

# Ordonnancement d'atelier coopératif et réactif : une approche multi-agents

Bernard ESPINASSE - Erwan TRANVOUEZ

*DIAM-IUSPIM, Université Aix-Marseille, Campus Scientifique de Saint Jérôme, 13397, Marseille, Cedex 20, FRANCE,  
tel: ++33 (0)4 91 05 60 30 - fax: ++33 (0)4 91 05 60 33  
E-mail : bernard.espinasse@iuspim.u-3mrs.fr*

---

**Résumé :** Pour préserver leur compétitivité, les entreprises manufacturières adoptent de nouveaux types d'organisation leur permettant d'accroître leur flexibilité et leur réactivité (ingénierie concurrente, entreprise virtuelle ou étendue, ...). Ces nouveaux modes d'organisation tendent à distribuer les activités, les compétences, les responsabilités et capacités décisionnelles de l'entreprise sur des unités d'un degré d'autonomie important. La coopération de ces unités (parfois le temps d'un projet) nécessite la coordination de processus décisionnels distribués. Notre recherche s'intéresse à ce type de coopération dans le cadre de l'ordonnancement d'atelier. Dans un premier temps nous présentons un modèle coopératif et réactif d'ordonnancement d'atelier de type "job-shop" structuré en trois niveaux décisionnels distincts et permettant, par coopération d'unités définies à ces niveaux, d'absorber des perturbations. Ensuite nous nous focalisons sur le traitement par coopération de perturbations internes à une cellule de production composée de machines disposant d'un certain niveau d'autonomie. Nous distinguons différents types de coopérations possibles entre les machines permettant d'absorber une perturbation interne et nous développons plus en détail le cas des coopérations possibles entre deux machines. Dans la dernière partie, nous présentons un système multi-agents, actuellement en cours de développement, simulant l'ordonnancement réactif et coopératif d'une cellule de production selon le modèle précédent. Nous développons la structure interne des agents de ce système ainsi que les protocoles supportant leur coopération. Enfin, nous concluons sur les limites et perspectives de notre recherche.

**Mots clé :** système de production distribué, ordonnancement réactif, coopération, négociation , systèmes multi-agents.

**Abstract :** To preserve their competitiveness, manufacturing enterprises adopt new types of organisation allowing them to increase their flexibility and their reactivity (Concurrent Engineering, Virtual or Extended Company, Holonic Manufacturing Systems, ...). These new organisation modes tend to distribute activities, competence, responsibilities and decision making capacities of the enterprise on units having a high level of autonomy. The cooperation (sometimes as long as a project lasts) of these units requires the coordination of distributed decision-processes. Our research concerns the study of this cooperation category in a workshop scheduling context. First we present a "job-shop like" workshop scheduling model structured according to three decision levels and enabling thus to absorb through cooperation disruptions. Then we focus on the treatment by cooperation of perturbations internal to a production cell, cell which is composed of machines having some autonomy. We develop in detail the different possible cooperations between machines allowing them to absorb an internal perturbation. In the final part, we present a multiagents system, currently in development, simulating the reactive and cooperative scheduling of a production cell according to the preceding model. We describe the internal structure of agents which compose this system and the protocols supporting agents cooperation. Finally, we conclude on limits and perspectives of our research.

**Key Words :** Distributed Manufacturing Systems, Reactive Scheduling, Distributed Decision Making, Cooperation, Negotiation, Multiagent Systems.

---

## 1. Introduction

L'entreprise manufacturière est confrontée de plus en plus à un marché global et à la prise en compte d'une production centrée client. Ce contexte de production se traduit par la demande de produits de plus grande qualité, disponibles dans des délais minimaux, moins chers et personnalisés [Vernadat 94][de Terssac & al. 92]. Pour faire face à ces exigences, les entreprises manufacturières adoptent de nouveaux types d'organisation leur permettant d'accroître leur flexibilité et leur réactivité,

citons notamment l'entreprise étendue (extend enterprise) [Browne & al.94], l'entreprise virtuelle (virtual enterprise) [Rolstadås 94], l'ingénierie concourante [Molina & al. 94], les systèmes holoniques de manufacture (holonic manufacturing systems) [Herath 94] [Deen 94] et enfin le D-CIM (Distributed-CIM)[Skjeullaug & al.90].

Ces nouveaux types d'organisation ont de nombreuses similitudes, ils mettent notamment l'accent sur deux points fondamentaux : (i) la distribution de responsabilités et de capacités décisionnelles de l'entreprise conduisant à la caractérisation d'unités autonomes internes ou externes à celle-ci et (ii) le développement de politiques de coopération entre ces unités. Ainsi, une entreprise, ses différents départements, ses sous traitants, ses clients et ses fournisseurs peuvent être appréhendés comme un réseau dont les nœuds sont des "unités autonomes" qui coopèrent et constituent ainsi l'entreprise manufacturière distribuée. La notion d'unité autonome peut aussi être appliquée à un niveau d'abstraction plus bas, aux machines d'une chaîne de fabrication ou de montage qui coopèrent. La coopération entre ces unités autonomes nécessite la coordination de processus décisionnels eux-mêmes distribués. Les possibilités offertes par les nouvelles techniques de traitements de l'information et des connaissances apportent de nouveaux challenges, notamment celui de développer de nouveaux outils informatisés, systèmes d'aide à la décision distribués, supportant cette coordination d'activités organisationnelles (humaines et/ou artificielles) se développant dans des environnements fondamentalement distribués.

Dans la conception et la réalisation de tels outils, les systèmes multi-agents (SMA), issus des développements de l'Intelligence Artificielle Distribuée (IAD) [Bond & Gasser 88] s'avèrent des plus pertinents : (i) leur puissance en tant qu'outil de modélisation de connaissances, plus précisément leur capacité à décrire des domaines complexes et leur souplesse permettant la mise à jour des modèles ; (ii) leur capacité d'exécution et/ou de simulation permettant la validation de modèles et (iii) leur capacité à appréhender des notions telles que la distribution, la coopération, la coordination, la négociation. L'usage des SMA comme outil de modélisation pour décrire des systèmes industriels complexes n'est pas nouveau [Fox 85] [Jennings 94] [Camarinha-Matos 94] [Spinosa & al.95]. Un nombre croissant d'applications industrielles apportent des solutions originales et efficaces dans les domaines de la gestion de production, de l'ingénierie simultanée, du pilotage ou encore de la gestion de systèmes d'information [Kouiss & Pierreval 95], [Parunak 96], [Chaib-draa 95].

Notre recherche s'inscrit dans cette problématique et concerne l'usage des SMA dans le domaine de l'ordonnancement d'atelier et plus précisément de type *job-shop*. L'ordonnancement en général constitue un terrain fertile en méthodes de résolution et est l'objet de nombreux travaux pluridisciplinaires [Gotha 93][Slimani & Ramudhin 95]. Ordonnancer un ensemble de *tâches*, c'est programmer leur exécution en leur allouant les *ressources* requises en respectant un ensemble de contraintes temporelles. Dans le cas du problème d'atelier, par "tâches" il faut comprendre les opérations élémentaires (usinage, fraisage, coupage...) et par "ressources", les machines sur lesquelles ces tâches seront effectuées. Un travail (job) correspond alors à une liste ordonnée d'opérations. Il s'agit ainsi de proposer une organisation de l'emploi du temps des machines telle que les *contraintes potentielles* propres à chaque opération (*contrainte d'antériorité* : il existe une contrainte d'antériorité entre deux opérations  $i$  et  $j$ , si l'opération  $i$  ne peut être commencée avant la fin de l'opération  $j$ : " $i.>.j$ " - *contrainte de localisation temporelle* : la date au plus tôt (respectivement la date au plus tard) à laquelle doit commencer (respectivement doit finir) une opération), soient respectées [Carlier & Chrétienne 88].

