

CHAPITRE

# Les dialogues entre agents avec des préférences conflictuelles

Leila Amgoud\* and Simon Parsons†

\* IRIT-UPS, 118 route de Narbonne 31062  
Toulouse Cedex, France.  
amgoud@irit.fr

† Department of Computer Science, University of Liverpool  
Liverpool L69 7ZF, United Kingdom.  
s.d.parsons@csc.liv.ac.uk

## Résumé

Un nombre croissant d'applications est conçu en utilisant la notion d'agent autonome. Les agents autonomes ont besoin de communiquer afin de résoudre leurs différences d'opinions ainsi que leurs conflits d'intérêts. Ils ont besoin de pouvoir s'engager dans des dialogues qui prennent en compte leurs différences de préférences.

Dans cet article nous proposons un modèle général de dialogue qui répond parfaitement aux attentes ci-dessus des agents. Le modèle est basé sur un système d'argumentation traitant des préférences conflictuelles. Il étend un précédent travail sur les dialogues basés sur l'argumentation où les agents sont supposés avoir les mêmes préférences.

**Mots-clés** : Dialogue, Argumentation, Préférences

## 1 INTRODUCTION

Plusieurs applications logicielles ont été conçues et implémentées en utilisant la notion d'agent autonome. Ces applications varient du commerce électronique à de larges applications industrielles. Dans tous ces différents cas, la notion d'autonomie est employée pour dénoter le fait que le logiciel a la capacité de décider par lui-même quels buts adopter et comment ces buts devraient être réalisés.

Dans la plupart des applications d'agent, les composantes autonomes ont besoin d'interagir à cause des liens existant entre elles. Elles ont besoin de communiquer afin de résoudre les différences d'opinions et les conflits d'intérêts, de travailler ensemble afin de résoudre des dilemmes, trouver des preuves, ou tout simplement pour échanger des informations.

Ces exigences de communication ne peuvent pas être réalisées par l'échange d'un simple message. Les agents concernés ont besoin d'échanger des séquences de messages, en d'autres termes ils ont besoin de dialoguer.

Plusieurs travaux ont été faits à cet effet. Cependant, ils s'intéressent uniquement à un seul type de dialogue : soit la négociation [9, 13, 14], où les agents essaient de trouver un compromis quant à l'utilisation des ressources, soit la persuasion [5, 6, 7, 8, 11, 16], où un agent essaye de changer le point de vue d'un autre agent.

Dans un travail antérieur [2, 3], nous avons présenté un système général qui capture différents types de dialogue. Comme tous les autres travaux cités ci-dessus, le système est basé sur l'utilisation de l'argumentation et il modélise le dialogue entre deux agents uniquement.

Une limite importante de tous ces travaux est le fait de ne pas prendre en compte les préférences des agents ([7, 8, 11, 16]), ou bien comme dans [2, 3], les préférences des agents sont considérées mais avec l'hypothèse simplificatrice et très restrictive que les agents ont les mêmes préférences.

Une autre limite de la plupart des travaux sur le dialogue basé sur l'argumentation est le fait qu'ils considèrent uniquement la façon dont les arguments sont traités (i.e quel argument contrarie d'autres arguments). Ils ne s'intéressent pas à la façon dont ces arguments sont utilisés dans un dialogue et aussi à la façon dont ils sont construits au cours d'un dialogue.

Dans cet article nous proposons une extension du système de dialogue proposé dans [2, 3]. Il s'agit d'un système général de dialogue qui permet à *plusieurs agents* de s'engager dans des discussions en prenant en compte leurs différences de préférences. Cette partie de l'article utilise un système d'argumentation basé sur des préférences contextuelles développé dans [4].

## **2 UN SYSTÈME D'ARGUMENTATION BASÉ SUR DES PRÉFÉRENCES CONFLICTUELLES**

L'argumentation est une approche de raisonnement avec des informations inconsistantes, basée sur la construction d'arguments et de contre arguments,

la comparaison de ces arguments et enfin la sélection d'arguments jugés acceptables.

Dans les travaux précédents [2, 3], nous avons décrit en détail un système formel d'argumentation et avons montré comment il peut être employé pour soutenir des dialogues entre des agents sous l'hypothèse que tous les agents ont le même ensemble de préférences.

Dans ce papier nous étendons ce système pour tenir compte de différents ensembles de préférences et cela en adoptant l'approche que nous avons suggérée dans [4].

L'idée est d'associer à une même base de connaissances plusieurs préordres. Chaque préordre représente des préférences exprimées dans un contexte particulier d'où le nom de *préférences contextuelles*.

$\Sigma$ , une base de connaissances inconsistante et non déductivement close, contient des formules d'un langage propositionnel  $\mathcal{L}$ .  $\vdash$  dénote l'inférence classique et  $\equiv$  dénote l'équivalence logique.

Les préférences sur les éléments de  $\Sigma$  dépendent du contexte dans lequel elles sont exprimées.  $\mathcal{C} = \{c_1, \dots, c_n\}$ , l'ensemble de tous les contextes, est supposé muni d'un ordre  $\sqsupset$ . Ainsi pour deux contextes  $c_1, c_2 \in \mathcal{C}$ ,  $c_1 \sqsupset c_2$  veut dire qu'une proposition dans le contexte  $c_1$  est préférée à une autre proposition dans le contexte  $c_2$ . Chaque contexte  $c_i$  est muni d'un préordre  $\gg_i$  qui donne les préférences sur les éléments de  $\Sigma$ .

### Définition 1

Un argument est une paire  $A = (H, h)$  où  $h$  est une formule de  $\mathcal{L}$  et  $H$  un sous-ensemble de  $\Sigma$  tel que :

1.  $H$  est consistant;
2.  $H \vdash h$ ;
3.  $H$  est minimal, il n'existe pas un sous-ensemble de  $H$  qui satisfait 1 et 2.

$H$  est appelé le support de l'argument, dénoté par  $H = \text{Support}(A)$  et  $h$  sa conclusion, dénotée par  $h = \text{Conclusion}(A)$ .

Puisque la base  $\Sigma$  est inconsistante,  $\mathcal{A}(\Sigma)$ , l'ensemble de tous les arguments construits à partir de  $\Sigma$ , contient des arguments qui attaquent d'autres arguments.

### Définition 2

Soient  $A_1$  et  $A_2$  deux arguments de  $\mathcal{A}(\Sigma)$ .  $A_1$  attaque  $A_2$  ssi  $\exists h \in \text{Support}(A_2)$

tel que  $h \equiv \neg \text{Conclusion}(A_1)$ .

En d'autres termes, un argument est attaqué s'il existe un argument pour la négation d'un élément de son support.

