

Mémoire de Master Recherche

Spécialité Génie Informatique, Université de Rouen

Laboratoire LITIS

(Laboratoire d'Informatique, de Traitement de l'Information et des Systèmes)

Modélisation cognitive des interactions dans un processus de négociation : approche multi-agent

Présenté par :

Charlène FLOCH

Encadré par :

Nathalie CHAIGNAUD

Alexandre PAUCHET

Soutenu le 12 juillet 2007

charlene.floch@gmail.com



À N.B.

Remerciements

Mes remerciements s'adressent (en premier lieu, mais ce n'est pas qu'une question de forme) à Nathalie Chaignaud, ma directrice de stage, pour m'avoir proposé ce sujet, pour ses conseils précieux, pour la confiance qu'elle m'a accordée, et pour les conditions de travail idéales dont j'ai pu bénéficier durant mon stage au laboratoire,

Je tiens ensuite à remercier Alexandre Pauchet, pour m'avoir guidée notamment dans la compréhension de ses travaux de thèse, pour sa réactivité par mail, ses conseils et remarques avisés sur mon travail malgré la distance,

Je remercie l'ensemble des membres du laboratoire LITIS :

- ceux dont j'ai suivi les enseignements ces dernières années, et qui sont sans aucun doute responsables de mon envie de poursuivre vers la recherche,
- ceux dont j'ai croisé la route, à Rouen ou Le Havre, et qui se reconnaîtront, merci pour les discussions passionnantes,

Merci aux stagiaires, ingénieurs, personnels, des sites de Mont-Saint-Aignan (J2) et du Madrillet, pour l'ambiance sympathique et chaleureuse, merci à Jean-François pour les très bonnes conditions matérielles durant les expérimentations,

Je tiens à remercier toutes les personnes qui ont participé aux expérimentations et aux tests de la plate-forme qui ont précédé, merci d'avoir *joué le jeu* !

Un merci tout particulier à Tapu ;-)

Merci Turbu, Bruno, Fred, Corwin, Cyco, and *Motards & Co.* !

Enfin, merci Yaya, pour les conseils, coups de main, critiques constructives, et autres parties de ping-pong verbal !

Table des matières

REMERCIEMENTS.....	3
TABLE DES MATIÈRES.....	4
TABLE DES FIGURES.....	6
INTRODUCTION.....	7
CONTEXTE.....	9
1.1. AGENTS ET SYSTÈMES MULTI-AGENTS.....	9
1.2. LE SYSTÈME BDIGGY.....	12
1.3. LA NÉGOCIATION.....	18
2. LA NÉGOCIATION DANS LES SMA.....	19
2.1. LA NOTION DE NÉGOCIATION DANS LES SMA.....	19
2.2. MODÉLISATION DE LA NÉGOCIATION DANS LES SMA.....	25
2.2.1. <i>Les éléments de la négociation</i>	26
2.2.2. <i>Les deux types de négociation</i>	27
2.2.3. <i>L'aide à la concertation et à la négociation</i>	29
2.2.4. <i>L'approche théorie des jeux</i>	30
2.2.5. <i>Théorie des jeux et IAD</i>	32
2.3. LA COMMUNICATION ENTRE AGENTS.....	33
2.3.1. <i>Modes de communication</i>	33
2.3.2. <i>Les langages de communication</i>	34
2.3.3. <i>Notion de protocole</i>	35
2.3.4. <i>Les protocoles de négociation entre agents</i>	35
2.3.4.1. Le protocole Contract Net.....	35
2.3.4.2. Le protocole de Sian.....	37
2.3.4.3. Le protocole SANP.....	39
2.3.4.4. Les protocoles dans les systèmes d'enchères.....	43
2.3.4.5. Le protocole dans GENCA.....	48
3. LES EXPÉRIMENTATIONS.....	50
3.1. LA PROBLÉMATIQUE DES EXPÉRIMENTATIONS.....	50
3.1.1. <i>Le recueil des données et la question de la verbalisation</i>	51
3.1.2. <i>Le choix du problème</i>	53
3.2. LE CADRE EXPÉRIMENTAL.....	54
3.2.1. <i>Description du jeu</i>	54
3.2.2. <i>Le logiciel d'expérimentation</i>	57
3.2.3. <i>Le déroulement des expérimentations</i>	59
4. ANALYSE DES TRACES.....	61
4.1. LES TRACES.....	61
4.2. NIVEAUX MICRO ET MACRO.....	62
4.3. LES ÉCHANGES.....	63
4.4. ÉTUDE STATISTIQUE.....	68
4.5. CRITIQUE DES MODÈLES DE NÉGOCIATION FONDÉS SUR LE CONTRACT NET.....	71

5. MODÈLE PROPOSÉ.....	77
5.1. LES AGENTS DE BDIGGY : DE LA COOPÉRATION INCONDITIONNELLE À LA NÉGOCIATION.....	77
5.1.1. <i>L'objet de la négociation.....</i>	78
5.1.1.1. Prédicat Transaction pTrans.....	78
5.1.2. <i>Les connaissances de l'agent.....</i>	80
5.2. MODÉLISATION DES INTERACTIONS.....	80
5.2.1. <i>Les performatives.....</i>	80
5.2.2. <i>Les automates de conversations.....</i>	81
5.2.2.1. Automates de BDIGGY utilisés.....	82
5.2.2.2. Automates de discussion.....	83
5.2.2.3. Automate de relance.....	85
5.2.2.4. Automate de mise en garde.....	86
5.2.2.5. Automate de mise en vente.....	86
5.2.3. <i>Un exemple.....</i>	87
5.2.4. <i>Choix de conception.....</i>	88
5.2.5. <i>La sémantique des performatives.....</i>	90
5.2.5.1. Sémantique des descriptifs.....	91
5.2.5.2. Sémantique des directifs.....	93
5.2.5.3. Sémantique des engageants.....	96
5.2.5.4. Adaptation de la sémantique.....	97
5.2.6. <i>La notion de satisfaction.....</i>	98
5.2.7. <i>La mémoire.....</i>	99
5.2.8. <i>L'architecture.....</i>	100
5.2.9. <i>Parallèle avec l'approche théorie des jeux.....</i>	102
CRITIQUE DE NOTRE TRAVAIL.....	103
CONCLUSION ET PERSPECTIVES.....	105
Bibliographie.....	107

Table des figures

FIG. 1 - UN AGENT INTELLIGENT DE NEWELL ET SIMON, TIRÉ DE (NEWELL, 1982, P.88).....	9
FIG. 2 - LISTE DE PERFORMATIVES DANS LE MODÈLE BDIGGY (PAUCHET, 2006, P.87).....	14
FIG. 3 - AUTOMATE DE DEMANDE D'INFORMATION Q_{INI} , TIRÉ DE (PAUCHET, 2006, P.113).....	15
FIG. 4 - AUTOMATE DE DEMANDE D'INFORMATION Q_{INT} , TIRÉ DE (PAUCHET, 2006, P.113).....	16
FIG. 5 - FLUX DE L'ARCHITECTURE BDIGGY (PAUCHET, 2006, P.137).....	17
FIG. 6 - LES DIFFÉRENTS ÉLÉMENTS D'UNE NÉGOCIATION ET LEURS STRUCTURES (MÜLLER, 1996).....	27
FIG. 7 - LES DIFFÉRENTS TYPES DE NÉGOCIATION (SIMOS, 1990).....	28
FIG. 8 - MATRICE DE GAIN DU DILEMME DU PRISONNIER.....	31
FIG. 9 - INTERACTIONS DANS LE PROTOCOLE CONTRACT NET ET LE PROTOCOLE CONTRACT NET ITÉRÉ (FIPA).....	36
FIG. 10 - LE PROTOCOLE CONTRACT NET.....	36
FIG. 11 - GRAPHE DE LA CONVERSATION DANS LE PROTOCOLE DE SIAN.....	37
FIG. 12 - PERFORMATIVES UTILISÉES DANS LE PROTOCOLE DE SIAN.....	38
FIG. 13 - DIAGRAMME DE STRATÉGIE (CHANG ET WOO, 1994).....	40
FIG. 14 - SOUS-DIAGRAMME DE RAISONNEMENT DE L'ATTACQUANT, DANS LE PROTOCOLE SANP.....	41
FIG. 15 - EXEMPLES DE VERBES D'ACTE DE DISCOURS DANS LES CATÉGORIES UTILISÉES DANS SANP.....	41
FIG. 16 - EXTRAIT D'UNE EXÉCUTION DE LA PLATE-FORME KASBAH (CHAVEZ ET MAES, 1996).....	45
FIG. 17 - PLATE-FORME EXPÉRIMENTALE.....	54
FIG. 18 - CAPTURE D'ÉCRAN DU LOGICIEL D'EXPÉRIMENTATION, VUE POUR UN ACHETEUR.....	58
FIG. 19 - CAPTURE D'ÉCRAN DU LOGICIEL D'EXPÉRIMENTATION, VUE POUR UN VENDEUR.....	59
FIG. 20 - LES INTERACTIONS OBSERVÉES DURANT LES EXPÉRIMENTATIONS.....	62
FIG. 21 - LES NIVEAUX MICRO ET MACRO.....	63
FIG. 22 - LISTE DES PERFORMATIVES.....	66
FIG. 23 - TYPES D'INTERVENTIONS SUR L'ENSEMBLE DES GROUPES.....	69
FIG. 24 - DURÉE DES VENTES PENDANT UNE PARTIE.....	70
FIG. 25 - TYPES D'ÉCHANGES ENTRE LES PARTICIPANTS DURANT UNE VENTE.....	70
FIG. 26 - PROTOCOLE DE NÉGOCIATION DANS GENCA.....	72
FIG. 27 - LES SOURCES DE CONNAISSANCES DE L'AGENT.....	80
FIG. 28 - TYPES D'ÉCHANGES.....	81
FIG. 29 - AUTOMATES D'ENVOI SPONTANÉ D'INFORMATION.....	82
FIG. 30 - AUTOMATE DE L'INITIAUTEUR DE LA DEMANDE D'INFORMATION.....	82
FIG. 31 - AUTOMATE DE L'AGENT RECEVANT LA DEMANDE D'INFORMATION.....	83
FIG. 32 - AUTOMATE DE DISCUSSION D_{INI} DE L'ACHETEUR.....	84
FIG. 33 - AUTOMATE DE DISCUSSION D_{INT} DU VENDEUR.....	84
FIG. 34 - AUTOMATE DE DISCUSSION D_{INI} DU VENDEUR.....	85
FIG. 35 - AUTOMATE DE DISCUSSION D_{INT} DE L'ACHETEUR.....	85
FIG. 36 - AUTOMATES DE RELANCE.....	86
FIG. 37 - AUTOMATES DE MISE EN GARDE.....	86
FIG. 38 - AUTOMATE DE MISE EN VENTE.....	87
FIG. 39 - UN EXEMPLE DE SITUATION D'INTERACTION ENTRE LES PARTICIPANTS.....	88
FIG. 40 - AUTOMATE DE DEMANDE D'INFORMATION Q_{INI}	99
FIG. 41 - STRUCTURE D'ORIGINE DE LA MÉMOIRE DANS BDIGGY.....	100
FIG. 42 - MÉMOIRE MODIFIÉE DANS BDIGGY.....	100
Fig. 43 - Modifications de l'architecture d'un agent.....	101

Introduction

[15:09:35] 2, Jack SAY Bonjour tout le monde
[15:10:38] 3, William SAY Bonjour
[15:10:42] 4, Averell SAY salut
[15:10:58] 5, William SAY il manque joe
[15:11:08] 6, Joe SAY Bonjour, my name is Joe
[15:11:14] 7, Averell SAY il fait la peau à Lucky Luke
[15:11:22] 8, Jack SAY tout le monde est là, on peut y aller ?
[15:11:27] 9, Averell SAY ok
[15:11:31] 10, Joe SAY ok

Groupe d'expé. n°8

Notre travail se situe dans les domaines de l'Intelligence Artificielle Distribuée (IAD) et de la Modélisation Cognitive (MC). Dans la continuité des travaux de Nathalie Chaignaud, Amal El Fallah Seghrouchni et Alexandre Pauchet sur la résolution coopérative de problèmes, ce stage a pour objectif d'enrichir les interactions des agents du système BDIGGY avec la négociation. Avec l'essor du commerce électronique, la conception d'agents logiciels capables de participer à des négociations fait l'objet de nombreux travaux de recherche. L'originalité de notre approche est qu'elle est basée sur les résultats d'une expérimentation psychologique, réalisée sur des sujets humains placés en situation de négociation.

Le système BDIGGY (Pauchet *et al.*, 2005) (Pauchet, 2006) a été conçu pour simuler des comportements humains de planification pour laquelle la coopération est indispensable. Il implémente un modèle de planification coopérative humaine et un modèle d'interaction humaine. Ces modèles cognitifs sont tous deux fondés sur l'analyse de traces de résolution de problèmes, recueillies au cours d'une expérimentation psychologique. BDIGGY est né de la fusion entre le système IGGY (Chaignaud et Levy, 1996) et une architecture BDI (Beliefs, Desirs, Intentions) étendue à la résolution coopérative de problèmes.

L'enjeu de notre étude est, d'une part, d'enrichir le modèle BDIGGY, en ajoutant des capacités de négociation au modèle d'interaction, et d'autre part, de montrer la générnicité de ce modèle en l'utilisant sur un autre type d'application que celui pour lequel il a été conçu.

Le contenu de ce stage de Master est la première étape du projet, et s'est concentré sur :

- la mise en place de l'expérimentation psychologique,
- la formalisation des échanges en situation de négociation pour aboutir à un protocole de négociation entre agents,
- les modifications à apporter au modèle BDIGGY.

Mots-clés.: Systèmes Multi-Agents (SMA), modélisation cognitive, interactions, négociation.

Plan du mémoire

Dans une première partie, nous présentons le contexte dans lequel s'inscrit le projet : le domaine des systèmes multi-agents (SMA), le système BDIGGY et la négociation.

La deuxième partie est consacrée à la façon dont est abordée la négociation dans les SMA. Après avoir discuté la notion de négociation, nous proposons une définition. Nous présentons ensuite un état de l'art de la modélisation de la négociation en nous focalisant sur les protocoles d'interactions entre agents.

Dans une troisième partie, nous décrivons notre démarche, le choix du problème, le logiciel développé, la mise en place et le déroulement des expérimentations.

La quatrième partie présente l'analyse des traces recueillies et l'étude statistique réalisée. Nous mettons ensuite en évidence l'inadéquation des protocoles se basant sur le Contract Net pour modéliser ce que nous avons observé.

Dans une cinquième partie, nous détaillons le modèle proposé, les éléments ajoutés au modèle BDIGGY, et les modifications de l'architecture.

Nous terminons par le bilan du travail réalisé et les perspectives envisagées.

Contexte

Dans cette partie, nous commençons par présenter le domaine des systèmes multi-agents. Nous présentons ensuite le modèle BDIGGY : le modèle de planification (sur lequel nous passons rapidement car il ne concerne pas la négociation), le modèle de l'interaction, et l'architecture. Nous présentons ensuite ce que signifie *négociation*, et la démarche que nous souhaitons adopter pour enrichir le modèle.

1.1. Agents et Systèmes Multi-Agents

La notion d'*agent* est à la base des débuts de l'intelligence artificielle. Cette discipline s'est d'abord focalisée sur la conception d'une entité unique pour résoudre les problèmes. Depuis la fin des années 70, les chercheurs ont commencé à travailler sur la résolution distribuée de problèmes, ce que l'on a alors commencé à appeler Systèmes Multi-Agents (Briot et Demazeau, 2001). On a ainsi fait la différence entre l'IA classique et l'IAD (Intelligence Artificielle Distribuée). La première s'intéresse à la modélisation du comportement d'une seule entité, alors que la seconde traite de la modélisation de systèmes, au sein desquels interagissent plusieurs entités.

En 1980, Newell fut le premier à parler d'*agent*. Chercheur en IA classique et psychologie cognitive, il présenta un *agent intelligent*, lors de son discours d'inauguration de AAAI (*American Association for Artificial Intelligence*, maintenant appelée *Association for the Advancement of Artificial Intelligence*) dont il fut le premier président. Il y présentait ses travaux sur le Niveau Connaissance *Knowledge Level* (Newell, 1982), l'*agent* étant le système du *Niveau Connaissance*.

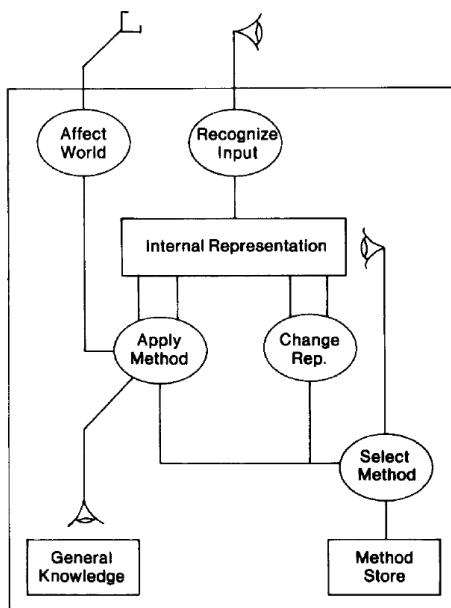


FIG. 1 - Un agent intelligent de Newell et Simon, tiré de (Newell, 1982, p.88)

Le schéma de la FIG. 1 (Newell, 1982, p.88), est une vue fonctionnelle d'un agent intelligent dans un environnement de tâches. Newell définit l'agent comme une entité ayant une structure très simple, il a :

- un *corps physique*, avec lequel il peut interagir avec l'environnement,
- un *corps de connaissances*, qui contient tout ce que l'agent connaît à un moment donné,
- des buts.

Depuis la proposition de Newell, de nombreuses définitions et architectures ont été proposées. Il n'existe pas de consensus sur la définition d'un agent. Nous adopterons la définition de Ferber (Ferber, 1995, p.13), communément utilisée dans le domaine SMA :

« *On appelle agent une entité physique ou virtuelle*

- *qui est capable d'agir dans un environnement,*
- *qui peut communiquer directement avec d'autres agents,*
- *qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser),*
- *qui possède des ressources propres,*
- *qui est capable de percevoir (mais de manière limitée) son environnement,*
- *qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune),*
- *qui possède des compétences et offre des services,*
- *qui peut éventuellement se reproduire,*
- *dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit. »*

On distingue habituellement les agents réactifs des agents cognitifs.

Un **agent réactif** ne possède pas de représentation explicite de son environnement. Il fonctionne sur un mode de type stimulus/réponse. Les formes d'interaction entre agents réactifs sont simples, ils communiquent de façon indirecte *via* l'environnement, en y laissant des signaux (comme les phéromones des fourmis par exemple). Les organisations d'agents réactifs possèdent généralement un grand nombre d'agents.

Un **agent cognitif**, quant à lui, possède une représentation plus élaborée de son environnement, des buts explicites et des capacités de raisonnement individuelles sur le problème et sur les autres agents. Il peut être doté de capacités de mémorisation de ses actions passées, de capacités de planification, etc. Les interactions avec les autres agents peuvent se faire directement par envois de messages (voir partie 2.2.6.1. Modes de communication).

On présente souvent ces deux types d'agent comme opposés. Comme l'explique Ferber dans (Ferber, 1996), il ne faut pas prendre cette opposition réactif-cognitif comme « *une opposition d'essence* », mais plutôt comme « *des repères dans l'analyse des SMA* ».

Il existe en effet des agents appelés **agents hybrides**, qui combinent des caractéristiques réactives et cognitives. Comme le fait remarquer Guessoum dans (Guessoum, 1996, p.22), les termes *réflexe* et *réflexion* ont la même racine, et l'expression *réflexe contrôlé* prend tout son sens dans ce type d'architecture d'agent. Les agents hybrides ont la possibilité d'adapter leur comportement en temps réel, à l'évolution de l'environnement.

Dans notre travail, les agents communiquent directement entre eux et *via* l'environnement, mais n'ont pas de position au sens physique du terme dans cet environnement. Ils sont *non situés*, communicants, et de type BDI (Beliefs Desires Intentions).

Les architectures des **agents BDI** sont fondées sur les trois notions suivantes :

- les **croyances** (la connaissance que possède un agent sur son environnement),
- les **désirs** (les états de fait qu'il souhaite voir réalisés, ce sont ses buts),
- les **intentions** (les désirs qu'il a l'intention de réaliser, et qui correspondent aux actions qu'il va entreprendre).

Le modèle BDI (Rao et Georgeff, 1995) est issu des travaux sur l'intentionnalité, comme ceux de Bratman par exemple (Bratman, 1987). Dans les SMA, on peut citer les travaux influents de Wooldridge, notamment l'ouvrage (Wooldridge, 2000).

Pour définir un **SMA**, dans (Briot et Demazeau, 2001), les auteurs prennent la définition usuelle du système (un ensemble organisé d'éléments) : c'est « *un ensemble organisé d'agents* ». Dans un SMA, il existe une ou plusieurs organisation(s) qui structure(nt) les règles de cohabitation et de travail collectif entre agents. Il s'agit d'une **métaphore de l'organisation collective**. Chaque agent a ses propres compétences, et a besoin d'interagir avec les autres pour résoudre des problèmes. L'objectif d'un SMA est de résoudre ces problèmes ou bien de modéliser des comportements complexes à l'aide d'un ensemble d'agents autonomes, proactifs et adaptatifs (Guessoum, 2003, p.8) :

- l'agent est **autonome**, il agit sans l'intervention des humains ou des autres agents, il contrôle ses actions en fonction de son état interne et de sa perception de l'environnement,
- il est **proactif**, c'est-à-dire qu'il agit de lui-même, il a sa propre activité dirigée vers ses buts (contrairement aux objets dont l'action est simplement une réponse aux messages reçus),
- il est **adaptatif**, il est capable de réguler ses aptitudes (comportementales, communicationnelles, etc.), en fonction des agents avec qui il interagit, et/ou en fonction de l'environnement dans lequel il évolue.

Les SMA sont fondés sur un modèle organisationnel. Ils permettent de faire apparaître des capacités différentes et globalement supérieures à celles des agents qui les composent. Les agents interagissent entre eux et avec l'environnement, et les solutions aux problèmes qu'ils traitent émergent de ces interactions. À noter que l'on peut trouver des modèles comprenant des agents qui sont eux-mêmes des SMA, selon le *principe de récursion* (Demazeau, 1995).

1.2. Le système BDIGGY

La thèse d'A. Pauchet (Pauchet, 2006) présente une étude de comportements humains de résolution coopérative de problèmes en connaissances incomplètes. Une expérimentation psychologique a permis de collecter des traces qui ont été analysées du point de vue de la planification et du point de vue des interactions. Ainsi, un modèle de la planification humaine et un modèle de l'interaction humaine ont été conçus et intégrés à une nouvelle architecture d'agent appelée BDIGGY.

Le problème de l'agence de voyage

Dans le problème choisi pour l'expérimentation, chacun des trois employés d'une agence de voyage était chargé d'organiser un voyage pour un client. Les employés possédaient des compétences différentes : l'un était spécialisé dans le transport aérien, le deuxième dans le transport ferroviaire et le dernier dans le transport routier. Aucun voyage ne pouvait se faire *via* un unique moyen de transport. Ainsi, chaque sujet¹ résolvait son problème et participait à la résolution des deux autres en interagissant par envois de mail.

Le modèle de planification

Ce modèle est fondé sur les notions de *phase*, *état d'esprit*, *stratégie*, *tactique*, *observation* et *personnalité* proposées par N. Chaignaud dans sa thèse (Chaignaud, 1996) : en fonction de sa personnalité et des observations qu'il a faites à un instant, un sujet se retrouve dans une phase avec un certain état d'esprit, planifiant selon des stratégies mises en œuvre par des tactiques.

Nous ne détaillons pas ce modèle ici, car nous nous intéressons à l'interaction lors de négociation.

Le modèle d'interaction

Dans ce mémoire nous utilisons le terme *performative* avec le sens qui lui est donné dans le domaine de la conception d'ACL, les langages de communication entre agents (Agent Communication Language). Il s'agit du type de message envoyé par un agent.

¹ Le terme sujet désigne les sujets humains ayant participé à l'expérimentation psychologique par opposition au terme agent qui désigne les agents logiciels implémentés.

Le modèle de l'interaction humaine

- utilise un modèle du discours, représenté par des automates temporisés, pour décrire la dynamique des conversations humaines,
- s'appuie sur la théorie des actes de langage pour modéliser les énoncés (Searle, 1969) (Vanderveken, 1988), à l'aide d'un ensemble de performatives appliquées à des états mentaux.

Le modèle de la dynamique conversationnelle

Le terme *échange* est utilisé ici dans le même sens que dans (Vanderveken, 1999). Un *échange* est composé d'une *intervention* ou un ensemble d'*interventions*, ayant une visée commune. Nous utiliserons le terme *message* dans le même sens que le terme *intervention*.

Pour pouvoir représenter la séquentialité et la temporalité des messages envoyés ainsi que conserver une souplesse pour simuler les capacités humaines d'interaction, les échanges sont représentés par des automates temporisés. Gérer la temporalité est utile quand un sujet prend trop de temps pour répondre à une demande. Il peut alors recevoir un ou plusieurs messages de relance et l'échange peut être considéré comme terminé par ses interlocuteurs sans qu'il y ait émission explicite d'une performative de terminaison.

Huit (4x2) automates ont été construits, un pour le locuteur et un pour l'interlocuteur de chaque type d'échange. Les échanges observés peuvent être classés selon : demande d'information, proposition d'information, envoi spontané d'information et traitement des erreurs. La première a une intention discursive primitive de type directif, la seconde d'engageant, la troisième de descriptif et la dernière de directif. La terminaison des échanges définit leur satisfaction.

Le modèle de l'énoncé

Les performatives observées proviennent des trois classes suivantes : les descriptifs, les directifs et les engageants.

Le modèle de l'énoncé repose sur le fait que le type de performative est lié au contenu du message :

- un **directif** s'applique à un désir du locuteur : il est utilisé quand le locuteur cherche à faire faire quelque chose à l'allocataire,
- un **descriptif** est appliqué à une croyance : il permet au locuteur de décrire la manière dont il perçoit le monde,
- un **engageant** s'applique à un désir de l'interlocuteur : le locuteur utilise un engageant pour signifier à l'allocataire qu'il s'engage à satisfaire un des désirs qu'il lui prête, désir qu'il transmet pour une éventuelle confirmation.

Les performatives sont décrites dans le tableau de la FIG. 2.

La sémantique² associée aux performatives est présentée sous la forme d'une règle générique de réduction précisant formellement les pré-conditions à l'envoi et à la réception de la performative ainsi que les actions à effectuer. Elle est liée aux automates temporisés et aux croyances de l'agent.

Le terme *prédictat* est utilisé dans le sens qu'on lui donne en logique (logique du premier ordre) : c'est un attribut qui peut prendre comme valeur *vrai* ou *faux*.

Descriptifs	
inform	S envoie spontanément à H une information. Dans le cadre du Problème de l'Agence de Voyage, S informe H de l'état d'avancement de son problème. <i>"Mission accomplie, merci encore."</i>
notUnderstood	S ne comprend pas un des précédents messages de H. <i>"Je n'ai jamais demandé cette réservation."</i>
reply	S répond à une demande de H. Il s'agit ici d'un envoi d'horaire ou de réservation. <i>"Non, il n'y a pas de gare au Chatelard."</i> <i>"Voilà qui est fait."</i>
thank	S remercie H. <i>"Merci beaucoup!"</i>
Directifs	
acceptProposal	S accepte une proposition faite par H. <i>"Oui, réserve-moi cet horaire."</i>
cancel	S demande à H de pas prendre en compte un de ses précédents messages. Il a été observé que les <i>cancels</i> n'étaient appliqués qu'à des messages de type <i>query</i> ou <i>reply</i> . <i>"Je n'ai réservé qu'une personne pour le trajet précédent, je corrige."</i>
query	S demande ou redemande à H de lui envoyer des informations. Il s'agit d'une (re)demande d'horaire ou de réservation. <i>"Combien coûte le trajet Paris-Annecy pour trois ?"</i> <i>"Pourrais-tu me réserver cet horaire ?"</i>
refine	S demande à H plus d'information pour pouvoir satisfaire sa demande. <i>"Lille aéro ou gare ?"</i> <i>"Il y a des cars mais pour quel horaire ?"</i>
refuseProposal	S refuse une proposition faite par H. Pour ce problème, c'est une demande de précision de lieu (entre gare ou aéroport) ou de temps. <i>"Non, ce sont les horaires qui ne collent pas."</i>
Engageants	
propose	S propose à H de lui fournir une information. Ici, il s'agit d'une proposition d'horaire ou de réservation. <i>"Je peux te proposer un trajet Orléans-Montpellier puis Montpellier-Toulouse."</i> <i>"Dois-je réserver ?"</i>

FIG. 2 - Liste de performatives dans le modèle BDIGGY (Pauchet, 2006, p.87)

² Le terme sémantique est utilisé ici dans le domaine informatique, et désigne la formalisation de la règle d'envoi/réception des performatives.

Voici par exemple la sémantique de la performative *query* :

$$\left[\begin{array}{l} pD(A_S S) \\ pB(!pMeans(S)) \\ !pB(S) \\ !pB(!S) \end{array} \right] \frac{Q_{ini}(ini, s2, s4, s7, s8) \xrightarrow{\text{Send(query)}} Q_{ini}(s1)}{aAdd(pB(pSent(M)); aUpdateTA(M))}$$

$$\frac{Q_{int}(ini, s1, s2, s4, s5) \xrightarrow{\text{Receive(query)}} Q_{int}(s1)}{aAdd(pB(pD(A_S S)); aUpdateTA(M))}$$

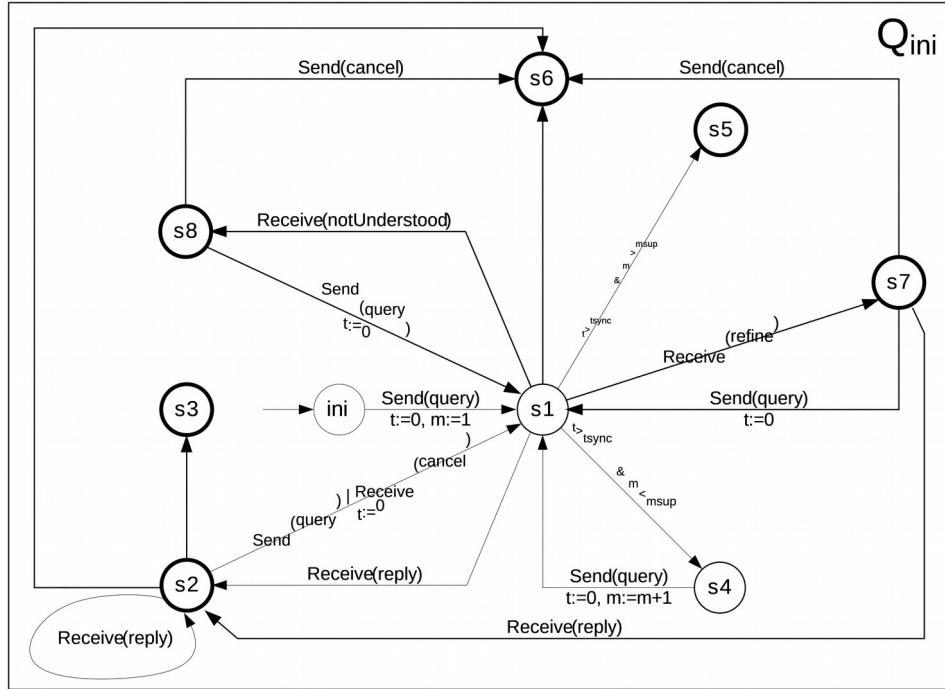


FIG. 3 - Automate de demande d'information Q_{ini} , tiré de (Pauchet, 2006, p.113)

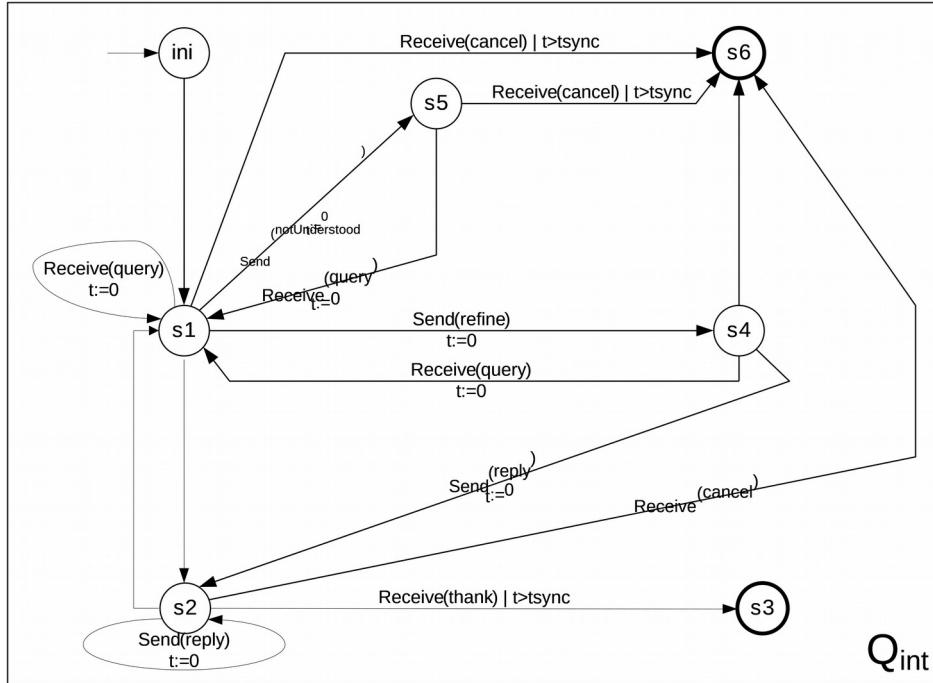


FIG. 4 - Automate de demande d'information Q_{int} , tiré de (Pauchet, 2006, p.113)

A_s est l'agent émetteur du message (*sender*). S (*stage*) est l'étape d'un voyage. $pMeans(S)$ est vrai si S est une étape dont le moyen de transport est celui dont l'agent a la charge. Les pD sont les prédictats sur les désirs (Desires), les pB les prédictats sur les croyances (Beliefs), s_1, \dots, s_8 sont les états des automates Q_{ini} et Q_{int} (FIG. 3 et 4) de demande d'information, ouverts respectivement par l'initiateur de la demande d'information (Q_{ini}), et par son interlocuteur (Q_{int}).

Les pré-conditions de la performative *query* signifient :

- $pD(A_s S)$: l'agent a le désir de connaître S ,
- $pB(!pMeans(S))$: l'agent a la connaissance qu'il n'a pas le moyen de transport correspondant à l'étape S ,
- $!pB(S)$: l'agent n'a pas la connaissance que S est valide pour son voyage,
- $!pB(!S)$: l'agent n'a pas la connaissance que S n'est pas valide pour son voyage.

Les agents réalisent systématiquement l'action $aUpdateTA(M)$, (M étant le message reçu ou envoyé), qui permet de mettre à jour les automates ($TA = \text{Temporised Automaton}$).

Tout envoi de message génère l'action $aAdd(pB(pSent(M)))$ pour que les agents gardent en mémoire les messages envoyés.

Contrairement à la réception d'un message, aucune action autre que la mise à jour des automates et la mémorisation de l'envoi n'est effectuée à l'envoi d'un message. Un envoi de message est une simple action élémentaire externe alors qu'une réception de message nécessite la mise à jour des états mentaux de l'agent.

$aAdd(pB(pD(A_s S)))$ signifie qu'à la réception du message, l'agent destinataire ajoute à ses croyances le désir de l'agent émetteur A_s de connaître S .