Les problèmes d'ordonnancement d'atelier présentent les caractéristiques suivantes : (i) pas de contrainte d'antériorité entre opérations de travaux (jobs) différents, (ii) hypothèse de non préemption (non-morcelabilité des opérations), et enfin (iii) un travail (job) ne peut être effectué que sur une seule machine à la fois, auxquelles on rajoute, pour le cas du job shop, (iv) l'hypothèse que chaque produit ou famille de produits possède une gamme de fabrication spécifique [Blazewicz & al. 96], [Gotha 93]. L'ordonnancement sera qualifié de prédictif ou statique s'il est calculé (ou obtenu) à partir de données prévisionnelles (on considère alors que rien ne viendra les perturber) et ce off-line (ou hors-ligne). A contrario, on parlera d'ordonnancement réactif ou dynamique (on / off-line) si celui-ci est effectué en temps réel [Gotha 93].

Dans le cadre de notre recherche, nous nous situons dans le cas où un ordonnancement prévisionnel existe. Notre objectif n'est pas de construire (du moins dans l'immédiat) un ordonnancement de manière réactive [Burke & Prosser 94], mais plutôt de développer un modèle et les outils associés permettant d'absorber, par coopération entre unités autonomes, des perturbations internes et externes aux unités. Dans un premier temps nous présentons un modèle coopératif et réactif d'ordonnancement d'atelier type "job-shop" articulé selon trois niveaux décisionnels et permettant par coopération entre

unités autonomes, d'absorber des perturbations. Ensuite, en nous situant au niveau d'une cellule composée de machines disposant d'un certain niveau d'autonomie (niveau décisionnel 3), nous nous focalisons sur le traitement par coopération de perturbations internes à la cellule. Nous distinguons différents types de coopérations possibles et nous développons plus en détail celles impliquant seulement deux machines. Dans la dernière partie, nous présentons un système multi-agents, actuellement en cours de développement, simulant l'ordonnancement réactif et coopératif d'une cellule de production selon le modèle précédent. Nous développons la structure interne des agents de ce système ainsi que les protocoles supportant leur coopération. Enfin, nous concluons sur les limites et perspectives de notre recherche.

## **2. Un modèle d'ordonnancement d'atelier coopératif et réactif**

Notre modèle d'ordonnancement réactif et coopératif d'atelier de type "job-shop" a pour ambition, à travers la coopération d'unités disposant d'un certain niveau d'autonomie, cellules et machines composant ces dernières, d'absorber des perturbations internes ou externes. Nous précisons tout d'abord les trois niveaux décisionnels sur lesquels notre modèle s'articule et auxquels sont associés des niveaux d'autonomie et de réactivité. Ensuite nous développons comment, par différentes étapes caractérisées par des coopérations intra ou inter niveaux, l'absorption d'une perturbation interne ou externe à une cellule de production est possible.

### **2.1. Une organisation à trois niveaux décisionnels**

Nous avons cherché à distribuer dans le système de production les capacités décisionnelles et cognitives selon trois niveaux décisionnels distincts. A chacun de ces niveaux est associé un niveau d'autonomie et des capacités cognitives permettant cette autonomie. L'intérêt d'une telle approche est de fournir aux unités d'une organisation industrielle les moyens de résoudre les problèmes dès leur apparition, et ce au niveau décisionnel le plus bas. Ainsi, en diminuant les flux d'information entre les différents niveaux décisionnels, nous réduisons d'autant le temps de réaction du système et par la même l'impact des perturbations. Les trois niveaux décisionnels que nous proposons s'inspirent des principes développés dans [Kallel & al. 85] :

- le niveau 1 assure la gestion globale d'un certain nombre de cellules de production. Il se charge de l'évaluation des besoins et les moyens pour les satisfaire, il élabore et transmet les plans de fabrication aux cellules sous sa responsabilité (niveau 2).
- le niveau 2 se situe au niveau de la cellule de production et constitue un centre décisionnel local. Il assure la gestion de la cellule et il joue le rôle de coordinateur entre les machines de la cellule mais aussi d'interface de communication avec le niveau 1 et avec les autres cellules pour d'éventuelles coopérations.
- le niveau 3 est associé aux processus décisionnels de plus bas niveau. Il est assimilé aux machines composant la cellule de production. Les capacités cognitives qui lui sont associées sont relativement limitées comparées à celles associées aux niveaux précédents. Ce niveau sera le plus souvent le premier à être confronté aux perturbations et à essayer de les résorber.

Ces trois niveaux décisionnels permettent de mesurer le degré d'importance d'une perturbation. Une perturbation non maîtrisée à un niveau  $n$  nécessitera l'utilisation de méthodes de résolution disponibles au niveau immédiatement supérieur ( $n-1$ ). Suivant le nombre de niveaux décisionnels impliqués (1, 2 ou 3), une perturbation sera considérée comme faiblement, moyennement ou fortement perturbante comme le propose [Baillet 94]. Cette architecture à trois niveaux est illustrée par la figure 1.

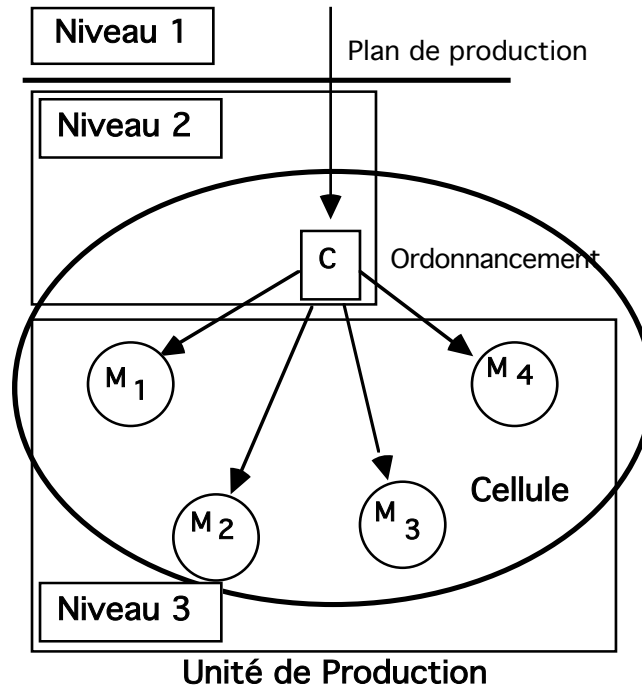


Figure 1 : Les différents niveaux décisionnels.

## 2.2. Gestion d'une perturbation interne à une cellule par coopération

A travers ces trois niveaux nous espérons aboutir à un haut niveau de flexibilité en limitant l'impact et le traitement d'une perturbation au niveau décisionnel le plus bas possible, afin que celle-ci ne se propage pas à l'ensemble du système de production. En nous intéressant au traitement d'une perturbation interne à une cellule de production (niveau 3), nous proposons une démarche de réaction s'articulant autour de sept étapes chronologiques distinctes :

- Étape 1 : la perturbation est limitée et est absorbée par la machine elle-même grâce à ces capacités cognitives;
- Étape 2 : la perturbation est absorbée par coopération entre la machine perturbée et une autre machine de la cellule possédant (ou non) des fonctionnalités en commun (étapes 2.a et 2.b);
- Étape 3 : la perturbation est absorbée par coopération entre l'ensemble des machines de la cellule de production;
- Étape 4 : la perturbation est absorbée par le centre de décision local à la cellule à travers un réordonnancement total (niveau 2);
- Étape 5 : la perturbation est absorbée grâce à une coopération avec d'autres cellules de production (coopérations aux niveaux 2 et 3);
- Étape 6 : il n'est pas possible d'absorber la perturbation au niveau de la cellule ou par coopération entre cellules, il faut alors passer au niveau décisionnel 1.