Chaque préordre  $\gg_i$  défini sur la base  $\Sigma$  peut être utilisé pour définir un préordre sur l'ensemble des arguments  $\mathcal{A}(\Sigma)$ . On peut ainsi définir une relation de préférence  $\text{Pref}_i$ , dans un contexte  $c_i$ , basée sur le préordre  $\gg_i$ .

Dans [1] plusieurs relations de préférences entre arguments ont été proposées. Certaines d'entre elles supposent que  $\gg_i$  est un préordre partiel et d'autres le supposent total. Dans ce qui suit, nous présentons un exemple d'une relation de préférence qui utilise un préordre total.

Lorsque le préordre est total, il est équivalent de considérer la base comme étant stratifiée en plusieurs sous-bases  $\Sigma_1, \dots, \Sigma_n$ . Les éléments de  $\Sigma_i$  ont tous la même préférence et ils sont préférés à tout élément d'une sous-base  $\Sigma_j$  avec  $j > i$ .

Le *niveau de préférence* d'un sous-ensemble non vide  $H$  de  $\Sigma$ ,  $\text{niveau}(H)$ , est le numéro de la plus basse state rencontrée par  $H$ .

### Définition 3

Soient  $A_1$  et  $A_2$  deux arguments de  $\mathcal{A}(\Sigma)$ .  $A_1$  est préféré à  $A_2$  selon  $\text{Pref}_i$  ssi  $\text{niveau}(\text{Support}(A_1)) \leq \text{niveau}(\text{Support}(A_2))$ .

### Exemple 1

Soit  $\Sigma = \{\neg a, a, a \rightarrow b, \neg b\}$ . Supposons que dans un contexte  $c_i$ , nous avons la stratification suivante:  $\Sigma = \Sigma_1 \cup \Sigma_2 \cup \Sigma_3$  avec  $\Sigma_1 = \{\neg a\}$ ,  $\Sigma_2 = \{a, a \rightarrow b\}$  et  $\Sigma_3 = \{\neg b\}$ .  $(\{\neg a\}, \neg a)$  et  $(\{a, a \rightarrow b\}, b)$  sont deux arguments de  $\mathcal{A}(\Sigma)$ . L'argument  $(\{\neg a\}, \neg a)$  attaque  $(\{a, a \rightarrow b\}, b)$ . Le niveau de préférence de  $\{a, a \rightarrow b\}$  est 2 tandis que celui de  $\{\neg a\}$  est 1, donc  $(\{\neg a\}, \neg a)$  est préféré selon  $\text{Pref}_i$  à  $(\{a, a \rightarrow b\}, b)$ .

$\text{Pref}_1, \dots, \text{Pref}_n$  dénotent les différentes relations de préférence entre les arguments induites respectivement à partir des préordres  $\gg_1, \dots, \gg_n$ .

Notons que puisque les préordres sur  $\Sigma$  sont conflictuels, les préordres sur  $\mathcal{A}(\Sigma)$  peuvent aussi être conflictuels. Ainsi, pour deux arguments  $A_1$  et  $A_2$ ,  $A_1$  peut être préféré à  $A_2$  dans un contexte  $c_i$  et  $A_2$  peut être préféré à  $A_1$  dans un autre contexte  $c_j$  tel que  $i \neq j$ . Nous pouvons maintenant définir formellement le système d'argumentation que nous utiliserons dans la suite.

### Définition 4

Un système d'argumentation basé sur des préférences contextuelles (SAPC) est un tuple  $\langle \mathcal{A}(\Sigma), \text{Attaque}, \mathcal{C}, \sqsupset, \gg_1, \dots, \gg_n \rangle$  tel que :

- $\mathcal{A}(\Sigma)$  est l'ensemble des arguments;

- *Attaque* est une relation de contrariété entre les arguments;
- $\mathcal{C}$  est l'ensemble des contextes;
- $\sqsupset$  est un préordre (partiel ou total) sur  $\mathcal{C} \times \mathcal{C}$ ;
- $\gg_i$  est un préordre (partiel ou total) sur  $\Sigma \times \Sigma$  issu du contexte  $c_i$ .

Les différents préordres  $Pref_1, \dots, Pref_n$  permettent de distinguer différents types de relations entre les arguments et cela en fonction de la façon dont les arguments s'attaquent mutuellement. Un argument *se défend seul* contre un attaquant s'il est préféré à ce dernier (dans un contexte préféré). Un ensemble d'arguments peut *défendre* un argument en attaquant tous les arguments contre lesquels le seul argument ne peut pas se défendre seul.

### Définition 5

Soit  $A_1, A_2$  deux arguments de  $\mathcal{A}(\Sigma)$ .

- Si  $A_2$  attaque  $A_1$  alors  $A_1$  se défend seul contre  $A_2$  ssi  $\exists c_i \in \mathcal{C}$  tel que :
  1.  $A_1 Pref_i A_2$
  2.  $\forall c_j$  tel que  $A_2 Pref_j A_1$  alors  $c_i \sqsupset c_j$ .
- Un ensemble d'arguments  $\mathcal{S}$  défend  $A$  ssi  $\forall B$  attaque  $A$  et  $A$  ne se défend pas seul contre  $B$  alors  $\exists C \in \mathcal{S}$  tel que  $C$  attaque  $B$  et  $B$  ne se défend pas seul contre  $C$ .

$C_{Attaque, \sqsupset}$  contient les arguments non attaqués et ceux qui se défendent seuls contre leurs attaquants.

Dans [4], nous avons montré que l'ensemble  $\underline{\mathcal{S}}$  de tous les arguments acceptables du système d'argumentation  $\langle \mathcal{A}(\Sigma), Attaque, \mathcal{C}, \sqsupset, \gg_1, \dots, \gg_n \rangle$  est le plus petit point fixe de la fonction  $\mathcal{F}$  définie comme suit :

$$\mathcal{S} \subseteq \mathcal{A}(\Sigma), \mathcal{F}(\mathcal{S}) = \{A \in \mathcal{A}(\Sigma) \mid A \text{ est défendu par } \mathcal{S}\}$$

### Définition 6

L'ensemble des arguments acceptables du système d'argumentation  $\langle \mathcal{A}(\Sigma), Attaque, \mathcal{C}, \sqsupset, \gg_1, \dots, \gg_n \rangle$  est :

$$\underline{\mathcal{S}} = \bigcup \mathcal{F}_{i \geq 0}(\emptyset) = C_{Attaque, \sqsupset} \cup [\bigcup \mathcal{F}_{i \geq 1}(C_{Attaque, \sqsupset})]$$

Un argument est acceptable si et seulement s'il est membre de l'ensemble  $\underline{S}$ .

