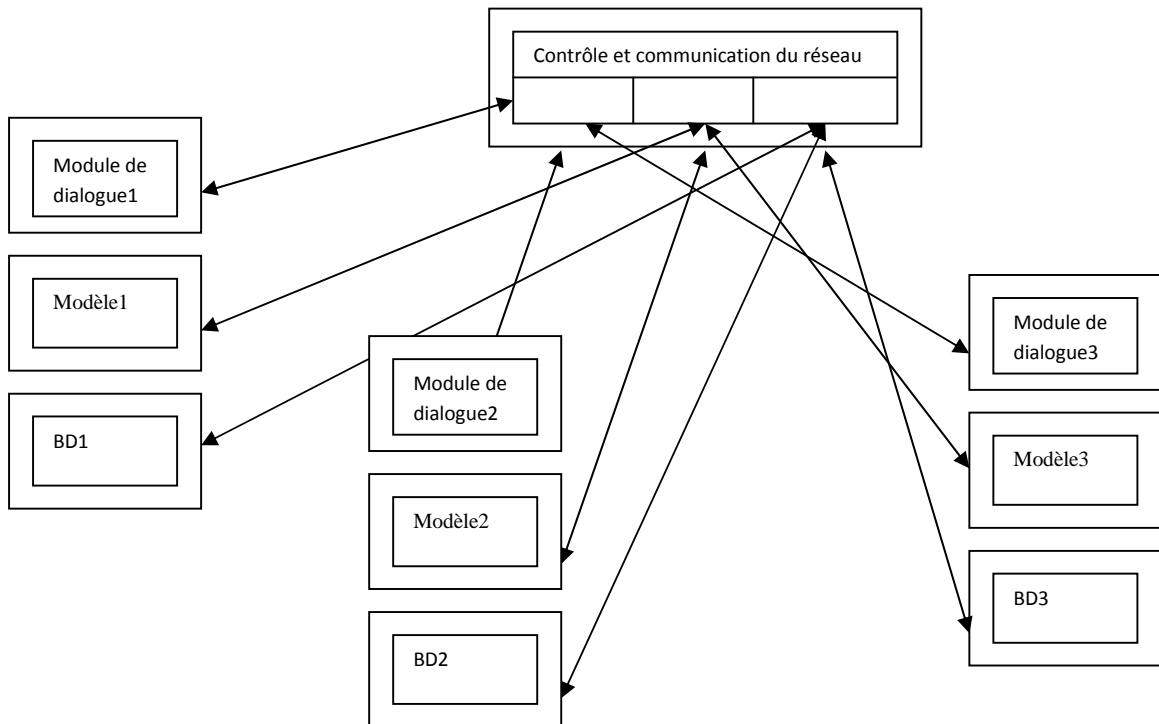


Figure 3: Architecture en réseau d'un S.I.A.D

b- Architecture centralisée :

Dans ce cas, chaque modèle relève d'un unique module de dialogue et communique avec une seule base de données.

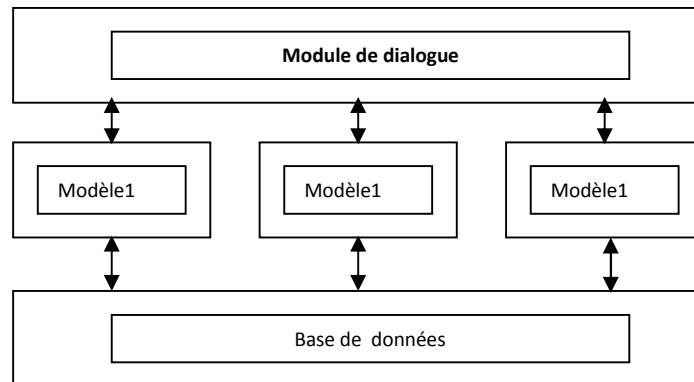


Figure 4 : Architecture centralisée d'un S.I.A.D

Avec ce type d'architectures, l'intégration des différents modèles est excellente. Le partage d'une base de données facilite les échanges d'informations entre les modèles. Le contrôle se fait par le dialogue.

Le principal inconvénient est le manque de souplesse. Il est difficile d'introduire un nouveau modèle.

c- Architecture hiérarchisée :

Elle se rapproche du système centralisé. La principale différence est que le module de dialogue est divisé tandis que le module « base de données » est muni d'une couche supplémentaire. Cette couche est destinée à remédier au principal défaut de l'architecture centralisée en permettant des adaptations plus faciles (ajout et suppression de modèles) ;

Le module de dialogue est divisé en deux parties : le module de dialogue commun qui gère la partie des échanges avec l'utilisateur, qui ne relève pas directement d'un modèle et des modules de dialogues qui relèvent des modèles. Le superviseur contrôle le dialogue utilisateur et les modèles.

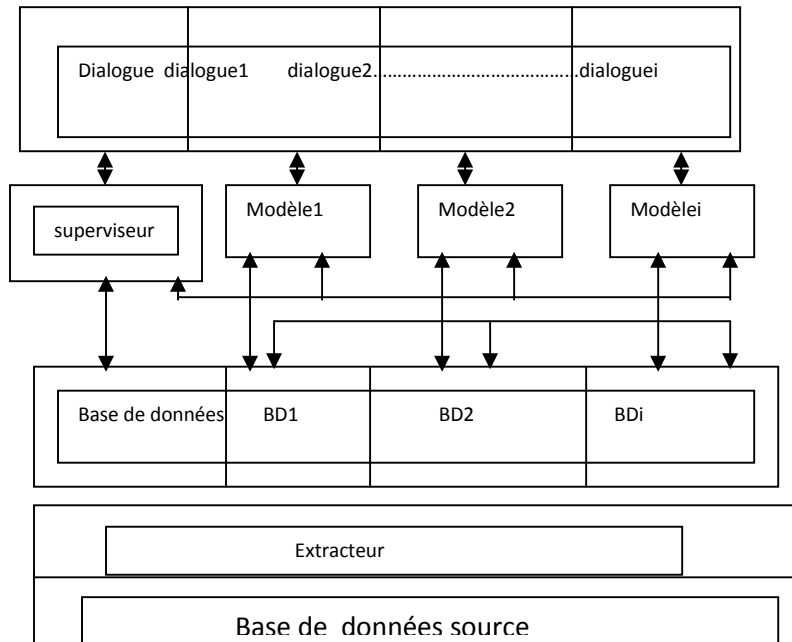


Figure 5: Architecture hiérarchisée d'un S.I.A.D

Après avoir présenté les différentes architectures, nous allons résumer les avantages et les inconvénients de chacune d'entre-elle dans le tableau qui suit :

En réseau	Avantages : - Architecture ouverte ; - Grande modularité.	Inconvénients : - Faible intégration ; - Manque d'unités de dialogue ; - Difficultés pour échanger les données ; - Difficulté de conception du contrôle.
Centralisée	Avantages : -Grande intégration -Unité des dialogues -Facilité d'échange des données ; -Relative facilité de réalisation.	Inconvénients : - Grande difficulté pour faire des modifications, en particulier pour l'introduction de nouveaux modèles ; - Manque de confidentialité dans l'accès aux données.
Hiérarchisée	Avantages : -Grande intégration -Unité des dialogues -Facilité de création des BD ; - Facilité d'emploi.	Inconvénients : - Difficulté de réalisation du superviseur et d'un extracteur suffisamment universels pour supporter les modifications et les ajouts ultérieurs.

Tab 1 : Les avantages et les inconvénients des architectures des SIAD.

4. Classification des SIAD :

On peut distinguer généralement quatre types de classification des SIAD :

- Au niveau utilisateur ;
- Au niveau conceptuel ;
- Au niveau système et opérationnel;
- Au niveau connaissances.

a- Classification au niveau utilisateur :

Keen [1987] propose un regroupement par niveau d'aide interactive c'est-à-dire d'après son niveau d'implication dans la prise de décision. Nous trouvons :

- **SIAD passif** : Il donne un avis mais reste moins puissant que celui de l'utilisateur ;
- **SIAD traditionnel**: Ce système est un assistant dont la principale utilisation concerne les interactions de type « what if ? » ;
- **SIAD étendu** : Il assure les fonctions de consultant et se place au même niveau que l'utilisateur ;
- **SIAD nominatif** : Il domine le processus de décision, l'utilisateur ne fait que le rôle d'aide à la décision. Cette typologie spécifie la manière et le niveau de coopération Homme machine ;

b- Classification au niveau conceptuel :

Selon les auteurs, il existe plusieurs classifications des SIAD. On peut citer celle de [Hackathorn et Keen, 1981] qui distingue trois catégories de systèmes de ce type, mais qui sont en corrélation :

- aide personnelle ;
- aide de groupe ;
- aide d'organisation.

Pour leur part, Holsapple et Whinston [1996] classifient les SIAD en six catégories :

- les SIAD orientés textes (Text oriented) ;
- les SIAD orientés bases de données (Database-oriented DSS) ;
- les SIAD orientés Bilan (Spread sheet-oriented DSS) ;
- les SAD orientés résolution (Solver-oriented DSS) ;
- les SIAD orientés Règle (Rule-oriented DSS) ;
- les SIAD composés (Compound DSS).

Utilisant le mode d'assistance comme critère, Power [2002] différencie entre :

- les SIAD dirigés par les modèles (*model-driven DSS*) soulignant l'accès et la manipulation d'un modèle statistique, financier, d'optimisation, ou de simulation. Ce système utilise des données et des paramètres fournis par des utilisateurs pour aider les décideurs dans l'analyse d'une situation ;

- les SIAD dirigés par la communication (*communication-driven DSS*) supportant plus d'une seule personne travaillant sur une tâche partagée ;

- les SIAD dirigés par les données (dits aussi orientés données) (*data-driven DSS*) soulignant l'accès et la manipulation d'une série chronologique de données internes d'entreprise et, parfois, de données externes ;

- les SIAD dirigés par les documents (*document-driven DSS*) qui contrôlent, recherchent, et manipulent l'information non structurée dans une variété de formats électroniques ;

- les SIAD dirigés par la connaissance (*knowledge-driven DSS*) fournissant l'expertise de résolution des problèmes, stockée comme faits, règles, procédures, ou en structures semblables. Un concept lié est l'Extraction de Connaissances à partir de Données (ECD).

c- Classification au niveau système et opérationnel:

Power [1997] distingue entre des SIAD d'entreprise et ceux de bureau :

- Un SIAD d'entreprise : sert à plusieurs dirigeants de l'entreprise, il est muni de bases de données contenant une masse de données très importante.

- Un SIAD simple utilisateur ou de bureau : correspond à un logiciel servant d'aide à un seul dirigeant, il réside uniquement dans un PC.

Depuis les travaux de Scott Morton [Morton, 1971], le domaine des SAD n'a cessé d'évoluer. Le concept de système d'aide à la décision est extrêmement vaste et ses définitions dépendent du point de vue de chaque auteur. Un tel système peut prendre de nombreuses formes et peut s'utiliser de diverses manières. D'une manière générale, on peut le définir comme étant "un système informatique qui facilite le processus de prise de décision".

On peut opérer différentes classifications des systèmes d'aide à la décision. Mais en axant sur l'aspect opérationnel de tels systèmes, on peut adopter la typologie proposée par **Turban [1993]** qui en distingue trois catégories :

- **Les Executive information system (EIS)** : ce sont des outils fournissant au décideur l'information utile. Ces applications ne permettent que de recenser les informations pertinentes par l'amélioration des accès et des manipulations des informations existantes pour confronter objectifs et observations [**Watson et al., 1993**] [**Turban et al., 1995**] [**Arnott et al., 2007**].

- **Les Executive Support System (ESS)** : l'ESS va au-delà de l'EIS car il inclut des communications, des outils d'analyse et de l'intelligence. L'ESS doit permettre l'analyse des données et permettre d'avoir une appréciation du futur de l'organisation. Ainsi pour passer de l'EIS à l'ESS ou de l'information à l'intelligence, il faut pouvoir affecter le futur (axe temporel et visualisation des liens) [**Kuo, 1998**].

- **Les Decision support System (DSS)** : un DSS (SIAD : Système Interactif d'Aide à la Décision) est un système informatique qui aide les décideurs de façon interactive à utiliser des données et des modèles pour résoudre des problèmes de décision. Un SIAD doit offrir un langage sophistiqué et des possibilités d'analyser les effets d'éventuels changements de l'environnement sur l'organisation [**Turban et al., 1998 ; Dargam et al., 2006**].

Les deux premières catégories privilégient surtout l'étape d'Intelligence. Celle-ci est formée des activités d'exploration nécessaires pour identifier les situations où il existe un problème, ou encore les opportunités de décider et d'agir. Précisément, il permet :

- La compréhension du problème : en donnant un modèle précis de la situation avec les diverses hypothèses ;

- La génération des solutions possibles à partir de la manipulation des modèles implémentés, en procurant des éclaircissements et des idées sur les solutions possibles du problème ;

- La faisabilité des solutions : elles sont analysées en termes d'impact sur l'environnement.

Les SIAD sont des systèmes mettant en œuvre l'importance de l'Interaction Homme-Machine (IHM) dans la prise de décision. Les SIAD devraient être adaptés aux styles cognitifs des décideurs [**Lévine et al., 1989 ; Power, 2002**].

- **PSS (Planning Support System)** : ils permettent une analyse de faisabilité des procédures ou des décisions retenues (c'est-à-dire fournir aux décideurs une assistance intelligente).

4. Classification au niveau connaissance :

Les principaux SIAD appartenant à cette classification sont :

- Les systèmes à base de savoir théorique ;
- Les systèmes à base de savoir experts ;
- Les systèmes à base de savoir factuel ;
- Les systèmes informatiques classiques.

5. Exemples de domaine d'application des SIAD :

Depuis une dizaine d'années, de nombreux SIAD ont été développés pour des secteurs aussi variés que les télécommunications, le transport aérien et ferroviaire, la santé, l'ordonnancement et la gestion de projet. En fait, si l'on souhaitait résumer l'ensemble des

domaines d'application des SIAD, on pourrait dire que les SIAD peuvent être mis en place au niveau de chaque activité humaine, nécessitant un processus de décision élaboré. Dans cette section, nous présentons quatre exemples de SIAD fonctionnant au quotidien dans des entreprises.

a- Un système d'allocation de wagon :

À la SNCF (société nationale de chemin de fer français), la répartition des wagons de marchandises est effectuée, sur le plan national, par un service « central de répartition » [Lévine et Pomerol, 1990]. Des experts, au sein de ce service, appelés répartiteurs, reçoivent par téléphone les demandes de wagons vides en provenance des zones. Ils centralisent alors les informations sur les wagons vides disponibles et assurent la répartition entre les zones déficitaires et les zones excédentaires. Autrement dit, ils prennent des wagons dans les zones excédentaires pour les envoyer vers les zones utilisatrices.

Outre les demandes ou les offres téléphoniques, les répartiteurs disposent des données de GCTM (Gestion Centralisée du Transport Marchandise) qui fournissent un état de la situation à trois heures du matin, état qui sort sur l'imprimante vers huit heures du matin à l'arrivée des répartiteurs. Cette situation donne un état des besoins et des wagons présents dans les zones ainsi que la moyenne des besoins et des ressources des jours précédents.

Le SIAD mis en place dans ce cadre a pour objectif d'aider le répartiteur dans sa tâche. Il est basé sur l'utilisation d'un système à base de connaissances. La minimisation des coûts de déplacement à vide, le respect des délais et la satisfaction des clients sont les critères principaux qui régissent le travail des répartiteurs et qui doivent se retrouver dans le SIAD.

Il s'agit là d'un premier exemple de SIAD qui peut être généralisé à l'ensemble des problèmes d'allocation de ressources en extrayant les caractéristiques propres à chaque problème.

b- La gestion de la production :

Un autre type de SIAD pouvant être utile dans l'industrie concerne les systèmes de gestion de production. Un projet de ce type a été mené par le Carnegie-Mellon Robotics Institute sous la direction de Fox [1981]. Dans ce genre d'application, il s'agit d'optimiser la production en influant sur certains critères : changer les techniques de conception, changer le matériel, anticiper les pannes en proposant des solutions de remplacement et synchroniser les différentes étapes de production. Le système mis en place permet de faire un suivi de la production. Il permet aussi d'effectuer des simulations par rapport à certaines modifications de paramètres et ainsi de trouver des solutions de rechange en cas de problèmes.

c- Un système de surveillance pour les sites industriels à hauts risques :

Les SIAD peuvent aussi être utiles dans la gestion des risques industriels. La présence de sites industriels à très hauts risques technologiques (centrale nucléaire, centrale thermique et autres) en milieu urbain représente, comme leur nom l'indique, un risque non négligeable pour la population [Boukachour et al., 2000]. Une solution serait l'installation d'un réseau de sirènes qui doit permettre le confinement des populations. Il s'agit pour un opérateur préposé à la détection des risques majeurs de déclencher ces sirènes au moment jugé opportun. Pour prendre sa décision, il doit être en mesure d'avoir une vision globale de la situation et des événements survenus ainsi que de leurs conséquences (réactions en chaîne). La connaissance concernant l'évolution d'une situation est dispersée auprès de différents experts. Pour avoir une évaluation de la situation, on va devoir interroger des experts en chimie, en météorologie, en traitement des situations d'urgence (gestion des populations, traitement des blessés par les

services concernés, traitement des accidents routiers par les pompiers, et autres). L'ensemble des informations à traiter et à synthétiser par l'opérateur, qui prendra la décision finale de déclencher ou non les sirènes, est vaste. Le SIAD devra donc permettre de mettre en évidence l'information pertinente en ayant recours à cette multi-expertise.

d- Pioneer Natural Resources :

Dans l'industrie du pétrole et du gaz, un grand nombre de variables sont liées à l'exploitation d'une société d'énergie, notamment les coûts du développement et de la production. A cause de la relation complexe entre toutes ces variables, les gestionnaires ont du mal à déterminer la rentabilité de leurs décisions. Pioneer Natural Resources (PNR) de Las Colina dans l'état du Texas aux États-Unis a décidé de créer un SIAD qui pourrait fournir des informations plus précises à cet égard.

En 1995, les cadres de cette entreprise ont commencé à repérer toutes les variables de gestion et ont tracé des diagrammes de tous les processus de l'entreprise pour créer un modèle qui pourrait montrer les conséquences du changement d'une ou de plusieurs de ces variables. L'entreprise a ensuite construit un prototype de SIAD. Les cadres ont d'abord testé le système en simulant la situation instable de PNR sur la côte du Golfe du Mexique, dont le temps de production était toujours long. Devant l'efficacité du système, les responsables de la société estiment que chacune des cinq divisions de la compagnie pourrait accroître ses revenus de 25 à 40 % en utilisant un outil similaire.

IV- Les systèmes interactifs d'aide à la décision à base de web (web-based DSS) :

Jusqu'à présent, les SIAD que nous avons vus avaient une dimension géographique qui se limite à la dimension du site dans lequel ils sont implantés. Le réseau du web fait que ces limitations géographiques peuvent être largement dépassées du moment qu'on utilise les technologies du web pour concevoir et implémenté les SIAD. A ce moment-là, il serait plus juste de définir ces derniers comme étant un *SIAD à base de Web*.

Les technologies web peuvent être utilisées pour mettre en place tout type de SIAD précédemment définis. Dans le passé, la plupart des systèmes étiquetés "SIAD à base de Web" étaient reliés à un entrepôt de données local, mais aujourd'hui ce n'est certainement plus le cas. En effet, du moment qu'il existe des entrepôts de données un peu-partout dans le monde, il n'est plus nécessaire d'avoir son entrepôt de données, il suffit juste d'avoir accès à ceux qui existent déjà. Avec un SIAD à base de Web, aucun composant de l'application du système d'aide à la décision ne doit figurer sur l'ordinateur de client. Un navigateur Web et une connexion à Internet offrent les fonctionnalités d'aide à la décision à l'utilisateur. **Kuljis et Paul [2001]** passent en revue sur le Web des études de simulations et de développements.

1. Définitions d'un SIAD à base de Web (Web-based DSS) :

Les technologies Web sont utilisées pour améliorer la capacité de système d'aide à la décision par des modèles de décision, le traitement d'analyse en ligne (OLAP) et les outils d'extraction de données qui permettent la publication normalisée et le partage des ressources de décision sur internet. Dans un système d'aide à la décision à base de Web, tous les processus décisionnels impliquant des opérations sont exécutés sur un serveur réseau d'ordre n afin de bénéficier la distribution de la prise de décision et l'indépendance de la plate-forme,

des périodes d'apprentissage plus courtes pour des utilisateurs déjà familiers avec les outils web et la navigation web, de la diffusion des logiciels à bas prix, la mise à jour des logiciels et la réutilisabilité de modules de décision et d'information sur internet grâce à une panoplie de protocoles normalisés [Huaiqing et al., 2002].

Selon **Friedman [2005]**, l'importance d'utiliser les SIAD à base de Web résulte de l'utilisation croissante de l'information disponible qui devrait être identifiée, contrôlée et accédée à distance, en utilisant les outils à base de Web pour soutenir la réutilisabilité des modules de décision intégrés. De tels systèmes aident les entreprises globales à gérer et améliorer les processus de décision, le contrôle des processus, le service client, et rendre la nouvelle conception plus flexible.

En utilisant les SIAD à base de Web, les décideurs peuvent partager des modules de prise de décision ouverts sur internet, en utilisant des protocoles normalisés tels que le HTTP, en un format normalisé comme XML ou DAML.

D'après **Shaw et al. [2002]**, les systèmes à base de Web sont considérés comme des "plates-formes de choix" pour fournir l'aide à la décision tout en tenant compte de nombreuses considérations techniques, économiques et sociales. La migration vers les SIAD à base de Web, dénote un changement des générateurs des SIAD (ce qui permet aux utilisateurs de développer des applications spécifiques caractérisées par un déploiement limité, une rigidité à intégrer des orientations au travers des applications qui soulignent la réutilisation des applications et des composants. En déployant les capacités de Web, les bases de connaissances multiples et les techniques de traitement de la connaissance peuvent être utilisées. La conception des systèmes interactifs d'aide à la décision a été affectée par la disponibilité d'une large gamme d'outils du web, des techniques et des technologies. L'utilisation des outils de Web a permis de remodeler la description des relations entre les composants de l'information et les modules de décision de telle façon que, la conception physique et la logique du SIAD se trouvent affectées. En conséquence, l'architecture fondamentale du SIAD à base de Web est passée des architectures centralisées, aux systèmes client-serveur, au web et aux systèmes répartis basés sur la technologie de réseau, qui permettent l'intégration de grandes quantités de données et des outils d'aide à la décision provenant des sources hétérogènes pour la disposition (fourniture) d'informations à valeur ajoutée, en utilisant la découverte de connaissance et les outils d'extraction de données.

Les systèmes d'aide à la décision à base de web sont utilisés par des organisations comme outils aides à la décision pour les employés aussi bien que les clients. Il s'agit d'utiliser les SIAD à base de Web comme assistants pour aider les clients à configurer le produit et le service selon leurs besoins. Ce sont des SIAD grand public.

Ces systèmes permettent aux différents clients de concevoir leurs propres produits par le choix d'un menu d'attributs, des composants, des prix et des options de livraison. Par exemple, sur les sites web de la plupart des producteurs des ordinateurs de bureau (www.dell.com, www.compaq.com, and www.ibm.com), les individus peuvent commencer par une configuration de base définie par un modèle et une vitesse de processeur, et continuer ensuite à spécifier la configuration complète avec leur choix de taille de disque dur, mémoire et accessoires comme les CD-ROMs, moniteurs et imprimantes. Des cas semblables peuvent être trouvés dans : (1) l'industrie d'habillement (www.landsend.com, www.blair.com, and www.weddingchannel.com), qui permet à un utilisateur un modèle virtuel pour concevoir son costume, (2) dans l'industrie de finances (www.calvertgroup.com), qui permet à l'utilisateur

d'essayer la retraite différente sauvant (économisant) des plans, et (3) dans l'industrie de jouets (www.vermontteddybear.com), où les enfants peuvent faire sur mesure les ours en peluche, ils décident de la couleur, la taille et le type du manteau.

2. Les travaux de recherche récents dans le SIAD à base de Web (Web-based DSS) :

Cette section présente un état de l'art sur les SIAD à base de Web (Web-based DSS), organisée autour de deux principaux points que nous allons développer un peu plus loin, à savoir: *architectures et technologies* et *applications et réalisations*. Un certain nombre d'articles ont passé en revue des matières plus spécifiques liées au SIAD à base de Web. Par exemple, **Kuljis et Paul [2001]** ont passé en revue la simulation à base de Web et le **Kersten et le Noronba [1999]** ont décrits l'aide à la négociation à base de Web.

a. Architectures et technologies :

Un certain nombre d'articles discutent les issues architecturales, les cadres, la rentabilité, et d'autres matières de technologie qui sont généralement applicables au SIAD à base de Web.

Bharati et Chaudhury [2004] ont entrepris une étude empirique pour étudier la satisfaction de clients avec des systèmes interactifs d'aide à la décision à base de Web. Dans leur modèle conceptuel, ils ont proposé trois variables indépendantes (qualité de système, qualité de l'information, et présentation de l'information) qui ont influencé la satisfaction des utilisateurs (variable dépendante). Ils ont montré que la qualité de système et la qualité de l'information sont fortement corrélées avec la satisfaction des utilisateurs, mais la présentation de l'information n'a pas d'impact significatif sur la satisfaction des utilisateurs.

Iyer et al. [2005] se sont intéressés à la gestion du modèle pour l'aide à la décision dans un environnement de calcul où les données et les modèles d'entreprise sont distribués. Ils ont défini des couches de connaissances pour la gestion du modèle et ont proposé une architecture préliminaire pour un environnement virtuel d'affaires (VBE).

Guntzer et al. [2007] ont proposé des modèles de service structurés qui emploient une variante de modèles structurés. Cette approche proposée peut aider les utilisateurs à trouver des ressources d'information disponibles comme un service en ligne dans un Intranet.

Zhang et Goddard [2007] ont appliqué des architectures de logiciel à la conception des SIAD à base de Web. Une architecture multicouches de logiciel a été proposée pour aider les développeurs à concevoir et mettre en application un SIAD dans un environnement distribué ; une telle architecture peut fournir une vue formelle et hiérarchique du SIAD qui aide la conception d'un SIAD à base de Web. En outre, les auteurs ont présenté un cadre à base de composant, 3CoFramework, pour aider l'exécution du SIAD.

Mitra et Valente [2007] ont fourni un aperçu sur l'optimisation à base de Web pour les SIAD guidés par les modèles. Leur étude a porté sur deux paradigmes (ASP et e-Services), et ont développé un fournisseur de service d'optimisation (OSP) " et le "WEBPOT " et ont trouvé de nouveaux avantages permis par l'implémentation du e-Service. Certains de ces avantages incluent la découverte de service, la gestion de service et la gestion de qualité.

Pour récapituler, ces études discutent les métadonnées, la gestion du modèle et la rentabilité pour le SIAD à base de Web. En outre, **Mitra et Valente [2007]** ont démontré que le SIAD à base de Web peut efficacement incorporer des modèles et des outils d'optimisation. Ces efforts d'aborder les questions fondamentales dans le calcul à base de Web sont les pierres angulaires essentielles pour l'usage pratique de l'aide à la décision à base de Web. La

recherche précédente indique que (a) les utilisateurs de Web ont besoin de détailler des informations sur le SIAD pour organiser et comprendre le contenu disponible, (b) les systèmes devraient être conçus pour inclure les constructions et les objets façonnés qui aident la livraison d'informations de haute qualité, et (c) les nouvelles approches pour la gestion du modèle sont nécessaires pour faciliter le stockage, la recherche, la récupération, l'assortiment, et la composition d'une bibliothèque des modèles de décision. Les avancées de recherches dans ces secteurs seront critiques au développement des orientations et des possibilités pour établir un SIAD à base de Web efficace.

b. Applications et réalisations :

Beaucoup de chercheurs et fournisseurs se sont intéressés à des études de cas basées sur les SIAD à base de Web et le développement des applications prototypes. **Kohli et al. [2001]** ont rapporté une étude de cas d'un SIAD à base de Web pour la gestion d'hôpital appelé système de Physician Profiling (PPS). Le PPS est un système de gestion de rapport de client (CRM) pour la gestion de rapport de médecin. Les auteurs ont démontré le développement du PPS et ont présenté une analyse coûts-avantages du projet.

Ngai et Wat [2005] ont développé et mis en application un SIAD à base de Web qui a employé un modèle basé sur la théorie des ensembles brouillées pour procéder à l'analyse de risque pour le développement du e-commerce. Les chefs de projet aidés par le prototype identifient, analysent, et donnent la priorité au risque impliqué dans un développement d'e-commerce. Dans leur prototype, le web browser sert de composant d'interface utilisateur et l'ensemble des modèles et des bases de données ont été hébergés par le serveur.

Dong et al. [2004] ont développé un cadre de SIAD à base de Web pour le choix de brochure. Ils ont employé le traitement analytique en ligne (OLAP) et une machine virtuelle parallèle (PVM) pour améliorer l'exécution globale de leur prototype du système de choix de brochure à base de Web (WPSS). Puisque le web browser est le composant d'interface utilisateur et que le serveur contient tous les modèles et des bases de données, le prototype permet aux ordinateurs multiples de manipuler un problème à grande échelle d'aide à la décision efficacement.

Sundarraaj [2004] a identifié les questions clés dans les contrats d'entretien de gestion et a développé un prototype qui peut soutenir le procédé de la planification d'un directeur.

L'avantage le plus important du prototype est l'étalonnage de la gestion des contrats d'entretien parce que le système à base de Web permet à beaucoup de directeurs d'accéder et d'employer un système simple.

Ray [2007] a rapporté une étude de cas qui démontre l'exécution des technologies d'aide à la décision à base de Web. Il a discuté une application spécifique développée pour le département du Delaware du transport (De IDOT). Le système a utilisé un réseau à base d'outils d'optimisation et des systèmes interactifs d'aide à la décision spatiaux pour améliorer la gestion des mouvements de véhicules à poids excessifs.

Delen et al. [2007] ont développé un SIAD à base de Web, appelé le gourou de MovieForecast, pour aider des décideurs dans l'industrie du cinéma. Pour prévoir des reçus de caisse, le système proposé adopte un méta-modèle de fusion de l'information qui emploie le rendement d'un réseau neurologique, des arbres de décision, d'une régression logistique ordinale, et d'une analyse discriminante. Leur évaluation exploratoire du système a indiqué que les utilisateurs ont été satisfaits de sa rentabilité en termes de plénitude de qualité de

l'information, de qualité de système, et d'utilisation (Cf, [Davis, 1989] [DeLone et McLean, 1992]).

V- Barrières au succès des SIAD :

Bien que les SIAD aient été le sujet de nombreuses recherches depuis plus d'une vingtaine d'années, ils n'ont pas nécessairement connu beaucoup de succès, en particulier du point de vu des utilisateurs [Rudnicka et Madey, 2001]. Le domaine des SIAD est trop vaste pour tenter de dresser une liste exhaustive des raisons qui font que ce type de systèmes suscite moins d'intérêt en pratique. Cependant, nous pouvons diviser les différents facteurs d'échec des SIAD en trois grandes catégories [Gachet, 2001]: *les facteurs humains, les facteurs conceptuels et les facteurs techniques*.

En ce qui concerne les facteurs humains, **Ghasemzadeh et Archer [2000]** font remarquer qu'en général le décideur n'est pas assez impliqué dans le processus de prise de décision avec un SIAD. Ce manque d'implication risque de provoquer une certaine réticence et un manque de confiance du décideur vis-à-vis du système. Un autre facteur responsable du manque d'intérêt des usagers par rapport aux SIAD traditionnels peut être expliqué par l'intérêt croissant pour «l'informatique de l'utilisateur final» [Kreie *et al*, 2000]. Ce concept fait référence aux personnes qui développent des logiciels pour leur usage personnel ou pour d'autres sans être des spécialistes de l'informatique. Le développement de cette tendance s'explique d'une part, par le coût du matériel informatique qui est devenu de plus en plus bas tout en étant de plus en plus performant, et d'autre part, par les environnements de programmation qui sont de plus en plus intuitifs [Gachet, 2001]. Même si ce type de systèmes est généralement de qualité inférieure aux SIAD traditionnels [Kreie *et al*, 2000], les utilisateurs préfèrent cette manière de travailler qui augmente leur satisfaction et élimine les problèmes de communication avec un spécialiste de l'informatique. Finalement, les SIAD sont généralement des systèmes complexes et difficiles à utiliser pour des non-spécialistes de l'informatique [Gachet, 2001]. Or, un bon SIAD doit être compréhensible et facile d'utilisation pour les décideurs [Sprague et Watson, 1993], sinon ils auront tendance à garder leurs distances vis-à-vis de tels systèmes.

Les facteurs conceptuels font référence aux problèmes rencontrés par les SIAD à cause d'un mauvais choix lors de leur design. Les SIAD sont utiles pour rechercher de l'information, l'évaluation et le choix d'une décision, mais ils sont moins utiles pour des tâches comme l'exploration d'un problème [Huber, 1982]. Une des tâches principales d'un SIAD consiste à modéliser l'environnement ou le contexte du problème à résoudre [Gachet,2001]. Cependant, **Hättenschwiler [1993]** constate plusieurs lacunes relatives à cette modélisation dans la plupart des SIAD :

- un manque de standards et de concepts pour la modélisation;
- un manque de supports pour la réutilisation des modèles existants; et
- un manque de convivialité permettant la modélisation.

En dépit des progrès technologiques considérables de ces dernières années, le développement et l'exploitation des SIAD continuent de connaître des problèmes tant au niveau matériel que logiciel. Le temps de développement inhérent à la construction d'un SIAD est souvent trop long et sa complexité non négligeable [Gachet, 2001]. Construire un SIAD requiert de l'expertise dans des domaines aussi variés que le design d'interface, la programmation et l'analyse de décision ce qui, dans certains cas, peut être compliqué à réunir.

Hättenschwiler et al. [1998] attribuent le manque d'intérêt pour les SIAD aux techniques traditionnelles de conception qu'ils jugent trop rigides. Ceci entraîne alors des coûts de développement trop élevés, une architecture statique et un manque de réutilisabilité du système. Comme toutes applications interactives, ces systèmes présentent souvent des dépassements de budgets avec des coûts de maintenance très élevés [**Palanque et al., 1994**]. Certaines évaluations d'applications interactives [**Bohm 1976**] ont révélé que leur maintenance peut représenter jusqu'à 70 % du coût total de développement. Ce coût de maintenance est occasionné principalement par trois types de problèmes [**Farenc et al., 1994**]:

- la correction des erreurs apparues lors de l'utilisation de l'application ;
- l'addition de nouvelles fonctionnalités ou la modification de fonctionnalités existantes de l'application; et
- l'amélioration de l'IHM suite aux réticences exprimées par les utilisateurs ou suite aux difficultés qu'ils éprouvent lors de l'utilisation d'une ou de plusieurs fonctionnalités.

Construire une interface homme/machine pour un SIAD constitue un autre facteur technique sensible [**Gachet, 2001**]. Un système avec une interface usager surchargée, inadaptée ou pas assez claire sera généralement inutilisable en pratique [**Druzdzel et Flynn, 1999**].

Finalement, les SIAD sont des systèmes complexes souvent composés de plusieurs sous-systèmes hétérogènes (base de données, bibliothèques mathématiques, etc.) qu'il est difficile d'intégrer dans un seul système productif [**Gachet, 2001**].

VI- Conclusion :

Au cours de ce chapitre, nous avons présenté des notions qui se réfèrent à la prise de décision et aux systèmes interactifs d'aide à la décision. En résumé, nous avons vu qu'un SIAD peut être vu comme un système permettant à un décideur de prendre une décision en mettant à sa disposition la meilleure information possible. Il assiste le décideur sans jamais se substituer à lui.

Aussi, nous avons présenté une classe particulière des SIAD, à savoir les SIAD à base de web qui permettent de tirer profils et avantages des technologies du web pour augmenter le niveau d'aide aux décideurs, permettant de récolter, filtrer et organiser dans un océan de données, celles qui sont utiles à l'organisation et aux décideurs de celle-ci.

Par la suite, nous avons mentionné quelques exemples de SIAD réels et opérationnels au sein de systèmes de grande importance et quel est leur rôle au sein de ces entreprises.

Enfin, nous avons recensé les limites et barrières qui empêchent les SIAD de connaître le succès qu'ils auraient pu avoir dans le domaine de l'aide à la décision sous tous ses aspects.

CHAPITRE III

SYSTEMES MULTI-AGENTS

&

COORDINATION

PARTIE I :

**AGENTS ET SYSTEMES MULTI-
AGENTS.**

I- Introduction :

Un peu d'histoire...

En 1950, A.M. Turing se demande si « une machine peut penser » dans un article demeuré célèbre et dans lequel il invente le fameux test qui porte son nom et qui doit permettre de répondre à cette question qui, pour beaucoup est un sacrilège voire un blasphème pour quelques autres.

Si le « test de Turing » est contesté dès son invention par les psychologues, il pose les premières pierres de ce qu'on désignera dès 1956 sous l'expression osée *d'intelligence artificielle*. En quelques années seulement, ce champ de recherche progresse rapidement. D'abord, on a créé le premier système expert (DENDRAL) dès la fin des années 60, ensuite ce succès en appellera beaucoup d'autres dans la sphère technologique avec les outils de diagnostic et d'expertise utilisés aujourd'hui et de façon plus triviale, dans le domaine ludique des jeux d'échecs qui révèlent, sinon l'intelligence des processeurs, du moins leur formidable puissance de calcul et de rapidité. L'ancien champion du monde d'échecs se souvient encore de sa médiatique défaite face à l'ordinateur d'IBM « Deep Blue » en 1997... mais ce domaine a très vite connu ses limites, on a vu trop grand, trop rapidement.

D'ailleurs à cet effet, **Caldéroni [2002]** souligne que, malgré ces succès, « *l'Intelligence Artificielle classique n'apporte aucune solution à des problèmes extrêmement simples, à la portée d'un enfant de quelques mois seulement, comme par exemple reconnaître un visage humain en quelques millisecondes* ». Prolongeant son regard critique, il poursuit en considérant que « *de manière plus générale, l'Intelligence Artificielle a échoué dans ses tentatives de proposer des systèmes intelligents universels qui sachent s'adapter aux problèmes qu'ils rencontrent avant de les résoudre* ».

Cette notion d'adaptation aux problèmes nous semble être essentielle dans la mesure où elle renvoie en réalité, à la notion même de l'intelligence. Longtemps développée sur des ordinateurs monoprocesseurs, cette intelligence artificielle renvoyait une image trop simplifiée et centralisatrice de l'intelligence. Mais, une intelligence que l'on qualifierait de « biologique », semble renvoyer nécessairement à la notion d'interaction déjà largement discutée à l'époque. C'est en effet dans le contact avec l'Autre, en immersion totale dans un environnement inconnu, que l'intelligence se déploie ; les neurones développant des réseaux synaptiques au fil des interactions.

En effet, si on reprenant l'analyse de Caldéroni, « *on ne peut que constater que l'Intelligence Artificielle s'engageait dès le départ dans une impasse, négligeant la dimension sociale de l'intelligence* ».

Pour résoudre ce problème et prendre en charge l'aspect social de l'intelligence, un nouveau souffle intervint au début des années 70 avec le développement de l'Intelligence Artificielle Distribuée (IAD). Partant des limites avérées des approches classiques basées sur la centralisation et l'unicité du fonctionnement de l'intelligence, l'IAD a repensé la notion d'expertise à partir de la coopération de plusieurs entités informatiques, capables en quelque sorte d'associer leurs efforts pour accroître leur *intelligence collective*. Le concept de coopération est donc central mais en induit d'autres d'importance égale comme la coordination, la communication ou la négociation qui sont au principe même de la question première de l'interaction. L'Intelligence Artificielle Distribuée vise l'amélioration des performances des systèmes dit intelligents, par le développement de langages concurrents et

d'architecture parallèles ou distribuées. D'autre part, la résolution distribuée des problèmes s'intéresse non seulement à la manière de diviser un problème posé en un ensemble d'entités distribuées et coopérantes, mais aussi à celle de partager la connaissance du problème et d'en obtenir la résolution.

Dans ce contexte, les *Systèmes Multi-Agents (SMA)* constituent selon Caldéroni l'un des trois axes fondamentaux de l'IAD.

Dans les pages suivantes, nous allons définir et déterminer les concepts de bases et les particularités se rapportant aux SMA. Mais avant cela, nous devons remarquer que dans l'appellation "Système Multi-Agents", il y a un concept fondamental qui est celui d'"Agent". C'est quoi un agent ? Comment est-il ?, ce sont les questions que nous devons nous poser avant même d'essayer de comprendre ce que sont les SMA ; raison pour laquelle nous allons d'abord commencer par introduire et définir la notion d'agent.

II- Les notions relatives aux Agents :

1. Définitions :

a. Agent :

La notion d'agent peut prêter à confusion, car utilisée dans plusieurs domaines. En effet, comme le rappelle utilement FERBER [1995], cette notion, comme tous les concepts fondamentaux, semble pouvoir être comprise de manière évidente. Cependant, elle recouvre plusieurs acceptions, chacune renvoyant à un courant de recherche particulier, par exemple un *agent physique* peut être un ordinateur, un robot ou encore un humain, de même, un *agent formel ou abstrait* peut être vu comme un programme informatique ou un processus. Cependant, l'auteur a rédigé une vulgate en 1995 qui est reprise unanimement depuis, dans la bibliographie sous une forme approachante et que nous adopterons pour la suite de notre travail, un **agent** représente une entité *autonome* réelle ou abstraite qui, *située* dans un environnement (réel ou virtuel) et plongée dans une structure sociale est capable :

- 1- d'*agir* sur elle-même et sur son environnement ;
- 2- de *percevoir* et de se représenter partiellement son environnement (et les autres) ;
- 3- de *communiquer* directement ou indirectement avec d'autres agents ;
- 4- de se *conserver* et de se *reproduire* ;

Par ailleurs, il est dirigé par des tendances internes (buts, recherche de satisfaction, utilité) et présente un comportement autonome qui est la conséquence de ses perceptions, de ses représentations et de ses communications ». Cette définition de l'agent renvoie donc immédiatement à ses fonctions essentielles, qui sont :

- **La fonction cognitive** regroupant les capacités de représentation, de mémorisation et de raisonnement de l'agent ;

- **La fonction conative (ou fonction de contrôle)** qui renvoie à la question centrale de la motivation de l'agent et à sa prise en compte différenciée des contraintes normatives du système qui le conduit à une sélection d'actions qu'il s'apprête à effectuer.

