

E. Tranvouez, B. Espinasse (1999), « Protocoles de coopération pour le réordonnement d'atelier ». JFLADSMA 99, 7^e Journées francophones d'intelligence artificielle et systèmes multi-agents -, Hermès Ed., Saint-Gilles, La réunion, 8-10 nov. 1999.

Protocoles de coopération pour le réordonnement d'atelier

Erwan Tranvouez — Bernard Espinasse

DIAM-IUSPIM, Université Aix-Marseille, Domaine Scientifique de Saint Jérôme, 13397, Marseille, Cedex 20, France,

tel: ++33 4 91 05 60 30 - fax: ++33 4 91 05 60 33

E-mail : etranvouez@chez.com ; bernard.espinasse@iuspim.u-3mrs.fr

RESUME. Les systèmes multi-agents proposent aujourd'hui des solutions à de nombreux problèmes industriels. Notre recherche s'intéresse au réordonnement d'atelier confronté à des événements perturbateurs. Nous présentons une approche de la gestion de perturbation basée sur une résolution coopérative et distribuée de problème. Tout d'abord nous introduisons la problématique générale de notre recherche, pour ensuite exposer la modélisation multi-agents que nous avons adoptée. Nous décrivons ensuite le formalisme de représentation des comportements d'agents retenu puis son application aux protocoles de coopération pour la gestion des perturbations. Enfin, après avoir évoqué le prototype actuellement en cours de développement, nous concluons sur les limites et les perspectives de notre recherche.

ABSTRACT., Multiagent systems today offer solutions to a large number of industrial problems. Our research is centred on the workshop scheduling in a disrupted manufacturing context. This paper presents a disruption management approach based on a co-operative and distributed problem solving. First, we introduce the general problematic of our research and we develop the multiagent modelling adopted. We then describe the formalism we propose to represent the agents' behaviours and its use in the co-operative protocols developed by the agents in the disruption management. Finally, we conclude on the prototype under development, the limits and the perspectives of our research.

MOTS-CLES : systèmes multi-agents, résolution coopérative et distribuée de problèmes, ordonnancement d'atelier, gestion des perturbations, protocoles de rôles.

KEY WORDS: multiagent systems, co-operative and distributed problem solving, workshop scheduling, disruption management, role protocols.

1. Introduction

Dans la conduite de Systèmes de Production et face à une perturbation, le gestionnaire n'a souvent que deux options : lancer un calcul de réordonnement global coûteux en temps et source de modifications drastiques des plans de production initiaux, ou faute de meilleure solution disponible, accepter cette perturbation et les retards qu'elle peut engendrer.

Ces limitations proviennent en grande partie du manque de prise en compte de la nature intrinsèquement distribuée du problème d'ordonnancement. Dès lors, une approche multi-agents s'avère pertinente [PAR 98]. Néanmoins l'étude de ces approches ([BUR 94], [MEB 96],...) a révélé une sous-utilisation de processus coopératifs [ESP 98]. Notons quelques exceptions tel que [SAS 97] où le problème d'ordonnancement est perçu comme un problème d'allocation de ressources ou encore les travaux récents de [MIY 98] et [RAB 98] qui ont recours largement à des processus coopératifs.

En conséquence, nous nous focaliserons sur l'intégration de la coopération dans la gestion d'aléas de production. Pour cela, nous nous basons sur un modèle de système de production hiérarchique auquel nous rajoutons des techniques de résolution relativement simples [TRA 99]. Ces techniques inspirées des opérations de réparation d'ordonnancement (*repairs solutions* [ZWE 93]) sont utilisées ensuite dans des protocoles de coopération.

Nous présentons tout d'abord la problématique du réordonnancement d'atelier ainsi que la méthode de résolution que nous proposons. Puis nous développons la modélisation multi-agents qui en découle et certaines questions soulevées par cette modélisation. Nous décrivons ensuite en détail les protocoles de coopération dynamisant le processus de gestion de perturbation par coopération après avoir introduit notre formalisme de représentation. Nous terminons sur l'état des recherches en cours ainsi que certaines de leurs perspectives de recherche.

2. Réordonnancement coopératif d'atelier

L'activité purement productrice d'une entreprise manufacturière peut être modélisée selon trois niveaux décisionnels, à chacun desquels sont associées des responsabilités et des missions spécifiques :

- *Niveau 1*: responsable de la gestion globale de la production de l'entreprise. Il est chargé de la planification, i.e. de définir les produits et leur quantité à fabriquer à moyen terme. Équivalent au niveau hiérarchique "Responsable de Production".
- *Niveau 2*: responsable du bon fonctionnement de l'atelier (ou cellule) et du respect des directives transmises par le niveau supérieur. Il est chargé de produire un ordonnancement correspondant aux besoins spécifiés par ces directives et par la suite d'intervenir si ces objectifs sont en danger. Équivalent au poste "chef d'atelier".
- *Niveau 3*: responsable de l'exécution du plan de charge déterminé par le niveau supérieur. Son horizon est à très court terme. C'est souvent à ce niveau que seront perçus en premier les effets d'une perturbation. Équivalent au poste de "Technicien" assigné à une machine outil.

Nous dotons le troisième niveau de capacités décisionnelles lui donnant les moyens d'absorber une perturbation dès son apparition. Nous évitons ainsi les trop nombreux allers-retours d'information au niveau supérieur qui diminuent d'autant la réactivité de l'entreprise. Par perturbation nous entendons tout événement prévisible ou non, pouvant remettre en cause le plan de production initial. Suivant que les causes à l'origine de l'apparition d'une perturbation soient directement liées au dysfonctionnement d'une machine ou non, nous parlerons de perturbation interne ou externe. Une perturbation interne pourra par exemple être une panne sur une machine-outil conduisant à un retard, et une perturbation externe un problème d'approvisionnement de stock portant sur une pièce nécessaire à une machine pour terminer la fabrication d'un produit en cours de fabrication. Dans tous les cas, il sera nécessaire de trouver une solution permettant à l'atelier de subir au minimum les effets de cette perturbation.