### Exemple 2

(Suite de l'exemple 1) Supposons que  $\mathcal{C} = \{c_1, c_2\}$  tel que  $c_2 \sqsupset c_1$ .  $c_1$  utilise la stratification donnée dans l'exemple 1 et  $c_2$  utilise la stratification suivante de  $\Sigma$ :  $\Sigma = \Sigma_1 \cup \Sigma_2 \cup \Sigma_3$  avec  $\Sigma_1 = \{a, a \rightarrow b\}$ ,  $\Sigma_2 = \{\neg a\}$  et  $\Sigma_3 = \{\neg b\}$ . L'argument  $(\{a, a \rightarrow b\}, b)$  est dans  $\mathcal{C}_{Attaque, \sqsupset}$  car il se défend seul contre ses attaquants  $(\{\neg a\}, \neg a)$  et  $(\{a, \neg b\}, \neg(a \rightarrow b))$ .  $(\{a, a \rightarrow b\}, b)$  est préféré à ses deux attaquants dans le contexte privilégié  $c_2$ . Par conséquent,  $(\{a, a \rightarrow b\}, b)$  est dans l'ensemble  $\underline{S}$ .

## 3 UN SYSTÈME DE DIALOGUE

Cette section présente la manière dont les dialogues inter-agents sont capturés en utilisant le système d'argumentation présenté ci dessus. Nous commençons par présenter un système abstrait de dialogue en discutant ses éléments de base à savoir : un ensemble d'agents, une relation de préférence entre les agents, un protocole et un ensemble de stratégies.

### 3.1 Définition du système général

Un des éléments de base d'un système de dialogue est l'ensemble des agents impliqués dans ce dialogue. Les agents peuvent être des humains ou des systèmes informatiques.

Les travaux sur la modélisation du dialogue [3, 7, 8, 11] considèrent le dialogue uniquement entre deux agents. Cependant, il existe d'autres formes de dialogue où plusieurs agents sont invoqués. Le meilleur exemple est le dialogue juridique où au moins trois parties sont invoquées (le juge, l'avocat et le procureur).

Chaque agent a un *nom* et un *rôle* dans le dialogue. Le rôle d'un agent peut aider à déterminer le poids des coups avancés par cet agent au cours du dialogue. Un agent est aussi supposé avoir une base de connaissances qui contient ses croyances, ses désirs ainsi que ses intentions. Elle peut aussi contenir certaines connaissances sur les croyances des autres agents. Nous supposons que la base de connaissance est équipée d'un *préordre (partiel ou total)* représentant les préférences de l'agent.

Comme suggéré par MacKenzie [7], chaque agent possède une autre base, accessible à tous les agents, contenant ses engagements au cours du dialogue. Cette base est appelée *tableau de conversation (CS)*.

Une base de connaissance est écrite dans une *logique*, et les agents peuvent utiliser différentes logiques. La logique gouverne le dialogue de différentes façons. Par exemple, si on demande à un agent de justifier sa proposition, cette dernière doit être déduite logiquement de la justification, ou bien si une proposition faite par un agent X n'est pas déduite logiquement de la base de connaissances d'un agent Y alors ce dernier peut refuser la proposition (si on suppose que l'agent Y est prudent). La logique peut aussi aider un agent à maintenir la cohérence de sa base.

**Définition 7**

Un agent est un tuple  $\langle A_i, R\acute{o}le_{A_i}, \mathcal{L}_{A_i}, \Sigma_{A_i}, \gg_{A_i}, \mathcal{CS}_{A_i} \rangle$  avec:

1.  $A_i$  est le nom de l'agent
2.  $R\acute{o}le_{A_i}$  dénote le rôle de l'agent  $A_i$
3.  $\mathcal{L}_{A_i}$  dénote la logique utilisée par l'agent  $A_i$
4.  $\Sigma_{A_i}$  est la base de connaissances de l'agent  $A_i$
5.  $\gg_{A_i}$  est un préordre (total ou partiel) sur  $\Sigma_{A_i}$
6.  $\mathcal{CS}_{A_i}$  est le tableau de conversation de l'agent  $A_i$

L'ensemble de tous les agents sera dénoté par  $\mathcal{A} = \bigcup_i \{ A_i \}$ .

Le rôle d'un agent a une influence sur le choix de la logique à adopter. Dans les dialogues où des agents échangent des informations et à la fin un juge détermine le gagnant, une logique classique est suffisante pour les différents agents. Par contre le juge requiert une logique non classique plus sophistiquée pour pouvoir traiter les conflits entre les agents.

Dans [12], les auteurs mentionnent que dans une entreprise, par exemple, les décisions respectent le niveau hiérarchique de l'agent qui les a exprimées. Par exemple, les décisions du directeur général sont prioritaires à celles du directeur de Marketing. Pour capturer cette idée de *poids* d'un agent, nous supposons que l'ensemble des agents est équipé d'un préordre  $\succ$  (partiel or total).

Comment définit t-on un dialogue? Un dialogue est une séquence non vide de coups avancés par les agents. Formellement :

**Définition 8**

Un coup est un tuple:  $M = \langle S, H, Acte \rangle$ .

- $S$  est l'agent qui a avancé l'acte :  $S = Speaker(M)$ .
- $H$  est l'agent à qui le coup est adressé :  $H = Hearer(M)$ . Quand le coup est adressé à tous les agents, nous dénotons  $A_\Omega$ .
- $Acte = Acte(M)$  est l'acte lui-même.

Un dialogue est une séquence non vide de coups  $M_1, \dots, M_p$  telle que :

- $Speaker(M_j) \neq Hearer(M_j)$
- $\mathcal{CS}_{A_i}(0) = \emptyset, \forall i = 1, \dots, n$ . Notons que  $\mathcal{CS}_{A_i}(0)$  est le commitment store de l'agent  $A_i$  à l'étape 0.
- Pour tous coups  $M_j, M_k$ , si  $j \neq k$  alors  $Act(M_j) \neq Act(M_k)$
- Pour tout  $j < p$ :  $Speaker(M_j) \neq Speaker(M_{j+1})$ .

La première condition interdit à un agent d'adresser un coup à lui-même. La seconde condition dit que les  $\mathcal{CS}$ s sont vides au début du dialogue. La troisième condition évite de répéter un coup déjà avancé au cours du dialogue. Cette condition garantit un dialogue non circulaire. La dernière condition interdit aux agents d'avancer plusieurs coups au même temps. Ainsi, elle garantit que le dialogue ne sera pas un monologue.

