

UN MECANISME DE COOPERATION DANS UNE ORGANISATION HIERARCHIQUE

W. ALSHABI¹, M. ITMI¹

(1) LITIS Laboratory, INSA – Rouen, 76131 Mt. St.
Aignan Cedex, Rouen, France.
{waled.alshabi, mhamed.itmi, habib.abdulrab}
@insa-rouen.fr

S. RAMASWAMY², A. HABIB¹

(2) Computer Science Department, Univ. of Arkansas
at Little Rock, AR 72204 USA.
srini@acm.org

RESUME : *L'évolution des recherches actuelles dans les systèmes multi-agents (SMA) permet aujourd'hui l'utilisation de ce paradigme pour résoudre des problèmes dans des applications complexes. La coopération est souvent présentée comme l'un des concepts fondamentaux des SMA. Or, la plupart des stratégies de coopérations existantes sont ad-hoc dans la façon dont elles sont développées. L'objectif de cet article est de présenter un nouveau modèle hiérarchique qui permet une coopération flexible et efficace dans les SMA. Nous décrivons le fonctionnement de ce modèle et nous détaillons tout particulièrement le mécanisme de choix des agents. En fin d'article, un exemple illustre cette approche.*

MOTS-CLES : *Coopération, coordination, organisation, systèmes multi-agents.*

1. INTRODUCTION

Les systèmes multi-agents (SMA) proposent une approche originale de conception des systèmes intelligents et coopératifs. Ces derniers sont des systèmes distribués composés de plusieurs entités autonomes et flexibles appelées agents. Les agents interagissent, le plus souvent, selon des modes de coopération, de concurrence ou de coexistence (Chaib-draa 1994; Chaib-draa 1995; Moulin and Chaib-draa. 1996), en vue de réaliser conjointement une tâche ou d'atteindre conjointement un but particulier.

La coopération (S. Cammarata, D. McArthur et al. 1983; E. H. Durfee, V. R. Lesser et al. 1989) est une caractéristique très importante dans les systèmes multi-agents. La résolution distribuée d'un problème est le résultat de l'interaction coopérative entre les différents agents. Selon Ferber (Ferber 1995), « On dit que des agents coopèrent s'ils s'engagent dans une action commune après avoir identifié et adopté un but commun ».

L'implantation de la coopération entre les agents est une question fondamentale qui doit être traitée dans le cadre des systèmes multi-agents (Demazeau and Müller 1991). Dans (Durfee and Lesser 1989), les auteurs ont proposé quatre buts génériques pour établir la coopération dans un groupe d'agents:

- augmenter le taux de finalisation des tâches grâce au parallélisme;
- augmenter le nombre de tâches réalisables grâce au partage de ressources (information, expertise, dispositifs physiques, etc.);

- augmenter les chances de finaliser des tâches en les dupliquant et en utilisant éventuellement des modes de réalisation différents;
- diminuer les interférences entre tâches en évitant les interactions négatives.

Les stratégies de coopérations employées jusqu'à aujourd'hui sont pour la plupart soit ad-hoc dans la façon dont elles sont développées, soit difficiles à réadapter et à réutiliser dans d'autres applications. Dans ce papier, nous essayons de définir une structure simple, juste et efficace pour la coopération dans les SMA.

- Simple, car tout à fait intuitive dans son approche. Elle peut être facilement adaptée et appliquée dans des multiples domaines et avec différents types de contraintes.
- Juste (équitable) car tous les agents pouvant fournir les services nécessaires, obtiennent des opportunités égales pour la réalisation des différentes tâches.
- Efficace parce que l'approche est pragmatique dans la manière dont la coopération est conçue et orchestrée. Elle sépare les stratégies du travail (attentes économiques) de la conception de la solution. De plus, un mécanisme très efficace est fourni pour non seulement mettre en œuvre la coopération, mais aussi contrôler les points qui l'étouffent (grâce à *DAB*¹, que nous expliquerons par la suite).

La technique proposée dans ce papier s'applique à diminuer la politique de contrôle global. Elle permet, en même temps, un contrôle maximal concernant les décisions (d'offre) pour chaque agent. Cela soutient notre

¹ Degree of Agent's Believability.

volonté de disjoindre dans le modèle de coopération la hiérarchie d'exécution dynamique (en temps réel) de la hiérarchie organisationnelle (du point de vue métier, par exemple).