L'architecture BDIGGY

BDIGGY (voir FIG.5) est structuré autour d'une architecture BDI qui comprend :

- un **module de perception** qui observe l'environnement et génère des croyances,
- un **module de planification** humaine, IGGY, qui construit des plans abstraits de façon opportuniste,
- un **interpréteur de plans** qui interprète les épisodes d'IGGY et effectue les actions intentionnées,
- et un **module de communication** qui permet à l'agent d'interagir.

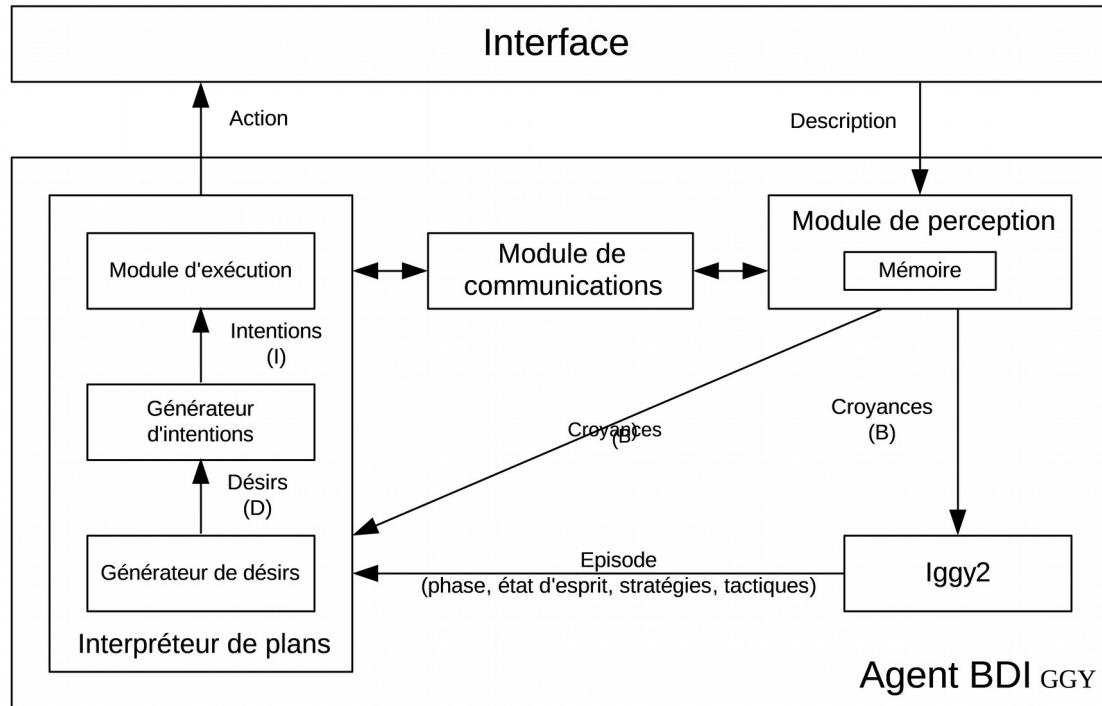


FIG. 5 - Flux de l'architecture BDIGGY (Pauchet, 2006, p.137)

La validation des modèles et de l'architecture d'agent a été réalisée par des experts en comparant manuellement un ensemble de traces réelles et un ensemble de traces générées

artificiellement, selon un test *à-la-Turing*. Le principal résultat est qu'il n'a pas été possible de re-séparer de façon significative les 2 classes de traces.

1.3. La négociation

Dans le problème de l'agence de voyage, les agents ont chacun un problème différent à résoudre : l'organisation d'un voyage. Ils coopèrent car chacun a besoin des deux autres. Par exemple, l'agent aérien demandera un horaire de train à l'agent ferroviaire. Ce dernier s'occupera alors de trouver et transmettre l'horaire, utile pour le problème de l'agent aérien, mais n'ayant aucun rapport avec le sien. Nous avons voulu ajouter des situations où les objectifs des agents sont interdépendants, les faire travailler sur le même problème, et qu'ils se retrouvent dans l'obligation de négocier pour parvenir à leur objectif.

Négocier vient du latin *negotiari*, « faire du commerce ». La négociation est la « *série d'entretiens, d'échanges de vues, de démarches qu'on entreprend pour parvenir à un accord, pour conclure une affaire* » (s.v. négociation Le Petit Robert). Elle est utilisée dans des situations très variées : par téléphone dans une salle de marchés, lors d'un entretien avec son patron pour discuter d'une augmentation de salaire, ou bien encore par mail avec un potentiel acheteur d'un objet que l'on aurait déposé sur un site d'enchères en ligne.

On trouve dans la littérature, de nombreuses définitions de la négociation. Toutes s'accordent sur le fait qu'il s'agit d'un processus. Commerciale, syndicale ou politique, la négociation est un processus humain visant à trouver un accord entre plusieurs parties. Dans le monde animal, le rapport de force supprime toute forme possible de négociation.

Notre recherche a donc pour objectif de s'appuyer sur un processus de négociation entre humains, pour modéliser les interactions d'agents logiciels qui négocient.

Nous reprendrons la définition de Ferber dans (Ferber, 1995, p.67), pour définir l'interaction entre des agents :

« *Une interaction est une mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques. Les interactions s'expriment ainsi à partir d'une série d'actions dont les conséquences exercent en retour une influence sur le comportement futur des agents.* »

Nous nous intéressons à la formalisation des messages et de la dynamique des échanges entre les sujets, sans traiter la prise de décision de ces derniers. Nous souhaitons dans cette première série d'expérimentations, n'observer que *les symptômes*. Il s'agit, en partant de l'observation du processus de négociation entre humains, de dégager un protocole de négociation pour agents logiciels. L'idée étant, pour reprendre l'expression de Maudet dans (Maudet, 2001, p.12), de « *tendre vers le plus humain* ».

2. La négociation dans les SMA

L'étude de la négociation est abordée par différentes disciplines : l'économie, les sciences des organisations, les sciences de l'aide à la décision, la psychologie, la sociologie. Il existe également une science dite de la négociation. La négociation est abordée depuis plus récemment par l'Intelligence Artificielle Distribuée.

Après une discussion sur différentes définitions de la négociation dans le domaine des SMA, nous proposerons une définition. Nous présenterons ensuite les différentes problématiques de modélisation de la négociation, la communication entre agents, indispensable pour la négociation, et les protocoles de négociation les plus connus.

2.1. La notion de négociation dans les SMA

« Ce qui s'énonce bien, se conçoit clairement. »

C.Brassac, S.Pesty
La pelouse fourmilière. De la coaction à la coopération.
Actes JFIADSMA'96, p. 251-263.

Bussmann et Müller dans (Bussmann et Müller, 1992) propose la définition suivante de la négociation :

« La négociation est un processus de communication d'un groupe d'agents, dans le but d'aboutir à un accord mutuel sur un sujet. »³

Cette définition met en avant la communication, indispensable pour envisager une négociation. Elle manque cependant de précision, et pourrait correspondre à une délibération. Les membres d'un conseil d'administration par exemple, peuvent être amenés à chercher un accord sur un sujet. Ils s'apporteront des informations mutuellement, dans un processus de communication, sans forcément avoir besoin de négocier quoi que soit.

Rahwan *et al.* dans (Rahwan *et al.*, 2007) adaptent la définition de (Walton et Krabbe, 1995) en philosophie de l'argumentation :

« La négociation est une forme d'interaction dans laquelle un groupe d'agents, avec des intérêts conflictuels et un désir de coopérer, essaie d'aboutir à un accord mutuel acceptable sur la division de ressources insuffisantes.»⁴

³ “negotiation is the communication process of a group of agents in order to reach a mutually accepted agreement on some matter.”

⁴ “Negotiation is a form of interaction in which a group of agents, with conflicting interests and a desire to cooperate, try to reach a mutually acceptable agreement on the division of scarce resources.”

Cette définition correspond mieux à la négociation. Cependant nous pensons que même pour des agents logiciels, la négociation ne porte pas forcément sur le partage de ressources insuffisantes, ni de ressources tout court. Si on peut voir un créneau horaire comme une ressource de temps (pour la prise de rendez-vous par exemple), on peut également négocier des interprétations (voir partie 2.2.7.2. Le protocole de Sian), des responsabilités, le traitement d'une tâche, le contenu d'une loi.

Ferber définit également la négociation comme une forme d'interaction. Dans (Ferber, 1995, p.59), il propose la formule suivante :

$$\text{coopération} = \text{collaboration} + \text{coordination d'actions} + \text{résolution de conflits}$$

La **négociation** est l'une des solutions, avec l'arbitrage, pour résoudre ou éviter les conflits parmi les agents. L'arbitrage, à partir de la définition de règles de comportement (comme la justice pour la société humaine), impose des contraintes pour limiter les conflits entre agents au sein d'un système. La négociation, elle, laisse les agents régler eux-mêmes le conflit dans une recherche d'accord.

La **collaboration** consiste à travailler à plusieurs sur un projet, une tâche commune. C'est l'ensemble des techniques permettant à des agents de se répartir des tâches, des informations et des ressources, de manière à réaliser une œuvre commune. Résoudre un problème de collaboration consiste donc à répondre à la question « *qui fait quoi ?* » par rapport à un travail donné (Ferber, 1995, p.89).

Ferber cite (Malone, 1998) pour définir la **coordination** d'actions : c'est l'ensemble des activités supplémentaires qu'il est nécessaire d'accomplir dans un environnement multi-agent et qu'un seul agent poursuivant les mêmes buts n'accomplirait pas. Il explique que le problème de la **coopération** peut se ramener à « *qui fait quoi, de quelle manière et avec qui* », pour résoudre les sous-problèmes que constituent la collaboration par répartition de tâches, la coordination d'actions et la résolution de conflits. La négociation est ainsi vue comme une forme de coopération.

Comme l'explique (Brassac et Pesty, 1996), les emplois respectifs de ces concepts sont « *peu stabilisés dans le domaine* ». Ce qui amène des débats sur leur utilisation (Brassac et Pesty, 1996) (Lenay, 1996) (Ferber, 1996).

Pour (Smith, 1980, p.1105), « *la négociation comprend 4 points importants : 1) c'est un processus local n'impliquant pas de contrôle centralisé, 2) il y a un échange bilatéral de l'information 3) chaque partie prenant part à la négociation, évalue les informations de son propre point de vue, et 4) l'accord final est obtenu par sélection mutuelle.* »⁵

⁵ “For our purposes, negotiation has four important components: 1) it is a local process that does not involve centralized control, 2) there is a two-way exchange of information, 3) each party to the negotiation evaluates the information from its own perspective, and 4) final agreement is achieved by mutual selection.”

Nous rapprochons cette définition de celle d'une délibération, dans laquelle les parties s'apportent des informations et qu'il n'y a pas forcément de divergence dans leurs préférences. Dans une négociation, on s'échange plus que des informations. D'autre part nous pensons qu'il peut y avoir négociation sans que celle-ci n'aboutisse nécessairement à un accord. Les protagonistes qui ne sont pas parvenus à trouver d'entente ont néanmoins négocié.

La définition de (Verrons, 2004) met l'accent sur le fait que la négociation est une discussion. On peut parler de négociation lorsque les acheteurs peuvent discuter l'offre. Dans le *take it or leave it offer* ou *offre à prendre ou à laisser*, un des négociateurs fait une offre, son interlocuteur doit choisir de l'accepter ou la refuser. Ce n'est pas de la négociation, car il n'y a pas discussion. Il y a communication, mais pas d'échange en vue d'aboutir à un accord. Le vendeur fixe son prix et l'acheteur accepte ou refuse.

Elle propose cette définition, qui précise également que le but de la négociation est de satisfaire au mieux tous les participants :

« Il y a négociation lorsqu'il y a une discussion, des propositions entre les protagonistes et lorsque l'accord final satisfait au mieux tous les participants. »

Le célèbre exemple de la prise de rendez-vous est une situation qui correspond à cette définition. Il y a des propositions, et souvent l'accord final satisfait au mieux les participants (le patient qui téléphone et la secrétaire du dentiste). Mais une prise de rendez-vous n'est-elle pas une simple coordination, une *recherche de correspondance*, un *appariement*, entre deux emplois du temps ? Le but est de trouver un créneau horaire commun. La négociation suppose, implique, la discussion. Quelle est la limite au-delà de laquelle les participants passent-ils de la simple coordination, à de la négociation ? Nous pensons que c'est lorsque les participants prennent conscience que leurs préférences sont incompatibles et qu'ils entament une démarche de recherche d'accord. Les différentes interventions vont modifier les préférences de chacun et permettre de trouver une entente. Une prise de rendez-vous avec le secrétariat d'un dentiste qui ne peut modifier l'emploi du temps du dentiste, n'est pas une négociation. Il n'y a pas possibilité de faire évoluer ses *préférences*, et les choix possibles sont les intersections des deux emplois du temps.

La même réflexion peut être menée sur une situation en carrefour, où deux véhicules doivent se croiser. Ils doivent adapter leurs actions de manière à ne pas entrer en collision. Considèrera-t-on qu'ils interagissent pour se coordonner ou qu'ils négocient dans le but d'éviter la collision (le conflit) ? La négociation serait-elle une interprétation de l'observateur extérieur, ou une action intentionnelle des automobilistes ? Cette question rejoint celles de Brassac et Pesty dans (Brassac et Pesty, 1996, p. 261) :

« En quoi une fourmi, constituante d'une fourmilière, négocie-t-elle avec une (ou des) congénère(s) ? N'est-elle pas trop réactive pour être capable de négocier ? Ou bien, possède-t-elle suffisamment d'intentionnalité pour que le signal (versus message) qu'elle émet serve de support à la négociation ? Existe-t-il une négociation qui soit non intentionnelle ? »

Nous pensons que la fourmi a les mêmes intérêts et objectifs que ses congénères de la même fourmilière (entre autres construire la fourmilière et la maintenir en état), et qu'il n'y a donc

pas de négociation : elle n'est pas nécessaire car les fourmis oeuvrent dans le même sens. On parlera de coordination. Dans le cas où l'autre fourmi est *d'un autre clan*, le rapport entre les deux fourmis est un rapport de force physique. La négociation est de notre point de vue, profondément humaine. Utiliser cette notion pour des agents logiciels est donc d'emblée sujet à débat.

Outre le fait que la fourmi est plutôt *dans l'instant* (et qu'il est difficile d'envisager des promesses par exemple, ou un accord dont les conséquences sont futures), il se pose la question de la part d'individualisme qu'elle possède. Nous déplaçons donc la question de l'intentionnalité de la fourmi, à celle de son ipséité :

La fourmi possède-t-elle suffisamment d'individualisme pour négocier ? (On définira l'individualisme comme la tendance d'un individu à s'affirmer indépendamment des autres).

La fourmi *joue en équipe*. Elle fait partie d'un groupe où l'individu, seul, n'existe pas. Il n'existe que par rapport à l'organisation dont il fait partie (en l'occurrence la fourmilière). Ce qui n'est pas forcément le cas de l'humain.

Pour la fourmi, il n'y a jamais incompatibilité entre intérêt individuel et intérêt collectif. C'est ce qui amène certains chercheurs en SMA à utiliser la théorie des jeux (qui étudie cette incompatibilité entre intérêt individuel et collectif) dans leurs recherches en négociation (voir partie 2.2.5. Théorie des jeux et IAD).

Pour en revenir aux automobilistes, la négociation implique des interactions qui ne sont pas forcément verbales (on peut négocier avec des gestes, par écrit, par des regards). Si l'on pousse cet exemple à l'extrême, les automobilistes ont donc la possibilité de négocier, s'ils veulent chacun passer en premier. L'objet de la négociation serait *celui qui passera en premier*. Une menace de l'un peut par exemple influencer la décision de l'autre. Il peut y avoir une séquence d'échanges par gestes. Là encore se pose la question de la limite à partir de laquelle on bascule dans la négociation.

Nous pensons, comme dans l'exemple de la prise de rendez-vous, que c'est en observant les échanges entre les protagonistes que l'on déterminera la nature de la discussion. Lorsqu'il apparaît que les intérêts sont conflictuels, et qu'il y a une volonté commune de trouver un accord, on pourra parler de négociation. Dans ces deux exemples (la prise de rendez-vous et le cas extrême du croisement en carrefour qui mène au conflit), la négociation serait un moyen de se coordonner. C'est le point de vue de (Malone et Crowston, 1994) (Lizotte et Chaib-Draa, 1995), pour qui la coordination subsume les autres mécanismes d'interdépendance.

(Chang et Woo, 1994) met l'accent sur le caractère de lutte que présente une négociation : ils appuient leur protocole entre autre sur le *Struggle model* de (Ballmer et Brennenstuhl, 1981) et le cite p.367 (voir partie 2.2.7.3. Le protocole SANP) :

« La lutte commence quand une des parties impliquées fait une revendication, et que l'autre n'est pas d'accord. Les deux parties s'engagent ensuite dans une bataille

verbale compétitive : elles débattent l'une l'autre, ce qui aboutit soit à la victoire de l'une et la défaite de l'autre, soit à un compromis. »⁶

La négociation commencerait à la constatation du désaccord entre les parties, dû à des préférences différentes. Nous préférions parler de *non accord*, le terme étant *mois fort* et plus représentatif des situations de négociation cordiale, comme les prises de rendez-vous. Entre humains, cela peut alors entraîner des échanges animés, qualifiés ici de *bataille verbale*. On retrouve cet aspect de la négociation dans *Saint-Germain ou la négociation* de Francis Walder, qui emploie « *escrime verbale* » pour qualifier les échanges (Walder, 1958, p.70). La définition de Chang et Woo limite néanmoins le résultat de la négociation : les auteurs n'envisagent pas le cas où elle n'aboutit pas.

Par rapport à ces différents points, nous proposons de définir la négociation de la façon suivante :

La négociation est un processus de recherche d'accord sur un objet, entre plusieurs parties ayant des préférences différentes mais susceptibles d'évoluer, et qui implique des interactions permettant de modifier à la fois la plage de possibilités et les préférences de chacun sur les possibilités.

La négociation naît donc de préférences différentes (non accord) et n'aboutit pas forcément à un accord. Le résultat est une prise de décision commune des parties sur les modalités de l'accord, si la négociation aboutit ; ou un désaccord (voir un conflit), si elle échoue.

Notre définition exclut notamment :

- **les systèmes de vote** : il y a bien une recherche d'accord mais il n'y a pas d'interaction entre les protagonistes, on obtient un résultat gagnant-perdant(s), le vote a permis de *trancher*,
- **le *take it or leave it offer*** : il y a interaction, mais aucune évolution possible des préférences de chacun, il s'agit d'un choix binaire à effectuer,
- **certaines formes d'enchères**, qui ne laissent pas la place à la discussion (voir partie 2.2.7.4. Les protocoles dans les systèmes d'enchères),
- **les situations de type dilemme du prisonnier** : il n'y a pas d'interaction, et les choix possibles sont contenus à l'avance dans une matrice de gains. Il s'agit de situations d'interdépendance dans un cadre déterminé à l'avance. Nous aborderons cependant ce cadre (voir partie 2.2.4. L'approche théorie des jeux), car il met en évidence l'incompatibilité entre intérêt individuel et intérêt collectif. Il peut être utilisé à titre individuel par un protagoniste, pour raisonner durant une négociation, mais il ne s'agit pas de situation de négociation,

⁶ “The struggle starts when one party makes a claim and the other party disagrees. Then, both parties engage in a competitive verbal dispute : they argue with each other, which may result in either the victory of one and defeat of the other or in compromise.”

- la **prise de rendez-vous**, si celle-ci est un simple appariement sur des plages horaires de plusieurs emplois du temps, et qu'il n'y a pas de possibilité pour tous les protagonistes de faire évoluer leur emploi du temps (comme le secrétariat d'un dentiste qui ne pourrait déplacer les rendez-vous déjà pris par exemple),
- la **délibération**, qui, selon le contexte :
 - est bien une discussion entre les parties concernées par l'objet discuté (par exemple lors d'un conseil d'administration), mais qui consiste à effectuer un choix. La délibération est parfois précédée ou entrecoupée de phases de négociation, mais nous considérons qu'elle n'est pas en elle-même de la négociation,
 - aboutit à un verdict (d'un juge par exemple). Les parties concernées s'en remettent à une autre pour trancher, la décision n'est pas forcément une de celles évoquées durant les discussions qui ont précédé. On rapprochera ces situations de l'arbitrage. En SMA, on peut citer le système PERSUADER de Sycara, qui met en jeu un médiateur. Voir par exemple (Sycara, 1989).

Il manque dans ces situations que nous ne prenons pas en compte, le processus interactif de recherche des possibilités que constitue la négociation. Lorsque l'on négocie, on ne fait pas un choix dans un ensemble fini des possibilités. On construit conjointement, et au fur et à mesure des échanges, ces possibilités. Le jeu de la négociation est justement de pouvoir faire évoluer la plage de possibilités, et les préférences de ses interlocuteurs sur ces possibilités, pour tenter d'aboutir à un accord.

2.2. Modélisation de la négociation dans les SMA

De nombreux travaux de recherche ont été effectués pour modéliser la négociation entre agents logiciels et entre agents logiciels et sujets humains. Dans la communauté SMA, on distingue souvent trois approches (Jennings *et al.*, 2000, p.28), (Rahwan *et al.*, 2007, p.3) :

- l'approche mathématique avec des analyses en théorie des jeux,
- la négociation fondée sur des heuristiques,
- et la négociation fondée sur l'argumentation.

Cependant, les travaux existants ont des problématiques bien différentes et il est difficile de les classifier en trois parties. Un système multi-agent d'allocation de ressources peut, par exemple, utiliser un protocole d'encheres, pour décider de l'emplacement de stockage de nouvelles données. Les différents serveurs faisant leurs offres en fonction de l'utilité qu'elles représentent pour eux (Schwartz et Kraus, 1998). Ou bien encore un agent qui négocie *par argumentation*, peut utiliser un modèle de raisonnement en théorie des jeux (voir partie 2.2.4. L'approche théorie des jeux) ou fondé sur la logique (Kraus *et al.*, 1998).

Certains travaux s'intéressent à la modélisation des préférences (Bichler et Lee, 2003), et aux mécanismes de négociation. On peut citer les recherches au LAMSADE, par exemple (Bellotta *et al.*, 2004). Certains s'attachent aux applications dans le commerce électronique, comme (Chavez et Maes, 1996), l'allocation de ressources, les fonctions d'utilité et la modélisation des préférences, par exemple (Estivie, 2006).

Dans le commerce électronique, les applications possibles pour ces agents logiciels sont :

- l'assistance des utilisateurs pour des enchères électroniques,
- le e-sourcing, qui a pour but d'optimiser l'amont de l'achat en standardisant et automatisant le plus possible les processus d'approvisionnement pour les entreprises (de produits ou de services). Ces techniques permettent de réaliser des économies qui proviennent principalement de la diminution des coûts de négociation et de transaction avec les fournisseurs.

Des recherches concernent également la formalisation des actes de langage, la modélisation des arguments et des engagements, par exemple (Bentahar *et al.*, 2003).

Selon les travaux, on trouve des types de négociations différents :

- les négociations multi-attributs, où il faut négocier un ensemble d'attributs pour le même objet (comme le prix, la livraison, etc),
- les négociations multi-niveaux, lorsqu'une négociation est décomposée en plusieurs *sous-négociations*. Par exemple, pour la prise de rendez-vous, on peut décomposer le processus en la recherche de la semaine, du jour, puis de l'heure, etc.,
- les négociations combinées, qui consistent à négocier des objets interdépendants avec des vendeurs différents. Il faut obtenir l'ensemble des objets (les composants d'une chaîne hi-fi par exemple) et avoir un mécanisme de liaison entre les différentes négociations. Si l'ensemble des objets ne peut être obtenu, aucun ne doit l'être.

(Zeng et Sycara, 1997) s'intéressent à l'apprentissage entre les agents, (Kraus, 1995) à la notion de temps pendant les négociations. La négociation par argumentation est traitée par exemple dans (Kraus *et al.*, 1998) et (Schroeder, 1999).

D'autres travaux, dans un cadre de génie logiciel, visent à fournir des plates-formes génériques. Elles permettent d'implémenter différents types de négociation. On peut citer GNP (Bartolini *et al.*, 2002) et GENCA (Verrons, 2004). Nous reviendrons sur la plate-forme GENCA dans la partie 2.2.7.5. Le protocole dans GENCA.

Après avoir détaillé les différents éléments de la négociation, et les deux types de négociation, nous présentons le domaine de l'aide à la concertation et à la négociation, et les travaux en théorie des jeux. Ces deux approches, bien que différentes, ont un lien avec la nôtre : la première, dans sa démarche de modélisation qui fait participer l'humain, et la seconde dans le fait qu'elle traite des situations d'intérêts conflictuels. Nous nous focalisons ensuite sur les protocoles de négociation existants.

2.2.1. Les éléments de la négociation

Müller dans (Müller, 1996), cité dans (Le Bars, 2003), fait la distinction entre trois éléments dans la négociation (voir FIG. 6) :

1. les langages, qui comportent des primitives de communication pour la négociation, leurs sémantiques, et leurs usages dans les protocoles. Les primitives concernent les envois et réceptions de messages et intègrent des actes illocutoires. La structure de l'objet de la négociation fait appel à un langage de description de l'objet. Le protocole spécifie les séquences possibles d'actions et les conditions suivant lesquelles une requête peut être effectuée,
2. les processus de négociation qui peuvent être, soit des procédures pré-établies, soit des comportements. Le processus concerne la proposition de solutions, l'analyse et la révision des solutions préférables,

- la décision individuelle qui peut être prise en fonction de critères d'utilité, de stratégie, de systèmes de préférences, d'une fonction de comparaison et de mise en correspondance (*matching*).

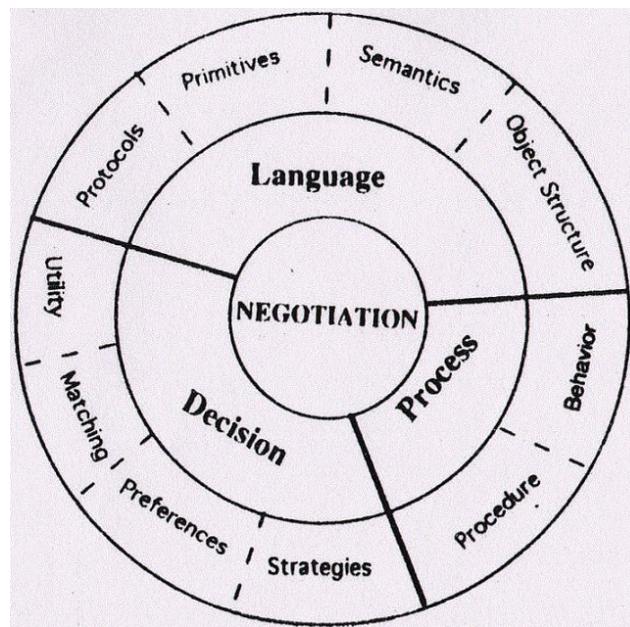


FIG. 6 - Les différents éléments d'une négociation et leurs structures (Müller, 1996)

Dans le cadre de ce stage, nous nous intéressons aux deux premiers points, l'étude du raisonnement des agents, dépendant du domaine, sera traitée par la suite.

2.2.2. Les deux types de négociation

Dans la littérature, on distingue deux grands types de négociation, en fonction de la conduite des négociateurs :

- la **négociation intégrative** (aussi parfois appelée **négociation coopérative** ou **concertative**) conduit généralement à un accord dans lequel les deux parties s'estiment gagnantes (gagnant-gagnant). Les agents prennent en compte les intérêts de chacun dans leur décision : ils poursuivent un but commun et la négociation leur permet de se coordonner. Il s'agit d'une recherche distribuée dans un espace de solutions.
- la **négociation distributive** (aussi parfois appelée **négociation compétitive** ou **conflictuelle**) permet de résoudre un conflit. Chaque agent négociateur poursuit des buts individuels et cherche à être le gagnant de l'échange. Chacun campe sur ses positions et la négociation est vue comme une confrontation. L'accord risque d'être gagnant-perdant, voire perdant-perdant.

Simos (Simos, 1990) propose de situer ces deux types de négociation en en définissant les limites (voir FIG. 7) :

- l'**affrontement** : seul le rapport de force joue, il y a recherche de la soumission totale de l'adversaire,
- l'**entente** : les interlocuteurs oublient leurs intérêts propres et se retrouvent en situation de résolution de problèmes, c'est-à-dire vont rechercher une solution optimale (du point de vue mathématique).

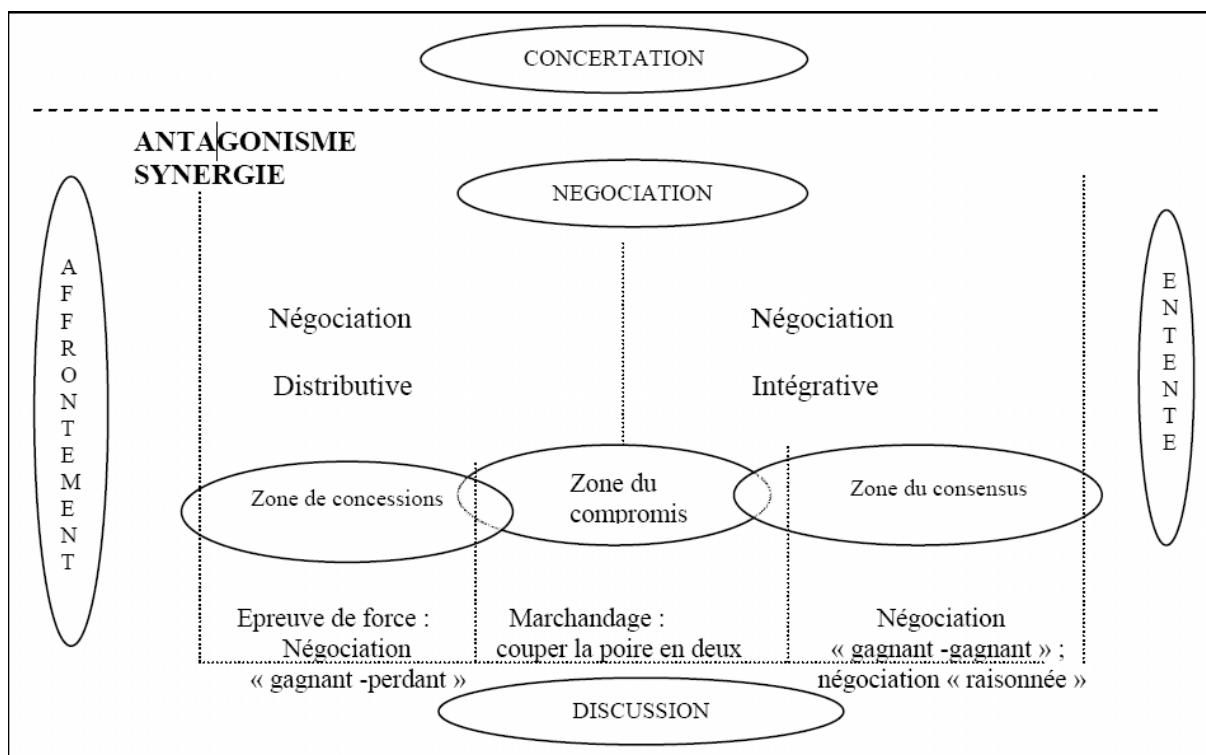


FIG. 7 - Les différents types de négociation (Simos, 1990)

Dans notre travail, nous ne cherchons pas à modéliser l'un ou l'autre type de négociation. Nous souhaitons permettre aux agents de négocier, et le type de négociation qui résultera des interactions entre agents pourra être qualifié *a posteriori* de la négociation.

Tout comme la discussion sur la coopération entre agents a trait à l'intentionnalité des agents par rapport au résultat obtenu (coopération entre agents), l'aspect type de négociation (intégrative ou distributive) a trait à la propension à tenir compte (ou non) pour un agent, des préférences des autres agents. Dans notre modèle c'est donc dans les traits de caractères d'un agent, contenu dans la personnalité⁷ que s'exprimera cet aspect de la négociation. (Guicherd, 2005) a par ailleurs étudié le rôle de la personnalité, dans les rapports durant la négociation.

⁷ Voir le modèle IGGY (Chaignaud, 1996).

2.2.3. L'aide à la concertation et à la négociation

Une autre approche vise l'aide à la concertation et à la négociation pour les humains. On peut citer les études réalisées au CEMAGREF (Centre National du Machinisme Agricole, du Génie Rural, des Eaux et Forêts), et au CIRAD (Centre de coopération Internationale en Recherche Agronomique pour le Développement). Par exemple, dans (Le Bars, 2003) l'objectif est de fournir aux négociateurs, dans le cadre de la gestion collective de l'eau, un instrument permettant de tester les conséquences d'une réglementation. Les négociateurs ont alors toute liberté pour modifier cette réglementation, en tester les conséquences, afin d'aboutir à un compromis acceptable.

Dans le cadre des recherches au CIRAD (Centre de coopération internationale en Recherche Agronomique pour le Développement), plusieurs modèles d'aide à la décision collective ont été proposés. On peut citer par exemple (Bousquet *et al.*, 2002). Un des principaux buts de ces modèles, conçus sur la plate-forme multi-agent CORMAS (CCommon-pool Resources and Multi-Agent Systems) (Bousquet *et al.*, 1998) est d'utiliser les simulations multi-agents en tant qu'outils d'accompagnement et d'aide à la concertation, dans des situations d'interactions homme-machine.

(Ruffez, 2003) dans le cadre d'un stage de recherche dans l'équipe GREEN (Gestion des Ressources renouvelables, Environnement) avait pour objectif d'améliorer SYLVOPASTJEU, un jeu de rôle hybride permettant à des acteurs réels de communiquer avec des agents virtuels.

Il s'inscrit dans le projet SYLVOPAST (Etienne, 2003), axé sur le Sylvopastoralisme et la prévention des incendies en région méditerranéenne. Il s'agit d'une simulation sous forme ludique de la gestion de la forêt, prenant en compte les aspects écologiques (évolution de la forêt) et l'action des principaux acteurs impliqués : les forestiers et les bergers, leurs buts et leurs stratégies d'utilisation de la forêt sont différents. Les forestiers font de la prévention d'incendie, de la production de bois ou recherchent la biodiversité de la forêt, les bergers ont eux pour but de faire pâturer leur troupeau dans les meilleures conditions possibles.

Le partage des ressources forestières créant des conflits entre les acteurs, la négociation est un aspect majeur de la gestion de ressources. La simulation se déroule sur une dizaine d'années. A la fin de chaque année, le berger et le forestier négocient une série de travaux à effectuer en fonction de leur budget. Ils peuvent pour une parcelle de terrain donnée, semer de l'herbe, débroussailler, replanter des arbres. Seul le forestier est habilité à prendre la décision finale de choisir une action. Le berger a la possibilité de financer tout ou partie des travaux, mais c'est le forestier qui réalise concrètement les actions. Le protocole de négociation proposé est basé sur le protocole Sian, que nous présentons dans la partie 2.2.7.2. Le protocole de Sian.

Cette approche utilise des jeux de rôle pour améliorer une simulation multi-agents, et cette même simulation pour améliorer les négociations entre sujets humains.

Une fois un modèle validé, le jeu de rôle correspondant est créé. Les acteurs remplacent les agents, et jouent une série de parties. A l'issue de ces séries, les comportements sont analysés et réinjectés dans le modèle original. Un jeu de rôle plus précis dans la modélisation est ensuite recréé, une série de parties rejouée, générant une nouvelle compréhension des comportements, et ainsi de suite.

Cette approche est la plus proche de la nôtre, dans le sens où elle tente de révéler les comportements et les relations entre des sujets humains, pour créer des modèles.