Les agents ne sont pas libres d'avancer n'importe quel coup à n'importe quel moment. Leurs coups sont gérés par un *protocole de dialogue*, un ensemble de règles qui gouverne l'interaction entre les agents. Un protocole spécifie les différents actes que les agents peuvent avancer (affirmation, question, etc...), soit  $\mathcal{M}$  l'ensemble des actes permis. Pour chacun de ces actes, le protocole définit les pré-conditions qui doivent être vérifiées avant de l'avancer. Ces pré-conditions dépendent étroitement de la nature de l'agent. Un *agent rationnel*, par exemple, affirme uniquement des informations qu'il croit. Par contre, un agent irrationnel peut affirmer des informations que lui-même peut contredire. Un *agent coopératif* est contraint de répondre aux questions qui lui sont posées par d'autres agents. Le protocole spécifie aussi pour chaque acte l'ensemble des réponses possibles à cet acte. Les règles de mise à jour des tableaux de conversation sont aussi considérées comme une partie du protocole car elles dépendent de l'acte avancé. Ces trois composantes du protocole du dialogue peuvent être vues comme étant trois fonctions dénotées respectivement par  $\mathcal{R}$ ,  $\Pi$  et  $\mathcal{U}$ . La fonction  $\mathcal{R}$ , qui spécifie les pré-conditions, prend en entrée un agent et un acte de langage et retourne un booléen qui dit



si l'acte peut être avancé ou pas. La fonction  $\Pi$  retourne à partir du dernier acte avancé l'ensemble des actes (parmi  $\mathcal{M}$ ) qui peuvent être joués au coup suivant. La fonction  $\mathcal{U}$  décrit comment un tableau de conversation d'un agent peut être mis à jour après un coup donné.

**Définition 9**

Un protocole de dialogue est un tuple  $\mathcal{P} = \langle \mathcal{M}, \mathcal{R}, \Pi, \mathcal{U} \rangle$  tq :

- $\mathcal{M}$  : est l'ensemble des actes permis.
- $\mathcal{R} : \mathcal{M} \times \mathcal{A} \mapsto \{0, 1\}$
- $\Pi : \mathcal{M} \mapsto 2^{\mathcal{M}}$
- $\mathcal{U} : \mathcal{A} \times \mathcal{M} \times \mathcal{CS} \mapsto \mathcal{CS}$

Le protocole du dialogue spécifie les règles d'interaction entre les agents et les différentes réponses possibles après un acte donné. En général plusieurs réponses sont possibles et la réponse exacte choisie est le résultat de la stratégie adoptée par cet agent. La stratégie peut être vue comme une fonction qui prend en entrée l'ensemble des actes identifiés par le protocole et retourne un seul acte, c'est l'acte qui sera joué par l'agent. Formellement :

**Définition 10**

Une stratégie du dialogue est une fonction  $\langle \mathcal{S} : 2^{\mathcal{M}} \mapsto \mathcal{M} \rangle$  telle que pour  $T \subseteq \mathcal{M}$ ,  $\mathcal{S}(T) \in T$ .

Nous sommes maintenant en mesure de définir formellement un système de dialogue :

**Définition 11**

Un système de dialogue est un tuple  $\langle \mathcal{A}, \succ, \Pi, \mathcal{S}, \mathcal{D} \rangle$  où :

- $\mathcal{A}$  est un ensemble d'agents
- $\succ$  est un préordre sur les éléments de  $\mathcal{A}$
- $\Pi$  est un protocole
- $\mathcal{S}$  est un ensemble de stratégies
- $\mathcal{D}$  est un dialogue

Dans la suite de cette section nous illustrons l'idée d'un système de dialogue en instantiant chacun de ses éléments.

## 3.2 Illustration

### L'ensemble des agents

Nous considérons un ensemble d'agents  $\mathcal{A} = \{A_1, \dots, A_n\}$ ,  $n \geq 2$ , avec  $A_i$  est le nom du  $i$  ème agent. Dans toute la suite nous ne parlerons pas du rôle des agents. Notons uniquement que ces rôles sont importants pour la définition du préordre  $\succ$  sur l'ensemble  $\mathcal{A}$ .

Comme dans [9] nous supposons que chaque agent  $A_i$  a un ensemble de croyances  $B_{A_i}$ , un ensemble de désirs  $D_{A_i}$ , et un ensemble d'intentions  $I_{A_i}$ . Cependant pour l'instant nous ne traitons que des bases propositionnelles. Les croyances, désirs et intentions sont donc modélisés en partitionnant la base de manière appropriée. La *base de connaissances de base* d'un agent  $A_i$  est :

$$\Sigma_{A_i}^B = B_{A_i} \cup D_{A_i} \cup I_{A_i}$$

Ainsi, une information  $p$  est une croyance d'un agent  $A_i$  si elle appartient à  $B_{A_i}$ , elle représente une intention de  $A_i$  si elle appartient à  $I_{A_i}$  et enfin c'est un désir de  $A_i$  si elle appartient à  $D_{A_i}$ .

Pour pouvoir représenter le type d'informations échangées au cours d'un dialogue de type  *négociation*, nous étendons le langage propositionnel. Dans [12], la négociation est définie comme un échange d'offres (requêtes) entre des agents. Un agent  $A_i$  accepte la requête d'un autre agent  $A_j$  si ce dernier accepte la requête de  $A_i$ . Par exemple, "Si vous me prêtez votre ordinateur, je vous laisserai utiliser mon imprimante". Pour capturer ce genre d'informations, nous avons ajouté un nouveau connecteur  $\Rightarrow$  dans [3]. Ainsi nous avons un nouveau langage  $\mathcal{L}'$  qui contient des formules propositionnelles et des formules du type  $p \Rightarrow q$  telles que  $p$  et  $q$  sont des formules propositionnelles.

Comme nous l'avons décrit plus haut, chaque agent  $A_i$  utilise un tableau de conversation  $\mathcal{CS}_{A_i}$  pour garder trace de ses différents engagements au cours du dialogue.

Au cours d'un dialogue les agents peuvent échanger deux types d'informations : des connaissances et des préférences. Nous supposons donc qu'un tableau de conversation a deux parties : la première partie dénotée par  $\mathcal{CS}.Pref_{A_i}$  va contenir les engagements de  $A_i$  en termes de préférences et la deuxième partie dénotée par  $\mathcal{CS}.Kb_{A_i}$  va contenir les engagements de  $A_i$  en termes de connaissances.

$$\mathcal{CS}_{A_i} = \mathcal{CS}.Pref_{A_i} \cup \mathcal{CS}.Kb_{A_i}$$

Un agent  $A_i$  connaît tous ce qu'il y a dans les différents tableaux de conversation  $\cup_{j=1}^n \mathcal{CS}.Kb_{A_j}$ . Il connaît aussi certaines croyances des autres agents  $\cup_{j \neq i} \Sigma_{A_j}^{B'}$  avec  $\Sigma_{A_j}^{B'} \subseteq \Sigma_{A_j}^B$ . La connaissance globale disponible pour un agent  $A_i$  à n'importe quel moment est donc la suivante :

$$\Sigma_{A_i} = \Sigma_{A_i}^B \cup [\cup_{j \neq i} \Sigma_{A_j}^{B'}] \cup [\cup_{j=1}^n \mathcal{CS}.Kb_{A_j}]$$

Un agent possède des préférences sur sa base de connaissances. Ainsi, la base  $\Sigma_{A_i}^B$  d'un agent  $A_i$  est équipée d'un préordre (total ou partiel) dénoté par  $\gg_{A_i}^B$ . Les préférences sont données sous forme de paires  $(a, b)$  qui veut dire que la proposition  $a$  est préférée à la proposition  $b$ .