- **La fonction interactionnelle** quant à elle, traite des modalités de la connexion avec ce qui l'entoure (l'environnement et les autres agents) et de la communication elle-même.

- Enfin, la fonction *végétative* renvoie aux capacités de conservation et de reproduction de l'agent.

Une seconde définition proposée par Jennings, Sycara et Wooldridge [Jennings et al., 1998a], définit un agent comme étant un système informatique, *Situé* dans un environnement, qui agit de façon *Autonome* et *Flexible* pour atteindre les objectifs qu'on lui a assignés.

A partir de ces deux définitions, nous pouvons d'ores-et-déjà souligner et retenir les propriétés essentielles d'un agent, qui sont : *autonomie, flexibilité, action, perception, communication, réactivité, rationalité, engagement et intention*. Que devons-nous comprendre ?

- *Situé* : l'agent est capable d'agir (*Action*) sur son environnement, à partir des entrées sensorielles de cet environnement (*Perception*).
- *Autonome* : l'agent est capable d'agir tout seul (*Engagement*) sans l'intervention d'un tiers et contrôle ses propres actions et son état interne (*intention*).
- *Flexible* : dans ce cas, l'agent doit être :
 - *Capable de répondre à temps (Réactivité)* : à partir de ces perceptions et des stimuli qu'il reçoit, l'agent doit établir une réponse dans un temps adéquat.
 - *Proactif* : l'agent n'agit pas simplement aux stimuli de son environnement, il est aussi capable de démontrer des comportements dirigés par des buts en prenant des initiatives, il doit montrer un comportement opportuniste, tout en prenant la bonne initiative au bon moment (*Rationalité*).
 - *Social* : l'agent doit être en mesure d'interagir (*Communiquer*) avec les autres agents quand la situation l'exige, afin d'accomplir ses tâches ou aider les autres agents à accomplir les leurs.

Il va de soi qu'en applications, on peut garder toutes les propriétés citées ci-dessus, en ajouter d'autres ou bien n'utiliser que certaines d'entre-elles selon nos besoins. En effet, l'utilisation de certaines propriétés comme l'autonomie, la flexibilité et la sociabilité donne naissance au paradigme agent tout en le distinguant des systèmes conventionnels comme les systèmes experts, les systèmes distribués ou encore les systèmes orientés objets.

Remarquons tout-de-même qu'à partir de ces propriétés, on peut distinguer deux tendances propres à chaque agent :

- *Une tendance sociale* relative aux connaissances et aux comportements de l'agent au sein d'une communauté d'agents.
- *Une tendance individuelle* comprenant les règles et les connaissances régissant le fonctionnement interne de l'agent.

b. Déterminant d'un agent :

On appelle *déterminant* d'un agent, l'ensemble nécessaire et suffisant, composé des caractéristiques environnementales, structurelles et comportementales, permettant d'expliquer et de comprendre les actions d'un agent.

En ce qui concerne les caractéristiques environnementales, elles concernent la perception de l'agent de son environnement et de lui-même. Quant aux caractéristiques structurelles elles concernent les composants internes de l'agent. Enfin, nous définissons les

caractéristiques comportementales comme étant celles qui contraignent le comportement de l'agent, en accord avec les caractéristiques environnementales.

2. Les différents types et architectures d'agents:

On distingue habituellement deux grandes familles d'agents selon leur niveau de complexité individuelle c'est-à-dire leur granularité. Si l'agent ne peut que percevoir son environnement en réagissant de manière réflexe, il est dit *réactif*, la métaphore de la fourmilière comme société d'agents « réactifs » développée par **Drogoul [1993]** en constitue l'exemple archétypique. Quand ce dernier est en mesure de se construire une représentation personnelle du monde virtuel dans lequel il évolue et d'élaborer des stratégies pour atteindre son objectif, il est dit *cognitif*.

Cela dit, nous pouvons trouver d'autres agents plus spécialisés, que nous allons essayer de présenter un peu plus loin au cours de ce chapitre.

a. Les agents cognitifs :

Les systèmes d'agents cognitifs sont fondés sur la coopération d'agents, capables à eux seuls d'effectuer des tâches complexes. Pour ce faire, de tels agents ont hérité de l'IA classique, d'un concept très fort, que l'on appelle *la planification*. Pour faire court, on peut dire que la planification est un sous domaine de l'IA, qui a pour objectif de répondre à la question suivante : " *Que doit-on faire ? et à quel moment ?*".

Un tel système est muni d'un petit nombre d'agents qui disposent d'une capacité de raisonnement à partir d'une base de connaissances, d'une aptitude à traiter les informations relatives à un domaine d'application, et celles qui sont liées à la gestion des interactions avec les autres agents et l'environnement.

Chaque agent cognitif est assimilé selon son niveau de capacités à un système expert plus ou moins sophistiqué. On parle alors d'agents de forte granularité. Ces agents sont munis généralement des composantes suivantes :

- Un modèle symbolique de l'environnement de l'agent, typiquement dans un sous-ensemble limité de la logique du premier ordre ;
- Une spécification symbolique des actions que l'agent peut accomplir, généralement sous forme de *pré-condition*, *action*, *effet*, qui représentent l'état de l'environnement avant que l'action ne soit posée ainsi que les conséquences de la dite action ;
- Un algorithme de planification qui est apte à manipuler les symboles définis et qui génère en sortie, un plan représentant les actions que devra poser l'agent pour atteindre son but.

Le cycle de contrôle d'un agent cognitif peut être implémenté comme suit :

```
State state = initialize();
while (true) {
  percepts = perceive();
  state = updateState (state, percepts);
  action = deliberate(state, plans);
  execute(action);
};
```

Après avoir pris connaissance de ce que sont les agents cognitifs, il serait judicieux de décrire leur architecture et d'où provient leur "*intelligence*".

Avant de commencer à présenter l'architecture des agents, il faut d'abord la définir. Même s'il n'y a pas consensus sur ce point, on peut dire que **l'architecture d'un agent** est une description de son organisation interne : les données et les connaissances de l'agent, les opérations qui peuvent être effectuées sur ses composantes et le flux de contrôle des opérations. Le choix d'une architecture ou d'une autre est bien sûr, lié à la structure conceptuelle de l'agent et représente la décision du concepteur sur la façon de bâtir l'agent artificiel.

La figure suivante représente l'architecture générique relative aux agents cognitifs :

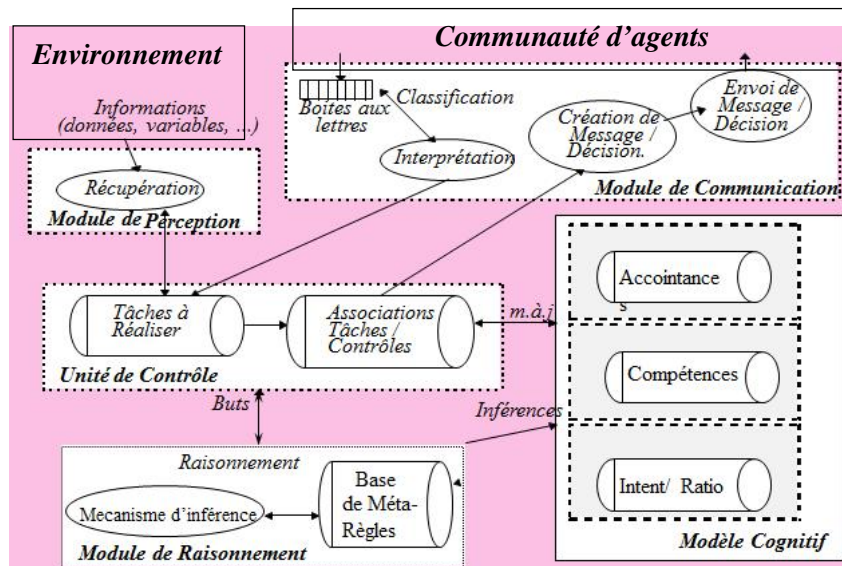


Figure 1 : Architecture d'agents cognitifs [Kabachi].

Nous pouvons remarquer d'après cette figure que l'architecture interne d'un agent est une architecture modulaire dont les principaux modules sont :

- ✓ **Module de connaissances du domaine** : où on trouve généralement :
 - Une base de faits conçue pour définir :
 - **Les Concepts de Compétences CC** : c'est-à-dire les idées directrices pour déclencher les méthodes de compétences, **Ex** : améliorer les performances du système ;
 - **Les Méthodes de Compétences MC** : qui représentent les connaissances sur les méthodes, les heuristiques et les différents paramétrages, **Ex** : les méthodes de gestion de production ou celles de la gestion du personnel ;
 - Une base de règles : contenant principalement :
 - **Les règles de conception des solutions ;**
 - **Les règles de raisonnement ;**
 - **Les Méta-règles.**
- ✓ **Module de contrôle** : représentant l'intelligence proprement dite de l'agent c'est-à-dire :
 - Les intentions.
 - Les croyances.
 - Les décisions.
 - Les rationalités.

- *Les engagements.*
- ✓ **Module de communication** : permet essentiellement à l'agent de communiquer avec ses semblables pour échanger des données de toutes sortes. Nous trouvons :
 - *Accointances.*
 - *Protocoles de communication.*
 - *Messages.*

Nous avons défini l'ensemble des modules composant l'agent. Il faut souligner que la modularité introduit beaucoup de souplesse dans la gestion des capacités en autorisant à moindre coût l'échange de modules dans un but d'évolutivité ou de test. Une architecture modulaire fait de l'agent un système ouvert.

b. Les agents réactifs :

Depuis les années 70 jusqu'au début des années 80, la recherche sur la planification s'est surtout concentrée sur les trois aspects exposés ci-dessus. Principalement, on voulait démontrer l'efficacité des algorithmes de planification développés. Malheureusement, on s'est vite aperçu que, bien qu'ils donnent de bons résultats pour des petits problèmes, les algorithmes de planification de l'époque avaient une performance plutôt médiocre lorsqu'on les applique à des problèmes du monde réel. En effet, comme la taille de l'espace de recherche des algorithmes de planification croît de façon exponentielle avec la complexité de la tâche pour laquelle on cherche à établir un plan, et que les tâches du monde réel sont des tâches très complexes, on obtient par conséquent un espace de recherche extrêmement large, ce qui donne des temps de recherche proportionnellement longs.

Devant l'échec des chercheurs à développer des algorithmes de planification efficaces dans le monde réel, certains chercheurs ont commencé à émettre de grandes réserves quant à l'approche *logicienne* (appelée aussi *approche symbolique*) utilisée jusqu'à maintenant.

Un des grands noms parmi les critiques du raisonnement symbolique fut Brooks qui, par le biais de plusieurs papiers [Brooks, 1986, 1991a, 1991b], exposa son opposition au modèle symbolique et proposa une approche alternative qu'on appelle aujourd'hui *IA réactive*. Selon lui, dans un système multi-agents, il n'est pas nécessaire que chaque agent soit individuellement intelligent pour parvenir à un comportement global intelligent. En effet, des mécanismes simples réagissant à des stimuli/événements peuvent faire émerger des comportements correspondant aux objectifs poursuivis. Ainsi, dans cette optique, au sein de son programme de recherche, Brooks a développé l'architecture *subsumption* (Figure 2). Dans cette architecture, on bâtit des *agents réactifs* sans utiliser de représentation symbolique ni de raisonnement. Un agent réactif est alors vu comme un ensemble de comportements accomplissant une tâche donnée. Chaque comportement est une machine à états finis qui établit une relation entre une entrée sensorielle et une action en sortie.

Cette approche propose la coopération d'agents de *faible granularité* mais en nombre beaucoup plus important.

A partir de là, on peut définir les *agents réactifs* comme étant des agents de bas niveau qui ne disposent que d'un protocole et d'un langage de communication réduits (Figure 3), leurs capacités et leur cycle de construction répondent uniquement à la loi *Stimuli/Action*, comme on le décrit ci-dessous :

```
do {
  percepts = perceive();
  action = getAction(percepts, rules);
  execute(action);
} while (true);
```

- Exemple de règle

```
if (light.isOn()) then open-the-eyes
```

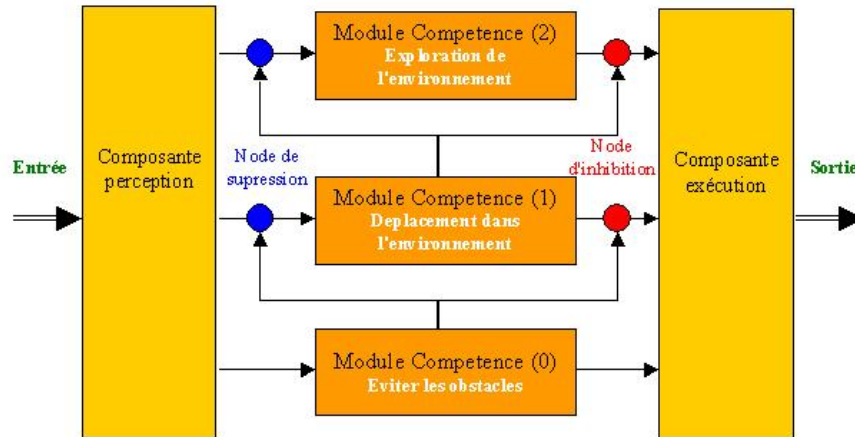


Figure 2 : Architecture réactive de subsumption [Brooks, 1986].

Ce que nous devons retenir de cette architecture, ce sont les trois modules de compétences désignés respectivement par *Module de compétence (M1)*, *(M2)* et *(M3)*.

Si on utilise cette architecture pour construire un robot d'exploration, on peut définir :

- M0 - un module qui a la compétence d'éviter les obstacles ;
 - M1 - un module M1 qui est responsable des déplacements dans l'environnement tout en évitant les obstacles à l'aide de M0 ;
 - M2 - un module qui a la compétence supérieure, la plus abstraite, de faire l'exploration systématique de la planète en se déplaçant grâce aux actions du module M1.
- Un module sur une couche inférieure a une priorité plus grande qu'un module situé sur une couche plus élevée, parce qu'il est responsable d'une tâche plus simple mais plus "urgente". Dans ce but, le fonctionnement d'un module situé sur une couche supérieure est subordonné à un module inférieur. Un module sur une couche inférieure peut modifier l'entrée d'un module supérieur au moyen d'un nœud de suppression, et invalider l'action du module supérieur au moyen d'un nœud d'inhibition, comme on peut le voir dans la figure 2. Par exemple, si notre robot veut se déplacer vers l'Est en partant d'une certaine position et qu'il n'y a pas d'obstacle dans cette direction, l'action exécutée par la composante exécution est celle commandée par M1 de se déplacer vers l'Est. Si, par contre il y a un obstacle, le module M0 prend en compte cet obstacle par sa perception de l'environnement et inhibe le déplacement vers l'Est. M1 essaiera alors de se déplacer dans une autre direction. C'est cette organisation qui justifie l'appellation de subsumption de l'architecture.

Le fonctionnement de l'agent est décrit par un ensemble de règles de comportement, "behaviour rules" en anglais. Une **règle de comportement** est semblable à une règle de production, elle a deux parties : une condition **c** et une action **a**. La condition correspond à

une perception p de l'environnement, et l'action a une possibilité d'action d'un module de compétence. Soit $\mathbf{Comp} = \{(c, a) \mid c \text{ dans } P, a \text{ dans } A\}$.

A partir de cette architecture, nous pouvons à présent déterminer celle qu'on attribue aux agents réactifs. Cette architecture se veut d'être générique et plus à-même de répondre aux agents de nature abstraite telle que nous la décrivons dans la figure 3 ci-après:

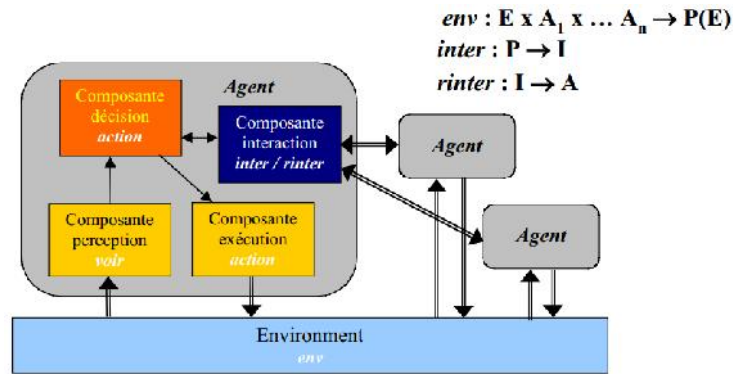


Figure 3 : Architecture d'agents réactifs.

Bousquet et Lepage [2001] reviennent en détail sur ce type d'approche généralement utilisée pour la simulation ; malgré la simplicité apparente et les bons résultats obtenus pour certaines applications, bien des reproches ont été adressés à cette approche dite "réactive", parmi lesquels, il convient de citer les suivants :

- si les agents ne possèdent pas de modèle de leur environnement, ils doivent posséder suffisamment d'informations locales leur permettant de choisir une action acceptable;
- comme les agents basent leurs décisions sur des informations locales, il est difficile de voir comment ils pourraient tenir compte des informations non-locales;
- il est difficile de voir comment un agent purement réactif peut apprendre de son expérience et améliorer ainsi ses performances;
- le comportement global d'un agent *devrait émerger* des interactions entre les divers comportements qui le composent, cette *émergence* rend donc très difficile la tâche de construire un agent dans le but d'effectuer une tâche spécifique;
- s'il est assez simple de bâtir un agent qui comporte très peu de couches, l'exercice devient beaucoup plus compliqué lorsqu'on a besoin de plusieurs couches. Les interactions dynamiques entre les diverses couches deviennent trop complexes à comprendre.

Les deux types d'agents étant définis, nous proposons le tableau récapitulatif suivant qui traite des différences les plus importantes entre eux :

<i>Agents Cognitifs</i>	<i>Agents Réactifs</i>
Représentation explicite de l'environnement.	Pas de représentation explicite de l'environnement.
Peut tenir compte de son passé.	Pas de mémoire. Aucun historique.
Agents complexes.	Fonctionnement Stimuli/Action.
Petit nombre d'agents pour résoudre un problème.	Nombre important d'agents pour résoudre un problème.

Tableau 1 : Agents Cognitifs Vs Agents Réactifs

c. Les agents BDI :

BDI est un acronyme qui signifie, en anglais, *Belief, Desire, Intentions*, ce qui se traduit en français par *croyances, désirs et intentions*. Les agents se basent donc sur ces trois aspects pour choisir leurs actions. Dans ce cadre, **Wooldrige [1999]** propose une architecture ayant sept composantes, telles que présentées dans la figure 4 :

- Un ensemble de *croyances* courantes, représentant les informations que l'agent possède à propos de son environnement courant ;
- Une *fonction de révision des croyances*, qui prend les entrées des capteurs et les croyances actuelles de l'agent et qui détermine un nouvel ensemble de croyances ;
- Une *fonction de génération des options*, qui détermine les options disponibles pour l'agent (i.e. ses désirs), en se basant sur les croyances courantes de l'agent à propos de son environnement et sur ses intentions courantes ;
- Un ensemble de *désirs*, représentant les options disponibles à l'agent ;
- Une *fonction de filtre*, qui représente le processus de délibération de l'agent et qui détermine les intentions de l'agent en se basant sur ses croyances, ses désirs et ses intentions courantes ;
- Un ensemble d'*intentions* courantes, représentant le centre d'attention actuel de l'agent, c'est-à-dire les buts envers lesquels il s'est engagé et envers lesquels il a engagé des ressources ;
- Une *fonction de sélection des actions*, qui détermine l'action à effectuer en se basant sur les intentions courantes de l'agent.

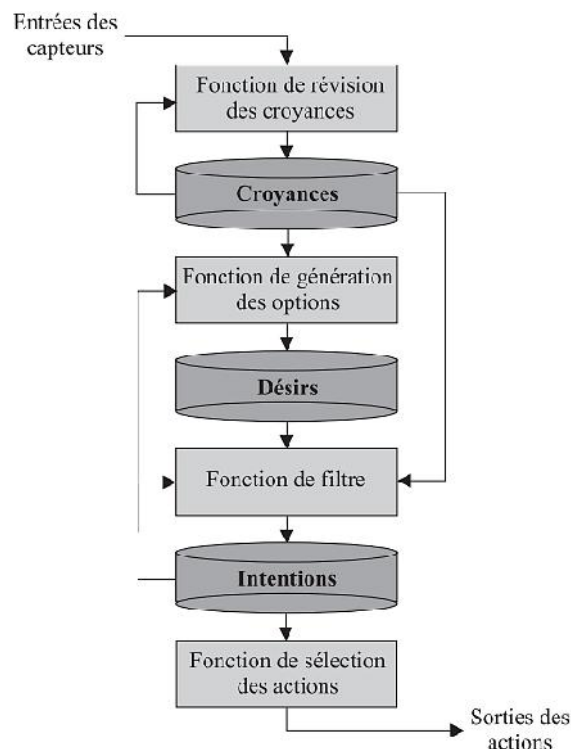


Figure 3 : architecture BDI [Wooldrige, 1999].

En résumé, un agent BDI doit donc mettre à jour ses croyances avec les informations qui lui proviennent de son environnement, décider quelles options lui sont offertes, filtrer ces options afin de déterminer de nouvelles intentions et poser ses actions en se basant sur ses intentions.

d. Les agents hybrides :

A la croisée des chemins, entre les *agents cognitifs* et les *agents réactifs*, les *agents hybrides* [Ferber, 1995] ; [Muller et Pischel, 1994], ont vu le jour comme étant une synthèse entre les deux orientations cognitives et réactives. Ces agents combinent donc les propriétés des deux orientations et sont généralement logées dans des modules différents. Ce type d'architecture pose le problème du contrôle à deux niveaux : la coordination classique entre agents, et la coordination entre modules au sein même de l'agent.

Les architectures hybrides apportent une solution au problème de l'intégration des aspects réactifs et cognitifs. Ces architectures présentent des qualités indéniables de génie logiciel. Leur organisation modulaire permet entre autres l'intégration de capacités très hétérogènes en termes de programmation. Cette hétérogénéité est nécessaire pour répondre à l'intégration des tâches très diversifiées qu'un agent doit remplir depuis le raisonnement jusqu'à l'interaction, ces tâches peuvent faire appel à des concepts de système des plus spécialisés. L'intérêt est de pouvoir réutiliser des méthodes existantes notamment des méthodes d'intelligence artificielle [Guessoum et Ocello, 2001].

Cependant, les architectures hybrides sont complexes et ne seraient pas appropriées à la modélisation d'agents dont le comportement est très simple par exemple, basé sur le principe de l'éco-résolution [Ferguson, 1992]. Ces architectures hybrides ne résolvent pas le problème de granularité. Elles ne permettent pas la conception de systèmes multi-agents dont la granularité des agents est variable.

Plusieurs architectures hybrides ont été conçues et utilisées. Une des architectures hybrides les plus connues est celle du **système InteRRaP** ("Integration of Reactive Behavior and Rational Planning" = Intégration du comportement réactif et planification rationnelle). L'architecture InteRRaP est une architecture en couches avec des **couches verticales** où les données d'entrée, notamment les perceptions, passent d'une couche à l'autre, comme on le voit sur la figure 4. En cela, elle est un peu différente de l'architecture de subsomption qui est une architecture en **couches horizontales**, où les perceptions sont transmises directement à toutes les couches à la fois (figure 3).

Un agent InteRRaP est comparable à un agent BDI qui a des buts à atteindre (les buts sont les mêmes que les désirs) et qui est capable de coopérer avec d'autres agents InteRRaP pour accomplir ces buts. Les buts d'un agent sont divisés en trois catégories :

- **réactions** : ce sont des buts simples à accomplir en fonction des perceptions sur l'environnement ;
- **buts locaux** : ce sont des buts que l'agent peut accomplir par lui-même ;
- **buts coopératifs** : ce sont les buts qui peuvent être accomplis uniquement par une coopération avec d'autres agents dans le système.

Il est intéressant d'observer que, dans cette conception, la réactivité de l'agent est conçue toujours comme un but, c'est-à-dire au niveau cognitif, mais comme un but très simple à réaliser.

L'architecture InteRRaP est composée de **trois couches de contrôle** et **trois bases de connaissances** associées qui représentent l'agent et l'environnement à divers niveaux d'abstraction, comme il est indiqué dans la figure 4. Chaque couche a un ensemble d'opérations spécifiques associées et une couche supérieure utilisant les opérations plus simples de la couche d'au-dessus pour exécuter ses opérations plus élaborées. Le flux de contrôle passe de bas en haut, une couche prend le contrôle lorsque la couche antérieure ne peut plus contribuer, par ses opérations, à l'accomplissement des buts. Chaque couche comprend deux modules: un **module pour l'activation** des buts et la reconnaissance des situations (**AR**) et un **module de planification et d'exécution** (**PE**). Les perceptions sur l'environnement sont transmises au module AR de la première couche et, de module en module, vers le sommet de la hiérarchie. Le flux de contrôle des actions passe de haut en bas pour arriver à la fin au module PE de la dernière couche, et les actions associées sont exécutées sur l'environnement.

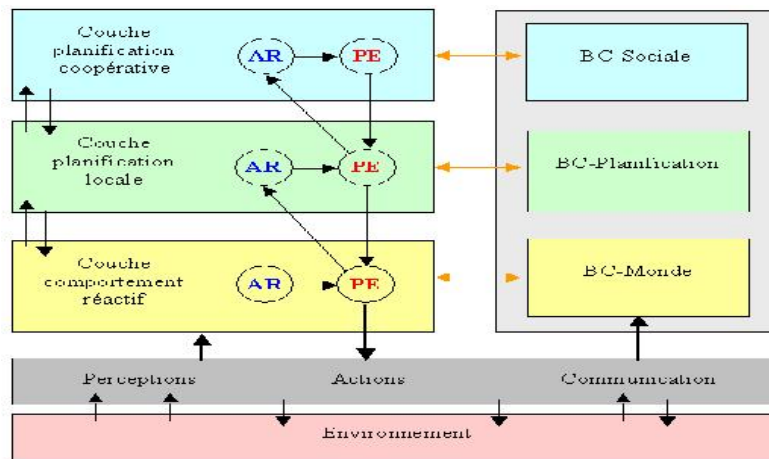


Figure 4 : Architecture hybride InteRRaP [Mul 94]

Les architectures hybrides se sont avérées efficaces dans beaucoup d'applications. Il y a cependant deux reproches qu'on peut leur faire : il n'y a pas de modèle formel associé comme dans le cas d'une architecture BDI pure, et on n'a pas encore validé de véritable méthodologie pour guider l'utilisateur dans la conception d'applications utilisant de telles architectures.

Jusque-là, nous avons présenté des types d'agents génériques qui représentent des moules pour d'autres agents plus spécialisés et plus spécifiques à certains domaines. Parmi eux, nous pouvons citer les plus importants comme :

e. Les agents logiciels :

Les premiers et les plus simples de cette catégorie ce sont les agents *Démons* d'UNIX, qui sont des processus informatiques autonomes, capables de se réveiller à certains moments ou sous certaines conditions.

Une version plus sophistiquée de ce type d'agent, sont les virus informatiques qui sont très doués pour se reproduire et faire du mal autour d'eux.

De toutes les façons on peut définir ce type d'agents comme étant des logiciels par opposition aux agents physiques, tel un robot.

f. Les agents mobiles :

L'idée principale derrière ce type d'agents, est d'offrir la possibilité à un agent informatique de se déplacer d'une machine à une autre pour traiter ainsi les données localement. Le premier objectif étant de minimiser les communications distantes et ainsi minimiser les coûts des traitements car il est plus facile de faire transiter un code source qu'une quantité volumineuse de données à traiter.

La première plate-forme de ce type d'agents est *telescript* née au milieu des années 90 et qui a ensuite donné naissance à plusieurs variantes, extensions du langage Java telles que Aglets, Voyager, etc....

g. Les agents assistants :

Pour pouvoir dépasser les limitations des interactions Homme-Machine à manipulation directe (rigidité, complexité...). Les agents assistants (appelés aussi agents intelligents) apportent une adaptation au profil de l'utilisateur et une capacité d'anticiper leurs besoins (automatiser certaines tâches, rappeler certaines informations...). L'exemple le plus courant est les agents PDA des agendas électroniques.

On peut citer d'autres types d'agents comme *les agents autonomes, les agents sociaux* ou encore *les agents flexibles...etc.*

L'agent étant défini, on peut tenter à-présent une définition des *Systèmes Multi-agents* (SMA).

III- Les Systèmes Multi-Agents (SMA) :

1. Définitions :

Une définition simple d'un SMA reviendrait à le considérer comme un ensemble d'agents qui se déploient dans un environnement et qui évoluent dans l'espace des interactions entre eux. Le SMA pouvant être structuré selon une organisation qui représente un niveau de description intermédiaire entre les agents et le système.

Dans son ouvrage intitulé « concepts et méthodologies multi-agents. P26-27 » **Ferber [1995]**, estime que trois concepts fondamentaux sont au principe même d'un SMA. Il s'agit :

- En premier lieu de *l'autonomie* de l'activité de l'agent, « c'est-à-dire sa capacité à exécuter une action de sa propre initiative, en contrôlant son comportement afin d'atteindre une plus grande satisfaction et en décidant d'aider ou d'empêcher les autres de satisfaire leurs buts ».
- En second lieu, le concept central qui est la *sociabilité* des agents, « c'est-à-dire le fait qu'ils agissent en relation avec d'autres agents dans une perspective sociale. Un agent dans un SMA n'est pas une entité isolée mais un élément d'une totalité plus vaste que l'on appelle société d'agents. Une société d'agents émerge des interactions qui ont lieu entre les agents, mais en même temps son organisation contraint le comportement des agents en leur attribuant des rôles qui vont restreindre leurs possibilités d'action ».
- En dernier lieu, c'est évidemment le concept *d'interaction* qui constitue la clé de voûte des SMA et qui représente la passerelle entre *l'autonomie* et la *sociabilité* des agents puisque comme le rappelle l'auteur, « c'est par cet entrelacement d'actions effectuées par un ensemble d'agents (chaque action étant décidée individuellement par chaque agent dans son for intérieur) que des formes émergent, que des structures organisées apparaissent pour venir en retour, contraindre et modifier le comportement de ces agents. De ce fait, c'est au travers

de ces interactions qu'émergent des *patterns d'interaction* que sont la coopération, les conflits ou la concurrence, qui produisent des formes d'organisation plus ou moins stables, qui viennent structurer l'action individuelle de chacun des agents ».

En mêlant les concepts d'autonomie et de sociabilité d'agents en interaction dans un univers virtuel, les SMA se sont affirmés en une dizaine d'années seulement comme un outil majeur de modélisation mais en transcendant la notion même d'outils.

En effet, dans la mesure où ils sont des sociétés d'agents en interaction, les SMA deviennent ce que Ferber appelle des « microcosmes » ou encore des « micro-mondes artificiels » sur lesquels on peut pratiquer toute expérimentation.

Quant à **Durfee et al. [1989]**, définit les SMA comme « *un réseau faiblement couplé d'entités agissant ensemble pour résoudre des problèmes qui dépassent leurs capacités individuelles* ».

Une autre définition considère qu'un système multi-agents est une société d'entités autonomes et hétérogènes appelés agents. Le comportement de chaque agent est guidé par deux impératifs : un *impératif local* et un *impératif global*. Le niveau local touche individuellement chaque agent qui agit en fonction de ses buts personnels. Le niveau global concerne l'ensemble des agents qui coopèrent afin d'atteindre leurs objectifs communs. Ces objectifs peuvent ne pas être accessibles à un seul agent, mais seulement par la collaboration d'un ensemble d'individus. Ainsi, le SMA vu comme une société exprime bien les relations entre les agents : ceux-ci peuvent collaborer, coordonner leurs actions et/ou négocier pour résoudre des conflits locaux [**Erceau et Ferber, 1991**].

2. Caractéristiques des SMA :

Un SMA est un système distribué composé d'un ensemble d'agents. Contrairement aux systèmes d'IA, qui simulent dans une certaine mesure les capacités du raisonnement humain, les SMA sont conçus et implantés idéalement comme un ensemble d'agents inter-agissants, le plus souvent, selon des modes de *coopération*, de *concurrence* ou de *coexistence* [**Chaib-draa, 1994 ; 1996**].

Un SMA est généralement caractérisé par :

- chaque agent possède des informations ou des capacités de résolution de problèmes limitées, ainsi chaque agent a un point de vue partiel;
- il n'y a aucun contrôle global du système multi-agent;
- les données sont décentralisées;
- le calcul est asynchrone.

Les SMA sont des systèmes idéaux pour représenter des problèmes possédant de multiples méthodes de résolution, de multiples perspectives et/ou de multiples solveurs. Ces systèmes possèdent les avantages traditionnels de la résolution distribuée et concurrente de problèmes comme la modularité, la vitesse (avec le parallélisme), et la fiabilité (due à la redondance). Ils héritent aussi des bénéfices envisageables de l'IA comme le traitement symbolique (au niveau des connaissances), la facilité de maintenance, la réutilisation et la portabilité mais surtout, ils ont l'avantage de faire intervenir des schémas d'interactions sophistiqués. Les types courants d'interactions incluent la coopération (travailler ensemble à la résolution d'un but commun) ; la coordination (organiser la résolution d'un problème de telle sorte que les interactions nuisibles soient évitées ou que les interactions bénéfiques soient exploitées) ; et la négociation (parvenir à un accord acceptable pour toutes les parties concernées).

Pour **Ferber [1995]**, on appelle système multi-agents un système composé des caractéristiques suivantes:

1. Un environnement **E**, c'est à dire un espace disposant généralement d'une métrique.
2. Un ensemble d'objets **O**. Ces objets sont situés, c'est à dire que, pour tout objet, il est possible à un moment donné d'associer une position dans E. Ces objets sont passifs, c'est à dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents.
3. Un ensemble **A** d'agents, qui sont des objets particuliers représentant les entités actives du système.
4. Un ensemble de relations **R** qui unissent des objets (et donc des agents) entre eux.
5. Un ensemble d'opérations **Op** permettant aux agents de A de percevoir, produire, consommer, transformer et manipuler des objets de O.
6. Des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on appellera les lois de l'univers."

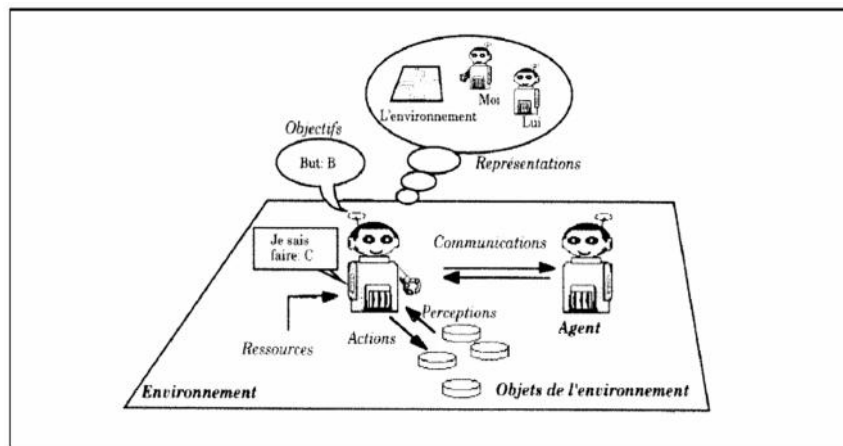


Figure 5 : Représentation d'un agent en interaction avec son environnement et les autres agents [Ferber, 1995].

Cela étant dit, bien que les SMA offrent de nombreux avantages potentiels, ils doivent aussi faire face à beaucoup de difficultés. Les problèmes inhérents à la conception et à l'implémentation des SMA se présentent comme suit : **[Bond et Gasser, 1988]**, **[Franklin et Graesser, 1997]**, **[Iglesias et al., 1997]** :

- Comment formuler, décrire, décomposer, et allouer les problèmes et synthétiser les résultats ?
- Comment permettre aux agents de communiquer et d'interagir? Quoi et quand communiquer ?
- Comment assurer que les agents agissent de manière cohérente i) en prenant leurs décisions ou actions, ii) en gérant les effets non locaux de leurs décisions locales et iii) en évitant les interactions nuisibles?
- Comment permettre aux agents individuels de représenter et raisonner sur les actions, plans et connaissances des autres agents afin de se coordonner avec eux ? Comment raisonner sur l'état de leurs processus coordonnés (comme l'initialisation ou la terminaison) ?
- Comment reconnaître et réconcilier les points de vue disparates et les intentions conflictuelles dans un ensemble d'agents essayant de coordonner leurs actions ?

- Comment trouver le meilleur compromis entre le traitement local au niveau d'un seul agent et le traitement distribué entre plusieurs agents (traitement distribué qui induit la communication) ? Plus généralement, comment gérer la répartition des ressources limitées ?
- Comment éviter ou amoindrir un comportement nuisible du système global, comme les comportements chaotiques ou oscillatoires ?
- Comment concevoir les plates-formes technologiques et les méthodologies de développement pour les SMA ?

Ce sont autant de questions que nous pouvons nous poser et autant de défis à relever et à surmonter pour un système multi-agents.

3. Décomposition d'un SMA :

A partir des travaux menés par N. Taghezout [Taghezout 2010, 2011], nous remarquons qu'il existe plusieurs décompositions relatives aux SMA, mais il y'en a une particulièrement qui a fait l'unanimité auprès de la communauté de chercheurs et c'est précisément celle-là que nous allons adopter et présenter. Il s'agit de la décomposition introduite par Yves Demazeau [Demazeau, 1995]. Dans cette décomposition, nous pouvons identifier un axe Agent, un axe Environnement, des Interactions et une structure Organisationnelle explicite ou pas.

$$\text{SMA} = \text{Agent} + \text{Environnement} + \text{Interaction} + \text{Organisation}$$

a. Agent : est une sorte d'organisme vivant dont le comportement se résume à communiquer, à agir et à éventuellement se reproduire, dans le but de satisfaire des besoins et des objectifs. Ces besoins sont déterminés à partir de l'état interne de l'agent et de l'environnement.

b. Environnement : un environnement est un espace représentant le monde dans lequel les agents évoluent. On fait généralement une distinction entre les agents qui sont les entités actives, et les objets passifs qui se situent dans l'environnement. Lorsque ce dernier dispose d'une métrique (cas général), la capacité de perception d'un agent est sa capacité à reconnaître les objets situés (position, relations entre objets) et la capacité d'action d'un agent est celle de transformer l'état du système en modifiant les positions et relations qui existent entre les objets. Par exemple dans un univers de robots, l'environnement est l'espace géométrique euclidien dans lequel se déplacent les robots (agents) et où sont situés des objets physiques que les robots peuvent manipuler ou doivent éviter.

c. Interaction : [Ferber, 1995] définit une interaction comme étant la mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques. Une interaction peut se caractériser par une simple trace qu'un agent laisse dans l'environnement (déplacement d'un objet par exemple) et la perception de cette trace par un autre agent. Elle peut également se caractériser par un message que s'échangent deux agents ou encore par un acte de langage entre ces deux derniers.

d. Organisation : est une structure décrivant comment les membres d'une organisation sont en relation et interagissent afin d'atteindre un but commun. Cette structure peut être statique, donc conçue *a priori* par le programmeur du système ou dynamique.

4. Interactions entre les agents au sein d'un SMA :

Selon Ferber [1995], pour un agent, l'interaction constitue à la fois la source de sa puissance et l'origine de ses problèmes. Comme on l'a déjà vu, l'une des principales propriétés de l'agent dans un système multi-agents est celle d'interagir avec les autres. Ces interactions sont généralement définies comme étant toute forme d'action exécutée au sein de la société qui a pour effet de modifier le comportement d'un autre agent. Elles donnent à

l'agent la possibilité de participer à la satisfaction d'un but global. Cette participation lui permet alors d'évoluer vers un des objectifs qui le composent. L'interaction peut être décomposée en trois phases non nécessairement séquentielles [Chaib-draa, 1996] :

- la réception d'informations ou la perception d'un changement ;
- le raisonnement sur les autres agents ;
- une émission de messages(s) ou plusieurs actions (plan d'actions) modifiant

l'environnement. Cette phase est le résultat d'un raisonnement de l'agent sur son propre savoir-faire et celui des autres agents.

a. L'interaction par la communication :

La communication, dans les Systèmes multi-agents, est souvent l'un des moyens utilisés pour échanger des informations entre agents (ex. plans, résultats partiels, buts, etc.). L'interaction constitue un niveau supérieur à la notion de communication et d'action. Le problème est alors de savoir combiner ces éléments (communications et actions) afin de coordonner et de contrôler les échanges entre plusieurs agents pour avoir un comportement collectif cohérent du système [Mazouz, 2001], [Taghezout, 2010].

Ainsi, les questions abordées par un modèle de communication peuvent être résumées par l'interrogation suivante : qui communique quoi, à qui, quand, pourquoi et comment ?

Pourquoi communiquer ?

- Les agents communiquent et interagissent pour synchroniser leurs actions et pour résoudre des conflits (de ressources, de buts ou d'intérêts).
- Pour s'aider mutuellement ou pour suppléer aux limites de leurs champs de perception.

Quand et avec qui communiquer ?

- Les agents communiquent lorsqu'ils sont face à un problème qu'ils ne savent pas résoudre (manque de compétences ou de ressources),
- Lorsqu'il est nécessaire de coordonner leurs actions,
- Lorsqu'il y a un conflit entre plusieurs agents.

Comment communiquer ?

Deux procédures de communication existent pour véhiculer les messages (porteurs d'informations ou d'actions) entre agents :

- Communication par partage d'informations (tableau noir).
- Communication par envoi de messages.

Pour la *communication par partage d'information*, les informations relatives au système complet sont centralisées et sauvegardées dans une structure de données globale appelée tableau noir. Les agents viennent lire et écrire dans cette base de données pour faire évoluer le système qui contient initialement les données du problème.

L'inconvénient majeur de cette technique est qu'elle porte atteinte à l'un des paradigmes le plus important d'un SMA c'est-à-dire l'aspect social. En effet, le comportement des agents se trouve conditionné et fortement lié au contenu du tableau, et donc pas d'échange entre eux.

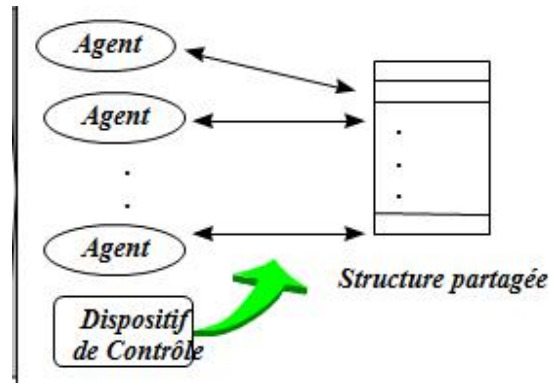


Figure 6 : Mécanisme du Blackboard.