D'une façon générale, le problème de (ré)ordonnement consiste à (re)programmer l'exécution d'un ensemble de tâches en leur allouant les ressources requises et en fixant leur date de début [GOT 93]. Dans le cas du problème d'atelier, les tâches correspondent à des opérations élémentaires réalisées sur des machines de production. Nous considérons le cas particulier d'un atelier de type *Job-Shop* dans lequel celui-ci est chargé de traiter des "jobs" (ou travaux) i.e. des séquences ordonnées de tâches élémentaires nécessaires à sa complétion (correspondant en général à la fabrication d'un produit).

Au niveau de cet atelier, le traitement des perturbations que nous proposons repose sur 6 groupes de stratégies de résolution basées sur des techniques de réordonnement par réparation ainsi que des techniques de Résolution Coopérative et Distribuée de Problèmes (RCDP) [TRA 99]. La nature distribuée et difficilement décomposable du problème d'ordonnement justifie le recours à une approche de RCDP [LES 90]. Dans l'état actuel de nos travaux, nous nous sommes limités à l'étude de la gestion d'une perturbation interne.

Le schéma ci-dessous (fig. 1) présente l'approche générale du traitement de la perturbation au niveau de l'atelier (ou cellule) :

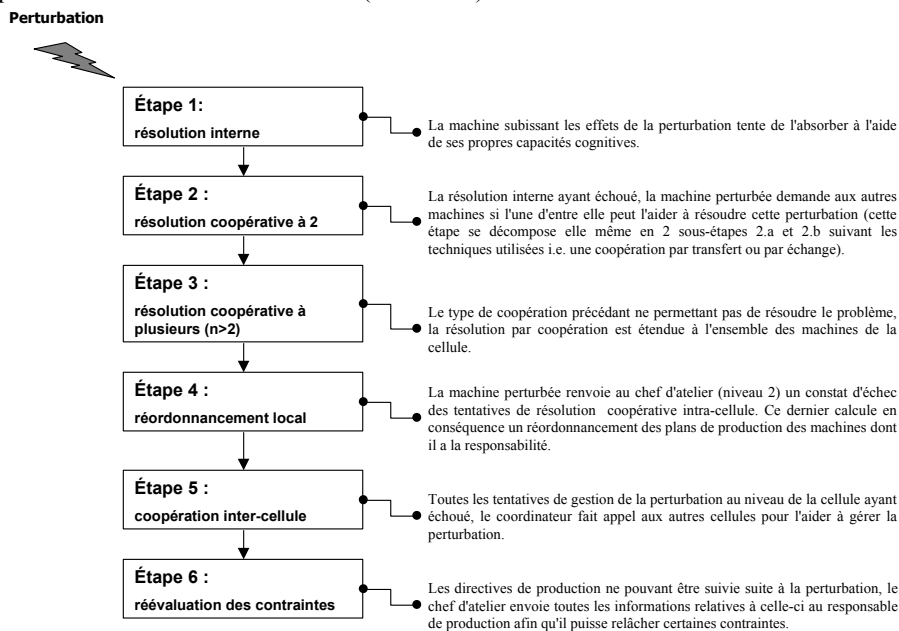


Figure 1. Traitement général d'une perturbation

La gestion de perturbations externes repose sur les mêmes principes que le traitement d'une perturbation interne, à l'existence près d'une étape supplémentaire. Cette étape précède les autres étapes et nécessite de la part du responsable de cellule (premier à être informé de cette perturbation) d'analyser les impacts de cette perturbation. Il en déduit les répercussions individuelles sur les machines, puis les en informe à leur tour du résultat de ces analyses. Chaque machine traitera alors cette information comme une perturbation interne.

3. Modélisation Multi-Agents d'un atelier de production

A partir de cette formulation du problème de gestion de perturbations dans un atelier, nous avons élaboré un modèle opératoire permettant de mettre en œuvre les principes développés dans cette formulation. Le problème à résoudre étant distribué et les méthodes de résolution se basant sur des processus coopératifs, le recours à une approche multi-agents s'est imposé. Dans cette section nous exposons brièvement les principales caractéristiques des agents ainsi que les plans comportementaux spécifiant le comportement des agents dans un contexte donné selon un formalisme de représentation adapté que nous introduisons. Le schéma ci-dessous décrit l'organisation multi-agents retenue.

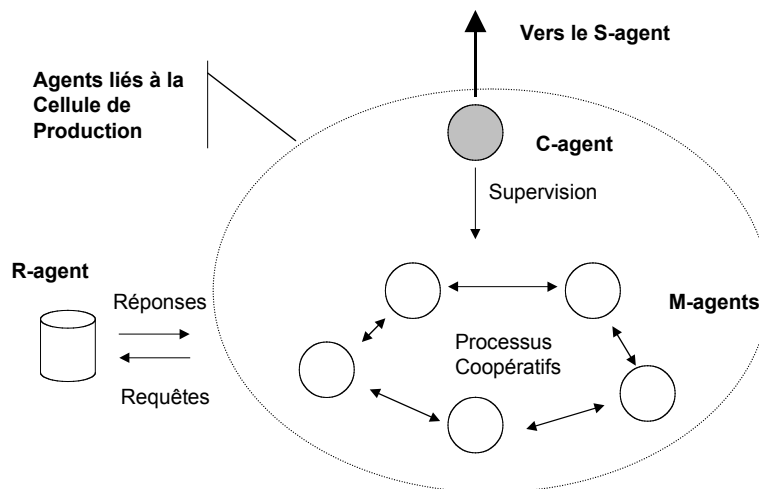


Figure 2. Organisation Multi-Agents

3.1. Caractéristiques des agents du SMA

Considérons un atelier de production composé de machines de production auquel on rajoute une capacité de réaction face aux perturbations grâce à un SMA. Afin de correspondre au système physique auquel il est relié, le SMA sera constitué de 4 types d'agent : un S-agent, un C-agent, des M-agents et un R-agent.