Dans le cadre de sa thèse, N. Franchesquin (Franchesquin, 2001) visait à modéliser la négociation entre les différents acteurs de l'eau, et à fournir un outil d'aide à la gestion du système hydraulique de la Camargue, centré sur le respect des règles de préservation des écosystèmes du parc. La négociation porte sur la fixation d'un contrat d'objectifs dans le système Vaccarès, pour le couple (niveau d'eau, salinité) et par périodes (rizicole et non rizicole). Ces objectifs étant déterminés, le modèle hydrologique va aider à la définition de règles de gestion pour les respecter. Le protocole de communication utilisé est également issu de Sian.

De nombreux travaux de recherche portent sur l'aide à la décision, on pourra se reporter à (Le Bars, 2003) pour un état de l'art plus complet.

2.2.4. L'approche théorie des jeux

La théorie des jeux est l'approche mathématique de l'étude des conflits d'intérêts. Elle a connu un essor important après la publication, en 1944, de l'ouvrage *Théorie des jeux et du comportement économique*⁸ de (Von Neumann et Morgenstern, 1944). Von Neumann, à l'époque mathématicien à l'Institute for Advanced Study et Morgenstern, économiste à l'université de Princeton, sont considérés comme les fondateurs de la théorie des jeux, qui n'a depuis jamais cessé d'évoluer. Le mathématicien John Nash a également été très actif. Les outils de la théorie des jeux sont aujourd'hui utilisés en économie, mais également dans des sciences comme la sociologie, la psychologie, la biologie, les sciences politiques, l'informatique, dans lesquelles l'étude des situations de conflits est pertinente. Nous allons rapidement présenter quelques concepts de la théorie des jeux.

Equilibre de Nash et Pareto-Optimalité

Le dilemme du prisonnier est une célèbre illustration de l'incompatibilité entre l'intérêt individuel et l'intérêt collectif :

Deux individus sont arrêtés par la police et emprisonnés dans des cellules séparées. La police fait à chacun la même proposition : « *Tu as le choix entre dénoncer ton complice ou rester muet. Si tu le dénonces et qu'il te*

⁸ *Theory of Games and Economic Behavior*

dénonce aussi, vous aurez chacun 2 ans de prison. Si tu le dénonces et que ton complice te couvre, tu sortiras et lui prendra 3 ans de prison. Si vous vous couvrez mutuellement, vous prendrez tous les deux 1 an. »

Dans cette situation, il est clair que si les deux restent muets, ils s'en tireront globalement mieux que si l'un des deux dénonce l'autre. Mais l'un peut être tenté de s'en tirer encore mieux en dénonçant son complice. Le dilemme est *faut-il accepter de couvrir son complice (C comme coopérer) ou le trahir (D comme dénoncer)* ? Il existe des versions itérées du dilemme où un joueur peut se souvenir du comportement de l'autre à son égard et développer une stratégie en rapport (par exemple, si l'adversaire d'un joueur ne coopère jamais, son intérêt sera de ne pas coopérer non plus). De nombreux travaux ont étudié les stratégies et leur confrontation, par exemple (Axelrod, 1981), (Axelrod, 1997), (Beaufils, 2000). Des versions étendues du dilemme (n prisonniers) permettent de faire des parallèles avec l'étude des comportements en sciences sociales, voir par exemple (Chavaliarias, 2005). Pour la version simple du dilemme, les résultats sont présentés dans la matrice de la FIG. 8.

	C	D
C	1 an, 1 an	3 ans, 0 an
D	0 an, 3 ans	2 ans, 2 ans

FIG. 8 - Matrice de gain du dilemme du prisonnier

Nash a été le premier à présenter une définition d'une stratégie optimale pour un jeu à plusieurs joueurs, dite **équilibre de Nash**. L'équilibre de Nash caractérise la rationalité individuelle. Il s'agit d'une combinaison stratégique où **chaque joueur joue sa meilleure réponse quelque soit le choix de l'adversaire**. C'est une stratégie de *non regret*. Le prisonnier qui dénonce prend au pire 2 ans de prison, et au mieux sort de prison. Pour le dilemme du prisonnier simple, D/D est la seule combinaison stratégique en équilibre de Nash.

Pareto, sociologue et économiste, a défini l'**optimum de Pareto** qui caractérise la rationalité collective. C'est une situation dans laquelle **un individu ne peut améliorer sa situation sans détériorer celle d'un autre individu**. Les trois issues C/C, C/D, D/C sont pareto-optimales, car dans chacune, il est impossible d'augmenter le gain d'un des joueurs sans baisser celui de l'autre. D/D n'est pas pareto-optimale car il existe une issue préférée par les 2 prisonniers : C/C.

2.2.5. Théorie des jeux et IAD

Des chercheurs comme Rosenschein, Kraus, Sandholm, analysent les interactions entre agents d'un SMA d'un point de vue théorie des jeux.

Dans les travaux de Rosenschein et Zlotkin (Rosenschein, 1994), les agents sont omniscients et sont conçus pour maximiser leur utilité. Ils ont tous les mêmes capacités et des ressources illimitées pour atteindre leurs buts. Les actions des agents ne sont pas dépendantes des éventuelles négociations passées, et leurs stratégies sont publiques et visibles des autres agents. Comme l'explique (Chaib-draa *et al.*, 2001), les négociations modélisées dans ce cadre s'appliquent mal à des problèmes du monde réel. Les négociations entre humains se déroulent dans l'incertitude, mettent en jeu des critères beaucoup plus complexes qu'une utilité à une seule dimension, et ces critères sont en général inconnus des autres agents.

Comme l'explique (Kraus, 2001), il est difficile de qualifier une négociation de *bonne* ou de négociation *réussie*. Ces travaux se placent du point de vue des performances globales du système et proposent des critères pour évaluer un protocole de négociation :

- la rapidité : le temps de négociation des agents doit être le plus court possible, car il faut prendre en compte le coût des communications entre les agents, et le réduire,
- l'efficacité : il est plus intéressant d'avoir un grand nombre d'agents satisfaits par le résultat de la négociation, il est donc préférable que les agents trouvent une solution pareto-optimale,
- la simplicité : le processus de négociation doit être simple, les stratégies mises en oeuvre par les agents également : un agent doit pouvoir utiliser une stratégie dans un temps raisonnable,
- la stabilité : une stratégie stable est une stratégie qui reste intéressante lorsque le nombre d'agents (et de stratégies) qui interagissent dans le système augmente,
- le transfert d'argent : il peut être utilisé entre les agents durant une négociation. Comme il nécessite une gestion de la sécurité des transferts, et des coûts de maintenance, il est préférable de ne pas l'utiliser.

Ces critères n'ont pas de sens pour modéliser un processus de négociation entre humains. Ces différents travaux ne s'intéressent pas à la façon dont les humains interagissent, et comment simuler ce comportement, mais cherchent à *créer des environnements sociaux pour les machines*. Ils s'intéressent à l'établissement de règles sociales entre des agents dits rationnels (notion qui se base ici sur la maximisation de leur utilité propre, sur l'égoïsme). Leur objectif est de produire des comportements bénéfiques pour les agents.

Dans les situations de la théorie des jeux, comme le célèbre dilemme du prisonnier, les deux joueurs prennent une décision sans se concerter. Il ne s'agit pas de négociation car il n'y a pas d'interaction. Le raisonnement se base sur des matrices *préconstruites*, contenant les décisions possibles et leurs conséquences. De plus, ils s'appuient sur l'hypothèse de rationalité des individus (au sens où ils cherchent à maximiser une fonction d'utilité). Ce qui est difficilement acceptable si l'on souhaite modéliser des comportements réalistes.

Notre démarche s'appuie, elle, sur l'observation et la description du phénomène humain et social qu'est la négociation, en vue de le modéliser et de le simuler de façon la plus réaliste possible. Nous ne sommes pas dans une démarche d'optimisation. Nous montrerons néanmoins le parallèle qui peut être fait entre le raisonnement des agents de notre modèle et certains aspects de la théorie des jeux, dans la partie 5.2.6. Parallèle avec l'approche théorie des jeux.

2.3. La communication entre agents

2.3.1. Modes de communication

Communiquer signifie « établir une relation avec » (s.v. communiquer Le Petit Robert). Comme l'explique Pesty dans (Koning et Pesty, 2001) en termes de systèmes multi-agents, l'interaction (« action réciproque » s.v. interaction Le Petit Robert) comprend la communication ainsi que l'action sur le monde. « Les termes interaction et communication se rejoignent dans le fait que la transmission d'information induit parfois chez l'autre un comportement spécifique. Communiquer revient alors à une forme d'action particulière qui, au lieu de s'appliquer à la transformation de l'environnement, tend à une modification de l'état mental de l'agent destinataire. »

Il existe essentiellement 2 modes de communication entre agents (Koning et Pesty, 2001) :

- la communication par signaux, *via* l'environnement,
- la communication *directe* par envoi de messages.

Pour la communication par envois de messages, les agents ont besoin d'un langage commun. La modélisation des langages de communication entre agents, dont nous allons présenter très brièvement quelques aspects, fait l'objet de nombreux travaux de recherche.

2.3.2. Les langages de communication

Les travaux sur les langages de communication entre agents s'inspirent des communications humaines. La théorie des actes de langage a ainsi influencé les SMA, parce qu'elle identifie les constituants élémentaires de la communication. Cette théorie a été proposée sous la forme de la théorie classique des actes de langage par (Austin, 1962). Elle a été formalisée par (Searle, 1969) et (Vanderveken, 1988).

L'acte de langage est l'unité minimale de l'énonciation et correspond au niveau de description le plus fin. Il correspond à un but communicatif, c'est-à-dire à la marque d'une intention particulière d'un locuteur (un but illocutoire).

Deux langages, KQML (Knowledge Query and Manipulation Language) et FIPA-ACL (*Agent Communication Language*) sont considérés, en IAD, comme des standards.

KQML a été proposé en 1993, par l'ARPA-KSE (Knowledge Sharing Effort), dans le but de favoriser l'intéoperabilité d'agents hétérogènes, et leur permettre de partager des bases de connaissances.

Le langage FIPA-ACL, proposé par la FIPA (*Foundation for Intelligent Physical Agents*) a bénéficié des résultats de recherche de KQML. Il est donc assez semblable à KQML, avec une description plus fine des actes de communication, et l'ajout de protocoles régissant les échanges de ces messages. On trouve des protocoles élémentaires, comme par exemple le *FIPA Query Interaction Protocol*, permettant à un agent de demander une information à un autre agent, ou bien encore le *FIPA Request Interaction Protocol* qui permet à un agent de demander à un autre agent d'effectuer une action. La FIPA propose également une spécification du protocole Contrat Net, sur lequel nous allons revenir car il est dédié à la négociation entre agents (voir partie 2.2.7.1. Le protocole Contract Net).

Comme le met en évidence (Brassac *et al.*, 1996), ces langages ont été largement critiqués. Ils sont bien adaptés aux interactions entre agents, mais pas pour notre problématique de représenter au mieux les interactions entre sujets humains. Comme le rappelle Brassac, p.246, ce n'était pas l'objectif des concepteurs de ces langages.

Nous souhaitons ici, identifier les énoncés utilisés durant des négociations. Nous appellerons *performatives*, les types de messages utilisés par les agents, un énoncé étant une performative appliquée à une proposition. Nous souhaitons également modéliser la façon dont s'enchaînent ces différentes performatives, en formalisant ce que l'on observera par un protocole d'interaction entre agents.

2.3.3. Notion de protocole

La modélisation des communications entre agents utilise la notion de protocole. Beer *et al.* dans (Beer *et al.*, 1998, p.85), définissent un protocole comme « *l'ensemble des règles qui gouvernent l'interaction* ».

Les protocoles définissent l'ensemble de séquences valides des messages qui peuvent être échangés entre les agents. Les protocoles peuvent être formalisés à l'aide de différents outils comme par exemple les réseaux de Pétri ou les automates état-transition. Ces derniers sont très utilisés, les noeuds représentent l'état d'une conversation, et les transitions l'émission-réception d'un acte de langage. C'est ce formalisme qui a été utilisé dans BDIGGY, et que nous avons également utilisé ici.

De nombreux protocoles de négociation ont été proposés en Intelligence Artificielle Distribuée. Nous allons présenter les plus connus.

2.3.4. Les protocoles de négociation entre agents

2.3.4.1. Le protocole Contract Net

De nombreux travaux s'appuient sur le modèle général du Contract Net (Smith, 1980), qui est considéré comme le précurseur des travaux en négociation automatique. Il s'agit d'un protocole fondé sur l'échange de contrats, qui met en relation un agent : le gestionnaire (*manager*), avec plusieurs autres agents : les contractants (*contractors*). Il s'agit de négociation de *1 vers n* agents, qui interagissent entre eux au moyen de performatives. Un contrat peut être élaboré entre deux participants : un contractant et le gestionnaire. Le contractant est garant de l'exécution de la tâche et de la transmission de ses résultats au gestionnaire. Le gestionnaire, quant à lui, se porte garant de la gestion de la tâche et du traitement des résultats. C'est pourquoi ce modèle peut être vu comme un protocole d'allocation de tâches par réseau contractuel.

Le Contract Net fait partie des spécifications de la FIPA (Foundation for Intelligent Physical Agents) qui propose également une version itérée (voir FIG. 9).

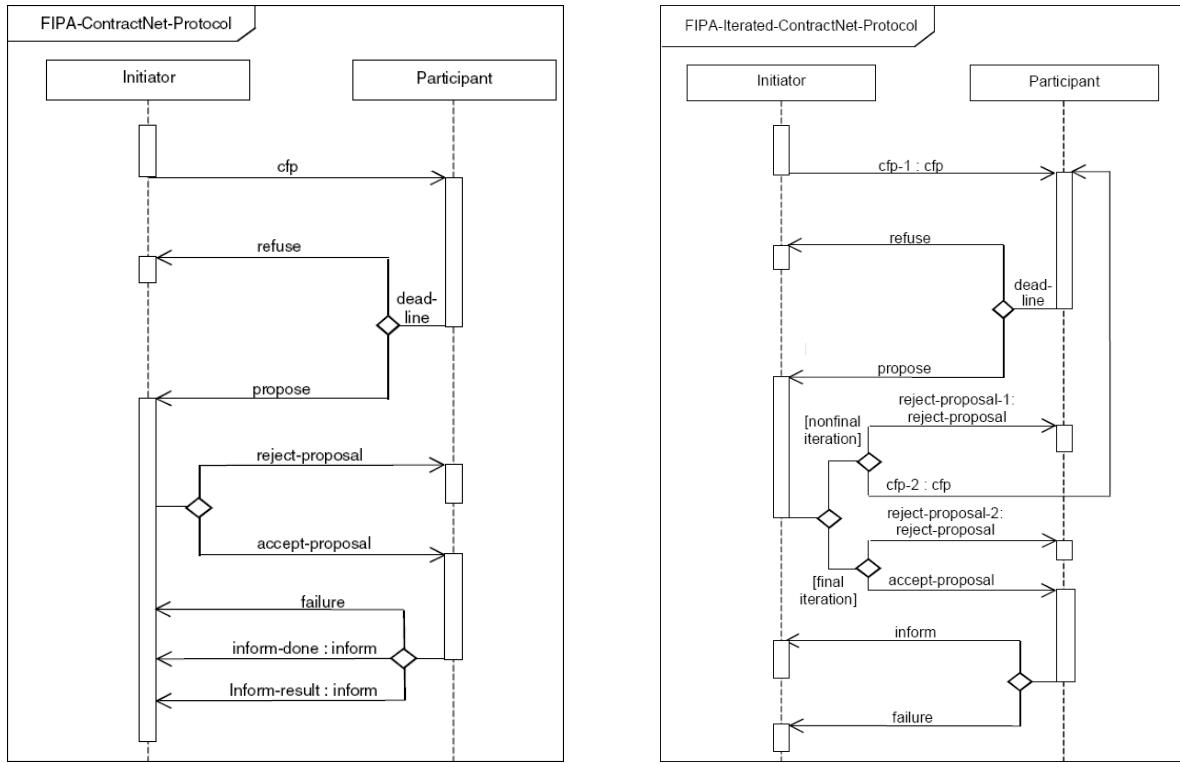


FIG. 9 - Interactions dans le protocole Contract Net et le protocole Contract Net Itéré (FIPA)

Le protocole se déroule comme suit (voir FIG. 10) :

1. Annonce de la tâche par le gestionnaire,
2. Soumission d'une offre par un ou des contractants,
3. Attribution de la tâche à un contractant par le gestionnaire,
4. Transmission du message de fin de tâche au gestionnaire par le contractant.

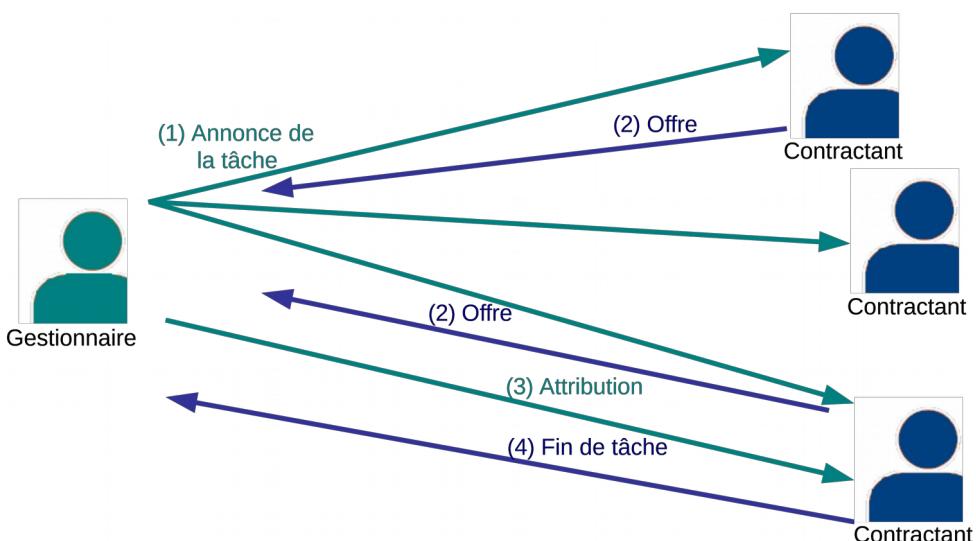


FIG. 10 - Le protocole Contract Net

Le Contract Net est pionnier et constitue une référence parmi les protocoles de négociation. Il a fait l'objet de nombreuses extensions, qui ont apporté des améliorations comme la possibilité d'effectuer des contre-propositions (Koo, 1988), des envois d'informations entre agents et l'itération des demandes (Conry *et al.*, 1988). On peut citer également l'ajout d'un modèle de décision basé sur un calcul de coûts (pour les choix de l'agent contractant), le regroupement des tâches interdépendantes, la résolution du problème de la congestion des messages, l'ajout d'un message indiquant aux contractants qu'ils n'ont pas été retenus (Sandholm, 1993), des niveaux d'obligation vis à vis des contrats attribués (Sandholm et Lesser, 1995).

De nombreux protocoles de négociation se basent sur le Contract Net. A l'origine, ce protocole avait pour objectif de proposer **un modèle de distribution des tâches, fondé sur des appels d'offres dans des marchés publics**. Nous montrerons qu'il ne convient pas pour modéliser la négociation que nous avons observée entre sujets humains (voir partie 4.2. Critique des modèles de négociation basés sur le Contrat Net).

2.3.4.2. Le protocole de Sian

Ce protocole de négociation a été conçu au sein de MALE (*Multi-Agent Learning Environment*) par Sian en 1991 (Sian, 1991). Il s'agit de permettre à un système multi-agent d'interpréter les événements de l'environnement. La perception du système est distribuée et les agents négocient pour adopter une interprétation commune de l'environnement. Le graphe de la FIG. 11 présente le schéma de la conversation.

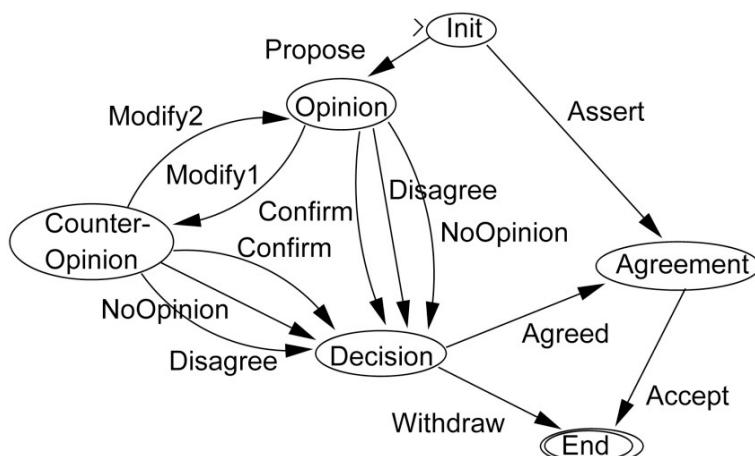


FIG. 11 - Graphe de la conversation dans le protocole de Sian

Les agents proposent des hypothèses, et pour chacune, il existe trois phases :

1. La *génération* de l'hypothèse,
2. La *discussion* entre les agents pour se mettre d'accord sur l'hypothèse,
3. L'*intégration* de l'accord.

Le tableau de la FIG. 12, présente les actes de langages disponibles pour chaque agent, dans une phase donnée.

PHASE	PERFORMATIVE	DESCRIPTION
1	assert	assertion non modifiable
1	propose	propose une nouvelle hypothèse
2	modify	donne une version modifiée de l'hypothèse proposée
2	disagree	indique un désaccord avec une hypothèse
2	noopinion	indique qu'il n'y a pas d'avis sur une hypothèse
2	confirm	indique un accord avec une hypothèse
3	agree	change l'état de l'hypothèse de proposée à agréée
3	accept	produit l'acceptation d'une hypothèse agréée
3	withdraw	retrait d'une hypothèse proposée

FIG. 12 - Performatives utilisées dans le protocole de Sian

Les agents communiquent *via* un tableau noir. Les propositions sont inscrites sur le tableau et chaque agent peut les lire, les accepter, les refuser ou les modifier.

Ici, les agents ont tous le même objectif : trouver une bonne interprétation de la situation. Avec sa perception locale de l'environnement, chaque agent va proposer des hypothèses. Les agents mettent en commun leurs hypothèses, et n'ont pas d'intérêts conflictuels. La discussion aboutit forcément à un accord, qui consiste en accepter (*accept*) ou retirer (*withdraw*) l'hypothèse du tableau.

2.3.4.3. Le protocole SANP

Le protocole SANP (Speech Act Based Negotiation Protocol) a été défini par Chang et Woo en 1992 (Chang et Woo, 1992) (Chang et Woo, 1994). Il propose une négociation entre deux agents : l'attaquant et le défenseur. L'agent initiant la conversation est appelé l'attaquant : il va tenter de faire accepter sa proposition à son partenaire, appelé le défenseur.

Le protocole se décompose en plusieurs phases :

1. *Situation de départ* : les agents établissent un modèle commun du sujet et estiment si une discussion est nécessaire,
2. *Emission d'une proposition et réception de l'avis* : l'attaquant émet sa proposition et le défenseur lui répond son accord ou son refus,
3. *Attaque* : si le défenseur refuse la proposition, l'attaquant va émettre ses arguments,
4. *Phase tactique* : le défenseur va défendre sa position et l'attaquant essayer de la contrer par des arguments. Ce processus continue jusqu'à ce qu'ils identifient leurs différences et tentent de les réduire,
5. *Résolution du problème* : chaque partie propose un compromis et tente de contrer celui de l'autre,
6. *Résultat final* : le résultat de la négociation peut être le compromis d'une partie, un compromis mutuel ou une demande d'arbitrage par une tierce partie.

A chaque phase correspond un ensemble de stratégies (voir FIG. 13) et de sous-diagrammes de transitions d'états (voir FIG. 14) suivant le raisonnement de l'agent. Le protocole est inspiré de la théorie des actes de langage en linguistique et la recherche sur la négociation en sociologie. Les auteurs se basent sur le *Struggle Model* et l'*Institutional Model* de (Ballmer et Brennenstuhl, 1981).

TACTICAL PHASE

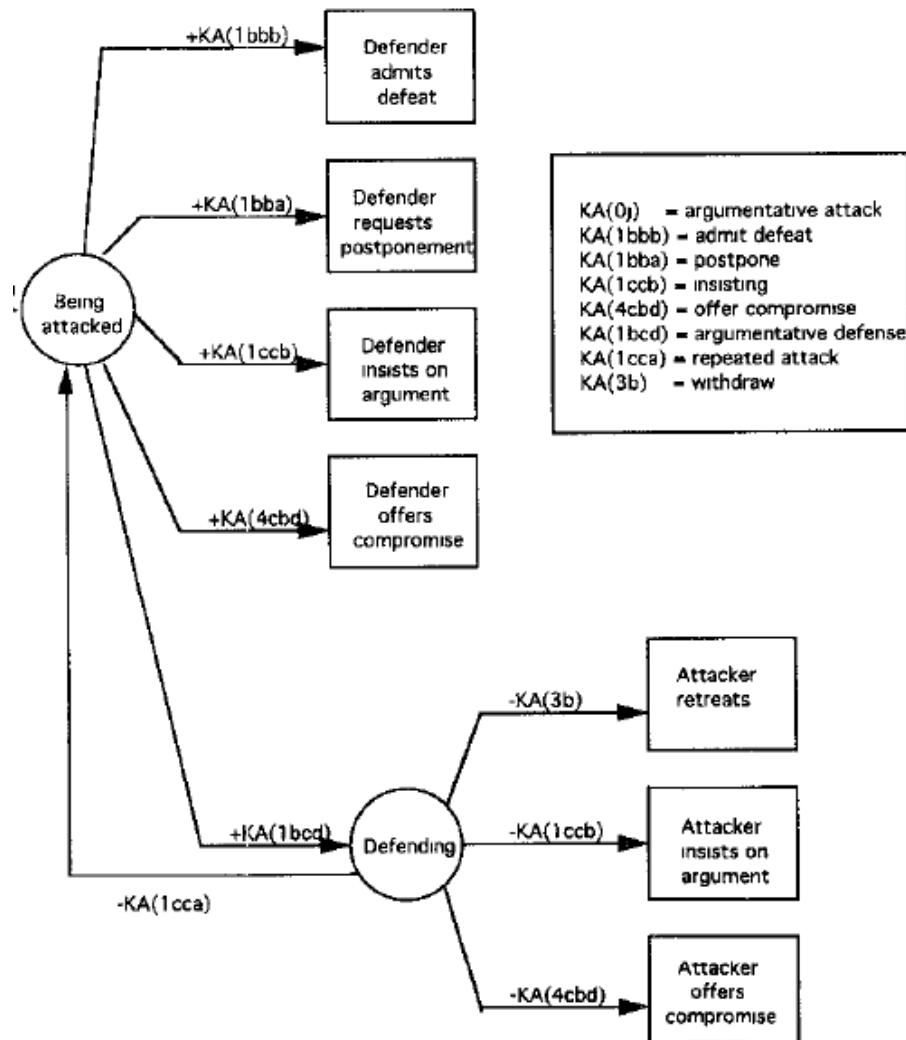


FIG. 13 - Diagramme de stratégie (Chang et Woo, 1994)

Une phrase *en SANP* est de la forme :

<function> <content>

où *function* est le nom d'une catégorie d'acte de langage (voir FIG. 15), et *content* dépend du domaine de l'application.

ATTACKER OFFERS COMPROMISE

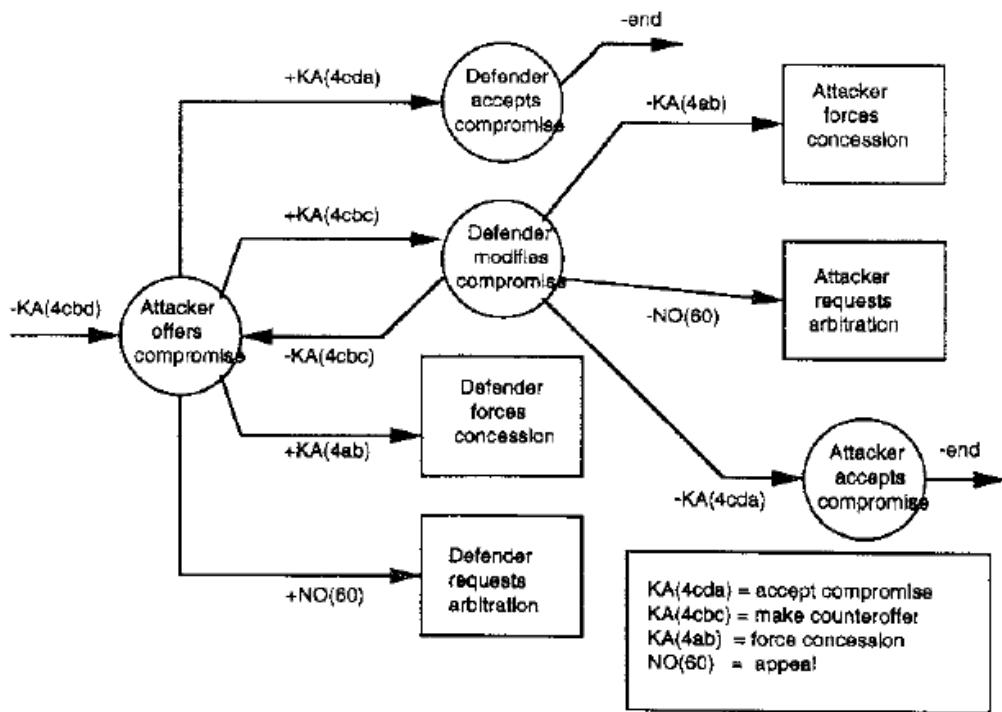


FIG. 14 - Sous-diagramme de raisonnement de l'attaquant, dans le protocole SANP

CODE	CATEGORY NAME	EXAMPLE SPEECH ACT VERBS
KA _{-2a}	make claim	ask, assert, claim opinion
KA _{-1a}	agree	agree, share the same opinion
KA _{-1c}	dissent	break with someone, have words with someone
KA _{0j}	argumentative attack	affirm, claim, confront
KA _{1bbb}	admit defeat	admit, agree to, give in, give way
KA _{1ccb}	insist	persist in, press, pursue
KA _{4ab}	force concession	force, overrule
KA _{4cbd}	offer compromise	make a contract, make an agreement
KA _{4cbc}	make counteroffer	accept in part, make stipulation
NO ₆₀	appeal	appeal to someone, bring forward
NO _{6c}	examine	hear, question
NO _{6da}	testify	give testimony, show proof

FIG. 15 - Exemples de verbes d'acte de discours dans les catégories utilisées dans SANP

Chang et Woo ont avis qu'on ne peut pas tout automatiser, et qu'il faut se demander ce que l'on peut automatiser et ce que l'on peut proposer, pour ce qui n'est pas automatisable. Des tâches comme rechercher de façon précise un terrain commun entre les protagonistes d'une négociation, ou faire des calculs d'équations (si l'on négocie un budget par exemple), sont

plus faciles à effectuer pour une machine. Par contre, il n'est pas possible, et non désirable, qu'un conflit soit résolu par la machine.

Pour évaluer les limites de leur modèle, Chang et Woo ont demandé à des sujets humains d'utiliser leur système pour résoudre des conflits. Deux groupes de deux étudiants ont été sollicités pour négocier sur le sujet *où aller déjeuner ?*

Malgré le fait qu'ils aient conçu leur protocole pour de la communication machine-machine, les auteurs expliquent qu'ils ont utilisé une expérimentation avec des sujets humains, pour ne pas altérer la conception à cause des limites des agents. Il fallait faire la part des choses entre les limites du protocole et les capacités limitées des agents logiciels. Utiliser le modèle avec des agents aurait faussé son évaluation, à cause de leurs limites (en terme de représentation des connaissances sur le domaine, et de capacités de raisonnement par exemple).

Avant l'expérimentation, il a été demandé aux sujets de faire « comme s'ils étaient des agents logiciels intelligents sans émotion ». Cela s'apparente à la technique de simulation appelée *Magicien d'Oz*, qui était souvent utilisée avant la mise en service d'un système comportant une interface utilisateur. Elle consiste à charger un humain de simuler le fonctionnement supposé du système, de sorte que l'interaction soit fluide et que le système semble réellement fonctionnel. Les usagers croyant s'adresser à une machine, ont alors des comportements spontanés. Ils ne sont informés qu'*a posteriori*, que le système était simulé.

Dans chaque binôme, les deux étudiants communiquaient par e-mail, et un expérimentateur était présent pour récolter leurs impressions et répondre à leurs éventuelles questions.

L'expérimentateur leur demandait les raisons de leur utilisation de telle ou telle fonction. La verbalisation concomitante et consécutive a été utilisée : des commentaires leur étaient demandés pendant et après l'expérimentation (voir partie 3.1.1. Le recueil des données et la question de la verbalisation).

Ils ont ainsi pu mettre en évidence les fonctions manquantes du protocole. Par exemple, quand ils ont conçu le protocole, ils ont supposé que les agents avaient assez d'informations pour mener les négociations. Mais les sujets humains ont manqué d'informations lorsqu'ils négociaient. Le défenseur, au lieu de s'opposer directement à la revendication de l'attaquant d'aller dans tel restaurant, demandait des informations supplémentaires sur le restaurant en question (« possédait-il une terrasse ? » par exemple). Dans le protocole, si l'attaquant manquait d'information, il devait retirer la revendication. Si le défenseur en manquait, il devait demander un ajournement de la négociation. Chang et Woo se sont ainsi rendu compte qu'il fallait incorporer des actes de langage permettant aux agents d'obtenir des informations sur les autres agents et des informations supplémentaires sur ce qui était négocié (comme le menu du restaurant).

Ils ont également constaté que les deux binômes ne négociaient pas de la même façon :

- dans le premier, l'attaquant tenait vraiment à aller dans un restaurant donné, et essayait de convaincre le défenseur de l'y suivre. La négociation s'est résumée donc plutôt à *va-t-on ou non dans ce restaurant ?*

- dans le second groupe, l'attaquant n'est pas resté fixé sur son choix et les deux sujets ont discuté de plusieurs restaurants. L'objet de la négociation était donc plutôt *lequel de ces restaurants allons-nous choisir* ?

Il s'est avéré que dans le second cas, le protocole ne pouvait traiter les échanges : les auteurs concluent qu'il faudrait un mécanisme de niveau supérieur, permettant pour la même négociation, d'utiliser le protocole pour chaque restaurant discuté.

Bien que les auteurs avaient également l'objectif de se rapprocher des négociations humaines, leur approche est différente de la nôtre : c'est après avoir conçu le modèle qu'ils ont fait intervenir des sujets humains. Dans notre modèle, c'est en amont de la conception que nous avons observé les humains. Par exemple, comme nous l'expliquerons dans la partie 4.1.1.3. Les échanges, nous avons constaté la part importante des demandes d'informations pendant les négociations, ce qui nous a permis d'en tenir compte pour la modélisation.

2.3.4.4. Les protocoles dans les systèmes d'enchères

Les types d'enchères

Il existe différents sites d'enchères en ligne (on peut citer eBay, Yahoo enchères, onSale), et de nombreux types d'enchères. Les plus utilisés sont les suivants :

Les enchères publiques ou enchères anglaises : chaque participant annonce publiquement son offre. Le participant qui propose la plus grande, gagne l'objet au prix de son offre.

Les enchères ascendantes : le vendeur diminue tout le temps le prix jusqu'à ce que l'un des participants achète l'objet au prix courant.

Enchères descendantes ou enchères hollandaises : le vendeur augmente tout le temps le prix jusqu'à ce que l'un des participants achète l'objet au prix courant.