Ce papier est organisé de la manière suivante : la deuxième section donne une vue globale de notre modèle hiérarchique. La dynamique de ce modèle ainsi que les quatre phases qui le composent sont présentées dans la troisième section. La quatrième section explique en détail le mécanisme de choix permettant une coopération flexible et efficace entre de multiples agents. Un exemple explicatif simple illustre la démarche dans la cinquième section. Dans la sixième section, nous discutons le modèle développé. Nous concluons le papier dans la septième section.

2. UN MODELE HIERARCHIQUE

Bond (Bond 1990), décrit deux types d'organisations (architectures) pour les sociétés d'agents :

- Structure horizontale : Dans de telles sociétés, tous les agents sont au même niveau, il n'y a pas d'agents maîtres et d'agents esclaves. C'est le cas, par exemple, d'un groupe d'agents ayant des spécialités différentes qui travaillent pour la résolution d'un même problème.
- Structure verticale : Les agents sont ici structurés par niveaux. Dans un même niveau, nous retrouvons une structure horizontale.

Pour des applications nécessitant la coopération d'agents, l'augmentation du nombre d'agents fait que le système risque de voir ses performances s'effondrer à cause des nombreux liens entre les agents et la quantité d'informations échangées. Pour chaque action, un agent du système doit tenir compte des possibilités d'actions de tous les autres agents qu'il connaît. En cas de conflit, l'agent entre en négociation avec ses concurrents. Il est donc difficile de gérer plusieurs agents vu que la complexité du système augmente avec le nombre d'agents.

Face aux limites contraignantes du modèle multi-agents classique, le modèle multi-agents hiérarchique constitue une alternative en organisant les agents en structure verticale.

Nous avons développé un modèle hiérarchique (Al-shabi, Ramaswamy et al. 2006; Al-shabi, Itmi et al. 2007; Al-shabi, Ramaswamy et al. 2007) focalisant sur la coopération efficace entre les agents. L'orientation vers un modèle hiérarchique a pour objectif de surmonter les inconvénients qui apparaissent dans les modèles classiques des systèmes multi-agents (Bensaid and Mathieu 1995; Bensaid and Mathieu 1997; Briot and Demazeau 2001).

Dans notre modèle, les agents sont des entités autonomes ayant le contrôle de leurs ressources aussi bien que les

compétences qui leur permettent de coopérer, communiquer et travailler avec les autres agents. Chaque agent est capable de fournir des solutions spécifiques et de résoudre des problèmes locaux d'une manière autonome. Les agents peuvent fournir ou avoir besoin de l'aide d'autres agents pour accomplir une tâche particulière en raison du manque d'information, de ressource, etc. Ce besoin d'aide peut apparaître lorsqu'un agent ne possède pas les compétences nécessaires pour réaliser efficacement une tâche spécifique, ou bien lorsque l'agent préfère une solution coopérative.

La compétence d'un agent est une notion importante caractérisant les agents de notre modèle. Chaque agent peut avoir simultanément une ou plusieurs compétences spécifiques qui vont donc lui octroyer des rôles spécifiques. Plusieurs agents peuvent avoir la même compétence simultanément, d'où l'importance d'intégrer la notion de négociation parmi les différents agents pour choisir l'offre la mieux appropriée.

La notion de compétence est une notion qui évolue au rythme du travail. Autrement dit, les agents peuvent apprendre de nouvelles compétences comme ils peuvent également en perdre durant leurs interactions.

Les agents peuvent aussi être répartis dans plusieurs groupes. Chaque groupe est composé d'agents coopératifs, appelés agents professionnels. Un membre supérieur du groupe est appelé agent coordinateur. La section suivante décrit brièvement la fonctionnalité de notre modèle.

3. LE CPS-PROCESSUS DU MODELE PROPOSE

Dans cette section, nous donnons une vue d'ensemble de la fonctionnalité du modèle proposé ainsi que les quatre phases qui la composent. Ces différentes phases sont expliquées de manière plus détaillée dans un précédent article (Al-shabi, Ramaswamy et al. 2007). Les phases de notre CPSP² sont les suivantes :

3.1. La reconnaissance (Recognition):

Le processus commence lorsqu'un agent, ou un groupe d'agents, identifie le besoin d'une action coopérative. Ce besoin apparaît lorsqu'un agent doit accomplir une tâche sans en avoir la compétence, ou bien lorsque l'agent a une préférence pour une solution coopérative (Wooldridge and Jennings 1999).