Le S-agent est associé au 1^{er} niveau décisionnel. Son niveau de responsabilité peut être assimilé à celui d'un Système d'Aide à la Décision Exécutif. Cet agent n'est pas à l'étude pour l'instant.

Le C-agent directement lié à l'atelier est associé au 2nd niveau décisionnel, et dispose des compétences requises pour assumer le rôle que lui confère son poste de responsable de la cellule. Il a aussi accès à toutes les informations détenues par les M-agents de "sa" cellule. Il peut intervenir et forcer un agent à accepter une modification de son plan de charge.

Le M-agent, sur lequel se base les étapes 1 à 3 de la gestion de perturbation, est associé à une machine-outil spécifique de l'atelier. Deux types de compétences le caractérisent : des compétences physiques et cognitives. Les compétences physiques se rapportent aux fonctionnalités de la machine dont il est le représentant dans le système. Ces compétences seront utilisées lors des calculs de réordonnancement. Les compétences cognitives décrivent toutes les aptitudes de coopération, de calcul de

solution, de communication etc. mises à disposition de l'agent pour qu'il puisse évoluer de façon autonome dans le SMA.

Le R-agent remplit le rôle d'agent répertoire ou encore d'agent annuaire [FIPA 97]. Sa conception est plutôt issue de préoccupations multi-agents, il n'est donc pas lié à la représentation du problème d'ordonnancement. Il centralise une partie des informations sur les compétences des agents. Ainsi, un R-agent est capable de répondre à des requêtes de type "Quels sont les agents possédant telle fonctionnalité". A chaque cellule on associe un R-agent dont les connaissances sur le système sont limitées à celle des agents de cette même cellule.

Dans ce qui suit nous utilisons indifféremment les termes M-agent et machine, ces deux mots décrivant tous deux le système (agent artificiel - machine de production). Les termes agent perturbé et agent coopérant apparaissent également dans la suite de cet article. L'agent perturbé est le M-agent qui subit les effets d'une perturbation, et qui cherche par conséquent la minimiser. L'agent coopérant est un M-agent contacté par l'agent perturbé pour lui prêter assistance, décidant de le faire.

3.2. Plans comportementaux pour le traitement des perturbations

Le développement de tout SMA amène à s'interroger sur la distinction entre les comportements d'un agent purement indépendant des actions des autres agents du système, de ceux faisant suite à des interactions avec ces mêmes agents [WOO 99]. Ces réflexions ont abouti à certains choix de représentation des comportements en distinguant d'une part les comportements *autarciques* ne nécessitant dans leur accomplissement aucun contact avec les autres agents, et d'autre part les comportements *sociaux* spécifiant à un moment ou à un autre un échange de communication ou toute autre "ouverture sur l'extérieur". Le modèle d'agent que nous avons défini reprends cette distinction en proposant un modèle individuel et un modèle social. La figure ci-dessous illustre ce propos en présentant la modélisation d'un M-agent (figure 3).

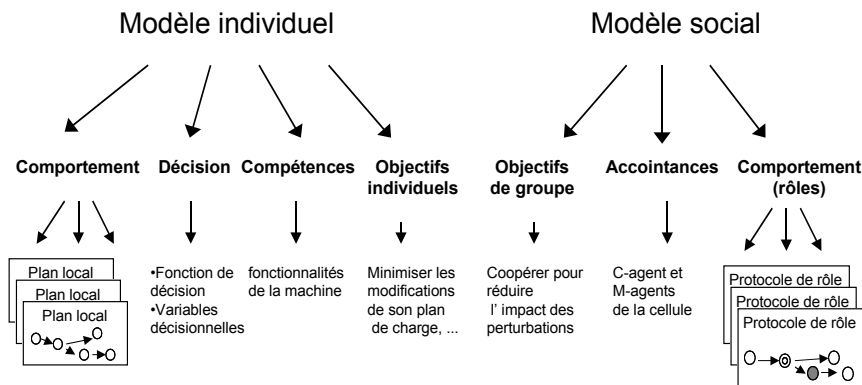


Figure 3. Modélisation du M-agent

Nous distinguons deux types de représentation de comportement d'agent selon le "modus operandi" choisi. Ces représentations (sous forme de graphe d'états) sont utilisées par l'agent pour résoudre les problèmes auxquels il pourra être confronté. Son comportement se basera alors sur :

- *un plan local* : s'il n'a pas besoin de demander d'information ni d'en envoyer (principales interactions possibles) aux autres agents. Tout changement d'état sera alors dû à des événements internes. L'ensemble des plans locaux est défini au sein du modèle individuel.

- *un protocole de rôle* : s'il doit interagir avec un autre agent. Ceci lui spécifiera sa part de travail i.e. son rôle, dans la réalisation de la coopération (informationnelle ou effective i.e. non plus limitées à des actes d'échange d'information) à laquelle il participe. A tout protocole de rôle correspondra au moins un protocole de rôle dual permettant de représenter tous les rôles coexistant lors d'une coopération. Les protocoles de rôles sont définis dans le modèle social.

Ces deux types de représentation seront décrits sous le terme générique de *plan comportemental*. Lors de son exécution, un agent a à sa disposition un ensemble de plans comportementaux (autarciques ou sociaux) pour résoudre les problèmes qu'il rencontre. Les agents ayant pour but de traiter une perturbation, disposent d'une série de plans comportementaux correspondants à autant de méthodes de gestion de perturbation (cf. fig. 1). L'agent dispose aussi de plans supplémentaires couvrant d'autres aspects de son fonctionnement, notamment des opérations de mise à jour de connaissances. Les protocoles de rôles de mise à jour des connaissances sont activés dès l'apparition de modifications dans l'ordonnancement d'une machine. Ainsi, un agent peut jouer plusieurs rôles (successivement ou simultanément) selon le contexte des conversations auxquelles il participe.