Un agent  $A_i$  peut connaître certaines préférences d'un autre agent  $A_j$ . Ces préférences, qui sont un sous-ensemble de  $\gg_{A_j}^B$ , seront dénotées par  $\gg_{A_j}^{B'}$ . Chaque agent connaît les préférences sur lesquelles se sont engagées les autres agents,  $\gg_{j=1, n} \mathcal{CS}.Pref_{A_j}$ . L'ensemble global des préférences disponibles pour un agent  $A_i$  est :

$$\gg_{A_i} = \gg_{A_i}^B \cup [\cup_{j \neq i} \gg_{A_j}^{B'}] \cup [\cup_{j=1, n} \mathcal{CS}.Pref_{A_j}]$$

Chaque agent est supposé équipé d'un système d'argumentation du type présenté dans la section 2. En utilisant la base  $\Sigma_{A_i}$ , l'agent  $A_i$  peut construire des arguments en faveur de ses croyances, ses désirs et ses intentions. Les agents sont considérés comme étant différents contextes. Ainsi, le système développé dans la section 2 peut être adapté en remplaçant l'ensemble des contextes  $\mathcal{C}$  par l'ensemble des agents  $\mathcal{A}$  et le préordre  $\sqsupseteq$  sur les contextes par le préordre  $\succ$  sur l'ensemble des agents. Chaque agent  $A_i$  utilisera donc le système d'argumentation suivant:

$$\langle \mathcal{A}(\Sigma_{A_i}), \text{Attaque}, \mathcal{A}, \succ, (\gg_{A_1}^{B'} \cup \mathcal{CS}.Pref_{A_1}), \dots, (\gg_{A_i}^B), \dots, (\gg_{A_n}^{B'} \cup \mathcal{CS}.Pref_{A_n}) \rangle$$

Le système d'argumentation est utilisé pour aider un agent à maintenir la cohérence de ses croyances, et pour s'assurer qu'il n'affirme que des informations vraies et des arguments acceptables. Ainsi, il aide à rendre opérationnelle la notion d'*agent rationnel*, et assure que les dialogues se terminent lorsque les agents sont d'accord sur l'acceptabilité des arguments avancés au cours du dialogue.

## Protocole du Dialogue

Cette section présente un protocole de dialogue en termes d'un ensemble d'actes  $\mathcal{M}$ . A chaque acte est associé un ensemble de règles : les règles de rationalité, les règles de dialogue et les règles de mise à jour. Les règles de rationalité représentent les pré-conditions d'un acte et les règles de dialogue précisent les réponses possibles à un acte donné. Les règles de mise à jour spécifient comment un tableau de conversation est mis à jour après un coup donné.

Dans la suite nous supposons que l'agent  $A_i$  adresse le coup aux autres agents.

### *Les actes de base*

Les actes de base sont *Affirme*, *Préfère*, *Défie* et *Question*.

**Affirme( $p$ )** où  $p$  est une formule de  $\mathcal{L}$ . Cet acte permet d'échanger des informations telles que "il fait beau" ou "j'ai l'intention d'écrire un article".

**Rationalité:** L'agent utilise son SAPC pour vérifier s'il a un argument acceptable en faveur de  $p$ .

**Dialogue:** –

- *Accepte*( $p$ )
- *Affirme*( $\neg p$ )
- *Défie*( $p$ )

**Mise à jour:**  $\mathcal{CS}.Kb_{t(A_i)} = \mathcal{CS}.Kb_{(t-1)(A_i)} \cup \{p\}$  et  
 $\mathcal{CS}.Kb_{t(A_j)} = \mathcal{CS}.Kb_{(t-1)(A_j)}, \forall j \neq i.$

L'information affirmée est ajoutée au  $\mathcal{CS}$  de l'agent qui a joué le coup. Notons qu'un autre agent  $A_j$  peut répondre uniquement si la règle de rationalité associée à la réponse est vérifiée.

**Affirme( $S$ )** où  $S$  est le support d'un argument.

**Rationalité:** L'agent utilise son SAPC pour vérifier si l'argument correspondant est acceptable.

**Dialogue:** –

- *Accepte*( $S$ )
- *Affirme*( $\neg q$ ), où  $q \in S$
- *Défie*( $q$ ), où  $q \in S$
- *Promets*( $q \Rightarrow r$ ), où  $q \in S$
- *Préfère*(( $a_1, b_1$ ), ..., ( $a_n, b_n$ )) où  $a_i, b_i$  sont des formules de  $\mathcal{L}$

**Mise à jour :**  $\mathcal{CS}.Kb_{t(A_i)} = \mathcal{CS}.Kb_{(t-1)(A_i)} \cup S$  et  
 $\mathcal{CS}.Kb_{t(A_j)} = \mathcal{CS}.Kb_{(t-1)(A_j)}, \forall j \neq i.$

De manière informelle, en réponse à une telle affirmation, un agent peut accepter tout le support de l'argument, défier ou nier un élément de ce support, promettre quelque chose en échange d'un élément de ce support ou tout simplement présenter ses préférences. Un agent peut présenter ses préférences grâce à l'acte suivant :

**Préfère**(( $\mathbf{a}_1, \mathbf{b}_1$ ), ..., ( $\mathbf{a}_n, \mathbf{b}_n$ )) où  $a_i, b_i$  sont des formules de  $\mathcal{L}$ .

**Rationalité :** Il n'y a pas de règle de rationalité (il n'existe aucune condition).

**Dialogue :** -

- *Préfère*(( $a'_1, b'_1$ ), ..., ( $a'_j, b'_j$ ))
- *Affirme*( $S$ )
- *Question*( $q$ )
- *Requête*( $q$ ), où  $q = b_i$
- *Promets*( $x \Rightarrow a_i$ )

**Mise à jour :**  $\mathcal{CS}.Pref_{t(A_i)} = \mathcal{CS}.Pref_{(t-1)(A_i)} \cup \{(a_1, b_1), \dots, (a_j, b_j)\}$   
 et  
 $\mathcal{CS}.Pref_{t(A_j)} = \mathcal{CS}.Pref_{(t-1)(A_j)}, \forall j \neq i.$

De manière informelle, un agent  $A_j$  peut répondre à l'agent  $A_i$  en présentant ses propres préférences qui doivent être différentes de celles de  $A_i$ . Il peut aussi donner un argument, poser une question, ou encore faire une requête sur quelque chose non préférée par  $A_i$ . Une autre possibilité pour un agent  $A_j$  est de promettre à  $A_i$  un élément que ce dernier préfère en échange d'autre chose.