Dans le premier cas, l'agent est incapable de réaliser seul la tâche demandée (quelle que soit la raison). Dans ce type de situation, seule la coopération permet d'accomplir l'action désirée. Ainsi, la coopération est vue comme une « obligation » et pas un choix, on parle alors de *coopération passive*. Dans le deuxième cas, l'agent préfère l'action coopérative alors qu'il est capa-

² Cooperative Problem Solving Process.

ble de réaliser la tâche sans la coopération des autres agents. Dans ce type de situation on parle de *coopération active*.

L'identification du besoin de l'action coopérative est la phase de reconnaissance.

3.2. La recherche des compétences (Skills' Search):

Dans cette phase, l'agent qui a identifié le besoin de l'action coopérative sollicite l'assistance et la coopération des autres agents. Cette sollicitation est réalisée par un processus spécial appelé le processus de recherche des compétences (*skills' search process - SSP*). Le *SSP* est expliqué en détail dans (Al-shabi, Ramaswamy et al. 2007).

3.3. Le choix d'Agents (Agents' Choice):

Dans ce papier, nous focalisons sur cet aspect particulier. Pendant cette phase, les agents (ou l'agent) qui possèdent les compétences nécessaires pour la réalisation de la tâche demandée, appelée T_d , seraient identifiés comme des agents compétents capables d'accomplir les besoins du client. Ces agents compétents seraient contactés par l'agent initiateur. Ce dernier va commencer un processus de négociation avec l'ensemble des agents compétents afin de choisir l'agent le mieux placé pour accomplir ou aider à accomplir la tâche T_d . Ce processus est souvent répétitif; puisque l'agent choisi peut répéter le même processus (devient également initiateur) indépendamment pour un ensemble de tâches (tâche décomposable). Cette phase nous permet de former la structure de coopération initiale (Changhong, Minqiang et al. 2002; d'Inverno, Luck et al. August 1997) appelée collectif.

3.4. Exécution (Execution):

Dans cette phase, les membres du collectif formé dans la phase précédente vont exécuter les rôles qui ont été négociés. À la fin de l'exécution, ces agents informent les initiateurs et envoient des informations concernant l'exécution de la tâche T_d (*feedback*). Ces informations peuvent être utilisées ultérieurement pour juger la qualité du service fourni dans l'accomplissement de la demande du client. Les initiateurs sont alors capables d'évaluer la performance des agents du collectif à exécuter le travail tel que négocié auparavant.

4. LE MECANISME DE CHOIX

Dans la section précédente nous avons mentionné les quatre phases qui constituent le *CPSP* de notre modèle. Dans cette section, nous nous intéresserons au mécanisme de choix.

L'objectif de ce processus est de permettre un choix approprié parmi les agents qui ont les compétences requises pour accomplir une tâche nécessaire (appelée aussi opération). Avant d'entrer dans les détails de ce proces-

sus; nous commençons par récapituler et clarifier le résultat du mécanisme de recherche discuté précédemment, et pouvant être l'une des possibilités suivantes:

- Aucun agent ne possède les compétences requises. Dans ce cas l'agent initiateur n'a pas de possibilité de choix en raison de la non-existence de candidats pour exécuter la tâche requise.
- Un seul agent possède les compétences requises. Dans ce cas l'agent initiateur va trouver un agent compétent unique (éventuellement lui-même) à la fin du processus de recherche *SSP*. Par conséquent, l'agent initiateur n'a pas de choix réel à faire.
- Plusieurs agents possèdent les compétences nécessaires. Dans ce cas l'agent initiateur trouve, à la fin du processus de recherche *SSP*, plusieurs agents compétents qui possèdent simultanément les compétences nécessaires. C'est la raison pour laquelle, l'agent initiateur a besoin de la négociation et d'un mécanisme de choix qui lui permet de choisir parmi les candidats disponibles.

Le besoin d'un mécanisme de choix apparaît lorsque nous avons plusieurs agents ayant les compétences nécessaires. Par conséquent, l'initiateur doit choisir parmi ces agents compétents. La sélection concerne ceux qui offrent une solution. Pour atteindre cet objectif notre mécanisme de choix appliquera la négociation pour obtenir la solution la plus appropriée.