3.3. Formalisme de représentation de comportement d'agents autonomes

Dans un document de travail récent, la FIPA a souligné la nécessité de développer des formalismes de représentation des interactions entre agents d'un SMA. Au cours de la phase de modélisation des M-agents de notre système, nous avons recherché un formalisme adéquat pour représenter les comportements d'agents. Le langage COOL (*COOrdination Language*) [BAR 97] a paru à cet égard intéressant. Basé sur un mode de représentation par machines à états finis, il est centré sur la notion de conversation. Cependant certaines limitations nous ont conduit à faire un autre choix. Parmi ces limitations, la plus contraignante est l'impossibilité de concevoir plusieurs niveaux d'abstraction des protocoles de rôles, et ainsi de représenter des protocoles imbriqués. Par ailleurs, notons également le cadre de coordination CAT développé dans notre laboratoire [CLO 99] fondé sur les notions de contrats et d'accords liant des agents partageant un ensemble de conventions. Conçue pour être employée au sein d'architectures fortement distribuées aussi bien physiquement que structurellement (entreprise réseau), cette approche s'appuie néanmoins sur un formalisme qui dépasse largement nos propres besoins.

En conséquence, nous avons développé notre propre formalisme de représentation des comportements d'agents. Ce formalisme permet de décrire *a posteriori* le fonctionnement des agents mais également d'aider à la conception des interactions entre les agents du système. Le concepteur doit pouvoir entrevoir les liaisons qui existent entre le macro (i.e. comportement du système) et le micro (i.e. comportement de l'agent) de façon intégrée (de même que le préconise [WOO 99]).

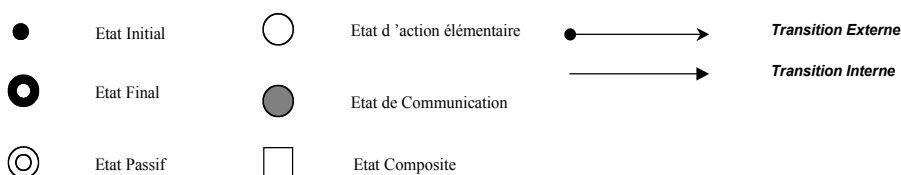


Figure 4. Convention de représentation utilisée

Le choix de représentation des plans comportementaux s'est porté sur des graphes d'états du fait de leur facilité relative de compréhension et d'utilisation face à d'autres représentations (Graphcnet, Réseaux de Pétri, ...) qui s'avèrent lourdes dès lors qu'il

faut représenter parallèlement le comportement de plusieurs agents. La Figure 4 au-dessous représente les différents types d'états et de transitions, ainsi que leurs symboles associés, utilisés dans notre formalisme de représentation.

Deux types d'événements peuvent apparaître pour dynamiser le graphe : des événements *internes* et des événements *externes* chacun activant des transitions de même nom. Les événements externes correspondent à l'aspect communication du SMA, ils sont liés à la réception de messages. Les événements internes correspondent aux activités propres à l'agent. Les transitions permettront de passer d'un état à un autre. Ces états pourront être *passifs*, il s'agira alors d'états d'attente ne pouvant être modifiés qu'à l'arrivée d'un événement externe (ex. arrivée d'un message), ou *actifs*, i.e. associé à l'exécution d'une action particulière. Dans ce dernier cas, nous distinguons trois type d'actions :

- *Etat d'action élémentaire* : exécute une action dite élémentaire, i.e. ne nécessitant pas de procédures complexes pour sa réalisation.
- *Etat d'action composite* : déclenche l'exécution d'un plan comportemental. Cet état permet de concevoir de manière modulaire les comportements d'agents.
- *Etat de communication*: envoie un message. Cet état particulier constitue avec les transitions externes, les points d'entrées et de sorties des flux d'informations existant entre les agents, sans lesquels aucune coordination ou coopération ne serait possible. L'envoi d'un message se traduira par un événement externe chez un autre agent et donc une transition externe.

La section suivante décrit certains plans comportementaux en reprenant ce formalisme.

4. Plans comportementaux de gestion de perturbation

Dans cette section, nous décrivons en détail les plans comportementaux permettant à un ensemble de machines d'absorber une perturbation. Ces plans sont présentés à l'aide du formalisme introduit dans la section précédente. Après avoir présenté le plan comportemental supervisant le traitement d'une perturbation, nous donnons deux exemples de ce traitement à travers deux types de résolution différentes. Le dernier plan comportemental explique comment la cohérence des connaissances des agents est maintenue.

4.1. Le plan local de gestion d'une perturbation interne

Ce plan pilote les autres plans et contrôle le bon déroulement du processus en cours. Il est activé dès la réception d'un message signalant l'apparition d'une perturbation. Sa représentation dans le formalisme précédent est donnée par la figure 5.

L'état de sélection de plan décide quelle méthode de résolution (i.e. quel plan comportemental) activer selon la situation rencontrée (i.e. si un plan précédent à échoué etc.). Deux exemples de plans comportementaux sont présentés dans les sections suivantes.

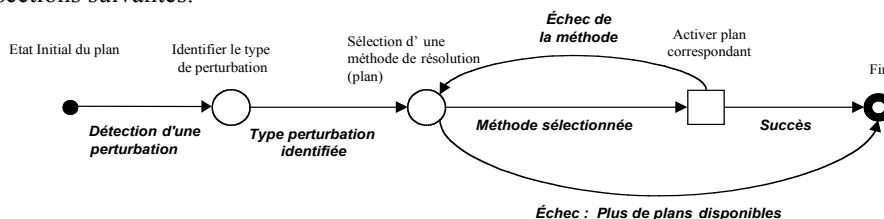


Figure 5. Le plan local de gestion d'une perturbation interne

4.2. Le plan local de résolution interne

Ce plan ne constitue pas une résolution de type coopérative puisque l'agent ne compte que sur ses propres ressources pour gérer une perturbation. Cependant, il interagit par la suite lorsqu'il doit informer les autres agents des changements intervenus en cas d'obtention d'une solution. Ceci constitue en conséquence une coopération informationnelle.