Ces sept étapes sont illustrées par la figure 2.

Développons maintenant chacune de ces sept étapes :

**Étape 1 :** La perturbation est perçue par une machine (ici M4). Celle-ci vérifie d'abord si cette perturbation peut être absorbée par ses marges de fonctionnement [Baillet 94]. Si ce n'est pas le cas, alors la machine essaye de la résorber en effectuant une permutation temporelle de certaines tâches. Il est néanmoins nécessaire de vérifier si cette permutation ne remet pas en cause les ordres de fabrication des autres machines de la cellule.

**Étapes 2-3 :** Pour ces étapes, nous passons à une phase de résolution du problème d'ordonnancement distribuée et coopérative au travers de tractations entre les différentes machines de la cellule : négociations de contraintes, transferts ou permutations de tâches entre machines. Nous développons plus en détail ces étapes dans la partie suivante.

**Étape 4 :** Si toutes les tentatives de coopération pour absorber la perturbation ont échoué, la machine perturbée renvoie au centre de décision de la cellule un constat d'échec (4.a). Ce centre tente alors d'absorber la perturbation par le calcul d'un réordonnancement complet des charges de travail des machines en respectant les contraintes associées au plan de production

fixé par le niveau 1 (4.b). Notons que les étapes précédentes (1 à 3) réalisent aussi des ordonnancements partiels et de complexité croissante, définissant une échelle de mesure du niveau de complexité d'un calcul d'ordonnancement, basée sur l'importance de la zone d'influence de la perturbation. Ainsi, si le niveau le plus bas, i.e. l'étape 1, présente une zone d'influence limitée à une machine, l'étape 5 voit cette zone s'étendre à toute l'unité de production.

**Étape 5 :** Le centre de décision local de la cellule n'ayant pas réussi à absorber la perturbation par un réordonnancement, il s'adresse aux autres cellules de production existantes et susceptibles de l'aider. Les propositions faites par les cellules contactées sont ensuite évaluées par le centre de décision de la cellule. Deux niveaux d'intervention peuvent être identifiés. Le premier consiste à déterminer la (ou les) tâche(s) impliquée(s) dans la perturbation interne, et à les transmettre aux autres cellules sous forme de contraintes/perturbations ou encore sous forme de plan de production. Le second, plus complexe, considère les cellules impliquées comme formant une macro-cellule sur laquelle on effectue un réordonnancement partiel. Il serait préférable néanmoins de vérifier si cette dernière approche ne présente pas un coût calculatoire trop important. Il ne faudrait pas que trop de flexibilité réduise la réactivité du système de production.

**Étape 6 :** Enfin, à ce stade, toutes les tentatives de coopération pour absorber la perturbation ont échoué. Il est alors nécessaire de répercuter la perturbation au niveau supérieur. Le centre de décision local se chargera alors de transmettre toutes les informations liées à la perturbation ainsi que ses conséquences.

On remarquera que le scénario précédent se base sur une perturbation limitée à une machine. Dans le cas où plusieurs machines sont impliquées, il suffit alors de procéder à tour de rôle. Le scénario semble assez robuste pour gérer plusieurs perturbations.

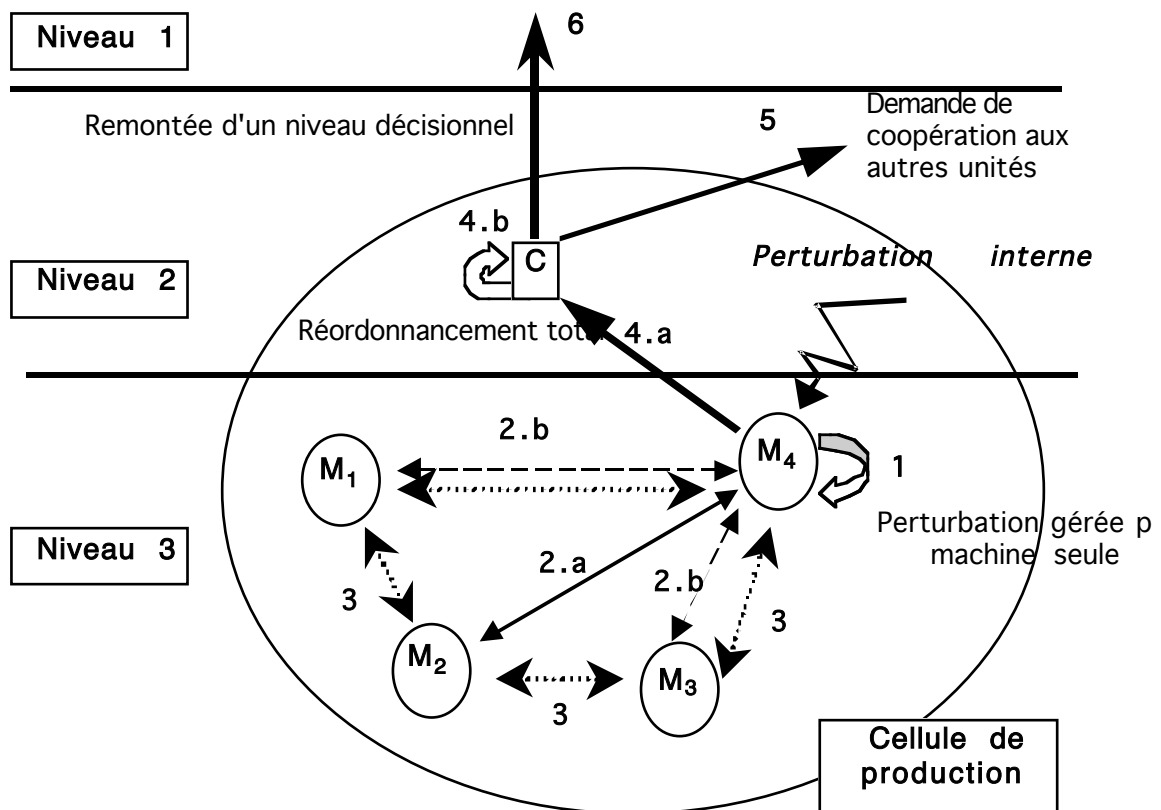
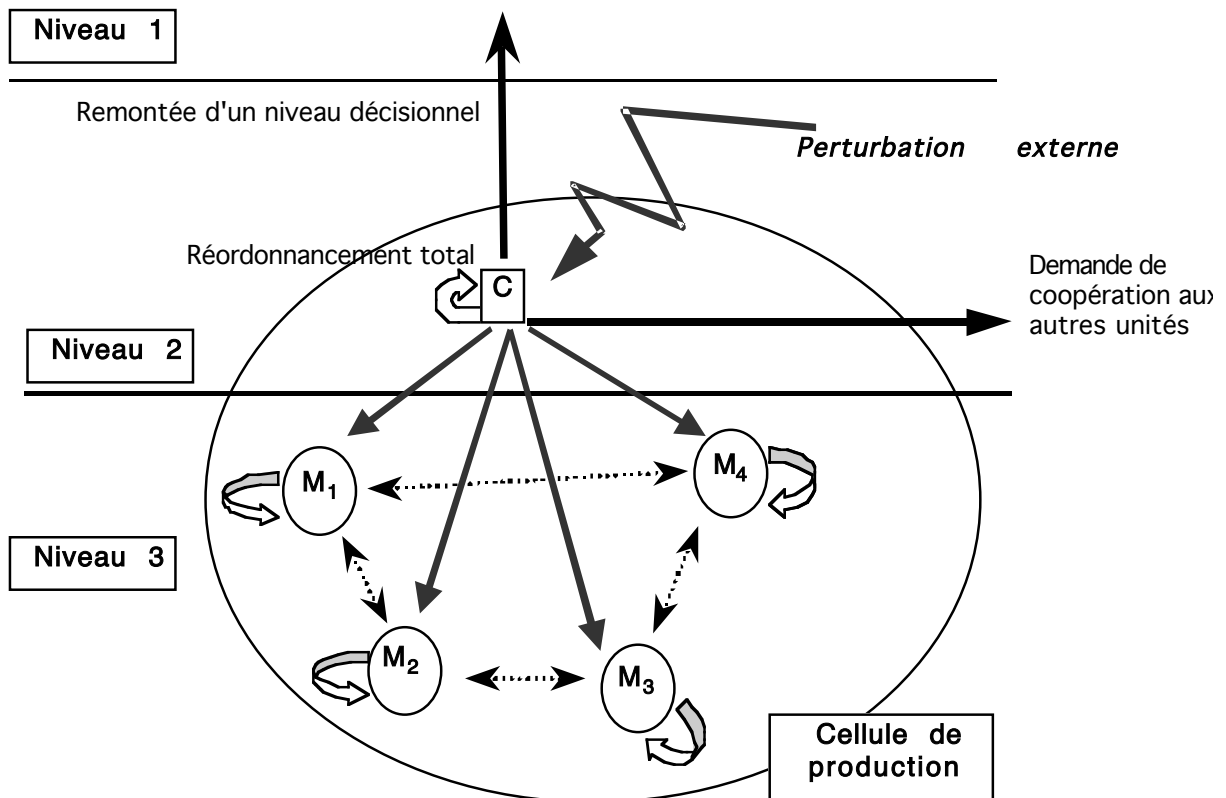


