

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE

UNIVERSITE FERHAT ABBAS – SETIF  
(UFAS). (ALGERIE)

**MEMOIRE**

Présenté à la Faculté des Sciences de l'Ingénieur

Département d'Informatique

Pour l'Obtention du Diplôme de

**MAGISTER**

Option : Sciences et Technologies de l'Information et de la Communication

Par

MR BENAZI Makhlouf

**Thème**

Négociation des offres dans les enchères par les systèmes multi-  
agents.

Soutenu le :

devant la commission d'examen :

Dr M. Touahria	MC à l'université de Sétif	Président
Dr M. Saidi	CC à l'université de Sétif	Examineur
Dr A. Boubetra	CC à CU de BBA	Examineur
Dr A. Khababa	MC à l'université de Sétif	Rapporteur

2006/2007



## Remerciements

الحمد لله الذي هدانا لهذا الذي كنا لنهتدي لولا أن هدانا الله

Il est toujours délicat de remercier l'ensemble des personnes qui ont contribué à l'aboutissement de mes travaux de recherche. Que ceux qui ne sont pas mentionnés ne m'en tiennent pas rigueur.

Je tiens à remercier tout d'abord Khababa abdallah pour avoir accepté de m'encadrer également messieurs Saidi et Boubetra pour avoir examiné ce document et Touahria pour avoir accepté de présider mon jury de mémoire et pour l'intérêt qu'ils ont porté à mes travaux.

Je remercie également tous les enseignants d'Informatique de l'université de setif qui m'ont accueillie et qui m'ont permis de m'épanouir aussi bien dans mes recherches que dans mes enseignements

Je remercie enfin ma famille et plus particulièrement mes parents, qui m'ont toujours soutenue dans mes études

# SOMMAIRE

<b>INTRODUCTION .....</b>	<b>1</b>
<b>ÉTAT DE L'ART .....</b>	<b>3</b>
1.1 KASBAH.....	4
1.2 AUCTIONBOT.....	4
1.3 FISHMARKET.....	5
1.4 MAGNET .....	8
1.5 GNP .....	11
1.6 A GENERIC SOFTWARE FRAMEWORK FOR AUTOMATED NEGOTIATION .....	14
1.7 CONCLUSION .....	18
<b>LES SYSTÈMES MULTI-AGENTS RAPPELS ET CONCEPTS .....</b>	<b>19</b>
2.1 INTRODUCTION .....	20
2.2 QU'EST-CE QU'UN AGENT ? .....	21
2.3 QU'EST-CE QU'UN SYSTÈME MULTI-AGENT ?.....	22
2.4 AGENTS ET OBJETS.....	23
2.5 AGENTS ET SYSTÈMES EXPERTS.....	24
2.6 LES AGENTS ET L'IA .....	25
2.7 LES AGENTS ET LEURS ENVIRONNEMENTS.....	25
2.8 CARACTÉRISTIQUES DES AGENTS INTELLIGENTS .....	26
2.8.1. <i>L'autonomie</i> .....	26
2.8.2. <i>Capacité à communiquer et à coopérer</i> .....	26
2.8.3. <i>Capacité à raisonner, à réagir à leur environnement</i> .....	27
2.8.4. <i>La mobilité</i> .....	27
2.8.5. <i>Autres propriétés</i> .....	27
2.9 DIFFÉRENTES CATÉGORIES ET MODÈLES D'AGENTS .....	28
2.9.1. <i>Agents cognitifs</i> .....	28
2.9.2. <i>Agents réactifs</i> .....	29
2.9.3. <i>Agents hybrides</i> .....	31
2.10 APPLICATION SMA : .....	32
2.11 LA COMMUNICATION ENTRE LES AGENTS : .....	33
2.12 ENVIRONNEMENT DE DÉVELOPPEMENT : .....	33
2.13 CONCLUSION .....	34
<b>LA NÉGOCIATION AUTOMATISÉE DANS LES SYSTÈMES MULTI-AGENTS ET LES ENCHÈRES .....</b>	<b>36</b>
3.1 GÉNÉRALITÉS .....	37
3.2 PROBLÉMATIQUE DANS LES SMA.....	37
3.3 COMPOSANTES DU MODÈLE DE NÉGOCIATION DANS LES SMA .....	38