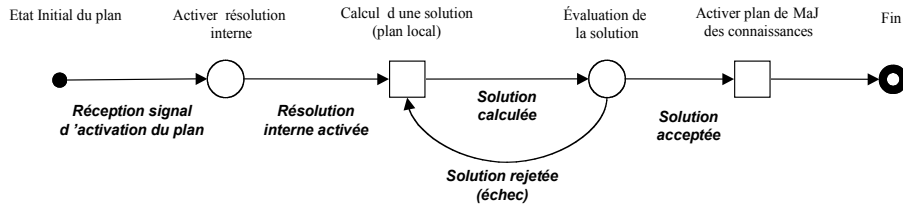


Figure 6. Le plan local de résolution interne

4.3. Résolution coopérative

Ce type de résolution met en jeu **1** agent perturbé et **p** agent(s) coopérant(s) ($p \geq 1$). L'agent perturbé est responsable du bon déroulement de la coopération. L'étude des comportements des agents en coopération s'étant révélée identique au mode de calcul prés, nous ne présentons ici que des protocoles de rôle neutres. En effet, les modes de résolution par transfert ou par échange ne se distinguent "que" par les types de messages envoyés (identifiant le type de coopération demandé et les solutions correspondantes) et par le mode de calcul employé (et par conséquent le plan local de calcul correspondant). Afin de d'aider au mieux à la conception et à la compréhension des comportements des agents, une représentation simplifiée "globale" est tout d'abord proposée.

4.3.1. Point de vue général de la résolution coopérative

La figure 8 ci-dessous décrit les grandes étapes par lesquelles passent les deux types d'agents perturbé et coopérant. Cette figure permet d'identifier les points d'entrée et de sortie des flux de communication entre les agents ainsi que les actions effectuées par ces mêmes agents pour traiter une perturbation. A partir de cette représentation, deux protocoles de rôles sont développés. Les paragraphes suivants détaillent ces protocoles de rôles.

4.3.2. Protocoles de rôles : point de vue de l'agent perturbé

L'agent perturbé, initiateur de la résolution coopérative, assure la gestion de la coopération correspondante. Aussi c'est à lui que reviendra le rôle d'envoyer l'appel d'offre aux agents adéquats et de sélectionner la solution la plus efficace ou du moins la moins perturbante. Ce rôle particulier de coordinateur apparaît clairement dans le graphe comportemental (cf. fig. 8).

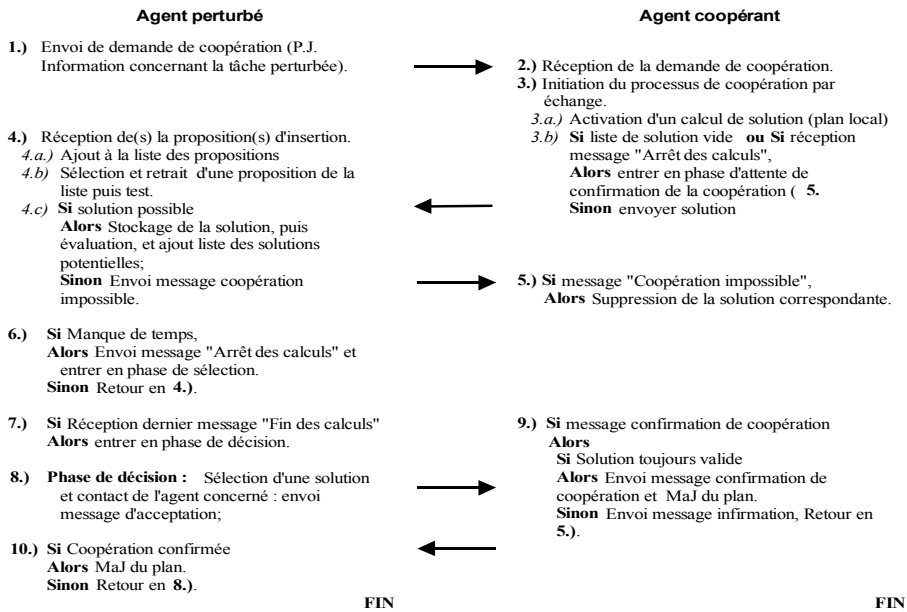


Figure 7. Vue générale d'une résolution coopérative

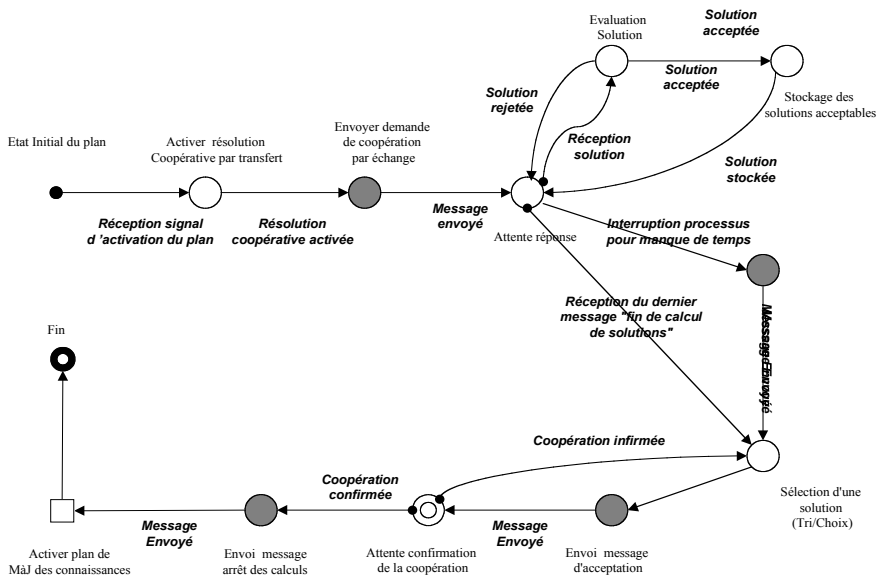


Figure 8. Protocole de rôle de résolution coopérative : point de vue de l'agent perturbé