Figure 2 : Gestion d'une perturbation interne à une cellule.

### 2.3. Gestion d'une perturbation externe à une cellule par coopération

Le traitement d'une perturbation externe est assez similaire au traitement d'une perturbation interne. En effet, dans le cas d'une perturbation externe (cf. figure 3), celle-ci est directement perçue par le centre de décision local de la cellule (niveau 2). C'est ensuite, après évaluation, qu'il la répercute sur les machines de la cellule sous forme de perturbation interne qui pourra être traitée comme nous venons de le définir.



**Figure 3 :** Gestion d'une perturbation externe.

Dans le cadre de notre recherche nous n'avons pas étudié plus en détail le traitement d'une perturbation externe, pour porter notre effort essentiellement sur le traitement d'une perturbation interne au sein d'une cellule par coopération entre machines. En effet, face à la complexité du problème, il nous est apparu préférable de nous limiter, dans un premier temps, au cas d'une perturbation interne. Cette restriction n'est néanmoins que partielle, puisque toute avancée dans la définition des méthodes de résolution à ce niveau, se repercutera automatiquement sur tout le modèle.

### 3. Traitement d'une perturbation interne par coopération

On suppose que la machine perturbée n'a pas pu elle-même résorber la perturbation (étape 1). Elle va s'adresser aux machines appartenant à la même cellule de production qu'elle. Cet appel à coopération local se fera tout d'abord selon un principe de coopération/communication se limitant à deux machines (étape 2) puis pourra s'étendre à l'ensemble des machines de la cellule (étape 3). Il est cependant indispensable de vérifier que les contraintes d'antériorité sur un même travail (job) soient respectées, sans quoi on pourrait aboutir à des solutions irréalisables.

#### 3.1. Coopération Machine - Machine (étape 2)

Une machine en difficulté peut recourir à différents types de coopération possible. En considérant d'une part les résolutions centrées fonctionnalité et d'autre part celles centrées produit (ou job), nous distinguons deux catégories de méthodes de résolution coopérative.

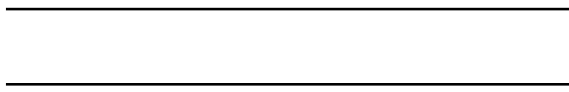
##### Coopération entre deux machines ayant des fonctionnalités en commun (étape 2.a)

Lorsqu'une machine subit une perturbation, c'est généralement un retard dans la réalisation d'une tâche sur un produit voire une panne. Dans le premier cas il lui faut prendre en compte ce retard et mesurer son impact sur la fabrication des produits qui suivent. On distingue deux types de coopération centrées fonctionnalité possibles entre deux machines : soit par transfert de charges, soit par permutation ou échange de tâches.

##### Coopération par transfert de charges entre machines

Dans cette coopération la machine perturbée demande aux machines disposant de la fonctionnalité de la tâche perturbée, si celles-ci peuvent se charger de la réaliser afin de lui permettre de respecter ses

contraintes temporelles. Si une machine sollicitée accepte de réaliser une tâche affectée à la machine perturbée, il y a transfert de charges de travail d'une machine sur une autre (cf. figure 4). A l'issue du transfert, les contraintes temporelles associées aux deux machines doivent être respectées.

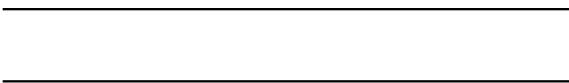


**Figure 4** : Transfert de charges entre machine de mêmes fonctionnalités.

Dans la figure 4, la machine M2 se charge d'effectuer l'opération 2 appartenant initialement à la machine M1. Il n'y a pas de contrepartie, il s'agit d'un simple transfert de charge.

#### Coopération par échange de tâches

Cette coopération est plus évoluée que la précédente : la machine perturbée demande à d'autres machines de se charger de la réalisation d'une ou plusieurs tâches et, en contrepartie, elle (la machine perturbée) se propose de réaliser une ou plusieurs tâches que l'autre ou les autres machines devaient réaliser. De tels échanges de tâches peuvent conduire à une négociation. A l'issue de l'échange, toutes les contraintes temporelles associées aux deux machines doivent être respectées. Comme l'illustre la figure 5, la machine M2 aidera la machine M1 si cette dernière peut prendre en charge les modifications qui découleraient de cette coopération. Il faudra alors prendre en compte les événements/dates signalés par la flèche, i.e. les dates au plus tôt du produit/opération perturbé ainsi que celui amené à le remplacer, la date à partir de laquelle la machine M1 redevient disponible (partie de la perturbation non absorbée par la marge) etc. En fait cela consiste à s'assurer que toutes les contraintes (temporelles et de ressource) sont vérifiées.



**Figure 5** : Permutation/échange inter machines de tâches.

On notera que dans les deux cas il sera impératif de vérifier que cette coopération ne provoque pas de perturbations supplémentaires. En effet, un transfert ou un échange de traitement pourrait se traduire par un décalage sur la chaîne de production, réduisant par la même l'intérêt d'une telle approche.

#### **Coopération entre deux machines ayant un produit en commun (étape 2.b)**

Dans le cas où la coopération avec des machines de mêmes fonctionnalités n'est pas possible ou ne permet pas d'absorber la perturbation, la machine perturbée va tenter de coopérer avec des machines traitant les mêmes jobs. Cette coopération est plus complexe. La machine perturbée pourra par exemple demander à une autre machine, de permuter certaines opérations afin qu'elle puisse faire de même, tout en respectant les contraintes potentielles. Une solution consiste à réaliser une réorganisation par permutation de tâches, des plans de fabrication de chacune des machines concernées, comme l'illustre la figure 6.