## SOMMAIRE

---

3.4	QUELQUES MODÈLES DE NÉGOCIATION .....	38
3.4.1	<i>Le take it or leave it offer</i> .....	38
3.4.2	<i>Le vote</i> .....	38
3.4.3	<i>Négociation dans la théorie des jeux</i> .....	39
3.4.4	<i>Contract Net Protocol</i> .....	39
3.4.5	<i>Modèle générique de négociation Genca</i> .....	42
3.4.6	<i>Négociation à base de connaissances</i> .....	43
3.4.7	<i>Les négociations multi-attributs</i> .....	43
3.4.8	<i>Les négociations combinées</i> .....	44
3.5	NÉGOCIATION DANS LA VENTE AUX ENCHÈRES .....	44
3.5.1	<i>Les enchères de base</i> .....	45
3.5.2	<i>Enchères multi-objets</i> .....	48
3.5.3	<i>Les enchères combinatoires</i> .....	50
3.6	CONCLUSION .....	51
	<b>MODÈLE ET RÉALISATION</b> .....	<b>53</b>
4.1	FONDEMENTS MATHÉMATIQUES D'ENCHÈRE .....	54
4.2	ARCHITECTURE GÉNÉRALE: .....	56
4.3	CONCEPTION .....	57
4.3.1	<i>Diagramme de classe</i> .....	58
4.3.2	<i>Diagramme de collaboration</i> .....	60
4.3.3	<i>Diagramme de séquence</i> .....	62
4.3.4	<i>Diagramme d'états- transitions</i> .....	65
4.4	RÉALISATION : .....	68
4.5	CONCLUSION .....	74
	<b>CONCLUSION</b> .....	<b>75</b>
	<b>ANNEXE A</b> .....	<b>78</b>
	<b>BIBLIOGRAPHIE</b> .....	<b>80</b>

## Table de figures

<b>Figure 1</b> : Protocole d'enchères descendantes de Fishmarket .....	7
<b>Figure 2</b> : Scénario d'utilisation de GNP .....	12
<b>Figure 3</b> : Diagramme d'activité de négociation .....	15
<b>Figure 4</b> : Architecture abstraite : sous-rôles et relations .....	17
<b>Figure 5</b> : L'agent intelligent. ....	26
<b>Figure 6</b> : Modèle d'un agent cognitif .....	28
<b>Figure 7</b> : Modèle d'un agent réactif .....	30
<b>Figure 8</b> : Structures des modèles d'agents .....	31
<b>Figure 9</b> : Modèle d'agents InteRRap .....	32
<b>Figure 10</b> : Description schématique du Contract Net Protocol .....	40
<b>Figure 11</b> : Actes de langages dans le Contract Net Protocol étendu .....	41
<b>Figure 12</b> : Actes de langages dans le Contract Net Protoco .....	41
<b>Figure 13</b> : Equation de réduction .....	55
<b>Figure 14</b> : La fonction d'évaluation. ....	56
<b>Figure 15</b> : Architecture générale .....	57
<b>Figure 16</b> : Diagramme de classe .....	59
<b>Figure 17</b> : Diagramme de collaboration .....	60
<b>Figure 18</b> : Diagramme de séquence : l'inscription d'un vendeur .....	63
<b>Figure 19</b> : Diagramme de séquence : l'inscription d'un Acheteur .....	63
<b>Figure 20</b> : Diagramme de séquence : le déroulement d'un enchère .....	64
<b>Figure 21</b> : Diagramme d'état : le comportement d'agent coordinateur .....	65
<b>Figure 22</b> : Diagramme d'état : le comportement d'agent vendeur .....	66
<b>Figure 23</b> : Diagramme d'état : le comportement d'agent acheteur .....	68
<b>Figure 24</b> : La fenêtre principale .....	69

<b>Figure 25</b> : Initialisation de vendeur .....	70
<b>Figure 26</b> : Stratégie d'augmentation de prix .....	71
<b>Figure 27</b> : La stratégie de la diminution de la quantité .....	72
<b>Figure 28</b> : Suivre de l'enchère .....	73

# **INTRODUCTION**

La négociation dans les systèmes multi-agents est un problème qui a longtemps été abordé et pour lequel plusieurs solutions ont été déjà apportées. Ces solutions permettent à ces entités autonomes « agents » de dialoguer pour atteindre un compromis. Il existe actuellement de nombreux protocoles de négociation. Néanmoins, chacun de ces protocoles présente des avantages et des faiblesses qui délimitent son application à certains domaines. Dans le cadre de notre mémoire, nous nous intéressons essentiellement à la négociation combinée dans le commerce électronique c'est-à-dire l'enchère combinatoire où le négociateur veut acquérir plusieurs services ou items interdépendants.

Afin de réaliser ce mémoire nous avons débuté, comme il se doit, par un état de l'art des enchères automatisé à base d'agent. Dans le deuxième chapitre nous avons abordé le concept d'agent et le système multi-agent, tout en précisant leurs caractéristiques et catégories. Par la suite, nous avons présenté une vaste revue de la littérature des domaines de la négociation automatique et des enchères tout en précisant les modèles de négociation existants et les types des enchères en débutant par les quatre types d'enchères de base : L'enchère anglaise, L'enchère hollandaise, L'enchère premier prix à enveloppes scellées et L'enchère Vickrey en suite les enchères multi-objets et en fin les enchères combinatoires et les problèmes liées à ce dernier type d'enchère. Dans la première partie du quatrième chapitre nous avons proposé un modèle formelle pour les enchères combinatoires La deuxième partie présente la conception de l'application permettant de simuler ce modèle au moyen d'un système multi-agents. la troisième partie de ce chapitre est consacrée pour présenter quelques interfaces de notre application. Et en fin une conclusion générale et prospective.