L'agent perturbé se charge de sélectionner les destinataires des appels d'offre (sélection des contractants potentiels) en fonction de ses croyances sur les

compétences des agents du système. Il ne s'adressera qu'aux agents possédant la fonctionnalité nécessaire à l'accomplissement de la tâche à effectuer (les autres ne pouvant pas *physiquement* le faire). Cette sélection des agents s'effectue à l'aide du R-agent puisqu'il connaît l'ensemble des agents ainsi que leurs compétences. Enfin, dans un soucis d'équilibrage des charges de travail entre les agents, la charge d'informer les autres agents des changements des plans de charge est attribuée à l'agent perturbé.

4.3.3. Protocoles de rôles : point de vue de l'agent coopérant

L'agent coopérant est véritablement le moteur de la coopération. C'est lui qui alimente le processus de résolution coopérative en solutions potentielles permettant d'absorber la perturbation. Pour des raisons de lisibilité du graphe nous avons omis la transition interne partant de l'état d'attente vers l'état final intervenant lorsque suite à un changement de situation plus aucune des solutions envoyées précédemment ne sont valables. Dans ce cas l'agent se retire de la coopération. La figure 10, ci-dessous décrit le comportement d'un agent coopérant lors d'une résolution coopérative.

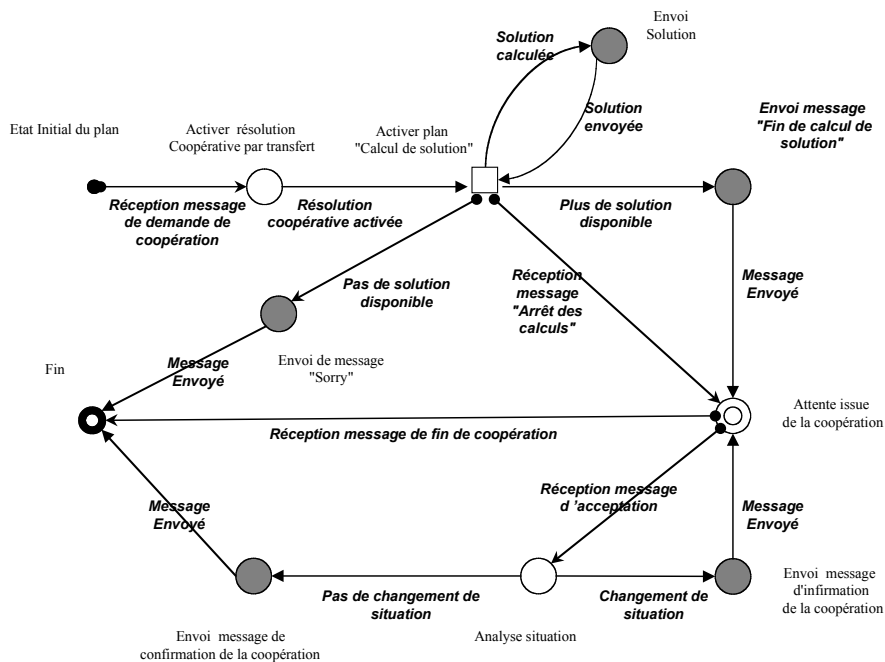


Figure 9. Protocole de rôle de résolution coopérative : point de vue de l'agent perturbé

4.3.4. Mécanisme de calcul de solution

Les plans locaux de calculs de solutions (dénotés par des carrés dans les représentations) s'appuient sur des opérations de réparation d'ordonnancement (traduction de *Repairs Solutions* [ZWE 93]) et des stratégies de résolution (coopérative ou non). Une opération de réparation peut être définie comme une construction itérative de solution de réordonnancement basée sur des modifications locales et limitées d'un ordonnancement. Les stratégies de résolution regroupent et définissent des cycles d'opérations de réparation. Pour exemple, une opération de réparation consiste à insérer, permuter des tâches sur un ou plusieurs plans de charges.

Pour prendre en compte les fortes interconnexions existant entre les données sur lesquelles portent ces calculs, nous avons définis des marges temporelles afin qu'un agent intègre dans ses connaissances les contraintes de ses voisins [Tranvouez et al 99]. Ces marges dépendent de l'état des plans de charge des M-agents, nécessitent régulièrement d'être mise à jour (cf. section suivante). Notons que les M-agents ne sont pas totalement altruistes : un M-agent acceptera de coopérer avec un autre jusqu'à un certain point. Le degré de coopération d'un agent correspondant à l'arbitrage entre objectifs locaux et objectifs sociaux (cf. fig 3), il est déterminé au sein d'une stratégie de résolution.

4.4. Coopération informationnelle : mise à jour des connaissances

Comme nous l'avons vu dans le § III.C, cette coopération sera entreprise dès lors que les connaissances, concernant le plan de charge d'une machine, se trouveront modifiées. Deux cas principaux sont possibles. Premièrement, une perturbation sur une tâche t d'un job j , détectée au niveau d'une machine M1 peut n'avoir aucune répercussion sur le plan de la machine M1, mais peut, par le jeu des contraintes d'antériorité sur le job j , en avoir sur une autre machine M2. Dans ce cas, M1 doit prévenir M2 pour que cette dernière puisse gérer, si nécessaire, les modifications conséquentes à cette perturbation. Deuxièmement, et c'est le cas le plus courant, un calcul de réordonnement amène un changement de la base de connaissance d'un agent. Cet agent doit alors prévenir les agents en amont et aval des tâches sur lesquelles des modifications de leur variables temporelles ont eu lieu (date de début, marges temporelles ...).