Le deuxième mode est la *communication par envoi de message* qui se caractérise par le fait que chaque agent possède une représentation propre et locale de son environnement et va la partager avec ses pairs, de telle sorte que chaque agent puisse avoir une vision assez complète de tout le système (environnement, buts,...). Ce type de communication se fait soit en mode point à point, soit en diffusion. Cette communication permet de réaliser un véritable système multi-agents puisque chaque agent possède sa propre base de connaissance. Par contre, il est plus difficile d'assurer une convergence globale du système. De plus, dans un environnement distribué, la communication entre un grand nombre d'agents peut très vite amener à une saturation du réseau compte tenu de la grande quantité de messages échangés.

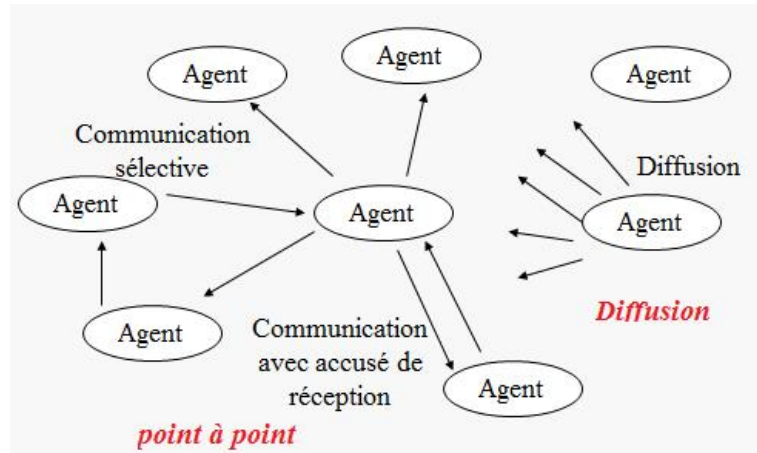


Figure 7 : Communication par envoi de messages.

Un acte de communication est un message au sens de FIPA, c'est-à-dire une structure de données comprenant les champs suivants :

Sender : l'identifiant de l'agent émetteur, caractérisé par son adresse (agent ID).

Receiver : l'identifiant du ou des agent(s) destinataire(s). Dans ce modèle, nous utilisons l'agent ID spécifique.

all pour indiquer que le message est transmis par broadcast à tous les agents du système.

Performative : le performatif du message.

Content : le contenu du message, dont la nature dépend fortement du performatif.

Conversationid : l'identifiant de la conversation, qui permet aux agents de suivre les protocoles.

Messageid : l'identifiant du message dans la conversation, que nous supposons unique.

b. L'interaction par la coopération d'agent :

Les agents travaillent à la satisfaction d'un but commun ou individuel, dont l'objectif est d'améliorer le mode de travail des agents en termes de :

- validité et rationalité des informations échangées et des comportements ;
- efficacité des stratégies de résolution employées ;
- cohérence entre planification locale et globale, et
- rééquilibrage dynamique de la charge de travail.

Des agents totalement coopératifs peuvent changer leurs buts pour répondre aux besoins des autres agents, afin d'assurer une meilleure coordination entre eux. Cela peut résulter en coûts de communication élevés.

Les agents antagonistes par contre, ne vont pas coopérer et dans ce cas, leurs buts respectifs vont se trouver bloqués. Dans de tels systèmes, les coûts de communication sont minimaux.

Les interactions des agents d'un SMA sont motivées par l'interdépendance des agents selon ces trois dimensions:

- leurs buts peuvent être compatibles ou non ;
- les agents peuvent désirer des ressources que les autres possèdent ;
- un agent X peut disposer d'une capacité nécessaire à un agent Y pour l'accomplissement d'un des plans d'action Y.

c. L'interaction par la négociation entre ressources :

La négociation est avant tout un mécanisme de résolution de conflit. Cette résolution peut alors s'effectuer dans un contexte favorable ou défavorable. Dans un contexte favorable, toutes les parties sont prêtes à faire les concessions nécessaires pour arriver à un accord. Dans un contexte défavorable l'objectif principal pour au moins l'un des participants, est d'obtenir un accord qui le favorise même aux dépens des autres. De ce fait, la négociation peut être décrite comme une coopération dont l'objectif commun est l'obtention d'un accord.

5. Protocoles d'interactions :

Un système multi-agents est un ensemble organisé d'agents interactifs. L'interaction est ainsi un concept clé des systèmes multi-agents. Pour définir les interactions entre agents, différents protocoles d'interactions ont été définis dont les plus importants sont ceux qui ont été menés et normalisés par FIPA [FIPA, 2002].

Les protocoles d'interaction sont des descriptions de patterns standards d'interactions entre deux ou plusieurs agents. La majorité des travaux existants, notamment les travaux de FIPA, décrivent en effet les séquences de messages échangés et leur contenu (émetteur, receveur, performative...).

Les représentations proposées des protocoles d'interaction fournissent plusieurs facilités pour concevoir des systèmes multi-agents [Ocello, 2003]. Cependant, les actions locales et les processus de décision d'un agent ne sont pas toujours explicitement représentés.

L'implémentation des protocoles d'interaction a souvent été désignée comme un objectif par la FIPA. Cependant, le *Contract Net* est le seul protocole d'interaction qui est implémenté dans la plate-forme JADE [Guessoum, 2003]; celle-ci ne fournit que des méthodes abstraites que l'utilisateur doit implémenter.

Cette section définit les protocoles d'interaction et les comportements nécessaires à leur exécution.

a. Protocoles de Coordination :

Dans des environnements à ressources limitées, la coordination se traduit par un comportement individuel visant à servir ses propres intérêts, tout en essayant de satisfaire le but global du système. Les actions des agents doivent être coordonnées pour plusieurs raisons :

1. il y'a des dépendances entre les actions des agents ;
2. aucun agent n'a suffisamment de compétence, de ressources et d'informations pour atteindre tout seul le but du système complet ou son propre but ;
3. éviter les redondances dans la résolution de problèmes.

Pour [Jennings et al., 1998], la coordination est caractérisée par deux aspects étroitement liés qui sont les engagements et les conventions. Les engagements fournissent la structure nécessaire pour des interactions prévisibles, de sorte que les agents puissent prendre en compte les dépendances inter-agents des futures activités, des contraintes globales ou des conflits d'utilisation des ressources. Pendant que les situations changent, les agents doivent évaluer si les engagements existants sont encore valides.

Les conventions fournissent des moyens pour contrôler les engagements dans des circonstances changeantes, toutes les notions relatives à la coordination seront données en détail dans la deuxième partie de ce chapitre.

b. Protocoles de Coopération :

Une stratégie de base partagée par plusieurs protocoles pour la coopération est de décomposer un problème en tâches puis les distribuer [Mazouz, 2001]. Le processus de décomposition doit prendre en considération les ressources et les capacités de chaque agent. La décomposition peut être spatiale ou fonctionnelle. Cependant un certain nombre de critères doit être respecté :

1. éviter la surcharge des ressources critiques ;
2. assigner les tâches aux agents ayant des capacités correspondantes ;
3. assigner des tâches interdépendantes à des agents proches spatialement ou sémantiquement pour limiter les couts de communication ;
4. réassigner des tâches pour accomplir les plus urgentes.

c. Protocoles de Négociation

Une forme fréquente de l'interaction qui se produit entre des agents avec des buts différents est la négociation. La négociation est un processus par lequel une décision commune est prise par deux agents ou plus, où chacun d'entre eux essaye d'atteindre ses buts ou objectifs propres. Les dispositifs principaux de la négociation sont (1) le langage utilisé par les agents participants, (2) le protocole suivi par les agents dans le processus de négociation et (3) la procédure de décision que chaque agent utilise pour déterminer ses positions, concessions et critères pour l'accord.

6. Quelques exemples d'applications des SMA :

De nos jours, la technologie multi-agents a trouvé sa place dans les systèmes manufacturiers, les systèmes financiers, les loisirs, les télécommunications, le contrôle-commande, les systèmes embarqués, et pas mal d'autres applications.

Nous n'en exposerons que quelques exemples d'applications utilisant cette technologie, mais pour plus de détails, il serait intéressant de lire l'article de Jennings [Jennings et al., 1998] pour un ensemble d'exemples plus complet.

Nous commençons tout d'abord par un domaine d'application très riche pour les agents: les télécommunications, ensuite nous enclencherons avec d'autres systèmes, l'un œuvrant dans le domaine de la gestion du processus d'affaires, le système ADEPT, le deuxième œuvrant dans le domaine médical, le système GUARDIAN, et finalement le dernier œuvrant dans les environnements riches en information, le système NETSA.

a. Application des SMA dans les télécommunications :

Ces dernières années, les télécommunications ont notamment introduit une conception de services décentralisée dans le contexte du Web, créé de nouveaux services de médiation tels que les portails et engendrés l'apparition de nombreux fournisseurs de services réseaux qui ne disposent pas de leurs propres services réseaux. L'obtention de tels services décentralisés ne peut évidemment être réalisée que grâce à des logiciels pour lesquels les données et le contrôles sont forcément distribués. De ce fait, il est clair que les SMA semblent convenir aux télécommunications. C'est pourquoi les principaux acteurs de télécommunications mènent actuellement d'intenses activités de recherche sur la technologie agent : British Telecom, France Télécom, Deutch Telekom, NTT, Nortel, Siemens, etc.

Le système ADEPT (Advanced Decision Environment for Process Tasks):

Les gestionnaires de grandes compagnies effectuent des prises de décisions en se basant sur une combinaison de jugement et d'informations provenant de plusieurs départements.

Idéalement, toutes les informations pertinentes devraient être rassemblées avant qu'une décision ne soit prise. Cependant, le processus d'obtention d'informations, qui sont à jour et pertinentes, est très complexe et prend énormément de temps, pour cette raison, plusieurs compagnies ont cherché à développer des systèmes informatiques afin de les assister dans leur processus de décision.

Le système ADEPT réalisé par **J. A. Alty, D. Griffiths, N. R. Jennings, E. H. Mamdani, A. Struthers, et M. E. Wiegand** en 1994, s'attaque à ce problème en considérant le processus d'affaires comme un ensemble d'agents qui négocient et qui offrent des services. Chaque agent représente un rôle distinct ou un département de l'entreprise et est en mesure de fournir un ou plusieurs services (solutions).

Les agents qui requièrent les services d'autres agents le font par une négociation qui permet d'obtenir un coût, un délai temporel et un degré de qualité qui sont acceptables aux deux parties.

Le résultat d'une négociation terminée avec succès constitue un engagement entre les deux parties.

Le système GUARDIAN :

Le système GUARDIAN [**B. H. Roth, M. Hewett, R. Washington, R. Hewett, and A. Seiver**] réalisé en 1989 a pour but de gérer les soins aux patients d'une unité de soins intensifs chirurgicale. Les principales motivations de ce système sont: premièrement, le modèle des soins d'un patient dans une unité de soins intensifs est essentiellement celui d'une équipe, où un ensemble d'experts dans des domaines distincts coopèrent pour organiser les soins des patients; deuxièmement, le facteur le plus important pour donner de bons soins aux patients est le partage d'informations entre les membres de l'équipe de soins critiques. Particulièrement, les médecins spécialistes n'ont pas l'opportunité de superviser l'état d'un patient minute par minute; cette tâche revient aux infirmières qui, quant à elles, ne possèdent pas les connaissances nécessaires à l'interprétation des données qu'elles rassemblent.

Le système GUARDIAN répartit donc le suivi des patients à un certain nombre d'agents de trois types différents. Les agents *perception/action* sont responsables de l'interface entre GUARDIAN et le monde environnant, établissant la relation entre les données des senseurs et une représentation symbolique que le système pourra utiliser, et traduisant les requêtes d'action du système en commandes pour les effecteurs. Les agents en charge du raisonnement sont responsables d'organiser le processus de prise de décision du système. Finalement, les agents en charge du contrôle (il n'y en a habituellement qu'un seul) assurent le contrôle de haut niveau du système.

Les systèmes d'informations coopératifs (SIC) :

Les SIC sont généralement caractérisés par la grande variété et le grand nombre de sources d'informations. Ces sources d'informations sont hétérogènes et distribuées soit sur un réseau local (Intranet) soit sur l'Internet. De tels systèmes doivent être capables d'exécuter principalement les tâches suivantes :

- la découverte des sources : trouver la bonne source de données pour l'interroger;
- la recherche d'informations : identifier les informations non structurées et semi-structurées;
- le filtrage des informations : analyser les données et éliminer celles qui sont inutiles;
- la fusion des informations : regrouper les informations d'une manière significative.

Le système multi-agents « Warren » pourrait constituer un exemple spécifique de l'utilisation des agents dans ce type d'application. C'est un système d'agents intelligents pour l'aide des usagers dans la gestion des portefeuilles. Ce système combine les données du marché financier, les rapports financiers, les modèles techniques et les rapports analytiques avec les prix courants des actions des compagnies. Toutes ces informations sont déjà disponibles sur le Web; "Warren" ne fait que les intégrer via des agents spécialisés, les agents d'informations et ensuite les présenter aux usagers. Pour ce faire, "Warren" dispose de six agents ressources, deux agents de tâches et un agent utilisateur pour chaque usager. L'agent utilisateur affiche (via le web) les informations financières de son usager, lui permettant de faire des simulations d'achat et de vente des actions. Il affiche également les prix courants des actions et les nouvelles informations du marché financier. Le même agent permet également d'accéder aux rapports produits par les deux agents de tâches. Ces deux agents fournissent d'une part, une intégration graphique des prix et des nouvelles concernant les actions et, d'autre part, une analyse fondamentale des actions en tenant compte de leurs historiques. Les agents d'informations accèdent à différentes sources d'informations, comme les pages Web, les nouvelles informations, les rapports financiers électroniques du FMI ainsi que d'autres rapports sous un format texte. "Warren" n'est qu'un exemple ; il existe actuellement plusieurs autres systèmes qui touchent à ce genre d'application. Parmi ces applications, nous pouvons citer :

❖ **Infosleuth** : C'est un système multi-agents pour la recherche coopérative d'informations dans des bases de données distribuées. Ce système a été appliqué aux domaines médicaux [Nodin et al., 1998].

❖ **NetSA (pour "Networked Software Agents")**: C'est un système proche d'Infosleuth et est dédié aux environnements riches en informations [Côté et Troudi, 1998].

❖ **UMDL** : C'est un système d'informations coopératif pour la recherche des documents dans une librairie digitale [Vidal et al., 1998].

À titre d'exemple, le système NETSA est un système multi-agents coopératif, développé à l'université Laval [Côté et Troudi, 1998] et destiné aux environnements riches en informations. Ce système comporte plusieurs types d'agents:

- Un agent *utilisateur* en charge de la cueillette et du filtrage des informations provenant et allant vers l'utilisateur;
- Un agent *courtier* servant de répertoire pour les agents qui évoluent au sein de NETSA;
- Des agents *ressources* reliés chacun à une ressource d'informations et pouvant rapatrier et mettre à jour les données;
- Un agent *d'exécution* en charge de la décomposition des tâches et du suivi du déroulement d'exécution des différentes sous-tâches;
- Un agent *ontologie* en charge du maintien de la cohérence des concepts utilisés par les agents.

7. Quelques Plates-formes de développement des SMA :

Des plates-formes génériques sont apparues pour le développement de systèmes reposant sur l'utilisation du paradigme agent. Après avoir examiné plusieurs de ces plates-formes, très peu apparaissent comme pouvant servir réellement à la conception de tout type d'application. En effet, la plupart d'entre eux s'adapte souvent à la conception d'un type d'applications ou d'une classe d'applications mais rarement plus. Il apparaît de plus que la définition d'un agent est beaucoup trop vague pour qu'un modèle générique et universel puisse exister.

Parmi les plates-formes les plus connues on peut citer [Taghezout, 2010]:

- *SWARM* [Burkhart, 1994][Daniels, 1999] [Daniels, 2000],
- *Voyager* [<http://www.objectspace.com/products/voyager/>],
- *DIMA* [Guessoum, 1996], *MADKIT* [Madkit, 2001],
- *Zeus* [Azvine et al., 2000],
- *JADE* [Bellifemine&al., 1999], [Bellifemine& al., 2000], [Bellifemine& al., 2005], [Bellifemine& al., 2008].

Aujourd'hui, l'intérêt d'une méthodologie de conception et d'implémentation de systèmes multi-agents n'est plus à démontrer. De nombreux travaux de recherche continuent à explorer les possibilités que fournirait une méthodologie telle que Aalaadin, ADLEFE, ADEPT, et d'autres. Toutefois, la plupart de ces propositions se base sur une architecture d'agent purement conceptuelle, car l'objectif visé n'est pas de déduire une solution opérationnelle à partir d'une spécification du problème posé.

IV- Conclusion : partie I.

Nous avons souligné, au cours de cette première partie du chapitre, les notions de bases relatives aux agents y compris leurs types et architectures. Ensuite, nous avons montré l'importance croissante apportée aux nouvelles approches à base d'agents et de systèmes multi-agents.

Des exemples relatifs aux SMA dans le monde réel ont été présentés ainsi que les plates-formes les plus importantes permettant d'implémenter les SMA.

Nous avons vu aussi, que la conception de tels systèmes exploitant ces nouvelles approches doit se baser sur les méthodes de conception et d'évaluation adaptées (considérant les notions d'autonomie et d'adaptation).

La deuxième partie de ce chapitre sera consacré aux concepts de bases liés à la coordination au sein des SMA.

PARTIE II :
COORDINATION AU SEIN DES SMA.

Coordination entre agents

1. Introduction :

La coordination est l'un des modes d'interaction les plus importants au sein de n'importe quelle organisation sociale. En effet, dans la vie quotidienne, nombreux sont les exemples où la coordination est primordiale pour garantir un minimum de cohérence ; à ce titre, nous pouvons citer deux personnes qui doivent se coordonner pour parler à tour de rôle ou bien nous pouvons citer l'exemple le plus célèbre qui est celui des déménageurs qui doivent coordonner leurs actions pour s'entre-aider à porter un objet lourd.

La coordination au sein des systèmes sociaux artificiels, a toujours sollicité énormément de réflexion. En effet, ce problème a déjà été posé en IA, puis dans les systèmes distribués et a pris une ampleur beaucoup plus importante au sein des SMA.

Ce problème consiste à organiser la résolution d'un problème, de telle sorte que les interactions nuisibles soient évitées ou que les interactions bénéfiques soient exploitées.

Dans cette deuxième partie du chapitre, nous allons donner une description globale de la coordination en commençant d'abord par définir la problématique, ensuite nous parcourrons quelques travaux déjà réalisés dans le domaine des SMA, et enfin, nous présenterons les types de coordination.

2. Définitions :

La **coordination** est définie par Thomas Malone et Krevin Crowston [**Malone et Crowston, 1990**] comme l'acte de gérer les interdépendances des différentes activités exécutées pendant la réalisation d'un but. Les interdépendances regroupent les pré-requis (le résultat d'une activité qui est nécessaire à une autre activité), le partage de ressources et la simultanéité (il existe une synchronisation entre l'exécution des activités). Suivant cette définition, la coordination recouvre les indices de coopération se rapportant au partage des ressources, à la coordination des actions et au parallélisme des actions.

Malone [1997] note que deux des composantes fondamentales de la coordination entre agents sont l'allocation de ressources rares et la communication de résultats intermédiaires. Dans ce contexte, les agents doivent être capables de communiquer entre eux de façon à pouvoir échanger les résultats intermédiaires.

Pour l'allocation des ressources partagées, les agents doivent être capables de faire des transferts de ressources. Ceci peut d'ailleurs imposer certains comportements à des agents particuliers.

Malone conclut qu'il peut être utile de distinguer les liens de contrôle comme une catégorie spéciale de liens de communication, par lesquels certains agents transmettent des instructions que d'autres vont suivre.

En étudiant les communautés humaines, **Mintzberg [1999]** a identifié trois processus fondamentaux de coordination :

- *ajustement mutuel* ;
- *supervision directe* ;
- *coordination par standardisation*.

L'*ajustement mutuel* est la forme de coordination la plus simple qui se produit quand deux ou plusieurs agents s'accordent pour partager des ressources en vue d'atteindre un but commun. Habituellement, les agents doivent échanger de nombreuses informations et faire plusieurs ajustements à leurs propres comportements en tenant compte des comportements

des autres agents. Dans cette forme de coordination aucun agent n'a un contrôle sur les autres agents et le processus de décision est conjoint.

La *supervision directe* apparaît quand un ou plusieurs agents ont déjà établi une relation dans laquelle un des agents a un contrôle sur les autres. Cette relation est habituellement établie par ajustement mutuel, comme par exemple dans le cas d'un employé ou d'un sous-contractant qui accepte de suivre les instructions du superviseur. Dans cette forme de coordination, l'agent superviseur contrôle l'utilisation des ressources partagées (comme par exemple les ressources humaines, le temps de calcul ou l'argent) par les agents subordonnés. Il peut aussi imposer certains comportements.

Dans les cas de *coordination par standardisation* le superviseur coordonne les activités en établissant des procédures que doivent suivre les subordonnés dans des situations identifiées. On trouve par exemple de telles procédures dans les entreprises, mais aussi dans les systèmes informatiques.

Malone [1997] suggère qu'en utilisant ces processus de coordination fondamentaux "à la Mintzberg", il est possible de construire des systèmes de coordination sophistiqués dont les plus répandus sont les hiérarchies et les marchés basés respectivement sur les processus de supervision directe et d'ajustement mutuel. Le travail en petit groupe se prête bien à l'ajustement mutuel. Par contre, dès que la taille du groupe grandit et que le nombre de tâches augmente, le nombre de liens et la quantité d'informations échangées peuvent devenir rapidement un handicap sérieux.

Un groupe important peut être partagé efficacement en sous-groupes de manière à ce que la plupart des interactions s'effectue dans les sous-groupes et les quelques interactions nécessaires entre les sous-groupes soient prises en charge par les superviseurs de ces sous-groupes.

Les sous-groupes peuvent être coordonnés par contrôle hiérarchique ou par ajustement mutuel suivant les caractéristiques du domaine d'application et des tâches à accomplir. Les marchés sont aussi considérés comme une activité d'organisation de groupe basée sur l'ajustement mutuel dans laquelle, chaque agent contrôle des ressources rares (par exemple la main d'œuvre, les matières premières, les marchandises et l'argent). Les agents s'accordent pour partager leurs ressources respectives afin d'atteindre un but commun. Les ressources échangées ont un prix explicite ou implicite. Lorsqu'un contrat est conclu, il y a un accord pour que l'agent contracteur devienne le superviseur de l'agent contractant. De nombreuses organisations mettent en œuvre des processus de coordination mixtes basées sur l'ajustement mutuel, la supervision directe et la standardisation. Ces cadres organisationnels peuvent être des sources d'inspiration quand on doit établir une structure organisationnelle pour un SMA.

Bien entendu, la coordination est une question centrale pour les SMA et la résolution de systèmes distribués. En effet, sans coordination, un groupe d'agents peut dégénérer rapidement en une collection chaotique d'individus. On pourrait penser que la façon la plus simple de s'assurer un comportement cohérent du groupe d'agents serait de le faire par un agent centralisateur qu'on appellera tout au long de ce travail *un agent coordinateur*, qui détiendrait des informations de haut niveau sur ces agents. Ainsi, l'agent coordinateur pourrait créer des plans d'action et assigner les tâches aux divers agents du groupe. Cette approche est très difficile à mettre en œuvre dans des applications réalistes en raison de la difficulté à réaliser un tel agent coordinateur qui puisse tenir compte des buts, des connaissances et des activités de chaque agent : la charge en communication serait énorme, sans compter qu'on

perdrait les avantages d'un SMA composé d'agents autonomes. Le contrôle et les informations doivent alors être distribués parmi les agents.

Pour résoudre ce problème, **Jennings [1995]** propose un modèle intéressant qui se base sur l'hypothèse de la centralité des engagements et des conventions. Les engagements sont vus comme des promesses en vue de réaliser certaines actions, alors que les conventions constituent le moyen de faire le suivi de ces engagements dans des circonstances changeantes. Grâce aux engagements des autres agents, un agent peut prédire les actions qu'il est susceptible d'effectuer et déterminer ainsi les interdépendances avec ses propres actions. Mais comme le monde extérieur et les croyances des agents évoluent constamment, un agent doit aussi posséder un moyen pour déterminer si les engagements existants sont encore valides. Les conventions offrent un tel mécanisme en définissant les conditions sous lesquelles les engagements doivent être réévalués et en spécifiant les actions à entreprendre dans de telles circonstances.

Du point de vue du concepteur d'un SMA diverses questions doivent être traitées comme notamment **[Ferber, 1995]** : Avec quels agents un agent doit-il coordonner ses actions? Quand et où ces actions de coordination doivent-elles être accomplies? Comment détecter et traiter les interactions entre actions (conflits et renforcement)? Détecter les relations existant entre des actions est une activité nécessaire lorsqu'on veut coordonner ces actions.

Von Martial [1990] de son côté, a identifié deux grandes catégories de relations pouvant exister entre les actions accomplies simultanément par plusieurs agents : les *relations négatives* et les *relations positives*. Les relations négatives (ou conflictuelles) sont celles qui gênent ou empêchent plusieurs actions de s'accomplir simultanément et sont dues en général à des incompatibilités de buts ou des conflits de ressources. Par exemple, dans une vente aux enchères, agent X et agent Y veulent acquérir un même meuble M. Les relations positives (ou synergiques) sont celles qui permettent aux actions de bénéficier les unes des autres. Ainsi, la réalisation d'une action à accomplir par l'agent Y réalise du même coup une action $x = b$ que devait accomplir l'agent X ou favorise la réalisation d'une action C par l'agent Z. Par exemple X, Y et Z sont dans une pièce dont les fenêtres sont fermées et les stores baissés. X a chaud et Z aimerait avoir de la lumière. Y monte les stores et ouvre une fenêtre.

3. Problématique de la coordination :

A partir des travaux menés par **El Fallah-Seghrouchni [2001]**, où un certain nombre de problèmes liés à la coordination au sein des SMA est présenté, nous nous sommes inspirés de ce travail pour mieux appréhender l'utilité et la notion même de la coordination à travers ces différentes problématiques.

Pour **[BON 88]**, les SMA nous proposent une approche originale pour la conception des systèmes dits intelligents et coopératifs. Comme nous l'avons déjà cité, les SMA se composent d'agents plus ou moins autonomes et hétérogènes, évoluant dans un environnement partagé et dynamique. Ce type de conception induit de nouveaux problèmes liés directement à la vision partielle des agents, c'est-à-dire la poursuite d'objectifs locaux et le chevauchement des activités des agents. Raison pour laquelle, si nous voulons que la coopération entre agents ait un comportement global cohérent, il faut opter pour des mécanismes élaborés de coordination, afin d'éviter tout conflit potentiel et favoriser la synergie des activités des agents.

Au sein des SMA, la coordination fait souvent référence au processus qui contrôle la prise de décision et guide le comportement global inhérent à l'exécution d'une collection d'agents. Dans le même ordre d'idée, on peut la considérer comme *un problème de prise de décision distribuée sous incertitude*, ou bien comme *un ensemble de questions relatives à la connaissance et à l'action*, ou encore comme dans notre cas *un problème de conception*, le cas par exemple de la coordination des frameworks dans les organisations Homme/Machine [GAS 92].

Le processus de coordination, engage à un instant donné, un ensemble d'agents du SMA et les aide à atteindre un état global cohérent où les états locaux des différents agents sont compatibles en termes d'objectifs, d'actions, d'utilisation de ressources,...etc. ainsi, tout au long de ce processus, les agents vont œuvrer pour éviter des situations conflictuelles (antagonisme, famine, inter-blocage...etc.) pouvant être engendrées par une mauvaise utilisation des ressources partagées, par l'existence de connaissances et de visions incompatibles...etc. Au-delà de la résolution ou de l'évitement d'éventuels conflits, la coordination présente un intérêt majeur qui est celui de l'évitement de toutes les situations d'interactions défavorables ou négatives et la favorisation des situations d'interactions favorables ou positives et ce, en poussant les agents à une parfaite synergie tout en leur permettant de bénéficier de leurs capacités respectives et de tirer profit des actions des uns et des autres pour avoir un comportement global des plus cohérents (l'union fait la force !!).

A travers ce que nous venons de décrire comme problématiques, nous pouvons déduire l'importance de la coordination dans le domaine des SMA, mais plus important encore est que cela nous permet de faire un tour d'horizon sur les solutions, les techniques et les implémentations qui ont été explorées pour résoudre le problème de la coordination. Notons tout de même que dans ce chapitre et pour la suite de notre travail, nous n'allons présenter que la coordination comme un acte *conscient et délibéré* applicable à des SMA composés d'agents *cognitifs* et/ou *rationnels*, par opposition à la coordination *réactive* [FER 95], qui repose essentiellement sur le mécanisme de perception/réaction et suppose que le contexte est très évolutif rendant ainsi l'anticipation sur la suite des actions difficilement réalisable.

4. Objectifs et motivations de la coordination dans les SMA :

a. Pluralisme :

L'un des premiers objectifs de la coordination a été l'incorporation et l'intégration de points de vue conflictuels ou divergents lors de la résolution distribuée des problèmes. Les initiateurs de ce concept sont Kornfeld et Hewitt [KOR 81], dans leur travail s'intitulant *Métaphore de la communauté scientifique* où la concurrence et la divergence de points de vue ont été la clé de son succès. Dans le même travail, ils ont développé un langage appelé **ETHER** qui offre un grand degré de parallélisme aux systèmes de résolution de problèmes.

En 1991, Carl Hewitt [HEW 91], a proposé un cadre conceptuel permettant le raisonnement sur des processus ayant des points de vue conflictuels.

Dans cette perspective Bond et Lessser [BON 88], définissent l'objectif de la coordination comme étant le processus qui *permet d'aligner des activités de différentes perspectives*. Aussi, elle permet de résoudre les problèmes suivants :

- Comment garantir que les agents agissent de manière cohérente lors d'une prise de décision ou du choix des actions futures, ainsi que lors de la recherche d'un

compromis entre les effets locaux et globaux des décisions afin d'éviter les interactions négatives ?

- Comment permettre aux agents de raisonner sur les actions, les plans et les connaissances des autres agents pour se coordonner avec eux ? et de quelle façon peuvent-ils percevoir l'état du processus de coordination, c'est-à-dire l'initialisation ou l'achèvement de ce dernier ?

- Comment et à quel moment pouvons-nous reconnaître une situation conflictuelle pour pouvoir réconcilier les points de divergences à travers un ensemble d'agents en leurs permettant de coordonner leurs actions ?

b. Recherche de la cohérence globale du système :

La deuxième motivation pour les travaux sur la coordination a été la recherche de la cohérence globale du SMA comme entité, en dépit de l'absence d'un contrôle centralisé ayant un point de vue global. Les Gasser [GAS 92], considère que la coordination est un problème de contrôle distribué où on est appelé à gérer les décisions de contrôle et les décisions relatives aux choix d'actions futures des agents lors de la résolution d'un problème. Si on se réfère à cette définition, nous pouvons affirmer que le degré de coordination dépend fortement de la qualité de gestion des deux types de décisions précédemment cités par l'auteur. Cette gestion pour quelle puisse être de qualité, on est dans l'obligation de connaître les alternatives, les critères de décision et enfin, connaître les procédures qui sont appliquées aussi bien aux critères de décisions, qu'aux alternatives.

Dans cette perspective, et en fonction de ces dimensions, plusieurs variantes de systèmes de contrôle ont été définies ; ce que nous retiendrons c'est que deux systèmes de contrôle sont équivalents s'ils mènent à la même séquence de choix.

Plusieurs travaux se sont également intéressés à la coordination dans plusieurs domaines, mais il y a un travail qui a tout particulièrement attiré notre attention, où on retrouve la définition proposée par Sascha Ossowski [OSS 99] qui est intéressante. Dans ce travail on distingue deux grandes familles de coordination : la coordination centralisée et la coordination distribuée ou décentralisée.

La *coordination centralisée* repose sur l'existence d'un agent central ou coordinateur. La résolution de la coordination peut se ramener à la résolution d'un problème d'optimisation ou à celle d'un problème de satisfaction de contraintes.

Dans le cas où un problème de coordination est exprimé sous la forme d'un problème d'optimisation, il peut être défini comme un triplé (V, D, U) où :

- $V = \{v_1, \dots, v_n\}$ est l'ensemble fini de variables de décision représentant les objets de la coordination ;

- $D = \{D_1, \dots, D_n\}$ est l'ensemble des domaines respectifs des variables v_1, \dots, v_n ;

- $U: X \rightarrow \mathbb{R}$ est une fonction d'utilité où $X = D_1 \times \dots \times D_n$ désigne l'ensemble des instanciations possibles représentant l'espace des décisions, c'est-à-dire l'ensemble de toutes les combinaisons possibles des objets de coordination.

Formulé ainsi, résoudre un problème de coordination revient à trouver une *instanciation optimale* x , c'est-à-dire $\forall y \in X, U(y) \leq U(x)$.

Dans le cas où un problème de coordination est exprimé sous-forme d'un problème de satisfaction de contraintes, la fonction d'utilité est substituée par un ensemble de contraintes représentant les dépendances entre les objets de la coordination. Le problème de la

coordination est ainsi défini comme un triplé (V, D, C) où V et D sont définis comme dans le modèle précédent (*i.e.* résolution d'un problème d'optimisation), alors que $C = \{C_i\}$ représente l'ensemble des contraintes de dépendance. Chaque contrainte C_i peut être considérée comme un sous-ensemble de $X =_{D_1} x_1 \dots x_{D_n}$.

Exprimé ainsi, résoudre un problème de coordination consiste à trouver une *instanciation consistante* x , qui satisfait toutes les contraintes de dépendances.

La *coordination distribuée* est particulièrement adaptée dans le cas d'agents rationnels possédant éventuellement des intérêts conflictuels, c'est-à-dire que chaque agent développe une stratégie propre pour maximiser sa fonction d'utilité. La coordination cherche à trouver un équilibre ou un compromis entre les intérêts des agents impliqués dans le processus de coordination. Notons cependant qu'une telle coordination requiert souvent un mécanisme de négociation ou de médiation entre agents [MÛL 96 ; ZLO 96].

5. Dimensions de la coordination :

Appréhender un modèle de coordination dans sa globalité nécessite d'intervenir selon trois dimensions fortement corrélées [El Fallah, 2001]:

- 1) la structure de l'agent qui détermine le sujet de la coordination : *que coordonner ?*
- 2) le modèle de spécification du problème de coordination : *à travers quelle représentation ?*
- 3) les techniques et mécanismes de coordination : *comment ?*

a. Structure d'un agent :

Situé dans un environnement, un agent est amené à interagir avec d'autres agents et à réguler son comportement en conséquence. Le comportement d'un agent dépend de sa structure interne (croyances, intentions, connaissances, etc.) et de sa perception externe au sens large (perception, communication, interaction).

La structure interne d'un agent est un ensemble de composants internes généralement inaccessibles aux autres agents (Figure 1). Elle définit son architecture qui dépend du type d'agent considéré. Ainsi, on trouvera le concept des états mentaux ou attitudes intentionnelles [BRA 87 ; RAO 91] dans le cas d'agents intentionnels, les connaissances et les mécanismes de raisonnement, de délibération et de prise de décision dans le cas des agents cognitifs ainsi que des engagements dans le cas d'agents sociaux [CAS 98].

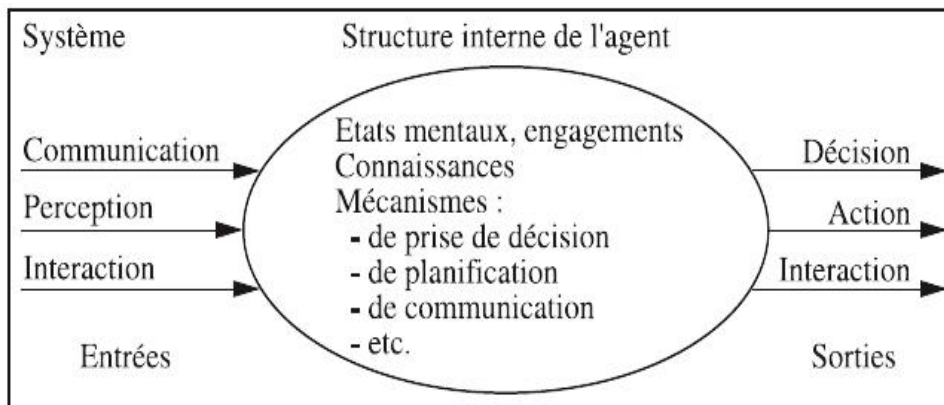


Figure 1 : Paramètres de la coordination [El Fallah, 2001].

Pour un observateur externe du SMA, un agent peut être perçu comme une entité qui subit l'influence de son environnement et réagit en conséquence. En effet, l'agent dispose (outre ses connaissances propres, telles que son expertise d'un domaine donné) de connaissances sur son environnement incluant les autres agents. Ces connaissances sont dynamiquement mises à jour en fonction de la perception qu'il a de son environnement et des communications ou interactions qu'il opère avec les autres agents. Le comportement d'un agent est donc perceptible en termes d'actions, de plans, de décisions ou de buts déclarés, d'interactions, etc., qui découlent de sa structure et ses mécanismes internes.

Coordonner le comportement d'un agent avec ceux d'autres agents revient donc à agir sur l'une de ses composantes internes pour influencer ses actions, ses plans, ses décisions, etc. Ce qui répond à la question : « Que faut-il coordonner ? »

Dans le scénario des agents déménageurs, chaque agent peut être amené à modifier ses intentions qui donnent lieu à des actions (par exemple : soulever un objet, le porter, le charger, s'engager dans un couloir étroit en portant un objet encombrant, etc.) ; ou encore à ses décisions et plans (décider de s'engager dans le couloir, synchroniser son plan avec celui d'un autre, etc.).

b. Modèle de spécification du problème de coordination :

La spécification d'un modèle de coordination nécessite la caractérisation des situations de coordination. Il s'agit de représenter les conditions d'une situation qui requiert la coordination, par exemple une situation critique (deux actions qui font appel à une même ressource à un seul point d'accès). Après la caractérisation des situations de coordination positives ou négatives, l'agent doit être en mesure de les détecter automatiquement et de déterminer le *sujet* de la coordination.

Le modèle de spécification du problème de coordination répond à la question : « *A travers quelle représentation ?* » Cette représentation dépend du sujet de la coordination. Par exemple, s'il s'agit d'une coordination de plans, le modèle de spécification sera fonction du formalisme de représentation de plans. Si, par contre, les agents doivent coordonner leurs décisions, le modèle de spécification dépendra de la représentation des décisions, des buts, etc.

Dans la littérature consacrée aux SMA, les modèles utilisés sont souvent issus de l'intelligence artificielle ou des modèles des systèmes concurrents. Ainsi, sont utilisés des modèles basés sur des formalismes logiques (logique des prédicats, logique modale, logique déontique, logique temporelle et leurs variantes, etc.), la théorie des graphes de buts et/ou l'algèbre des processus, des formalismes pour la concurrence tels que les réseaux de Pétri, etc.

c. Mécanismes de coordination : modèles orientés tâches Vs modèles orientés agents :

Les mécanismes de coordination correspondent aux techniques et moyens mis en œuvre pour coordonner les comportements d'agents à différents niveaux d'abstraction. Ils tentent de répondre à la question : « Comment coordonner ? » et dépendent du modèle de spécification adopté pour un sujet de coordination donné. On distingue deux grandes classes de modèles de coordination : les modèles orientés tâches et ceux orientés agents.

c1. Modèles de coordination orientés tâches :

Il s'agit de modèles applicables dans un contexte de résolution distribuée de problèmes (*Distributed Problem Solving*). Dans ces modèles de coordination, on présuppose l'existence

d'un but global partagé par un groupe d'agents au sein d'une organisation. C'est le cas par exemple de nos agents déménageurs dans une situation coopérative où le gestionnaire a déjà distribué la tâche en sous-tâches, et alloué chacune d'elles à un agent ou un groupe d'agents :

-le but global des agents est d'accomplir la tâche T de déménagement allouée par le gestionnaire de l'agence ;

- le but individuel de chaque agent A_j est d'accomplir une sous-tâche T_j selon l'allocation du gestionnaire (par exemple : déménager une pièce).

La coordination est nécessaire pour optimiser l'efficacité globale du groupe d'agents et non pas d'un point de vue individuel des agents. Les agents sont typiquement dans une situation de coopération. Des interactions apparaissent au fur et à mesure que les agents essaient de réaliser leur tâche individuelle, par exemple :

S'entraider pour porter un objet lourd, ou se synchroniser pour ne pas emprunter un couloir étroit simultanément, etc.

c2. Modèles de coordination orientés agents

Il s'agit de modèles applicables dans le contexte d'agents (semi-) autonomes ne partageant pas *forcément* un but global sinon la cohérence et la viabilité du système. Dans ce cas de figure, les agents, en tant qu'entités individuelles, cherchent à optimiser leurs propres objectifs (par exemple : gagner le maximum dans la situation de compétition des agents déménageurs). Même en situation de compétition, la coordination est *nécessaire* pour accomplir des tâches qu'un agent ne peut réaliser seul, ce qui lui permet d'augmenter ses gains.

A nouveau, dans le cas de modèles orientés agents, on peut distinguer deux grandes tendances : la coordination d'agents pris individuellement (par exemple : coordonner les plans individuels des agents déménageurs A_1 et A_2) et la coordination d'un groupe d'agents (par exemple : A_1 et A_2 élaborent un seul plan appelé Plan-Joint [GRO 96 ; HUB 96 ; KIN 92], Agency, Teamwork, etc.) qui vont mener des actions jointes à partir de connaissances et/ou croyances collectives (connaissances communes, attitudes intentionnelles jointes, croyances mutuelles, activités de groupe, etc.).

6. Approches de coordination d'agents :

Cette section présente quelques approches des plus significatives dans le domaine de la coordination des SMA, aussi bien celles orientées tâches que celles orientées agents. La section 7 sera entièrement consacrée aux travaux concernant la coordination par planification qui constitue une approche majeure de la coordination au sein des SMA.

a. Coordination basée sur la résolution distribuée de problèmes :

Un des premiers modèles de coordination orientés tâches a été proposé par Lesser dans [LES 81]. Ce modèle appelé FA/C (*Functionally Accurate Model*) (modèle fonctionnel exact) peut être caractérisé par la distribution du contrôle, et de ce fait, il a constitué une approche novatrice au début des travaux sur la coordination des SMA. Son intérêt réside dans le fait que les agents n'ont pas besoin d'avoir localement toutes les informations pour résoudre leur problème, mais doivent interagir pour acquérir dynamiquement les informations requises et se coordonner en fonction de leurs résultats partiels. Il s'agit de définir des stratégies de coordination basées sur un mécanisme de contrôle distribué, qui suppose une interdépendance entre les activités des agents. Ces derniers échangent des informations de haut niveau (dites *méta*) tels que leurs choix, leurs préférences, leurs priorités, leurs buts, etc.