Par exemple, lorsque l'agent initiateur cherche une compétence, tous les agents compétents (les agents qui possèdent la compétence recherchée) vont répondre à l'initiateur en envoyant leurs offres. Ces agents sont considérés comme des candidats potentiels. Par conséquent, leurs offres seront évaluées par l'initiateur selon plusieurs paramètres. Nous examinons cette situation.

Le choix d'une solution doit être effectué après l'évaluation des offres données par les agents (candidats potentiels). Cette évaluation est réalisée par une fonction spéciale définie à cette fin par l'utilisateur selon le domaine d'application. Par exemple, le choix peut concerner les agents qui offrent le meilleur prix ou ceux qui offrent le meilleur temps pour réaliser la tâche nécessaire.

Nous avons mentionné auparavant que lorsqu'un agent souhaite qu'une tâche soit exécutée (par un ou plusieurs agents), l'agent initiateur commence par effectuer une recherche pour trouver les agents compétents. Puis, l'initiateur va solliciter des propositions de ces agents en envoyant un appel à propositions (*call for proposals - cfp*) voir (FIPA00037 2002). Dans cet appel à propositions l'initiateur va spécifier la tâche T_d , ainsi que les conditions exigées pour son exécution. Les participants recevant l'appel à propositions sont considérés comme des candidats potentiels. Ils peuvent produire plusieurs réponses (en forme d'offres). Les agents peuvent également refuser de faire une proposition ou répondre négativement pour exécuter la tâche demandée. La proposi-

tion de chaque agent devrait contenir l'offre de l'agent et son Facteur de Succès (*Success Factor - SF*) (celui-ci sera détaillé dans les paragraphes suivants).

L'initiateur va évaluer les propositions reçues. Il choisit l'agent ayant la proposition la plus satisfaisante pour l'exécution de la tâche T_d . L'évaluation des différentes offres est réalisée en utilisant la formule (1) pour la réalisation d'une tâche spécifique T_d :

$$\left[CI_{\text{expected}} \times \left(\frac{FV}{SF_{A_i}} \right) \times Bid_{A_i}^{\text{evaluation}} \times DAB_{A_i} \right]_{A_i} \quad (1)$$

L'évaluation à l'aide de la formule (1) est répétée de manière systématique pour toutes les offres. Chaque paramètre dans (1) tient un rôle important dans le processus d'évaluation. Nous détaillons ces différents paramètres dans ce qui suit :

- CI_{expected} : représente l'indicateur de coopération attendu (*Cooperation Indicator expected*). Ce paramètre a un rôle fondamental. Il assure une flexibilité à notre modèle. Le CI_{expected} indique le niveau de coopération souhaité et attendu des agents. Il oriente l'organisation des agents vers la structure nécessaire pour modéliser à la fois la coopération passive et active selon le besoin de l'utilisateur (*C.f. section 3.1*). La valeur de CI_{expected} attendue est donnée à l'initialisation par l'utilisateur. Cette valeur dépend des besoins de l'application. La figure 1 montre la notion d'indicateur de coopération. C'est une sorte de curseur (position de l'un des triangles gris).

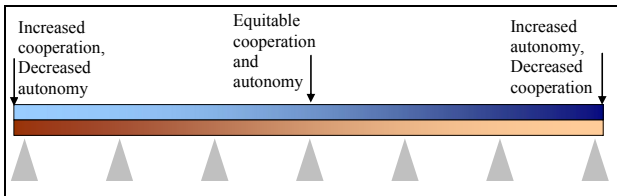


Fig. 1. L'indicateur de coopération

- FV : est utilisé pour représenter la valeur d'équité (*Fairness Value*) entre les agents. C'est un paramètre important dans notre mécanisme de choix. FV peut être considéré comme la valeur optimale que chaque agent doit atteindre dans une perspective idéale. Cette valeur précise le taux de tâches à réaliser par chaque agent. Cette valeur est utilisée pour améliorer le succès des différents agents pendant le processus de coopération. Le FV est également donné par l'utilisateur à l'initialisation.
- SF_{A_i} : on note SF pour représenter le Facteur de succès (*Success Factor*). Cette notion a été mentionnée brièvement précédemment. Nous avons indiqué que les agents ayant les compétences nécessaires pour la réalisation d'une tâche peuvent répondre à la sollicitation de l'agent initiateur en envoyant leurs offres ainsi que le SF de chaque agent A_i . Cette valeur de