Les figures 10 et 11 décrivent les rôles respectifs d'un agent calculant et informant une mise à jour (agent émetteur) et un agent intégrant cette mise à jour dans ses croyances (agent récepteur).

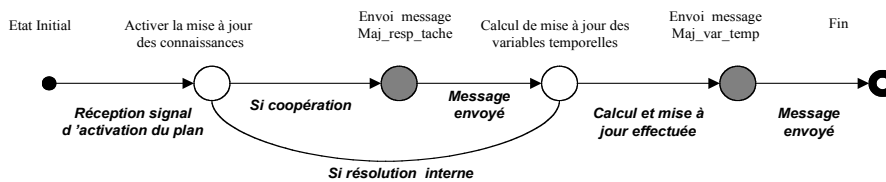


Figure 10. Mise à Jour des connaissances : rôle émetteur

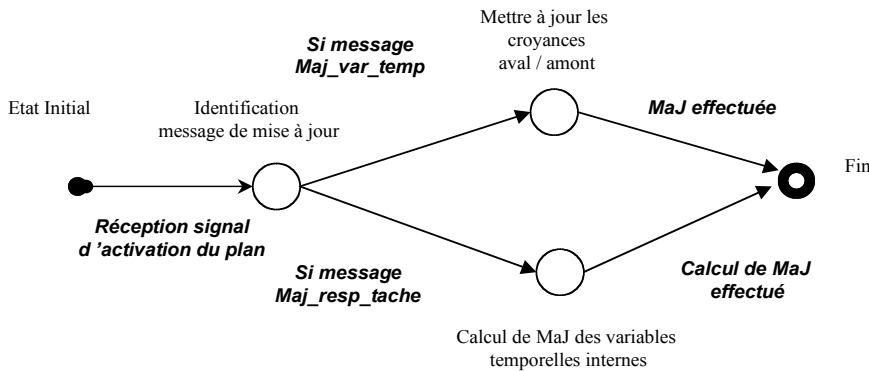


Figure 11. Mise à Jour des connaissances : rôle récepteur

Cette mise à jour des connaissances porte sur deux types de question toutes deux centrées tâches : le qui et le comment.

- *Le qui* : cette question est liée à la nécessité de savoir qui fait telle tâche (ex : la mise à jour des variables temporelles). En effet, la première conséquence d'une coopération, qu'elle soit par insertion ou par échange, demeure avant tout la modification du "lieu" de traitement d'une ou plusieurs tâches. Ainsi une tâche (2,3) traitée à l'origine par le M-agent M1, peut se voir transférée chez un autre M-agent M3. Le message de mise à jour correspondant sera intitulé : *Maj_resp_tache*.
- *Le comment* : il regroupe toutes les informations décrivant les caractéristiques de la tâche, à savoir les contraintes physiques et temporelles. Remarquons néanmoins que les problèmes de mise à jour concerneront principalement les contraintes temporelles. Ainsi, les messages traitant du *comment*, traiteront essentiellement de la mise à jour de variables temporelles telles que *d_min* ou *d_max*. Le message de mise à jour correspondant sera intitulé : *Maj_var_temp*.

Dans notre modèle nous considérons que c'est à l'agent responsable de la modification des informations sur une tâche que revient la charge d'informer les autres agents de cette modification. Remarquons que la mise à jour des variables temporelles chez l'agent récepteur ne débouche heureusement pas sur un nouvel envoi de messages de mise à jour, ce qui évite un risque d'explosion combinatoire.

5. Prototype et validation

Afin de valider notre approche nous avons développé, dans un premier temps, un prototype à partir de la plate-forme de développement de SMA LALO¹. Diverses difficultés liées aux options conceptuelles de cette plate-forme nous ont conduit à préférer le couple (Java / CLIPS). Le langage Java est un langage particulièrement adapté pour un fonctionnement en réseau et donc pour développer des agents artificiels. CLIPS est un système de production de règles en chaînage avant, permettant de développer des comportements intelligents par la mise en place de faits et de règles. Le développement s'appuie sur un portage Java de CLIPS (Jess²) associé à une bibliothèque de fonctions de communication développée dans notre laboratoire, basée sur le standard ACL [FIPA 97]. Les plans comportementaux décrits précédemment, sont traduits en règles Jess de deux types : des règles de transition et des règles d'action. Les règles de transition (interne et externe) définissent les conditions de transition entre états. Les règles d'action (correspondants aux états actifs et de communication) peuvent donner lieu à des appels de fonctions Jess ou Java (ex. envoi d'un message ACL).

Pour valider notre principe de réordonnement coopératif, nous avons élaboré un cas pratique inspiré de cas industriels réels, dans le contexte de l'atelier flexible présenté dans la figure 12. Cet atelier produit des pignons de vitesses à partir de pièces brutes selon un cycle de mesure, de tournage, de nettoyage et de vérification des produits finis. L'hétérogénéité des six machines composant cet atelier se traduit par une spécialisation des modes de coopération (fonctionnalités différentes, contraintes physiques, etc.). Les processus de coopératifs s'établissent entre les différentes machines le composant.

La définition du cas pratique a permis d'affiner la modélisation du problème d'ordonnement en précisant notamment les notions de gammes, de jobs ou encore d'opérations et de tâches. La validation de notre approche s'effectuera en vérifiant la capacité de cet atelier à absorber des perturbations.

¹ Cette plate-forme fut présentée en Session Poster lors des JFIAD-SMA'97.