La critique majeure adressée à ce modèle est que le niveau *méta* est défini de manière statique, ce qui rend l'adaptation du mécanisme de coordination difficile.

Depuis ce modèle, plusieurs tentatives d'amélioration du mécanisme de distribution du contrôle ont été proposées [LES 91].

Le modèle proposé par E. H. Durfee et V. R. Lesser [DUR 91] appelé planification globale partielle (PGP : *Partial Global Planning*) se caractérise par sa flexibilité et sa dynamique. Contrairement au modèle précédent (FA/C), ici le niveau *méta* est dynamique. Dans ce modèle, les agents coordonnent dynamiquement leurs plans en échangeant des informations pertinentes pour l'exécution de ces plans. Les informations pertinentes sont définies à trois niveaux de représentation des plans ; et chaque agent gère les trois niveaux comme suit :

- *plan local* : chaque agent développe un plan local détaillé dont il contrôle l'exécution ;
- *plan nodal* : ce plan est une abstraction du plan local qui inclut des informations pertinentes destinées à être communiquées aux autres agents ;
- *plan global partiel* : ce plan contient les buts des autres agents, dans ce sens il est global. Par contre, il est partiel car il ne concerne qu'un sous-ensemble d'agents restreint à ceux susceptibles d'avoir des interactions avec l'agent propriétaire du plan.

Les interactions concernent la communication des plans et des buts à un niveau d'abstraction approprié. Les agents récepteurs infèrent les buts et les actions des agents émetteurs et décident des actions à planifier en leur faveur. Ainsi, cet échange de plans ou buts améliore les capacités de prédiction d'un agent relativement aux plans des autres agents en vue d'une coordination plus efficace.

Ce modèle a été appliqué au problème de la surveillance de la circulation de véhicules dans un espace divisé en zones (DVMT : *Distributed Vehicle Monitoring Testbed*). Chaque agent est responsable d'une zone et les interactions en terme de surveillance portent sur les recouvrements de ces zones. On remarquera ici que la distribution du problème porte sur l'espace à surveiller et que les tâches allouées aux agents sont similaires. Dans ce cas particulier, les interactions sont limitées aux agents responsables des régions voisines selon une topologie spatiale.

Decker a ensuite généralisé ce modèle [DEC 92] en proposant le modèle GPGP (*Generalized Partial Global Planning*), où la coordination doit respecter des contraintes d'exécution en temps réel. Par conséquent, la structure des buts a été augmentée et affinée à travers un arbre de buts étendu, organisant les buts de manière hiérarchique et incluant diverses informations telles que : l'ordre des buts, leurs durées, leurs priorités, les ressources qui leur sont nécessaires, le plan solution en cours, les contraintes imposées aux solutions attendues, etc.

Dans cette généralisation (GPGP), le mécanisme de coordination se base sur trois relations de dépendance :

- les relations de coordination entre actions : inhiber, annuler, causer, faciliter, contraindre, etc. ;

- les relations entre les graphes de buts : recouvrement, extension, suffisance, concurrence, etc. ;

- les relations temporelles : dures (*hard*), négociables, légères (*soft*) pour représenter des préférences locales des agents.

Ainsi, après avoir construit un plan local des activités à moyen terme, et après avoir déterminé les relations de dépendance entre ces activités, chaque agent communique avec ses voisins et met à jour sa représentation individuelle à partir des échanges. L'intérêt principal de ce modèle est la possibilité *de prédire* les sujets de coordination et de les traiter en temps utile voire d'anticiper les traitements [DEC 93 ; DEC 95].

b. Coordination basée sur les structures organisationnelles

Les structures organisationnelles jouent un rôle important pour définir et guider les rôles des agents et par conséquent leurs comportements au sein d'une organisation. De manière externe, une organisation peut être vue comme une délimitation des responsabilités et des actions des agents. En contrepartie, la structure organisationnelle peut constituer un moyen de pallier les limitations des agents en terme de compétences ou d'aptitudes à réaliser des tâches complexes.

Plusieurs types d'organisation existent dans la littérature consacrée aux SMA. Une caractéristique commune aux organisations est que l'on peut les voir comme des distributions (statiques ou dynamiques) selon trois dimensions pouvant être combinées :

- *la distribution fonctionnelle* : elle consiste à répartir les spécialisations, les rôles ou les fonctions des agents au sein d'une organisation et ce, en tenant compte de leurs aptitudes, de la nature des tâches qui leur incombent et des interactions possibles entre eux (par exemple, une spécialisation des agents déménageurs par type d'objet à déménager : porteurs d'objets lourds, porteurs d'objets fragiles, etc., ou encore une hiérarchie des agents : gestionnaire et agents déménageurs sous ses ordres, etc.) ;

- *la distribution spatiale* : une telle distribution concerne l'affectation des agents à des régions bien délimitées suite au découpage de l'environnement dans lequel ils évoluent (par exemple : affectation des agents déménageurs aux pièces de la maison). Cette répartition de l'espace n'est pas nécessairement une partition car les recouvrements peuvent être introduits pour améliorer la robustesse des traitements, notamment dans le cas des transitions d'une région à une autre voisine (par exemple : dans l'exemple DVMT, les agents ont des rôles identiques mais sont affectés à des régions différentes de l'espace qui le recouvrent) ;

- *la distribution temporelle* : ce type de distribution relève des organisations dynamiques. C'est le cas des structures où les agents peuvent évoluer, apprendre, acquérir de nouvelles connaissances. Par conséquent, les processus inhérents à leurs exécutions, leurs communications ainsi que leurs interactions peuvent changer et influencer la structure organisationnelle elle-même (par exemple : coalitions ponctuelles des agents déménageurs pour porter des objets lourds).

Selon [GAS 84], du point de vue de la coordination, et à un plus haut niveau d'abstraction, une organisation peut être vue comme un mécanisme de coordination qui se situe au niveau global, c'est-à-dire au niveau du réseau d'agents. Cette organisation peut être soit *statique* ou bien *dynamique*.

b1. Organisations statiques :

Lorsqu'elle est définie statiquement, une structure organisationnelle est figée *a priori*. Lors de la conception du SMA, les comportements des agents sont à un certain niveau d'abstraction coordonnés et réglementés par la structure qui les réunit. Dans ce type de

structure, on retrouve les techniques de *coordination par réglementation* qui consistent à introduire des règles de comportement que les agents doivent respecter afin d'éviter des conflits potentiels. De ce fait, plusieurs conflits sont résolus *a priori*.

L'intérêt généralement escompté de l'emploi de telles techniques est la réduction des coûts de communication entre agents et l'élimination de conflits potentiels dès la conception. En revanche, les lois pèsent sur l'*autonomie* des agents dont il faut *délimiter* le comportement.

b2. Organisations dynamiques :

Un problème de coordination dans le cas d'une organisation dynamique peut être formulé selon trois niveaux [SYC 89] :

1) au niveau de la conception de l'organisation : comment organiser et répartir les tâches dynamiquement pour atteindre les objectifs du système pouvant eux-mêmes varier dans le temps ?

2) au niveau de la dynamique de l'organisation : comment sauvegarder les compétences et les connaissances du système quand l'organisation change ?

3) au niveau cognitif : comment représenter les connaissances organisationnelles et quel niveau de partage (connaissances, contrôle, etc.) est requis entre agents ?

Différentes approches pour la dynamique des organisations ont été explorées. On peut citer à titre d'exemple les agents flexibles définis par Toru Ishida [ISH 92], qui peuvent se restructurer en fonction des évolutions environnantes (autres agents, environnement, etc.).

Un autre domaine très significatif pour les organisations dynamiques est celui de la formation de coalitions dans les SMA que nous allons aborder un peu plus loin. Par définition, les coalitions sont des regroupements ponctuels d'agents compétitifs (auto-intéressés). L'intérêt des coalitions est qu'elles se forment et se dissolvent en fonction des entrées du système (par exemple : les buts nouvellement arrivés, les agents qui intègrent ou quittent le système, etc.) garantissant ainsi une meilleure adaptation aux changements de l'environnement.

c. Coordination basée sur les protocoles de coopération :

A l'origine des langages d'interaction, où se situent les langages de coordination proposés par Gelernter et Carriero [GEL 92]. Ces auteurs les explicitent ainsi : «*Une computation doit communiquer (interagir) avec son environnement ou ne servira à rien. L'environnement peut inclure des humains, d'autres processus, etc. La forme de cette interaction est du domaine de la coordination et le point de mire des langages de coordination*».

Une approche de plus en plus développée repose sur les protocoles de coordination, appelés également protocoles d'interactions. Dans ce cas, la coordination est basée sur la communication entre agents, celle-ci respecte un protocole défini et agréé par les agents. Ces derniers se coordonnent en émettant des requêtes aux autres agents, en leur envoyant des informations, en leur déléguant des tâches, etc.

Généralement, un protocole de coopération décompose une tâche en sous-tâches et les distribue, suivant une stratégie donnée, sur des agents du système. Cette stratégie doit respecter les critères suivants [DUR 87] :

- éviter le recouvrement en ressources critiques ;
- allouer les tâches aux agents en fonction de leurs capacités ;
- tenir informés les agents des différentes allocations de tâches ;

-allouer les responsabilités aux agents de manière couvrante pour assurer la cohérence globale ;

-allouer les tâches interdépendantes aux agents en optimisant leur distance spatiale pour minimiser les coûts de communication ;

- réallouer les tâches s'il est nécessaire de les accomplir de manière urgente.

Un des premiers protocoles de coopération a été proposé par D. Smith. Il est connu sous le nom du protocole du réseau contractuel (*Contract Net protocol*) [SMI 80] et respectant les critères énoncés ci-dessus. Plus récemment, plusieurs protocoles de coopération ont été proposés dans le cadre des langages d'interaction d'agents tels que FIPA-ACL (ACL : *Agent Communication Langage*). D'autres protocoles et plusieurs de leurs variantes ont été proposés dans la littérature SMA en vue de rendre opérationnels des protocoles pour les enchères, les appels d'offre, etc.

d. Modèles de négociation et de prise de décision distribuée

La négociation dans les SMA constitue une étape importante dans un processus de coordination [DAV 83]. La négociation est un processus dynamique qui vise à établir des accords entre agents et à induire des engagements de leur part à court ou à moyen terme.

Les approches fondées sur la négociation concernent souvent, mais pas exclusivement, des agents compétitifs et/ou rationnels (au sens de la rationalité économique). Chaque agent possède une stratégie de prise de décision qu'il utilise pour maximiser sa *fonction d'utilité locale*. Dans le cas où l'agent pourrait définir une stratégie dominante, c'est-à-dire une stratégie gagnante indépendamment des stratégies des autres, il n'y a pas réellement besoin de coordination. Cependant, cette stratégie n'existe jamais puisque les stratégies individuelles peuvent s'influencer mutuellement (si cette stratégie existait, elle serait applicable par tous les agents). L'agent, à travers sa rationalité individuelle, est amené à se coordonner avec les autres au moyen d'un protocole de négociation, afin de converger vers des solutions globales acceptables par tous les participants à la négociation.

Les protocoles proposés dans ce contexte cherchent à satisfaire généralement deux critères : 1) garantir l'efficacité globale sans affaiblir aucune fonction d'utilité locale ; 2) veiller à la stabilité du système, c'est-à-dire assurer un équilibre symétrique entre agents où les gains sont répartis équitablement. Les travaux les plus significatifs du domaine proposent des modèles de négociation inspirés de la théorie des jeux, de la théorie d'aide à la décision ou combinent les deux.

e. Modèles de coordination fondés sur la formation de coalitions :

Les coalitions permettent aux agents de satisfaire des besoins qui requièrent la synergie de compétences des différents agents comme, par exemple, dans le cadre de la résolution de tâches complexes pour lesquelles des agents agissant seuls seraient incapables ou moins efficaces (par exemple : déménagement d'un objet lourd). Une coalition peut être définie comme une organisation à court terme basée sur des engagements spécifiques et contextuels, ce qui permet aux agents de bénéficier de leurs compétences respectives. Dans un contexte économique par exemple, plusieurs compagnies s'allient « *virtuellement* » pour répondre à des offres nécessitant des compétences variées. La formation de coalitions offre plusieurs avantages : 1) le concept d'engagement ponctuel permet aux agents de réagir de façon opportuniste et de réviser dynamiquement leurs intérêts et par conséquent leurs objectifs ; 2)

la formation et la dissolution des coalitions est contexte-dépendante, ce qui permet aux agents d'adapter leur production ; 3) contrairement aux organisations statiques, la formation de coalitions permet d'appréhender de façon plus flexible les environnements ouverts et dynamiques.

Du point de vue des SMA, les recherches sur la formation de coalitions essaient de proposer des protocoles d'automatisation des coalitions qui soient les plus « réalistes » possibles. En effet, la recherche de solutions optimales (généralement mesurées en termes de profit total maximum) nécessite des algorithmes NP-complets ; et à cause de cette complexité, différentes hypothèses simplificatrices sont introduites pour générer des solutions calculables en un temps raisonnable. Une des hypothèses les plus fortes consiste à supposer que les agents sont coopératifs, voire altruistes. Or dans un contexte de rationalité économique, les agents sont libres de se comporter comme ils le désirent, et même d'avoir des objectifs égoïstes comme en commerce électronique.

Le problème de la formation de coalitions consiste à trouver une solution qui soit la plus satisfaisante pour l'ensemble des agents.

Une autre critique qui peut être adressée aux travaux existants sur la formation de coalitions concerne les dépendances entre les coalitions telles que définies dans [KET 94] : dans une formation itérative, un agent ne peut pas prendre en compte la participation future éventuelle de tel ou tel agent. Or, l'intérêt pour une coalition dépend de l'ensemble de ses membres et non seulement de l'augmentation de sa valeur à l'étape en cours. De même, très peu de travaux se sont intéressés aux partitions multiples dans le processus de formation de coalition. Un agent est le plus souvent contraint de n'appartenir qu'à une seule coalition à la fois (partitions primaires). Ce qui simplifie considérablement le problème mais très souvent le dénature [SEN 00].

7. Coordination d'actions par planification multi-agent et distribuée :

Dès le début des années quatre-vingt, plusieurs travaux en coordination ont adopté la planification multi-agent comme technique de coordination des activités des agents [DUR 88, GEO 83]. La plupart de ces travaux se sont focalisés sur la planification [GEO 90] comme stratégie de coopération entre agents cognitifs pouvant assumer une *post-planification*. En effet, souvent les agents génèrent d'abord leurs plans et cherchent à les coordonner ultérieurement en vue de les exécuter. L'obtention d'un plan global cohérent nécessite la détection de sous-buts en interactions positives ou négatives et, ensuite, la résolution des conflits potentiels.

Dans ce cadre, plusieurs approches de coordination par planification multi-agent ont été proposées dans la littérature SMA, nous y ferons un tour d'horizon par la suite.

a. La coordination par synchronisation d'actions :

Cette approche s'apparente aux techniques utilisées dans les systèmes distribués. Il s'agit de spécifier l'enchaînement des actions et de gérer leur parallélisme potentiel. Cette technique n'intègre pas de mécanismes cognitifs et peut être vue comme un support de base à toute autre forme de coordination d'actions. A titre d'exemple, on peut citer les travaux de M. Georgeff [GEO 83]. Celui-ci a introduit un agent *synchroniseur* qui gère les interactions entre plans. Les agents envoient leurs plans au *synchroniseur*. Ce dernier les examine, détecte les régions critiques dans les plans reçus et résout les conflits par l'ajout d'actions de synchronisation (par

exemple, des primitives de synchronisation qui garantissent l'exclusion mutuelle entre les actions conflictuelles).

b. La planification multi-agent comme spécialisation de la résolution distribuée de problème :

Dans [DUR 99], la planification multi-agent est présentée comme une *spécialisation* de la résolution distribuée de problème. En effet, dans cette perspective, chaque agent se charge de la résolution d'un sous-problème qui, dans le cas d'une planification multi-agent, devient un problème de planification : génération d'un sous-plan pour un sous-problème. De nouveaux problèmes sont alors posés aux agents planificateurs (voir ci-dessus : scénario des déménageurs) :

- Comment décomposer un problème en sous-problèmes ? Quelle stratégie le gestionnaire de l'agence doit-il adopter pour décomposer efficacement la tâche T en sous-tâches ?
- A quels agents allouer ces sous-problèmes ? Quelle stratégie le gestionnaire de l'agence doit-il adopter pour allouer efficacement les sous-tâches aux agents déménageurs ?
- Comment générer les sous-plans ? Chaque agent déménageur génère un plan en fonction de ses connaissances (description du monde incluant les connaissances sur les autres agents, l'état courant, l'état final à atteindre, etc.) ;
- Quand échanger les solutions ou les résultats partiels ? Chaque fois qu'un agent détecte (ou prédit) une situation d'interaction positive ou négative, il en informe les agents concernés et entre en phase de coordination ;
- Quand et comment faire la synthèse des plans concurrents ? Pour des environnements dynamiques, il est préférable voire indispensable d'entrelacer synthèse de plans et exécution. En effet, un plan à long terme peut s'avérer obsolète car, contrairement à la planification mono-agent, l'hypothèse du monde clos n'est pas viable dans un contexte SMA.

c. La coordination par planification multi-agent centralisée ou distribuée :

Les agents élaborent un *plan multi-agent* qui spécifie leurs actions futures et leurs interactions. La synthèse des plans exige un raisonnement sur la manière dont les actions des différents agents peuvent interférer et, par conséquent, la représentation d'actions parallèles. L'exécution non contrôlée de plans concurrents peut engendrer des situations de blocage dues par exemple à une utilisation incohérente des ressources (par exemple : le couloir étroit est une ressource critique dans le scénario des agents déménageurs). De ce fait, la synchronisation des plans doit être effectuée à *différents points* de la résolution d'un problème: durant la phase de décomposition du problème [COR 79], lors de la construction des plans en bâtissant des plans hiérarchisés qui interagissent faiblement [COR 79], après la construction des plans [GEO 83 ; KAB 95], en alignant des plans partiels de façon incrémentale [DEC 92 ; DUR 91], ou en introduisant dans le processus de planification un mécanisme de raisonnement sur les interactions et les dépendances possibles entre actions et plans [EPH 95b ; ELF 96b ; ROS 82 ; VMA 92].

d. Planification multi-agent centralisée et plans distribués :

Ici, le *plan multi-agent* est généré de manière centralisée par un agent *coordinateur* et son exécution est distribuée sur les agents. De ce fait, le plan généré est nécessairement un plan partiellement ordonné et les actions parallèles sont exécutées par différents agents de manière concurrente.

Après avoir distribué le plan en sous-plans potentiellement synchronisés, le coordinateur les transmet aux agents exécutants en vue de leur achèvement sous la forme de plusieurs processus concurrents. Dans le scénario des agents déménageurs, cette situation correspondrait au cas où l'agent gestionnaire allouerait à chaque agent un plan correspondant à la sous-tâche qu'il doit effectuer.

Les systèmes proposés en *planification multi-agent centralisée* [GEO 83 ; GEO 88] présentent des options intéressantes pour la résolution des conflits et la convergence vers une solution globale, mais présentent tous les mêmes désavantages liés à la centralisation du contrôle au niveau d'un seul agent, à savoir :

- si le coordinateur est défaillant, le système est incontrôlable ;
- le coordinateur central constitue un goulet d'étranglement au niveau de la communication et du traitement, c'est-à-dire que les différents agents travaillant de façon concurrente dans le réseau doivent envoyer leurs actions au coordinateur afin qu'il résolve les conflits éventuels et gère leurs interactions ;

-la charge de communication est très importante puisque des quantités importantes de données non nécessairement utiles sont envoyées vers le coordinateur en pénalisant le débit utile du réseau.

f. Planification multi-agent distribuée

En *planification multi-agent distribuée*, c'est le processus de la synthèse du plan multi-agent qui est distribué. A nouveau, on peut distinguer deux cas de figure selon que le plan multi-agent est centralisé ou non.

Planification distribuée pour plan centralisé

Dans certains contextes (par exemple celui de la planification pour des problèmes complexes) la synthèse de plans requiert plus d'un agent. Dans ce cas, le plan sera élaboré par plusieurs agents planificateurs. Reprenons l'exemple des agents déménageurs qui doivent planifier la tâche T. Après la décomposition de la tâche en sous-tâches, chaque agent responsable d'une sous-tâche génère le sous-plan correspondant et, ensuite, les agents interagissent pour fusionner les sous-plans en un plan global. L'interaction peut concerner l'échange de sous-plans à synchroniser, des sous-plans partiels à raffiner, inclure un mécanisme de négociation, etc. De nouveau, le coût de communication peut être important et sans garantie de convergence vers un plan global. Toutes les techniques utilisées en résolution distribuée de problèmes restent applicables pour résoudre les conflits éventuels entre agents planificateurs.

Planification distribuée pour plan distribué

Contrairement aux approches précédentes, ici les processus de synthèse et d'exécution d'un plan multi-agent sont distribués.

On distingue à nouveau deux approches : une approche *orientée tâches* et une approche *orientée agents*. La différence concerne l'existence ou non d'un but global.

Dans le cas d'une approche *orientée tâche*, le problème est décomposé en sous-problèmes et chaque agent prend en charge la résolution du sous-problème qui lui a été confié, c'est-à-dire qu'il génère un sous-plan et l'exécute. La synthèse des sous-plans porte sur les *résultats* et non plus sur les plans.

L'approche *orientée agents* (souvent appelée planification distribuée) s'intéresse aux systèmes où plusieurs agents peuvent planifier et exécuter des plans de manière concurrente et autonome [EPH 95a ; GEO 90 ; VMA 92] indépendamment de l'existence d'un plan global ou d'un but global au système. La coordination, ici, consiste à fournir les mécanismes nécessaires pour que l'exécution des plans concurrents soit possible.

Comme les agents planifient dans un environnement partagé, des événements liés à l'exécution concurrente de plans peuvent se produire durant les phases de synthèse ou d'exécution de plans. Par conséquent, la planification multi-agent requiert des modèles d'actions et de plans multi-agents [GEO 90] permettant la gestion de la concurrence potentielle entre les activités des agents. Ceci a conduit à l'élaboration de formalismes offrant une grande puissance d'expression du parallélisme, de concurrence d'actions, d'événements non atomiques, etc., ainsi que l'expression de nouveaux buts tels que les buts de maintenance ou d'ordonnancement d'actions.

Différentes approches ont été proposées en planification distribuée et plus particulièrement en planification *incrémentale*. Celle-ci consiste à considérer un ensemble de plans déjà coordonnés dans lequel on insère un nouveau plan à coordonner avec les plans existants.

g. Processus de coordination d'actions par planification :

Un processus de coordination par planification peut être décrit par le cycle suivant :

- 1) caractérisation des situations d'interaction ;
- 2) détection *statique ou dynamique* des situations d'interaction ;
- 3) résolution des situations d'interaction positives et négatives ;
- 4) modification des plans.

8. Critères d'évaluation des systèmes de coordination :

Les techniques et mécanismes de coordination peuvent être classifiés selon le spectre de l'autonomie, de la flexibilité, du degré de contrôle de la coordination [LES 91]. L'évaluation d'un modèle de coordination se base sur les critères suivants ou une de leurs combinaisons :

- efficacité en termes de solutions atteintes ou non (complétude) ;
- qualité des solutions produites (consistance) et leur opportunité ;
- adaptabilité du système (apprentissage de concepts individuels ou de représentations et modèles des autres, etc.) et gestion de l'incertain ;
- clarté et observabilité du système ;
- fonctionnement en mode dégradé, tolérance aux pannes, etc. ;
- réactivité aux changements, degré et coût des informations échangées, etc. ;
- etc.

Dans DVMT, par exemple, Lesser et son équipe ont conceptualisé le degré de coopération en mesurant la performance comme le nombre total des sources de connaissances activées par tous les nœuds de la collection quand le système atteint sa solution optimale. Plus la collection s'approche de la solution optimale (nombre d'activations connues) moins on gaspille d'effort et plus la performance est meilleure.

9. Conclusion : Partie II

Au cours de cette deuxième partie du chapitre, nous avons pu constater que les travaux développés dans le domaine de la coordination d'agents combinent souvent plusieurs des mécanismes présentés dans ce chapitre. Citons à titre d'exemple les travaux en planification multi-agent qui incluent des mécanismes de prise de décision distribuée et de négociation.

Globalement, la plupart des approches de coordination se basent sur la distribution du contrôle et des données. Cette distribution est inhérente à l'architecture intrinsèque d'un SMA. Le contrôle distribué signifie que chaque agent possède un minimum d'autonomie pour générer de nouvelles actions et décider quels buts il va poursuivre. En revanche, cette distribution induit une dispersion des connaissances et une vue partielle rendant ainsi la cohérence globale difficile à atteindre, d'où la nécessité de la coordination qui reste un mécanisme-clé dans le domaine des SMA. Plusieurs solutions ont été explorées mais il reste beaucoup à faire dans ce domaine.

CHAPITRE IV
CONCEPTION

I. Introduction :

Selon [Friedman, 2005], l'importance d'utiliser les SIAD à base du Web résulte de l'utilisation croissante de l'information disponible qui devrait être identifiée, contrôlée et accédée à distance en utilisant les outils à base du Web pour soutenir la réutilisabilité des modules de décision intégrés. De tels systèmes aident les entreprises globales à gérer et améliorer les processus de décision, le contrôle des processus, le service client, et rendre la nouvelle conception plus flexible.

Dans ce chapitre nous présenterons les différents agents sur lesquels se base la modélisation de notre approche en mettant en évidence le rôle de l'agent central qui est l'agent coordinateur. Ce dernier accomplit une tâche spécifique à laquelle seront associés plusieurs scénarios de coordination. Nous décrirons enfin dans ce chapitre les éléments les plus importants qui caractérisent ces stratégies tels que les temps d'exécution, les temps d'attente et les autres temps d'interaction des agents.

II. Architecture générale du Web-DSS :

Le schéma donné à la figure 1 représente l'architecture générale du système, nous pouvons remarquer que ce système est composé de trois sous-systèmes différents lui permettant d'effectuer une gestion dynamique des pannes qui se produisent au sein de l'atelier de production, ces trois sous-systèmes sont:

- *Atelier de production ;*
- *Poste de contrôle ;*
- *SIAD à base de web.*

Aussi, nous pouvons remarquer qu'entre l'atelier de production et le poste de contrôle nous définissons un échange d'informations basé sur une architecture Client/Serveur, et ce pour répondre efficacement au types de traitements que nous allons effectuer au sein de notre système.

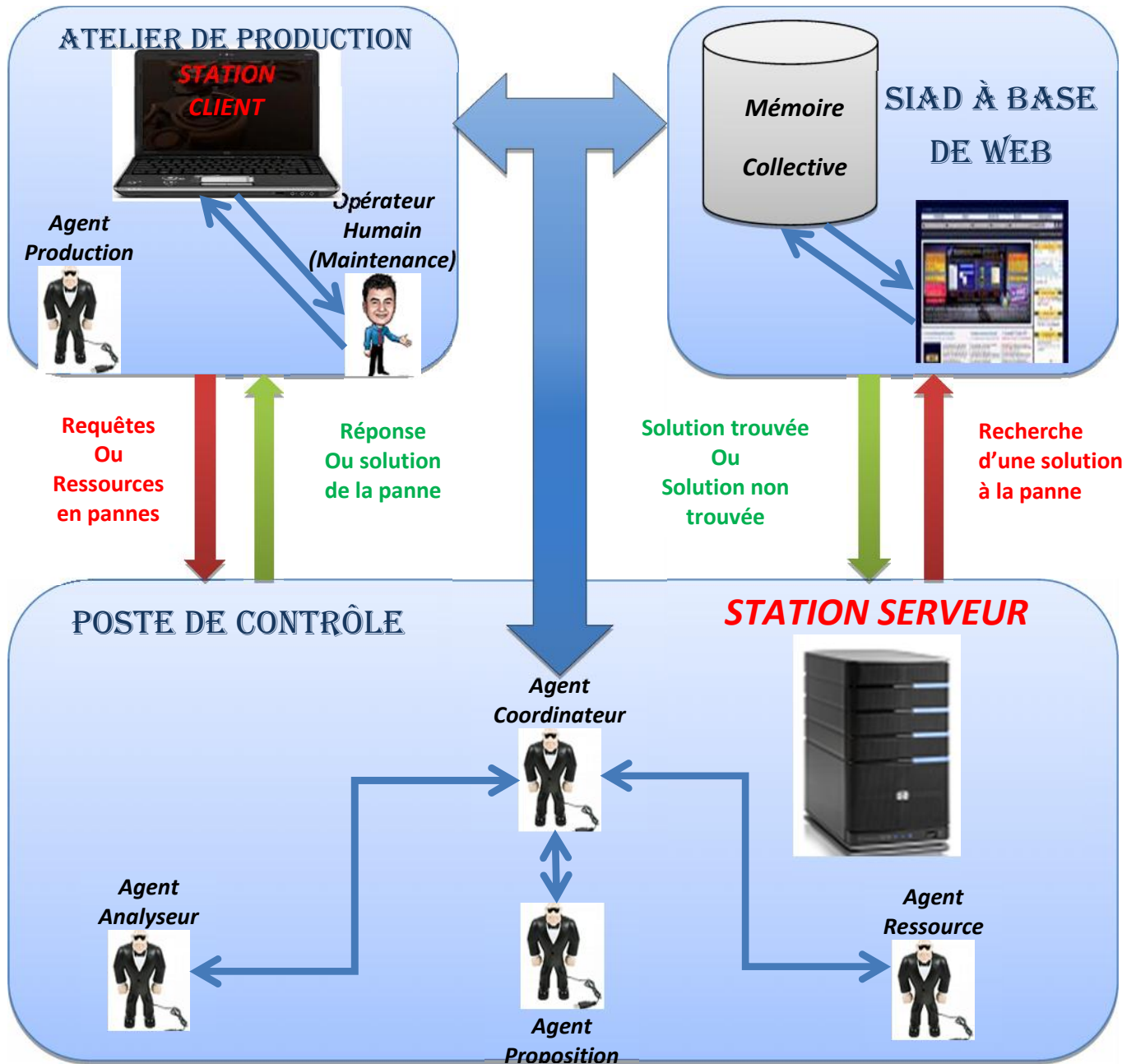


Figure1 : Architecture générale du Web-DSS

Partant de ce principe nous commençons d'abord par définir les sous-systèmes que voici :

➤ **Atelier de production:** par abus de langage nous désignerons par le terme *atelier de production* tout lieu où se trouve un certain nombre de machines et d'employés, et ce quelques soit leurs nombre ou la superficie sur laquelle ils produisent du moment que ce lieu de production possède un système de gestion informatique, donc cela peut varier du simple atelier jusqu'à une chaîne de production regroupant plusieurs usines. Pour aller dans le sens de nos idées nous dirons que c'est le lieu où la production proprement dite se fait.

L'atelier sur lequel porte notre étude est composé de plusieurs machines munies d'un certain nombre de capteurs qui sont reliées à un contrôleur électronique ayant des interfaces programmables, ce contrôleur est lui-même relié à une station qu'on appellera dans ce système une **station client**, ce client héberge notre **agent de production**, qui grâce à la communication entre le client et le contrôleur peut avoir toutes sortes d'informations sur l'état de fonctionnement de chaque machine, y compris si une panne c'est produite, sur quelle machine et surtout quelle pièce est défectueuse.

Le rôle de l'agent de production sera abordé en détails plus loin dans ce chapitre lorsque nous détaillerons notre SMA.

➤ **Poste de contrôle:** comme son nom l'indique, il s'agit de la salle de contrôle à partir de laquelle toutes les machines sont pilotées et surveillées, et c'est la raison pour laquelle nous avons choisi de mettre la **Station Serveur** dans cette salle. Le serveur en question est utilisé pour répondre à toutes sortes de requêtes dont les requêtes concernant les pannes qui se sont produites dans l'atelier. Pour ce faire le serveur contient la partie la plus importante du SMA, à savoir:

- ✓ *L'agent analyseur;*
- ✓ *L'agent proposition;*
- ✓ *L'agent ressource;*
- ✓ *L'agent coordinateur;*

1. Le rôle des composants du Web-DSS :

Afin de mieux appréhender le fonctionnement général du système, et avant de décortiquer chacun des sous-systèmes qui le compose, nous allons donner un aperçu sur le rôle de ces derniers ainsi que leur utilité dans le système. Cette démarche permet d'aller du plus spécial qui est facilement compréhensible au plus général qui n'est pas toujours évident de comprendre dès le départ, en d'autres termes affiner le système global successivement jusqu'à l'obtention d'un noyau dur qui représente pratiquement le petit composant du système et qui est tout à fait banal à comprendre.

➤ **SIAD à base de web :** la conception du SIAD ayant fait l'objet d'un précédent travail soutenu et validé [Bessedik, 2010), nous nous sommes fixés un objectif d'étude qui consiste à recenser toutes les stratégies possibles de coordination parmi celles ayant été suivies par l'agent coordinateur durant une session de travail c'est-à-dire une prise de décision globale, et pour satisfaire cet objectif nous avons enrichi le modèle proposé par **La mémoire collective (MC)**:

➤ cette mémoire est utilisée afin que les responsables de la maintenance (désignés par **opérateur humain** dans la figure correspondant à l'architecture générale du système), puissent consigner toutes les solutions auxquelles ils sont parvenus pour réparer les pannes qui se sont produites dans les machines. Aussi, cette mémoire permet de mentionner les adaptations apportées aux solutions tirées de la base des cas de l'agent coordinateur et elle

permet à ce dernier de se procurer des solutions aux pannes si elles existent au niveau de la MC.

Comme nous pouvons le voir dans la figure précédente, nous avons choisi à juste titre d'intégrer la MC directement dans les SIAD à base de web. En effet, ce choix s'avère être judicieux pour les raisons suivantes :


1. Sur le plan économique : l'utilisation d'une seule MC au niveau d'un SIAD à base de web s'avère être beaucoup plus économique que l'utilisation d'une MC pour chaque SMA à travers le web.

2. Gain de temps considérable au niveau de la recherche : prenant le cas où nous avons une MC pour chaque SMA, en lançant une recherche pour trouver une solution à une panne à travers le web, imaginez tout le temps nécessaire pour parcourir toutes les MC traitants du même problème, alors qu'avec la solution que nous avons proposé, on mettra beaucoup moins de temps à faire la même recherche puisque nous n'avons besoins de parcourir que la MC des SIAD à base de web qui eux, regroupe plusieurs SMA du même type.

3. Gain de temps considérable au niveau des MAJ de la MC : comme nous le savons la MC est avant tout une BDD qui a besoin d'être MAJ régulièrement, nous savons aussi combien de temps pourrait mettre une MAJ lorsqu'il s'agit d'une BDD de taille importante, à présent imaginons que serait ce temps en le multipliant par autant de MC qu'il y a de SMA !! là encore la solution que nous avons apportée permet de réduire ce temps considérablement et de façon exponentielle.

4. Rapidité de conception et réutilisation : étant chercheur informaticien, notre plus grand souci est de permettre aux utilisateurs du système de concentrer leur énergie à produire plus et de ne pas perdre de temps à reproduire ce qui a déjà été fait, alors la solution apportée permet de ne créer la MC qu'une seule fois par le concepteur du SIAD ce qui permet de gagner du temps et permet aux autres SIAD de réutiliser la MC en totalité ou partiellement selon les cas.

Nous pouvons remarquer que cette mémoire est liée à tous les autres composants du système, car cette dernière joue un rôle primordial dans ce système. Afin de mieux illustrer nos propos, on va décrire chacun de ces liens.

 **Entre MC et la station client** : la station client offre une interface à l'agent de maintenance pour pouvoir consulter la solution proposée par l'**agent coordinateur** pour réparer une panne donnée, ou bien d'ajouter, de modifier ou de mettre à jour des solutions qui seront sauvegardées dans la MC au cas où la solution nécessite une adaptation de la part de l'agent de maintenance ou bien si celui-ci est solliciter pour participer à une solution par le biais des SIAD à base de web. Un autre aspect très important est le fait que chaque responsable de maintenance à travers le monde est en mesure de consulter les solutions aux problèmes via le web, ceci leurs permet d'identifier les problèmes, les solutions et même de travailler ensemble sur une solution, et c'est là qu'interviennent les aspects de la coopération et de la négociation qui donnent toute la puissance à l'utilisation d'une MC.

En effet, les responsables de maintenance seront en mesure de coopérer pour trouver une solution, où de négocier des compromis satisfaisant une solution à une panne complexe.

Un autre aspect important, est le fait que toute cette coopération et cette négociation peut être sauvegardée dans la MC pour laisser des traces pour les futurs utilisateurs, pour qu'ils sachent pourquoi une telle solution a été approuvée, et comment les négociations se sont déroulées pour arriver à un tel résultat. Ceci permet de gagner énormément de temps lorsqu'on se trouve face à une panne qui a déjà été résolue par le passé. Donc le principe est d'avoir une traçabilité pour chaque solution ainsi que pour les négociations qui ont conduit vers cette solution.

✚ *Entre la MC et le poste de contrôle via l'agent coordinateur:* l'interaction présentée ici, permet à l'agent coordinateur de consulter les solutions adaptées ou modifiées au niveau de la MC si jamais sa base de cas ne contient pas la solution à la panne en question ou bien si la solution proposée ne résout pas le problème, alors l'agent coordinateur peut soumettre une requête aux différents responsable de maintenance via la MC, en faisant participer un groupe de technicien pour résoudre ce problème.

Après avoir donné un aperçu sur l'architecture du système ainsi que sur le fonctionnement de ses composants, nous allons à présent rentrer dans le vif du sujet en décrivant notre système multi-agents (SMA) et le rôle de chaque agent qui le compose en mettant l'accent sur l'**agent coordinateur** qui est le noyau ou bien la plaque tournante de ce SMA.

II. Structure générale du Système Multi-Agents (SMA) :

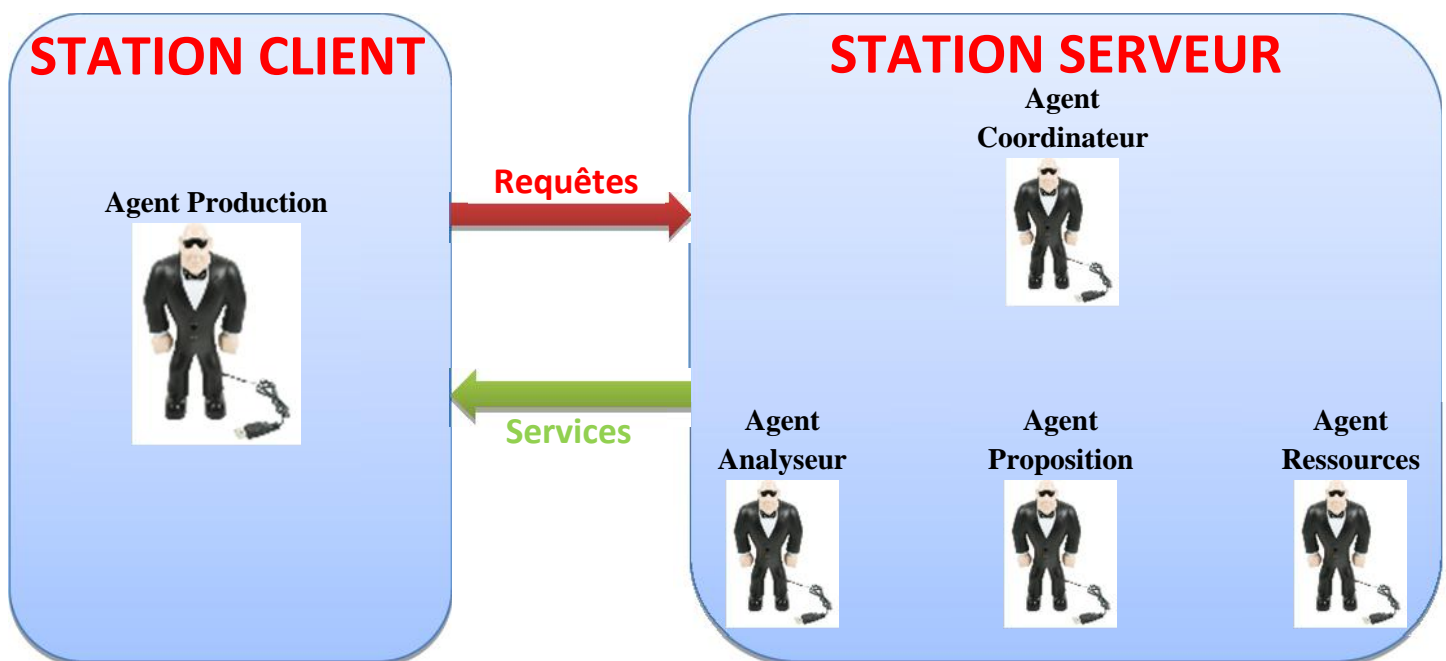


Figure 2 : répartition géographique et logistique des agents du système multi-agents.

Afin de bien comprendre la structure générale ainsi que le fonctionnement du système multi-agents que nous présentons, nous devons d'abord discuter la structure ainsi que la répartition géographique des agents qui le compose.

Le système multi-agents sur lequel est basée notre Web-DSS est composé des agents suivants :

- ✚ *Agent Production (APr)* ;
- ✚ *Agent Analyseur (AA)* ;
- ✚ *Agent Proposition (AP)* ;
- ✚ *Agent Ressource (AR)*.
- ✚ *Agent Coordinateur (AC)* ;

Ces agents sont répartis sur deux sites distincts, l'*Agent Production* est hébergé au niveau de la *Station Client* dans l'atelier de production, quant aux autres agents ils sont hébergés dans la *Station Serveur* au niveau du poste de contrôle. Le rôle ainsi que la structure des agents seront abordés dans la section suivante.

III. Architecture des agents du SMA :

Avant de commencer à présenter les structures des agents qui composent notre SMA, nous devons d'abord mentionner le fait que nous avons à faire dans ce système principalement a des agents de *type cognitif* , exception faite pour l'agent production qui lui, répond à une architecture hybride, cette dernière sera aborder lorsque nous décrierons l'agent en question.