---

---

**Figure 6** : Réorganisation des plans de fabrication par permutation de tâches.

Dans la figure 6, après vérification des contraintes, ici une contrainte d'antériorité sur le produit 3, les machines M1 et M2 s'accordent pour réorganiser leur plan de fabrication en parallèle.

### 3.2. Coopération entre plus de deux machines (étape 3)

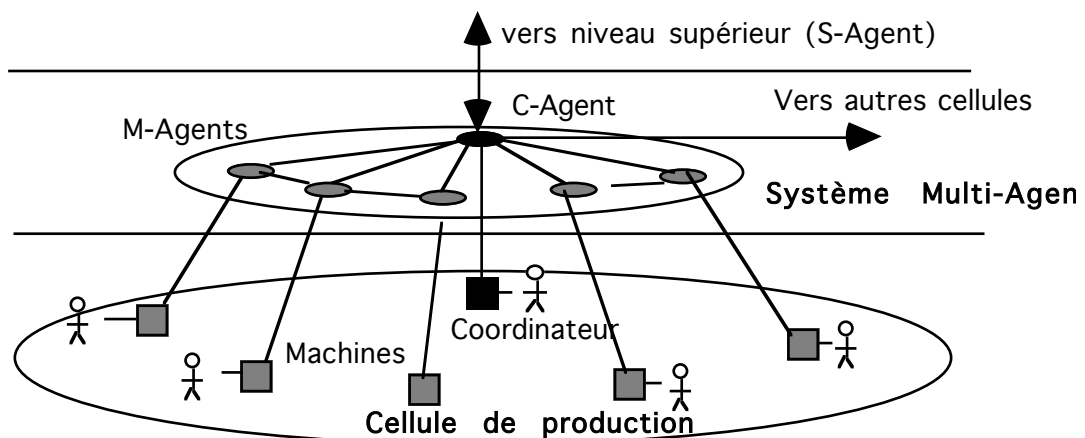
Lorsque la coopération entre deux machines ne permet pas d'absorber la perturbation, on peut envisager une coopération à plus grande échelle. Toutes les machines se concertent afin d'élaborer une solution absorbant la perturbation et acceptable/réalisable par toutes les machines. Cela pourrait consister à explorer toutes les possibilités de permutation de tâche à la fois sur la machine perturbée mais aussi sur les autres. Cependant on peut d'ores et déjà observer qu'une telle approche risquerait de déboucher sur une explosion combinatoire des possibilités.

## 4. Un système multi-agents pour l'ordonnancement coopératif

Afin de tester notre modèle d'ordonnancement d'atelier type "job-shop" permettant par coopération d'absorber des perturbations nous développons actuellement un système multi-agents simulant l'ordonnancement réactif et coopératif d'une cellule de production. Avant de développer la structure interne des agents de ce système ainsi que les protocoles supportant leur coopération, nous présentons l'architecture multi-agents que nous avons retenue.

### 4.1. Architecture multi-agents retenue

Notre approche consiste donc à percevoir le problème d'ordonnancement comme naturellement distribué. La modélisation multi-agents du problème qui va suivre prend parfaitement en compte cette caractéristique. Non seulement la résolution de ce problème d'ordonnancement sera distribuée, mais elle sera en plus coopérative. Il sera donc nécessaire de s'intéresser à la descriptions des agents (acteurs du système) et à leur organisation (source d'interactions). L'une des contraintes incontournables de tout système multi-agents lors de son élaboration reste l'adaptabilité. En effet, une architecture multi-agents se doit de "coller" à la réalité mais aussi d'être capable de proposer des solutions. Dans cet article nous nous limiterons à la modélisation d'une cellule de production comme l'illustre la figure 7 :



**Figure 7** : Architecture générale du système multi-agents.



Au sein d'une cellule de production nous considérons deux types d'agents : un agent coordinateur ou "C-Agent" et un agent machine ou "M-Agent". Un troisième type d'agent, l'agent manager associé au niveau supérieur ou "S-Agent" peut aussi être introduit. Ce dernier correspond au niveau décisionnel le plus élevé et ne sera pas considéré ici, puisque nous nous sommes restreint dans cette recherche à la cellule de production.

Pour concevoir et développer notre système multi-agents nous nous sommes inspirés du paradigme de Programmation Orienté Agent (POA) défini par [Shoham 93]. Brièvement, la POA est une évolution de la Programmation Orienté Objet (POO), dans laquelle l'objet a été doté d'un "état mental". Par état mental il faut comprendre croyances, compétences et engagements (i.e. processus de décision). Cet anthropomorphisme de la programmation se justifie dans notre cas, puisqu'il s'agit bien d'appréhender les machines comme des agents autonomes et intelligents, comme nous le permet la POA de Shoham.

La plate-forme multi-agents que nous utilisons pour le développement de notre système est LALO [Gauvin & al.96] qui s'inspire du paradigme de Shoham. Cette plate-forme nous contraint dans la définition de la structure interne des agents ainsi que dans celle des interactions entre les agents. Nous retrouvons dans LALO la notion d'état mental (croyances, compétences ou capacités et engagements) appelées "capacités". Comme la plate-forme Agent-K [Davies & Edwards 94], LALO intègre le protocole de communication entre agents KQML (Knowledge Query and Manipulation Language). Dédié à l'origine à la gestion distribuée des bases de données [Finin et al 94], ce protocole propose un ensemble de performatives inspirées de la théorie des actes de langages, à partir desquelles peuvent être élaborés des messages.

Nous présentons ci-dessous l'état mental des deux types d'agents essentiels composant notre système, C-Agent et M-Agent. Conformément au paradigme de POA nous nous intéressons plus particulièrement aux croyances, aux capacités et aux engagements des agents ainsi qu'à leurs accointances. En ce qui concerne les choix ou décisions que seront amenés à prendre les agents, nous utiliserons les scénarios de coopération précédemment présentés.

## **4.2. L'agent coordinateur : C-Agent**

L'agent coordinateur, C-Agent, correspond au niveau décisionnel local (niveau 2). En tant que tel, il joue un rôle de supervision et de coordination. Il a pour rôle de contrôler le bon fonctionnement de la cellule dont il dépend. En cas d'échec de coopération local, il peut effectuer un réordonnancement local et gérer si nécessaire les communications et les négociations avec les autres cellules. Ces dernières lui enverront alors des propositions de coopération qu'il pourra évaluer, puis accepter ou refuser. Il pourra aussi traduire une perturbation externe en perturbations internes qu'il transmettra aux agents machines concernés.

Cet agent dispose d'une représentation partielle de l'univers qui l'environne, ceci au travers de ses croyances. Celles-ci sont constituées des connaissances sur les capacités des agents Machine de sa cellule et de leurs plans de production. Les accointances constituent aussi une partie de ses croyances, puisque c'est à partir d'elle qu'il va élaborer des processus de coopération. Elles comprennent l'agent Manager, les agents coordinateurs des autres cellules et les agents Machine appartenant à la cellule qu'il supervise.

A partir de ces éléments, l'agent coordinateur s'engagera auprès de ses accointances à effectuer certaines actions. Ainsi s'agissant du S-Agent, le C-Agent se charge de l'exécution du plan de fabrication que lui a fourni celui-ci. Dans le cas d'une coopération avec une autre cellule, il prend l'engagement de participer au traitement d'une opération ou d'un travail. En ce qui concerne les M-Agents, il calcule un réordonnancement local (engagement pris en cas d'échec de la coopération locale) et transmet des nouveaux ordres de fabrication aux agents Machines. Les communications qui en résultent concernent ainsi : (i) des appels d'offre envoyés aux autres cellules pour coopération; (ii) des constats d'échec de toutes les coopérations; (iii) la diffusion d'une perturbation externe; (iv) et enfin l'envoi d'un réordonnancement total.