² Disponible sur "<http://herzberg.ca.sandia.gov/jess>"

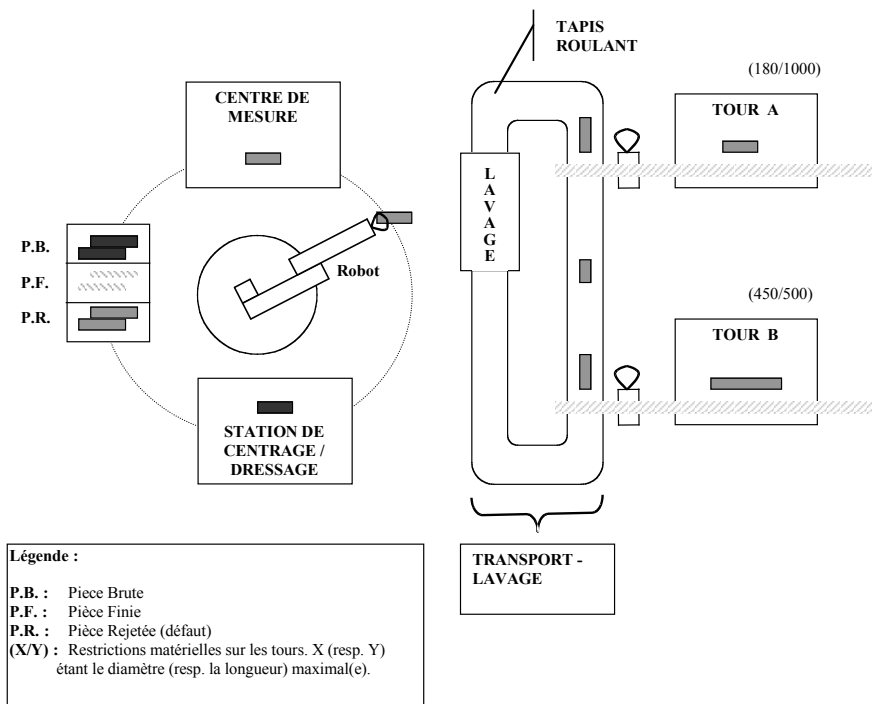


Figure 12. Atelier flexible de production.

6. Conclusion et perspectives

Dans cet article, nous avons présenté notre approche de gestion de perturbations dans le contexte d'un problème d'ordonnancement d'atelier. Nous avons tout d'abord rappelé les concepts et principes sur lesquels s'appuie notre démarche de réordonnancement coopératif. Nous avons proposé également un formalisme de représentation des comportements des agents facilitant la conception de ces mêmes agents. Nous avons appliqué ce formalisme au traitement d'une perturbation (avec ou sans coopération) en définissant les comportements des M-agents par des plans locaux et des protocoles de rôles.

Actuellement, les stratégies de résolution coopérative se limitent au niveau des machines. Nous envisageons d'étendre ces stratégies à des coopérations entre ateliers sur la base de principes similaires. Par ailleurs, si le SMA actuel n'intègre pas l'utilisateur humain, rien n'empêche d'ajouter un agent le représentant. L'ajout peut se limiter à un seul agent artificiel omnipotent (i.e. pouvant agir sur n'importe quel autre agent) ou à rajouter autant d'agents qu'il existe de M-agents. Ce travail permettrait de rajouter une dimension aussi bien didactique qu'opérationnelle en vue de son utilisation en entreprise. Enfin, il nous reste à confronter notre prototype avec le cas pratique que nous avons défini.

7. Bibliographie

[BUR 94] BURKE, P., PROSSER P., "The Distributed Asynchronous Scheduler", in *Intelligent Scheduling*, Monte Zweben & Mark S. Fox Eds, Morgan Kaufmann Publishers.

- [BAR 97] BARBUCEANU M. ET FOX M.S. "The design of a Coordination Language for Multi-Agents Systems", in *Intelligent Agents III : Agent Theories, Architectures and Languages*, J.P. Mueller, M.S. Wooldridge and N.R. Jennings (eds), LNAI 1193, Springer Verlag, 1997, pp 341-357.
- [CLO 99] CLOUTIER L. "Une approche multi-agents par conventions et contrats pour la coordination de l'entreprise manufacturière réseau", Thèse à paraître, DIAM, Université d'Aix-Marseille, Juin 1999.
- [ESP 98] ESPINASSE B., TRANVOUEZ E. "Ordonnancement d'atelier coopératif et réactif", *Journal of Decision Systems*, Vol. 7 – Special Issue, 1998, pp-215-237.
- [FIPA 97] FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS (FIPA), FIPA 07 Specification part 2 – Agent Communication Language, 1997 disponible à : <http://www.fipa.org>.
- [GOT 93] GOTHA, Les problèmes d'ordonnancement, *Recherche opérationnelle / Operations Research*, vol.27, n°1, 1993, p.77-150.
- [LES 90] LESSER V.R., An Overview of DAI: Viewing Distributed AI as Distributed Search, *Journal of Japanese Society for Artificial Intelligence*, Vol. 5, N°4, R. Nakano et S. Doshita (eds), pp. 392-400.
- [MEB 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.
- [PAR 98] PARUNAK H., VAN D., Application of Distributed Artificial Intelligence in Industry, in *Artificial of Distributed Artificial Intelligence*, Chap. 4, Ed., pp.139-164, 1996.
- [RAB 98] RABELO, R.J., CAMARINHA-MATOS L.M. "Generic Framework for Conflict Resolution in Negotiation-Based Agile Scheduling Systems", *IFAC- Workshop on Intelligent Manufacturing Systems / IMS'98*, Gramado, Brésil, 1998.
- [SAS 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.
- [TRA 99] TRANVOUEZ E., ESPINASSE B., FERRARINI A., "Résolution coopérative et distribuée de problèmes : Une application Multi-Agents au réordonnancement d'atelier", *CIGI'99*, Montréal, Canada, 26-28 Mai 1999.
- [WOO 99] WOOLDRIDGE M., JENNINGS N. R., KINNY D. "A Methodology for Agent-Oriented Analysis and Design", Proc. 3rd Int. Conf. On Autonomous Agents (Agents 99), Seattle, WA (à paraître), 1999.
- [ZWE 93] ZWEBEN M., DAVIS E. , DAUN B. , DEALE M. "Scheduling and Rescheduling with Iterative Repair", in *IEEE Transactions on Man and Cybernetics*, Vol 23., N°6, Nov-Déc 1993.