1. Agent Production (APr):

a- Architecture et composants de l'Agent Production:

L'agent de production est composé de sept sous-systèmes:

- Sous-système d'information : permet d'assurer la communication entre les sous-systèmes et de gérer les informations locales.
- Sous-système de communication : assure la liaison entre l'agent de production et les autres agents constituant le système de production.
- Sous-système d'interface : permet l'échange d'informations entre l'opérateur humain et l'agent.
- Sous-système de contrôle / commande : assure la commande des ressources de production et transmet les informations de suivi au Sous-système d'information.
- Sous-système de décision : ce sous-système est responsable de l'ensemble des décisions prises au cours du processus de réalisation du problème.
- Un module d'analyse et de réaction : afin de prendre en compte à tout instant l'état de l'atelier [Rouba, 2005].
- Un ensemble de comportements associés à chaque agent.

Comme nous pouvons le remarquer à partir de cette figure l'architecture de l'Agent Production est une architecture qui met en avant le concept de couches. Nous distinguons alors deux zones de couches :

- La zone de gauche représente tous ce qui est connaissances de l'agent.
- La zone de droite quant à elle, représente la structure de contrôle de l'agent.

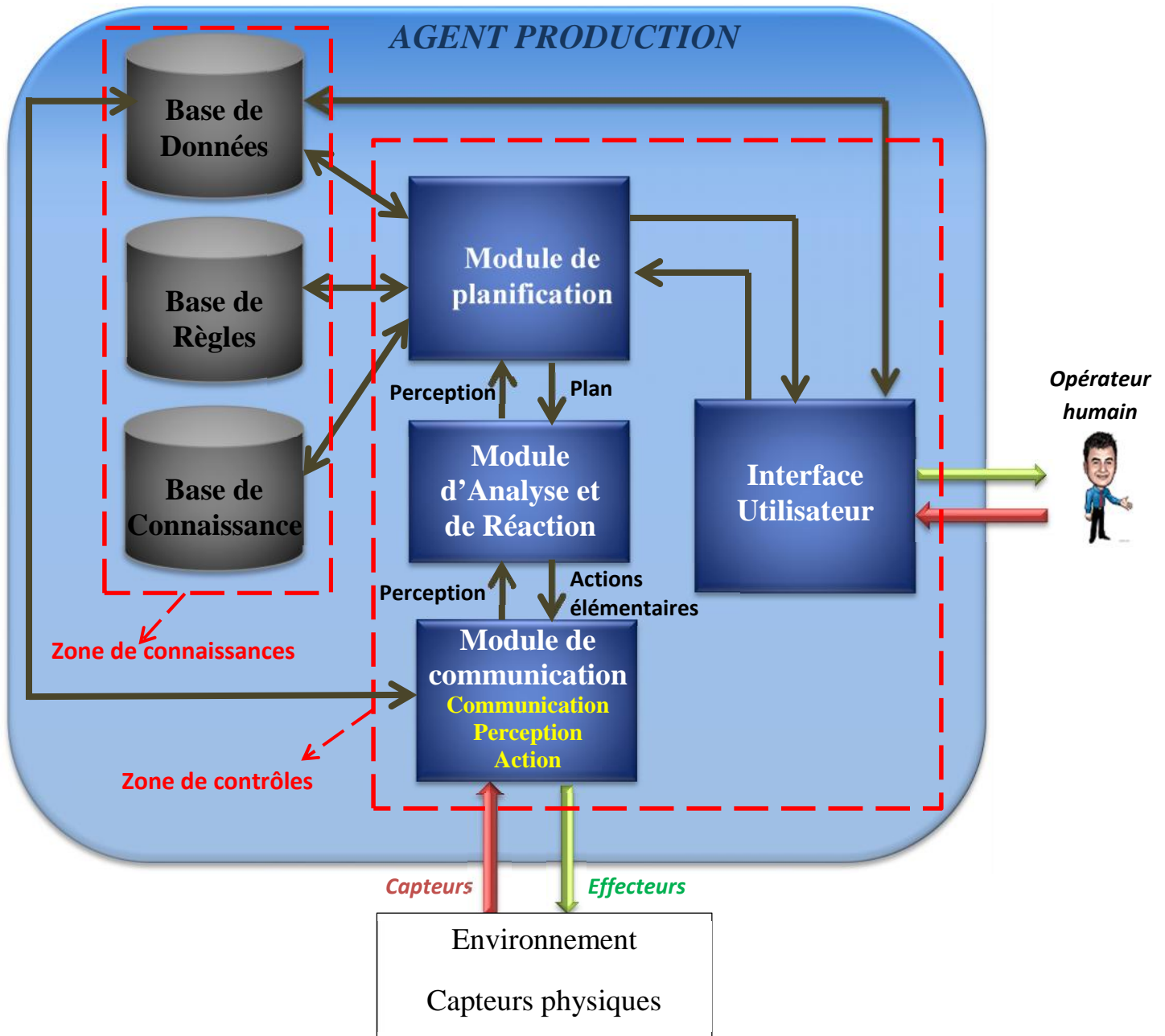


Figure 4 : Structure de l'Agent Production

Ces deux zones contiennent des couches agencées verticalement pour mettre en évidence le traitement en série de la perception de l'APr qui s'effectue de bas en haut. Ces couches sont les suivantes en commençant par la zone de contrôle :

- ✓ *Le module de communication ;*
- ✓ *Le module d'analyse et de réaction ;*
- ✓ *Le module de planification ;*
- ✓ *L'interface utilisateur.*

La deuxième zone est celle représentant les connaissances et elle se compose de :

- ✓ *La base de données ;*
- ✓ *La base de connaissances ;*
- ✓ *La base de règles.*

b- Rôles des composants de l'Agent Production :

Dans cette section nous allons décrire à quoi sert chacun des composants du système de l'APr. Ceci nous permet de décrire de façon concise le fonctionnement interne de cet agent.

b1. Rôle du module de communication :

La communication, dans les systèmes multi-agents comme chez les humains, est à la base des interactions et de l'organisation sociale. Sans communication, l'agent faisant partie du SMA n'est qu'un individu isolé, sourd et muet aux autres agents, refermé sur sa boucle perception-délibération-action. C'est parce que les agents communiquent qu'ils peuvent coopérer, coordonner leurs actions, réaliser des tâches en commun et devenir ainsi de véritables êtres sociaux. La communication s'exprime comme une forme d'interaction dans laquelle la relation dynamique entre les agents s'exprime par l'intermédiaire de signaux, qui, une fois interprétés, vont produire des effets sur ces agents.

On l'a bien compris, la communication au sein d'un SMA est on ne peut plus importante, d'où l'importance et la présence du **module de communication** au sein de tous les agents composants notre SMA.

Pour pouvoir communiquer et faire parvenir des perceptions de l'environnement et agir sur celui-ci par des actions relatives au fonctionnement de l'agent, le module de communication doit impérativement être muni de :

- un **protocole de transport** : un mécanisme de transport de données (Ex. TCP/IP, SMTP, HTTP, ...) qui doit être opaque pour les agents.
- un **langage de communication** : un médium au travers duquel les attitudes relatives au contenu de l'échange sont communiquées.
- un **protocole d'interaction** : une stratégie de haut niveau (Ex: une stratégie de négociation) qui gouverne des conventions qui doivent être respectées lors de l'interopérabilité (ou la coopération), Ex. KQML ou FIPA-ACL.

b2. Rôle du module d'analyse et de réaction :

La couche d'analyse et de réaction est en fait une couche réactive traitant les évènements perçus par la couche de communication, que ce soit par la perception directe du

monde, ou bien par la réception de messages provenant des autres agents. Cette couche a accès à des modèles de comportements stockés dans la base de connaissance, via *la couche de planification*.

Cette couche peut être considérée comme une couche comportementale, car c'est à ce niveau que le type de comportement de l'agent sera déterminé, il s'agit de construire les *désirs* de ce dernier. En effet, en fonction de la perception reçue, ce module peut adopter soit *un comportement réactif* soit *un comportement cognitif*, et ceci nous mène précisément à la fonction d'analyse de cette couche.

Le comportement sélectionné par la fonction d'analyse répond aux stimuli reçus de l'environnement, mais ce comportement peut-être de plusieurs natures. En effet, en cas de réponse à un stimulus simple, comme la scrutation périodique de l'état de fonctionnement de l'atelier de production ou la perception d'une panne, le comportement plus ou moins primaire sélectionné sera directement transmis au module de communication pour exécution, dans ce cas il s'agit d'examiner directement le type de la panne et de prendre connaissance de la référence de la pièce en panne. Après avoir détecté une panne, il faut lui trouver une solution, dans ce cas le problème devient plus complexe, à ce moment-là, ce module n'ayant pas les compétences nécessaires pour traiter ce genre de problèmes, l'action qui doit être effectuée est celle de faire appel à la couche de plus haut niveau dans la zone de contrôle, c'est-à-dire *la couche de planification* et ce afin de créer un plan.

b3. Rôle du module de planification :

Comme nous venant de le citer, le module d'analyse et de réaction va envoyer la perception qu'il a reçue au module de planification, ce dernier va lui renvoyer un plan à suivre, ensuite, c'est au module d'analyse et de réaction d'interpréter le plan et de le traduire en actions élémentaires compréhensibles par le module de communication qui aura la charge de le faire appliquer.

Avant d'aller plus loin, nous devant d'abord définir ce qu'est un **Plan**. Un plan est un découpage d'un problème en plusieurs sous-problèmes, ces derniers devront être résolus par coopération avec les autres agents du SMA car cet agent seul n'a aucun moyen de réponse au problème initial, c'est-à-dire un but qui ne peut pas être atteint par l'utilisation simple d'un comportement primitif. Ce découpage de but à atteindre se fait bien évidemment en plusieurs autres buts, jusqu'à obtenir une succession de buts réalisable par l'agent (c'est à dire par une succession de buts pouvant être atteints par des primitives).

Cependant, il existe deux types de plans, le premier concerne les plans qui sont réalisables en local (l'agent peut résoudre toutes les sous-tâches tout seul), alors que le second type de plans intervient lorsqu'une tâche ne peut pas être réalisée par un seul agent, et que des communications inter-agents sont nécessaires afin de coordonner leurs actions pour résoudre le problème.

Pour réaliser un plan, le module de planification se base sur la zone de connaissances, celle-ci est composée de trois couches distinctes à savoir **la Base De Données (BDD)**, **la Base**

De Connaissances (BDC) et enfin **la Base De Règles (BDR)**. Le rôle de chacune de ces bases sera défini par la suite, mais pour le moment concentrons-nous sur leur rôle par rapport au module de planification, ce dernier doit être en mesure de reconnaître les agents avec lesquels il doit coordonner ses actions et coopérer pour résoudre un problème donné, ces informations se trouvent au niveau de la BDC. Aussi, il doit être en mesure de consulter les croyances de l'agent sur son environnement et ce à partir de la BDD. A présent que ce module dispose de toutes les informations lui permettant de reconnaître un problème au niveau de l'atelier et avec quels agents il doit coopérer pour résoudre ce dernier, le comportement du module de planification doit être régi par des règles, ces dernières se trouvent dans la BDR et permettent de définir le comportement non seulement de ce module mais aussi celui de l'agent de production et par conséquent elles permettent au module de planification de créer un plan pour la résolution d'un problème soit au niveau local soit en faisant intervenir les différents agents composant notre SMA.

b4. Rôle de l'interface utilisateur :

Dans ce système, ce composant a un rôle qui est commun à toute interface logicielle et répond à une définition générale commune à toute interface, ceci étant, nous avons choisi de développer une définition qui se rapproche le plus du rôle que nous lui attribuons au niveau de ce système, à savoir : Une **interface** définit la frontière de communication entre deux entités ou plus, comme entre des éléments logiciel, entre des composants matériel, ou avec des utilisateurs.

Une interface se réfère généralement à une image abstraite qu'une entité fournit d'elle-même à l'extérieur. Cela permet de distinguer les méthodes de communication avec l'extérieur et les opérations internes, et autorise à modifier les opérations internes sans affecter la façon dont les entités externes interagissent avec elle et ce en fonction des niveaux d'abstractions définis.

A partir de cette brève définition nous pouvons mettre en évidence les fonctions de cette interface qui ne sont pas des moindres. En effet, cette couche joue un rôle primordial dans ce système car elle permet :

1. Assurer l'interaction Homme/Machine pour permettre au technicien de maintenance de visualiser la solution proposée pour résoudre une panne ainsi que pour l'aider à accéder à la mémoire collective se trouvant au niveau du SIAD pour les raisons que nous avons citées auparavant.
2. Encapsulation des données du système, on ne permettant à son utilisateur d'accéder qu'aux données qui lui sont autorisées via cette interface.
3. Permettre à l'opérateur humain de mettre à jour toutes les bases se trouvant dans la zone de connaissances.

Il existe d'autres fonctions qu'on peut attribuer à ce module mais comme elles ont déjà été présentées dans la partie concernant le module de planification vu qu'elles ont un lien direct avec ce dernier, nous nous abstenons de les citer à nouveau.

Après avoir présenté tous les composants constituant la zone de contrôle, nous allons désormais décrire brièvement les bases se trouvant dans la zone de connaissances, qui peut être décrite comme étant « l'intelligence » de l'agent.

b5. Rôle de la zone de connaissances :

Cette zone est composée de trois bases distinctes, à savoir **la Base De Données (BDD)**, **la Base De Connaissances (BDCo)** et enfin **la Base De Règles (BDR)**, chacune de ces bases a un rôle bien précis dans ce système et contiennent toutes les données nécessaires au bon fonctionnement de l'agent.

La Base De Données, contient tout ce que sait l'agent sur le monde dans lequel il évolue, cela va des agents qu'il a rencontrés, aux objets de l'environnement, la topographie de l'atelier et celle du réseau avec tout ce qui se rapporte aux passerelles, @ IP...

Les données se trouvant dans la BDD lui proviennent directement des capteurs physiques se trouvant dans l'atelier via le module de communication ou bien à travers l'interface utilisateur si l'opérateur humain a quelque chose à modifier ou à ajouter. Ces données représentent la perception de l'agent c'est-à-dire ses propres croyances, mais ces dernières peuvent être incorrectes, incomplètes ou incertaines et, à cause de cela, elles sont différentes des connaissances de l'agent, qui sont des informations toujours vraies. Les croyances peuvent changer au fur et à mesure que l'agent recueille des informations qui lui proviennent soit par sa capacité de perception, soit par l'interaction avec les autres agents ou avec l'opérateur humain.

La seconde base est la **Base Des Règles**, qui peut être considérée comme une base de patrons de comportement contenant une liste de comportements primitifs, qui sont des unités de connaissances appelées **les règles**. Chacune de ces règles est établie en respectant les conditions suivantes :

- Les règles sont formulées comme suit: **Si Prémises Alors Conclusions**.
- Ce sont des Connaissances déclaratives et révisables.
- Une règle n'est pas désignée par un nom, mais par ses conditions de déclenchement (prémises) et ses effets (conclusions) : mode d'accès associatif.
- L'ensemble des règles forme la **base de règles**.

Comme on dit « un exemple vaut mieux qu'un long discours », alors nous allons donner un petit exemple sur les règles qu'on peut employer dans ce système :

R1 : SI panne=true ALORS chercher_la_référence_de_la_pièce_en_panne

R2 : SI ¬panne=true ET système_défaillant=false ALORS envoi_message_bonne_condition

R3 : SI ¬système_défaillant=false ALORS alerter_opérateur

R4 : SI avoir_référence ET liste_des_ressources_en_panne ALORS
envoyer_la_liste_à_agent_coordinateur

Remarque : Les règles utilisées sont celles de la logique des propositions : ordre zéro.

Enfin, vient la **Base des Connaissances** dans laquelle on sauvegarde des données et des informations toujours vraies et persistantes à travers le temps et à travers les changements que peut connaître l'atelier ou le réseau dans lequel évolue l'agent production. Ces connaissances peuvent être de types différents et de natures différentes mais elles ont toutes une caractéristique commune, qui est celle de décrire le bon fonctionnement de l'agent.

En effet, les informations qui sont enregistrées dans cette base se rapportent aux protocoles de communication, de négociation, de coordination et des plans pour accomplir soit des tâches avec plusieurs agents à la fois, soit des tâches en locale par le biais du module de planification pour la détection et la résolution des pannes au sein de l'atelier. Donc elles fournissent à chaque module composant l'agent production, les connaissances dont il a besoin pour son fonctionnement interne.

c- Rôle de l'Agent Production :

L'agent production a pour rôle d'effectuer les principales tâches suivantes :

- ✓ Détecter les pannes et les anomalies pouvant se produire au niveau des machines dans l'atelier de production. Cela, à partir des capteurs se trouvant au niveau des machines via le module de communication;
- ✓ Mettre en place des plans à travers lesquels cet agent doit envoyer une liste de toutes les ressources en pannes ainsi que leurs références respectives à l'agent de coordination. Ces plans sont réalisés par le module de planification après avoir reçu des perceptions par le module d'analyse et de réactions;
- ✓ Mettre en pratique les plans concernant les solutions aux pannes détectées provenant de l'agent coordinateur, ces plans sont reçus par le module de communication qui les mettra en pratique via ses effecteurs mais après que ces derniers soient traités par le module de planification ensuite par le module d'analyse et de réaction qui va les interpréter et envoyer des instructions élémentaires au module de communication pour qu'ils soient compréhensibles par ce dernier;
- ✓ Permettre à l'opérateur humain de consulter aussi bien les solutions proposées, ainsi que les plans établis par l'agent et les pannes détectées ;
- ✓ Permettre au technicien de maintenance de consulter la MC ou de faire une recherche à partir de celle-ci au cas où le SMA ne parvient pas à trouver une solution ou que la solution proposée n'est pas tout à fait adéquate à une panne donnée.

d- variables temporelles de l'agent production :

Jusque-là, nous avons décrit l'agent production mais sans vraiment prendre en considération l'aspect temporel qui est très importants dans ce système d'autant plus qu'on se propose de faire une étude statistique qui nous permettra de désigner précisément qu'elle est la meilleure stratégie de coordination pour la résolution des pannes dans les plus brefs délais.

Donc la connaissance des valeurs de ces variables est impérative pour décrire les performances du système, pour ce faire ces variables doivent être réparties dans le système et donc chaque agent du SMA possède les siennes selon son rôle dans le système.

Pour l'agent production, nous pouvons définir une seule variable temporelle non négligeable :

- **Temps de recherche de la panne (T_{RP})** : c'est le temps nécessaire à l'agent pour détecter une panne au niveau de l'atelier, sur quelle machine et quelle est la pièce défectueuse, aussi il doit prendre connaissance des informations relatives à la pièce en question c'est-à-dire la référence de la pièce, son producteur, ses différentes capacités...etc. Ce temps sera initialisé au niveau du module de planification dès lors qu'il reçoit la perception indiquant qu'une panne c'est produite, après, lorsqu'il reçoit les informations concernant la panne il va calculer le temps qui a été consommé pour faire cette recherche, ce temps sera enregistré temporairement dans la BDD pour être supprimé ensuite dès qu'il sera communiquer à l'agent coordinateur.

$T_{RP} = T_i - T_0$ où T_0 représente l'heure à laquelle la panne c'est produite et T_i représente l'heure à laquelle l'agent reçoit la solution concernant la ressource en panne.

Après avoir décrit le seul agent résidant dans la station client, nous allons maintenant décrire tour à tour les agents se trouvant dans la station serveur, et qui sont au nombre de quatre comme on l'a déjà cité précédemment.

e- Algorithme du fonctionnement de l'agent production :

L'agent de production exécute deux fonctions en parallèles :

- **La fonction de scrutation** : cette fonction permet de parcourir la liste des ressources afin de trouver d'éventuelles pannes. A chaque fois qu'une nouvelle panne est trouvée, l'agent met à jour sa perception. A intervalle régulier, cette fonction envoie une requête avec la liste des ressources en panne au serveur.

- **La fonction de résolution** : cette fonction est exécutée quand l'agent reçoit la solution transmise par le serveur. Elle met à jour sa perception et l'indique à l'utilisateur via son interface.

```
// GetRessourcesState : Cette fonction renvoie un l'état des ressources sous forme d'un tableau  
//d'entiers  
//Chaque case de ce tableau contient « -1 » si il n y a pas de panne, sinon elle contient l'identifiant  
//de la panne
```

```
Function Scrution()  
Const scrutTime = 10 ;  
Type PanneRef = record  
ressource_id, pann_id : integer ;  
to: time ;  
end ;  
Type PanneRefList = List Of PanneRef ;  
Var tmp, tmp2 : Time ;  
pannList : PanneRefList ;
```

```

    ressources :array of integer ;
    ref : PanneRef ;
Begin
    CreateEmptyPanneRefList(pannList);
    While(true)
        Begin
            Tmp = GetTime();
            Repeat
                Ressources = GetRessourcesState();
                For i = 1 to SizeOf(ressources) do
                    Begin
                        If(ressources[i] <>-1)
                            Begin
                                Ref.ressources_id = i ;
                                Ref.pann_Id = ressources[i] ;
                                Ref.To = getTime();
                                If(not alreadyContains(pannList,
ref))
                                    Append(pannList , ref);
                            End ;
                        End ;
                    End ;
                Until(tmp2 – tmp > scrutTime)
                SendRequest(panneList);
                EmptyList(PanneList);
            End ;
        End ;
End ;

Type Solution = record
    Solved : boolean ;
    Ressource_id, panne_Id : integer ;
    To, ti, trp, t_analy, t_ress, t_prop : Time ;
    Solution : string ;
End ;
Function Resolve(Sol : Solution)
Begin
    Sol.ti = GetTime();
    Sol.trp = sol.ti – sol.to ;
    UpDateDataBase();
    ShowResults();
End ;

```

2. Agent Analyseur (AA) :

a- Architecture et Composants de l'Agent Analyseur (AA) :

Comme ça été le cas pour l'agent production, cet agent est aussi composé de deux zones distinctes, la zone de connaissances comprenant toutes les bases,

- ***Base de Données ;***
- ***Base de règles ;***

- *Base de connaissances.*

et la zone de contrôles représentant l'aspect fonctionnel de l'agent et comprenant les autres modules, qui sont :

- *Module d'interface ;*
- *Module d'analyse ;*
- *Filtre ;*
- *Générateur de propositions ;*

La figure ci-après nous montre de quelle façon sont agencés les composants de l'AA et quels sont les liens existants entre ces derniers.

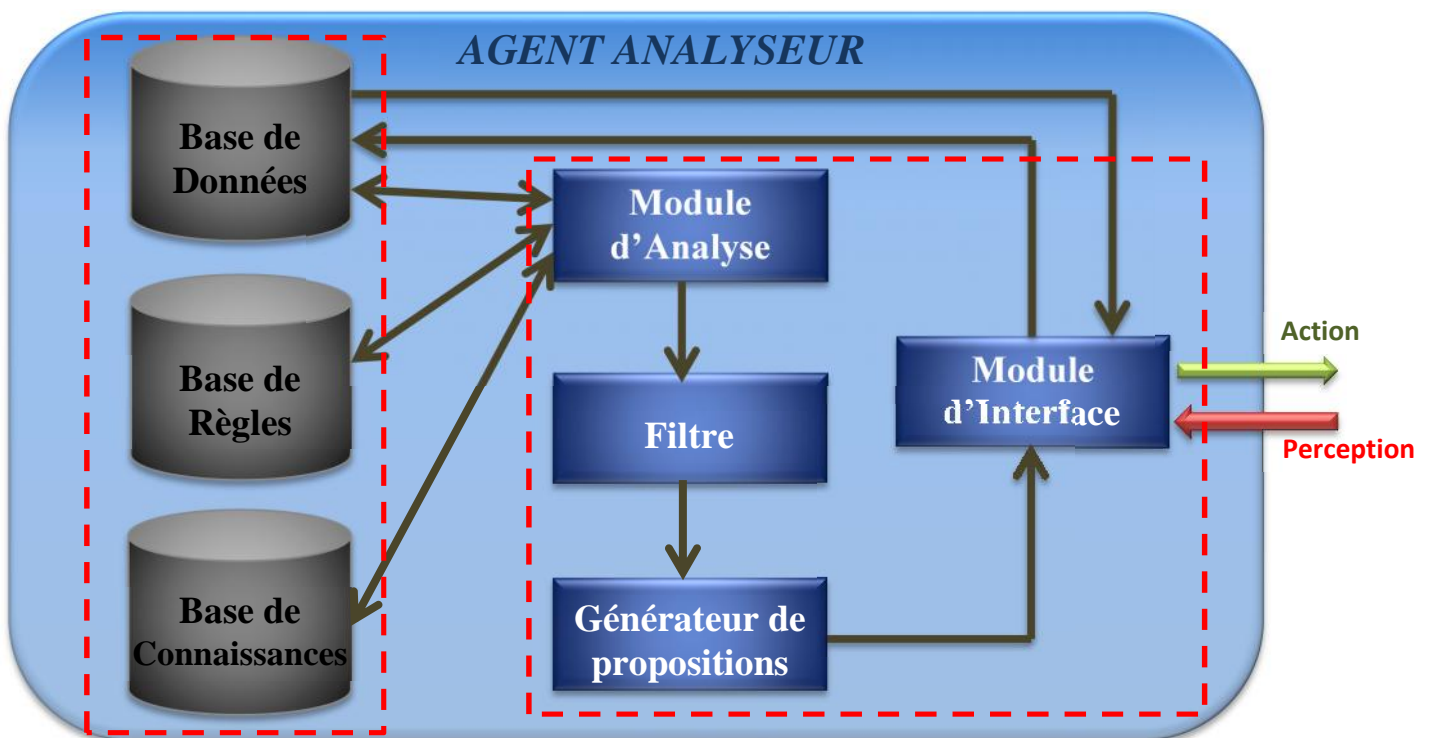


Figure 5 : Structure de l'Agent Analyseur

b- Rôles des composants de l'Agent Analyseur (AA) :

A partir de cette figure nous pouvons distinguer chacun des modules qui composent l'agent analyseur, cependant, il est impératif de décrire le rôle de chacun d'eux dans ce système afin de bien comprendre le fonctionnement interne de cet agent. Comme pour l'agent production, la description se fera de droite à gauche c'est-à-dire en commençant d'abord par la zone de contrôle et en respectant le cheminement de la perception de l'agent depuis le module d'interface jusqu'au générateur de propositions.

b1. Rôle du module d'interface :

Au fur et à mesure qu'on va présenter les agents de notre SMA, nous allons nous apercevoir qu'il y a certains modules que nous allons retrouver chez plusieurs agents et qui

jouent les mêmes rôles, donc, afin d'éviter de refaire ce qui a été déjà fait et pour éviter toute redondance d'informations, nous n'allons présenter que les modules qui n'ont pas été présentés ou que leur rôle dans le système diffère d'un agent à un autre.

Pour le module d'interface qu'on va retrouver dans tous les agents, nous dirons que ce module joue les mêmes rôles que ceux du module de communication précédemment décrit (agent production) à l'exception que pour cet agent, ce module n'échange ses informations qu'avec les autres agents composants notre SMA ou bien dans certains cas il peut échanger des informations via le web, mais il n'a pas de relation directe ni avec l'atelier de production ni avec l'opérateur humain.

Pour la zone de connaissances, les rôles de chaque composant restent les mêmes pour tous les agents composants ce système.

b2. Rôles du module d'analyse et du filtre:

Ces deux modules ont des rôles complémentaires d'où le fait qu'on va les présenter ensemble, pour permettre une meilleure compréhension. Notons juste que ces modules jouent des rôles très importants dans ce système, car :

- Le module d'analyse permet d'analyser et de construire la perception de l'AA à partir de la BDD, une analyse qui permet de prendre connaissance de la liste des ressources en panne à partir de sa perception de l'environnement;
- Dès qu'il reçoit cette liste qui lui provient de l'Agent de Coordination, il associe à chaque ressource en panne les références adéquates sans lesquelles aucune recherche ne pourra avoir lieu ;
- Si la vérification effectuée mentionne qu'il y a une quelconque erreur au niveau des références, ce module le mentionne directement à l'agent coordinateur ;
- Par contre si la vérification effectuée ne mentionne aucune erreur liée aux références des pièces défectueuses alors, le module d'analyse commence à trier la liste des ressources en pannes selon un ordre chronologique de type FIFO, cet ordre permet de résoudre d'abord les pannes les plus anciennes;
- Dès lors que ce tri est effectué une autre liste est créée mais cette fois-ci en respectant l'ordre préalablement établi ;
- La liste étant ainsi établie, elle sera envoyée au filtre qui va faire en sorte que cette liste ne contient aucun doublon c'est-à-dire que si jamais deux pièces ayant les mêmes références sont présentes dans cette liste, le filtre va faire en sorte d'éliminer l'une d'elle en augmentant uniquement le nombre d'occurrence de cette panne ce qui revient exactement à mentionner le nombre de pièces en pannes ayant les mêmes références.
- Une autre fonction qu'on peut attribuer au filtre est qu'il revoit et filtre les intentions de l'AA, ces mêmes intentions qui seront traduites en actions élémentaires.

b3. Rôles du générateur de proposition :

Ce module permet de créer les plans relatifs à la liste des ressources en pannes, car après la réception de celle-ci, il doit envoyer une à une les ressources en pannes à l'agent de

coordination en lui demandant de lancer une recherche sur les références de la ressource en panne.

Dès qu'il reçoit la liste, ce module a pour objet de chercher un diagnostic à chaque à partir de la MC ou bien à partir du web.

c- Rôle de l'agent analyseur :

Principalement, nous pouvons attribuer les rôles suivants à l'AA :

- ✓ Pour commencer, dès lors que l'AA reçoit la liste des ressources en pannes de la part de l'AC, il va commencer par mettre à jour ça perception à partir de la BDD ;
- ✓ Ensuite, il va analyser et filtrer cette liste par le biais du module d'analyse et le filtre ;
- ✓ Le résultat de cette analyse est principalement l'ordonnancement des pannes (FIFO) et la suppression des doublons ;
- ✓ La liste en question étant établie, le générateur de proposition va veiller à l'attribution des références à chaque ressource en panne ;
- ✓ Enfin, ce générateur va créer un plan dans lequel il prévoit d'envoyer une à une les références des ressources en pannes à l'AC avec le diagnostic de chacune d'entre-elles.

d- Variables temporelles de l'agent analyseur :

Dans le cas de cet agent, on peut définir uniquement le temps nécessaire pour effectuer une analyse pour filtrer et dresser la liste des ressources en pannes. Si cette liste n'est pas très importante, ce temps peut être négligé. Par contre, si la liste en question est de taille importante, alors ce temps peut être conséquent :

- **Temps d'analyse et de filtrage (T_{AF}).**
- **Temps improductif (T_{IMP}).** Il représente le temps où la connexion au réseau n'est pas disponible, ou bien si l'agent est occupé ne pouvant pas prendre en charge tout de suite une nouvelle panne.
- **Temps de recherche à partir de la MC (T_{RM}).**
- **Temps de recherche à partir du web (T_{RW}).**
- **Temps de proposition de diagnostique (T_{PD}) = $T_{AF} + T_{IMP} + T_{RM} + T_{RW}$**

e- Algorithme de fonctionnement de l'agent analyseur :

L'agent analyseur reçoit une liste de variables PannRef contenant les références des ressources ainsi que celles des pannes. L'agent analyseur parcourt cette liste afin d'éliminer les doublons. Il revoit une autre liste contenant les références des pannes ainsi que le nombre d'occurrences de chaque panne.

```
// GetIndexOfPanne : cette fonction reçoit comme paramètres la liste des Pannes ainsi que  
//l'identifiant  
//de la panne recherchée dans cette liste. Elle renvoie l'index de cette panne dans la liste,  
//sinon elle renvoie « -1 »
```

```
Type PannCount = record
    Pann_Id, Count : integer ;
End ;
Type pannCountList = List Of PannCount ;
Function GetPannFiFoList(List : PannRefList) : PannCountList ;
Var i, pann, ps : integer ;
    cPann = PannCount ;
Begin
    PannCountList result = CreateEmptyPannCountList() ;
    For i = 1 to Count(List) do
        Begin
            Pann = list[i].pann_Id ;
            Ps = GetIndexOfPanne(result, pann) ;
            If (Ps = -1)
                Begin
                    cPann. Pann_Id = Pann ;
                    cPann.count = 1 ;
                    Append(result, cPann) ;
                End
            Else
                Begin
                    Inc(result[ps].Count) ;
                End ;
            End ;
        End ;
    End ;
Return result ;
End ;
```

3. Agent Ressources (AR) :

a- Architecture de l'agent ressources :

Comme pour les autres agents, cet agent est composé de deux zones distinctes :

La zone de connaissances comprend,

- *Base de Données ;*
- *Base de règles ;*
- *Base de connaissances.*

et la zone de contrôles de l'agent est composée des modules suivant :

- *Module d'interface et de communication ;*
- *Module de recherche ;*
- *Générateur de références ;*

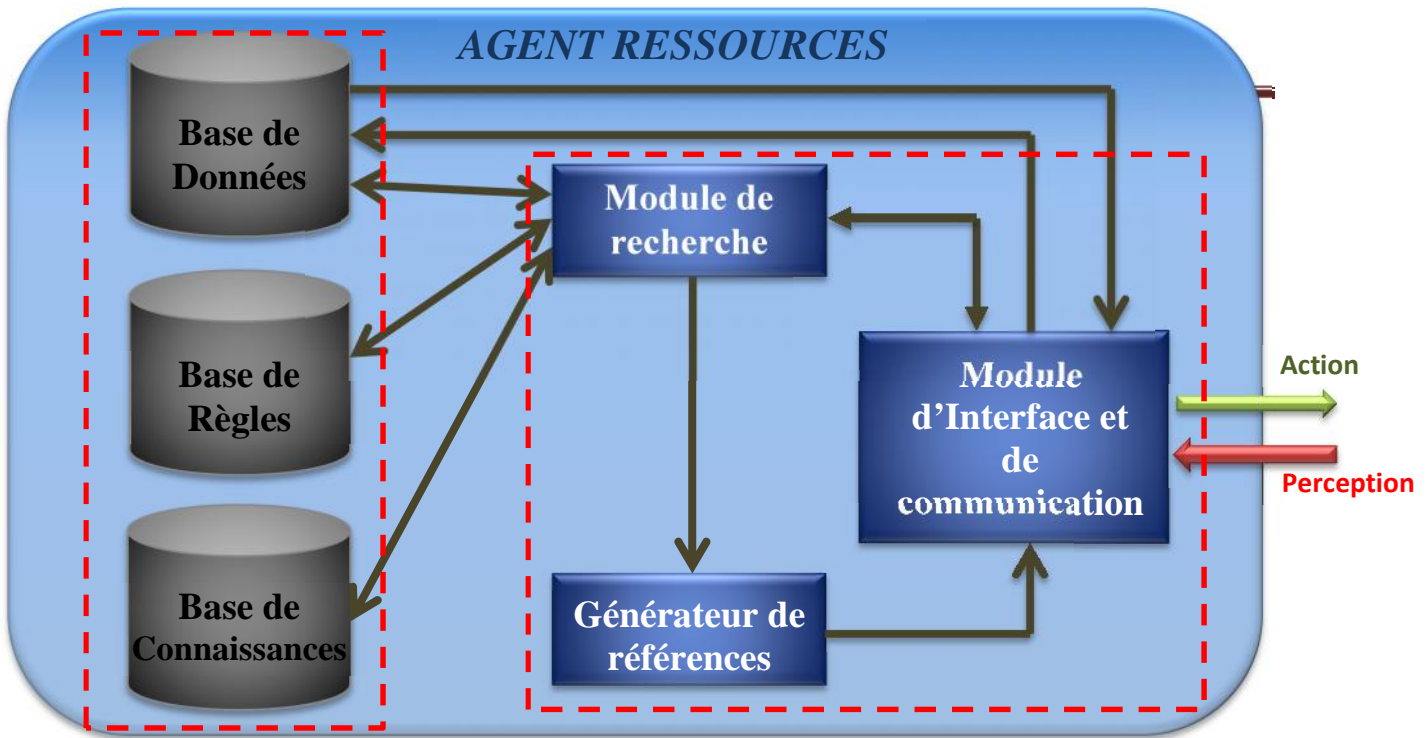


Figure 6 : Structure de l'Agent Ressources

b- Rôles des composants de l'agent ressources :

Concernant le module d'interface et de communication de cet agent est exactement comme celui que nous avons défini pour l'AP.

Pour le module de recherche de l'AR, il est lui aussi identique à celui de l'AP, sauf qu'il ne recherche pas de solutions aux pannes, mais il recherche les types et les références des ressources en pannes à partir de la MC ou bien sur le web.

Dès que la recherche est effectuée le générateur de référence récupère l'ensemble des références trouvées et construit un plan. Ce dernier sera envoyé à l'AC qui va à son tour le transmettre à l'AP dont nous avons décrit le fonctionnement à la section précédente.

c- Variables temporelles de l'agent ressources :

Pour cet agent, nous pouvons définir les variables temporelles suivantes :

- $T_{RR} = T_{RW} + T_{IMP}$.

Où : T_{RR} représente le temps de recherche de la référence d'une ressource en panne.

T_{RW} représente le temps de recherche sur le web.

T_{IMP} représente le temps improductif c'est-à-dire le temps où la connexion au réseau n'est pas disponible, ou bien si l'agent reçoit la référence de la ressource et qu'il est occupé ne pouvant rechercher une solution à la nouvelle panne.

d- Algorithme de fonctionnement de l'agent ressource :

L'agent ressources va chercher la référence de la panne dans le web. Il nous renvoie le type de la pièce en panne.

// SendWebRequest : cette fonction envoie l'identifiant de la panne à une base de données //spécialisée sur le web

//Elle nous renvoie directement le type et la référence de la pièce en panne sous forme de //chaîne de caractères

```
Type ressourceResult = record
    Conn : boolean ;
    ref : string ;
End ;

Function searchForRessource(pann_Id : integer) : ressourceResult;
Var result : ressourceResult ;
Begin
    Result.conn = VerifyConnection() ;
    If(Result.conn) then
        Result.ref = SendWebRequest(pann_id) ;
    Return result ;
End ;
```

4. Agent Proposition (AP):

a- Architecture de l'agent proposition :

Cet agent possède pratiquement la même architecture que l'AR précédemment décrit, à l'exception, qu'au lieu de chercher et de générer des références aux pannes, il recherche une solution à la panne.

Donc, au lieu d'avoir un module générateur de références, il possède un module générateur de solutions. Ainsi, l'architecture de cet agent peut être définie comme suit :

- *Base de Données ;*
- *Base de règles ;*
- *Base de connaissances.*

et la zone de contrôles représentant l'aspect fonctionnel de l'agent et comprenant les autres modules, qui sont :

- *Module d'interface et de communication ;*
- *Module de recherche ;*
- *Générateur de solutions ;*

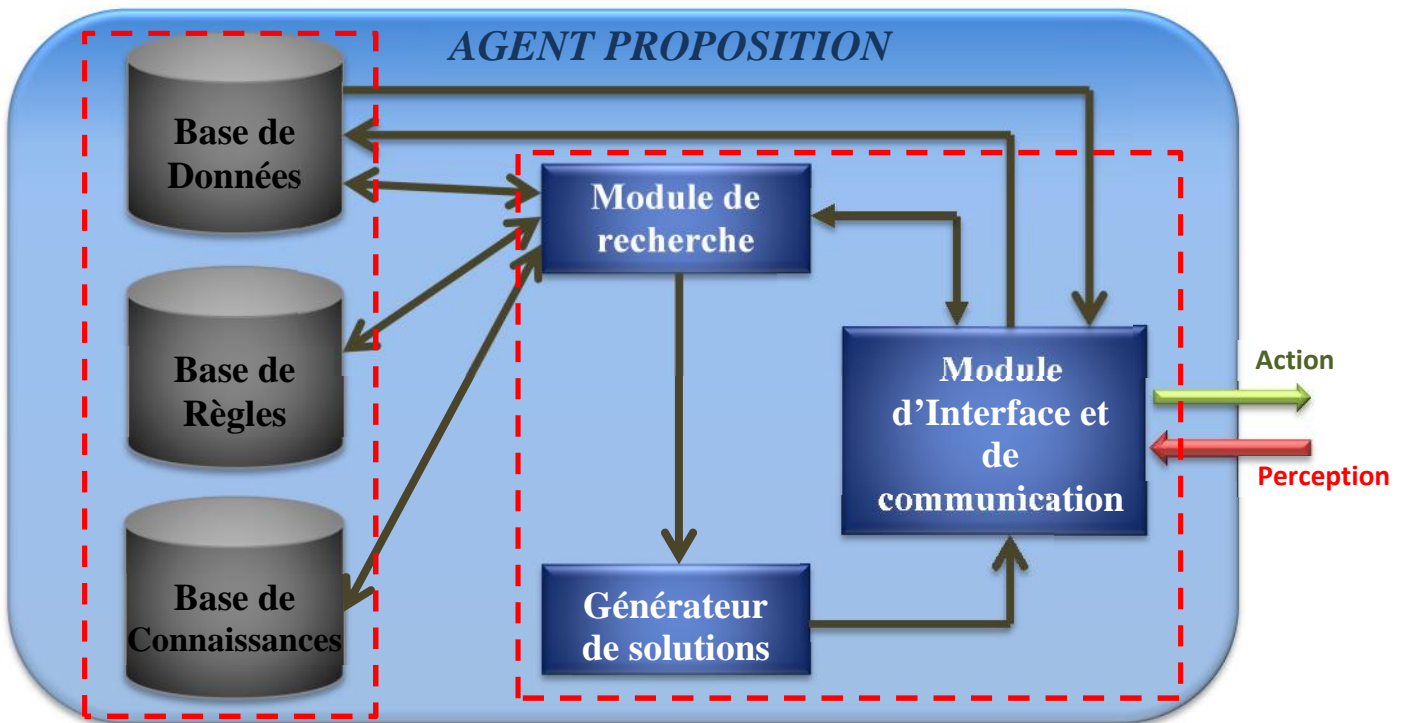


Figure 7 : Structure de l'Agent Proposition

b- Rôles des composants de l'agent proposition :

b1. Module d'interface et de communication :

Ce module joue les mêmes rôles que ceux que nous avons définis pour les autres agents, à l'exception que ce module a un accès à internet pour effectuer des recherches à la demande de l'agent coordinateur. Donc on munit cet agent de tous les protocoles nécessaires pour garantir un bon fonctionnement sur le web.

La raison pour laquelle cet agent est muni d'un accès à internet, est qu'il doit rechercher une solution à un problème donné soit à partir de la mémoire collective, soit à partir des autres SIAD se trouvant sur le web. Cette opération est effectuée à la demande de l'agent coordinateur dès qu'il reçoit la liste des références des pièces défectueuses.