## **4.3. L'agent machine : M-Agent**

L'agent machine, M-Agent, correspond au niveau décisionnel le plus bas (niveau 3), il est associé au contrôle d'une machine. Ses capacités sont de type physique et de type cognitif. Les capacités physiques concernent essentiellement la réalisation d'une opération élémentaire, les capacités

cognitives sont plus variées puisqu'elles concernent par exemple l'évaluation de l'importance d'une perturbation, la permutation de tâches sur la ligne de production et surtout la coopération avec les autres machines.

Le M-Agent s'engage à exécuter les ordres de fabrication provenant de son C-Agent ainsi qu'à coopérer avec les autres M-Agents, par exemple effectuer une permutation spatiale ou temporelle de tâches ou encore un transfert de charge. Afin de tenir et diffuser ses engagements, il est amené à échanger de l'information avec d'autres M-Agents pour s'assurer de la possibilité de réaliser un transfert ou une permutation de tâches et aussi avec son C-Agent pour l'informer d'un changement d'état (date au plus tard modifié etc.). Pour cela, le M-Agent possède une vision de son environnement, vision plus restreinte que celle dont dispose le C-Agent. Cette vision repose sur des accointances limitées au C-Agent dont il dépend ainsi qu'aux autres M-Agents de sa cellule. Ses croyances portent notamment sur des croyances nécessaires à la coopération, par exemple les capacités des autres M-Agents, ainsi que des croyances liées aux tâches dont il a la charge.

Afin de guider son action, il dispose aussi d'une fonction d'évaluation (ou encore fonction de préférence) lui permettant d'évaluer les solutions de coopération qu'il élabore ou qui lui sont proposées. Cette fonction d'évaluation est multi-critères. Les différents critères permettant d'évaluer une coopération peuvent être relatifs au nombre d'opérations impliquées dans la coopération, au nombre de machines impliquées dans la gestion de la perturbation, au retard encouru par la cellule, aux temps morts résultant etc. Une pondération de ces critères permet de spécifier des comportements spécifiques à un M-Agent donné. Ainsi pour une machine très spécialisée, la diminution des temps morts peut être un critère primordial. La mise en place d'une telle fonction d'évaluation fournit par ailleurs un "axe de référence" pour la machine durant les négociations auxquelles elle participe. Selon ses préférences, la machine peut orienter la négociation dans une direction ou une autre. De même, une machine sollicitée pour une coopération peut décider de ne pas participer en constatant, après évaluation, que le coût de sa participation dépasse un seuil critique.

Les capacités cognitives (appelée ainsi par opposition aux capacités physiques) d'un M-Agent peuvent être structurées en trois modules distincts : de communication, de génération et d'évaluation. Le module de communication assure la gestion des communications avec les autres M-Agents et le C-Agent, le module de génération élabore des solutions de coopération et le module d'évaluation évalue les solutions de coopération élaboré par le module de génération du M-Agent ou qui lui sont proposées par les autres M-Agents. Les agents étant définis, nous allons introduire le protocole de coopération définissant le cadre de leurs interactions.

#### **4.4. Exemple de protocole de coopération**

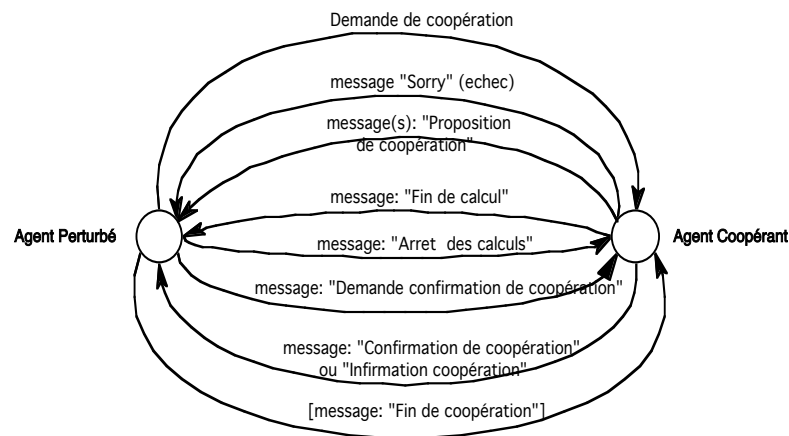
Il s'agit d'élaborer un contexte propre à favoriser la coopération entre les différents agents composant le système. Pour cela il est nécessaire de préciser les modes de communication entre agents ainsi que les protocoles de coopération. Notre projet se situe à mi-chemin entre la communication directe par message et la communication contractuelle "Contract Net" [Davis et Smith 81]. Afin de gérer les interactions dans le système multi-agents nous proposons différents protocoles de coopération prenant en compte, dans un premier temps, deux acteurs principaux : un agent perturbé et un agent candidat à la coopération, appelé agent coopérant. Nous espérons en effet, par la suite, étendre notre protocole à la coopération entre  $n$  agents (avec  $n > 2$ ).

Le protocole présenté ici correspond à l'étape 2. du scénario de gestion de perturbation interne. Ce protocole peut être appréhendé selon deux niveaux : un niveau global présentant les différentes interactions possibles entre un agent et un niveau plus détaillé présentant leurs impacts sur le comportement des agents du système. Il peut être appliqué aussi bien pour le cas de la coopération par transfert de tâche que pour le cas de coopération par échange de l'étape 2.b). Seuls changeront les types de messages (intitulé, information contenue) et le mode de calcul des solutions.

Dans ce schéma global ne sont présents que les interactions entre agents, i.e. tous les messages pouvant être échangés lors d'un processus de résolution. Ces messages seront représentés à l'aide de flèches. Ce schéma nous a amené à spécifier certains cas particuliers dont celui de la représentation de deux messages ayant des conditions d'envoi incompatibles (relation Ou-exclusive). C'est par exemple l'envoi d'un message signifiant l'échec d'un calcul de solution ne pourra cohabiter avec un message proposant une solution : cas sera représenté explicitement en donnant aux flèches respectives de ces messages des extrémités communes.

Par définition une coopération met en jeu au moins deux acteurs. Dans le cas des résolutions coopératives, nous considérerons un agent perturbé et  $p$  agents coopérants. Ces deux dénominations

d'agents ne correspondent pas à deux typologies exclusives mais à des rôles que peuvent remplir tout agent dans une situation de gestion d'une perturbation. Nous allons étudier ces deux rôles à travers deux graphes d'états étroitement reliés.



**Figure 8** : Protocole de coopération par échange: point de vue global.

Ces graphes d'états utilisent un formalisme adapté à notre approche, i.e. capable de pouvoir représenter les effets des interactions entre agents chez un agent particulier. Ainsi, nous proposons les définitions suivantes :

- *Transition interne* : transitions activées par des événements internes à l'agent (ex: fin d'un calcul, d'une identification, ...)
- *Transition externe* : transitions activées par des événements externes à l'agent (ex: signal d'un capteur, réception d'un message).
- *Etat actif* : Etat dont la désactivation ne dépend pas forcément d'un événement externe
- *Etat passif* : Il est associé à un état dans lequel l'agent attend un événement externe pour poursuivre son exécution. Typiquement, cela correspondra à des états d'attente de message.
- *Etat de communication* : Etat actif particulier, correspondant à une tâche d'émission de message. Ce type d'état représente donc avec les transitions externes le moteur des interactions engagées entre agents.

A l'aide de ce formalisme, nous présentons deux graphes d'états (figure 9), chacun correspondant à un rôle que tout agent peut endosser lors d'une résolution.