Les offres scellées au meilleur prix : ces enchères sont aussi parfois appelées *enchères premier prix offres cachées*, chaque participant soumet une offre sans savoir les offres des autres. Celui qui fait la plus grande, gagne l'objet et paye le montant de son offre.

Les offres scellées au second meilleur prix ou enchères de Vickrey : ces enchères sont aussi parfois appelées *enchères deuxième prix offres cachées*. Chaque participant soumet une offre sans connaître les offres des autres. Celui qui fait la plus grande, gagne l'objet et paye le montant de la seconde meilleure offre proposée.

Les enchères sont une forme très répandue de négociation, mais comme Verrons le met en évidence dans (Verrons, 2004), certaines ne sont pas de la réelle négociation. Dans les enchères à offres scellées par exemple, une seule proposition de prix est émise par les acheteurs : il n'y a pas de discussion entre le vendeur et les acheteurs. Nous partageons complètement ce point de vue et notre définition exclut ces types d'enchères.

Verrons ajoute par ailleurs que dans les ventes aux enchères, seuls le vendeur et le gagnant de l'enchère sont satisfaits. Ce qui ne constitue pas une négociation, qui doit satisfaire au mieux tous les participants. Nous pensons que la négociation ne s'effectue qu'entre un acheteur et un vendeur. Les acheteurs s'influencent entre eux, mais le *rapport de force*, la discussion de négociation, s'effectue entre un acheteur et un vendeur. Le fait qu'il y ait plusieurs acheteurs donne un certain pouvoir au vendeur, qui, du coup, n'est pas dépendant de l'acheteur avec qui il négocie. De façon symétrique, l'acheteur peut également aller *voir ailleurs*. Le fait qu'ils ne soient pas *seuls* leur donne un certain pouvoir, une marge de manœuvre. La transaction les concerne tous les deux, et les autres participants de la vente aux enchères ne sont pas acteurs de la négociation qui les occupent. Nous voyons les enchères comme plusieurs négociations parallèles qui s'influencent. Nous pensons qu'on ne peut parler de satisfaction qu'entre les agents qui effectuent la transaction (l'acheteur et le vendeur). C'est eux qui, si l'on peut dire, vont *signer le contrat*. (voir la partie 5.2.6. La notion de satisfaction).

Les systèmes implémentant différents types d'enchères sont nombreux. Nous allons présenter les protocoles dans les systèmes Kasbah et AuctionBot, qui figurent parmi les plus connus.

Kasbah

Kasbah (Chavez et Maes, 1996), développé au MIT Media Lab, propose des agents acheteurs ou des agents vendeurs pour des utilisateurs. Un utilisateur crée son agent, lui donne une direction stratégique, et l'envoie au marché central des agents. Chaque agent veut accomplir une transaction en utilisant les choix de l'utilisateur (prix souhaité, prix minimum et maximum acceptables, etc.). Kasbah permet trois stratégies de négociation : inquiet, sang-froid et frugal, qui correspondent respectivement aux fonctions linéaire, quadratique et exponentielle pour augmenter le prix proposé. Lorsqu'un accord est trouvé, les utilisateurs humains (s'ils le souhaitent) sont prévenus par mail, et peuvent effectuer la transaction.

Les méthodes à implémenter par un agent désirant participer sont les suivantes :

- `accept-offer?(agent, from-agent, offer)`

Cette méthode est utilisée pour demander à l'agent *agent* s'il accepte ou non l'offre *offer* de l'agent *from-agent*. *agent* renvoie *accepted* ou *rejected*.

- *what-is-price?(agent, from-agent)*

Cette méthode est appelée par *from-agent* pour demander à l'agent *agent* quelle est son offre de prix. Si *agent* est un acheteur, il s'agit du prix qu'il est prêt à payer pour le produit. Si *agent* est un vendeur, il s'agit du prix auquel il souhaite vendre son produit.

- *what-is-item?(agent, from-agent)*

Cette méthode est appelée par l'agent *from-agent* pour demander à l'agent *agent* quel produit il recherche (si *agent* est un acheteur) ou quel produit il souhaite vendre (si *agent* est un vendeur). Une description du produit est renvoyée par *agent*.

La FIG. 16 montre un extrait de la trace d'une négociation, enregistrée par le vendeur Agent 1 (chaque agent effectue un enregistrement). Dans cet exemple, Agent 2 et Agent 3 sont les acheteurs potentiels du produit mis en vente par Agent 1.

```

Date: Wed 24 Jan 21:21:13 | Event: Offered agent 3 87. Rejected.

Date: Wed 24 Jan 21:21:13 | Event: Asked agent 3 their price. Replied 78.

Date: Wed 24 Jan 21:21:31 | Event: Agent 3 offered 81. I rejected.

Date: Wed 24 Jan 21:21:31 | Event: Agent 3 asked my price. I replied 83.

Date: Wed 24 Jan 21:21:31 | Event: Offered agent 2 83. Rejected.

Date: Wed 24 Jan 21:21:31 | Event: Asked agent 2 their price. Replied 62.

Date: Wed 24 Jan 21:21:32 | Event: Agent 2 offered 62. I rejected.

Date: Wed 24 Jan 21:21:32 | Event: Agent 2 asked my price. I replied 83.

Date: Wed 24 Jan 21:21:50 | Event: Offered agent 3 79. Accepted.

```

FIG. 16 - Extrait d'une exécution de la plate-forme Kasbah (Chavez et Maes, 1996)

La négociation est simple mais il y a bien des interactions entre agents acheteurs et vendeurs, qui communiquent avec des actes de langage spécifiques à Kasbah. Le protocole n'est cependant pas basé sur des négociations observées dans la réalité, entre acheteurs et vendeurs humains.

AuctionBot

AuctionBot, développé au laboratoire d'intelligence artificielle de l'université du Michigan par Michael P. Wellman et Peter R. Wurman (Walsh *et al.*, 1998), est un serveur de ventes aux enchères. Chaque utilisateur crée une vente aux enchères en sélectionnant le type de la vente aux enchères (enchères anglaises, de Vickrey, etc), et en spécifiant ses **paramètres** (la date limite, la méthode pour résoudre les conflits entre des offres, etc.). Un vendeur règle le prix de réservation de sa vente aux enchères et laisse AuctionBot s'occuper de collecter les enchères des acheteurs, qui ont été placées dans une base de données lors de leur émission. C'est le *commissaire-priseur* du système qui gère la vente aux enchères, qui sélectionne les offres d'après un ensemble de règles défini, et informe les participants.

Les échanges possibles au sein du système sont par exemple :

- Bid : envoyé d'un agent au système. Il s'agit de proposer une offre de prix (pour acheter un produit ou le vendre) qui peut inclure des paramètres (comme la date d'expiration de l'offre par exemple)
- Bid Admittance/Bid Rejection : envoyé du système à un agent, pour lui indiquer que son offre satisfait/enfreint les règles de la vente aux enchères.

Ce système ne permet pas la négociation : il n'y a en effet aucune interaction entre vendeurs et acheteurs. C'est le système qui fait l'intermédiaire. L'objectif ici est d'automatiser une vente aux enchères, et non de permettre de négocier les offres. Ce système est utilisé pour la Trading Agent Competition, une compétition entre agents logiciels.

La Trading Agent Competition

La **Trading Agent Competition** (TAC) est une compétition créée en 2001 par Michael P. Wellman et Peter R. Wurman. Tous les ans, une trentaine d'équipes proposent des agents qui sont mis en concurrence pour préparer des voyages pour huit clients différents. Pour cela, un agent participant doit négocier les composants du voyage : les vols, les nuits d'hôtel, etc. Plusieurs types d'enchères sont organisés. Les critères de notation des agents portent sur les dépenses effectuées, les revenus obtenus, la satisfaction des clients. De nombreuses équipes de recherches, avec une implémentation de leur modèle d'agent, participent à cette compétition.

Le Supply Chain Management Game

Cette compétition s'intitule *gestion de la chaîne logistique d'approvisionnement*⁹. Il s'agit, pour les agents, de planifier et coordonner les activités d'approvisionnement sur une chaîne logistique, de l'acquisition des matières premières, jusqu'à la livraison des produits finis. Les agents en compétition doivent interagir avec les fournisseurs, gérer la chaîne de production, et obtenir des commandes des clients.

Le jeu a été conçu conjointement par une équipe de recherche du e-Supply Chain Management Lab à l'université de Carnegie Mellon et le SICS : Swedish Institute of Computer Science (SICS).

Nouvelle compétition CAT

La Trading-Agent Competition (TAC Classic) et la TAC SCM (Supply Chain Management) ont été inventées dans un objectif de développer des stratégies pour agents logiciels acheteurs et vendeurs, pour des salles de marchés. Les règles d'échanges commerciaux et les mécanismes d'interactions sont fixés par les organisateurs, et les participants créent des agents qui suivent les règles fixées.

Une nouvelle compétition CAT est proposée pour la prochaine édition (juillet 2007). CAT (qui est l'inverse de TAC mais qui signifie aussi *CATallactics*, la science des échanges) consiste à faire exactement l'inverse qu'à la TAC. Les agents logiciels sont créés par les organisateurs, et les participants doivent définir les règles pour faire interagir les agents vendeurs et acheteurs et pour paramétriser les frais des services. Les participants s'affrontent pour attirer les acheteurs et faire faire des profits aux vendeurs. Cela se termine lorsque les règles obtenues et les honoraires fixés amènent à un bon compromis entre l'attraction des acheteurs et le gain des vendeurs.

Il existe de nombreux protocoles s'inspirant de ceux que nous venons de présenter. Beaucoup d'entre eux s'inspirent du protocole Contract Net. On peut également citer (Vercouter *et al.*, 1999), qui proposent un protocole au croisement du protocole de Sian et du Contract Net.

Nous allons maintenant présenter le modèle et la plate-forme GENCA (Verrons, 2004), qui, au niveau de la communication, se base sur des améliorations du Contract Net. Nous revenons dessus dans la partie 4.2. Critique des modèles de négociation basés sur le Contract Net, pour mettre en évidence la non concordance entre les modèles de négociation s'inspirant du Contract Net, et ce que nous avons observé dans le processus de négociation entre humains.

⁹ Supply Chain Management

2.3.4.5. Le protocole dans GENCA

GENCA est un projet de l'équipe SMAC du LIFL. Dans un cadre de génie logiciel, l'objectif était de fournir un modèle et une plate-forme facilitant le développement des applications de négociation. GENCA signifie *modèle Général de Négociation de Contrats entre Agents*, et est également le nom de l'implémentation du modèle en une API Java (*Generic Negotiation of Contracts API*). La plate-forme vise la générnicité, la portabilité, l'uniformisation et l'automatisation des envois de messages. (Verrons, 2004) montre que dans tous les types de négociation, on retrouve des éléments similaires. Les couches communication, négociation et stratégie sont séparées. Pour la communication, GENCA est basé sur une version étendue du Contract Net.

La plate-forme permet donc de réaliser plusieurs formes de négociation différentes. Verrons propose une hiérarchie des principales formes de négociation qui peuvent être automatisées :

- la forme la plus primaire le *take it or leave it offer*, offre à prendre ou à laisser (qui ne consiste pas en de la négociation),
- les systèmes de vote,
- les enchères,
- le Contract Net Protocol,
- les négociations multi-attributs,
- les négociations multi-niveaux et combinées,
- la négociation par argumentation.

Les négociations multi-attributs sont situées entre le Contract Net Protocol et les négociations multiniveaux car elles peuvent se dérouler en un ou plusieurs tours.

GENCA se présente sous la forme d'une architecture en trois couches, qui sépare le coeur de la négociation des considérations spécifiques aux applications : la communication et la stratégie de négociation.

Les objectifs de la négociation utilisés dans le modèle sont des contrats portant sur des ressources. Ils représentent l'objet minimal qui sera négocié. Les utilisateurs sont ensuite libres d'y ajouter d'autres critères (comme un prix, une qualité de service, etc.).

La couche de **communication** permet de définir le mécanisme de communication entre les agents qui est spécifique à l'application réalisée. Un serveur de noms est utilisé pour centraliser les ressources qui seront négociées et mémoriser les participants inscrits pour différents types de négociation. Cette couche contient le format des messages échangés : les messages de liaison avec le serveur de noms et les actes de langage entre les agents.

La couche de **négociation** contient toutes les structures de données nécessaires à la gestion des négociations : la description des ressources, des contrats, des propriétés des contrats, les

mécanismes permettant les négociations simultanées, la gestion des situations d'interblocage (*deadlocks*), la possibilité de se rétracter et de renégocier des contrats. Elle contient également des outils pour la création de stratégies, qui permettent de récupérer des informations sur les négociations passées (par exemple la ressource la plus/moins proposée pour un initiateur donné, le nombre de fois qu'une ressource a été proposée par un initiateur donné, etc.) et sur les priorités (priorités sur les ressources et sur les participants).

La couche **stratégique** permet de séparer les mécanismes de raisonnement des agents. Les stratégies de négociation se présentent comme des interfaces à implémenter, l'une pour le rôle d'initiateur, l'autre pour le rôle de participant. Pour pouvoir utiliser un comportement par défaut, une implémentation de la stratégie de chaque rôle est fournie. Elles se basent sur les listes de préférences fournies par l'utilisateur sur les ressources et sur les autres participants.

La plate-forme peut être utilisée de façon automatique (avec des agents répondant aux différentes propositions GENCA), et de façon manuelle (l'utilisateur humain prend alors les décisions). L'utilisateur peut ainsi créer des contrats, répondre aux propositions qui lui sont faites, visualiser les messages envoyés et reçus par l'agent et surveiller les négociations en cours.

Plusieurs applications ont été réalisées avec GENCA : un système de prise de rendez-vous, un système d'enchères à offres scellées, un système d'enchères hollandaises, un système de création d'emplois du temps, un système de choix d'un restaurant, et un jeu de négociation.

Nous reviendrons sur le protocole utilisé dans la partie 4.2. Critique des modèles de négociation basés sur le Contrat Net.

3. Les expérimentations

Comme nous l'avons vu, il existe des types variés de négociation. Nous avons restreint notre étude à un processus de négociation entre agents vendeur et acheteurs. Nous avons réalisé une expérimentation avec des sujets humains pour obtenir des comportements réels variés que nous avons enregistrés, pour les analyser afin d'en proposer un modèle. Malgré les contraintes matérielles et temporelles, les comportements observés sont représentatifs d'une plage assez large de situations réelles de négociation. Dans cette partie, nous commençons par présenter la problématique que pose l'organisation d'expérimentations et le recueil des données. Nous présentons ensuite le problème et le cadre expérimental choisis, puis le logiciel outil développé, et le déroulement des expérimentations réalisées.

3.1. La problématique des expérimentations

Pour disposer de données, il nous fallait un moyen d'observer un processus de négociation entre humains. Plusieurs possibilités s'offraient à nous. La situation dans laquelle se retrouve un sujet au cours de notre expérimentation est une situation dynamique, dans le sens où elle évolue au cours du temps, même en l'absence d'action de sa part et qu'il ne peut déterminer et contrôler que partiellement cette évolution (Hoc, 1996, p.12). Cette évolution est rythmée par les interventions des autres sujets, interventions orales ou actions modifiant l'état du jeu. Il s'agit de suivre ce qui est dit, ce qui est échangé ou vendu, et d'intervenir ou non dans les négociations.

(Hoc, 1996) présente trois méthodes de recherche permettant de recueillir des données exploitables sur des situations dynamiques :

- l'étude des *situations réelles*, au sens d'une situation de la vie courante. Une situation en laboratoire est bien sûr tout aussi *réelle*, Hoc emploie ce terme au sens où la situation n'est pas conçue spécifiquement pour les objectifs de la recherche. Dans notre cas d'étude, cela aurait consisté au recueil de données lors de négociations spontanées,
- l'étude des *situations simulées*, pour les situations où il est difficile de recueillir des observations. Elles permettent de paramétrier des facteurs inaccessibles dans les situations réelles. Il s'agit de reproduire la situation dite *situation-cible* sans en modifier les caractéristiques principales (cela peut poser des problèmes de réalisme et de pertinence des résultats),
- l'étude des *micro-mondes*, environnement spécialement conçu pour des travaux de recherche. Il vise à créer des situations dynamiques et paramétrables pour le besoin des expérimentations. Il n'a pas forcément pour objectif de recréer fidèlement les

caractéristiques d'une situation réelle. Il s'agit parfois de situations irréelles, ou simplifiées par rapport à la réalité, mais avec lesquelles il est possible de faire un parallèle (cela peut poser des problèmes de correspondance entre la situation étudiée et la situation réelle, en terme de transposition des résultats).

C'est cette troisième méthode que nous avons appliquée. Obtenir de la négociation spontanée, de la négociation entre humains ignorant qu'ils sont observés, par exemple dans la rue, dans des commerces, sur des marchés, n'était pas adapté à ce que nous recherchions. D'une part parce que cela est beaucoup trop lourd à mettre en place pour obtenir des données en quantité suffisante (cela nécessitait, pour le passage d'un support audio ou vidéo vers la machine, un temps d'extraction des données trop important pour être traité dans le cadre de ce stage). D'autre part, parce que la visée de notre étude est l'interaction d'agents logiciels entre eux et avec des humains. Faire communiquer les sujets via un environnement médiatisé par ordinateur était donc plus adapté.

Nous avons opté pour la mise en place d'un environnement, un *micro-monde*, permettant de recréer des situations de négociation dans un cadre de commerce électronique. Ce choix s'est porté sur ce type d'application car le commerce est à l'origine du concept de négociation.

3.1.1. Le recueil des données et la question de la verbalisation

Utilisée pour la première fois en IA par (Newell et Simon, 1972), la verbalisation est souvent employée dans l'étude des activités humaines. (Caverni, 1991), cité dans (Chaignaud, 1996) distingue plusieurs méthodes pour y recourir. Elles diffèrent par le type de consigne utilisée par l'expérimentateur pour la provoquer (verbalisation *de la procédure* quand le sujet doit décrire le plus simplement possible sa démarche, verbalisation *des justifications de la procédure* quand le sujet doit expliquer tout le cheminement de son raisonnement, verbalisation *directe* quand le sujet doit penser à voix haute). Un second point de choix est l'enchaînement de la verbalisation et de l'exécution de la tâche. La verbalisation est appelée verbalisation *anticipée* quand elle se fait avant l'exécution de la tâche, *concomitante* quand elle se fait au fur et à mesure de l'accomplissement de la tâche, ou *consécutive* quand elle est effectuée après la tâche. Pour plus de précision, se reporter à (Chaignaud, 1996).

Les données obtenues par la verbalisation sont importantes mais souvent insuffisantes pour inférer les activités cognitives sous-jacentes aux comportements (Hoc, 1996). La verbalisation est en général combinée avec d'autres sources d'observables pour contrôler leur validité respective. La méthode utilisée est la confrontation des résultats provenant de la verbalisation avec ceux de l'enregistrement de toutes les actions exécutées par le sujet du début à la fin du problème.

Nous avons fait le choix de ne pas recourir à la verbalisation pour cette première série d'expérimentations pour trois raisons :

- ne pas surcharger l'activité des sujets. Nous aurions eu besoin de la verbalisation concomitante, et l'activité effectuée par les sujets était déjà suffisamment prenante pour ne pas leur ajouter cette contrainte,
- ne pas influencer les sujets. Nous avons préféré ne pas risquer de les influencer par la présence d'un tiers, durant le déroulement du jeu. Un expérimentateur passait régulièrement les voir, et le contenu du *chat* était surveillé, afin d'éviter tout problème technique. Mais nous avons préféré ne pas laisser quelqu'un dans chacune des pièces durant toute la durée de l'expérimentation,
- enfin parce que nous nous intéressons dans un premier temps au déroulement de la négociation, et non pas à l'activité de résolution de problème. La résolution du problème est ici un prétexte à provoquer le processus de négociation. Nous ne cherchions pas à comprendre *le pourquoi* mais bien *le comment* des interventions. Nous nous intéressons à la dynamique des échanges entre les sujets, et non pas à la prise de décision de ceux-ci.

(Verrons, 2004) explique que le mode de communication n'a pas d'incidence sur la négociation (p.5). Si cela est vrai pour les protocoles de communication au niveau réseau¹⁰, cela n'est plus vrai au niveau des protocoles de plus haut niveau. Nous nous intéressons ici aux interactions entre humains, et il est vrai que nous ne communiquons pas de la même façon à l'oral qu'à l'écrit (Chapanis, 1979). Plusieurs expérimentations ont été proposées faisant intervenir différents moyens de communication : face à face, par voix seule, en écriture à distance ou au moyen d'un télécopieur. Les résultats montrent que la communication est plus riche (contient plus de mots) quand elle se fait à voix haute et les problèmes sont résolus plus rapidement. Il n'y a pas de différence significative quand la communication orale est faite face à face ou non. Une restriction du vocabulaire échangé entre sujets n'altère pas la résolution du problème ce qui démontre une flexibilité de la part des sujets dans leur expression orale ou écrite.

Pour notre expérimentation, il nous fallait définir la façon de récolter les données, et le type de données dont nous avions besoin. Nous aurions besoin d'un logiciel par la suite pour mettre en œuvre la négociation entre agents logiciels. Le développer en premier lieu permettait par la suite d'utiliser l'interface pour la mise en œuvre.

La problématique est de savoir *ce qu'on stocke, ce qu'on enregistre, sur quoi on se focalise*. Sans savoir exactement ce que l'on va observer, il est difficile de fixer à l'avance ce dont on a besoin. Soit on souhaite observer des paramètres précis, on peut alors organiser des expérimentations visant à effectuer des relevés quantitatifs par exemple ; soit on ne sait pas encore bien ce que l'on va obtenir, et il faut décider au mieux des données enregistrées, pour qu'elles soient exploitable en aval.

¹⁰ Nous considérons que le fait qu'un message échangé entre agents soit acheminé via un protocole TCP, ou un protocole UDP (par exemple), n'aura aucune incidence sur le traitement de ce message par les agents.

Nous avons décidé de permettre à tout moment la communication en temps réel d'un sujet vers tous les autres, *via* une messagerie instantanée (clavardage, communément appelé *chat*), et d'enregistrer toutes les conversations. La visée étant la modélisation des échanges sur le réseau internet entre *agents logiciels* et *agents logiciels ou humains*, nous supprimions de cette façon la part du contexte existant entre humains lors de communications orales (physique, expressif), dont nous ne devions pas tenir compte. Nous avons également voulu éliminer au maximum l'influence de la qualité des relations entre les sujets, sur les interactions. Pour cela, nous avons imposé l'anonymat lors de l'utilisation du logiciel (ce qui permet par exemple de supprimer la connaissance des rapports de hiérarchie entre les participants).

3.1.2. Le choix du problème

Après le choix du type d'expérimentation, il s'agissait de définir le problème à traiter : le contexte des négociations, l'objet des négociations, le nombre de participants, le déroulement. Une des difficultés était de trouver le bon équilibre entre la liberté que nous souhaitions laisser aux participants et les contraintes qu'imposerait l'expérimentation choisie (en terme de possibilités d'intervention et d'actions). Notre objectif était de créer une situation qui déclencherait un processus de négociation entre les participants, mais sans les guider ou les contraindre, dans la mesure du possible.

Nous souhaitions obtenir une situation propice à l'apparition d'un processus de négociation entre humains, pour pouvoir l'observer, l'analyser et le modéliser sans nous focaliser sur un type de négociation en particulier. Il a fallu créer un contexte, et celui que nous avons choisi s'apparente aux enchères. Il ne s'agit pas d'enchères à proprement parler car rien n'est imposé à un participant, ni l'obligation de faire une proposition, ni l'ordre d'interventions des participants. D'autre part le dialogue est autorisé et laissé libre entre les participants, ce qui n'est pas le cas dans les ventes aux enchères.

La problématique était de réduire les contraintes imposées par un type de négociation en particulier. Un exemple de contrainte serait la règle du *Beat-the-Quote*, qui impose que toute nouvelle offre soit supérieure à celle déjà faite précédemment. Ici, les participants avaient libre choix de l'évolution de leurs propositions.

3.2. Le cadre expérimental

3.2.1. Description du jeu

L'application choisie est un jeu de négociations. Nous avons opté pour le jeu Négoces conçu par François Tréca (<http://www.negoces.fr>), qui convient particulièrement bien à ce que nous recherchions. Les joueurs se retrouvent en effet dans des situations où ils sont obligés de négocier pour parvenir à leur objectif.

Afin d'observer les comportements des joueurs tout au long de la partie, nous avons développé un logiciel permettant de jouer à ce jeu sur ordinateur, en réseau (voir FIG. 17). Les participants, chacun isolé dans un bureau, communiquent durant la partie *via* une messagerie instantanée.

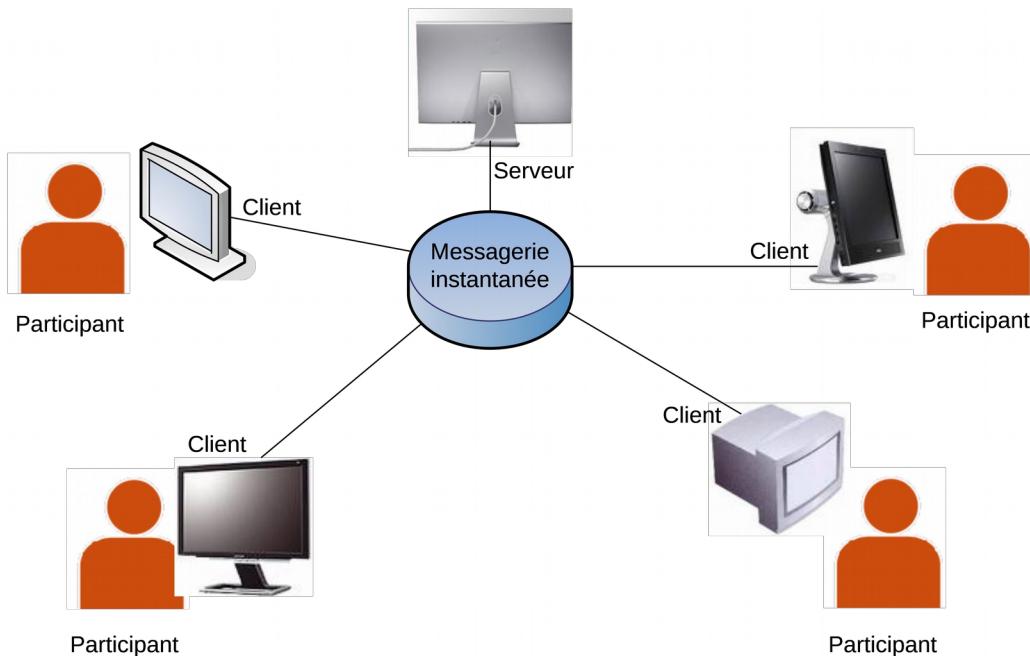


FIG. 17 - Plate-forme expérimentale

Voici les instructions données à chaque participant :

Bonjour,
Merci d'avoir accepté de contribuer à cette expérimentation.
Il s'agit de participer à un jeu de type jeu de négociations.
Vous êtes 4 participants.
Chaque joueur est un marchand. Vous avez, au départ, le monopole sur 2 marchandises et détenez des pièces d'or.

Pour gagner, vous devez détenir 8 marchandises différentes et garder suffisamment de pièces d'or.

Pour cela, des échanges sont organisés entre vous.

Durant la partie, plusieurs ventes ont lieu sur le marché. Au premier tour, il s'agit d'une vente aux enchères. Lors des tours suivants, il s'agit de ventes par négociation.

Au début du jeu, chaque joueur reçoit 15 pièces et 8 cartes marchandise : deux séries de 4 cartes identiques.



Le but pour chaque joueur est de posséder 8 cartes marchandises différentes et au moins 10 pièces.

Durant la partie, le nombre de pièces possédées par chacun est visible par tous les joueurs, mais les cartes marchandise ne le sont pas.

Pour la première vente, c'est la carte Diamant qui est placée sur le marché, c'est une marchandise normale qui comptera par la suite comme une autre.



Lors des négociations, vous utilisez la messagerie instantanée présente dans l'interface, et les boutons de l'interface pour valider vos décisions.

-> Utilisez la messagerie au maximum pour réussir à trouver un accord.

-> Vous jouez de façon anonyme, avec des pseudos, utilisez-les et ne divulguez pas votre réelle identité.

Premier tour : vente aux enchères d'une carte Diamant

La carte marchandise Diamant est mise aux enchères. Les joueurs commencent à proposer un certain nombre de pièces pour l'acquérir. C'est la seule vente où vous êtes 4 acheteurs. A chaque proposition, vous pouvez soit surenchérir d'un nombre de pièces supérieur à la meilleure offre courante proposée, soit vous retirer définitivement de la vente. Il n'y a pas d'ordre particulier, chacun intervient quand il le souhaite. **Le temps n'est pas limité**, c'est lorsqu'il ne reste plus qu'une proposition (parce que 3 joueurs se sont retirés de la vente) que le joueur restant remporte la carte marchandise Diamant. Les pièces qu'il a dépensé pour l'acquérir sont retirées de la partie. Ce joueur possède maintenant 9 cartes et devient le vendeur.

Tours suivants : ventes par négociations

Le joueur possédant 9 cartes est le vendeur. Il met en vente une de ses cartes marchandise. Une carte marchandise une fois proposée ne peut plus être récupérée

par le vendeur. Elle devra être vendue, même si aucune offre n'enthousiasme le vendeur.

Les autres joueurs deviennent alors les acheteurs. Ils négocient avec le vendeur, sans ordre particulier, l'achat de la carte sur le marché. Les acheteurs peuvent proposer au vendeur :

- **une carte (et une seule),**
- **des pièces,**
- **une carte et des pièces.**

(Noter que l'on peut proposer des "pièces négatives" : l'acheteur peut proposer "une carte et moins deux pièces", c'est-à-dire que le vendeur, s'il accepte son offre, devra lui payer deux pièces).

Une offre peut être augmentée, diminuée ou retirée jusqu'à la fin de la négociation. Les acheteurs peuvent former des alliances avec ou contre le vendeur. Seule la carte placée sur le marché peut être échangée : les acheteurs ne peuvent pas s'échanger de cartes entre eux.

Le temps n'est pas limité : le vendeur accepte l'offre de son choix, reçoit le paiement convenu (carte et/ou pièces) et donne sa carte à l'acheteur.

Le joueur qui possède maintenant 9 cartes en main devient le nouveau vendeur, il peut s'agir du même joueur qu'au tour qui vient de se terminer.

Fin de partie et vainqueur

Le premier joueur à posséder 8 cartes marchandise différentes et au moins 10 pièces remporte la partie. Si deux joueurs remplissent ces conditions après une vente, c'est l'acheteur qui remporte la partie.

Remarques

La première vente, celle de la marchandise Diamant, est la seule où vous êtes 4 acheteurs, et permet de déterminer qui sera le premier vendeur.

Avoir la main (être le vendeur) est très avantageux : ne la laissez pas trop longtemps au même joueur, c'est-à-dire accordez-vous entre acheteurs pour ne plus proposer que des pièces (car proposer une carte permet au vendeur de rester avec 9 cartes en main, et donc de rester vendeur au tour qui suit).

Si vous êtes vendeur trop longtemps, vous risquez de faire gagner un acheteur avant vous (voir Fin de partie et vainqueur).

Lorsque vous êtes vendeur, vous pouvez par exemple demander aux acheteurs quelle carte les intéresse, puis leur demander de proposer une carte qui vous intéresse, etc., vous pouvez faire des alliances entre vous.

Surveillez les offres, prévenez de leur erreur les acheteurs qui feraient une offre faisant gagner le vendeur.

Utilisez le "chat" un maximum.

Vous avez le droit de prendre des notes pendant la partie (utilisez la feuille jointe).

Le logiciel du jeu va maintenant vous être présenté oralement.

Ne fermez jamais l'application (le serveur s'en chargera une fois la partie terminée).

Nous avons fait quelques adaptations du jeu d'origine : le principe des propositions à tour de rôle du premier tour a été supprimé. Ceci afin d'éliminer le choix arbitraire du joueur qui proposait la première enchère et du sens de rotation (donc de la séquence imposée) pour les

premières propositions. Le premier tour se déroule comme les autres, sauf qu'il n'y a pas de vendeur, les quatre joueurs sont acheteurs et tentent d'acquérir une marchandise supplémentaire pour devenir le vendeur du tour suivant (le diamant).

Chaque participant est en concurrence avec les autres pour atteindre un objectif : réunir un certain nombre de marchandises différentes (tout en gardant une certaine quantité d'argent) avant les trois autres joueurs. Les participants ont ainsi des *buts incompatibles* : il ne peut y avoir qu'un seul gagnant.

Les **ressources** servant de monnaie d'échange sont des pièces, et il est possible d'échanger une carte marchandise en plus, ou à la place des pièces. Jamais plus d'une, ce qui implique que les ressources marchandises (cartes) sont quantitativement réparties durant toute la durée du jeu : un joueur, le vendeur courant, possèdera neuf cartes marchandises, et les trois autres en possèderont huit. Les ressources sont *suffisantes*, c'est-à-dire que les joueurs peuvent chacun réunir assez de marchandises différentes, le problème étant de le faire avant les autres.

En suivant les distinctions expliquées par Estivie dans (Estivie, 2006, p.6), on peut dire que les ressources ici sont :

- *discrètes* (les pièces peuvent également être vues comme une ressource continue, que l'on peut discréteriser, par exemple une somme de 6 pièces peut être transformée en un ensemble de 6 ressources discrètes de 1 pièce),
- *indivisibles*,
- *non-partageables* (elles ne peuvent pas appartenir à plusieurs agents en même temps),
- *statiques* : elles ne sont ni consommables (comme le serait par exemple du carburant), ni périssables, (comme par exemple de la nourriture),
- *multi-items* : il existe plusieurs cartes marchandises identiques, 4 par type de marchandise, et on y réfère en utilisant le même nom.
- *de type ressources* : à un niveau suffisant d'abstraction, un problème d'allocation de tâches peut être réduit à un problème d'allocation de ressources (Estivie, 2006, p.6). les tâches sont souvent interdépendantes, notre problème met en jeu des ressources matérielles non dépendantes les unes des autres.

3.2.2. Le logiciel d'expérimentation

Le logiciel développé utilise l'API RMI (Remote Method Invocation). Il s'agit d'une API Java permettant de manipuler des objets distants (comme un objet instancié sur une autre machine virtuelle, par exemple sur une autre machine du réseau) de manière transparente, c'est-à-dire de la même façon que si l'objet était sur la machine virtuelle de la machine locale. On dit généralement que RMI est une solution *tout Java*, contrairement à la norme CORBA (*Common Object Request Broker Architecture*) de l'OMG (*Object Management Group*) permettant de manipuler des objets à distance avec n'importe quel langage. Les connexions et les transferts de données dans RMI sont effectués par Java sur TCP/IP grâce à un protocole propriétaire JRMP (*Java Remote Method Protocol*) sur le port 1099.

Depuis la version Java 2 1.3, les communications entre client et serveur s'effectuent grâce au protocole RMI-IIOP (*Internet Inter-Orb Protocol*), un protocole normalisé par l'OMG et utilisé dans l'architecture CORBA.



FIG. 18 - Capture d'écran du logiciel d'expérimentation, vue pour un acheteur

Les traces enregistrées par le logiciel d'expérimentation sont constituées à la fois des actions effectuées dans l'interface et des conversations provenant de la fenêtre de clavardage.

Les participants disposent d'une interface permettant les actions suivantes :

- *Enchérir* (pour un acheteur),
- *Se retirer de la vente* (pour un acheteur),
- *Mettre en vente* (pour le vendeur),

et d'un *chat* permettant de s'exprimer librement, (voir FIG. 18 et FIG. 19).