SF définit la moyenne d'acceptation des offres pour chaque agent. Par exemple, si la valeur de SF concernant l'agent A_i est égale à 0.4, cela signifie que sur un total de 10 appels d'offres lancés il en a eu quatre seulement qui ont été couronnés de succès. dans le système global, le Facteur de Succès SF_{A_i} de chaque agent est calculé séparément. Cela n'est pas valable avec les deux premiers paramètres (CI et FV). Les valeurs de ces derniers sont prédéterminées pour le système entier. Par conséquent, chaque agent de notre modèle doit calculer les valeurs suivantes avec chaque événement transactionnel :

- le nombre total des offres demandées à l'agent. Nous le notons RB_{A_i} (*Requested Bids*). Il est non nul pour les agents concernés.
- le nombre total des offres couronnées de succès par l'agent. Nous l'appelons SB_{A_i} (*Successful Bids*).

Ces deux paramètres jouent un rôle capital pour calculer la valeur de SF pour chaque agent. À cette fin, nous divisons le nombre d'offres réussies par l'agent par le nombre d'offres demandées à cet agent; $SF_{A_i} = SB_{A_i} / RB_{A_i}$. Il est tel que $0 \leq SF_{A_i} \leq 1$.

Pour évaluer la valeur du SF_{A_i} résultant de l'équation précédente nous devons prendre en considération deux facteurs indispensables. Le premier est la valeur d'équité. Ce facteur a été expliqué dans le point précédent. Le deuxième facteur est la valeur de tolérance appelée (*Tolerance Value - TV*). La valeur de TV est définie pour le système dans son intégralité. Elle est donnée à l'initialisation. Dans le cas optimal, le facteur de succès concernant l'agent A_i devrait être égal à la valeur d'équité donnée, $SF_{A_i} = FV$. Dans un cas moins avantageux, le Facteur de Succès pour un agent A_i sera différent de la valeur d'équité. Néanmoins, le SF_{A_i} est situé dans une zone de tolérance définie par la valeur de tolérance, $TV_{\text{min}} \leq SF_{A_i} \leq TV_{\text{max}}$. Nous considérerons donc tous les agents ayant un SF se trouvant dans la zone de tolérance (définie par la TV) comme des agents ayant un bon degré de succès dans le système. Dans le plus mauvais cas, nous aurons des agents qui ont un SF se trouvant en dehors de la zone de tolérance. Deux situations sont possibles:

- $SF_{A_i} > TV_{\text{max}}$, dans cette situation le nombre des tâches réalisées par l'agent est supérieur aux valeurs de FV et de TV (données auparavant). Cela va pénaliser l'offre de l'agent par rapport aux autres agents (la situation suivante) et qui seront plus avantageux.
- $SF_{A_i} < TV_{\text{min}}$, dans cette situation le nombre des tâches réalisées par l'agent est inférieur aux valeurs de FV et de TV . Cela va

avantager l'offre de l'agent par rapport aux autres agents.

Ainsi, pour une tâche T_d les offres des agents seront évaluées en utilisant la formule (1). Le paramètre (FV/SF_{A_i}) favorisera les agents dans la situation (ii) et défavorisera les agents dans la situation (i). En conséquence, dans les deux situations la valeur de SF va tendre vers FV .

- $Bid_{A_i}^{evaluation}$: Comme son nom l'indique, ce paramètre représente l'évaluation d'offres de l'agent. Cette évaluation peut être effectuée par une fonction spéciale définie à cette fin par l'utilisateur selon le domaine de l'application. Cette fonction devrait prendre en considération les différents paramètres liés au domaine de l'application. Ces paramètres peuvent avoir ou ne pas avoir la même importance dans chaque application.
- DAB_{A_i} : La valeur de ce paramètre donne une idée du degré de crédibilité de l'agent (*Degree of Agent's Believability*). Lorsque les agents achèvent les tâches assignées, ils doivent informer l'initiateur et envoyer les informations concernant l'exécution des différentes tâches. Ces informations vont permettre à l'initiateur de commencer un processus d'évaluation dans lequel la performance de chaque agent va être évaluée. Cette évaluation vérifie si le résultat après l'exécution des tâches, est égal à l'offre proposée par l'agent avant l'exécution. L'évaluation de l'offre finale d'un agent A_i doit donc être inférieure (meilleure ici) ou égale à son offre de départ, $Bid_{A_i}^{final} \leq Bid_{A_i}^{initial}$. La valeur de DAB pour chaque agent est égale à un (valeur initiale par défaut), $DAB_{A_i}^0 = 1$: les agents sont supposés crédibles. Lorsqu'un agent réalise sa première tâche, la valeur de DAB est ensuite calculée en utilisant la formule suivante :

$$DAB_{A_i}^{new} = 1/2 \times (DAB_{A_i}^{last} + Bid_{A_i}^{initial} / Bid_{A_i}^{final}) \quad (2)$$