Nous avons vu au chapitre précédent, que le comportement d'un M-Agent en coopération s'élaborait dans l'interaction de ses trois modules de génération, d'évaluation et de communication. Ces trois modules dynamisent la coopération à travers une séquence proposition/évaluation/décision . Ainsi le module génération calcule et propose des scénarios (permutation de deux actions, transfert de charge...) pouvant constituer des solutions au problème d'ordonnancement rencontré. Il sera sollicité lors de la mise en place de la coopération par les agents coopérants et alimentera l'agent perturbé en propositions de coopération. Le module d'évaluation se charge de décider si une action est réalisable par l'agent et s'il a intérêt à l'effectuer. La faisabilité d'une action est déterminée par la vérification de la non violation des contraintes liées au plan de charge de la machine dont l'agent est responsable.

Remarquons que tous les aspects comportementaux des agents ne sont pas représentés dans ces deux graphes. Citons par exemple le calcul de solution qui diffère selon le type de coopération, ou encore la mise à jour des connaissances et de leur diffusion auprès des agents concernés. Cette dernière pose le problème de la maintenance de la cohérence des connaissances locales des agents aussi bien au niveau interne (existence de connaissances contradictoires au sein de l'agent) qu'au niveau externe à l'agent (connaissance sur son environnement inexacte ou dépassée). Ceci amène donc au développement de protocoles supplémentaires prenant en compte, encore une fois, les diverses interactions possibles entre les agents du système.



## 5. Conclusion

Nous avons tout d'abord proposé un modèle coopératif et réactif d'ordonnancement d'atelier de type "job-shop" structuré selon trois niveaux décisionnels distincts et permettant, par coopération d'unités définies à ces niveaux, d'absorber des perturbations. En nous intéressant au traitement d'une perturbation interne à une cellule de production composée de machines disposant d'un certain niveau d'autonomie (niveau 3), nous avons ensuite proposé une démarche de réaction s'articulant autour de sept étapes chronologiques distinctes auxquels sont associés différents types de coopérations. Nous avons limité notre étude aux coopérations entre deux machines d'une même cellule (étapes 2). Notre modèle n'est ainsi que partiellement développé tant au niveau du traitement d'une perturbation interne que d'une perturbation externe à la cellule de production. Nous entendons bien poursuivre son développement et le système multi-agents prototype, dont nous avons présenté l'architecture et qui est actuellement en cours de développement, devrait nous y aider fortement.

Si l'on considère les différentes méthodologies développées pour résoudre les problèmes d'ordonnancement, on remarquera la place importante de la notion de point de vue. En effet, selon que le problème soit perçu en terme de calculs combinatoires ([Gotha 93], d'aide à la décision ([Lopez & al. 96], [Bérard & al. 97]), d'allocation de ressource ([Shaw 87],[Pujot & al. 97]) ou encore de coopération comme nous l'avons choisi, il ne sera pas proposé les mêmes types de solution. Dans le domaine de l'ordonnancement réactif, il existe en fait très peu d'approches orientées coopération. A notre connaissance, elles s'intéressent toutes à une coopération circonscrite à des agents se situant à un niveau décisionnel relativement élevé. C'est le cas dans les travaux de Prosser [Prosser 89] dans lesquels la coopération intervient entre des "strategic agents" (correspondant à nos C-agents). D'autres travaux dont [Fisher & al.96] distinguent deux types de coopérations, l'une verticale (correspondant à une coordination entre un C-agent et un M-agent), indiquée pour une allocation de tâche et une autre horizontale se situant encore à un haut niveau d'agrégation d'information (les "Shipping Company Agents" - nos C-Agents). Dans notre approche, la coopération peut être mise en oeuvre dès le plus bas niveau décisionnel. Ainsi on tentera de traiter par coopération une perturbation au niveau décisionnel  $n$  auquel elle surgit. Si celle-ci ne peut être maîtrisée, on tentera de la traiter au niveau immédiatement supérieur ( $n-1$ ). En donnant aux agents de bas niveau des capacités décisionnelle et coopérative, il est possible d'absorber le plus tôt possible une perturbation en restreignant sa zone d'impact, ce qui conduit à un gain de réactivité et de flexibilité. Cependant, les solutions obtenues ne sont pas nécessairement optimales en émergeant d'une résolution locale. Notons que des approches assez similaires existent notamment en planification et ordonnancement dans le domaine du transport (coopération entre des agents "camion" de bas niveau) par [Sassi & Chaib-draa 97] ou encore dans la conduite de système de production [Pujot & al. 97].

Actuellement nous travaillons au développement de notre prototype de système d'ordonnancement coopératif multi-agents intra cellule. Plus précisément nous nous intéressons (i) à la représentation des connaissances des M-Agents (contraintes respectives, croyances, états, capacités, engagements possibles, préférences par une fonction multicritère adaptée...); (ii) au comportement cognitif de ces agents avec notamment la conception de ses modules de génération et d'évaluation (intégrant des fonctions d'évaluation, voire des règles de priorité [Mebarki & al. 96]) et enfin des mécanismes d'apprentissage permettant d'optimiser la qualité et la durée de la résolution); (iii) à étudier en détail les protocoles de coopération définissant les interactions entre ces agents; (iv) enfin à définir la coopération du système avec des décideurs humains associés à ces M-Agents. Ces résultats obtenus au niveau intra cellule devraient nous permettre d'appréhender par similitude, le niveau inter cellules mettant l'accent sur la coopération entre C-Agents notamment en les dotant de fonctions d'évaluation propres. Le traitement de cas réels avec notre prototype nous permettra de valider et de perfectionner notre modèle.

## 6. Remerciements

Les auteurs tiennent à remercier Jean Paul Kieffer, Aline Cauvin et Alain Ferrarini pour leurs commentaires constructifs ainsi que Jean Paul Chirac pour son efficace collaboration.