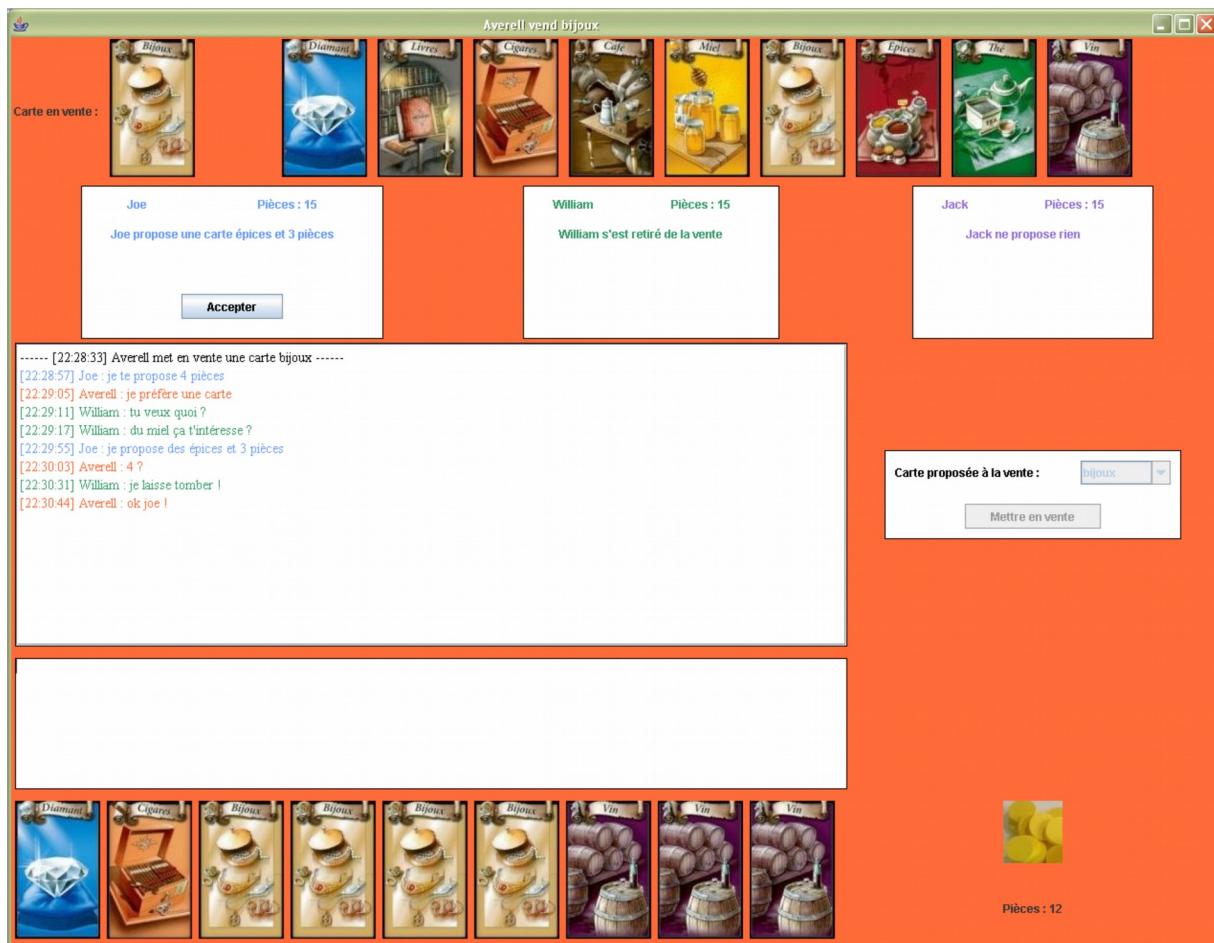


FIG. 19 - Capture d'écran du logiciel d'expérimentation, vue pour un vendeur

3.2.3. Le déroulement des expérimentations

Nous avons choisi d'impliquer **4 participants**. Ce qui est suffisamment élevé pour permettre la formation d'éventuelles alliances entre les joueurs, et suffisamment réduit pour permettre la réalisation de l'expérimentation sur le plan matériel.

Nous n'avons fixé **aucune limite de temps**. Si une négociation était engagée concernant la vente d'une marchandise, elle pouvait se poursuivre sans limite de temps.

En tout, nous avons fait jouer 14 parties de 4 participants, membres du laboratoire LITIS (enseignants-chercheurs, étudiants et administratifs). Un des groupes ayant rencontré des difficultés de compréhension des règles, nous avons décidé de retirer la partie jouée de l'ensemble. Nous avons ainsi recueilli 13 fichiers exploitables de type HTML¹¹, comportant tous les dialogues, toutes les actions effectuées par les joueurs dans l'interface, et leurs ressources en cartes et pièces à chaque vente de carte.

¹¹ HTML = Hypertext Markup Language. Nous avons utilisé ce type de fichier pour des raisons pratiques (meilleure visibilité lors de l'analyse). Les balises HTML ont été supprimées par un script Perl, lorsque nous avons eu besoin du texte brut pour le traitement automatique (scripts Perl).

Voici un extrait d'une trace recueillie durant l'expérimentation :

```
[15:28:10] 127, ACCEPT_SELL (Jack, COINS -, CARD miel, Averell, COINS -, CARD café)
[15:28:33] 128, William SAY Il me semble opportun de faire attention
[15:28:41] 129, Jack SAY qui veut une carte cigare ou épices ?
[15:28:58] 130, Averell SAY je propose 2 pieces
[15:29:16] 132, Jack SAY averell je suis ok
[15:29:27] 133, Jack AUCTION cigares
[15:29:29] 134, William SAY Je propose 3 pièces pour cigare
[15:29:34] 135, William PROPOSE (Jack, COINS -, CARD cigares, William, COINS 3,
CARD -)
[15:29:35] 136, Averell TAKE_OUT
[15:29:36] 137, Jack SAY ok william
[15:29:37] 138, ACCEPT_SELL (Jack, COINS -, CARD cigares, William, COINS 3, CARD -)
```

Cette trace correspond à une vente au cours d'une partie. Pour garder l'anonymat, les joueurs s'appellent Joe, William, Jack et Averell. Les lignes en noir correspondent aux actions effectuées sur l'interface, les lignes en couleurs sont les interventions dans le *chat*. Chaque joueur *parle* dans une couleur spécifique.

4. Analyse des traces

Cette partie présente l'analyse des fichiers générés par le logiciel d'expérimentations. Nous présentons tout d'abord les traces, les observations que nous avons pu faire au niveau des situations obtenues, puis l'analyse des échanges relevés au cours des parties effectuées, appuyée par une étude statistique. Nous finissons par une critique des modèles fondés sur le protocole Contract Net, en montrant, sur l'extrait d'une trace, que les échanges observés ne peuvent être réalisés avec ce type de protocoles.

4.1. Les traces

Cette section présente notre analyse des traces recueillies lors des expérimentations.

Dans ce qui suit, nous utiliserons les termes *transaction*, *vente* et *partie* :

- une *transaction* est l'échange effectif d'une carte contre une carte et/ou des pièces, les négociations observées portent sur les modalités de la transaction,
- une *vente* se déroule entre la transaction de la vente précédente et la transaction correspondant à l'accord trouvé entre le vendeur et l'acheteur de la carte, sauf pour la première vente, qui débute à partir de l'achat de la carte diamant,
- une *partie* est une expérimentation sur un groupe de 4 sujets, et comporte au minimum 5 ventes.

Les traces enregistrées par le logiciel d'expérimentation sont constituées à la fois des actions effectuées dans l'interface et de *français spontané* provenant de la fenêtre de clavardage. Nous avons récolté 13 fichiers HTML. Les parties ont durées entre 11 minutes 21 secondes et 1 heure 47 minutes 10 secondes. La durée totale d'enregistrement (sur l'ensemble des parties effectuées) est de 9 heures 8 minutes 24 secondes.

Comme nous l'avons expliqué, les participants disposaient d'une interface permettant de faire les actions (proposer une enchère par exemple), et d'un *chat* permettant de s'exprimer librement (voir partie 3.2.2. Le logiciel d'expérimentation). Le plus souvent, ils utilisaient les deux, en annonçant une offre dans le *chat*, et en validant cette offre *via* l'interface. Nous avons cependant observé des groupes qui ne se servaient de l'interface qu'à la fin de la vente. Tous les échanges de la vente se faisant en dialoguant *via* le *chat* et seule la transaction sur laquelle l'acheteur et le vendeur étaient tombés d'accord par écrit, faisait l'objet d'actions sur l'interface (bouton de proposition pour l'acheteur et bouton de validation pour le vendeur). A l'inverse, nous avons observé des ventes qui ne comportaient aucune intervention écrite dans le *chat*, et pour lesquelles tous les échanges de messages se faisaient *via* les actions sur les boutons de l'interface. Lors de l'étude des traces, nous avons bien sûr décompté *les doublons*.

4.2. Niveaux micro et macro

La première observation que nous avons pu faire porte sur la situation globale dans laquelle les sujets interagissent. Bien que la situation de problème place en interaction 4 personnes simultanément, nous observons que les négociations s'instaurent à chaque fois entre un acheteur et le vendeur. Deux acheteurs n'interagissent pas entre eux, du point de vue des négociations. Nous avons illustré cela avec le schéma de la FIG. 20. Lorsque qu'ils discutent entre eux c'est pour faire des commentaires à propos d'un autre acheteur (« *Je sens qu'Averell va faire une bonne affaire* »), des relances (sur lesquelles nous allons revenir) ou bien de l'humour (par exemple des jeux de mots « *black Jack* », « *thé fou où quoi ?* »).

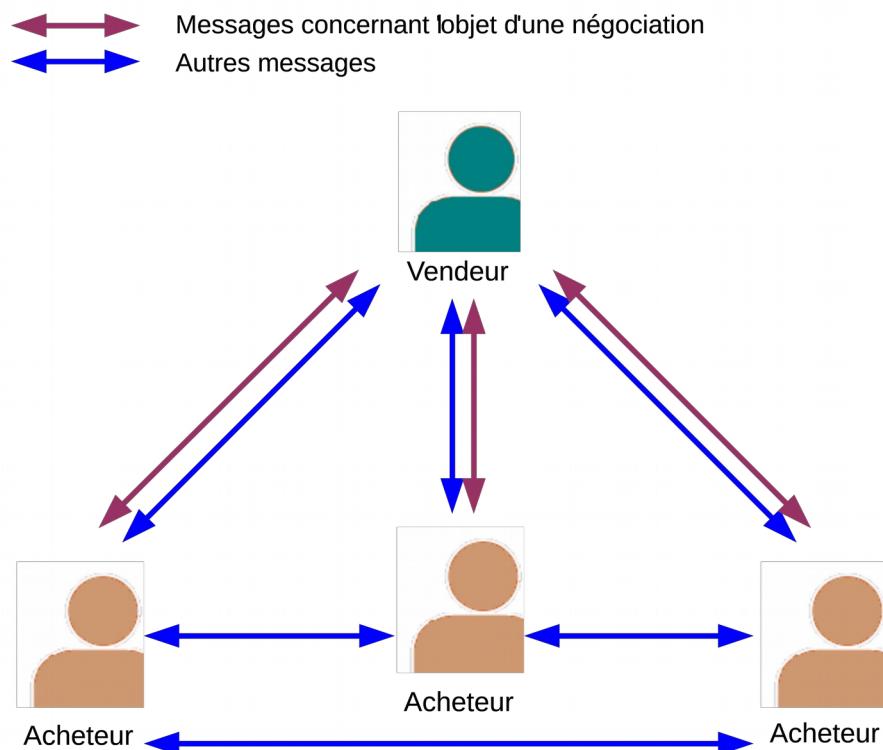


FIG. 20 - Les interactions observées durant les expérimentations

Les sujets devaient effectuer les transactions, afin d'avancer dans le jeu. Ils ont collaboré pour effectuer l'ensemble des ventes et terminer une partie. Il s'est instauré une situation de collaboration entre eux. Nous reprenons les concepts de *niveau micro* et *niveau macro* du domaine SMA pour illustrer :

« *La macro-situation, tout en étant la résultante des micros-situations qui la composent, introduit un ensemble de problèmes dont la résolution passe par la production d'une quantité de micro-situations elles-mêmes porteuses de problèmes* » (Ferber, 1995, p.77).

Au niveau *micro*, des situations de compétition ont motivé les négociations au moment d'une vente, à chaque fois entre un acheteur et le vendeur. Nous pouvons résumer cela par le schéma de la FIG. 21.

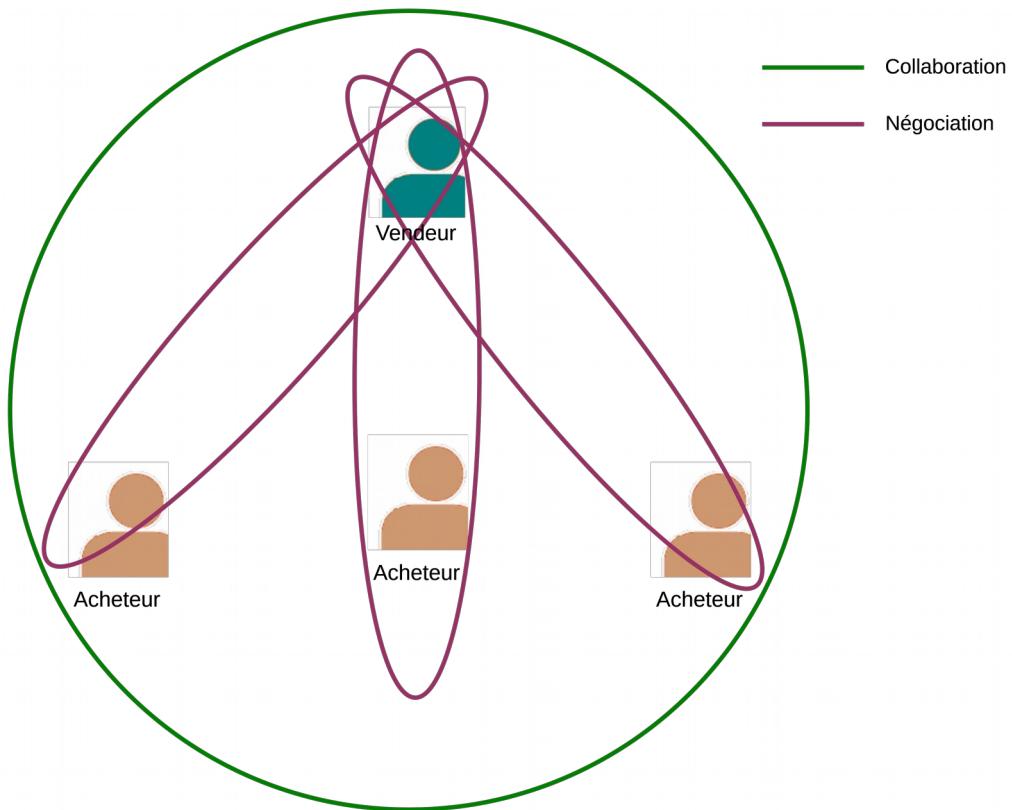


FIG. 21 - Les niveaux micro et macro

4.3. Les échanges

Nous observons de nombreuses demandes d'information dans les échanges entre les participants. Dans ces phases de recherche d'information, beaucoup de questions restent sans réponse.

De la part du vendeur les questions sont par exemple « *qui est intéressé par quoi ?* », « *qui peut me proposer des épices ?* ». De la part d'un acheteur, les demandes sont par exemple « *qu'est ce que tu peux nous proposer ?* », « *qu'est ce que tu veux acquérir ?* », « *qu'as-tu et que souhaites-tu ?* ».

Sans pouvoir expliquer pourquoi les participants ne répondent pas, nous pouvons supposer que les joueurs sont parfois concentrés sur les offres proposées, que le jeu va trop vite (la vitesse de l'arrivée des propositions de chacun et le flux de dialogue dans le *chat* nécessitait parfois une concentration accrue). Nous pouvons également penser que dans certains cas, ils ne se sentent pas concernés, ou bien encore qu'ils ne souhaitent pas y répondre (révéler la carte qu'ils recherchent ne leur paraissant pas judicieux à ce moment du jeu).

Nous observons également ce que nous avons appelé des *relances*. Un participant relance un ou plusieurs participants. Voici des exemples extraits de différentes parties : le vendeur relance un ou des acheteurs : « *aller joe* », « *alors les autres* », « *alors vous proposez ?* », un acheteur relance le vendeur par exemple : « *bon william on la fait cette affaire* ». La relance a également été observée entre les acheteurs : « *bon, vous vous décidez* » (de la part d'un acheteur ne participant pas aux négociations d'une vente, mais qui s'impatientait et souhaitait passer à la vente suivante).

Les quatre participants sont en compétition et il y a, comme nous l'avons dit, des échanges deux à deux, du point de vue de la négociation.

Les types d'échange observés durant les parties sont les suivants :

- des demandes d'information (obtenir une information sur les souhaits du vendeur ou des acheteurs, selon le rôle du participant),
- des mises en vente (uniquement pour le vendeur),
- des relances (accélérer le processus de décision d'un participant),
- des discussions (arriver à une entente),
- des informations (transmettre spontanément une information sur ses souhaits de vente/achat).

On ne retrouve pas forcément tous ces échanges durant une vente, mais au minimum une mise en vente et une discussion.

La discussion de négociation débute de deux façons différentes :

- lorsque l'acheteur fait une offre au vendeur,
- lorsque le vendeur demande à l'acheteur de lui en faire une.

Dans le second cas, nous parlons d'une situation où le vendeur souhaite réellement engager la négociation avec l'acheteur. Cela va plus loin que ce que nous appelons une *relance* : le vendeur est l'initiateur de la négociation.

Voici un extrait d'une trace, correspondant à une vente au cours d'une partie :

[15:28:10] 127, ACCEPT_SELL (Jack, COINS -, CARD miel, Averell, COINS -, CARD café)
[15:28:33] 128, William SAY Il me semble opportun de faire attention
[15:28:41] 129, Jack SAY qui veut une carte cigare ou épices ?
[15:28:58] 130, Averell SAY je propose 2 pieces
[15:29:16] 132, Jack SAY averell je suis ok
[15:29:27] 133, Jack AUCTION cigares
[15:29:29] 134, William SAY Je propose 3 pièces pour cigare
[15:29:34] 135, William PROPOSE (Jack, COINS -, CARD cigares, William, COINS 3, CARD -)
[15:29:35] 136, Averell TAKE_OUT
[15:29:36] 137, Jack SAY ok william
[15:29:37] 138, ACCEPT_SELL (Jack, COINS -, CARD cigares, William, COINS 3, CARD -)

Les lignes en noir correspondent aux actions effectuées sur l'interface, les lignes en couleur sont les lignes tapées dans le *chat*, (chaque participant a une couleur spécifique). Il s'agit ici d'une vente assez courte : elle commence par une mise en garde de William. Jack, qui est vendeur, demande aux acheteurs leurs préférences entre deux cartes, cigarettes ou épices. Question qui reste sans réponse, mais Averell fait une première proposition de 2 pièces. Jack exprime son intérêt pour la proposition d'Averell (il donne une information) et met en vente sa carte cigarettes. William fait alors une proposition de trois pièces. Averell se retire de la vente et Jack accepte la proposition de William. La ligne contenant ACCEPT_SELL correspond à la transaction effectuée via l'interface.

Nous avons observé que le processus de négociation ne commence pas forcément à partir de la mise en vente d'une carte marchandise.

C'est pourquoi nous dissocierons la mise en vente, qui fixe l'objet de la négociation, d'avec la discussion entre les deux sujets sur la transaction à effectuer.

L'ensemble des traces a été décrypté, et nous avons défini les actes de langage présentés FIG. 22). Trois d'entre eux sont propres à l'application (*auction*, *sold* et *takeOut*), et trois autres (*inform*, *query* et *reply*) existent déjà dans le modèle BDIGGY.

PERFORMATIVES	DESCRIPTIONS
auction ⁺	L'agent vendeur met en vente une carte. [10:36:49] 274, Joe SAY je vends du thé
takeOut ⁺	Un agent acheteur quitte la vente et la discussion de négociation éventuellement commencée. [10:32:04] 79, Jack TAKE_OUT [10:32:21] 80, Jack SAY je suis en cure
inform*	Un agent envoie spontanément une information avec <i>inform</i> , par exemple une carte qui l'intéresse. [10:52:23] 189, Jack SAY les épices m'interessent
query* / reply*	Un agent demande une information à un ou plusieurs agents (s) avec un <i>query</i> . Ce(s) dernier(s) lui répond(ent) avec un <i>reply</i> . [10:38:27] 63, Joe SAY qui peut me proposer une carte miel contre cigarette [10:38:34] 64, Jack SAY moi
bid	Un agent acheteur fait une proposition à l'agent vendeur. [11:12:24] 61, William SAY je propose 5 pieces et carte the

[†] Performatives propres à l'application du jeu de négociation.

[‡] Performatives déjà existantes dans le modèle BDIGGY.

acceptBid	L'agent vendeur accepte la proposition d'un agent acheteur avec <i>acceptBid</i> . Il prévient ensuite les acheteurs qu'il a vendu sa carte, avec un <i>sold</i> . [11:52:58] 375, Joe SAY 4 pièces [11:53:15] 376, Jack SAY 4 pieces ok
refuseBid	Un agent refuse la proposition d'un autre agent. [11:16:01] 298, William SAY 1 pièce [11:16:04] 299, Jack SAY non [11:16:07] 300, William SAY :P [11:16:10] 301, William SAY si si [11:16:15] 302, Jack SAY non
request	Le vendeur demande à un ou plusieurs agent(s) acheteur(s) de faire une proposition ou d'apporter une modification à leur proposition, par exemple proposer plus de pièces. [11:53:23] 683, Jack SAY monte un peu joe
relaunch	Un agent relance un ou plusieurs agent(s). [11:16:08] 87, Averell SAY bon jack tu te décides ??
warning	Un agent met en garde un ou plusieurs agents sur la position d'un autre. [10:32:52] 83, Averell SAY Je vous préviens que si vous vendez a joe, il a gagné.

FIG. 22 - Liste des performatives

Nous les appellerons *performatives*, et définirons leur sémantique dans la partie 5.2.5. La sémantique des performatives.

Chaque trace a ensuite été annotée par des marques correspondant aux performatives :

```
#b : bid
#i : inform
#q : query
#rp : reply
#r : refuse
#rl : relaunch
#rq : request
#w : warning
```

À noter que les types de messages *acceptBid*, *auction*, *bid*, *takeOut*, réalisées via l'interface étaient relevés par le logiciel, par les marques ACCEPT_SELL (qui marquait une transaction et la fin de la vente), AUCTION, PROPOSE, TAKE_OUT. Des annotations ont été ajoutées :

#to : takeOut, regroupe les sorties via le bouton de l'interface et les sorties annoncées dans le *chat*
#bc : bid doublons provenant de double-clic sur les boutons de l'interface
#bd : bid doublons d'un #b, lorsque l'offre était faite à la fois via l'interface et dans le *chat*

Voici pour exemple, l'extrait précédent annoté :

[15:28:33] 128, William SAY Il me semble opportun de faire attention
#w
[15:28:41] 129, Jack SAY qui veut une carte cigare ou épices ?
#q
[15:28:58] 130, Averell SAY je propose 2 pieces
#b
[15:29:16] 132, Jack SAY averell je suis ok
#i
[15:29:27] 133, Jack AUCTION cigars

[15:29:29] 134, William SAY Je propose 3 pièces pour cigare
#b
[15:29:34] 135, William PROPOSE (Jack, COINS -, CARD cigares, William, COINS 3, CARD -)
#bd
[15:29:35] 136, Averell TAKE_OUT
#to
[15:29:36] 137, Jack SAY ok william

[15:29:37] 138, ACCEPT_SELL (Jack, COINS -, CARD cigares, William, COINS 3, CARD -)

L'annotation des traces n'a pas toujours été évidente à réaliser. La compréhension des conversations *en contexte*, a parfois nécessité des inférences, pour dégager le sens donné (construit) par les participants. Voici, à titre d'exemple, deux lignes interprétées comme un refus, et qui met en évidence la nécessité d'étudier les traces *à la main* dans un premier temps :

[11:14:30] 282, Averell SAY 2
[11:14:33] 283, Jack SAY merci
[11:14:38] 284, Jack SAY non

4.4. Étude statistique

Des scripts en langage Perl¹² ont été codés et lancés sur les traces annotées, afin de réaliser une étude statistique.

Dans 6 groupes (parmi les 13), la mise en vente de la carte n'est pas la première action dans plus de 50% des ventes. La vente commence par une discussion pour décider de la carte à vendre, durant laquelle les participants font des demandes d'information, des propositions et des relances.

Nous avons observé des offres *à prendre ou à laisser* durant les discussions de négociation. 10 ont été relevées dans 7 groupes différents, exprimées par des acheteurs dans des phrases comme « *c'est mon dernier mot* », « *dernier prix* » ou encore « *dernière offre* » (voir le diagramme de la FIG. 23, où bidTl signifie *take it or leave it offer*). Nous observons que les acheteurs ayant fait une telle proposition s'y tiennent. Dans la moitié des situations où cela se produit, cette phrase fait céder le vendeur qui accepte la proposition, dans l'autre moitié il n'y aura plus aucun échange durant la vente entre l'acheteur et le vendeur. Dans tous les cas, nous observons que ce type de proposition coupe court à la discussion instaurée entre l'acheteur et le vendeur. Ce qui illustre les propos de Verrons, qui ne considère pas le *take it or leave it offer* (offre *à prendre ou à laisser*) comme de la négociation (Verrons, 2004, p.30), car il n'y a pas de discussion permettant la recherche de compromis. Nous constatons qu'en effet, cela interrompt brutalement la discussion de négociation entre les parties.

Une partie se termine lorsqu'un participant a réuni 8 cartes différentes (et gardé au moins 10 pièces). Il fallait donc au minimum 5 échanges de cartes dans une partie (donc 5 ventes comprenant obligatoirement une transaction carte-carte). Ce cas s'est produit dans 2 groupes sur les 13. Nous avons observé une moyenne de 12 ventes environ, avec un maximum de 23 ventes.

Nous n'avons relevé aucune corrélation entre le nombre de ventes et la durée de la partie, ni entre les interventions d'un participant et le rôle tenu par celui-ci. Nous ne retrouvons pas non plus de lien entre la durée d'une vente et le rôle des participants (voir à titre d'exemple de diagramme généré, la FIG. 24). Sur l'ensemble des groupes, la vente la plus courte a duré 15 secondes. Elle a consisté en une mise en vente, une proposition d'un acheteur et l'acceptation du vendeur. Cela reflète le minimum nécessaire pour une vente. D'autre part, elle n'a comporté qu'une seule intervention orale du vendeur, pour indiquer la carte qu'il mettait en vente.

La vente la plus longue a duré 1 heure 2 minutes et 15 secondes sur une partie de 1 heure 40 minutes et 13 secondes. Le vendeur et le dernier acheteur (les 2 autres acheteurs s'étant retirés de la vente) sont entrés dans une négociation qui a duré 54 minutes.

¹² Perl est un langage de programmation dont le nom signifie d'après son créateur (L. Wall), *Practical Extraction and Report Language* (langage pratique d'extraction et de génération de rapports) ou Pathetically Eclectic Rubbish Lister (collectionneur de déchets pathétiquement éclectiques).

Dans ce groupe nous avons observé des promesses (« *ok pour la prochaine vente* », « *je te donne la priorité au tour suivant* »), des propositions conditionnelles (« *je te donnerai la main, promis* »¹³). Dans ce groupe, et très rarement dans 3 autres groupes, nous avons observé des menaces (« *vite, sinon je baisse* »), des alliances¹⁴ (« *allions nous tous contre joe* »), des propositions conditionnelles (« *si tu me vends ta carte, je t'en proposerai une à la prochaine vente* »). (Bentahar *et al.*, 2003) par exemple, proposent des performatives pour modéliser les propositions conditionnelles. Pour notre part, toutes ces interventions ont été observées en quantité insuffisante pour que nous les prenions en compte dans la modélisation.

Nous ne prenons pas non plus en compte l'humour, qui est observé dans les groupes de façon très hétérogène. L'humour a des répercussions sur les relations entre sujets humains, et donc sur la négociation. Dans notre situation il permet, entre autres, de maintenir la cohésion du groupe des 4 participants. Mais nous ne considérerons pas ici ce type d'interactions : outre le fait qu'il ait été observé dans une minorité des groupes, les relever met en jeu une part d'interprétation importante dans la lecture des dialogues. La prise en compte de l'humour est d'autre part discutable en ce qui concerne des agents logiciels (nous ne pouvons ici en modéliser les répercussions). Enfin parce que l'humour observé ici est très lié au jeu (en rapport avec les personnages des Dalton par exemple), donc au problème choisi.

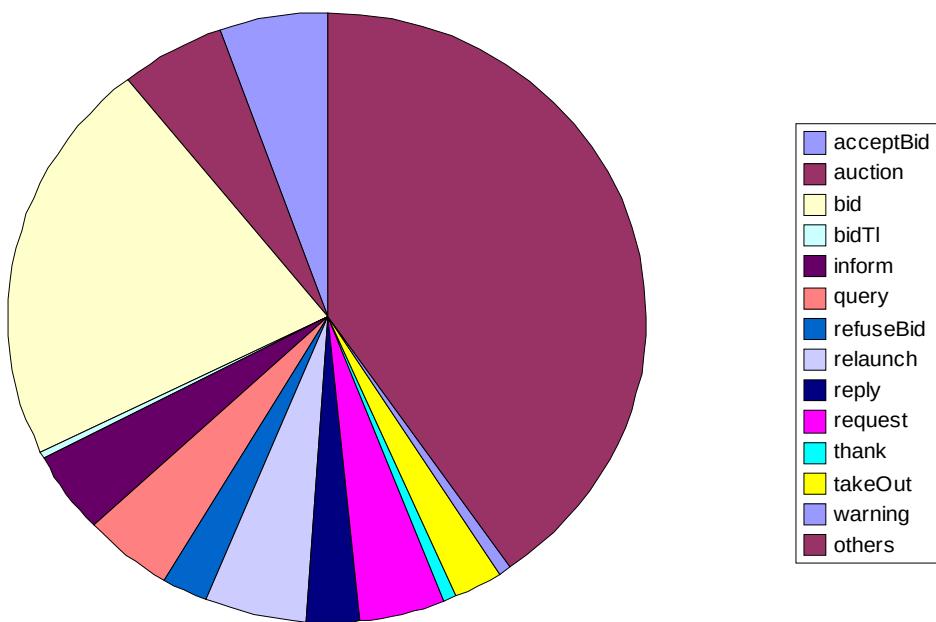


FIG. 23 - Types d'interventions sur l'ensemble des groupes

¹³ Prendre la main signifie devenir le vendeur, ce qui implique une transaction sans échange *carte-carte* : l'acheteur ne propose que des pièces et devient alors vendeur, car il possède une carte de plus que les autres participants.

¹⁴ Former une alliance entre acheteurs consiste à ne proposer que des pièces au vendeur, afin de changer de vendeur (le vendeur étant souvent considéré comme menant le jeu).

Des scripts ont également été codés et lancés sur les traces pour examiner les types d'échanges et les intervalles entre ces échanges. Aucune régularité n'a été relevée. Le diagramme de la FIG. 25, présente une vente réalisée durant une des parties effectuées. Les marques indiquent le type d'échange, l'axe des ordonnées est utilisé pour indiquer l'intervenant (Participant 1, Participant 2, etc.) et l'axe des abscisses indique le temps.

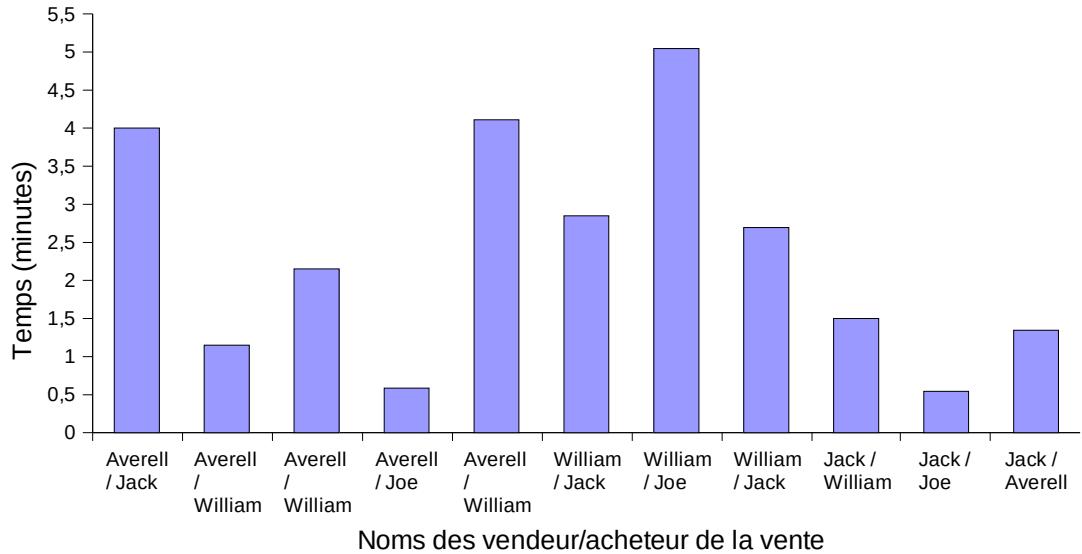


FIG. 24 - Durée des ventes pendant une partie

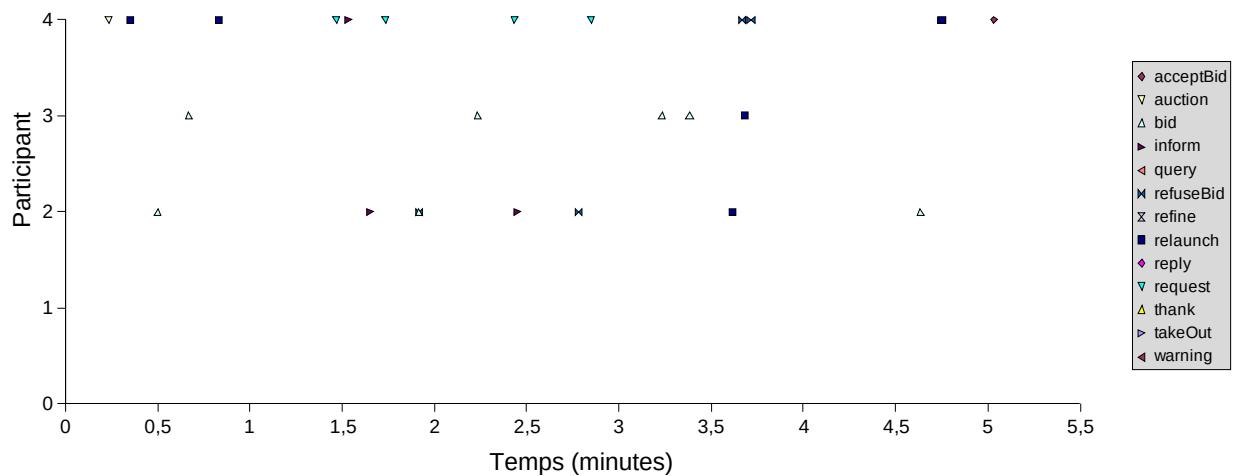


FIG. 25 - Types d'échanges entre les participants durant une vente

Cette étude nous a permis de faire un premier pas vers la compréhension de la dynamique des interactions entre les protagonistes d'une négociation. Nous proposons dans la partie 5. Modèle proposé, la sémantique des performatives relevées, et les automates d'interactions les utilisant.