Les deux prochains coups permettent aux agents de poser des questions.

**Question(p)**, où  $p$  est une formule de  $\mathcal{L}$ , dénote le fait que  $A_i$  pose la question "est ce que  $p$ ?".

**Rationalité:** Il n'existe aucune condition.

**Dialogue:** –

- *Affirme*( $p$ )
- *Affirme*( $\neg p$ )
- *Question*( $q$ )

**Mise à jour:**  $\mathcal{CS}.Kb_{t(A_i)} = \mathcal{CS}.Kb_{(t-1)(A_i)}$  et  
 $\mathcal{CS}.Kb_{t(A_j)} = \mathcal{CS}.Kb_{(t-1)(A_j)}$ ,  $\forall j \neq i$ .

**Défi(p)**, où  $p$  est une formule de  $\mathcal{L}$ , dénote le fait que  $A_i$  pose la question "pourquoi  $p$ ?".

**Rationalité:** Il n'existe aucune condition.

**Dialogue:** *Affirme*( $S$ ) où  $S$  est le support de l'argument ( $S, p$ ), ou bien  $S$  est le support d'un argument ( $S, h$ ) tel que  $p$  appartient à  $S$  et  $h$  est une intention de l'agent  $A_i$ .

**Mise à jour:**  $\mathcal{CS}.Kb_{t(A_i)} = \mathcal{CS}.Kb_{(t-1)(A_i)}$  et  
 $\mathcal{CS}.Kb_{t(A_j)} = \mathcal{CS}.Kb_{(t-1)(A_j)}$ ,  $\forall j \neq i$ .

### ***Les actes de négociation***

Les actes suivants permettent de capturer certains aspects de la négociation. Notons que ces actes peuvent être adressés uniquement à *un seul agent* ( $A_j$ ).

**Requête(p)** où  $p$  est une formule de  $\mathcal{L}$ .

**Rationalité:**  $A_i$  utilise son SAPC pour identifier un  $p$  dans  $\Sigma_{A_j}^{B'}$  tel que  $p \in H$  et ( $H, h$ ) est un argument acceptable en faveur de l'intention  $h$  de  $A_i$ .

**Dialogue:** –

- *Accepte*( $p$ )
- *Refuse*( $p$ )
- *Défi*( $p$ )

– *Promets*( $q \Rightarrow p$ )

**Mise à jour :**  $\mathcal{CS}.Kb_{t(A_i)} = \mathcal{CS}.Kb_{(t-1)(A_i)}$  et  
 $\mathcal{CS}.Kb_{t(A_j)} = \mathcal{CS}.Kb_{(t-1)(A_j)} \cup \{p\}$ .

Notons qu'une requête est stockée dans le  $\mathcal{CS}$  de l'agent à qui cette requête est adressée. La raison est simple : si l'agent accepte cette requête elle devient un de ses engagements. Une *Requête* est faite lorsqu'un agent ne peut pas réaliser ses intentions tout seul par manque de ressources. Contrairement à une affirmation, une requête ne peut pas être prouvée "vraie" ou "fausse". La décision pour l'accepter ou la rejeter dépend de la relation qu'elle a avec les intentions de l'autre agent.

**Promets**( $p \Rightarrow q$ ) où  $p$  et  $q$  sont des formules de  $\mathcal{L}$ .

**Rationalité :**  $A_i$  utilise son SAPC pour identifier un  $p$  dans  $\Sigma_{A_j}^{B'}$  tel que  $p \in H$  et  $(H, h)$  est un argument acceptable pour une intention de  $A_i$ , et vérifier s'il n'existe pas d'argument acceptable  $(H', h')$  pour une de ses intentions  $h'$  tel que  $q \in H'$ .

**Dialogue :** –

- *Accepte*( $p \Rightarrow q$ )
- *Refuse*( $p \Rightarrow q$ )
- *Promets*( $s \Rightarrow p$ )
- *Défi*( $p$ )
- *Préfère*(( $x, q$ ))

**Mise à jour :**  $\mathcal{CS}.Kb_{t(A_i)} = \mathcal{CS}.Kb_{(t-1)(A_i)} \cup \{q\}$  et  
 $\mathcal{CS}.Kb_{t(A_j)} = \mathcal{CS}.Kb_{(t-1)(A_j)} \cup \{p\}$ .

Un agent fait une promesse lorsqu'il a besoin de faire une requête  $p$  à un autre agent et qu'il possède une ressource  $q$  en échange de sa requête (car il n'a pas besoin de  $q$ ).

### *Les actes de réponses*

Ces actes sont joués en réponse à une affirmation ou à une requête. Les réponses dépendent de l'acte avancé au tour précédent.

**Accepte(p)** où  $p$  est une formule de  $\mathcal{L}$ . Après une affirmation ou une requête, un agent peut répondre avec une acceptation explicite.

**Rationalité:** En réponse à une affirmation,  $A_i$  utilise son système d'argumentation pour vérifier s'il a un argument acceptable en faveur de  $p$ . Si c'est le cas, alors le coup peut être avancé. En réponse à une requête,  $A_i$  vérifie s'il n'a pas un argument acceptable  $(H, h)$  en faveur d'une de ses intentions  $h$ , tel que  $p \in H$ . En d'autres termes, il est possible d'accepter une requête uniquement si elle n'invalide pas le support d'argument d'une de ses intentions<sup>1</sup>.

**Dialogue:** Les autres agents peuvent avancer n'importe quel acte sauf *Refuse*.

**Mise à jour:**  $\mathcal{CS}.Kb_{t(A_i)} = \mathcal{CS}.Kb_{(t-1)(A_i)} \cup \{p\}$  et  
 $\mathcal{CS}.Kb_{t(A_j)} = \mathcal{CS}.Kb_{(t-1)(A_j)}, \forall j \neq i.$

Notons que dans le cas d'une réponse à une requête,  $p$  est déjà dans le tableau de conversation de l'agent.

**Accepte(S)** où  $S$  est un ensemble de formules de  $\mathcal{L}$ .

**Rationalité:** Pour chaque  $p \in S$ ,  $A_i$  effectuera la même vérification que pour *Accepte(p)*. Ainsi, accepter un ensemble de formules est exactement comme accepter plusieurs formules individuellement.