## 7. Références

- [Baillet 94], Baillet P., *Contribution à l'amélioration de la réactivité dans les systèmes de production, notamment par la mise en œuvre des concepts de décentralisation des fonctions de décisions*, Thèse de doctorat en Sciences, soutenue au DIAM, Déc. 1994.
- [Berard & al. 95], Bérard C., Deschamps J-C., Farthouat P., A cooperative approach to schedule manufacturing systems, *7th Mini Euro Conference*, Bruges, Belgique, 24-27 Mars 1997.
- [Blazewicz & al. 96], Blazewicz J., Wolfgang Domschke W., Pesch E., The job shop scheduling problem: Conventional and new solution techniques, *European Journal of Operational Research*, Vol 93, N°1, , Aug. 1996, pp. 1-33.
- [Bond & Gasser 88], Bond H.A., Gasser L., *Readings in Distributed Artificial Intelligence* , Morgan Kaufman, 1988.
- [Browne & al. 94], Browne J., Sackett J.C., Wortmann, Industry Requirements and Associateds Research Issue in the Extend Enterprise, *Proceedings of the IMSE'94 European Workshop on Integrated Manufacturing Systems Engineering*, Grenoble, France, December 12-14, 1994.
- [Burke & Prosser 94], Burke P., Prosser P., "The Distributed Asynchronous Scheduler", in *Intelligent Scheduling*, Monte Zweben & Mark S. Fox Eds, Morgan Kaufmann Publishers, 1994.
- [Camarinha-Matos 94], Camarinha-Matos L.M., "Lecture Note, Special Session on Multiagent Systems in Manufacturing", *EURISCON'94, European Robotics and Intelligent Systems Conference*, Malaga, aug 22-26, 1994, Spain.
- [Carlier & Chretienne 88], Carlier J., Chretienne P., *Problèmes d'ordonnancement : Modélisation Complexité, Algorithme*, Ed. Masson, 1988.
- [Chaib-draa 95], Chaib-draa B., "Industrial Applications in Distributed AI", *Communication of ACM*, 38(11), pp. 49-53, 1995.
- [Davies & Edwards 94], Davies W.H.E., Edwards P., Agent K : An Integration of AOP and KQML, Proceeding of CIKM'94, Workshop on Intelligent Information Agents, Y Labron & T. Finin (eds.), National Institute of Standards and Technology, Gaithersburg, Maryland, USA, 1994.
- [Davis & Smith 81], Davis R., Smith R.G., Frameworks for Cooperation in Distributed Problem Solving, *IEEE transactions on Systems, Man and Cybernetics*, 1981.
- [Davis & Smith 83], Davis R., Smith R.G., Negotiation as a metaphor for distributed problem solving, *Artificial Intelligence*, 20 (1), 1983.
- [de Terssac & al. 92], de Terssac G., Dubois P., *Les nouvelles rationalisations de la production*, Cépadués-Editions, 1992.
- [Deen 94], Deen S.M., Cooperation Issues in Holonic Manufacturing Systems, *Information Infrastructure Systems for Manufacturing (B-14)*, Yoshikawa H. and Goossenaerts J. (Eds.), Elsevier Science B.V., (North Holland), 1993.
- [Espinasse & al. 95], B.Espinasse, L.M. Spinoza, E. Chouraqui, D-CIM et IAD : une approche orientée connaissance pour la modélisation de systèmes de production, *Congrès International de Génie Industriel de Montréal : La productivité dans un monde sans frontière*, Montréal, Canada, 18-20 octobre 1995, Volume II, pp. 915-925.
- [Finin & al. 94], Finin T., Fritzon R., McKay D., McEntire R., KQML as an Agent Communication Language, *Proceedings of the Third International Conference on Information Management (CIKM'94)*, ACM Press, Nov. 1994.
- [Fisher & al. 96] Fisher K., Müller J.P., Pischel M., Cooperative transportation scheduling : an application domain for DAI, *Applied Artificial Intelligence*, Vol.10, pp 1-33, 1996.
- [Fox 85], Fox M.S., "Knowledge Representation for Decision Support Systems", in *Knowledge Representation for Decision Support Systems*, edited by L.B.Methlie and R.M.Sprague, North Holland, 1985.
- [Gauvin & al. 96], Gauvin D., Marchal H., Saldanha, LALO: a new Agent Oriented Programming Language and Environment, *Canadian Artificial Intelligence*, falls 96.
- [Gotha 93], Gotha, Les problèmes d'ordonnancement, *Recherche opérationnelle/Operations Research*, vol.27, n°1, 1993, p.77-150.

- [Herath 94], Herath A., Towards a CKBS Model for Holonic Manufacturing Environments, Report N° DAKE/TR-94002, DAKE Centre, University of Keele, UK, 1994.
- [Jennings 94], Jennings N.R., *Cooperation Industrial Multiagent Systems*, World Scientific Publishing Co., World Scientific Series in Computer Sciences, vol.43, 1994.
- [Kallel & al. 85], Kallel G., Pellet X., Z. Binder Z., Conduite décentralisée coordonnée d'atelier, *APII Hermès Ed.*, 1985; Vol. 19, pp. 371-387.
- [Kouiss & Pierreval 95], Kouiss K., Pierreval H., Systèmes multi-agents: direction actuelles pour les systèmes de productions., *Congrès International de Génie Industriel de Montréal : La productivité dans un monde sans frontière*, Montréal, Canada, 18-20 octobre 1995, Volume II, pp. 2029-2039.
- [Lopez & al. 96], Lopez P., Haudot L., Esquirol P., Sicard M., Conception d'un système coopératif en ordonnancement de production : une approche pluridisciplinaire., *Conférence: 5th congrès International de Génie Industriel (GI5)*, Grenoble 2-4 avril 1996.
- [Mebarki & al. 96], Mebarki N, Pierreval H., Kouiss K., "Une approche multi-agents pour l'ordonnancement dynamique d'un système de production flexible", *Ingénierie des Systèmes d'Information*, Hermès, Vol.4, n°5, 1996, pp. 621-636.
- [Molina & al. 94], Molina A., Al-Ashaab, Elis T.I.A., Young R.I.M., Bell R., A Review of computer Aided Simultaneous Engineering Systems, in *Computer Aided Simultaneous Engineering Systems*, Manufacturing Engineering Department, Loughbrough University of Technology, England, 1994.
- [Parunak 96], Parunak H.Van D., Application of Distributed Artificial Intelligence in Industry, in *Artificial of Distributed Artificial Intelligence*, Chap. 4, Ed., pp.139-164, 1996.
- [Prosser 89], Prosser P., A reactive agent, in *International Joint Conference on Artificial Intelligence* , 1989, pp. 1004-1009.
- [Pujo & al. 97] Pujo P., Broissin N.et Bertrand J-C., Evaluation de génération de tâches dans le contexte du pilotage décentralisé d'un système de production automatisé et flexible, *Congrès International de Génie Industriel d'Albi : La productivité dans un monde sans frontière*, Albi, France, 3-5 septembre 1997.
- [Rolstadås 94], Rolstadås A., Beyond Year 2000 - Production Management in the Virtual Company, *IFIP WG5.7, Working Conference on Evaluation of Production Methods*, Gramado, Brazil, march 21-29, 1994.
- [Sassi & Chaib-Draa 97], Sassi M., Chaib-Draa B., "Planification et négociation dans les systèmes de transports considérés comme systèmes multi-agents", *JFSMA 97*, Sophia-Antipolis, avril 97, Hermès, pp.279-297.
- [Shaw 87], Shaw M.J., A distributed scheduling method for computer integrated manufacturing: the use of local area networks in cellular systems, *International Journal of Production Research*, Vol. 25, n°25, 1987, pp. 1285-1303.
- [Shoham 93], Shoham Y., Agent-Oriented Programing, *Artificial Intelligence*, n° 60, pp.51-93, 1993.
- [Skjeullaug & al. 90], B.Skjeullaug, H.Hämmäinen, H.Saathoff, Enterprises and Distributed CIM : Inter-organizational Communication, *Computers in Industry* , 9, 1990.
- [Slimani & Ramudhin 95], Slimani K., Ramudhin A., Applications des systèmes d'aide à la décision, des systèmes à base de connaissance et de l'intelligence artificielle aux problèmes d'ordonnancement: une revue de la littérature, *Congrès International de Génie Industriel de Montréal : La productivité dans un monde sans frontière*, Montréal, Canada, 18-20 octobre 1995, Volume II, pp. 405-414
- [Spinosa & al.95], Spinosa M.L., Espinasse B., Chouraqui E., Distributed CIM and DAI: for a Knowledge and Multiagent Approach, in *1995 IEEE International Conference on Systems, Man and Cybernetics*, oct. 22-25, 1995, Vancouver , Canada.
- [Sucur 95], Sucur M.D., Multiple objective decision-making in distributed problem solving framework for automated factory scheduling, *Preliminary Examination Proposal, Department of Mechanical and Industrial Engineering*, University of Illinois, Urbana, Illinois, 1995.
- [Sycara 89], Sycara K., 1989, Multiagent compromise via Negotiation, *Distributed Artificial Intelligence*, vol. 2, Morgan-Kaufmann Eds., pp 119-137.
- [Vernadat 94], Vernadat F., "Future R&D Directions for CIM Deployment", *European Workshop on Integrated Manufacturing Systems Engineering*, Grenoble, France, dec 12-14, 1994, pp. 3-6.