Avec, $DAB_{A_i}^{last}$ est la dernière valeur de DAB_{A_i} ,

$Bid_{A_i}^{initial}$ est l'offre initiale donnée par l'agent,

$Bid_{A_i}^{final}$ est l'évaluation de la performance de

l'agent après exécution. La valeur de DAB pour chaque agent sera enregistrée dans une matrice élaborée à cette fin. L'accès et la modification de cette matrice sont autorisés uniquement à l'agent initiateur jusqu'à la fin du processus.

Nous devons souligner que la formule (1) est utilisée pour évaluer les propositions des différents agents sauf lorsque l'initiateur possède la compétence demandée. Ce

dernier est autorisé à participer impartialement au processus de proposition. Dans ce cas, l'évaluation de la proposition de l'initiateur différera de l'évaluation des propositions des autres agents. Pour évaluer sa propre proposition, l'initiateur doit appliquer la formule suivante:

$$\left[(1 - CI_{expected}) \times \left(\frac{FV}{SF_{A_i}} \right) \times Bid_{A_i}^{evaluation} \times DAB_{A_i} \right]_{Initiator} \quad (3)$$

La distinction principale entre les deux formules (1) et (3) est dans la manière de calculer le premier facteur de la formule concernant l'indicateur de coopération. Dans la formule (1) la valeur de CI est égale à la valeur du CI attendu (des données initiales) tandis que, dans la formule (3) nous déduisons la valeur du CI attendu de 1. Cette opération favorise la coopération avec d'autres agents.

Le but d'avoir deux comportements différents, pour évaluer le premier facteur de la formule, est de permettre à notre modèle d'être flexible et d'éviter le problème de rigidité qui subsiste dans beaucoup d'autres modèles. Ainsi, dans des systèmes accordés pour offrir un degré de coopération plus élevé, nous donnons la priorité au meilleur candidat.

5. EXEMPLE

Pour illustrer l'idée proposée nous présentons un exemple simple. Considérons le scénario où trois agents A_1 , A_2 et A_3 ont une ou plusieurs compétences en même temps. Ces compétences permettent à l'agent de réaliser un certain nombre de tâches et par conséquent chaque agent tient un rôle spécifique dans le groupe. Supposons que l'agent A_1 possède les compétences S_1 , S_2 et S_3 , l'agent A_2 possède les compétences S_1 , S_4 et S_5 et l'agent A_3 possède les compétences S_1 , S_3 et S_6 (Figure 2). La nature et le

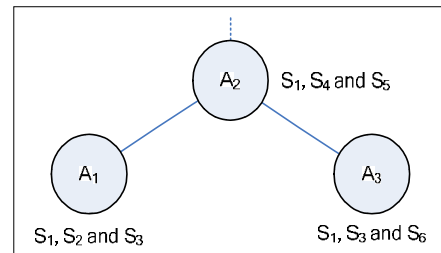


Fig. 2. Les compétences d'Agents

fonctionnement de ces compétences est ici sans intérêt. La hiérarchie organisationnelle proposée est aussi sans intérêt.

À l'initialisation, l'utilisateur devra fournir la valeur de paramètres essentiels qui sont l'indicateur de coopération attendu CI , la valeur de tolérance TV et la valeur de l'équité FV . Supposons que les valeurs données par l'utilisateur à l'initialisation sont égales à 80% pour le CI ,

20% pour la TV et 0.5 pour le FV . La valeur de CI donnée par l'utilisateur indique qu'il privilégie la coopération parmi les différents agents. Il préfère la solution coopérative dans 80% des cas. Le FV est égal à 0.5. Cela signifie que la tendance recherchée est que 50% des offres demandées doivent être réalisées par chaque agent dans ce groupe. En raison de l'existence de TV , le FV peut être étendu. Nous pouvons considérer que la valeur de SF_{A_i} est satisfaisante si elle vérifie $0.4 \leq SF_{A_i} \leq 0.6$.