Nous allons maintenant mettre en évidence les manques des protocoles existants basés sur le Contract Net, pour modéliser ce que nous avons observé.

4.5. Critique des modèles de négociation fondés sur le Contract Net

Avec le réseau contractuel, Smith définissait un modèle de distribution des tâches fondé sur des appels d'offres dans des marchés publics. Ce qui explique qu'il ne convient pas forcément pour traiter les interactions dans des négociations. Beaucoup de protocoles se basent pourtant sur le Contract Net itéré. Nous prenons le protocole utilisé dans GENCA, pour montrer qu'il ne convient pas pour modéliser ce que nous avons observé. Nous utilisons un exemple de négociation extrait d'une des traces de notre expérimentation.

Nous choisissons le protocole de GENCA car il fait partie d'un ensemble important des protocoles issus d'améliorations du Contract Net, et que parmi ces protocoles, il est relativement récent. C'est d'autre part un protocole utilisé dans une plate-forme qui se veut générique, nous pourrions donc *a priori* l'utiliser pour modéliser ce que nous avons observé.

Verrons précise que son modèle ne convient pas pour certaines formes de négociation comme les négociations combinées, les négociations multi-niveaux et la négociation par argumentation (Verrons, 2004, p.157).

Nous avons choisi un extrait ne contenant pas d'arguments, et se limitant à des échanges de types proposition, demande de modifications, refus, acceptation. La conversation est donc *a priori* modélisable avec ce protocole de négociation (voir FIG. 26).

Nous nous intéressons aux échanges entre les agents, donc à ce qui correspond à la couche communication de GENCA (voir partie 2.2.7.5. Le protocole dans GENCA).

Les primitives disponibles dans le protocole sont les suivantes :

Primitives de l'initiateur :

- ***propose (contrat)*** : c'est la première primitive que l'initiateur envoie aux participants pour leur proposer un contrat. Le contrat contient les différentes ressources à négocier
- ***demande modification (contrat)*** : ce message indique aux participants que le contrat ne peut être conclu sous sa forme actuelle et qu'il faut le modifier
- ***confirme (contrat)*** : ce message indique aux participants que le contrat est confirmé. La négociation a été un succès
- ***annule (contrat)*** : ce message indique aux participants que le contrat est annulé. La négociation a échoué

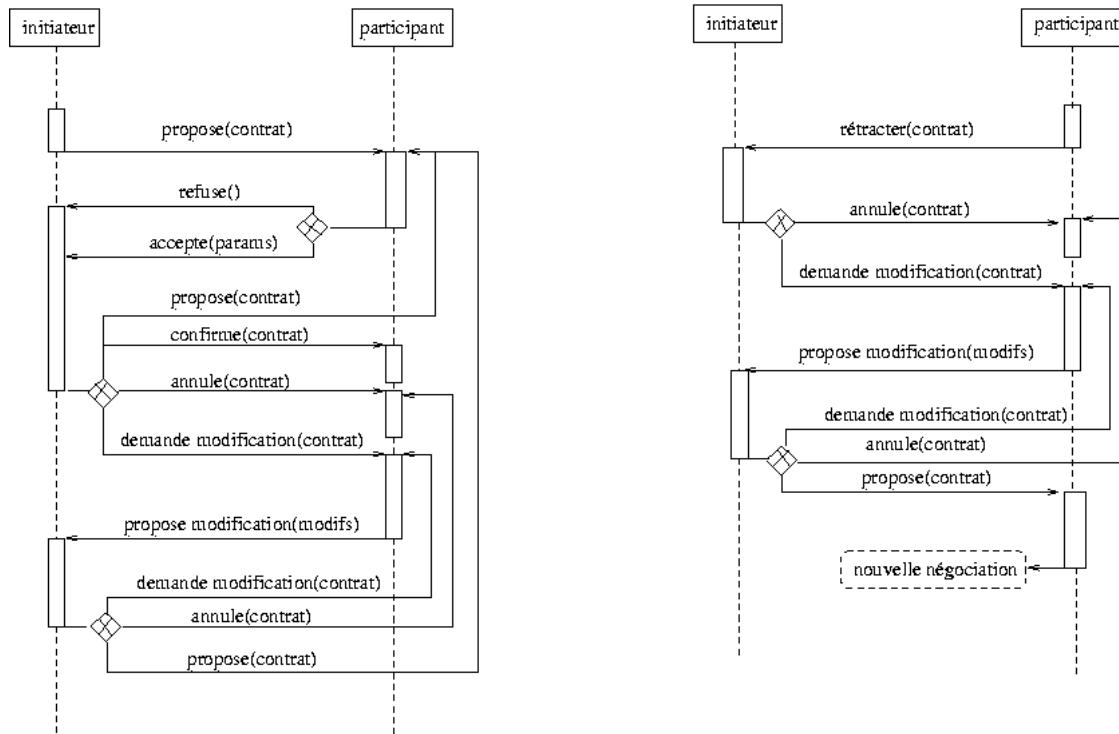


FIG. 26 - Protocole de négociation dans GENCA

Primitives du participant :

- **accepte (paramètres)** : ce message répond à la proposition de contrat faite par l'initiateur
- **refuse** : ce message répond à la proposition de contrat faite par l'initiateur. Le participant indique à l'initiateur qu'il refuse le contrat
- **propose modification (liste modifications)** : ce message répond à une demande de modification de la part de l'initiateur

Primitive commune :

- **rétractation (contrat)** : le contrat avait été confirmé mais le participant ou l'initiateur ne peut (ou ne veut) plus l'honorer. Il décide donc de se rétracter.

Dans ce protocole, le début de la négociation est la constitution du contrat : c'est l'initiateur qui fixe le contrat, et le propose au participant.

Dans notre expérimentation, nous avons observé plusieurs cas :

- la discussion de négociation commence avant la mise en vente de la carte, et la vente de la carte termine la négociation,
- la discussion de négociation commence avant la mise en vente de la carte, et se poursuit après,
- la discussion de négociation commence après la mise en vente de la carte.

La carte peut être mise en vente sans qu'il n'y ait aucune proposition de la part d'un acheteur, la négociation n'a pas lieu entre le vendeur et cet acheteur.

La mise en vente de la carte n'est donc pas considérée comme faisant partie d'une discussion de négociation : il s'agit d'une action n'entraînant pas forcément de processus de négociation. D'autre part, nous avons observé que le vendeur n'était pas forcément l'initiateur de la négociation, on trouve les deux cas : aussi bien le vendeur que l'acheteur peuvent débuter la négociation (voir partie 4. Analyse des traces).

Voici l'extrait d'une de nos traces choisi :

```
[10:41:49] 322, William AUCTION vin
[10:42:03] 323, Averell PROPOSE (William, COINS -, CARD vin, Averell, COINS 2, CARD thé)
[10:42:04] 324, William SAY alors qui commence
[10:42:11] 325, Jack PROPOSE (William, COINS -, CARD vin, Jack, COINS 10, CARD -)
[10:42:13] 326, Averell SAY 2 + une carte
[10:42:23] 327, Joe PROPOSE (William, COINS -, CARD vin, Joe, COINS 1, CARD -)
[10:42:24] 328, Jack SAY 10 p
[10:42:31] 329, Averell SAY oh la la
[10:42:34] 330, Joe SAY oj
[10:42:36] 331, Joe SAY oh
[10:42:48] 332, Joe SAY william va devenir riche
[10:42:50] 333, Joe SAY !
[10:42:52] 334, Jack SAY aller william
[10:43:21] 335, Joe SAY william a peur :D
[10:43:24] 336, William SAY tu peux changé la carte averell, j'accepte 2 + un carte miel
[10:43:34] 337, Joe SAY il se dit qu'est-ce qui se manigance !
[10:43:54] 338, Jack TAKE_OUT
[10:44:05] 339, Averell SAY je prpose un echange direct
[10:44:12] 340, Averell PROPOSE (William, COINS -, CARD vin, Averell, COINS -, CARD miel)
[10:44:23] 341, Joe SAY je te propose un échange de carte william
[10:44:27] 342, William SAY c'est pas suffisant
[10:44:28] 343, Joe SAY alors tu veux quoi
[10:44:37] 344, Averell SAY une carte miel
[10:44:47] 345, Joe SAY william tu veux quoi
[10:45:03] 346, Averell SAY laisse le respirer
[10:45:09] 347, Averell SAY ;)
[10:45:16] 348, Joe SAY je te propose une carte + des sous ;)
[10:45:31] 349, William SAY allez averell ajouter un peu
[10:45:32] 350, Joe SAY alors ?
[10:45:36] 351, ACCEPT_SELL (William, COINS -, CARD vin, Averell, COINS -, CARD miel)
[10:45:39] 352, William SAY vendu
[10:45:43] 353, Joe SAY :o
[10:45:46] 354, Averell SAY merci
```

Si l'on observe la négociation entre Averell (un acheteur) et William (le vendeur), on obtient la séquence suivante :

```

323 : Averell fait une proposition
336 : William demande de modifier la proposition
340 : Averell fait une autre proposition
342 : William la refuse
349 : William demande de modifier la proposition (d'augmenter la somme)
351 : William accepte la proposition
Averell (acheteur) : bid
William (vendeur) : request
Averell (acheteur) : bid
William (vendeur) : refuse
William (vendeur) : request
William (vendeur) : accept

```

Le protocole de GENCA ne permet pas de réaliser cet échange.

- D'une part, de par la distinction initiateur/participant dans les échanges :

Pour réaliser le dialogue entre William et Averell, deux cas s'offrent à nous :

- Si nous posons que l'initiateur est le vendeur, nous devons prendre en compte la mise en vente de la carte, en commençant par un message *propose (contrat)* de William (vendeur). La mise en vente de la carte ne constitue pas un début de négociation. La suite du dialogue n'est pas modélisable : Averell (acheteur) ne peut pas faire de proposition. S'il utilise *accepte* et ses paramètres pour faire une proposition, il peut poursuivre le dialogue. William (vendeur) fait alors une demande de modification du contrat avec *demande modification (contrat)*. La deuxième proposition d'Averell (acheteur) est alors envoyée via l'acte de langage *propose modifications (liste modifications)*. La suite du dialogue n'est pas modélisable : William (vendeur) ne peut refuser. S'il utilisait à nouveau *demande modification (contrat)* et que l'on considère que le refus est *sous-entendu*, le dialogue peut se poursuivre, mais sur un envoi de message seulement. En effet, la demande de modification suivante n'est plus possible (voir le schéma du protocole FIG. 26), et l'acceptation qui suit non plus. Le protocole ne permet pas d'envoyer ces types de messages de façon consécutive,
- si nous posons que l'initiateur est l'acheteur, Averell (acheteur) commence par un message *propose (contrat)*. La suite du dialogue n'est pas modélisable : William (vendeur) ne peut pas faire de demande de modification. L'acte de langage *accepte* comporte des paramètres et pourrait permettre une proposition (et par la suite une contre-proposition) de la part du participant. Mais le sens de *accept* est alors ambigu : ce n'est pas le contrat original proposé qui est alors accepté, mais une modification non encore validée par l'autre partie (l'initiateur).

Nous soutenons que ce qui diffère dans les actes de langage utilisés par les deux protagonistes, ne doit pas être fonction de celui qui entame la négociation, mais du rapport de

force qui existe entre les deux parties. Dans notre modèle, cela va dépendre du rôle de chacun (vendeur/acheteur, patron/employé, etc.). C'est ce que nous avons observé avec les rôles vendeur/acheteur.

- Le protocole utilisé dans GENCA n'autorise pas certaines **séquences** comme *refuser-accepter* ou *demande de modifier-accepter*. Cela se produit pourtant : nous avons observé que le vendeur pouvait accepter une proposition qu'il a tout d'abord refusée, ou une proposition dont il a demandé la modification sans réussir à l'obtenir. Ses deux interventions (*refuser*, *accepter*) sont donc consécutives dans le dialogue : sa décision a changé sans nouvelle intervention de l'interlocuteur (l'acheteur). Sa décision peut, en effet, changer en fonction des interventions des autres protagonistes (les autres acheteurs), mais également par rapport au temps qui s'écoule et qui peut avoir une répercussion sur sa prise de décision. Cela doit être géré au niveau du raisonnement de l'agent, et les actes de langage disponibles ne doivent pas contraindre les interventions d'un agent, ou lui en interdire.
- Comme d'autres protocoles, le protocole utilisé dans GENCA propose des réponses *par défaut*. En cas de non réponse durant des enchères par exemple, la réponse par défaut sera un refus (Verrons, 2004, p.81). Le modèle BDIGGY permet de prendre en compte l'absence de réponse, grâce aux automates temporisés (Pauchet, 2006). Nous l'avons observé chez les humains : l'absence de réponse, le refus de participer à la négociation et le retrait explicite des négociations, sont une réalité. Utiliser une réponse par défaut s'oppose à notre démarche de rendre les agents logiciels *plus humains*. Nous proposons un protocole plus souple, plus proche de ce que nous observons dans les négociations humaines.
- Une autre critique : l'initiateur est obligé d'*annuler* un contrat, et ne peut *refuser* une proposition du participant. De notre point de vue, le contrat est le résultat de la négociation, et les modalités du contrat sont discutées pendant la négociation. Or ici, l'initiateur décide du contrat, et ne peut recevoir de propositions de modifications s'il ne les a pas lui-même demandées. Lorsqu'il en reçoit, il ne peut pas indiquer simplement son accord ou son refus : soit il propose un nouveau contrat modifié au participant (qui l'accepte ou le refuse), soit il décide d'*annuler* le contrat, soit il demande de nouvelles propositions de modifications. En cas d'*annulation*, il a la possibilité de proposer un nouveau contrat. D'un point de vue conceptuel, nous pensons que si les protagonistes continuent à négocier les mêmes ressources, ils devraient pouvoir modifier le contrat en cours, et non pas devoir en élaborer un nouveau.
- Également d'un point de vue conceptuel : dans le cas d'une vente aux enchères, lorsque le vendeur trouve un accord avec l'un des acheteurs, il doit annuler des

contrats créés avec les autres acheteurs. Un contrat devrait être créé uniquement avec l'acheteur avec qui il fait affaire. Le vendeur pourrait alors prévenir les autres acheteurs qu'il a trouvé à vendre son produit, et ainsi mettre un terme à la négociation entamée avec eux, sans avoir à effectuer des ruptures de contrats avec chacun.

Dans ce protocole, seul l'initiateur crée et modifie les contrats. Comme le montre (Baker, 1994, p.5), la négociation exige une interaction quasi-égalitaire. Ce n'est pas le cas ici : le participant peut seulement accepter ou refuser un contrat, et proposer des modifications seulement si l'initiateur le lui demande. Il manque des interactions permettant la **co-construction du contrat**.

Il manque à l'*initiateur* la possibilité de refuser les modifications du participant (il peut accepter avec le *confirme*), et la possibilité de lui proposer des modifications.

Il manque au *participant* la possibilité d'interventions situées **entre l'acceptation et le refus**. C'est là que se situe précisément la négociation : le dialogue qui est permis entre le « oui » et le « non », entre l'accord et le désaccord. Il lui manque la possibilité de proposer des modifications de sa propre initiative, celle d'indiquer la raison de son refus, celle enfin de décider de ne pas entrer dans la négociation (le droit de ne pas participer).

Il ne s'agit pas de permettre aux agents de ratifier un contrat, mais bien de leur fournir la possibilité de le négocier.

5. Modèle proposé

« Je connais un critique qui est en même temps auteur...
ce qui le met en tant qu'auteur dans une situation critique ! »

Raymond Devos

Cette partie contient tout d'abord les modifications d'un point de vue général du modèle BDIGGY, pour pouvoir modéliser les interactions lors de négociation. Nous définissons ensuite le prédicat transaction, objet de la négociation pour notre application de jeu de négociation. Nous expliquons les éléments qui existent déjà dans le modèle et que nous utilisons pour la négociation, puis détaillons les nouveaux éléments, ajoutées au niveau du module de communication : les 10 performatives (dont 3 propres au domaine d'application) et les 5 paires d'automates (dont 1 propre au domaine d'application). Nous poursuivons en expliquant nos choix de conception, puis nous détaillons la sémantique ajoutée. Les adaptations de l'architecture de l'agent sont ensuite présentées. Nous terminons sur le parallèle qui peut être fait entre notre modélisation et les outils de la théorie des jeux.

5.1. Les agents de BDIGGY : de la coopération inconditionnelle à la négociation

Notre situation de problème (la négociation) diffère beaucoup de celle pour laquelle BDIGGY a été conçu (résolution coopérative de problème), et c'est d'ailleurs pour cela que nous l'avons choisi. Il nous a donc fallu adapter l'agent BDIGGY ; nos apports et modifications sont les suivants :

- la **communication** est publique et directe. Dans une situation comme celle de l'agence de voyage, les agents communiquent deux à deux par mail. Pour un agent, les messages qui circulent entre les deux autres agents, ne sont pas visibles. Il n'en a pas besoin, car chacun résout son propre problème. Dans la situation de négociation choisie, tous les échanges sont visibles par tous les agents, (ce qui est permis par le *chat*, dans l'application).
- l'**évolution** de certaines croyances. A la différence des croyances sur les horaires des trains par exemple, les croyances sur les désirs des agents évoluent sans cesse durant les négociations.
- le **rapport de force**. Pour un agent, la différence se situe au niveau du traitement des désirs des autres agents. Les agents ne traitent plus les désirs des autres agents comme les leurs. C'est comme l'agent au guichet de la gare, qui recherchera l'information que lui demande une personne, sans se poser de question : on peut considérer que la coopération se fait à 100%. Les deux personnes (l'employé au guichet, et la personne

lui demandant une information) ont les mêmes intérêts : obtenir l'information. Au niveau du modèle, l'*agent employé* prend le désir de l'*agent demandeur*, et le traite (car lui a accès aux informations nécessaires). Il traite ainsi ce désir *comme si c'était le sien*. La négociation est une forme de coopération, mais les agents n'ayant pas les mêmes intérêts, ils ne font *que* tenir compte des désirs des autres agents, pour la génération des leurs.

5.1.1. L'objet de la négociation

L'objet des négociations entre les participants porte sur les modalités d'une vente de carte marchandise. Il s'agit de déterminer qui sera l'acheteur et quelles seront les valeurs *carte-pièces* échangées. Nous avons appelé pTrans (pour *prédictat Transaction*) le prédictat définissant ces valeurs.

5.1.1.1. Prédictat Transaction pTrans

`pTrans(a1, coins_nb_1, typeCard_1, a2, coins_nb_2, typeCard_2)`

a1 : agent vendeur

coins_nb_1 : nombre de pièces éventuellement données par a1 en plus de la carte vendue

typeCard_1 : type de la carte vendue par a1

a2 : agent acheteur

coins_nb_2 : nombre de pièces données par a2 pour acheter la carte en vente

typeCard_2 : type de la carte éventuellement donnée par a2 en plus des pièces pour acheter la carte en vente.

Voici des exemples de son utilisation :

Averell est vendeur :

Averell : qui veut quoi ?

= Averell demande qui a le désir de lui acheter quelque chose, et quoi

(query(Averell, all, pD(?, pTrans(Averell, ?, ?, ?, *, *))))

Joe : je suis intéressé par des épices

= Joe répond qu'il a le désir d'acheter des épices à Averell

reply(Joe, Averell, pB(Joe, pTrans(Averell, 0, SPICES, Joe, *, *)))

Averell : qui est intéressé par du thé ?

= Averell demande qui a le désir de lui acheter du thé

query(Averell, all, pD(?, pTrans(Averell, *, TEA, ?, *, *)))

Joe : moi non

= Joe répond qu'il n'a pas le désir d'acheter du thé à Averell

reply(Joe, Averell, !pD(Joe, pTrans(Averell, *, TEA, Joe, *, *)))

Averell : Jack est-ce que le thé t'intéresse ?

= Averell demande à Jack s'il a le désir de lui acheter du thé

query(Averell, Jack, pD(Jack, pTrans(Averell, *, TEA, Jack, *, *)))

Averell : Jack, qu'est-ce qui t'intéresse ?

= Averell demande à Jack s'il a le désir de lui acheter quelque chose, et quoi

query(Averell, Jack, pD(Jack, pTrans(Averell, ?, ?, Jack, *, *)))

Averell : qui me propose quoi contre du café ?

= Averell demande qui a le désir de lui acheter du café et contre quoi ?

query(Averell, all, pD(?, pTrans(Averell, ?, COFFEE, ?, ?, ?)))

Averell : qui me propose du thé, contre du café ?

= Averell demande qui a le désir de lui acheter du café contre du thé

query(Averell, all, pD(?, pTrans(Averell, *, COFFEE, ?, *, TEA)))

Averell : Jack, qu'est-ce que tu me proposes, contre du café ?

= Averell demande à Jack s'il a le désir de lui acheter du café, et contre quoi

query(Averell, Jack, pD(Jack, pTrans(Averell, ?, COFFEE, Jack, ?, ?)))

Averell : Jack, est-ce que tu peux me proposer du vin, contre du café ?

= Averell demande à Jack s'il a le désir de lui acheter du café contre du vin

query(Averell, Jack, pD(Jack, pTrans(Averell, *, COFFEE, Jack, *, WINE)))

5.1.2. Les connaissances de l'agent

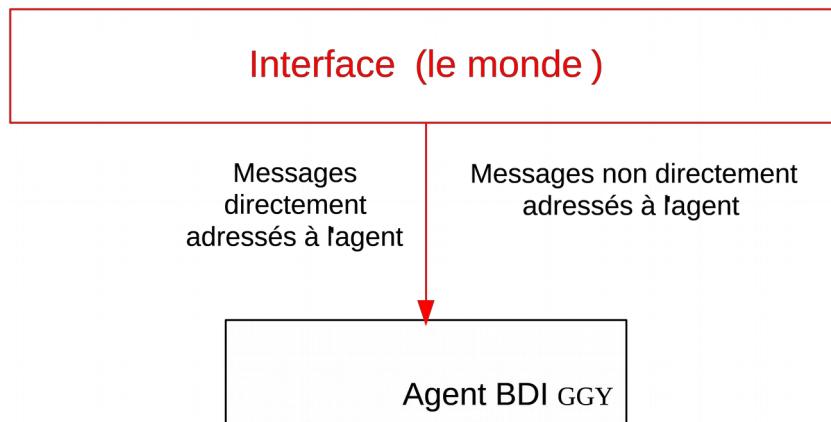


FIG. 27 - Les sources de connaissances de l'agent

Les connaissances de l'agent BDIGGY, formalisées par les prédictats croyances pB , proviennent de la perception de l'agent (voir FIG. 27) :

- d'une part les messages qui lui sont directement adressés, et qui seront traités via les automates,
- et d'autre part les messages défilant également dans le *chat*, qui ne lui sont pas directement adressés. Ils les lira ou non (en fonction de son attention), et en tiendra compte ou non dans ses prises de décision.

5.2. Modélisation des interactions

« quand dire, c'est faire plusieurs choses à la fois ;
quand dire, c'est faire une chose sous les apparences d'une autre. »

Dictionnaire d'analyse du discours
Patrick Charaudeau, Dominique Maingueneau

5.2.1. Les performatives

Comme lors de la conception de BDIGGY, l'analyse et la modélisation des interactions ont été réalisées aux niveaux des énoncés et du discours.

Au niveau des énoncés, chaque message correspond à une performative appliquée à un état mental. Au niveau du discours, tous les messages en relation avec un message initiateur sont

regroupés en une seule intervention (voir FIG. 28) dont l'intention discursive est donnée par le type du message initiateur (Pauchet, 2006).

Type d'échange	Intention discursive	Performative initiatrice	Performative de clôture
demande d'info.	directive	query	reply / -
discussion	engaging of the buyer	bid	acceptBid / leave
	directive of the seller	request	acceptBid / leave
envoi spontané d'info.	descriptive	inform	inform
relance	directive	relaunch	relaunch
mise en garde	descriptive	warning	warning
mise en vente	engaging	auction	takeOut / sold

FIG. 28 - Types d'échanges

En plus des performatives définies lors de l'analyse (voir tableau FIG. 22, partie 4.3. Les échanges) nous avons ajouté deux performatives :

- *sold*, qui est propre à l'application et permet au vendeur de prévenir les autres acheteurs qu'il a accepté la proposition d'un acheteur, et que la vente est terminée,
- *leave*, qui permet à un agent de quitter un automate de discussion.

5.2.2. Les automates de conversations

La force de la modélisation des interactions par des automates, est de pouvoir simuler plusieurs échanges simultanément. Ce qui est nécessaire pour modéliser tous les échanges observés entre les agents.

Les nouveaux automates du modèle ne sont pas *temporisés*. La temporisation est utile durant certains échanges, comme la demande d'information (où il est nécessaire, en cas de non réponse, de pouvoir fermer les automates : voir Le modèle de la dynamique des échanges dans 1.2. Le système BDIGGY). Par contre dans les situations observées, la communication permet clairement d'indiquer lorsque les échanges concernant la négociation se terminent. Un acheteur peut explicitement quitter la vente, tous sont prévenus lorsque le vendeur fait affaire avec un acheteur, etc., (voir partie 5.2.4. Choix de conception).

5.2.2.1. Automates de BDIGGY utilisés

Deux automates déjà présents dans le modèles sont utilisés pour la négociation : l'automate d'envoi d'information spontané (FIG. 29) et l'automate de demande d'information (FIG. 30 et 31). Le second n'est utilisé qu'en partie, les transitions utilisées sont indiquées en rouge sur les schémas.

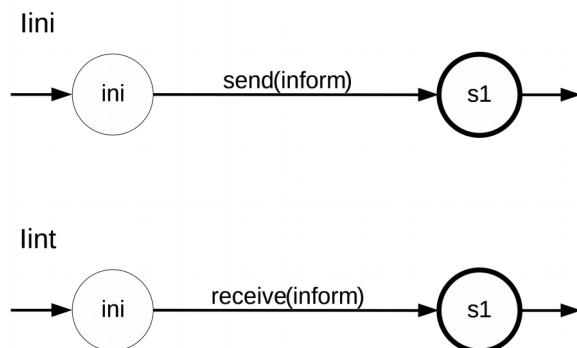


FIG. 29 - Automates d'envoi spontané d'information

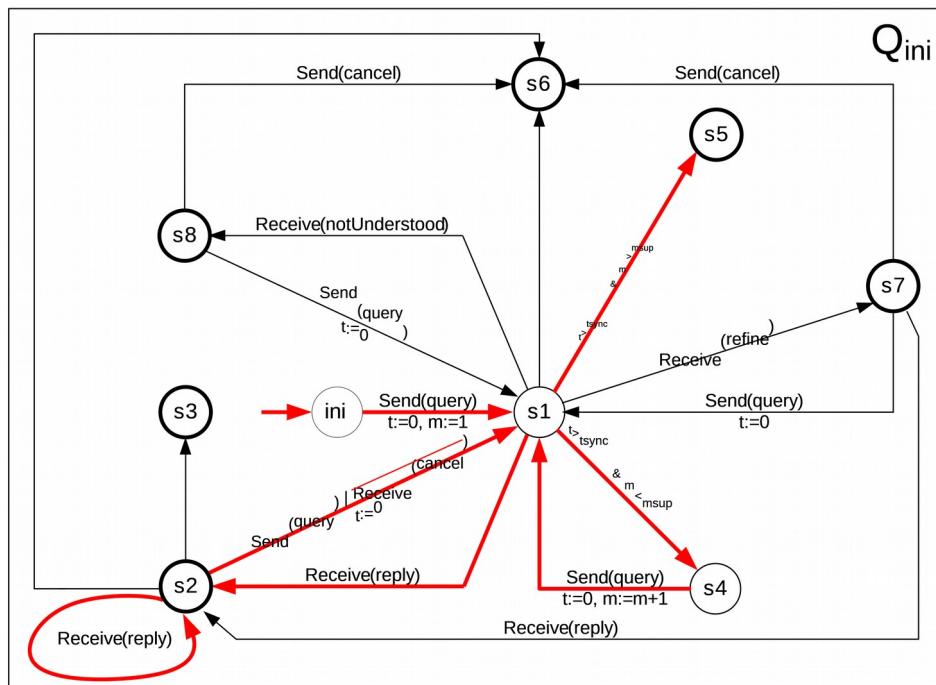


FIG. 30 - Automate de l'initiateur de la demande d'information

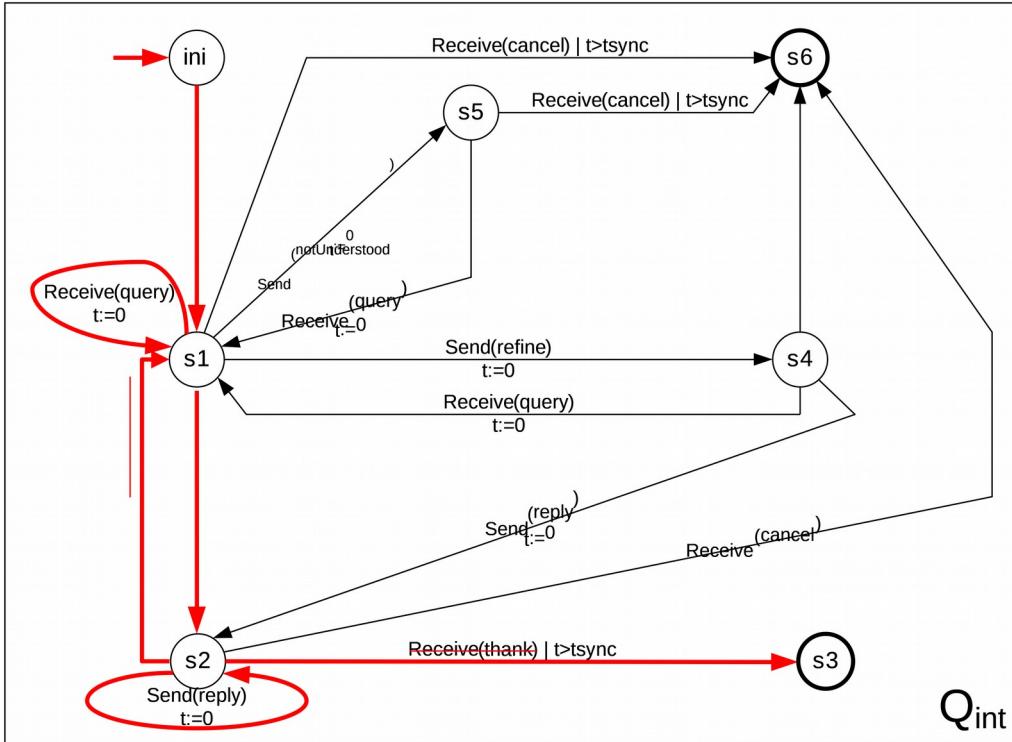


FIG. 31 - Automate de l'agent recevant la demande d'information

Notons que nous n'avons pas observé de *refine* correspondant à un désir de l'agent émetteur de voir une étape du voyage raffinée. Par exemple :

De : Agent routier A : Agent arien
 » Y a-t-il des cars de Montpellier à
 » Maguelone ?
 Il y a des cars mais quel horaire ?

pMessage(road air refine pD(road pD(air pStage(Montpellier Maguelone ? ? car * * *)))

5.2.2.2. Automates de discussion

Comme nous l'avons expliqué dans la partie analyse (voir partie 4.1.1.3. Les échanges), la discussion de négociation débute soit par une intervention de l'acheteur, soit par une intervention du vendeur. Pour modéliser ce phénomène, nous avons ajouté deux types d'automates. Les FIG. 32 et 33 montrent la paire d'automates utilisée dans le premier cas : ces automates sont ouverts lorsqu'un acheteur fait une offre (envoi de la performative *bid*) au vendeur. Les FIG. 34 et 35 correspondent au second cas : lorsque le vendeur demande à un acheteur de faire une offre (envoi de la performative *request*). Seuls la performative d'ouverture et l'état courant des automates après l'envoi/la réception de cette performative (état s1 ou s2), diffèrent, dans ces deux paires d'automates.

Dini Buyer

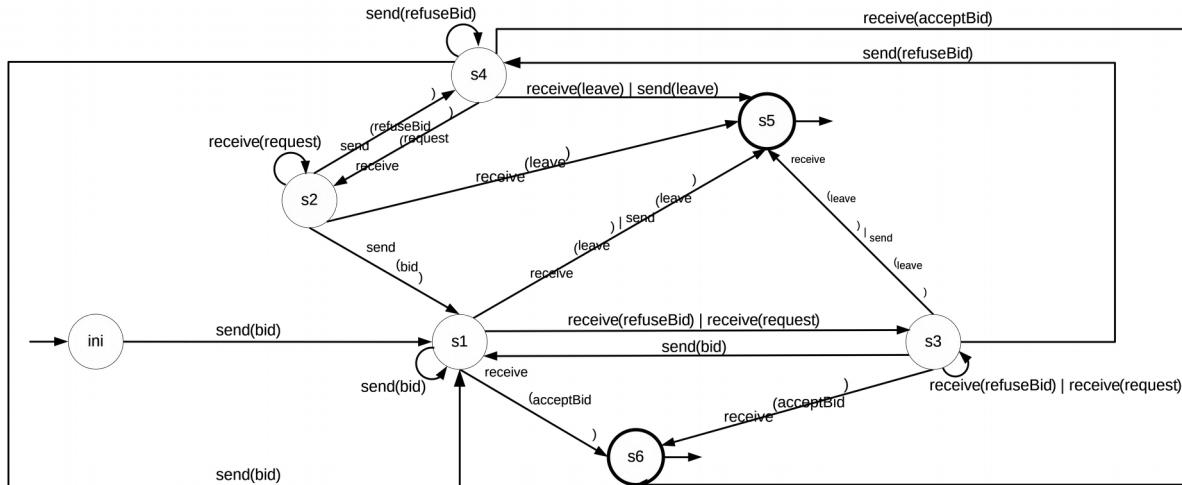


FIG. 32 - Automate de discussion D_{int} de l'acheteur

La conception de ces automates s'est faite de façon *incrémentale*, en parcourant les traces obtenues lors des expérimentations. On peut noter par exemple la séquence possible *receive(refuse) - receive(accept)*, observée dans les négociations humaines, qui est un exemple de séquence qu'on ne retrouve pas dans les protocoles de négociation existants.