**Dialogue:** Les autres agents peuvent jouer n'importe quel acte sauf *Refuse*.

**Mise à jour:**  $\mathcal{CS}.Kb_{t(A_i)} = \mathcal{CS}.Kb_{(t-1)(A_i)} \cup S$  et  
 $\mathcal{CS}.Kb_{t(A_j)} = \mathcal{CS}.Kb_{(t-1)(A_j)}.$

**Accepte(p  $\Rightarrow$  q)** où  $p$  et  $q$  sont des formules de  $\mathcal{L}$ .

**Rationalité:**  $A_i$  exécute la règle de rationalité correspondant à *Accepte(p)*.

**Dialogue:** Les autres agents peuvent avancer n'importe quel acte sauf *Refuse*.

**Mise à jour**  $\mathcal{CS}.Kb_{t(A_i)} = \mathcal{CS}.Kb_{(t-1)(A_i)}$  et  
 $\mathcal{CS}.Kb_{t(A_j)} = \mathcal{CS}.Kb_{(t-1)(A_j)}, \forall j \neq i.$

---

1. Un argument en faveur d'une intention peut être vu comme un plan pour achever cette intention, donc donner une ressource  $p$  invalide le plan.



**Refuse(p)** où  $p$  est une formule de  $\mathcal{L}$ .

**Rationalité:**  $A_i$  utilise son SAPC pour vérifier s'il existe un argument acceptable  $(H, h)$  en faveur d'une de ses intentions  $h$  tel que  $p \in H$ .

**Dialogue:** Les autres agents peuvent avancer n'importe quel acte sauf *Refuse*.

**Mise à jour:**  $\mathcal{CS}.Kb_{t(A_i)} = \mathcal{CS}.Kb_{(t-1)(A_i)} \setminus \{p\}$  et  
 $\mathcal{CS}.Kb_{t(A_j)} = \mathcal{CS}.Kb_{(t-1)(A_j)}, \forall j \neq i$ .

Un agent refuse les requêtes qui sont nécessaires pour achever ses intentions.  
Il peut aussi refuser une promesse :

**Refuse(p  $\Rightarrow$  q)** où  $p$  et  $q$  sont des formules de  $\mathcal{L}$ .

**Rationalité:**  $A_i$  utilise son SAPC pour vérifier s'il a un argument acceptable  $(H, h)$  en faveur d'une de ses intentions  $h$  tel que  $p \in H$ .

**Dialogue:** Les autres agents peuvent avancer n'importe quel acte sauf *Refuse*.

**Mise à jour:**  $\mathcal{CS}.Kb_{t(A_i)} = \mathcal{CS}.Kb_{(t-1)(A_i)} \setminus \{q\}$  et  
 $\mathcal{CS}.Kb_{t(A_j)} = \mathcal{CS}.Kb_{(t-1)(A_j)} \setminus \{p\}$ .

## Stratégie du dialogue

La stratégie sélectionne un seul acte parmi l'ensemble des réponses possibles à un acte donné. Pour l'ensemble des actes que nous avons présenté, nous pouvons identifier différentes stratégies reflétant différents types d'agents :

- *Accepter* chaque fois que cela est possible.
- *Accepter* uniquement lorsqu'il n'y a pas de raison de ne pas le faire.
- *Défier* uniquement lorsque cela est nécessaire.
- *Défier* chaque fois que cela est possible.
- Répondre toujours à une *promesse* par une meilleure contre-promesse.
- Poser des questions chaque fois que cela est possible.

Bien que nous ayons nommé ces stratégies, chacune d'elles n'est qu'une définition partielle de  $\mathcal{S}$ . Une définition complète devrait tenir compte de la nature du coup précédent et par conséquent du protocole du dialogue. Par exemple, la première stratégie (qui consiste à *accepter* chaque fois que cela est possible) définit comment un agent doit répondre à *affirme*( $p$ ) quand il n'a pas d'argument contre  $p$ . Cependant, s'il en possède un, alors une stratégie complète devrait choisir entre *affirme*( $\neg p$ ) et *Défi*( $p$ ).

Le choix d'une stratégie peut avoir un impact sur les propriétés du dialogue, en particulier sur celles reliées à la longueur et à la terminaison. Par exemple, la stratégie qui consiste à toujours poser des questions entraîne une chaîne infinie de questions. Un travail préliminaire sur ce sujet a été présenté dans [10]. Les auteurs ont montré que la terminaison est garantie pour certains types de dialogue et cela en utilisant un sous ensemble des actes de dialogue présentés dans ce papier.

Un autre aspect qui est lié à la notion de stratégie, est comment un agent choisit quel argument proposer au cours d'un dialogue. Une approche évidente serait de choisir l'argument le plus court (avec le plus petit support), afin de limiter son exposition à l'attaque.

## 4 EXEMPLE

Cette section présente un exemple de dialogue de type *persuasion*. Il s'agit d'un dialogue entre un journaliste P et un politicien C. C est supposé plus fiable que P donc  $C \succ P$ . Le dialogue est le suivant :

**P:** Les journaux n'ont pas le droit de publier l'information X.

**C:** Pourquoi?

**P:** Parce qu'elle concerne la vie privée de A et ce dernier n'est pas d'accord.

**C:** Mais A est un ministre, et toute information concernant un ministre est publique.

**P:** Je sais mais cela est moins important que la vie privée d'une personne.

**C:** Non, en politique la fonction d'une personne est plus importante qu'une autre chose.

	$\Sigma_P$	$\Sigma_C$
1	$\neg agr, pri, min$	$min, min \rightarrow \neg pri$
2	$pri \wedge \neg agr \rightarrow \neg pub$	$pri$
3	$min \rightarrow \neg pri$	

TAB. 1 – *Les bases de connaissances*

L'agent  $P$  est supposé muni de la base  $\Sigma_P = \Sigma_{P1} \cup \Sigma_{P2} \cup \Sigma_{P3}$  avec  $\Sigma_{P1} = \{\neg acc, pri, min\}$ ,  $\Sigma_{P2} = \{pri \wedge \neg acc \rightarrow \neg pub\}$ ,  $\Sigma_{P3} = \{min \rightarrow \neg pri\}$ . L'agent  $C$  est muni de la base  $\Sigma_C = \Sigma_{C1} \cup \Sigma_{C2}$  avec  $\Sigma_{C1} = \{min, min \rightarrow \neg pri\}$ ,  $\Sigma_{C2} = \{pri\}$ .  $acc$  dénote "A est d'accord",  $pri$  dénote "privée",  $min$  dénote "A est ministre", et  $pub$  dénote "les journaux peuvent publier X". (voir Table 1)

Dans notre modèle ce dialogue se déroule comme le montre la table 2. Le dialogue commence lorsque  $P$  présente un argument,  $A = (\{pri, \neg acc, pri \wedge \neg acc \rightarrow \neg pub\}, \neg pub)$ , en faveur de "ne pas publier X". Cet argument est initialement acceptable dans le système d'argumentation de l'agent  $P$  car il se défend seul contre son unique attaquant  $B = (\{min, min \rightarrow \neg pri\}, \neg pri)$ . L'agent  $C$  donne ses préférences qui sont en conflit avec celles de  $P$ . Puisque  $C$  est supposé plus fiable que  $P$ , alors le système d'argumentation de  $P$  conclura que l'argument  $A$  n'est plus acceptable car il n'est pas préféré à  $B$  dans le contexte  $C$ .