Soit le scénario suivant : Nous supposons que l'agent A_2 doit réaliser une tâche T . La réalisation de T nécessite la possession de la compétence appropriée S_3 . L'agent A_2 est incapable de réaliser cette tâche car il n'a pas de compétence appropriée. Dans ce groupe, les deux autres agents (A_1 et A_3) possèdent cette compétence S_3 . Par conséquent, l'agent A_2 (l'initiateur) sollicitera les compétences nécessaires et appropriées auprès des autres agents. Cela s'effectue par l'envoi d'un message où l'initiateur demande aux autres agents d'envoyer des propositions (cfp), comme mentionné auparavant.

Après l'évaluation de leurs capacités et leurs disponibilités pour la réalisation de la tâche T , la réponse donnée par les agents A_1 et A_3 est en synthèse égale à 38 pour A_1 et 41 pour A_3 . Ces valeurs peuvent représenter le coût de l'opération, sa durée, etc. Cela dépend du problème et de son contexte.

Pour choisir parmi ces deux agents compétents, nous devons utiliser la formule (1). Dans cette la formule l'utilisateur a donné la valeur des deux paramètres à l'initialisation, $CI = 80$ et $FV = 0.5$. Il reste trois paramètres à calculer pour les agents A_1 et A_3 séparément. Ces paramètres sont : DAB , SF_{A_i} et $Bid_{A_i}^{evaluation}$.

Étant au début du processus de coopération, il n'y a pas de valeur calculée. Nous procédons comme suit :

- la valeur du DAB est égale à 1 tant pour A_1 que pour A_3 (autrement la valeur du DAB est calculée par la formule (2)).
- l'évaluation des offres données par les agents dépend de l'importance des différents paramètres (temps, coût, ... etc.). Cette importance varie selon le domaine de l'application. Dans notre exemple, les offres des agents représentent le coût d'exécution de la tâche pour chaque agent. Celui-ci est l'unique paramètre qui permet l'évaluation des offres. Ainsi, $Bid_{A_i}^{evaluation} = 1/A_i^{offer}$.
- Pour les différents agents: si la valeur SF de l'agent A_i est nulle alors, on « cherche à le favoriser légèrement » en posant : $SF_{A_i} = \max(SF_{A_i}, 0.1)$.

Dans le cas de notre exemple, pour l'agent A_2 , l'offre de l'agent A_1 est égale à :

$$\left[CI_{expected} \times \left(\frac{FV}{SF_{A_1}} \right) \times Bid_{A_1}^{evaluation} \times DAB_{A_1} \right]_{A_1} \\ = \left[0.8 \times \left(\frac{0.5}{0.1} \right) \times 1/38 \times 1 \right]_{A_1} = 0.105.$$

Et l'offre de l'agent A_3 est égale à :

$$\left[CI_{expected} \times \left(\frac{FV}{SF_{A_3}} \right) \times Bid_{A_3}^{evaluation} \times DAB_{A_3} \right]_{A_3} \\ \left[0.8 \times \left(\frac{0.5}{0.1} \right) \times 1/41 \times 1 \right]_{A_3} = 0.098.$$

Nous remarquons que l'agent A_2 va favoriser la coopération avec l'agent A_1 en raison de son résultat (supérieur au résultat de l'agent A_3). Par conséquent, pour les prochaines offres de l'agent A_1 , la valeur de SF_{A_2} et la valeur de DAB_{A_2} vont être différentes des valeurs données dans la première évaluation.

Une fois que l'agent A_1 a fini la réalisation de la tâche T_3 , il doit informer son initiateur (A_2). Ce dernier va réévaluer le degré de crédibilité de l'agent A_1 , DAB_{A_1} . Le but de cette réévaluation est de vérifier que l'agent a respecté son offre.

Si l'agent initiateur A_2 possède la compétence S_3 , son offre serait évaluée par la formule (3). Le premier paramètre serait égal à $(1 - CI_{expected}) = 0.2$. Dans ce cas, même si l'initiateur possède la compétence nécessaire il serait défavorisé car l'utilisateur favorise la coopération à 80%.

6. DISCUSSION

Le mécanisme présenté est un mécanisme flexible. Il est capable de traiter la coopération passive et active. Ce mécanisme permet à l'utilisateur de déterminer le pourcentage de coopération attendue. Cela peut être réalisé par le FV donné par l'utilisateur à l'initialisation.