Dint Seller

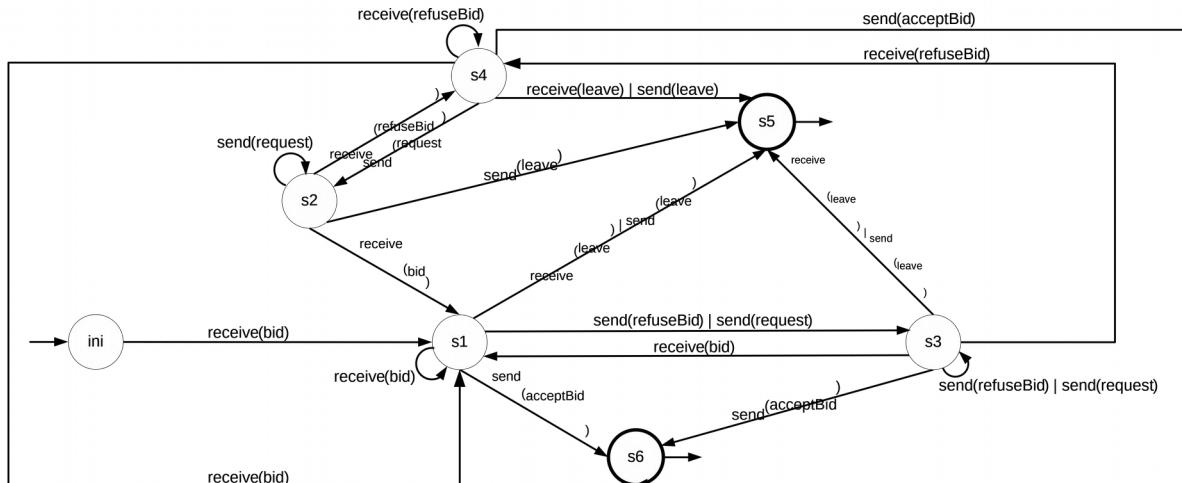


FIG. 33 - Automate de discussion D_{int} du vendeur

Dini Seller

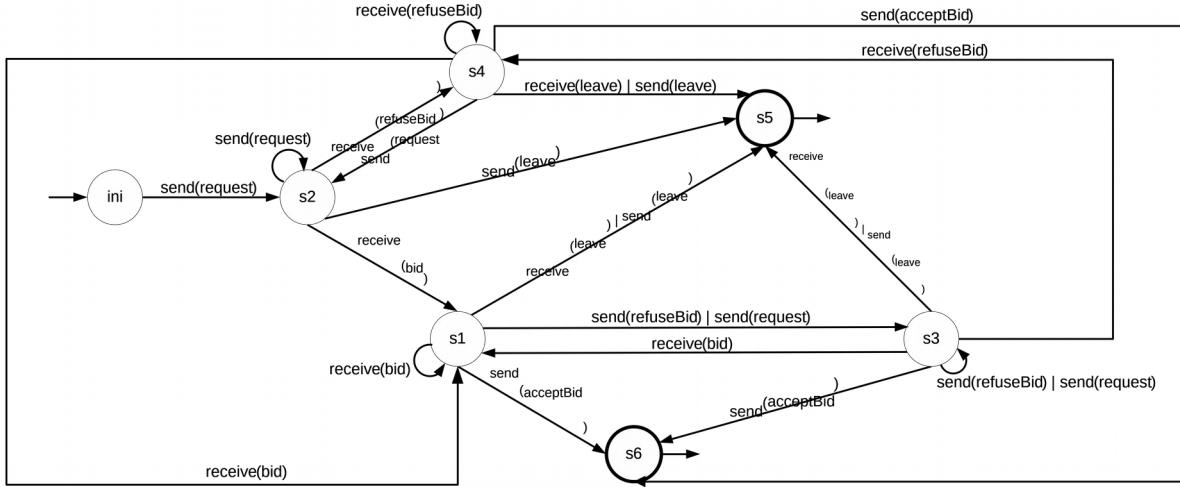


FIG. 34 - Automate de discussion D_{ini} du vendeur

Dint Buyer

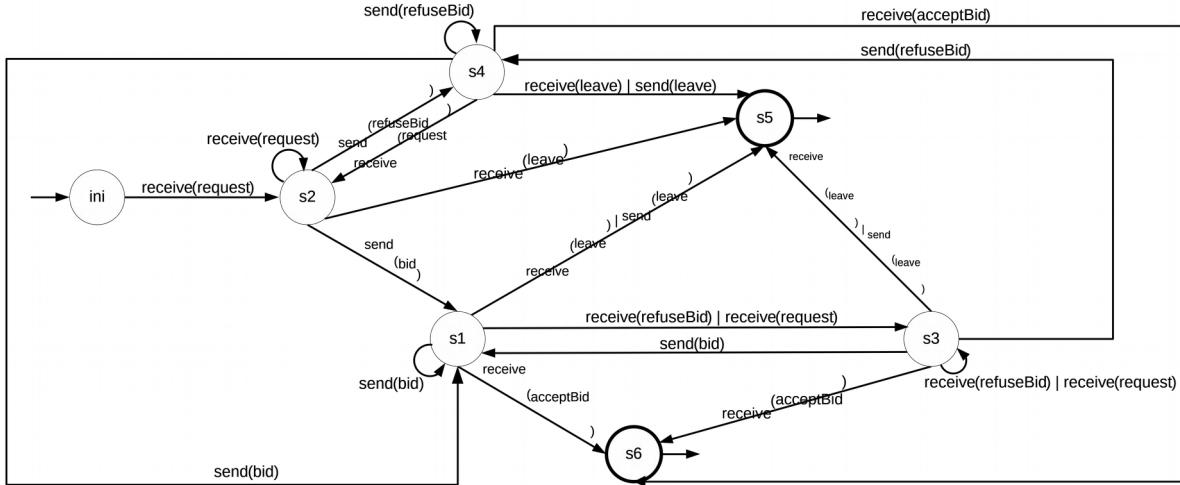


FIG. 35 - Automate de discussion D_{int} de l'acheteur

5.2.2.3. Automate de relance

L'automate de relance (voir FIG. 36) est très simple : il ne contient que deux états et une transition. En effet les messages de relance ont été observés à la fois au cours des discussions entre acheteurs et vendeur, et entre des acheteurs. L'initiateur du *relaunch* a même parfois quitté la vente, et souhaite que les autres se dépêchent (voir partie 4.1.1.3. Les échanges). Il a donc fallu rendre ce type de message disponible pour les agents, indépendamment des autres échanges.

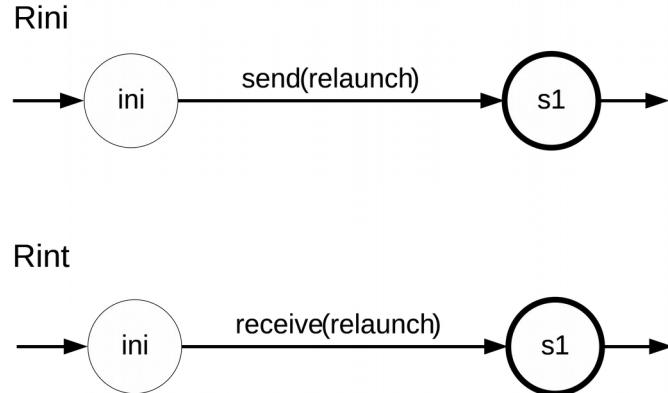


FIG. 36 - Automates de relance

5.2.2.4. Automate de mise en garde

L'automate de mise en garde (FIG. 37) comporte une seule transition permettant l'envoi/la réception du message de mise en garde. Nous avons observé les mises en garde parfois en dehors des négociations. Comme pour la relance, nous avons conçu l'automate de mise en garde de façon à ce qu'il soit possible pour un agent, d'envoyer une mise en garde (*warning*) aux autres agents, indépendamment des autres échanges au cours des négociations.

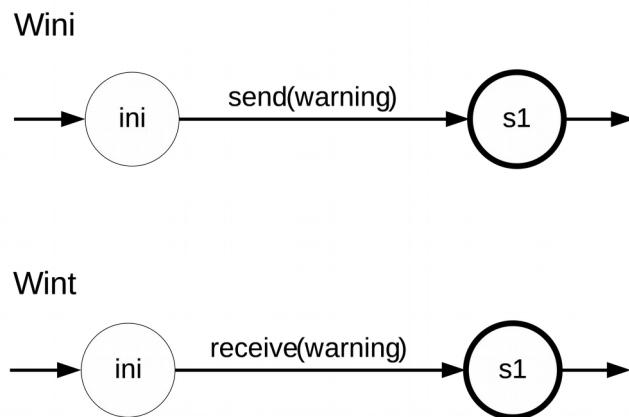


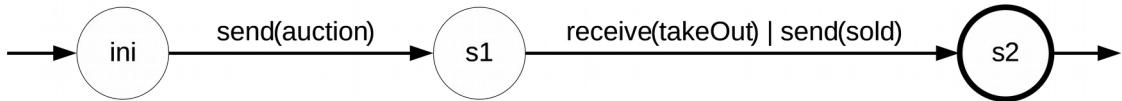
FIG. 37 - Automates de mise en garde

5.2.2.5. Automate de mise en vente

L'automate de mise en vente (voir FIG. 38), et les trois performatives *auction*, *takeOut* et *sold*, sont propres à l'application de commerce électronique. Il est ouvert par chaque agent acheteur à la réception d'une performative *auction*, et par l'agent vendeur, à chaque envoi d'un *auction*. Ce dernier a ainsi un automate A_{ini} ouvert, par acheteur participant. Les automates A_{ini} et A_{int} entre un agent vendeur et un agent acheteur, sont fermés lorsque l'acheteur se retire de

la vente en envoyant un *takeOut*, ou lorsque le vendeur termine la vente, en envoyant un message *sold* (vendu) à l'acheteur.

Aini



Aint

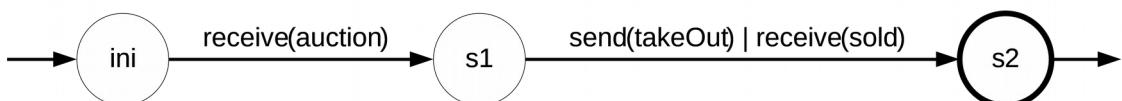


FIG. 38 - Automate de mise en vente

Nous avons choisi de dissocier ce qui tenait à la vente en ajoutant un *automate de mise en vente* et les deux performatives *takeOut* et *sold*. Ces trois éléments sont propres à l'application, et non au processus de négociation. L'automate de discussion $D_{\text{ini}}/D_{\text{int}}$ peut ainsi être utilisé de façon indépendante, sur d'autres applications. On peut par exemple imaginer une prise de rendez-vous, qui réutiliserait les automates proposés précédemment.

5.2.3. Un exemple

Le schéma de la FIG. 39 donne un exemple d'utilisation des automates. Lors de la mise en vente, l'envoi de la primitive *auction* provoque l'ouverture de l'automate A_{ini} de *mise en vente* pour le vendeur, et sa réception l'ouverture de l'automate A_{int} pour chaque agent acheteur. Dans la situation représentée, la mise en vente a déjà eu lieu, et des discussions de négociation sont en cours : entre le vendeur et l'acheteur 1, et entre le vendeur et l'acheteur 3.

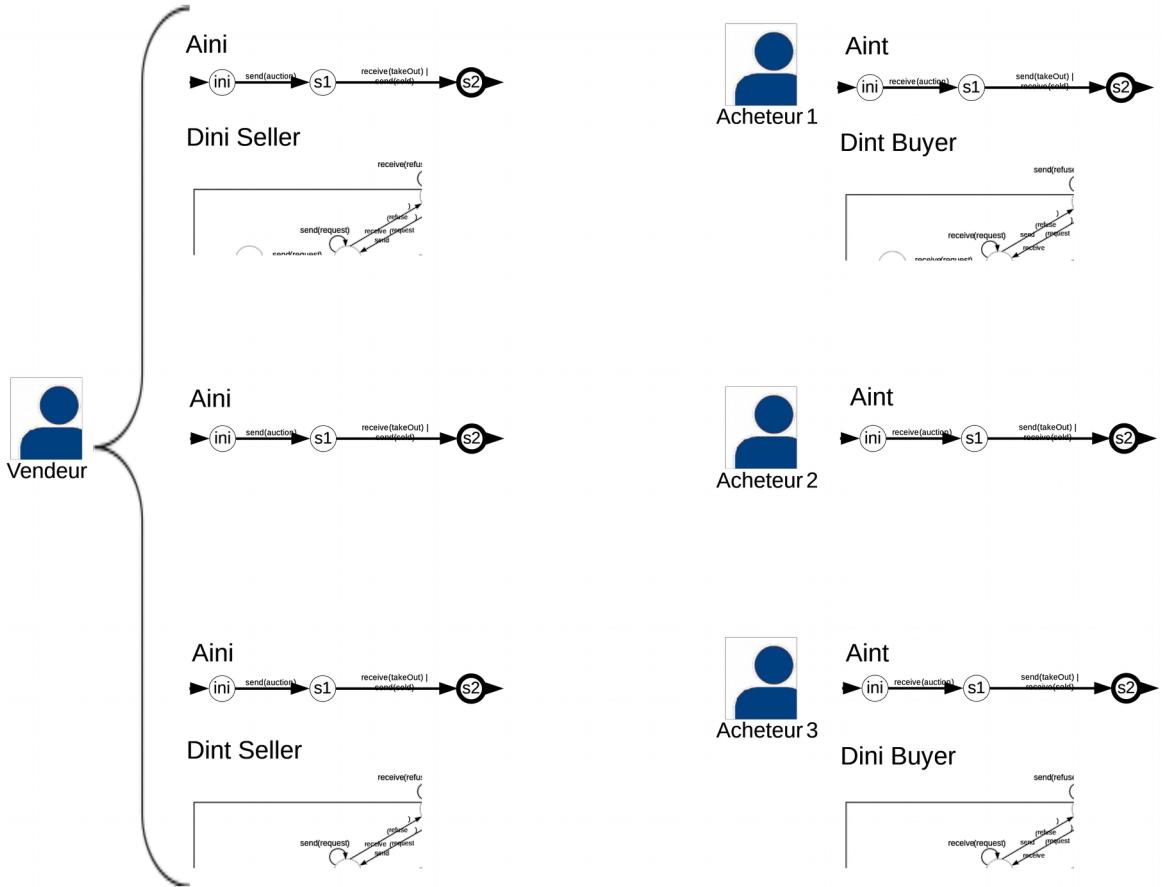


FIG. 39 - Un exemple de situation d'interaction entre les participants

La première a été initiée par le vendeur, et la seconde par l'acheteur 3. D'autres automates peuvent évidemment être ouverts et utilisés parallèlement. Par exemple si l'acheteur 1 s'impatiente et qu'il relance le vendeur, il ouvrira un troisième automate, l'automate de relance R_{ini} . Le vendeur ouvrira alors l'automate symétrique A_{int} , et ainsi de suite.

5.2.4. Choix de conception

Le modèle proposé a été réalisé en fonction des observations faites durant la phase d'analyse. La modélisation sous forme d'automates a été améliorée à plusieurs reprises. Un problème s'est notamment posé au niveau de la fermeture des automates de mise en vente et de discussion.

Un participant quitte les négociations, dans les cas suivants :

- le vendeur a trouvé un acheteur avec qui il fait affaire, la transaction est effectuée et il prévient les autres acheteurs qu'il a vendu son produit. L'automate de discussion ouvert avec cet acheteur est fermé par la performative *acceptBid*.
- un acheteur souhaite quitter la vente, il n'est plus intéressé par effectuer une transaction avec le vendeur, deux cas se présentent :

1. il était en train de négocier avec le vendeur et un automate de discussion était ouvert avec lui,
2. il n'était pas en train de négocier, et seul l'automate de mise en vente était ouvert.

Dans le premier cas, les discussions prennent fin sur l'initiative du vendeur, qui a trouvé un acheteur à qui il vend son produit. Dans le second cas, un acheteur exprime son retrait de la discussion et/ou de la vente (certains quittent la vente sans avoir négocié).

Dans les deux cas se posait la question de la fermeture des automates. En effet, que la négociation se termine sur une initiative du vendeur ou d'un acheteur, il s'agissait de fermer les automates correspondants aux conversations qui n'étaient plus d'actualité.

Pour palier ce problème, nous avons ajouté la règle suivante, propre à l'application de vente :

Lorsqu'un acheteur envoie un message *takeOut*, et qu'un automate de discussion est ouvert avec le vendeur, un message *leave* lui est également envoyé de façon automatique. Ces deux performatives ferment respectivement l'automate de mise en vente et l'automate de discussion, si celui-ci est ouvert.

Lorsqu'un vendeur envoie un message *sold*, un message *leave* est automatiquement envoyé à chaque acheteur avec qui une discussion est en cours. Ces deux performatives ferment respectivement l'automate de mise en vente et l'automate de discussion (s'il est ouvert), du côté du vendeur (par l'envoi de la (des) performative(s)) et du côté de chaque acheteur participant (par la réception de la (des) performative(s)).

D'autres solutions existaient, par exemple :

- nous pouvions rendre l'automate de mise en vente récursif et mettre l'état d'entrée de l'automate de discussion dans son état *s1* (voir l'automate de mise en vente FIG. 38). Cela n'était pas satisfaisant car, comme nous l'avons expliqué, les discussions de négociation débutaient parfois avant la mise en vente. D'autre part, au niveau du modèle, cela rendait les deux automates dépendants l'un de l'autre,
- nous pouvions modéliser les interventions concernant la vente (*auction*, *sold*, ...) et les discussions dans un même automate, et n'avoir à en fermer qu'un seul. Mais là encore, cela posait un problème de manque de généricité du modèle.

Nous avons ainsi opté pour ce découpage, le choix de ces performatives, et l'ajout de cette règle, propre à l'application, afin de ne pas trahir les observations effectuées et privilégier la généricité du modèle.

5.2.5. La sémantique des performatives

Nous avons repris les notations utilisées dans (Pauchet, 2006), pour écrire la sémantique¹⁵ des performatives ajoutées au modèle :

AGT est l'ensemble des agents

MEN est l'ensemble des états mentaux de l'agent, qui contient :

- BEL : l'ensemble des croyances,
- DES : l'ensemble des désirs,
- INT : l'ensemble des intentions.

PER est l'ensemble des performatives, PRE l'ensemble des prédictats, MES l'ensemble des messages possibles.

Un message échangé entre agents est décrit par le prédicat :

$$pMessage(A_s, A_r, P, O)$$

avec $A_s \in AGT$ l'agent émetteur, $A_r \in AGT$ le destinataire, $P \in PER$ la performative du message, $O \in MEN$ l'objet auquel s'applique la performative.

Lors de la réception et de l'envoi de messages, les changements d'état dans les automates s'effectuent grâce à l'action interne $aUpdate(M)$, avec $M \in MES$. Cette action permet de vérifier dans les automates ouverts s'il y en a un dont l'état courant permet l'envoi ou la réception du message en argument. Si c'est le cas, l'état actif de l'automate est modifié en conséquence et l'automate est fermé si nécessaire. Si aucun automate ne correspond, un nouvel automate est ouvert.

Par exemple, si l'agent Joe désire vendre sa carte vin pour 3 pièces, il possède dans ses états mentaux le désir :

$$pD(Joe, pTrans(Joe, ?, WINE, ?, ?, ?))$$

Dans la notation utilisée dans le modèle, les ? indiquent que les valeurs (dans l'exemple, les pièces échangées, la carte éventuellement échangée, et l'acheteur) sont à préciser, et les * peuvent être remplacées par n'importe quelle valeur, voir (Pauchet, 2006, p.140).

¹⁵ Le terme sémantique est utilisé ici dans le domaine informatique, et désigne la formalisation de la règle d'envoi/réception des performatives.

5.2.5.1. Sémantique des descriptifs

Tous les descriptifs sont appliqués à des croyances. Leur forme générale est :

$$pMessage(A_s A_r \text{performatif } pB) \\ \text{avec } A_s, A_r \in AGT, \text{performatif} \in PER \text{ et } pB \in BEL$$

Les deux types de descriptifs utilisés pour les négociations sont *inform* et *reply*. Les deux types de descriptifs ajoutés au modèle sont *leave* et *warning*.

inform

Syntaxe :

$$pMessage(A_s \text{all inform } pB(pD(A_s, pTrans | ! pTrans))) \\ \text{avec } A_s, A_r \in AGT, pTrans : \text{prédictat transaction}$$

Sémantique :

$$\frac{[pD(A_s, pTrans) \mid pD(A_s, ! pTrans)] \xrightarrow{I_{int}(ini) \xrightarrow{\text{send(inform)}} I_{int}(sI)} aUpdateTA(M)}{aAdd(pB(pD(A_s, pTrans | ! pTrans))) ; aUpdateTA(M)}$$

Description :

A_s prévient A_r qu'il a le désir ou non d'effectuer la transaction $pTrans$. À la réception de l'*inform*, A_r ajoute à ses croyances le fait que A_s a le désir d'effectuer la transaction $pTrans$ ou au contraire, qu'il ne l'a pas.

Exemple :

groupe 5 2007-01-18-10h13

[10:52:23] 189, Jack SAY les epices m'interressent

pMessage(Jack all inform pB(pD(pTrans(Jack * * * * SPICES))))

reply

Syntaxe :

$$pMessage(A_s A_r reply pB(pD(A_s, pTrans) | ! pTrans))) \\ \text{avec } A_s, A_r \in AGT, pTrans : \text{prédicat transaction}$$

Sémantique :

$$\frac{\left[pD(A_s, pTrans) | pD(A_s, ! pTrans) \right] \frac{Q_{\text{int}}(s1, s2, s4) \xrightarrow{\text{send(reply)}} Q_{\text{int}}(s2)}{aUpdateTA(M)}}{aAdd(pB(pD(A_s, pTrans) | ! pTrans)); aUpdateTA(M)}$$

$$\frac{Q_{\text{ini}}(s1, s2) \xrightarrow{\text{receive(reply)}} Q_{\text{ini}}(s2)}{aAdd(pB(pD(A_s, pTrans) | ! pTrans)); aUpdateTA(M)}$$

Description :

Suite à un *query* de A_r sur la transaction $pTrans$, A_s peut envoyer un *reply* sur son désir ou non de voir effectuée la transaction $pTrans$, s'il a le désir de communiquer à tous cette information. À la réception d'un *reply*, A_r ajoute à ses croyances la réponse de A_s , c'est-à-dire le désir ou non de A_s de voir effectuée la transaction $pTrans$.

Exemples :

groupe 1 2007-01-16-10h19

[10:38:27] 63, Joe SAY qui peut me proposer une carte miel contre cigare

[10:38:34] 64, Jack SAY moi

$pMessage(\text{Joe all query } pD(pTrans(\text{Jack} * \text{CIGARS} ? * \text{HONEY})))$

$pMessage(\text{Jack Joe reply } pD(pTrans(\text{Jack} * \text{CIGARS} \text{ Joe} * \text{HONEY})))$

leave

Syntaxe :

$$pMessage(A_s A_r leave pB(! pD(A_s, pTrans))) \\ \text{avec } A_s, A_r \in AGT, pTrans : \text{prédicat transaction}$$

Sémantique :

$$\frac{[! pD(A_s, pTrans)] \frac{D_{\text{ini}} / D_{\text{int}} \text{Buyer}(s1, s3, s4) \xrightarrow{\text{send(leave)}} D_{\text{ini}} / D_{\text{int}} \text{Buyer}(s5)}{aUpdateTA(M)}}$$

$$\begin{array}{c}
 \dfrac{D_{\text{ini}} / D_{\text{int}} \text{Seller}(s1, s2, s3, s4) \xrightarrow{\text{send}(\text{leave})} D_{\text{ini}} / D_{\text{int}} \text{Seller}(s5)}{! pD(A_s, pTrans); aUpdateTA(M)} \\
 \\
 \dfrac{D_{\text{ini}} / D_{\text{int}} \text{Seller}(s1, s3, s4) \xrightarrow{\text{receive}(\text{leave})} D_{\text{ini}} / D_{\text{int}} \text{Seller}(s5)}{aAdd(pB(! pD(A_s, pTrans)); aUpdateTA(M))} \\
 \\
 \dfrac{D_{\text{ini}} / D_{\text{int}} \text{Buyer}(s1, s2, s3, s4) \xrightarrow{\text{receive}(\text{leave})} D_{\text{ini}} / D_{\text{int}} \text{Buyer}(s5)}{aAdd(pB(! pD(A_s, pTrans)); aUpdateTA(M))}
 \end{array}$$

Description :

L'agent A_s quitte la discussion de négociation. Il envoie un *leave* aux agents avec qui il était en train de négocier. Quand un agent A_r réceptionne le *leave*, il ajoute à ses croyances la négation du désir de l'agent A_s concernant l'objet de la négociation (ici la transaction).

Exemple :

groupe 8 2007-01-19-14h : Joe vend des cigares :

[15:17:00] 40, Joe SAY je ne fume pas
[15:17:16] 42, Jack SAY j'achète
[15:17:24] 43, William SAY mauvais pour la santé sans moi
[15:18:07] 44, William TAKE_OUT

`pMessage(William all leave pB(!pD(pTrans(Joe * CIGARS William * *))))`

5.2.5.2. Sémantique des directifs

Tous les directifs sont appliqués à des désirs du locuteur. Leur forme générale est donc :

$$\begin{aligned}
 & pMessage(A_s A_r \text{performative } pD(A_s \phi)) \\
 & \text{avec } A_s, A_r \in AGT, \text{ performative} \in PERet \phi \in PRE
 \end{aligned}$$

Le directif *query* du modèle est utilisé. Les quatre types de directifs ajoutés sont *acceptBid*, *refuseBid*, *request* et *relaunch*.

acceptBid

Syntaxe :

$$pMessage(A_s A_r acceptBid pD(A_s, pTrans)) \\ \text{avec } A_s, A_r \in AGT, pTrans : \text{prédictat transaction}$$

Sémantique :

$$\frac{\left[\begin{array}{c} pD(A_s, pTrans) \\ pB(pD(A_r, pTrans)) \end{array} \right] D_{int}/D_{ini} Seller(s1, s3, s4) \xrightarrow{send(acceptBid)} D_{int}/D_{ini} Seller(s6)}{aUpdateTA(M)}$$

$$\frac{D_{int}/D_{ini} Buyer(s1, s3, s4) \xrightarrow{receive(acceptBid)} D_{int}/D_{ini} Buyer(s6)}{aAdd(pB(pD(A_s, pTrans))); aUpdateTA(M)}$$

Description :

A_s accepte la proposition de A_r . A_s peut envoyer un *acceptBid* s'il a le désir d'effectuer la transaction correspondant à $pTrans$, et s'il a la croyance qu' A_r l'a aussi.

À la réception d'un *acceptBid*, A_r sait que A_s désire la transaction également et met à jour ses croyances en conséquence.

Exemple :

groupe 7 2007-01-19-10h18

[11:52:58] 375, Joe SAY 4 pièces

[11:53:15] 376, Jack SAY 4 pieces ok

`pMessage(Jack Joe bid (pTrans(Jack - SPICES Joe 4 -), false))`

`pMessage(Jack Joe acceptBid pTrans(Jack - SPICES Joe 4 -))`

refuseBid

Syntaxe :

$$pMessage(A_s A_r refuseBid pD(A_s, pTrans)) \\ \text{avec } A_s, A_r \in AGT, pTrans : \text{prédictat transaction}$$

Sémantique :

$$\left[\begin{array}{l} pD(A_S, ! pTrans) \\ pB(pD(A_R, pTrans)) \end{array} \right] \frac{D_{ini}/D_{int} Buyer(s2, s3, s4) \xrightarrow{\text{send (refuseBid)}} D_{ini}/D_{int} Buyer(s4)}{aUpdateTA(M)}$$

$$\left[\begin{array}{l} pD(A_S, ! pTrans) \\ pB(pD(A_R, pTrans)) \end{array} \right] \frac{D_{ini}/D_{int} Seller(s1, s3) \xrightarrow{\text{send (refuseBid)}} D_{ini}/D_{int} Seller(s3)}{aUpdateTA(M)}$$

$$\frac{D_{ini}/D_{int} Seller(s2, s3, s4) \xrightarrow{\text{receive (refuseBid)}} D_{ini}/D_{int} Seller(s4)}{aAdd(pB(pD(A_S, ! pTrans))); aUpdateTA(M)}$$

$$\frac{D_{ini}/D_{int} Buyer(s1, s3) \xrightarrow{\text{receive (refuseBid)}} D_{ini}/D_{int} Buyer(s3)}{aAdd(pB(pD(A_S, ! pTrans))); aUpdateTA(M)}$$

Description :

A_s refuse la proposition de A_r . A_s peut envoyer un *refuseBid* après avoir traité le désir proposé par A_r et ne pas l'avoir intégré à sa pile de désirs. Quand A_r reçoit le *refuseBid*, il ajoute à ses croyances la négation du désir qui avait provoqué l'envoi du *bid/request*.

Exemple :

groupe 3 2007-01-17-10h48

- [11:16:01] 298, William SAY 1 pièce
- [11:16:04] 299, Jack SAY non
- [11:16:07] 300, William SAY :P
- [11:16:10] 301, William SAY si si
- [11:16:15] 302, Jack SAY non

pMessage(William Jack bid pD(Jack, pTrans(Jack - TEA William 1 -)), false)

pMessage(Jack William refuseBid pD(Jack, pTrans(Jack - TEA William 1 -)))

request

Syntaxe :

pMessage($A_S A_R$ request $pD(A_S, pTrans)$)
avec $A_S, A_R \in AGT$, $pTrans$: prédictat transaction

Sémantique :

$$\left[\frac{pD(A_S, ! pTrans)}{pB(pD(A_R, pTrans))} \right] \frac{D_{ini}/D_{int} Seller(s1, s2, s3, s4) \xrightarrow{send(request)} D_{ini}/D_{int} Seller(s2, s3)}{aUpdateTA(M)}$$

$$\frac{D_{ini}/D_{int} Buyer(s1, s2, s3, s4) \xrightarrow{receive(request)} D_{ini}/D_{int} Buyer(s2, s3)}{aAdd(pB(pD(A_S, ! pTrans))); aUpdateTA(M)}$$

Description :

Un agent demande à un ou plusieurs agent(s) de faire une action : faire ou modifier une proposition (proposer plus de pièces par exemple).

Exemple :

groupe 3 2007-01-17-10h48

[11:53:23] 683, Jack SAY monte un peu joe

pMessage(Jack Joe request pD(Jack, pTrans(Jack - TEA Joe x -)) up)

groupe 7 2007-01-19-10h18

[10:48:01] 116, William SAY Je ne te propose pas de renchérir, mais de faire un échange : garde tes pièces et propose moi une carte

pMessage(William Jack request pD(Jack, pTrans(William - COFFEE Jack * x)) move)

5.2.5.3. Sémantique des engageants

Tous les engageants sont appliqués à des désirs du destinataire. Leur forme générale est :

$$pMessage(A_S A_R \text{performatif } pD(A_R \phi))$$

avec $A_S, A_R \in AGT$, $\text{performatif} \in PER$ et $\phi \in PRE$

Les engageants observés dans les protocoles expérimentaux sont les *bid*.

bid

Syntaxe :

$$pMessage(A_S A_R bid pD(A_R, pTrans(A_R n typeCard1 A_S m typeCard2)), bool)$$

avec $A_S, A_R \in AGT$, $pTrans : \text{prédicat transaction}$, $bool \in \{ \text{true}, \text{false} \}$

Sémantique :

$$\left[\frac{pD(A_s, pTrans)}{!pB(pD(A_r, !pTrans))} \right] \frac{D_{ini}/D_{int} Buyer(ini, s1, s2, s3, s4) \xrightarrow{send(bid)} D_{ini}/D_{int} Buyer(s1)}{aUpdateTA(M)}$$

$$\frac{D_{ini}/D_{int} Seller(ini, s1, s2, s3, s4) \xrightarrow{receive(bid)} D_{ini}/D_{int} Seller(s1)}{aAdd(pB(pD(A_s, pTrans))); aUpdateTA(M)}$$

Description :

A_s peut envoyer un *bid* portant sur une transaction *pTrans*, si *pTrans* est une transaction qu'il désire et s'il n'a pas de croyance lui indiquant qu' A_r ne la désire pas. À la réception de ce *bid*, A_r ajoute à ses croyances le désir concernant la transaction *pTrans*. A_r ajoutera ce désir à sa pile de désirs seulement si cette transaction lui paraît intéressante (ce qui sera décidé hors du modèle de l'interaction). A_s peut préciser s'il s'agit d'une proposition à prendre ou à laisser (bool prend la valeur true), ou bien si l'offre reste négociable (bool prend la valeur false).

Exemples :

groupe 1 2007-01-16-11h05

[11:12:24] 61, William SAY je propose 5 pieces et carte the

pMessage(Jack William bid pD(William, pTrans(Jack - COFFEE William 5 TEA)), false)

groupe 6 2007-01-18-14h18

[14:32:46] 122, Jack PROPOSE (Averell, COINS -, CARD vin, Jack, COINS -, CARD épices)

[14:32:56] 124, Jack SAY épices ou rien

pMessage(Jack Averell bid pD(Jack,pTrans(Averell - WINE Jack 5 SPICES)) true)

épices ou rien est une offre à prendre ou à laisser.

5.2.5.4. Adaptation de la sémantique

Dans le problème de l'agence de voyage, le prédictat *pMeans(S)* est vrai si *S* est une étape dont le moyen de transport est celui dont l'agent a la charge (voir partie 1.2 Le système BDIGGY). Dans notre application, *pMeans* est utilisé dans une pré-condition de la performative *query*, avec un sens différent.

pB(!pMeans(S)), lors d'une négociation, signifie que l'agent sait qu'il ne peut pas avoir l'information (qu'il demande en envoyant un *query*). Nous l'avons donc renommé en

pCapabilities. Cela signifie que l'agent n'a pas la capacité d'obtenir par lui-même l'information (accès aux informations concernant un moyen de transport dans le problème de l'agence de voyage, ou bien connaissance des désirs d'un autre agent pour l'application de négociation). Ce prédicat est également utilisé dans les performatives suivantes (non observées durant les négociations) :

- *acceptProposal*,
- *refine*,
- *propose*.

Les prédictats *pSent* et *pNotUnderstood* du modèle sont génériques, et réutilisés *tels quels*.

5.2.6. La notion de satisfaction

« Négocier deviendrait impossible si chacun étaisait son jeu sur la table. Les prétentions apparaîtraient si crûment incompatibles, que ce serait à désespérer de toute solution. Il faut que vous masquiez vos intentions, que vous ne les découvriez que petit à petit, à tâtons, pour sentir par où elles s'accordent. [...] Surtout ne vous engagez pas, ne vous liez pas prématurément ! Rien n'est fait, rien n'est dit qu'au tout dernier instant. »

Saint-Germain ou la négociation. Francis Walder.

(Pauchet, 2006, p.110), précise, pour chaque état terminal des automates, la satisfaction de l'échange par une étiquette *Satisfaction* ou *Unsatisfaction*. Par exemple, dans l'automate de demande d'information Q_{ini} (FIG. 40), l'échange est satisfait si après une demande (*query*), on reçoit une réponse (*reply*) : l'état s_2 porte une étiquette *Satisfaction*. On considère par contre qu'il n'est pas satisfait si par exemple, l'envoi d'un *cancel* a suivi l'envoi du *query* : l'état s_6 porte une étiquette *Unsatisfaction*.

Pour la négociation, la notion de satisfaction est difficilement cernable. Comme l'explique (Simos, 1990), même en cas d'accord, les deux protagonistes ne sont pas forcément satisfaits (voir partie 2.2.2. Les deux types de négociation). Le résultat de la négociation peut être un résultat perdant-gagnant. Pour cette raison, nous n'avons pas utilisé ces étiquettes. La stratégie de négociation d'un agent peut être de demander plus pour obtenir ce qu'il souhaite. La concordance *demande de départ/résultat*, bien souvent, n'est pas révélatrice. Il paraît donc difficile, en observant les échanges durant la négociation, de déterminer si un agent est satisfait, très satisfait, peu satisfait (il semble s'agir plutôt d'un degré de satisfaction que d'un indicateur binaire), au terme des échanges. Nous parlerons de satisfaction d'un agent en fonction de l'adéquation de l'accord trouvé avec ses *préférences*, et, comme pour l'humain, la satisfaction sera une propriété intrinsèque à l'agent.