La recherche en question sera dirigée par le module de recherche présenté ci-après.

b2. Module de recherches :

Ce module joue un rôle très important dans ce système, car c'est lui qui permet de lancer et de diriger les recherches de solutions à partir de la mémoire collective ou à partir d'internet et ce par le biais du module d'interface et de communication.

Dès lors que cet agent reçoit une référence relative à une pièce en panne, de la part de l'AC il met à jour sa perception via la BDD. Les règles qui se trouvent dans la BDR lui permettent de réagir face à cette perception. A ce moment, le module de recherche récupère la référence et lance une recherche, en premier lieu, la recherche s'effectue à partir de la MC, si la solution n'existe pas alors la recherche sera redirigée vers le web.

Si nous supposons que la recherche s'avère fructueuse, alors, le résultat de la recherche va être envoyé au générateur de solutions.

b3. Générateur de solutions :

Ce module a pour objet de choisir et de sélectionner la solution la plus adéquate parmi celles qui lui parviennent du module de recherche. Ensuite, cette solution sera examinée et

décortiquée pour construire un plan qui sera transmis à l'AC pour qu'il soit appliqué par l'APr.

c- Rôle de l'agent proposition :

Cet agent joue un rôle primordial dans le SMA, car :

- ✓ Dès que l'AC reçoit la référence et le diagnostic de la ressource en panne, il la transmet à l'AP et va lui demander de rechercher une solution à cette panne ;
- ✓ L'AP, par le biais du module de recherche, va lancer une recherche approfondie pour trouver une solution. En premier lieu à partir de la MC. Si la solution n'existe pas dans la MC. Il va lancer une autre recherche mais cette fois-ci sur le web.
- ✓ Le résultat de ces recherches peut donner lieu à plusieurs solutions, qui seront transmises au générateur de propositions.
- ✓ Le générateur de propositions va analyser toutes les solutions qui s'offrent à lui et choisi l'une d'entre-elles.
- ✓ La solution choisie sera traduite en plan qui sera à son tour envoyé à l'AC.

d- Variables temporelles de l'agent proposition :

A partir de ce que nous venons de décrire, nous pouvons déduire que cet agent développe deux sortes de recherches : la première à partir de la MC ; et la seconde à partir du web. D'où, nous pouvons définir ce qui suit:

- **Temps de recherche d'une solution** (T_{RS}) = $T_{RM} + T_{RW} + T_{IMP}$.

Où : T_{RM} représente le temps de recherche dans la MC.

T_{RW} représente le temps de recherche sur le web. Ce temps de recherche doit être limité pour éviter des temps de recherches infinis.

T_{IMP} représente le temps improductif c'est-à-dire le temps où la connexion au réseau n'est pas disponible, ou bien si l'agent reçoit la référence de la ressource et qu'il est occupé ne pouvant rechercher une solution à la nouvelle panne.

e- Algorithme de fonctionnement de l'agent proposition :

Il reçoit le type de la pièce défectueuse ainsi que sa référence. Il va chercher et renvoyer une solution en consultant le web.

// SearchSolutionOnWeb : Cette fonction envoi une requête aux bases de données web
//pour rechercher une solution à la panne.

```
Type ProposedSolutionRecord = record
    Solved : boolean ;
    Sol : string ;
End ;
Function proposeSolution(ref : string, pann : Integer) : ProposedSolutionRecord;
Var result : ProposedSolutionRecord;
Begin
    Result.Solved = VerifyConnection();
    If(result.Solved ) then
        Result.sol = SearchSolutionOnWeb(ref, pann);
    Return result;
End ;
```

5. Agent Coordinateur (AC) :

a- Architecture de l'agent coordinateur :

L'AC est composé de ce qui suit :

La zone de connaissances comprend,

- *Base de Données ;*
- *Base de règles ;*
- *Base de connaissances ;*
- *Base de cas.*

et la zone de contrôles de l'agent est composée des modules suivant :

- *Module d'interface et de communication ;*
- *Module de recherche ;*
- *Processeur de coordination.*

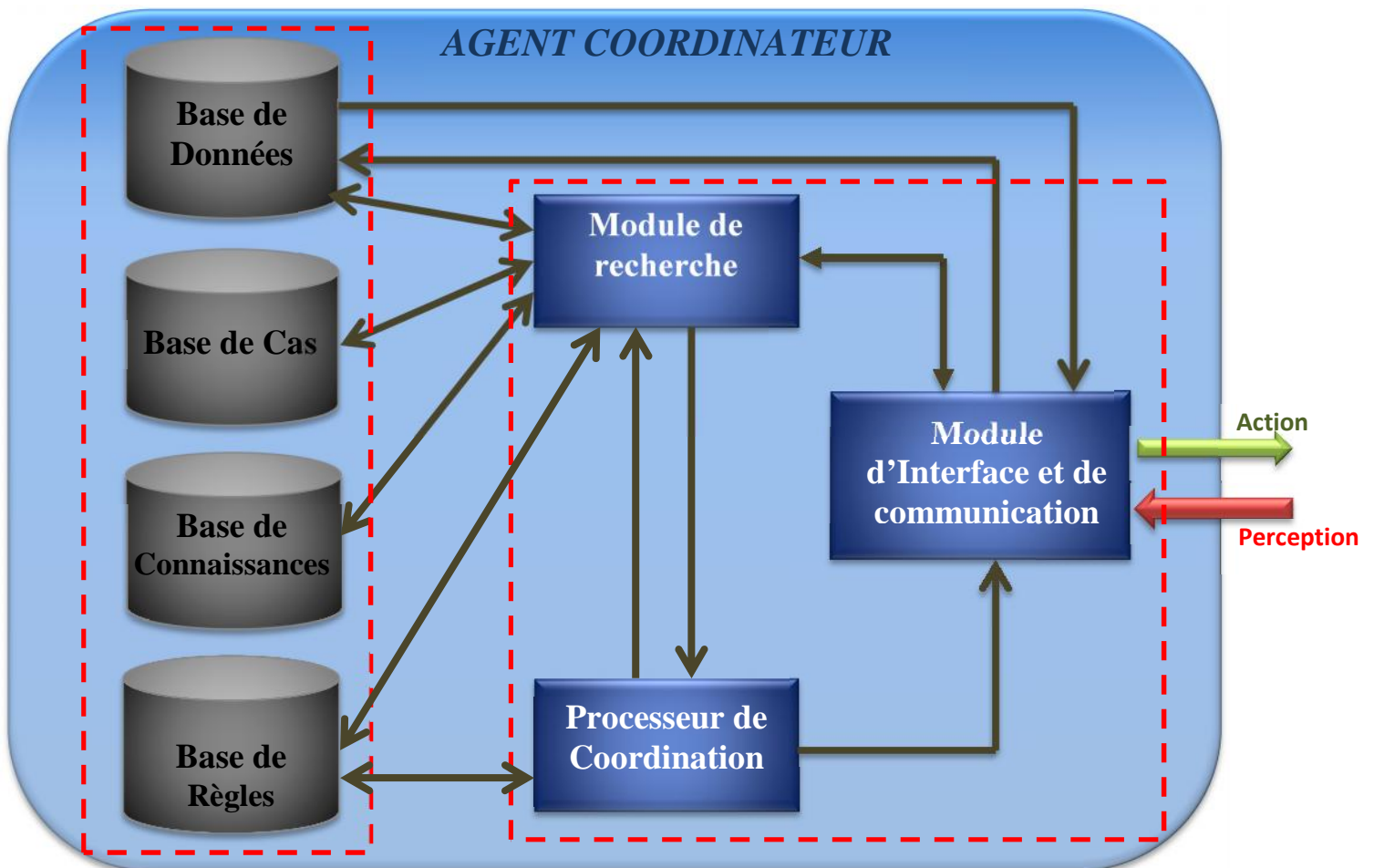


Figure 8 : Structure de l'Agent Coordinateur

b- Rôles des composants de l'agent coordinateur :

Cet agent joue un rôle très important au sein du SMA, il correspond au cœur de cette architecture. En effet, toutes les interactions entre les différents agents du SMA passent par lui.

Nous pouvons remarquer à partir de cette figure, que nous avons procédé à quelques changements au niveau de la zone de connaissances. Ces changements se caractérisent par l'ajout de la **Base De Cas (BDC)** que nous allons décrire dans ce qui suit. L'autre changement est appliqué à la BDR, que nous allons détaillée par la suite.

b1. La base de cas :

Nous avons intégré dans ce système, la base de cas pour pouvoir enregistrer les cas de pannes que nous rencontrons au cours des exécutions du SMA. Cependant l'enregistrement des cas n'est pas identique à celui qu'on fait dans la mémoire collective. En effet dans la base des cas, nous enregistrons pour chaque cas son identifiant (identifiant que nous supposons unique) et les différents temps relatifs à chacun des agents (les temps que nous avons définis pour chaque agent précédemment cité).

La raison pour laquelle nous enregistrons ces temps dans la base des cas et non pas à la MC, est que ces valeurs ne sont vrais que pour ce système. Dans des configurations différentes ces temps peuvent changés et donc ils ne servent à rien dans un autre système. Par contre dans notre système ces valeurs nous servent de métrique pour évaluer les stratégies de coordination.

b2. La base de règles :

Nous avons définis précédemment le rôle de cette base. Mais pour l'agent coordinateur, cette base a été augmentée par les stratégies de coordination relatives au fonctionnement du SMA. Ces stratégies seront présentées un peu plus loin dans ce chapitre.

b3. Le module de recherches :

Dans cette configuration, ce module a pour but de procéder à des recherches :

- Premièrement, à partir de la base des cas pour déterminer si la panne détectée est répertoriée dans cette base ou pas. Si cette panne est déjà répertoriée le module de recherche envoie l'identifiant de la panne au processeur de coordination pour qu'il puisse créer un plan et envoyer à l'AP cet identifiant pour que ce dernier récupère la solution de la panne à partir de la MC.
- Deuxièmement, si la panne n'est pas répertoriée dans la BDC. Le module de recherches commence par chercher la solution au niveau de la MC, car il est possible que d'autres SMA aient consignés une solution à cette panne (ce principe est l'une des raisons principales qui nous ont poussé à intégrer la MC dans notre SIAD ; pour plus d'informations, voir la section relative à la MC). Si la MC ne fournit pas de solution à cette panne, alors, le module de recherche retourne un résultat négatif au processeur de coordination.

b4. Le processeur de coordination :

Ce module est très important dans ce système, car c'est lui qui permet :

- De coordonner les actions des agents du SMA ;
- D'établir un plan permettant d'atteindre l'objectif commun sans affaiblir les agents tout en évitant toute situation conflictuelle ;
- De déclencher les règles qui régissent les interactions entre l'AC et les autres agents composant notre SMA ;
- De déclencher les stratégies de coordination se trouvant dans la BR, pour chaque cas qu'il rencontre ;

- Enfin, de consigner tous les temps relatifs à chaque agent dans sa BDC pour permettre une étude statistique.

c- Variables temporelles de l'agent coordinateur :

Pour l'AC, on ne peut définir qu'un seul temps important et qui résulte de la sommation de tous les temps précédemment définis pour chaque agent. Ce temps que nous désignerons par **Temps de Traitement Total (T_{TT})** aura pour effet de nous permettre d'avoir une idée claire sur le temps de traitement de chaque panne et pour chacune des stratégies que nous définirons par la suite.

d- Algorithme de coordination :

Cet algorithme est très important car il décrit non seulement le fonctionnement interne de l'AC mais en plus, il décrit chacune des interactions qui s'échangent entre les agents composants notre SMA.

```
// GetPannListAndEmptyBuffer : cette procédure va copier le contenu du buffer contenant
//les PannRef envoyés par l'agent production au serveur dans une nouvelle liste et vide le buffer
// SaveInDataBase : Cette procedure va enregistrer la solution dans la base de données ainsi que
//dans la mémoire collective
```

Procédure coordination()

```
Var    pannList : PannRefList ;
        Sol : Solution ;
        pannCounts : pannCountList ;
        analyseTime, resTime, propoTime : Time ;
        i, j : integer;
        resResult : ressourceResult;
        propResult : ProposedSolutionRecord ;
        resConn : Boolean;

Begin
    While(true)
        Begin
            pannList = GetPannListAndEmptyBuffer();
            if(Count(pannList))
                continue;
            analyseTime = getTime() ;
            pannCounts = GetPannFiFoList(pannList);
            analyseTime = GetTime() - analyseTime ;
            for i = 1 to Length(pannCounts) to
                begin
                    resTime = GetTime() ;
                    resResult = searchForRessource(pannCounts[i].Pann_Id);
                    resTime = GetTime() - resTime ;
                    if(resResult.Conn)
                        begin
                            propoTime = GetTime() ;
```

```
propResult = proposeSolution(resResult.ref, pannCounts[i]. Pann_Id );
propoTime = GetTime() – propoTime ;
    end
else
    propoTime = 0 ;
for j = 1 to Length(pannList) do
    begin
    if(pannList[j].Pann_Id == pannCounts[i].Pann_Id)
        begin
        Sol.Solved = resResult.Conn and propResult.Solved ;
        Sol.Ressource_id =pannList[j].Ressource_id ;
        Sol.Pann_Id = pannList[j].Pann_Id ;
        Sol.to = pannList[j].to ;
            If(Sol.Solved) then
                Begin
                Sol.t_analy = analyseTime ;
                Sol.t_ress = resTime ;
                Sol.t_prop = propoTime ;
                Sol.Solution = propResult .Sol ;
                End
            Else
                Begin
                Sol.Impro_ress = resTime ;
                Sol.impro_prop = propoTime ;
                End ;
                Resolve(Sol) ;
                SaveInDataBase(Sol) ;
            end ;
        end ;
    end ;
end ;
End ;
```

Après avoir décrit le fonctionnement et le rôle de chaque agent composant notre SMA. Nous allons consacrer la section suivante à la description du fonctionnement général du système.

IV. Fonctionnement général du Web-DSS:

1. Déroulement du système :

Notre système est un SIAD à base de web, spécialisé dans l'aide à la résolution de problème. Dans cette section nous allons résumer le fonctionnement de ce système. Cela nous permettra de mettre par la suite en évidence les stratégies de coordination au sein du SMA.

Pour la détection et la résolution d'une panne :

- L'Agent Production (**APr**) va récupérer toutes les informations concernant les ressources en pannes dans l'atelier et il va les envoyer aux agents qui se trouvent dans le serveur et plus précisément à l'Agent Coordinateur (**AC**).
- Au niveau du serveur, l'Agent Coordinateur (**AC**), va recevoir cette liste, laquelle sera envoyée à l'Agent Analyseur (**AA**) qui va l'analyser et la filtrer, ensuite elle va être renvoyée à l'Agent Coordinateur (**AC**).
- Ce dernier va chercher dans sa base de cas si la panne en question est déjà consignée.
- Si c'est le cas, alors l'**AC** va directement envoyer les informations de la ressource à l'agent proposition (**AP**) tout en lui demandant de chercher la solution de la panne à partir de la **MC**.
- Par contre si la panne n'est pas consignée, alors l'**AC** va envoyer les informations relatives à cette panne à l'agent ressources (**AR**).
- L'**AR** va procéder à une recherche sur le web pour essayer de trouver les références de la ressource en panne pour déterminer de quelle ressource s'agit-il.
- Le résultat de cette recherche sera envoyé à l'**AC**, qui à son tour va le transmettre à l'**AP** pour lancer une recherche sur le web pour trouver une solution à la panne.
- Si la recherche est fructueuse alors l'**AP**, va transmettre la meilleure solution à l'**AC** qui va la consignée dans la **MC**, et dans sa base de cas il va consigner uniquement les temps de recherche relatifs à la panne. Ceci avant de l'envoyer à l'**APr** pour qu'il mette en place la solution trouvée.
- Cependant, si la recherche de la solution s'avère infructueuse, alors l'**AP** va chercher et transmettre à l'**AC** un lien lui permettant de commander directement la ressource en question. Ce lien sera transmis ensuite à l'**APr** qui le proposera au technicien de maintenance au niveau de la station client (atelier de production) pour commander une nouvelle pièce.

2. Diagramme des séquences : les échanges entre les agents :

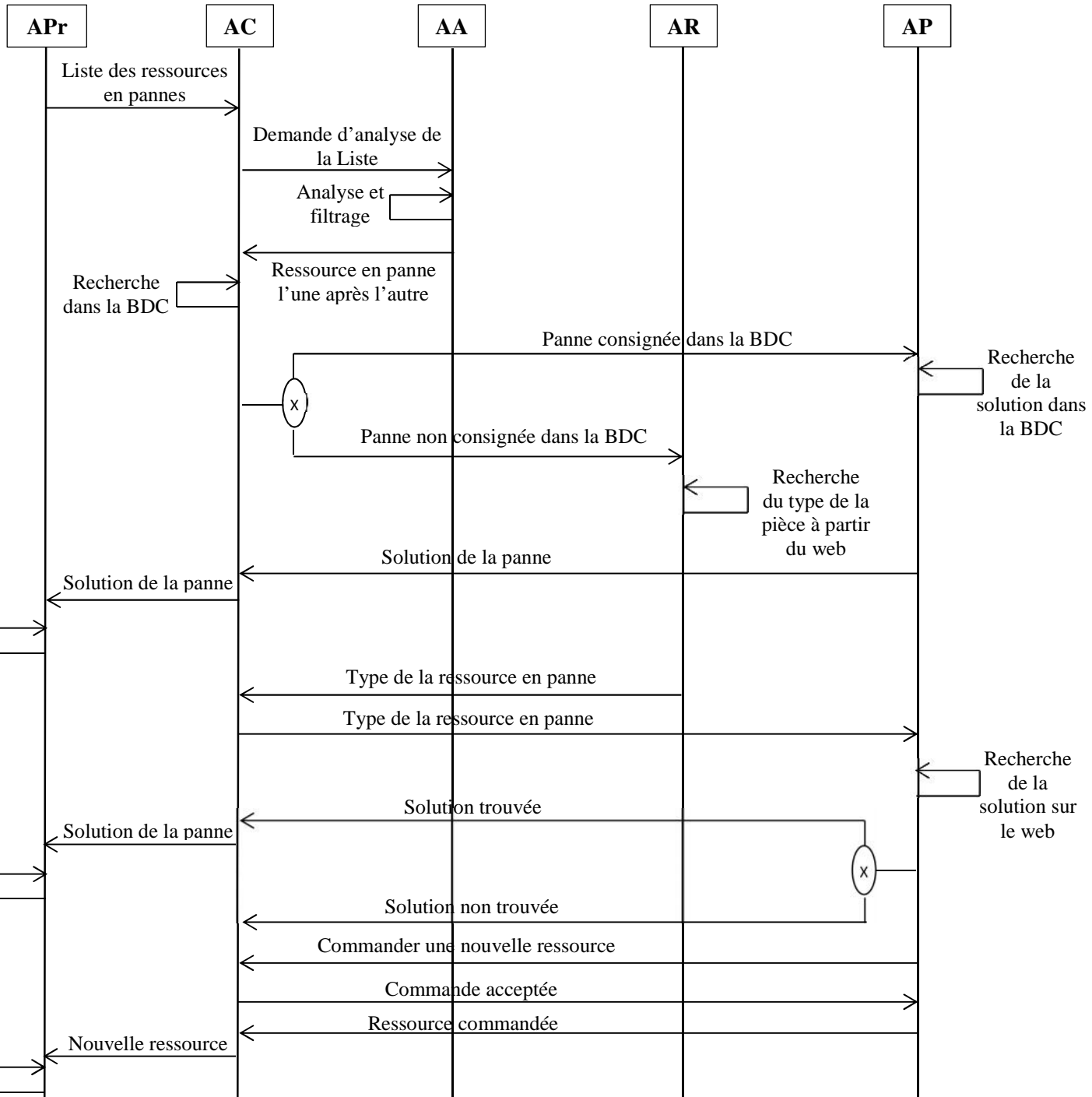


Figure 9 : Diagramme des séquences : échanges entre les différents agents

V. Les stratégies de coordination :

1. Stratégies de coordination :

Nous avons indiqué au cours du chapitre relatif aux SMA et à la coordination (chapitre 4), qu'il existe deux grandes familles de stratégies pour la coordination :

a- La coordination centralisée :

Dans ce type de système, il existe un agent coordinateur qui se charge de la coordination entre les différents agents et parfois dans les systèmes de production, il est même considéré comme étant un superviseur.

Cette stratégie de coordination permet d'éviter toute situation conflictuelle, puisque aucun agent n'a la possibilité d'interagir avec un autre sans passer par l'agent coordinateur.

b- La coordination décentralisée :

La particularité d'un tel système réside dans le fait, qu'il n'existe ni agent coordinateur ni hiérarchie entre les différents agents composant le SMA.

La coordination est mise en œuvre par l'ensemble des agents qui compose le système. En effet, les agents partagent leurs plans et leurs buts avec un niveau d'abstraction correcte et acceptable. Cela permet aux agents de se coordonner et de travailler ensemble pour atteindre un idéal commun et éviter toute situation conflictuelle.

2. Techniques de coordination :

Pour les techniques de coordination, nous pouvons citer principalement :

a- La coordination par structure organisationnelle :

Dans ce type de coordination, il existe des relations de contrôle entre les différents agents. Une relation qui peut être soit hiérarchique soit plate.

Dans les deux cas, nous dirons que chaque agent agit sur un espace de recherche qui lui est propre et dans lequel il agit en fonction de sa perception. Dans ce type de système la coordination se met en place toute seule car aucun agent n'interfère dans l'espace environnemental des autres agents et donc il ne peut pas y avoir de situation conflictuelle et ce quel que soit la stratégie de coordination mise en place c'est-à-dire centralisée ou décentralisée.

Nous pouvons citer comme exemple illustratif, une gestion de réseaux dans laquelle il existe deux agents A et B. on assigne à l'agent A la gestion de la zone 1 et à l'agent B, la zone 2. Chacun d'eux gère sa zone en fonction de sa perception sans se soucier des autres zones.

b- La coordination par échange de méta-informations :

N'ayant qu'une vue partielle de l'environnement dans lequel ils évoluent, les agents d'un tel système construisent une vue plus globale, en échangeant des informations relatives à leur perception partielle de l'environnement et travaillent ensemble pour atteindre un but

commun. La coordination se construit petit à petit jusqu'à ce que la perception de chaque agent soit connue par ses pairs et donc les buts partiels à atteindre de chacun des agents et mise en place pour atteindre la cohérence globale. Cette technique ne peut être mise en place que par une stratégie de coordination décentralisée.

c- La coordination par planification multi-agents :

Dans cette technique, l'espace de recherche ou bien l'environnement dans lequel évoluent les agents est connu à l'avance c'est-à-dire avant l'exécution du système. Ceci nous permet d'assigner à chaque agent sa perception, ses buts partiels et comment il doit se dérouler pour atteindre ces buts.

Cette technique de par le fait, qu'elle permet de tout contrôler, est excellente pour éviter tout conflit potentiel entre les agents. Mais il existe un inconvénient majeur dans le principe de cette technique relatif aux changements permanents de l'environnement. Ces changements contraignent l'élaboration des plans pour les agents qu'à très court terme.

Le deuxième inconvénient, concerne l'élaboration des plans. En effet, les plans que nous devons élaborés doivent prendre en considération tous les choix et toutes les contraintes possibles avant l'exécution du système et ce avant même son exécution, ce qui peut rendre le système très lourd.

Enfin, nous dirons que pour mettre en place une telle technique il faut avoir une grande quantité de ressources pour éviter tout conflit, ce qui est pratiquement impossible à réaliser.

3. Stratégies de coordination adoptées par notre système :

Le rappel que nous avons présenté au cours de la section suivante nous permet de situer notre système et les stratégies de coordination qui s'y rattachent.

Nous pouvons définir la coordination au sein de notre système comme étant une coordination centralisée mise en place par la technique de coordination dite « la coordination par structure organisationnelle ».

Pour ce qui est des stratégies retenues au sein de notre système, nous pouvons dénombrer celles-ci :

a- Stratégie de coordination locale :

Nous définissons pour cette stratégie une coordination et un traitement qui se font localement par rapport au SIAD à base de web que nous avons développé. En effet, la résolution des problèmes se fait au sein du SIAD sans avoir recours aux solveurs qui se trouvent sur le web.

Si nous reprenant le déroulement de notre système présenté à la section **IV**, et l'algorithme de coordination précédemment présenté, nous pouvons nous apercevoir qu'il existe deux façons de résoudre un problème. La première, celle que nous désignons par

traitement local, est relative à la recherche d'une solution au problème uniquement au sein de la mémoire collective.

b- Stratégie de coordination externe:

La seconde façon de rechercher une solution à un problème dans notre système, c'est de procéder à une recherche directement à partir du web (d'autres SIAD et MC sur le web).

c- Métriques de comparaison entre les deux stratégies :

L'évaluation de la performance de chacune des stratégies de coordination précédemment citées, est réalisée à l'aide des variables temporelles que nous avons implémentées pour chacun des agents composants notre SMA.

Ces variables nous permettent de déterminer :

- ✓ Quelle est la meilleure stratégie pour un même problème.
- ✓ Quel agent à consommer le plus de temps pour effectuer sa tâche.
- ✓ Quelles sont les raisons qui font que telle ou telle stratégie est moins efficace que l'autre. Exp. Problèmes liés à la connexion, indisponibilité des ressources, ou bien taux d'occupation des agents.

VI. Syntaxe des requêtes échangée entre agents :

Le modèle de requêtes que nous présentons est utilisé dans notre approche pour formaliser les besoins de l'utilisateur, mais aussi le contenu des messages échangés entre agents.

Parce qu'il est employé dans les interactions humain-agent et inter-agents, il comprend plusieurs types de requêtes, dont des questions, des commandes, des assertions, etc.

Une requête est une paire :

$$r = (,)$$

Où est un performatif et est le contenu (ou l'objet) de la requête.

Le performatif de la requête peut ainsi prendre les valeurs suivantes :

- **What-is** : pour les requêtes qui représentent les questions à propos des données de l'agent.
 - **Assert-is** : pour les assertions à propos des valeurs des données de l'agent (en réponse aux requêtes What-is).
 - **Order** : pour commander l'exécution d'une action étant donné un ensemble de paramètres, i.e. moyennant l'envoi d'un évènement.
 - **Ack** : pour les confirmations d'exécution d'actions (en réponse aux requêtes Order).
 - **Assert-cannot** : pour exprimer qu'un agent est incapable d'exécuter une action.
 - **Assert-can** : pour exprimer qu'un agent peut exécuter une commande, moyennant certains paramètres.
 - **Unknown** : pour exprimer qu'une entité demandée n'a pas été reconnue de l'agent.
 - **Missing** : pour communiquer les valeurs d'attributs que l'agent n'a pas pu interpréter.
- Cela se produit, lorsqu'une requête reçue présente des dépendances avec d'autres requêtes.

Ainsi, {What-is, Assert-is, Order, Ack, Assert-cannot, Assert-can, Unknown, Missing} est l'ensemble de performatifs que nous utilisons dans la formalisation des requêtes de l'utilisateur ainsi que dans la représentation des réponses des agents.

VII. Langage de communication entre agents :

Un langage de communication est conçu pour l'échange, entre logiciels (agents), d'informations, de connaissances ou encore de services [Koning et Pesty, 2001]. Dans notre architecture, les agents sont munis d'un ACL leur permettant de communiquer à propos des besoins de l'utilisateur et de construire des contenus de messages exprimant des informations à propos de leurs capacités [Charif et Sabouret, 2004].

1. Envoi des messages :

Un agent envoie un message dans deux cas de figure :

1. S'il a reçu un message nécessitant d'être traité, il construit alors un message pour répondre à celui qu'il a reçu et l'envoi à l'émetteur de ce dernier ;
2. S'il exécute une action dont les effets consistent en l'envoi d'un ou plusieurs messages, alors il construit ces messages et les envoient au(x) destinataire(s) spécifié(s).

2. Contenu des messages :

En contraste avec les messages KQML et FIPA-ACL, où est exprimée une seule valeur d'illocution appliquée à un seul contenu propositionnel, nos agents peuvent au contraire traiter un ensemble de requêtes, chacune comportant un performatif et un contenu.

En effet, les langages KQML et FIPA-ACL ont été conçu pour des besoins applicatifs, le plus souvent liés à l'échange d'informations et de connaissances. Ces modèles ne sont cependant pas adaptés à des interactions de types conversationnel, en particulier lorsqu'il s'agit de communautés impliquant agents artificiels et humains.

C'est dans cette optique que nous avons conçu notre ACL. Il est un moyen pour les agents non seulement de communiquer et d'exprimer une large gamme d'informations, mais aussi de pouvoir directement traiter les besoins de l'utilisateur avec lequel ils interagissent. En effet, ces besoins peuvent être complexes et doivent parfois être formalisés en un ensemble de requêtes qui se complètent et dont certaines dépendent des résultats des autres. Ces besoins peuvent ainsi être manipulés par les agents, moyennant leur ACL. Leur traitement s'effectuera ensuite, en fonction de la capacité des agents à répondre à tout ou partie de ces besoins, soit au niveau de leur module de raisonnement, i.e. le RPM, soit au niveau du protocole de coordination.

3. Syntaxe des messages :

Pour construire les messages échangés entre les agents, ceux-ci utilisent un langage de communication appelé FIPA-ACL [Fipa, 2002]. Un message FIPA-ACL comprend un niveau de communication et un niveau contenu :

1. **Le niveau communication** permet d'identifier les agents émetteur et récepteur, et précise l'identifiant du message ainsi que l'identifiant de celui auquel il est construit en réponse (i.e. le message d'origine). De plus, ce niveau affecte un identifiant de la conversation à laquelle appartient ce message, i.e. c'est un identifiant commun à l'ensemble des messages

générés pour la satisfaction d'un même ensemble de besoins de l'utilisateur, et les identifiants des requêtes auxquelles le message répond.

2. Le **niveau contenu** englobe le contenu du message. Celui-ci est constitué d'un ensemble de requêtes dont la syntaxe et la sémantique ont précédemment été définies.

Chaque message a ainsi la syntaxe suivante :

- **sender** est l'identifiant de l'agent émetteur ;
- **receiver** est l'identifiant de l'agent récepteur ;
- **conv-id** est l'identifiant de la conversation dans laquelle le message **m** est impliqué.

Une conversation réfère l'ensemble des messages générés pour satisfaire un même ensemble de besoins de l'utilisateur ;

- **reply-with** est l'identifiant du message **m**;
- **in-reply-to** est l'identifiant du message auquel **m** est construit en réponse ;
- **req-id** est un tableau regroupant les identifiants des requêtes auxquelles **m** répond ;
- **content** est le contenu du message.

VII. Conclusion :

Nous avons proposé dans ce chapitre l'architecture générale de notre application, ainsi que la structure des agents ainsi que le diagramme des séquences permettant de visualiser l'ensemble du fonctionnement de notre système.

Ensuite, nous avons présenté les stratégies de coordination ainsi que les différentes variables temporelles, qui nous servent de métrique pour évaluer et déterminer quelle stratégie est la plus appropriée pour résoudre un problème donné.

Aussi, nous avons défini le modèle de requêtes permettant à la fois de formaliser les requêtes de l'utilisateur, mais également le contenu des messages échangés par les agents. Nous avons vu que ce modèle permettait de formaliser à la fois des requêtes en services, des requêtes ou assertions à propos des capacités des agents mais aussi des requêtes présentant des dépendances avec d'autres, couvrant ainsi plusieurs domaines d'applications tels que le web.

Dans ce chapitre, nous avons également présenté l'ACL utilisé par les agents pour qu'ils puissent à la fois communiquer mais aussi gérer directement les besoins de l'utilisateur avec lequel ils interagissent.

Dans le chapitre suivant, nous allons proposer des scénarios d'exécution des modules implémentés avec la plate-forme JADE.

CHAPITRE V

IMPLEMENTATION

I. Introduction :

Dans ce dernier chapitre, nous allons décrire l'environnement d'implémentation et la structuration logique, enfin quelques imprimés écrans d'illustration de notre application.

L'application développée englobe trois parties : la technologie J2EE pour le Web, la plate-forme JADE pour un environnement SMA, une application développée sous JBuilder qui présente l'Atelier.

II. La technologie J2EE :

J2EE est l'acronyme de Java 2 Enterprise Edition. Cette édition est dédiée à la réalisation d'applications pour entreprise.

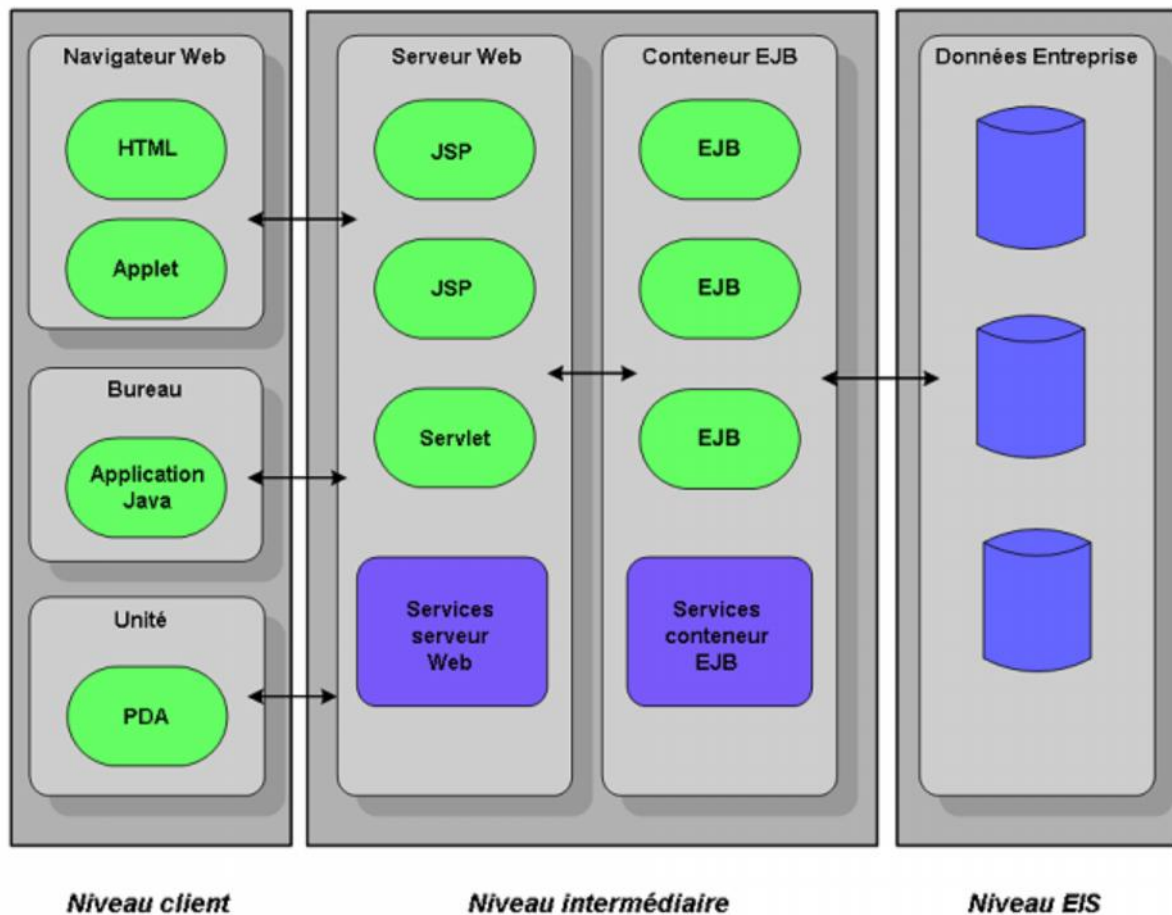


Figure 1 : Architecture générale du J2EE

Le client, dans une application J2EE, peut être une page HTML ou un applet s'exécutant dans un navigateur, une application Java sur une machine de bureau ou même un client Java sur un périphérique portable, comme un assistant numérique personnel (ANP) ou un téléphone cellulaire.

Le niveau intermédiaire peut contenir des pages JavaServer™ ou des servlets exécutés sur un serveur Web. Ces éléments constituent généralement la logique de présentation côté serveur. Un conteneur EJB fournit un environnement d'exécution pour les Enterprise JavaBeans™, qui contient la logique métier de l'application. Un serveur Web et un conteneur

EJB (Enterprise JavaBeans) fournissent tous deux des services aux composants qui s'exécutent sur eux. Puisque ces services sont toujours disponibles, les programmeurs n'ont pas à les inclure dans les composants qu'ils écrivent.

Le niveau de système d'informations d'entreprise (EIS, Enterprise Information System) est un référentiel pour les données de l'entreprise. Il est composé généralement de données d'un système de base de données relationnelle.

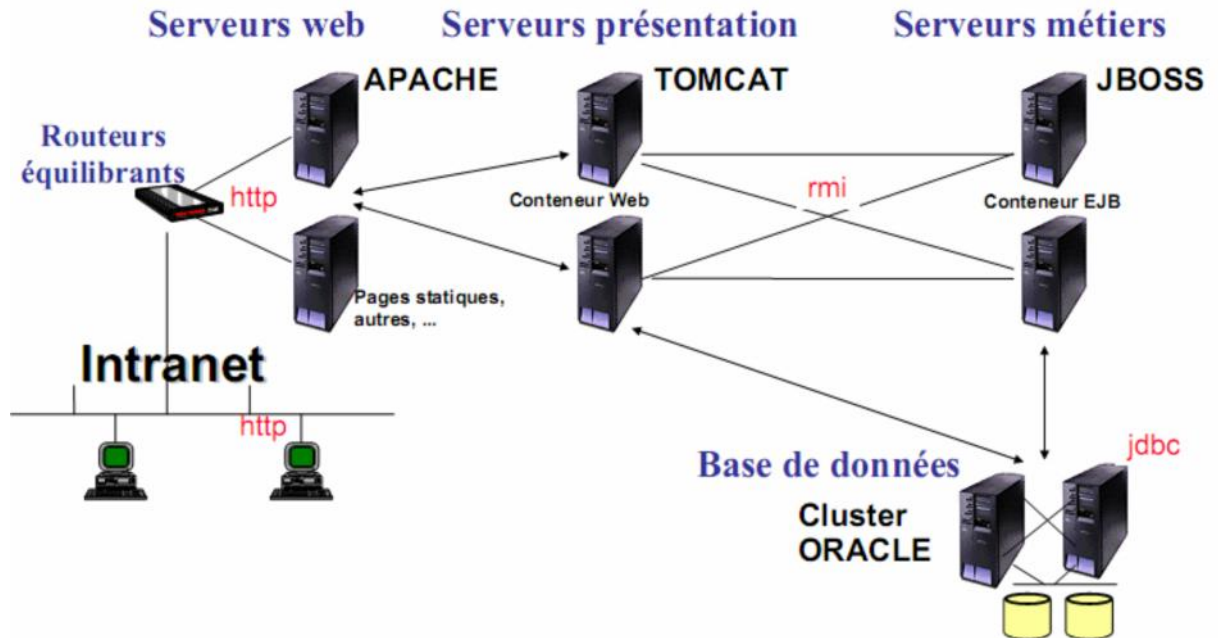


Figure 2 : Principe de fonctionnement du J2EE

1. les fonctions couvertes par J2EE

L'édition Entreprise de JBuilder comprend de nombreuses fonctions pour aider votre équipe à développer des applications J2EE. Voici les technologies dont dispose JBuilder pour vous aider à développer le niveau client :

a. Technologies du niveau client

- **Applets** : Les applets sont un type spécial d'application Java qui sont téléchargées et exécutées par un navigateur Web sur une machine client. Pour commencer le développement d'une applet dans JBuilder, utilisez l'expert Applet. Pour plus d'informations sur l'utilisation des applets, voir "Utilisation des applets" dans le *Guide du développeur d'applications web*.

- **Applications d'interface utilisateur Java** : JBuilder comporte plusieurs fonctions qui peuvent vous aider à développer une application s'exécutant sur une machine client. Commencez une application Java en utilisant l'expert Application. Poursuivez la conception de votre interface utilisateur à l'aide du concepteur d'interface utilisateur de JBuilder. Créez votre interface utilisateur en ajoutant des composants d'interface utilisateur à partir de la palette de composants de JBuilder. Pour plus d'informations sur le concepteur d'interface utilisateur de JBuilder, voir "Conception visuelle dans JBuilder" dans *Conception d'applications avec JBuilder*.

b. Technologies du niveau intermédiaire

Voici les technologies J2EE du niveau intermédiaire qui utilisent un serveur Web :

- **Servlets** : Une servlet est une application Java côté serveur pouvant traiter les requêtes issues des clients. Les servlets répondent aux requêtes en générant des sorties dynamiques renvoyées au client. Vous pouvez commencer à développer des servlets à l'aide de l'expert Servlet de JBuilder. Pour plus d'informations sur les servlets et leur développement.

- **Pages JavaServer (JSP)** : Les pages JavaServer (extension de la technologie servlet) offrent une façon simplifiée de développer des servlets. Comme les servlets, elles génèrent des sorties dynamiques qui sont renvoyées au navigateur Web du client.

Commencez le développement de pages JavaServer avec l'expert Page JavaServer de JBuilder.

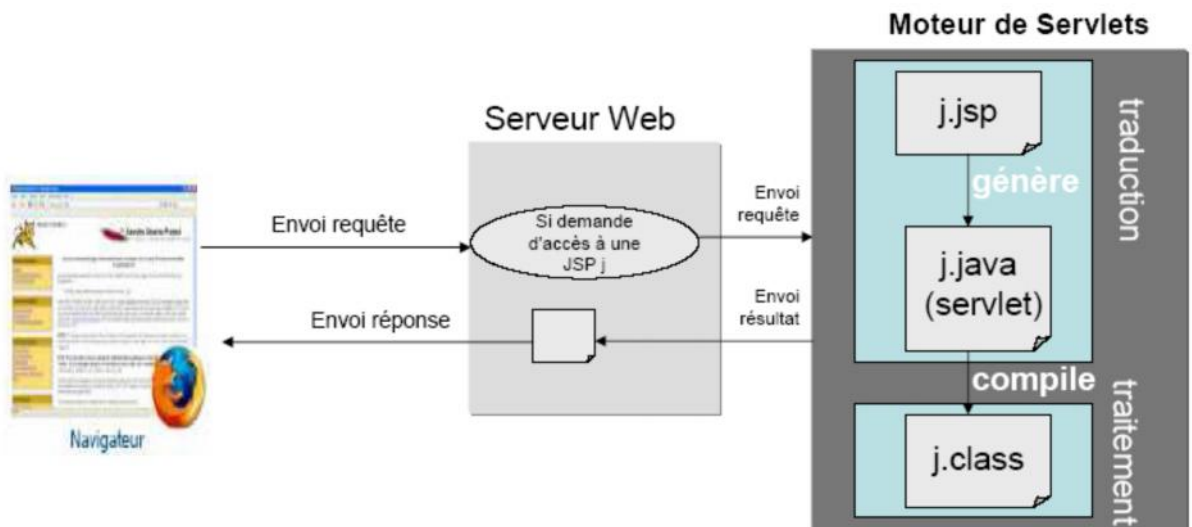


Figure 3 : fonctionnement d'une JSP

Voici les technologies J2EE du niveau intermédiaire qui utilisent un conteneur EJB :

- **Enterprise JavaBeans (EJB)** Les Enterprise JavaBeans sont des composants côté serveur qui contiennent la logique métier de l'application. JBuilder vous assiste dans la construction des composants EJB 1.x et EJB 2.0.

c. Autres technologies J2EE

Sans être confinées dans un niveau d'architecture particulier, ces technologies contribuent au bon déroulement des opérations :

- **JDBC (Java Database Connectivity)** : JDBC est la norme utilisée pour accéder à votre base de données sur le niveau des systèmes d'information d'entreprise (Enterprise Information Systems, EIS). JDBC définit une API Java que vous utilisez pour écrire des instructions SQL envoyées à votre base de données.