Le mécanisme présenté permet également de définir une valeur de tolérance par l'utilisateur. La conjonction de la valeur d'équité FV et de la valeur de tolérance TV permet de définir une zone de tolérance qui est ici pour l'exemple:

$$[FV-TV/2, FV+TV/2] = [0.4, 0.6].$$

Tous les agents placés dans cette zone sont satisfaisants du point de vue de l'utilisateur.

7. CONCLUSION

Pour la résolution de problèmes coopératifs, nous avons présenté un modèle hiérarchique hybride : il tient compte de l'existence d'une hiérarchie organisationnelle et d'une hiérarchie dynamique pour résoudre un problème. Le modèle de cette dernière hiérarchie décrit tous les as-

pects du processus de coopération : de l'identification du potentiel de coopération à l'exécution. Un mécanisme de coopération a été développé pour préciser le niveau de coopération voulue et attendue des différents agents dans le SMA. Cela fournit la structure nécessaire pour modéliser la coopération, tant passive qu'active, selon les besoins de l'utilisateur.

REFERENCES

- Al-shabi, W., M. Itmi, et al. (2007). A hierarchical model for transport application 7ème édition du congrès international pluridisciplinaire, (QUALITA'07).
- Al-shabi, W., S. Ramaswamy, et al. (2006). Coordination, Cooperation and Conflict Resolution in Multi-Agent Systems. International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering, (IEEE CISSE'06): 6.
- Al-shabi, W., S. Ramaswamy, et al. (2007). A cooperative problem-solving process in hierarchical organization Summer Computer Simulation Conference, (SCSC'07): 6.
- Bensaid, N. E. and P. Mathieu (1995). Un modèle d'architecture multi-agents entièrement écrit en prolog. IV Journées Francophones de Programmation Logique, JFPL'95. Toulouse-France: 381–385.
- Bensaid, N. E. and P. Mathieu (1997). A Hybrid Architecture for Hierarchical Agents.
- Bond, A. H. (1990). Distributed Decision Making in Organizations. IEEE International Conference on Systems, Man and Cybernetics, University of California, Los Angeles, California 90024.
- Briot, J.-P. and Y. Demazeau (2001). Principes et architecture des systèmes multi-agents.
- Chaib-draa, B. (1994). Distributed Artificial Intelligence: An overview. In A. Ken, J. G. Williams, C. M. Hall, and R. Kent, editors, *Encyclopedia Of Computer Science And Technology*. I. Marcel Dekker. volume 31: 215-243.
- Chaib-draa, B. (1995). Industrial applications of distributed AI. *Communications of the ACM*.
- Changhong, L., L. Minqiang, et al. (2002). Cooperation Structure of Multi-agent and Algorithms. proceedings of the 2002 IEEE International Conference on Artificial Intelligence Systems (ICAIS'02).
- d'Inverno, M., M. Luck, et al. (August 1997). Cooperation Structures. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)* Nagoya, Japan.
- Demazeau, Y. and J.-P. Müller (1991). *Proceedings of the Second European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-90)*. Decentralized AI 2 - . Elsevier Science Publishers B.V. Amsterdam, The Netherlands.
- Durfee, E. H. and V. Lesser (1989). Negotiating task decomposition and allocation using partial global planning. *Distributed Artificial Intelligence*, Pitman Publishing: London and Morgan Kaufmann: San Mateo, CA.
- E. H. Durfee, V. R. Lesser, et al. (1989). *Cooperative Distributed Problem Solving*, volume 4.
- Ferber, J. (1995). *Les systèmes multi-agents, vers une intelligence collective*.
- FIPA00037 (2002). *FIPA Communicative Act Library Specification*. FIPA TC Communication.
- Moulin, B. and B. Chaib-draa. (1996). G. M. P. O'Hare and N. R. Jennings, editors, *Foundations of Distributed AI. An overview of distributed artificial intelligence*, Chichester, England.
- S. Cammarata, D. McArthur, et al. (1983). *Strategies of Cooperation in Distributed Problem Solving*. Proceedings of the 8th International Joint Conference on Artificial Intelligence, volume 2, Germany, Karlsruhe, editor.
- Wooldridge, M. and N. R. Jennings (1999). "The Cooperative Problem-Solving Process" *J. Logic Computat* **9**(4): 563-592.