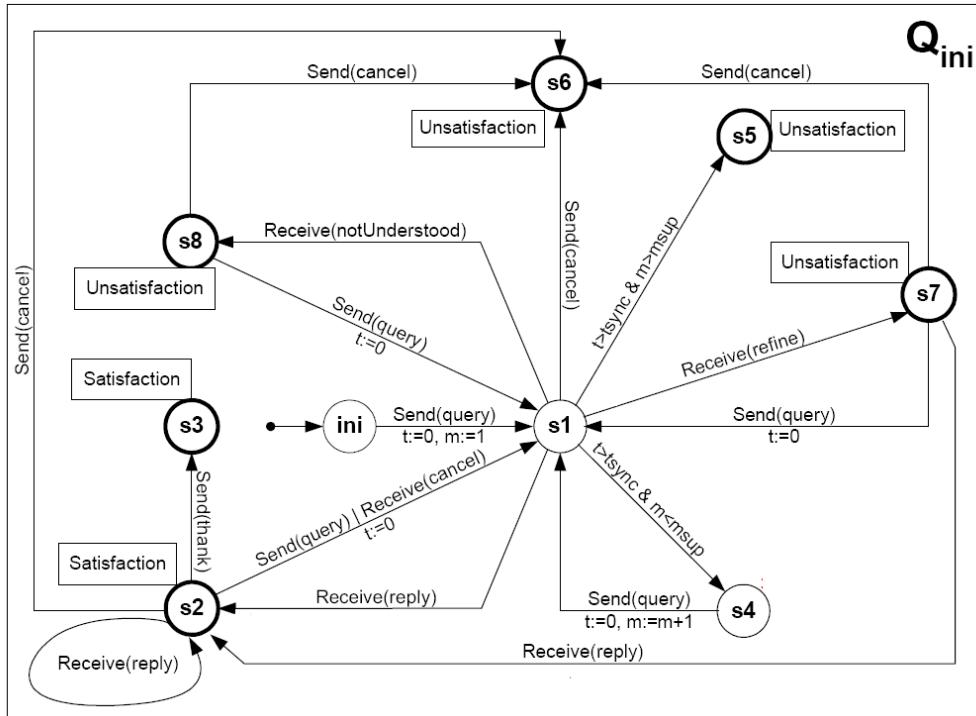


FIG. 40 - Automate de demande d'information Q_{ini}

5.2.7. La mémoire

Dans le problème de l'agence de voyage, lorsqu'un agent obtient un horaire pour le départ d'un train (par exemple), il stocke cet horaire dans sa mémoire et ne le *remettra pas en question*. C'est le cas lorsque l'on obtient une information d'un agent au guichet d'une gare. L'agent stocke les informations obtenues par les autres agents (les informations à propos des deux moyens de transports dont il n'a pas la charge, et les informations qu'il obtient de la base de données de son propre moyen de transport), dans une zone de mémoire à *long terme* appelée *Croyances persistantes* (voir FIG. 41). Dans une situation de négociation, les désirs des agents (motivés par leurs préférences) évoluent sans cesse. Les croyances sur les désirs des autres agents sont donc considérées par un agent comme des croyances susceptibles d'être modifiées ou supprimées. Nous avons ajouté à la mémoire à *court terme*, une zone appelée *Croyances ajustables*, qui contiendra ce type de croyances (voir FIG. 42).

	Représentation de l'interface	
	Panneau "Horaires/Prix" Panneau "Zone de travail" Panneau "Proposition" Panneau "Messagerie"	
	Croyances non traitées	
	Croyances sur des désirs à évaluer Croyances sur l'état des échanges en cours	
	Croyances persistantes	
	Informations obtenues par la base de données Informations obtenues auprès des autres agents	
	Représentation du problème	
	Problème à résoudre Représentation de la carte (villes, distances et transports) Connaissances implicites	

FIG. 41 - Structure d'origine de la mémoire dans BDIGGY

	Représentation de l'interface	
	Panneaux Décisions/Nombres de Pièces des participants Contenu de la fenêtre de clavardage	
	Croyances non traitées	
	Croyances sur des désirs à évaluer Croyances sur l'état des échanges en cours	
	Croyances ajustables	
	Croyances sur les désirs des autres agents	
	Croyances persistantes	
	Informations obtenues par la base de données Informations obtenues auprès des autres agents	
	Représentation du problème	
	Règles du jeu	

FIG. 42 - Mémoire modifiée dans BDIGGY

5.2.8. L'architecture

Le schéma de la FIG. 43 présente les modifications effectuées au niveau de l'architecture d'un agent BDIGGY. Les performatives et automates décrits précédemment sont contenus dans le module de communications. La mémoire a été modifiée (voir section précédente). Le module IGGY 2 (voir Voir FIG. 2 - L'architecture BDIGGY, partie 1.2. Le système BDIGGY), qui construit des plans, est remplacé par un module de raisonnement qui se basera sur des heuristiques, et

fournira les préférences de l'agent (la structure de ce module n'a pas été traitée dans le cadre de ce stage).

Une heuristique est une règle ou un ensemble de règles pour prendre une décision. Une méthode heuristique ne garantit pas l'obtention d'une solution optimale, mais fournit une solution, dans un laps de temps raisonnable et à un coût acceptable, en fonction de l'expérience ou de l'intuition. On utilise fréquemment des méthodes heuristiques en intelligence artificielle, à l'intérieur de ce qui est techniquement appelé un univers incertain. L'heuristique utilisée ici sera dépendante du domaine d'application. Comme nous l'avons expliqué, nous ne traitons pas la partie raisonnement des agents. Mais à titre d'exemple, pour notre application de jeu de négociation, les règles pourraient être de la forme :

- pour un acheteur :

si je sais que la carte x intéresse le vendeur, et si j'ai cette carte, alors je la lui propose,

si je ne connais pas ce qui intéresse le vendeur, alors je lui demande ce qui l'intéresse,

...

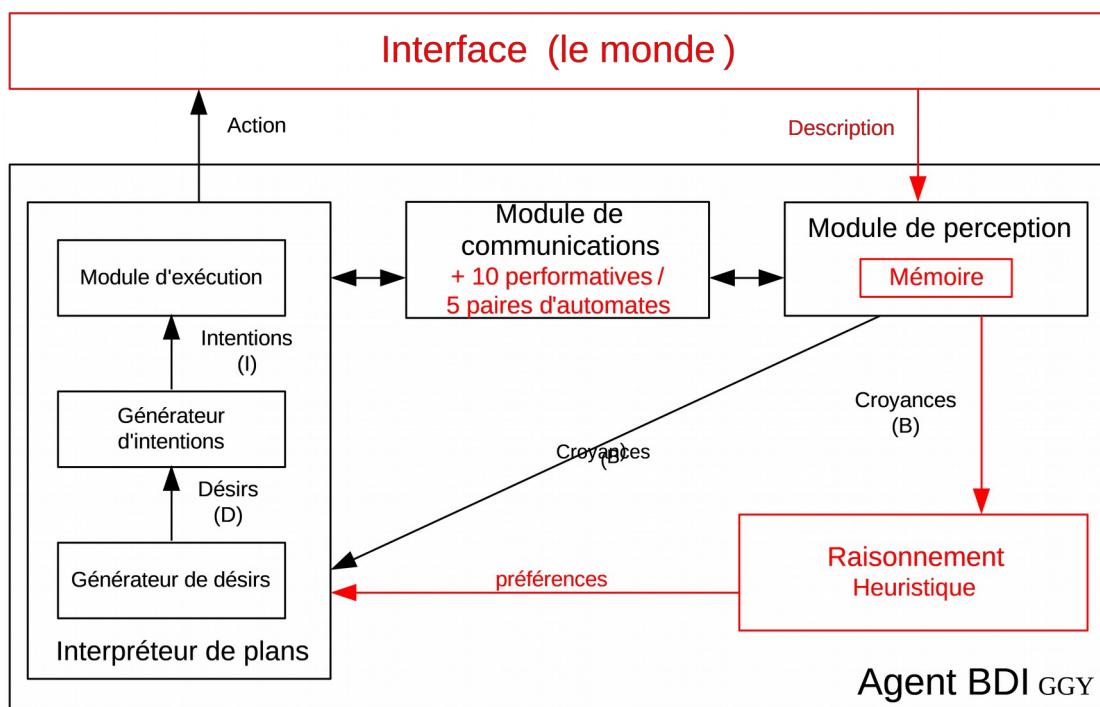


FIG. 43 - Modifications de l'architecture d'un agent

- pour un vendeur :

s'il ne reste plus qu'un acheteur présent dans la vente, et que celui-ci m'a fait une offre, alors j'accepte sa proposition,

...

L'heuristique utilisée donnera les *préférences* de l'agent. Il peut s'agir d'un résultat binaire, comme la décision de répondre ou non à une question (il n'est parfois pas judicieux, durant une négociation, de répondre à toutes les questions). Il peut s'agir également des préférences de l'agent sur l'objet négocié. Pour développer ce module, nous pourrons par exemple utiliser les notions de *critères*, du système IGGY (Chaignaud, 1996). Les préférences sont ensuite traduites en désirs puis en intentions par l'interpréteur de plans.

5.2.9. Parallèle avec l'approche théorie des jeux

Pour modéliser la partie raisonnement des agents, nous pourrions nous inspirer des outils proposés par la théorie des jeux. Par exemple, trouver un équilibre de gains, peut faciliter la recherche d'un compromis entre les protagonistes d'une négociation. Par ailleurs, le jeu du duel et le jeu des archers, présentés dans (Umbauer, 2004, p.51) se rapprochent également de la situation dans laquelle se trouvent les acheteurs de notre micro-monde.

Le duel est l'un des plus anciens jeux étudiés en théorie des jeux.

Deux cow-boys tiennent chacun un pistolet contenant une seule balle. Ils sont initialement distants de n mètres et avancent progressivement l'un vers l'autre, avec l'intention réciproque de s'abattre.

Plus les joueurs attendent pour tirer, plus ils ont de chances d'atteindre leur cible, car la distance qui les séparent se réduit, mais plus ils courrent le risque de voir l'autre dégainer avant eux, et mourir avant d'avoir eu l'occasion de tirer.

Le jeu des archers est une variante du jeu du duel.

Deux tireurs à l'arc, ne disposant que d'une seule flèche sont à une distance de 6 mètres de la cible. À chaque étape impaire, le joueur 1 choisit de tirer ou non son unique flèche, s'il n'a pas procédé au tir à une étape précédente. Il en va de même pour le joueur 2 à chaque étape paire. Chaque tireur, à chaque prise de décision, observe le comportement et le résultat des actions de son adversaire aux étapes antérieures. Si l'un des deux joueurs tire et atteint la cible, le jeu s'arrête. S'il tire mais n'atteint pas la cible, il sort du jeu. Dans ce cas, tout comme dans le cas où il choisit de ne pas tirer sa flèche, on passe à l'étape suivante. Enfin, entre deux étapes successives, les joueurs restants avancent d'un mètre vers la cible. La probabilité d'atteindre la cible augmente au fur et à mesure qu'elle se rapproche et vaut 1 à 0 mètre de la cible. Le jeu s'arrête dans les trois cas suivants : un joueur atteint la cible ; il n'y a plus de joueurs ; après l'étape 7, qui correspond au moment où les deux joueurs sont devant la cible (distance 0). Un joueur est dit vainqueur (montant du gain = 1) s'il atteint la cible. Dans tous les autres cas, il est perdant (montant du gain = 0).

Un parallèle peut être fait avec les acheteurs dans notre jeu de négociation. Plus ils attendent, plus la probabilité de proposer une offre qui intéresse le vendeur augmente. Ils peuvent en effet ajuster leur offre en fonction des informations que donne le vendeur sur ses désirs et en fonction des propositions des autres acheteurs. Cela dit, les joueurs n'ont pas d'ordre d'intervention à respecter, et s'ils n'atteignent pas la cible (satisfaire le vendeur) ils ne sont pas éliminés : c'est tout le jeu de la négociation.

La théorie des jeux sert donc surtout à modéliser des situations où des acteurs prennent des décisions individuelles, ayant un impact sur tous les acteurs. Elle offre un cadre théorique pour rechercher des équilibres, étudier la prise de risque, mettre en évidence des stratégies optimales. Elle est utile pour modéliser le raisonnement d'un agent, ce que nous n'avons pas traité dans le cadre de ce stage car le raisonnement est propre à l'application (Vercouter *et al.*, 1999) (Verrons, 2004).

Critique de notre travail

Lorsqu'une expérimentation est mise au point, il y a toujours des biais. Pour celle que nous avons réalisée, nous pouvons par exemple citer :

- **la situation choisie** pour observer la négociation. Une situation de négociation comprend des paramètres variés, comme le nombre de participants (un processus de négociation peut engager une discussion entre deux participants, d'un participant vers n participants, de n participants vers m participants), le type d'accord qui doit être trouvé, la quantité et la qualité des informations dont dispose chacun. Tous ces aspects ont des effets sur la négociation. Nous avons dû faire un choix de configuration (quatre participants, un vendeur, trois acheteurs). Les rôles des participants *tournent* néanmoins régulièrement,
- **la connaissance des sujets du fait qu'ils participaient à une expérimentation**. Les sujets savaient qu'ils participaient à une expérimentation psychologique. Nous avons tenté de minimiser au maximum les biais liés à ce qui est appelé le *paradoxe de l'enquêteur*¹⁶, en isolant chaque sujet dans une pièce. Il a fallu néanmoins surveiller le déroulement de l'expérimentation (en passant régulièrement dans les pièces en début de partie) pour éviter tout éventuel problème lié à la compréhension des règles par les sujets, ou à leur utilisation du logiciel.
- **l'interface graphique de l'application** : les actions rendues possibles dans l'interface sont au choix du concepteur. Nous avons cependant essayé de *coller* au maximum aux règles du jeu. Les boutons d'actions étaient nécessaires : il n'était pas possible de valider la fin d'une vente seulement en lisant la conversation des participants dans la fenêtre de *chat*. L'initiative de valider les *avancées* dans le jeu, revenait bien sûr aux participants. Les actions permises par les boutons de l'interface avaient pour unique objectif de permettre cette validation,
- **l'enjeu** : ici les sujets humains ont participé à un jeu. L'enjeu était de gagner (réunir un certain nombre de cartes avant les autres) et était loin d'être *vital*. Les négociations qui

¹⁶ Le *paradoxe de l'observateur* (popularisé par W. Labov en sociolinguistique) est *Comment observer la manière dont les gens agissent quand ils ne sont pas observés*? Leurs productions peuvent en effet être modifiées rien que par notre présence en tant qu'enquêteur.

impliquent l'avenir d'une personne ou d'une entreprise, présentent sans aucun doute des aspects différents,

- **L'échantillon des participants**, l'âge des participants, leur milieu social, par exemple, ont également un effet sur les négociations.

Nous pourrions citer d'autres biais. À noter que nous avons souvent constaté que les sujets étaient *plongés* dans la situation de commerce électronique. Assez pour penser que le micro-monde que nous avons mis en place était suffisamment réaliste pour être exploitable.

Conclusion et perspectives

L'objectif de ce stage était de permettre aux agents du système BDIGGY de négocier, en fondant la modélisation sur l'observation des interactions humaines. Après avoir précisé le concept de négociation, nous avons défini un protocole d'expérimentation utilisant un jeu de négociation et permettant d'observer le processus de négociation entre sujets humains. Les traces obtenues grâce au logiciel outil développé, ont permis d'analyser les types d'échanges entre les protagonistes et de dégager un protocole pour les agents du système. BDIGGY a été enrichi par de nouveaux éléments : des performatives couplées à des automates de modélisation des interactions. Nous avons effectué des adaptations de l'architecture existante pour prendre en compte ces nouvelles interactions. Le contenu du module *Raisonnement*, propre à l'application, n'a pas été traité. Il est l'objet du projet de fin d'études d'ingénieur d'Emmanuel Arnaud (PFE à l'INSA de Rouen qui débutera en septembre 2007). Les résultats permettront de mettre en œuvre et de tester le protocole de négociation proposé (simulation des processus observés).

Nous avons essayé de rendre le modèle le plus générique possible. Il serait intéressant de le tester sur une autre application, comme par exemple la prise de rendez-vous.

Sans prévoir de questionnaire particulier à donner aux participants des expérimentations, certains d'entre eux ont spontanément exprimé des impressions et remarques en fin de partie :

- un participant nous a expliqué avoir fait une erreur par rapport aux règles du jeu : il a confondu le vainqueur désigné en cas d'égalité après le dernier échange (si chacun des deux possède 8 cartes après la dernière vente). Pensant que c'était le vendeur et non l'acheteur, il a vendu sa carte à un acheteur qui a remporté la partie,
- certains ont menti durant les négociations,
- un participant a expliqué qu'il n'utilisait pas du tout le *chat* durant la partie : il considérait le dialogue comme inutile,
- certains participants ont expliqué que le déroulement était parfois rapide et difficile à suivre (conversations dans le *chat*, actions des autres joueurs, etc.),
- certains ont expliqué qu'ils avaient parfois refusé de traiter avec un participant en particulier, à cause du participant et non de ses offres (en représailles à de précédentes négociations).

Toutes ces remarques ouvrent des portes à de nouvelles expérimentations qui permettraient de mettre le *focus* sur des points précis, notamment :

- les types d'échanges que nous n'avons pas pu prendre en compte ici, comme les promesses. La situation choisie ici ne permettait pas forcément de faire beaucoup de promesses. Une situation permettant aux participants d'envisager un avenir commun

et donnant les moyens à chacun de pouvoir promettre à l'autre, nous permettrait d'observer davantage ce type d'interventions,

- la possibilité de dialogue privé entre les participants : nous pourrions par exemple étudier les différences entre ce qui est échangé en privé et ce qui est tapé dans le *chat*, observer si des informations sur les autres agents sont échangés entre deux agents, etc.. La confiance et la réputation des agents au sein d'un système sont étudiées dans le domaine SMA, voir par exemple (Muller *et al.*, 2005),
- la variation des paramètres : nous pourrions faire varier le nombre de participants, le nombre d'acheteurs et de vendeurs, le degré d'information réciproque. Les interventions durant la négociation dépendent des forces et faiblesses des protagonistes, des anticipations que chacun fait sur le comportement de ses interlocuteurs. Il serait intéressant d'étudier le rapport de force, en fonction des paramètres,
- la personnalité des participants : cela nous permettrait de vérifier si les caractères de la personnalité des agents de BDIGGY sont suffisamment précis pour ces nouvelles situations,
- la qualité de la relation entre les participants : effectuer davantage de parties avec un même groupe, étudier les représailles et les conséquences des accords trouvés sur la qualité des relations entre les protagonistes,
- les alliances : effectuer davantage de parties au sein d'un même groupe permettrait également d'observer la formation des alliances entre acheteurs (nous n'en avons que trop rarement observé, pour pouvoir en tenir compte dans la modélisation). Dans le domaine SMA, cet aspect a notamment été traité par (Vauvert, 2004).

L'étude que nous avons réalisée, comme celles qui ont permis la conception du système BDIGGY, met en évidence l'apport de la modélisation cognitive pour concevoir des protocoles d'interactions entre agents au sein de systèmes hétérogènes.

Bibliographie

[15:27:36] 79, Jack SAY oui, mais pour l'instant, j'ai le choix entre des livres ou du miel
[15:27:46] 80, Jack SAY il faut m'allécher un peu plus que ça
[15:27:50] 81, Averell SAY oui mais les livres ça se mange pas
[15:28:01] 82, Jack SAY non, mais ça peut se dévorer

Groupe d'expé. n°8

Austin J.L., (1962), *How to Do Things with Words*. Oxford University Press, Oxford.

Axelrod R., (1981), The Emergence of Cooperation Among Egoists, In *The American Political Science Review*, Vol. 75, p. 306-318, 1981.

Axelrod R., (1997), *The Complexity of Cooperation. Agent Based Models of Competition and Collaboration*, Princeton University Press, 1997.

Baker M. J., (1994), Argumentation, Explication et Négociation : analyse d'un corpus de dialogues en langue naturelle écrite dans le domaine de la médecine, In *Modélisation d'explications sur un corpus de dialogues : Actes de l'atelier de recherche*, p. 1-26. Rapport de Télécom Paris, No. 94-S-003.

Ballmer T. et Brennenstuhl W., (1981), *Speech Act Classification: A Study in the Lexical Analysis of English Speech Activity Verbs*. Springer-Verlag, Berlin.

Bartolini C., Preist C., et Jennings N.R., (2002), Architecting for reuse: A software framework for automated negotiation, In *Proceedings of 3rd International Workshop on Agent-Oriented Software Engineering*, Bologna, Italy.

Beaufils B., (2000), *Modèles et simulations informatiques des problèmes de coopération entre agents*. Thèse de doctorat, Université des Sciences et Technologies de Lille, 25 janvier 2000.

Beer M., d'Inverno M., Luck M., Jennings N.R., Preist C., Schroeder M., (1998), Negotiation in Multi-Agent Systems In *Knowledge Engineering Review*, 14(3), p. 285-289, 1999. Panel de discussions du *Workshop of the UK Special Interest Group on Multi-Agent Systems (UKMAS'98)*.

Bellosta M-J., Brigui I., Kornman S., Pinson S., Vanderpooten D., (2003), Un mécanisme de négociation multicritère pour le commerce électronique, In *Proceedings Reconnaissance des Formes et Intelligence Artificielle (RFIA'2003)*, Toulouse, janvier 2004. Paru dans les Annales du LAMSADE, No. 2, juin 2004, ISSN-1762-455X, p. 19-34.

Bentahar J., Moulin B., Chaib-draa B., (2003), Vers une approche pour la modélisation du dialogue basée sur les engagements et les arguments, In *Actes des Secondes Journées Francophones Modèles Formels de l'Interaction*, Lille, France, 20-22 mai 2003.

Bichler M., Kalagnanam J., Lee H.S., (2003), RECO : Representation and evaluation of configurable offers In H.K. Bhargava, N. Ye (eds), *Computational Modeling and Problem Solving in the Networked World : Interfaces in Computing and Optimization*. Kluwer Academic Publishers.

- Bousquet F., Bakam I., Proton H., Le Page C., (1998), Cormas : common-pool resources and multi-agent Systems. Lecture Note In *Computer system*, Springer. 1416: 826-838.
- Bousquet F., Barreteau O., d'Aquino P., Etienne M., Boisseau S., Aubert S., Le Page C., Babin D., Castella J.C., (2002), Multi-agent systems and role games : collective learning processes for ecosystem management, In *Complexity and Ecosystem Management. The theory and practice of multi-agent systems*. M.A. Janssen (ed.), Edward Elgar Publishing Limited, p. 248-285.
- Brassac C. et Pesty S., (1996), La Pelouse Fourmilière. De la coaction à la coopération, In *IA Distribuée et Systèmes Multi-Agents, 4èmes Journées Francophones sur l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents (JFIADSMA'96)*, Le Grau du Roi, AFCET, AFIA, Müller, Quinqueton, (eds), Hermes, p. 251-263.
- Bratman M.E., (1987), *Intention, Plans, and Practical Reason*. Harvard University Press : Cambridge, MA.
- Briot J-P. et Demazeau Y., (2001), *Principes et architecture des systèmes multi-agents*. Collection IC2, Hermes Science Publications, Briot et Demazeau (eds.), Paris.
- Bussmann S. et Müller J., (1992), A Negotiation Framework for Cooperating Agents, dans S.M.Deen (ed.), *Proceedings of the CKBS-SIG (CKBS'92)*, DAKE Centre, University of Keele, p. 1-17.
- Caverni J-P., (1991), La verbalisation comme source d'observables pour l'étude du fonctionnement cognitif In *Psychologie Cognitive : Modèles et Méthodes*, Caverni, Bastien, Mendelsohn et Tiberghien, PUG, p. 253-273.
- Chaib-draa B., I. Jarras I., Moulin B., (2001), Systèmes multiagents : Principes généraux et applications, In *Principes et Architecture des Systèmes Multi-Agents*, J.P. Briot, Y. Demazeau, Traité IC2, Hermès, Paris, France, novembre 2001.
- Chaignaud N., (1996), *Etude cognitive et informatique de la résolution d'un problème : analyse, modélisation et implantation*. Thèse de doctorat d'Informatique, Université Paris-Nord, 25 janvier 1996.
- Chaignaud N. et Levy F., (1996), Common sense reasoning : experiments and implementation, *European Conference on Artificial Intelligence (ECAI'96)*, Budapest (Hongrie), p. 604-608.
- Chang M.K et Woo C.C, (1992), SANP: A Communication Level Protocol for Negotiations, In *Decentralized A. I. 3*, E. Werner et Y. Demazeau (eds), p. 89-198.
- Chang M.K et Woo C.C, (1994), A Speech-Act-Based Negotiation Protocol : Design, implementation, and Test Use, In *ACM Transactions on Information Systems*, Vol. 12, No. 4, p. 360-382, octobre 1994.
- Chapanis A., (1979), Interactive communication: a few research answers for technological explosion, *Nouvelles tendances de la communication homme-machine*, Cours de la commission des communautés européennes, p. 33-67.
- Chavalarias D., (2005), *Modeling Endogenous Social Networks: The Example of Emergence and Stability of Cooperation without Refusal*, 36 p.
- Chavez A., Maes P., (1996), Kasbah: An Agent MarketPlace for Buying and Selling Goods, In *Proceedings of the first international Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96)*, Londres, p. 75-90.

- Conry S., Meyer R.A., Lesser V., (1988), Multistage negotiation in distributed planning, In *Readings in distributed artificial intelligence*, A. Bond et L. Gesser (eds), Morgan Kaufman, San Mateo, Californie, p. 367-384.
- Demazeau Y., (1995), From interactions to collective behaviour in agent-based systems, In *Proceedings of the 1995 European Conference on Cognitive Sciences*, Saint-Malo, France.
- Etienne M., (2003), SYLVOPAST : a multiple target role-playing game to assess negotiation processes in sylvopastoral management planning, In *Journal of Artificial Societies and Social Simulation*, Vol. 6, No. 2, mars 2003.
- Etivie S., (2006), *Allocation de Ressources Multi-Agent : Théorie et Pratique*. Thèse de doctorat de l'Université Paris-Dauphine, 12 décembre 2006.
- Ferber J., (1995), *Les systèmes multi-agents : Vers une intelligence collective*, InterEditions.
- Ferber J., (1996), Coopération, intention et interaction. Une réponse à la « pelouse fourmilière » de Brassac et Pesty, In *IA Distribuée et Systèmes Multi-Agents, 4èmes Journées Francophones sur l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents (JFIADSMA'96)*, Le Grau du Roi, AFCET, AFIA, Müller, Quinqueton, (eds), Hermes, p. 273-280.
- Franchesquin N., (2001), *Modélisation et simulation Multi-Agents d'écosystèmes anthropisés : une application à la gestion hydraulique en Grande Camargue*. Thèse d'informatique de l'Université de Droit, d'Économie et des Sciences d'Aix-Marseille, Aix-Marseille III.
- Guessoum Z., (1996), *Un environnement opérationnel de conception et de réalisation de systèmes multi-agents*. Thèse de doctorat de l'Université Pierre et Marie Curie (Paris 6), LAFORIA, Paris 6, mai 1996.
- Guessoum Z., (2003), *Modèles et architectures d'agents et de systèmes multi-agents adaptatifs*. Dossier d'habilitation de l'Université Pierre et Marie Curie (Paris 6), décembre 2003.
- Guicherd C., (2006), *Le rôle du pouvoir et de la personnalité des acteurs dans la négociation partenariale industrielle*. Thèse de doctorat, Université Pierre Mendès France Grenoble 2, 18 septembre 2006.
- Hoc J-M., (1996), *Supervision et contrôle de processus. La cognition en situation dynamique*, Presses Universitaires de Grenoble.
- Jennings N.R., Parsons S., Sierra C., et Faratin P., (2000), Automated Negotiation, In *Proceedings of 5th International Conference on the Practical Application of Intelligent Agents and Multi-Agent Systems (PAAM-2000)*, Manchester, UK, p. 23-30.
- Koning J.L. et Pesty S., (2001), Modèles de communication, In *Principes et architectures des systèmes multi-agents*, édité par Jean-Pierre Briot et Yves Demazeau, Collection IC2, Hermes Science Publications, Paris.
- Koo C.C., (1988), A commitment-based communication model for distributed office environments, In *Proceeding of Conference on Office Information System ACM Press*, New York, p. 291-298.
- Kraus S., Wilkenfeld J., Zlotkin G., Multiagent Negotiation Under Time Constraints, In *Artificial Intelligence Journal*, Vol. 75, No. 2 p. 297-345.
- Kraus S., Sycara K., Evenchik A., (1998), Reaching agreements through argumentation : a logical model and implementation, In *Artificial Intelligence*, Elsevier Science, 104, p.1-69.

Kraus S., (2001), Automated Negotiation and decision Making in MultiAgent Environments, In *Multi-Agent systems and Applications 9 th ECCAI Advanced Course, ACAI 2001 et Agent Link's 3rd European Agent Systems Summer School, EASSS 2001*, Lecture Notes In Artificial Intelligence 2086, p. 150-172.

Le Bars M., (2003), *Un Simulateur Multi-Agent pour l'Aide à la Décision d'un Collectif : Application à la Gestion d'une ressource Limitée Agro-environnementale*. Thèse de doctorat d'Informatique, Université Paris IX-Dauphine, 218p., 27 mai 2003.

Lenay C., (1996), Coopération et intentionnalité, In *IA Distribuée et Systèmes Multi-Agents, 4èmes Journées Francophones sur l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents (JFIADSMA'96)*, Le Grau du Roi, AFCET, AFIA, Müller, Quinqueton, (eds), Hermes, p. 265-271.

Lizotte S. et Chaib-Draa B., (1995), Coordination en situation non familières, In *Actes des Troisièmes Journées Francophones sur l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents (JFIADSMA'95)*, St Badolph, Savoie, France, p. 255-266, 15-17 mars 1995.

Malone T.W. et Crowston K., (1994), The interdisciplinary study of coordination, In *ACM Computing Surveys*, Vol. 26, No. 1, mars 1994.

Malone T.W., (1998), What is coordination theory, In *National Science Foundation Coordination Theory Workshop*, MIT.

Maudet N., (2001), *Modéliser les conventions des interactions langagières : la contribution des jeux de dialogue*. Thèse de l'Université Paul Sabatier, Toulouse, 18 mai 2001.

Muller G., Vercouter L., Boissier O., (2005), A trust Model for the reliability of Agent communications, In *8th Workshop on Trust in Agent Societies*, C. Castelfranchi, S. Barber, J. Sabater, M. Singh (eds.), p.106-118, Utrecht, Netherlands.

Müller H.J., (1996), Negotiation principles, In *Foundations of Distributed Artificial Intelligence*, G.M. P. O'Harre et N. R. Jennings (eds), John Wiley & Sons, p. 211-229.

Newell A. et Simon H.A., (1972), *Human Problem Solving*, Prentice-hall, Inc, Englewood Cliffs, New Jersey.

Newell A., (1982), The Knowledge Level, Presidential Address, American Association for Artificial Intelligence, AAAI80, Stanford University, 19 Aug 1980, publié dans *Artificial Intelligence Magazine* 2(2) 1981 et 18(1) : p. 87-127, 1982.

Pauchet A., Chaignaud N., El Fallah Seghrouchni A., (2005), Un modèle cognitif de l'interaction pour agents logiciels, In *Actes des Journées Francophones Modèles Formels de l'Interaction (MFI)*, Caen, p. 183-192.

Pauchet A., (2006), *Modélisation cognitive d'interactions humaines dans un cadre de planification multi-agents*. Thèse de doctorat, Université de Paris Nord, septembre 2006.

Rahwan I., Sonenberg L., Jennings N.R., McBurney P., (2007), STRATUM : A Methodology for Designing Heuristic Agent Negotiation Strategies, In *Applied Artificial Intelligence*, Vol. 21, No. 10, 41 p.

Rao A.S. et Georgeff M., (1995), BDI Agents : from theory to practice. In *Proceedings of 1st International Conference on Multi-Agent Systems (ICMAS'95)*, p. 312-319, San Francisco, CA.

- Rosenschein J-S. et Zlotkin G., (1994), *Rules of Encounter : Designing Conventions for Automated Negotiation Among Computers*, MIT Press, Cambridge, Massachusetts.
- Ruffez P., (2003), *Etude de la négociation dans le cadre d'un jeu de rôle sur le Sylvopastoralisme*, Rapport de stage de DEA IARFA, Encadrants : Le Page C., Drogoul A., 17 septembre 2003
- Sandholm T., (1993), An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations, In *Eleventh National Conference on Artificial Intelligence, AAAI-93*, Washington DC. (Acceptance Rate 24%), p. 256-262.
- Sandholm T., Lesser V., (1995), Issues in Automated Negotiation and Electronic Commerce : Extending the Contract Net Framework, In *First International Conference on Multiagent Systems (ICMAS-95)*, San Francisco. (Acceptance rate 33%), p. 328-335.
- Schroeder M., (1999), An Efficient Argumentation Framework for Negotiating Autonomous Agents, In *Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World : Multi-Agent System Engineering (MAAMAW-99)*.
- Schwartz R. et Kraus S., (1998), Bidding mechanisms for data allocation in multiagent environments, In Munindar P. Singh, Anand S. Rao, and Michael J. Wooldridge, editors, *Intelligent Agents IV: Agent Theories, Architectures, and Languages*, p. 61-75. Springer-Verlag.
- Searle J.R., (1969), *Speech Acts - An Essay in the Philosophy of Language*. Cambridge University Press, Cambrigde.
- Sian S.S., (1991), Adaptation based on cooperative learning in multi-agent systems, In *Decentralized A. I. 2*, Elsevier Science, p. 257-272.
- Simos J., (1990), *Evaluer l'impact sur l'environnement: une approche originale par l'analyse multicritère et la négociation*, Presse Polytechniques et Universitaires Romandes, Lausanne, Switzerland, 261 p.
- Smith R.G., (1980), The Contract Net Protocol : High-Level Communication and Control in a Distributed Problem Solver. *IEEE Transactions on Computers*, C-29(12), p. 1104-1113, December 1980. Reprinted in *Readings in Distributed Artificial Intelligence*, A.H. Bond et L. Gasser, (eds.), Morgan Kaufmann Publishers, Inc., 1988.
- Sycara K., (1989), Multi-agent compromise via negotiation. In *Distributed Artificial Intelligence*, Gasser L. et Huhns M., (eds.) Vol. 2, p. 119-138, CA, septembre 1989.
- TAC <http://www.sics.se/tac/>
- Umbhauer G., (2004), *Théorie des jeux*, ed. Vuibert, Collection Dyna'Sup, septembre 2004.
- Vanderveken D., (1999), Analyse et simulation de conversations. *De la théorie des discours aux systèmes multiagents*. Chapitre 2 : La structure logique des dialogues intelligents, p. 61-100. L'interdisciplinaire, Limonest.
- Vanderveken D., (1988), *Les actes de discours. Essai de philosophie du langage et de l'esprit sur la signification des énonciations*. Pierre Mardaga.
- Vauvert G., (2004), *Atteinte de consensus basée sur l'échange de préférences entre agents rationnels et autonomes : application à la formation d'alliances*. Thèse de doctorat d'Informatique de l'Université de Paris 13.

Vercouter L., Franchesquin N., Treuil J-P., Pinson S., (1999), Simulation multi-agent des processus de négociation en agronomie : un exemple de gestion locale sous contrainte collective, In *Cahier du LAMSADE* No. 115, juillet 1999.

Verrons M-H., (2004), GENCA : un modèle général de négociation de contrats entre agents. Thèse de doctorat, Université des Sciences et Technologies de Lille, 2 novembre 2004.

Von Neumann J. et Morgenstern O., (1944), *Theory of games and economic behavior*, Princeton University Press, 625 p.

Walder F., (1958), *Saint-Germain ou la négociation*, ed. Gallimard de septembre 2004, 239 p., prix Goncourt 1958.

Walsh W., Wurman P., Wellman M., (1998), The michigan internet auctionbot : A configurable auction server for human and software agents In *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*, Minneapolis, MN, USA, mai 1998.

Walton D.N. et Krabbe E.C.W., (1995), *Commitment in Dialogue : Basic Concepts of Interpersonal Reasoning*, State University of New York Press, Albany, NY.

Wooldridge M.J., (2000), *Reasoning about Rational Agents*, MIT Press.

Zeng D.D. et Sycara K., (1997), Benefits of Learning in Negotiation, In *Actes de la conférence AAAI-97*.