JBuilder inclut DataExpress, une bibliothèque de composants pour l'accès aux données de votre base de données. Elle connecte votre application à votre base de données à l'aide de pilotes JDBC.

JBuilder inclut également JDataStore, une bibliothèque de composants et une base de données imbriquée entièrement Java. Vous accédez à JDataStore par l'intermédiaire de JDBC.

Les beans entité, qui accèdent aux lignes de votre base de données, vous connectent également à vos données via JDBC.

- **JMS (Java Message Service)** : JMS est un service de messagerie d'entreprise qui achemine les messages entre les composants et s'exécute dans une application distribuée.

- **JNDI (Java Naming and Directory Interface)**: Tous les serveurs J2EE utilisent JNDI, un service de nommage Java utilisé pour localiser les objets distribués.

- **XML (Extensible Markup Language)** : Bien que XML ne soit pas réellement une technologie J2EE, ce langage est largement utilisé par les technologies J2EE. Par exemple, les descripteurs de déploiement des composants Web et des enterprise beans sont écrits en XML. Ces descripteurs décrivent le comportement des composants lorsqu'ils sont déployés. JBuilder comprend plusieurs fonctionnalités XML qui vous aident à accomplir les tâches de programmation communes que vous pouvez rencontrer dans vos projets J2EE.

III. L'environnement de développement

1. Choix de langage de programmation

Java est un langage de programmation récent (les premières versions datent de 1995) développé par Sun Microsystems. Il est fortement inspiré des langages C et C++.

On a choisi d'utiliser le langage Java par l'assistance de Jbuilder qui va nous permettre d'informatiser nos idées : construire notre application (acquérir dynamiquement les données, afficher graphiquement le résultat).

2. JBuilder : environnement de développement intégrant Java

JBuilder est un environnement intégré Java qui, associé à un serveur d'applications de sociétés comme Borland, BEA, IBM, SUN et Sybase simplifie et accélère considérablement le développement d'applications J2EE.

IV. La plate-forme JADE

Le meilleur moyen pour implémenter un système multi-agent (SMA) est d'utiliser une plate-forme multi-agent. Une plate-forme multi-agent est un ensemble d'outils nécessaires à la construction et à la mise en service d'agents au sein d'un environnement spécifique.

Ces outils peuvent servir également à l'analyse et au test du SMA ainsi créé. Ces outils peuvent être sous la forme d'environnement de programmation (API) et d'applications permettant d'aider le développeur. Nous allons étudier dans cette partie la plate-forme JADE (Java Agent DEvelopment framework).

1. Brève description de JADE

JADE (Java Agent DEvelopment framework) est une plate-forme multi-agent créée par le laboratoire TILAB et décrite par Bellifemine et al. Dans [Bellifemine et al, 1999] [Bellifemine et al, 2000]. JADE permet le développement de systèmes multi-agents et d'applications conformes aux normes FIPA [Fipa, 2002]. Elle est implémentée en JAVA et fournit des classes qui implémentent « JESS » pour la définition du comportement des agents. JADE possède trois modules principaux (nécessaire aux normes FIPA).

- *DF* « *Director Facilitator* » fournit un service de « pages jaunes » à la plate-forme ;
- *ACC* « *Agent Communication Channel* » gère la communication entre les agents ;

- *AMS* « *Agent Management System* » supervise l'enregistrement des agents, leur authentification, leur accès et l'utilisation du système.

Ces trois modules sont activés à chaque démarrage de la plate-forme.

2. La norme FIPA

La FIPA (Foundation for Intelligent Physical Agents) est une organisation à but non lucratif fondée en 1996 dont l'objectif est de produire des standards pour l'interopération d'agents logiciels hétérogènes. Par la combinaison d'actes de langages, de logique des prédicats et d'ontologies publiques, la FIPA cherche à offrir des moyens standardisés permettant d'interpréter les communications entre agents de manière à respecter leur sens initial, ce qui est bien plus ambitieux que XML, qui ne standardise que la structure syntaxique des documents. Afin d'atteindre ce but, le FIPA émet des standards couvrant :

- Les applications (applications nomades, agent de voyage personnel, applications de diffusion audiovisuelles, gestion de réseaux, assistant personnel...);
- Les architectures abstraites, définissant d'une manière générale les architectures d'agents ;
- Les langages d'interaction (ACL), les langages de contenu (comme SL, CCL, KIF ou RDF) et les protocoles d'interaction ;
- La gestion des agents (nommage, cycle de vie, description, mobilité, configuration);
- Le transport des messages : représentation (textuelle, binaire ou XML) des messages ACL, transport (par IIOP, WAP ou HTTP) de ces messages.

Ces standards évoluent, et sont régulièrement mis à jour, ainsi que de nouveaux standards qui sont nouvellement proposés. Les standards qu'édicte la FIPA ne constituent pas vraiment une plate-forme de construction multi-agents. Ce n'est pas non plus l'objectif que s'est fixé la FIPA. Tout au plus, la FIPA normalise une plate-forme d'exécution standardisée dans un but d'interopérabilité. Ces normes s'appliquent donc pour la plupart en phase de déploiement. Elles n'abordent pas les phases d'analyse ni de conception. Elles peuvent cependant guider certains choix d'implémentation.

3. Architecture logicielle de la plate-forme JADE

JADE reprend donc l'architecture de l'Agent Management Reference Model proposé par FIPA. Les différents modules présentés dans la figure suivante sont présentés sous forme de services. Les services de base proposés sont le Directory Facilitator (DF) et l'Agent Management System (AMS). Il est possible de lui demander de tenir en plus le service de Message Transport Service (MTS) pour communiquer entre plusieurs plates-formes. Mais ce service sera chargé à la demande pour ne conserver par défaut que les fonctionnalités utiles à tout type d'utilisation.

L'agent est l'acteur fondamental de la plate-forme, un Agent Identifier (AID) identifie un agent de manière unique. Le DF est un composant qui fait office d'annuaire. C'est un service de « pages jaunes » qui permet de mettre en relation les agents avec leurs compétences. Un agent peut enregistrer ses compétences dans le DF ou interroger le DF pour connaître les compétences proposées par les autres agents. L'AMS est un autre composant important car il contrôle l'accès et l'utilisation de la plate-forme et maintient un répertoire

contenant les adresses de transport des agents de la plate-forme. Ce service est plus un service de type « pages blanches » qui effectue la correspondance entre l'agent et l'AID.

Chaque agent doit s'enregistrer à un AMS pour avoir un AID. Il n'y a qu'un AMS par plateforme.

Le MTS est une méthode par défaut de communication entre agents de différentes plates-formes. Cela permet l'interconnexion entre systèmes hétérogènes ou tout au moins de système ne communiquant pas de la même façon. L'Agent Platform (AP) constitue l'infrastructure physique sur laquelle se déploient les agents. Il contient le DF, l'AMS et le MTS. Lorsqu'on parle de l'AP, on inclut souvent le matériel électronique, l'OS, le software et les composants cités ci-dessus avec les agents. Enfin, l'Agent Identifier (AID) est un identifiant précis d'un agent. On lui donne plusieurs paramètres tels que l'adresse de transport, l'adresse de service de résolution de nom, ... Un exemple est : name@HAP (Home Agent Platform)

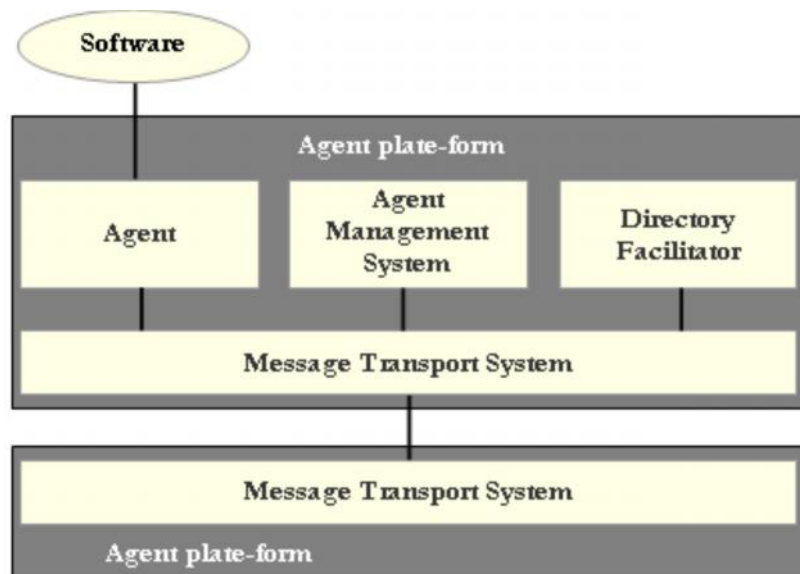


Figure 4 : Architecture logiciel de La plate-forme JADE

Dans la plate-forme JADE, deux méthodes sont fournies par la classe Agent afin d'obtenir l'identifiant de l'agent DF par défaut et celui de l'agent AMS : getDefaultDF() et respectivement getAMS(). Ces deux agents permettent de maintenir une liste des services et des adresses de tous les autres agents de la plate-forme. Le service DF propose quatre méthodes afin de pouvoir :

- Enregistrer un agent dans les pages jaunes (register).
- Supprimer un agent des pages jaunes (deregister).
- Modifier le nom d'un service fourni par un agent (modify).
- Rechercher un service (search).

Le service AMS s'utilise généralement de manière transparente (chaque agent créé est automatiquement enregistré auprès de l'AMS et se voit attribué une adresse unique). Ces deux services fournissent donc les annuaires qui permettent à n'importe quel agent de trouver un service ou un autre agent de la plate-forme.

4. Langage de communication de la plate-forme JADE

Le langage de Communication de la plate-forme JADE est FIPA-ACL (Agent Communication language). La classe ACLMessage représente les messages qui peuvent être échangés par les agents. La communication de messages se fait en mode asynchrone.

Lorsqu'un agent souhaite envoyer un message, il doit créer un nouvel objet ACLMessage, compléter ces champs avec des valeurs appropriées et enfin appeler la méthode send().

Lorsqu'un agent souhaite recevoir un message, il doit employer la méthode receive() ou la méthode blockingReceive().

Un message ACL dispose obligatoirement des champs suivants :

Performative : type de l'acte de communication

Sender : expéditeur du message

Receiver : destinataire du message

reply-to : participant de la communication

content : contenu du message

language : description du contenu

encoding : description du contenu

ontology : description du contenu

protocol : contrôle de la communication

conversation-id : contrôle de la communication

reply-with : contrôle de la communication

in-reply-to : contrôle de la communication

reply-by : contrôle de la communication

Tous les attributs de la classe ACLMessage peuvent être obtenus et modifiés par les méthodes set/get(). Le contenu des messages peut être aussi bien du texte que des objets car la sérialisation Java est supportée.

V. Exemple d'application

Dans notre application, on a une architecture *client* (*Atelier + Agent Production (APr)*) / *Serveur* (*Web + 4 agents*). Dans le coté *client*, l'utilisateur va créer un nouveau projet qui va contenir un ensemble de ressources. À un certain moment une ou plusieurs ressources tombent en panne. Pour la résolution de cette panne, on a proposé un outil d'aide à la décision à base de Web qui va aider l'utilisateur à choisir la tâche à effectuer par exemple :

réparer la ressource, commander une nouvelle ressource,.. À ce moment-là, l'*Agent Production (APr)* va récupérer la liste des ressources en panne et l'envoyer au serveur. La réception de cette liste va être par l'**Agent Analyseur (AA)**.

1. Création d'un nouveau projet

Du coté client, l'utilisateur va créer un nouveau projet nommé « prod » (*Figure 5*) et l'enregistrer (*Figure 6*).



Figure 5 : Création d'un nouveau projet

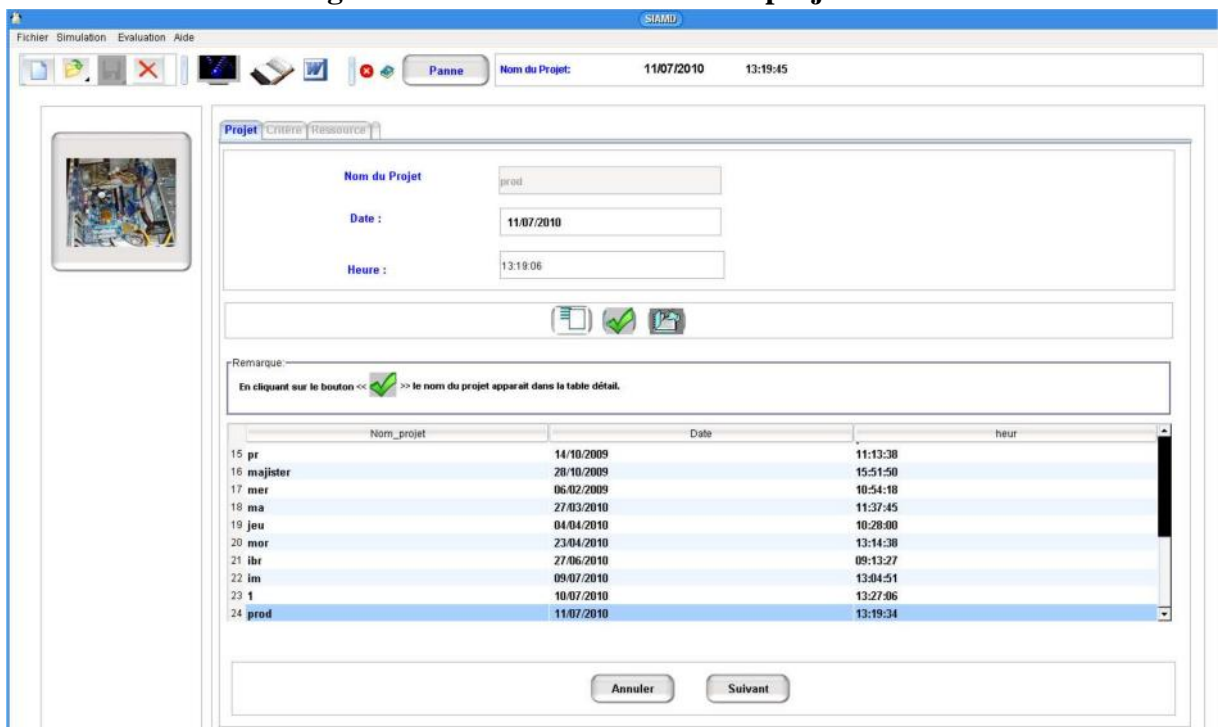


Figure 6 : L'enregistrement d'un nouveau projet

2. Affectation des ressources au projet

Le projet « prod » contient huit (8) ressources qui sont de différents types (Imprimantes, Ordinateurs_Portables). Chaque ressource est identifiée par : *Nom de ressource*, *Type de ressource*, *Temps_Début*, *Temps_Fin*, *Fournisseur* (*Figure 7*) (*Figure 9*), après la sélection du *Type de ressource* (Imprimante ou Ordinateurs_Portable), on clique sur le bouton pour sélectionner d'autres attributs : *Marque*, *Catégorie*, *Référence*. (*Figure 8*) (*Figure 10*).

Soit la *Ressource 1* de type *Imprimante* (*Figure 7*).

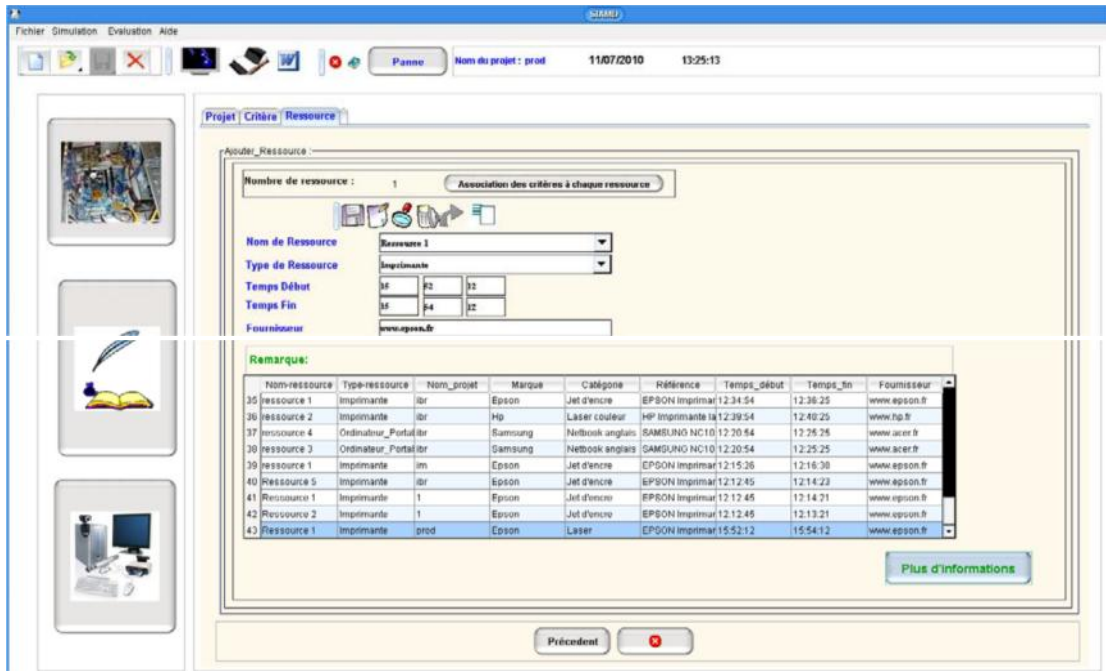


Figure 7 : Affectation des ressources au projet (Imprimante)



En cliquant sur le bouton la fenêtre suivante apparaîtra :



Figure 8 : Détails sur les ressources de type « imprimante »

Soit la Ressource 2 de type *Ordinateur_Portable* (Figure 9).

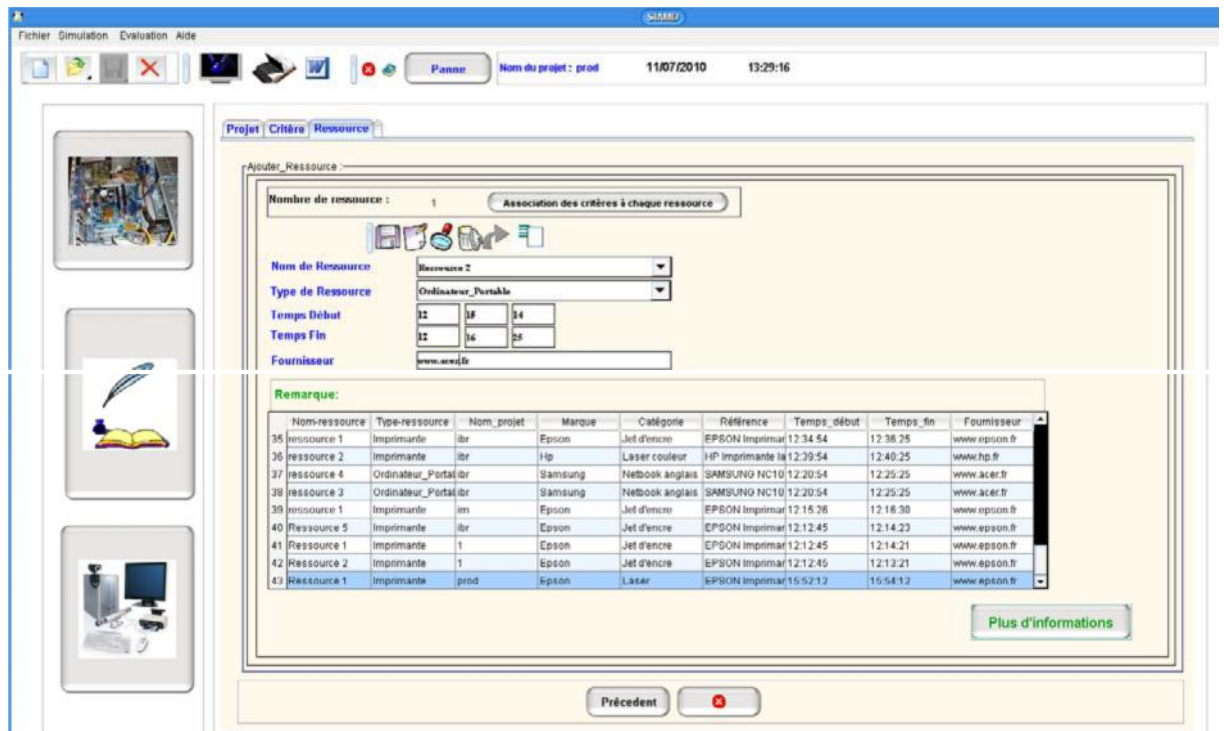


Figure 9 : Affectation des ressources au projet (Ordinateurs_Portables)



En cliquant sur le bouton la fenêtre suivante apparaît:

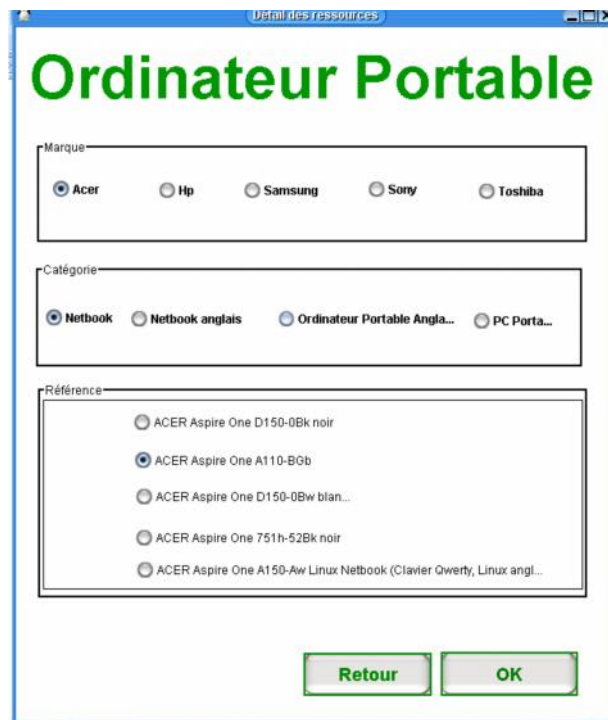


Figure 10 : Détails sur les ressources de type « Ordinateur Portable »

3. Traitement de l'évènement panne

a. Exécution des ressources

La Figure 11 montre l'exécution des ressources.

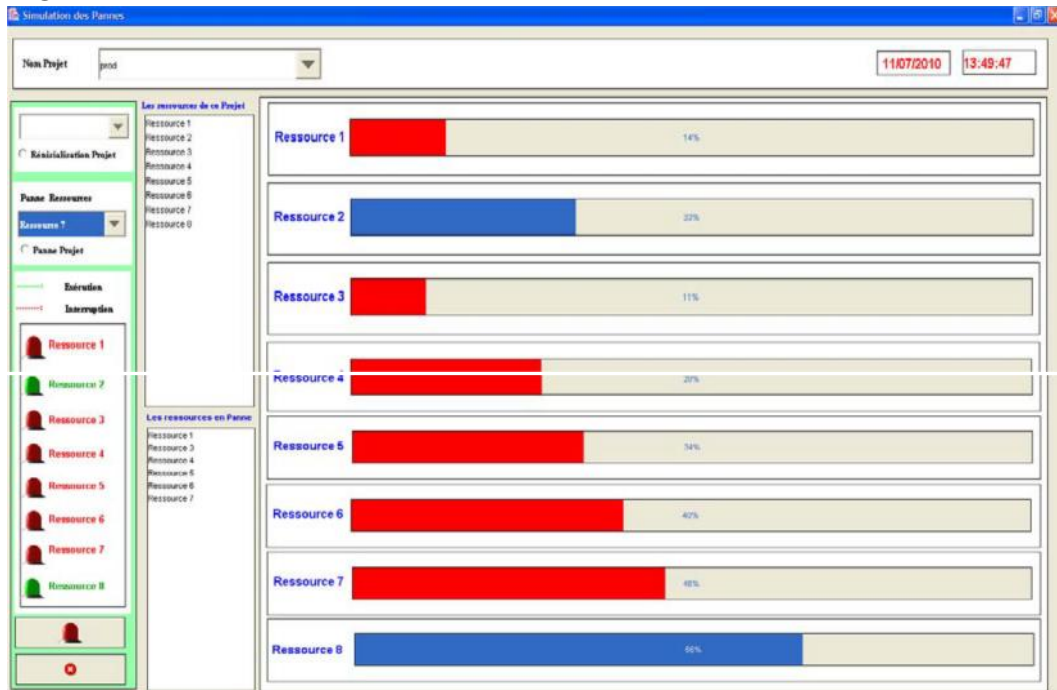



Figure 11 : Provocation de panne

Pour obtenir les détails sur la panne, on clique sur :  on aura la fenêtre suivante :

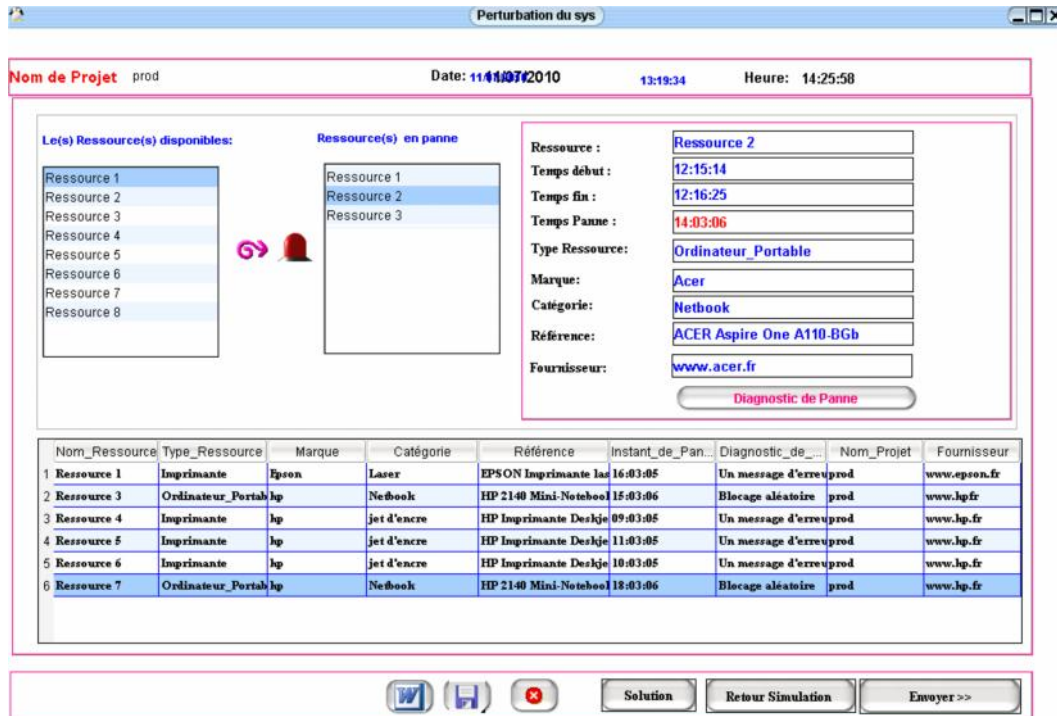


Figure 12 : Détail sur la panne

4. Résolution de panne

En cliquant sur le bouton « envoyer », *l'Agent Production (APr)* va récupérer la liste des ressources en panne. (*Figure 13*).

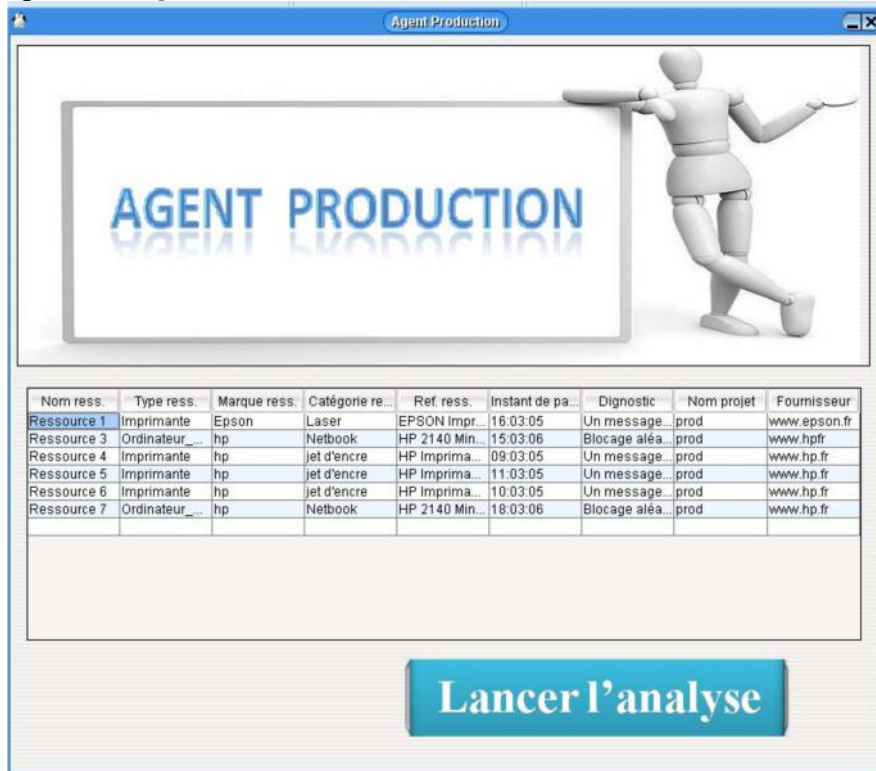


Figure 13 : l'interface de l'agent production

En cliquant sur le bouton, **Lancer l'analyse** on va activer *l'Agent Analyseur (AA)* qui va analyser et filtrer cette table. (*Voir la figure 14*).

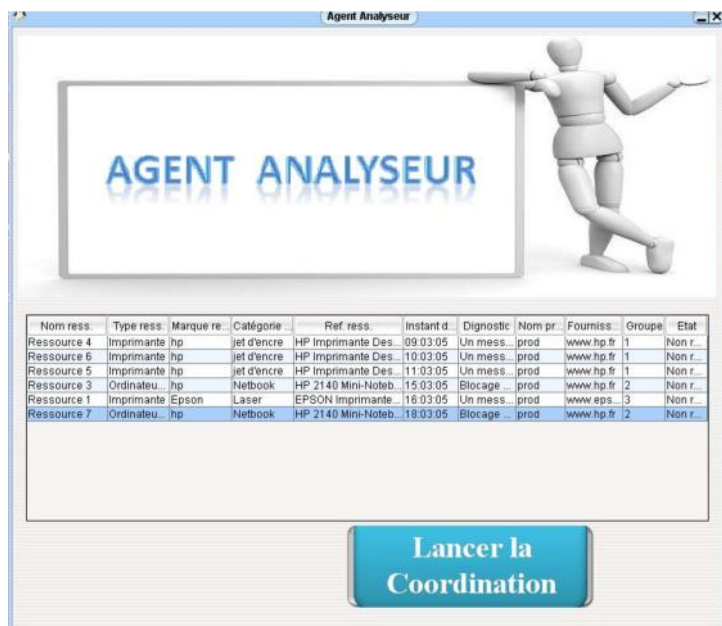


Figure 14 : L'interface de l'Agent Analyseur

Lancer la
Coordination

En cliquant sur le bouton on va activer *l'Agent Coordinateur (AC)* dont l'interface est montrée dans la *figure 15*.



Figure 15 : L'interface de l'Agent Coordinateur

On commence par la recherche de la première ressource, donc on clique sur le bouton « Recherche ressources sur Web » qui va activer *l'Agent Ressource*. (Figure 16).



Figure 16 : Interface de l'agent ressources

En cliquant sur le bouton « Recherche », l'Agent Ressource (AR) va chercher sur le web la référence de la ressource en panne en parcourant plusieurs pages web. (Figure 17).



Figure 17 : exemple sur les pages web
Comme résultat de la recherche, on aura la fenêtre suivante (Figure 18).

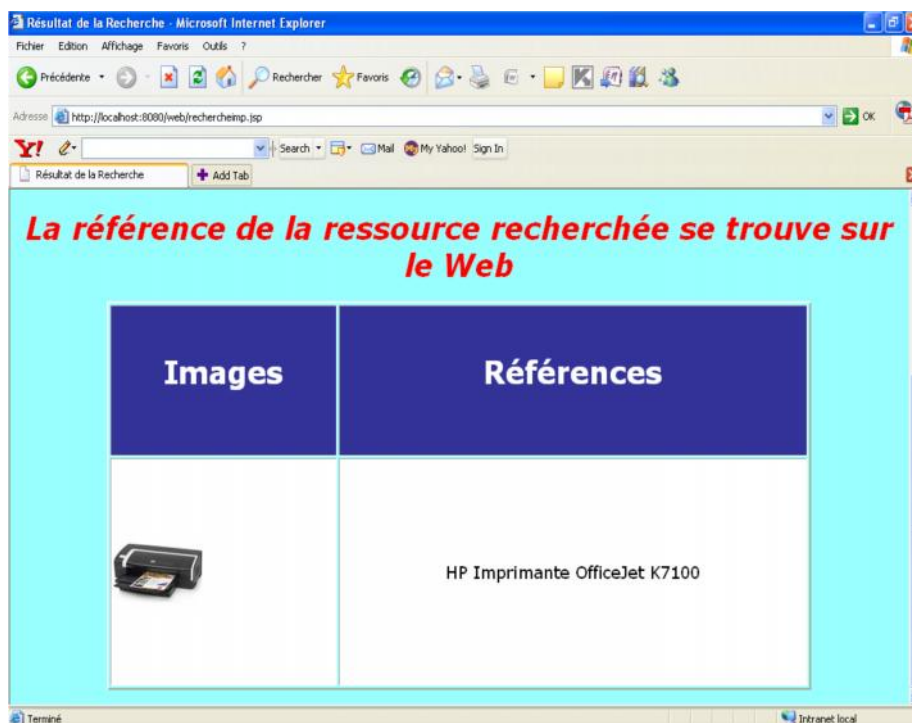


Figure 18 : Résultat de la recherche avec succès

Dans le cas contraire, c'est-à-dire en cas d'échec, on aura la fenêtre suivante. (*Figure 19*).

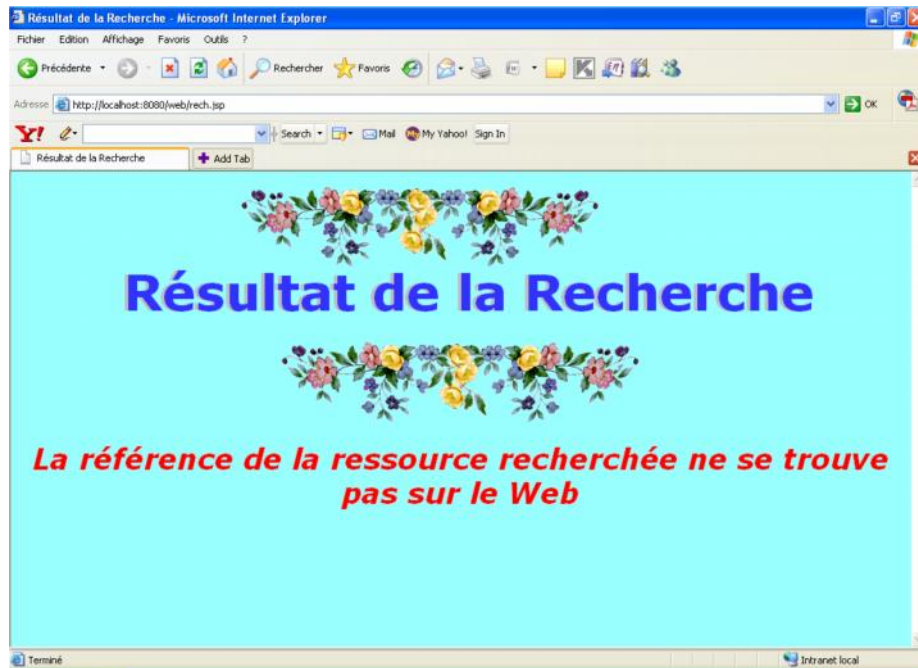


Figure 19 : Résultat de la recherche avec échec

En cliquant sur le bouton « Recherche solution sur le Web », on va activer *l'Agent Proposition (AP)* *Figure 20*. Ce dernier va lancer une recherche de la solution, comme cela est montré à la *figure 21*. La solution sera envoyée au client.



Figure 20 : Interface e l'agent proposition

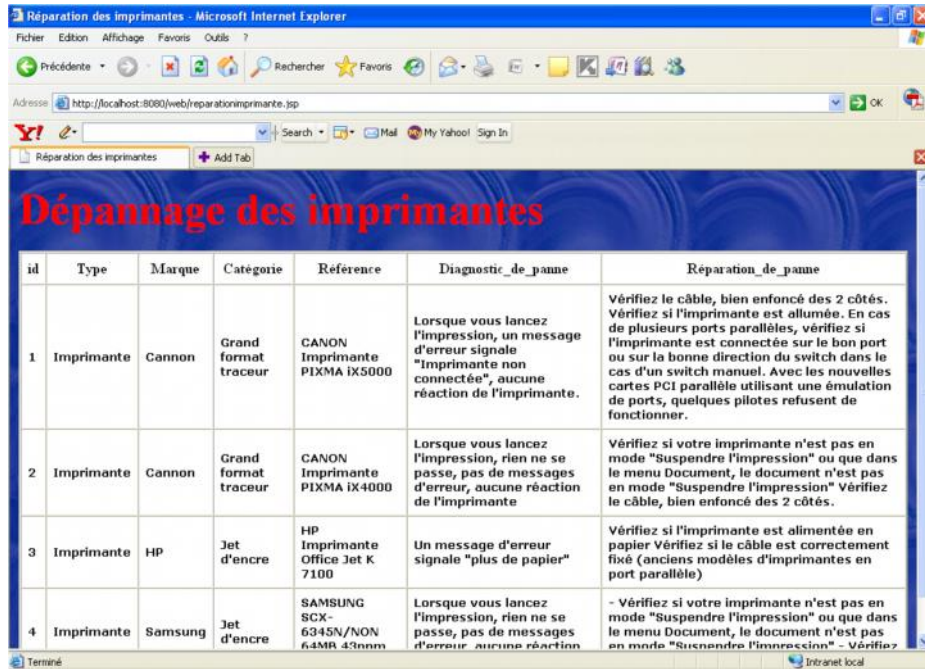


Figure 21 : exemple de solution

La communication entre les agents est présentée par la figure suivante : (Figure 22)

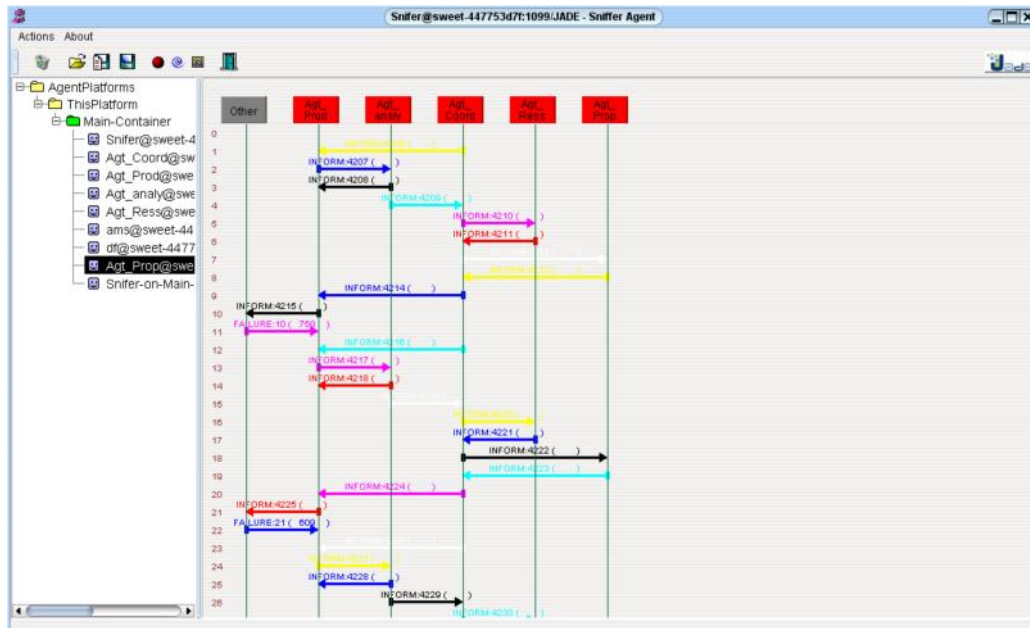


Figure 22 : interactions entre les agents (plate-forme JADE)

VI. Conclusion

Dans ce chapitre, nous avons défini l'architecture de développement ainsi que l'environnement d'interaction. Plusieurs écrans d'affichage sont proposés par la suite, pour donner un aperçu sur la prise de décision en utilisant la technologie Web et les différents agents composant notre système à savoir l'Agent de Production (APr), l'Agent Analyseur (AA), l'Agent Coordinateur (AC), l'Agent Ressource (AR) et l'Agent Proposition (AP).

CONCLUSION GENERALE

Conclusion générale et perspectives :

Les travaux de recherche réalisés dans le cadre de ce mémoire ont permis de présenter une modélisation d'un processus de coordination au sein d'un SMA dans une configuration de WEB-DSS appliqué à une gestion de production dynamique en temps réel.