## 5 CONCLUSION

Le travail présenté ici concerne la modélisation des dialogues inter-agents en utilisant des techniques d'argumentation. Notre première contribution est de présenter un système général de dialogue. Nous avons montré en détail les différentes composantes d'un tel système. Ensuite nous avons illustré ce système en instanciant chaque composante dans le but de capturer les différents types de dialogues discutés dans [15].

La deuxième contribution de ce travail est de prendre en compte les préférences (éventuellement conflictuelles) des agents. Au cours d'un dialogue, les agents peuvent échanger deux types d'informations: des connaissances et des préférences. Nous avons montré comment à la lumière de nouvelles

Move	CS
$\langle P, C, \text{affirme}(\neg pub) \rangle$	$\mathcal{CS}.Kb_P = \{\neg pub\},$ $\mathcal{CS}.Pref_P = \emptyset$ $\mathcal{CS}.Kb_C = \emptyset,$ $\mathcal{CS}.Pref_C = \emptyset$
$\langle C, P, \text{Défi}(\neg pub) \rangle$	$\mathcal{CS}.Kb_P = \{\neg pub\},$ $\mathcal{CS}.Kb_C = \emptyset$
$\langle P, C, \text{affirme}(\{pri, \neg agr, pri \wedge \neg agr \rightarrow \neg pub\}) \rangle$	$\mathcal{CS}.Kb_P = \{\neg pub, pri, \neg agr, pri \wedge \neg agr \rightarrow \neg pub\}$ $\mathcal{CS}.Kb_C = \emptyset$
$\langle C, P, \text{affirme}(\{min, min \rightarrow \neg pri\}) \rangle$	$\mathcal{CS}.Kb_P = \{\neg pub, pri, \neg agr, pri \wedge \neg agr \rightarrow \neg pub\}$ $\mathcal{CS}.Kb_C = \{min, min \rightarrow \neg pri\}$
$\langle P, C, \text{préfère}(\{pri, min \rightarrow \neg pri\}) \rangle$	$\mathcal{CS}.Pref_P = \{\{pri, min \rightarrow \neg pri\}\}$ $\mathcal{CS}.Pref_C = \emptyset$
$\langle C, P, \text{préfère}(\{min \rightarrow \neg pri, pri\}) \rangle$	$\mathcal{CS}.Kb_P = \{\neg pub, pri, \neg agr, pri \wedge \neg agr \rightarrow \neg pub\}$ $\mathcal{CS}.Pref_P = \{\{pri, min \rightarrow \neg pri\}\}$ $\mathcal{CS}.Kb_C = \{min, min \rightarrow \neg pri\}$ $\mathcal{CS}.Pref_C = \{\{min \rightarrow \neg pri, min\}\}$

TAB. 2 – Le changement des CS au cours du dialogue

préférences un agent peut changer sa croyance en une donnée. Parmi les extensions envisagées, citons en particulier l'étude de la stratégie adoptée par un agent pour choisir l'acte à accomplir. Dans ce papier nous avons présenté quelques idées et exemples de stratégies. Cependant elles ne sont pas exhaustives. Notre but est de relier cela aux travaux sur les actes de langage où un acte est supposé avoir des pré-conditions (appelées conditions préliminaires).

## RÉFÉRENCES

- [1] L. Amgoud, C. Cayrol et D. LeBerre. Comparing arguments using preference orderings for argument-based reasoning. In *Proceedings of the 8th International Conference on Tools with Artificial Intelligence*, pages 400–403, 1996.
- [2] L. Amgoud, N. Maudet et S. Parsons. Modelling dialogues using argumentation. In *Proceedings of the International Conference on Multi-Agent Systems*, pages 31–38, Boston, MA, 2000.

- [3] L. Amgoud, S. Parsons et N. Maudet. Arguments, dialogue, and negotiation. In *Proceedings of the Fourteenth European Conference on Artificial Intelligence*, pages 338–342, Berlin, Germany, 2000.
- [4] L. Amgoud, S. Parsons et L. Perrussel. An argumentation framework based on contextual preferences. In *Proceedings of the International Conference on Formal and Applied and Practical Reasoning*, pages 59–67, 2000.
- [5] G. Brewka. Dynamic argument systems: a formal model of argumentation process based on situation calculus. *Journal of Logic and Computation*, 11(2):257–282, 2001.
- [6] T. F. Gordon. The pleadings game. *Artificial Intelligence and Law*, 2:239–292, 1993.
- [7] J. MacKenzie. Question-begging in non-cumulative systems. *Journal of Philosophical Logic*, 8:117–133, 1979.
- [8] R. McConachy et I. Zukerman. Dialogue requirements for argumentation systems. In *Proceedings of IJCAI'99 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, 1999.
- [9] S. Parsons, C. Sierra et N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.
- [10] S. Parsons, M. Wooldridge et L. Amgoud. An analysis of formal inter-agent dialogues. In *The first International joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS. To appear*, 2002.
- [11] H. Prakken. On dialogue systems with speech acts, arguments, and counterarguments. In *7th European Workshop on Logic for Artificial Intelligence*, Malaga, 2000.
- [12] C. Sierra, N. R. Jennings, P. Noriega et S. Parsons. A framework for argumentation-based negotiation. In *Proceedings of the 4th International Workshop on Agent Theories, Architectures and Languages*, 1997.
- [13] K. Sycara. Persuasive argumentation in negotiation. *Theory and Decision*, 28:203–242, 1990.
- [14] F. Tohmé. Negotiation and defeasible reasons for choice. In *Proceedings of the Stanford Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning*, pages 95–102, 1997.
- [15] D. N. Walton et E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. State University of New York Press, Albany, NY, 1995.

- [16] S. Zabala, I. Lara et H. Geffner. Beliefs, reasons and moves in a model for argumentation dialogues. In *Proceedings of the Latino-American Conference on Computer Science*, 1999.