L'intérêt majeur d'un tel support est d'avoir un outil d'aide à la décision par rapport à des événements se produisant en temps réel. Cet outil a été réalisé par la conception d'un SMA dans lequel plusieurs agents cognitifs sont en interaction pour réaliser et atteindre les buts escomptés. L'utilisation d'un système multi-agents tel qu'il est présenté au chapitre 4, permet d'apporter beaucoup pour le SIAD à base de web dont la flexibilité, la rapidité d'exécution et surtout la collecte d'un maximum d'informations sur le problème en question pour apporter la meilleure solution, donc la meilleure aide aux décideurs. En utilisant le web pour rassembler des quantités très importantes d'informations pour proposer de l'aide, sa gestion constitue un problème majeur. En effet, comment stocker et comment exploiter toutes ces informations ont fait l'objet de plusieurs recherches ; mais aujourd'hui, grâce aux SIAD à base de web, nous dirons que la solution est toute trouvée.

Comme nous l'avons indiqué précédemment, les interactions au sein des SMA font à la fois la force et la faiblesse d'un système multi-agents. Dans ce but, nous avons présenté le contexte général des interactions possibles au sein d'un SMA tout en mettant l'accent sur les difficultés, non seulement à les mettre en œuvre, mais aussi sur le fait que ces dernières peuvent amener tout le système vers une incohérence globale et vers un comportement chaotique ; raison pour laquelle nous nous sommes fixés un objectif de taille qui est celui de permettre aux agents du SMA d'interagir entre eux, tout en gardant une certaine cohérence et surtout leur permettre de réaliser les objectifs que nous leur avons fixés au préalable.

Nous avons conçu un modèle de coordination englobant plusieurs stratégies mises en œuvre par un *agent coordinateur (AC)*, permettant de garder la cohérence du système et d'intégrer tous les temps relatifs à la résolution des problèmes dans une unité de production, afin d'avoir des statistiques qui permettent de faire un comparatif ou étude permettant d'affirmer ou de confirmer que face à tel ou tel problème c'est cette stratégie qui permet d'obtenir les meilleurs résultats.

Comme nous nous sommes basés sur un SIAD à base de web pour apporter de l'aide aux décideurs, nous avons conçu une mémoire collective au sein du SIAD pour permettre, entre autres, aux décideurs de coopérer, voire négocier une solution à un problème donné, tout en laissant une trace de ces choix et compromis, durant tout le processus de prise de décision. Cette mémoire permet d'apporter un plus dans ce domaine sur le plan scientifique mais elle apporte également de nombreux avantages au sein du SIAD à base de web comme nous l'avons déjà cité au chapitre 4.

Pour résumer l'approche que nous avons conçue, la mise en œuvre du processus de coordination nous a permis de recenser toutes les stratégies possibles de coordination, en particulier celles qui ont été suivies par l'agent coordinateur durant une session de travail, c'est-à-dire une prise de décision globale. Pour satisfaire cet objectif, nous avons enrichi le modèle proposé par une *mémoire collective (MC)*. Cette mémoire permet de mentionner les adaptations apportées aux solutions tirées de la base des cas de l'agent coordinateur et elle permet à ce dernier de se procurer des solutions aux pannes, si elles existent, au niveau de la MC.

De plus, nous pouvons évaluer, grâce à l'outil développé, chaque stratégie de l'agent de coordination par la mesure de certaines valeurs de temps. Six scénarios ont été testés sur l'application et montrent la faisabilité de l'approche.

Finalement, nous dirons que le travail que nous avons mené n'est pas une fin en soi ni un aboutissement, mais au contraire, nous espérons que ces recherches permettent d'apporter une modeste contribution dans ce domaine, constituant peut-être une ouverture vers d'autres horizons qui n'ont pas été pris en charge dans le cadre de ce mémoire.

Sur le plan personnel, ce travail nous a permis d'apprendre énormément sur plusieurs domaines d'activités et de recherches, puisque comme nous l'avons cité en introduction, le paramètre le plus important lors de la conception des SIAD à base de web est le fait d'intégrer presque tout ce que nous voulons comme technologies à base de web, même si elles proviennent de régions disciplinaires tout à fait différentes. Cette intégration de technologies est possible du moment qu'elle ramène un plus, mais surtout, il ne faut pas commettre l'erreur de demander toujours plus au point d'affaiblir le système en question. En fait, c'est comme une recette de cuisine où nous pouvons mélanger un tas d'ingrédients pourvu que l'on obtienne le meilleur ragoût possible.

Concernant les perspectives, nous pouvons dire qu'il y a beaucoup à faire dans ce domaine tant sur le plan recherches futures que sur le plan expérimentations.

Sur le plan expérimentations, nous essayerons de changer de type de coordination tout en gardant le même système. C'est-à-dire, passer d'une coordination centralisée à une coordination décentralisée, en implémentant tous les sous-types appartenant à ce type de coordination. En effet, cela pourrait être très enrichissant dans la mesure où nous pouvons faire une étude comparative entre les deux types de coordination pour chaque type de problème.

Sur le plan recherches futures, il serait intéressant de réaliser des travaux permettant d'aboutir à la création d'une ontologie de coordination pour chaque domaine d'application. Cette ontologie sera en mesure de garantir un fonctionnement cohérent global pour tout système lui faisant appel, tout en soulageant le concepteur de ce travail fastidieux.

Un autre axe de recherche pourrait être celui de la réalisation d'un outil comme le nôtre en utilisant uniquement des agents mobile.

Globalement, la plupart des approches de coordination se basent sur la distribution du contrôle et des données. Cette distribution est inhérente à l'architecture intrinsèque d'un SMA. Le contrôle distribué signifie que chaque agent possède un minimum d'autonomie pour générer de nouvelles actions et décider quels buts il va poursuivre. En revanche, cette distribution induit une dispersion des connaissances et une vue partielle rendant ainsi la cohérence globale difficile à atteindre, d'où la nécessité de la coordination qui reste un mécanisme-clé dans le domaine des SMA. Plusieurs solutions ont été explorées et plusieurs ouvertures restent à l'état d'investigation. Cependant, peu de systèmes intègrent le concept d'apprentissage dans leurs mécanismes de coordination, pour renforcer l'adaptabilité des agents. Beaucoup d'efforts sont également à fournir pour développer des mécanismes de coordination dynamiques et réellement distribués pour les agents.

RÉFÉRENCES
BIBLIOGRAPHIE

[AFGI, 91]: "Dictionnaire des termes de gestion industrielle" Editions AFGI, 1991 R. Biteau, A. Garreau, M. Gavaud.

[AFNOR, 90] : "Analyse de la valeur" Norme française NF-X 50-150, 1990.

[AFNOR, 91] : "Concepts fondamentaux de la Gestion de Production - Vocabulaire" Norme française NF X50-310, 1991.

[Alty et al., 94]: Alty, J. A., Griffiths, D., Jennings, N. R., Mamdani, E. H., Struthers, A. and Wiegand, M. E. (1994) ADEPT-Advanced Decision Environment for Process Tasks: Overview & Architecture. In: BCS Expert Systems 94 Conference (Applications Track), Cambridge, UK. pp. 359-371.

[Arnott et al., 07] Arnott D., Jirachiefpattana W., et O'Donnell P. Executive information B. H. Roth, M. Hewett, R. Washington, R. Hewett, and A. Seiver, 1989.

[Apics, 2005] Apics Dictionary, The American Production and Inventory Control Society, Inc., 2005, Edition No. 11.

[Archimède, 1991] B. Archimède, "Conception d'une architecture réactive distribuée et hiérarchisée pour le pilotage des systèmes de production", Thèse de doctorat, université de BordeauxI, 1991.

[Artigues, 1997] Artigues, C. 1997. *Ordonnancement en temps réel d'ateliers avec temps de préparation des ressources*. Thèse de Doctorat, Université Paul Sabatier de Toulouse, Laboratoire d'analyse et d'architecture des systèmes du CNRS.

[Azvine et al., 2000] : B. Azvine, N. Azarmi, D.D. Nauck (Eds.), Intelligent systems and soft computing XVII. 359 pages.2000.Vol. 1804.

[Bellifemine et al., 1999] Bellifemine F., Poggi A., Rimassa G., "JADE -- A FIPA-compliant agent framework", CSELT internal technical report. Part of this report has been also published in Proceedings of PAAM'99, London, pp.97-108, April 1999.

[Bellifemine et al., 2000] Bellifemine F., Giovanni C., Tiziana T. Rimassa G., "Jade Programmer's Guide" Jade version 2.6 (<http://sharon.csel.it/projects/jade/>), 2000.

[Bellifemine et al., 2005] Fabio Bellifemine, Federico Bergenti, Giovanni Caire, et Agostino Poggi. JADE - A Java Agent Development Framework, chapter 5. Bordini et al., 2005.

[Bellifemine et al., 2008] Bellifemine F., Caire G., Poggi A., JADE G., "A software framework for developing multiagent applications", Lessons learned *In Information and Software Technology* 50, 2008, p. 10-21.

[Bellman et al., 1982] Bellman, R., Esogbue, A.O., Nabeshima, I. 1982. *Mathematical aspects of scheduling and applications*. Pergamon press.

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

[Bessedik, 2010] Imene. Bessedik : Réalisation d'un WEB-DSS à base d'agents pour la gestion de production dynamique. 2010. Thèse de Magister.

[Bharati et Chaudhury, 2004] Bharati. P., and Chaudhury, A.,:« *An empirical investigation of decision-making satisfaction in Web- based decision supportsystems* », *Decision Support Systems* 37 (2), pp. 187-197, 2004.

[Blondel, 1997] François Blondel. Gestion de la production: comprendre les logiques de gestion industrielle pour agir. Fonctions de l'entreprise. Série Performance industrielle Fonctions de l'entreprise. Production-qualité. ISBN : 2100032836, 9782100032839. 353 pages.

[Bohm, 1976] Bohm , B. W. (1976). Software Engineering. IEEE Transactions on computing, p. 1226-1241.

[Bond et Gasser, 88]A.H. Bond and L. Gasser, "An Analysis of Problems and Research in Distributed Artificial Intelligence," in [Bond and Gasser 88b], 1988.

[BON, 88]Bond A.H. Gasser.L (dir) Readings in distributed Artificial intelligence, Morgan Kaufmann publishers. 1988.

[Boukachour *et al.*, 2000] Boukachour, H., C. Duvallet et A. Cardon (2000). Multiagent system to prevent technological risk. In Proceedings of ACIDCA'2000.

[BRA, 87] Bratman M.E., Intentions Plans and practical reasons. Harvard university Press. Cambridge, M.A. 1987.

[Breuil, 1984] Alain Dominique Breuil, "Outil de conception et de décision dans les organisations de gestion de production", Doctorat d'Etat es Sciences, Université de Bordeaux I, soutenue le 13 novembre 1984.

[Burkhart, 1994] Burkhart, R. « *The SWARM multi-agent simulation system* ».In Object Oriented Programming Systems, Languages and Applications (OOPSLA), 1994.

[Caldéroni, 2002] Calderoni Stéphane (2002).Ethologie Artificielle et Contrôle Auto-Adaptatif dans les Systèmes d'Agents Réactifs : de la Modélisation à la Simulation. Thèse à l'Université de La Réunion, 175 pages.

[Carlier et Chrétienne, 1988] Carlier, J. et Chrétienne, P. « *Problèmes d'ordonnancement* ». Masson. 1988.

[Carlier et al., 1993] Carlier, J., Chrétienne, P., Erschler, J., Hanen, C., Lopez, P., Munier, A., Pinson, E.,Portmann, M.C., Prins, C., Proust, C. et Villon, P. 1993. GOTHa, (Groupe de Recherche en Ordonnancement Théorique et Appliqué). Les problèmes d'ordonnancement. *RAIRO RO* , 27 (1) : 77-150.

[CAS, 98] Castelfranchi C., « Modelling social action for AI agents » Artificial intelligence. 103. P.1-2. 157-182. 1998

[Chabbat, 1997] Chabbat B., Modélisation Multiparadigme de textes réglementaires, Thèse de doctorat, LISI, Décembre 1997, 392p.

[Chaib-draa, 94] Distributed Artificial Intelligence: An overview. In A. Ken, J. G. Williams, C. M. Hall, and R. Kent, editors, *Encyclopedia Of Computer Science And Technology*, volume 31, pages 215-243. Marcel Dekker, Inc, 1994.

[Chaib-draa, 1996] Chaib-draa, B et Moulin, B, «*An overview of Distributed Artificial Intelligence* », in *Foundations of Distributed Artificial Intelligence*, 1, G.M.P. O'Hare et N.R. Jennings eds., J. Wiley and sons, pp. 3.55, 1996.

[Côté et Troudi, 1998] M. Côté and N. Troudi. Une architecture multiagent pour la recherche. *L'expertise Informatique*, 3(3), 1998.

[COR, 79] Corkill D.D « Hierarchical planning in distributed environment » dans *proceeding of IJCAT79*.p. 168-175.1979

[Courtois, 1995] Courtois. A., Pillet, M., et Martin. C., :« *Gestion de Production* ». Editions d'organisation, 2ème édition, 1995.

[Courbon, 1986] Courbon, J.C., "Les SIAD : outils, concepts et mode d'action", *AFCET/INTERFACE 9* (1986) 30-36.

[Courbon et Stabell, 1986] Courbon, J.-C. et C. B. Stabell (1986). Artificial intelligence and the design of decision support systems, tutorial of the conference on economics and artificial intelligence, Aix-en-Provence.

[Crossland et Wynne, 1995] Crossland , M. D. et B. E. Wynne (1995). "Spatial Decision Support Systems: an overview of technology and test of efficacy." *Decision Support Systems* 14(3): p. 219-235.

[Daniels, 1999] Daniels, M. « *Integrating simulation technologies with swarm* ». In *Agent Simulation: Application, Models, and Tools*, University of Chicago. 1999.

[Daniels, 2000] Daniels, M. « *An open framework for agent-based modeling* ». In *Applications of Multi-Agent Systems in Defence Analysis*, Los Alamos National Labs. 2000.

[Dargam et al. 06] Dargam F.C.C., et Zaraté P. Decision Support Systems: Methodologies and Applications, *Journal of Decision Systems*, vol. 15 (2-3), pp. 157-158, 2006.

[DAV, 83] R. Davis and R. Smith. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, 20(1) :63-109, January 1983.

[Davis, 1989] Davis, F.D.: « *Perceived usefulness, perceived ease of use, and user acceptance of information technology* », *MIS Quarterly* 13 (3) 319-340, 1989.

[Davis *et al*, 1986] Davis, G. B., M. H. Olson, J. Ajenstat et J.-L. Peaucelle (1986). Systèmes d'information pour le management. Volume 1 et 2, Montréal, édition G. Vermette.

[DEC, 92] DECKER K.S.. LESSER V.R.. « Generalizing the Partial Global Planning Algorithm ». Dans *international journal on intelligent Cooperative information Systems*. p. 319-346. 1992.

[DEC, 93] DECKER K..LESSER V., « A one-shot dynamic coordination algorithm for distributed sensor networks ».dans *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93)*, p. 210-216. 1993.

[DEC, 95] DECKER K..LESSER V..«Designing a **family** of coordination mechanisms ».dans *Proceedings of the First international Conf on Multi-agents Systems (JCMAS-95)*. p. 73-80. 1995.

[Delen *et al.*, 2007] Delen D., Sharda R., Kumar P., “Movie forecast guru: a Web based DSS for Hollywood managers”, *Decision Support Systems*, 43, 2007, p. 1151-1170.

[DeLone *et McLean*, 1992]. DeLone, W.H., et McLean, E.R.,:« *Information systems success : the quest for the dependent variable* », *Information Systems Research* 3(1), 60-95, 1992.

[Demazeau, 1995] Yves Demazeau. From interactions to collective behaviour in agent-based systems. In Pre-proceedings of the invited lectures of the 1st European Conference on Cognitive Science, St. Malo, France, March 1995

[Dong *et al.*, 2004] Dong J., Du H.S., Wang S., Chen K., Deng X., “A framework of Web-based decision support systems for portfolio selection with OLAP and PVM”, *Decision Support Systems*, No. 3,2004, p. 367-376.

[Doumeingts, 1984] Guy Doumeingts, “Méthode GRAI : Méthode de conception en productique”, Doctorat d'Etat es Sciences, Université de Bordeaux I, soutenue le 13 novembre 1984.

[Druzdzal *et Flynn*, 1999] M. J. Druzdzal, and R. R. Flynn, *Decision Support Systems*, Encyclopedia of Library and Information Science, A. Kent, Marcel Dekker, Inc., 1999.

[DUR, 87] Durfee E.D..Lesser V.R..Corkill D..«Cohérent coopération among communicating Problem solvers ».IEEE transactions on computer.C-36.vol. 11.n° 1275-1291. 1987.

[DUR, 88] E. H. Durfee. *Coordination of distributed problem solvers*. The Kluwer international series in engineering and computer science; Artificial intelligence. Kluwer, Boston, 1988.

[DUR, 91], DURFEE H.D.. Montgomery A.. «Coordination as Distributed Search in a hierarchical behavior Space ».IEEE Transactions on Systems. Man. and Cybernetics, Special Issue on Distributed Artificial Intelligence. SMC-2I(6). p. 1363-1378. 1991.

- [**DUR, 99**] Durfee E.D.« Dislributed Problem Solving ».dans G. Weiss (dir.). *Multi-Agent Systems: A modem approach to distributed art (fie ial intelligence*. MI 1 Press. 1999.
- [**Durfee et al., 1989**] E. H. Durfee and V. Lesser. Negotiating task decomposition and allocation using partial global planning. In L. Gasser and M. Huhns, editors, *Distributed Artificial Intelligence Volume II*, pages 229-244. Pitman Publishing : London and Morgan Kaufmann : San Mateo, CA, 1989.
- [**Eierman et Niederman, 1995**] Eierman, M. A. et F. Niederman (1995). "DSS theory: a model of constructs and relationships." *Decision Support Systems* 14(1): p. 1-26.
- [**El Fallah, 2001**] El-Fallah Seghrouni A. Modèles de coordination d'agents cognitifs. 2001.
- [**ELF, 96b**]El Fallah-Seghrouchni A.. Haddad S..« A Recursive Model for Distributed Planning ».dans*Proceedings of ICMAS'96 (international Conférence of Multi-agents Systems)*. AAAI Press. Kyoto. Japon. 1996.
- [**EPH, 95a**] Ephrati F.. Rosenschein J.S..« A Framework for the Interleaving of Execution and Planning for Dynamic lasks by Multiple Agents ».dans*Proceedings of ATAL'95*. 1995.
- [**EPH, 95b**] Ephrati F.. Pollack F.M. Rosexscieix S..« A tractable heuristic that maximizes global utility through local plan combination ».dans*Proceedings ofthe First International Conf. on Multt-agents Systems fJCMAS-95*).p. 94-101. 1995.
- [**Erceau & Ferber, 91**]. ERCEAU J. et FERBER J. (1991) L'intelligence artificielle distribuée. *La Recherche*, Juin.
- [**Erschler et al., 1992**] Erschler, J., Fontan, G., Roubellat, F. 1992. *Encyclopédie du management – Ordonnancementen ateliers spécialisés*. Helfer et Orsoni,Vuibert, Paris, 2 : 208-229.
- [**Farenc et al.,1994**] Farenc, C, P. Palanque et J. Vanderdonckt (1994). L'évaluation ergonomique de l'utilisabilité d'une application interactive, est-elle utilisable ? IHM'94. 6iemes journées sur l'ingénierie des interfaces homme-machine, Lille.
- [**Fer, 92**]. I. A. Ferguson. *TouringMachines : An Architecture for Dynamic, Rational, Mobile Agents*.PhD thesis, Clare Hall, University of Cambridge, UK, November 1992.(Also available as Technical Report No. 273, University of Cambridge Computer Laboratory).
- [**Fer, 95**] J. Ferber. *Les systèmes multi-agents, vers une intelligence collective*. InterEditions, 1995.
- [**Franklin et Graesser, 97**] Franklin, S., and A. C. Graesser. [1997] Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In *Intelligent Agents III*. Berlin: Springer Verlag.

[GEO, 83] Georgeff M.P « Communication and interaction in Multi-agent Planning » dans proceedings of AAAI'93.p. 125-129. 1983

[GEO, 90] Georgeff M.P “Readings in Planning” Morgan Kaufman Publishers. San Matteo.Californie. 1990.

[Ferber, 1995] J. Ferber. *Les systèmes multi-agents, vers une intelligence collective*. Inter Editions, 1995.

[FIPA, 2002] The FIPA ACL Message Structure Specifications. <http://www.fipa.org/specs/fipa00061/2002>.

[Finlay, 1994] Finlay, P. (1994). *Introducing Decision Support Systems*, Oxford: Blackwell.

[Friedman, 2005]Friedman T.L., *The World Is Flat: A Brief History of the Twenty-First Century*, Farrar, Straus and Giroux, 2005.

[Fox, 1981] Fox, M. (1981). Factory modeling, simulation and scheduling in the intelligent management system. In Proceedings of the 7th International JointConference on Artificial Intelligence.

[Gachet, 2001] Gachet, A. (2001). A Framework for Developing Distributed Cooperative Decision Support Systems - Inception Phase.Conference Proceedings, 2001 Informing Science Conference, June 19-22 Krakow, Poland.

[Gannon, 1979] Gannon, C. (1979). Error detection using **path testing and static analysis**. *Computer*.12 august.

[Garlatti, 1996] Garlatti, S. (1996). Tutorial: Multimédia et systèmes d'aide à la décision en situation complexe, in 43th meeting of the european working group "Multicriteria Aid for Decisions", Brest.

[GAS, 84] Gassler L., *The Social Dynamics of Routine Computer use in Complex Organizations*. Ph.D. thesis.University of California.Department of Computer Science. Irvine. 1984.

[GAS, 92] Gasser L., Briot J.P., *Object-Based Concurrent Programming and DAI*, in Avouris N.M., Gasser L. (Eds.) *Distributed Artificial Intelligence: Theory and Praxis*. Kluwer Academic Press, Dordrecht, pp.81. 1992.

[GEL, 92] Gelernter D..CARRIERO N..« Coordination Languages », *Communications of the ACM*, 35, 2, 1992.

[GEO, 83] Georgeff M.P..Communication and Interaction in Multi-agents Planning ».dans*Proceedings of AAAI'93*. p. 125-129. 1983.

[GEO, 88] Georgeff M.P. « A Theory of Action for Multi-agents Planning » dans A.H. Bond et L. Lesser (dir.). *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann Publishers. p. 205-209. 1988.

[GEO, 90] Georgeff M.P. *Readings in Planning*. Morgan Kaufmann Publishers. San Mateo. Californie. 1990.

[Ghasemzadeh et Archer, 2000] Ghasemzadeh, F. et N. P. Archer (2000). "Project portfolio selection through decision support." *Decision Support Systems and Electronic Commerce*.

[Giard, 1988] *Gestion de la production*, Edition Economica, 1988, 2^{ème} édition .V. Giard.

[Gorry et Scott-Morton, 1971] Gorry, G.A. and Scott Morton, M.S. "A Framework for Management Information Systems," *Sloan Management Review* (13:1), Fall 1971, pp.55-70.

[Grabot et Huguet, 1996] Bernard Grabot, Pierre Huguet "Reference models and objectoriented method for reuse in production activity control system design", *Computer in Industry*, 1996, No. 32, pp. 17-31.

[GRO, 96] Grosz B.J., KRAUS S.. « Collaborative plans for complex group action ». *Artificial Intelligence*. 86. p. 269-357. 1996.

[Gue, 01] Z. Guessoum and M.Ocello "principes et architectures des systemes multi-agents. Chapter environnements de developpement. Hermes. 2001.

[Guessoum, 1996] Guessoum, Z. « *Un Environnement Opérationnel de Conception et de Réalisation de Systèmes Multi-Agents* ». PhD thesis, Laforia, Paris6. 1996.

[Guessoum, 2003] April 2003. 9. Z. Guessoum, N. Faci, and J.-P. Briot. Adaptive replication of large-scale multi- agent systems - towards a fault-tolerant multi-agent platform.

[Guntzer et al., 2007] Guntzer U., Muller R., Muller S., Schimkat R.D., "Retrieval for decision support resources by structured models", *Decision Support Systems* 43, doi:10.1016/j.dss.2005.07.004, 2007, p. 1117-1132.

[Hackathorn et Keen, 81] Hackathorn R.D. et Keen P.G.W. Organizational Strategies for Personal Computing in Decision Support Systems. *MIS Quarterly*, vol. 5 (3), pp.21-28, 1981.

[Hansen et al., 1995] Hansen, J. V., R. D. Meservy et al. (1995). "Case-based reasoning application techniques for decision support." *Intelligent systems in accounting, finance and management*.

[Hättenschwile, 1993] Hättenschwiler, P. (1993). *Computer Assisted Top Down Modeling, Modeling Tools for Decision Support*, University of Fribourg: p. 101-131.

[**Hättenschwiler et al., 1998**] Hättenschwiler, P., M. Moresino et A. Schroff (1998). Rapid Prototyping of Decision Support System, conference proceedings of ICSC Symposium, Tenerife.

[**Hättenschwiler, 1999**] Hättenschwiler, P. (1999). Neue Konzepte der Entscheidungsuntersriitzung, Working Paper 99-4, Institute of Informatics, University of Fribourg.

[**HEW, 91**]Hewit.C “ Open information systems semantics for distributed artificial intelligence” *Artificial intelligence*.vol47. pages 79-106. 1991.

[**Holtzman, 1989**] Holtzman S. Intelligent Decision Systems. Addison Wesley, 1989.

[<http://www.objectspace.com/products/voyager/>]

[**Huaiqing et al, 2002**]. Huaiqing, W., Stephen L., Lejian L., “Modelling constraint-based negotiating agents”, *Decision Support Systems*, Vol. 33, No. 2, 2002, p. 201-217.

[**Huber, 1982**].Huber, G. P. (1982). Group Decision Support Systems as Aids in the Use of Structured Group Management Techniques. DSS-82, Conference Proceedings.

[**HUB, 96**] Huber M.J., Durfee.E.D “an initial assessment of plan recognition based coordination for multi agents teams” proceedings of the second international conf on multi agent systems (ICMAS-96).p.126-133.1996

[**Iglesias et al., 97**] C. A. Iglesias, M. Garijo, J. Centeno-Gonzalez, and J. R. Velasco. Analysis and designe of multiagent systems using MAS-common KADS. In Agent theories, Architectures, and Languages, Lecture Notes in Artificial Intelligence, pages 313- 327, **1997**.

[**ISH, 92**], Ishida. T, Gasser L., Yokoo.M “Organization self-design of distributed production systems" *IEEE trans on Knowl and data sys DKE4*. VOL2. Pages 123-134.1992

[**Iyer et al, 2005**] Iyer B., Shankaranarayanan G., Lenard M.L., “Model management decision environment: a Web service prototype for spreadsheet models”, *Decision Support Systems*, Vol. 40,No. 2, 2005, p. 283-304.

[**Jennings, 95**] N. R. Jennings. Controlling cooperative problem solving in industrial multi-agent systemsusing joint intentions.*Artificial Intelligence*, 74(2), 1995.

[**JEN, 98a**] N. R. Jennings, M. Wooldridge, and K. Sycara. A roadmap of agent research and development. *Int Journal of Autonomous Agents and Multi-Agent Systems*, 1(1) :7- 38,1998.

[**Jennings et al., 1998a**] N.R. Jennings and M.J. (eds) Wooldridge. *Agent Technology : Foundations, applications,and Markets*. Springer-Verlag : Heidelberg, Germany, 1998.

[Jennings et al., 1998b] N. R. Jennings, M. Wooldridge, and K. Sycara. A roadmap of agent research and development. *Int Journal of Autonomous Agents and Multi-Agent Systems*, 1(1) :7- 38, 1998.

[KAB, 95] Kabanza F., "Synchronizing Multiagent Plans using temporal logic specifications" dans *Proceedings of first international conference on Multi agent Systems. ICMAS-95. P 217-224.1995.*

[Keen, 1981] Keen, P. G. W., Ed. (1981). Decision Support Systems: A Research Perspective, in Decision Support Systems: Issues and Challenges, Pergamon Press.

[Keen, 87] Keen, P.G.W " decision support systems: the next decade". Septembre.1987. vol3.

[Kersten et le Noronba, 1999] : Kersten, G.E. and Noronba, S.J.: « WWW-based negotiation support design, implementation, and use », *Decision Support Systems* 25 (2), pp. 135- 154, March 1999.

[KET, 94] Ketchpel S..« Forming coalition in the face of uncertain rewards», dans *Proc, of the 12" National Conférence on Artificial Intelligence*, vol. 1. p. 414-419. Menlo Park. CA. A A AI Press. 1994.

[KIN, 92] KINNY D. L. jungberg M. RAOA.. Sonenberg b.. TIDHAR G..WERNER \L. *Planned Team Activity*.Preproceedings of the Fourth European Workshop on Modeling Autonomous Agents in a MultiAgent World, juillet 1992.

[Klein et Tixier, 1971] Klein, M. et V. Tixier (1971). SCARABEE: a data and model bank for financial engineering and research. IFTP congress, North Holland.

[Klein, 1988] Klein, D. A. (1988).Integrating Artificial Intelligence and Decision Theory to Forecast New Markets, IBM Research Division, Yorktown Heights.

[Kohli et al., 2001] Kohli R., Piontek F., Ellington T., VanOsdol T., Shepard M., "Managing customer relationships through E-business decision support applications: a case of hospital physician collaboration", *Decision Support Systems*, Vol. 32, 2001, p. 171-187.

[KOR, 81]Kornfeld; W.A..Hewitt C. « the scientific Community Metaphor ». *IEEE Transactions on System. Man and Cybernetics. vol. 11. n° 1. p. 24-33. janvier 1981.*

[Kreie et al., 2000] Kreie, J., T. P. Cronan , J. Pendley et J. S. Renwick (2000). "Applications development by end-users: can quality be improved?" *Decision Support Systems*.

[Kuljis et Paul, 2001] Kuljis J, Paul RJ. (2001). *Une évaluation des web-based simulation: où nous errons?.*Simulation Practice and Theory, 9. P 37-54.

- [**Kuo, 98**] Kuo F.Y. Managerial intuition and the development of executive support systems. *Decision Support Systems*, vol. 24 (2), pp. 89-103, 1998.
- [**Lamothe, 1996**] Lamothe, J. 1996. *Une approche pour l'ordonnancement dynamique d'un atelier de traitement de surface*. Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse.
- [**Lawler et al., 1989**] Lawler, E.L. , Lenstra, J.L. , Rinnooy Kan, A.H.G. , Schmoys, D.B. 1989. *Sequencing and scheduling : algorithms and complexity*. Rapport BS-R8909, Centre for Mathematics and Computer Science, Amsterdam.
- [**Le Moigne, 1974**] Jean-Louis Le Moigne, "Les systèmes de décision dans les organisations", Presses Universitaires de France, Paris, 1974.
- [**LES, 81**] Lessler V.R . Corkill D.D « functionally accurate cooperative distributed systems" IEEE trans on systems man and cybernetics SMC-11. P.81-96. 1981
- [**LES, 91**] Lessler V.R . "A retrospective view of FA/A distributed problem solving" dans IEEE trans on systems man and cybernetics. Special issue on distributed A.I.P.1347-1362. Vol 21.N°6. 1991.
- [**Lévine et al. 89**] Lévine P., et Pomerol J.C. *Systèmes interactifs d'aide à la décision et systèmes experts*, Hermès, 1989.
- [**Lévine et Pomerol, 1990**] LEVINE, Pierre; POMEROL, Jean-Charles Systèmes interactifs d'aide à la décision et systèmes experts.. Paris: Hermès, 1990. Note: Bibliogr. : p. 319-326.
- [**Lévy, 1996**] Lévy, M.L. 1996. *Méthodes par décomposition temporelle et problèmes d'ordonnancement*. Thèse de Doctorat, Institut National Polytechnique de Toulouse.
- [**Liberatore et Titus, 1983**] Liberatore, M. & Titus, G., "The Practice of Management Science in R&D Project Management", *Management Science*. 29 (8), (1983), 962-974.
- [**Little, 1970**] Little, J. D. (1970). "Models and Managers: The Concept of a Decision Calculus." *Management Science*.
- [**Lorino, 1992**] Philippe Lorino, "La gestion par les activités", dans « Évaluer pour évoluer » du Séminaire AFGI du 15 octobre 1992.
- [**Lorino, 1997**] Philippe Lorino, "Méthodes et pratiques de la performance – Le guide du pilotage", Les Editions d'Organisation, 512 pages, 1997.
- [**Madkit, 2001**] MADKIT.<http://www.madkit.org>.
- [**Malone, 1988**] :Malone T.W., « *What is coordination theory* », in national science foundation coordination theory workshop, MIT, 1988.

[Malone, 90] T. W. Malone. Organizing information processing systems : parallels between human organizations and computer systems. In W. W. Zachary and S. P. Robertson, editors, *Cognition, Computation and Cooperation*, pages 56-83. Ablex, 1990.

[Malone & Crowston, 90] Malone, T. W., & Crowston K. (1990). What is coordination theory and how can it help design cooperative work systems? (Tatar, Deborah, Ed.) Proceedings of the Third Conference on Computer-supported Cooperative Work (CSCW). 357–370.

[Malone et Crowston, 1994]: Malone, T.W., Crowston, K.: « *The interdisciplinary study of coordination* ». ACM Compus.Surv., 26 (1): 87-119, 1994.

[Marris, 1996] « le management par les contraintes » éditions d'organisation, 1996. P. Marris.

[Mazouz, 2001] H. Mazouz. "Ingénierie des protocoles d'interaction des systèmes distribués aux systèmes multi-agents". Thèse de Doctorat, Université Paris IX-Dauphine, 2001.

[Mélèze, 1972] Jacques Mélèze, "L'analyse modulaire des systèmes de gestion", A.M.S., Paris, Éditions Hommes et Techniques, 1972.

[Mesghouni, 1999] Mesghouni, K. 1999. *Applications des algorithmes évolutionnistes dans les problèmes d'optimisation en ordonnancement de la production*. Thèse de Doctorat, Université des Sciences et Technologies de Lille, Laboratoire d'Automatique et d'Informatique Industrielle de Lille.

[Mintzberg 79] H. Mintzberg. *The Structuring of Organizations*. Englewoods Cliffs, 1979.

[Mitra et Valente, 2007] Mitra G., Valente P., "The evolution of Web-based optimization from ASP to e-services", *Decision Support Systems*, Vol. 43, 2007, p. 1096-1116.

[Molet, 1997] « une nouvelle gestion industrielle » édition Hermès, 2ème édition, 1997. H. Mollet.

[Morton, 71] Morton M.S. *Management Decision Systems; Computer-based support for decision making*. Boston, Division of Research, Graduate School of Business Administration, Harvard University, 1971.

[MÛL, 96] MÛLLER .H.J "Negotiation principles" dans O'hare et Jennings (dir), *Foundations of distributed artificial intelligence* p.211-229.1996.

[Myers, 1995] Myers, B. A. (1995). "User Interface Software Tools." *ACM Transactions on Computer- Human Interaction*. 2(1): p. 64-103.

[**Ngai et Wat, 2005**] Ngai E.W.T., Wat F.K.T., “Fuzzy decision support system for risk analysis in e-commerce development”, *Decision Support Systems*, Vol. 40, No. 2, 2005, p. 235-255.

[**Nodin, W. Bohrer, and A. Hiong. 1998**] M. Nodin, W. Bohrer, and A. Hiong Ngu. Semantic brokering over dynamic heterogeneous data sources in infosleuth. Technical Report? MCC Corporation, 1998.

[**Ocello, 2003**] M. Ocello and Y. Demazeau. Building real time agent using parallel Blackboards and its use for mobile robotics. In IEEE int. conf. on systems. Man and cybernetics. San Antonio. Texas. USA. October. 1994

[**OSS, 99**] Ossowski. S co-ordination in Artificial agent societies, Springer Verlag LNAI 1535. 1999.

[**Palanque et al., 1994**] Palanque, P., J. B. Long, J. C. Tarby, M. F. Barthet et K. Y. Lim (1994). Conception d'applications ergonomiques : une méthode pour informaticiens et une méthode pour ergonomes. Actes du congrès ERGO-IA'94 (Ergonomie et Informatique Avancée), Biarritz, France.

[**Pellet, 1985**] X. Pellet, “Sur la hiérarchisation des décisions- Application à la conduite d'ateliers”, Thèse de doctorat, Institut national polytechnique de Grenoble, 1985.

[**Portmann & Bolzoni, 1994**] Portmann, M.C., Bolzoni, V. 1994. Specific developments of general software adaptation in order to solve some scheduling or production manufacturing problems. 4th *International Workshop on Project Management and Scheduling*, Leuven, pp. 192- 197.

[**Power, 1997**]: Power, D.J. : « *What is DSS ?* », The On-line Executive Journal for Data Intensive Decision Support 1 (3). 1997.

[**Power, 02**] Power D.J., *Decision Support Systems: Concepts and Resources for Managers*, Quorum Books, Westport Conn. 2002.

[**Probst, 1984**] Probst, A. R. (1984). "Les systèmes d'aide à la décision: rôle, structure et évolution." *Revue Gestion*: p. 13-19. Reix, R. (2000).

[**RAO, 91**] A. S. Rao and M. P. Georgeff. Asymmetry thesis and side-effect problems in linear time and branching time intention logics. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, pages 498-504, Sydney, Australia, 1991.

[**Ray, 2007**] Ray J.J., “A Web-based spatial Decision Support System optimizes routes for oversized/overweight vehicles in Delaware”, *Decision Support Systems*, Vol. 43, 2007, p. 1171-1185.

[Reix, 2000] Reix, R. (2000). Systèmes d'information et management des organisations. 3ième éd., Paris, Vuibert.

[ROS, 82] Rosenschein J.S. « Synchronization of Multi-Agents Plans » dans *Proceedings of AAAI'82*. P 115-119. 1982.

[Rudnicka et Madey, 2001] Rudnicka, A. et G. R. Madey (2001). A Framework for effective user interface design for web-based electronic commerce applications. Conference Preproceedings, 2001 Informing Science Conference, June 19-22 Kraków, Poland.

[VMA, 92] F. Von Martial, editor. *Coordinating Plans of Autonomous Agents*. Springer-Verlag :Heidelberg, Germany, 1992.

[Sage, 1991] Sage, A.P. (1991). Decision Support System Engineering. New York, Wiley.

[Sundarraaj, 2004] Sundarraaj, R.P.: "A Web-based AHP approach to standardize".

[Schroff [1998] Schroff, A. (1998). An Approach to User Oriented Decision Support Systems. Inaugural-Dissertation Nr. 1208, Druckerei Horn, Bruchsal.

[Simon, 1977] Simon H.A. The new science of management decision. Prentice Hall, New Jersey, 1977 systems development in an emerging economy. Decision Support Systems, vol. 42 (4), pp.

[SEN, 00] Sen.S., Sarathi dutta P., " searching for optimal coalition structures" dans Proc for ICMAS 2000. IEEE press. Boston. Massachusetts. 2000

[Shaw et al., 2002] Shaw N.G., Mian A., Yadav S.B., "A comprehensive agent-based architecture for intelligent information retrieval in a distributed heterogeneous environment", *Decision Support System*, Vol. 32, No. 4, 2002, p. 34-42.

[Shrague et Baker, 1978] Shrague, L.; Baker, K. (1978), Dynamic programming solution of sequencing problems with precedence constraints, *Operations Research*, Vol. 26, p.444-449.

[SYC, 89] Sycara K., "Argumentation : planning other agents' plan" in proceeding of IJCAT 86. Detroit Michigan. P. 517-523. 1989

[Tacquard et al., 1995] C. Tacquard, P. Baptiste et H. Manier, "A behavioral model to schedule and control a flexible manufacturing ring: application to the push and the pull flows", International conference on Industrial Engineering and Production Management, Marrakech, Maroc, pp. 127-136, 1995.

[Taghezout, 2010] Taghezout N., Bouamrane k., Bessedik I., "Decentralized Multi-Agent Coordination In a Web-based DSS: Application to Resource Allocation Problems", *Third International Conference on Web and Information Technology (ICWIT)*, ISBN 978-9954-9083-0-3, 16-19 June 2010a, Marrakech Morocco, p. 405-416.

[**Taghezout, 2011**] thèse de doctorat intitulée « *conception et développement d'un système multi-agents d'aide à la décision pour l'aide à la gestion dynamique* » soutenue en 2011.

[**Trentesaux, 1996**] Damien Trentesaux, "Conception d'un système de pilotage distribué, supervisé et multicritère pour les systèmes automatisés de production", Thèse de Doctorat en Automatique - Productique, Institut National Polytechnique de Grenoble, soutenue le 24 janvier 1996.

[**Totton et Flavin, 1991**] Totton, K. A. E. et P. G. Flavin (1991). "An overview of Computer Aided Decision Support."

[**Turban, 1993**] E. Turban. "Decision Support and Expert Systems", New York, Macmillan.

[**Turban et al., 95**] Turban E., et Walls J.G. Executive information systems - a special issue. Decision Support Systems, vol. 14 (2), pp. 85-88, 1995.

[**Turban et al., 98**] Turban, E., & Aronson J., (1998), Decision Support System and intelligent Systems, USA, Prentice-Hall, Inc., Upper Saddle River, Fifth Edition.

[**Vidal et al., 1998**] M.J. Vidal, T. Müller, P. Weinstein, and Durfee H. E. The University of Michigan Digital Library service market society. In K. P. Sycara and M. Wooldridge, editors, *Proc. of the 2nd Int. Conf. on Autonomous Agent (Agent-98)*, pages 269-276, 1998.

[**Von Martial, 90**] F. Von Martial (1990). « Interactions among autonomous planning agents ».

[**Watson et al. 93**] Watson, W. E., Kumar, K., & Michaelsen, L. K. 1993. Cultural diversity's impact on interaction process and performance: Comparing homogeneous and diverse task groups. *Academy of Management Journal*, 36: 590-602.

[**Widmer, 1991**] M. Widmer « Modèles mathématiques pour une gestion efficace des ateliers flexibles » Editions presses polytechniques romandes, 1991.

[**Woodward, 1965**] Woodward J., *Industrial Organization : theory and practice*, Oxford University Press, Oxford, 1965.

[**Zhang et Goddard, 2007**]: Zhang, S., Goddard, S.: « *A software architecture and framework for Web-based distributed decision systems* », *Decision Support Systems* 43, doi:10.1016/j.dss.2005.06.001, pp. 1133-1150, 2007.

[**ZLO, 96**] Zlokting. G., Rosenshen J.S " Mechanism design for automated negotiation and its application to task oriented domain" , *Artificial intelligence*, 86, p 195-244. 1996.