# **ÉTAT DE L'ART**

Les différentes formes de négociations sont étudiées par de nombreux chercheurs du monde entier et différentes plates-formes de négociation ont vu le jour. Dans ce premier chapitre nous allons présentons les principaux travaux en la matière.

## 1.1 Kasbah

L'une des avancées récentes les plus prometteuses de la technologie des agents reliée au courtage de marchands est le système Kasbah [Chavez and Maes, 1996] du MIT Media Lab.

Un utilisateur qui veut acheter ou vendre des biens crée un agent, lui donne quelques directives stratégiques et l'envoie dans un marché d'agents centralisé. Les agents Kasbah se mettent à la recherche active d'acheteurs ou de vendeurs potentiels et négocient avec eux pour le compte de leurs propriétaires. Le but de chaque agent est de conclure un marché acceptable, assujetti à des contraintes spécifiées par l'utilisateur, comme le prix désiré, le prix plafond (ou plancher) acceptable, la date à laquelle la transaction doit être effectuée et la stratégie de négociation choisie parmi les 3 proposées qui correspondent aux fonctions linéaire, quadratique et exponentielle pour le calcul de l'évolution du prix selon le temps.

La négociation dans Kasbah est simple. Une fois que les agents acheteurs et les agents vendeurs sont appariés, la seule action valide du protocole de négociation est que les agents acheteurs peuvent proposer une offre aux agents vendeurs, sans restriction de délai ou de prix. Les agents vendeurs répondent par un « oui » ou un « non » ayant force exécutoire.

Du point de vue de l'implémentation, Kasbah met en relation les agents ayant des buts communs, les communications entre agents se font de 1 vers 1. Le fonctionnement en parallèle des agents est simulé en accordant un time-slice à chacun à tour de rôle. Durant ce time-slice, l'agent détermine le prix courant désiré, décide avec quel agent communiquer et enfin communique avec celui-ci. Les agents communiquent via des actes de langages spécifiques à Kasbah.

## 1.2 AuctionBot

AuctionBot[Wurman et al., 1998] est une application agents développé et opérationnel au laboratoire d'intelligence artificielle de l'université du Michi-

gan par Michael P. Wellman et Peter R. Wurman. Elle fournit un serveur de ventes aux enchères : chaque utilisateur crée une vente aux enchères en sélectionnant le type de la vente aux enchères, et en spécifiant ses paramètres (la date limite, la méthode pour résoudre les conflits entre des offres, etc.). Les acheteurs et les vendeurs peuvent ainsi négocier en respectant les protocoles multilatéraux distribués de la négociation de cette vente aux enchères. En fait, un vendeur règle le prix de réservation de sa vente aux enchères et laisse AuctionBot diriger les acheteurs qui font des offres selon les protocoles et les paramètres de la vente aux enchères. De plus, AuctionBot est un marché électronique qui donne aux acheteurs la possibilité de créer ses agents autonomes propres : c'est à l'utilisateur de programmer grâce à ses API les stratégies de proposition des offres pour son agent. AuctionBot permet de négocier au cours d'une vente aux enchères, mais elle ne prend en compte que le prix.

Du point de vue de l'implémentation, les agents placent les enchères dans la base de données, tandis qu'AuctionBot collecte les enchères, détermine une allocation d'après un ensemble de règles d'enchères bien défini et avertit les participants. En revanche, il n'exécute pas les transactions, il n'impose pas les échanges ni ne vérifie la crédibilité des participants. Ce n'est donc pas l'agent initiateur du contrat qui gère l'enchère mais un programme d'AuctionBot.

### **1.3 Fishmarket**

Fishmarket [Noriega, 1998] est une agence d'enchères électroniques pour vendre du poisson, où les agents peuvent être soit humains, soit virtuels et qui a été développée à l'Institut de Recherche en Intelligence Artificielle en Espagne par Pablo Noriega.

Le marché au poisson est une place où se déroulent différentes scènes qui ont lieu en même temps mais en des pièces différentes et qui ont une continuité causale. Chaque scène implique divers agents qui agissent selon des règles bien définies à ce moment précis. La scène principale est l'enchère où les acheteurs font des offres pour des caisses de poisson présentées par un commissaire-priseur qui annonce les prix dans l'ordre décroissant (protocole d'enchères descendantes). Cependant, avant que ces caisses de poisson soient vendues, les pêcheurs ont dû livrer le poisson au marché (dans la scène d'admission des vendeurs) et les acheteurs ont dû s'enregistrer dans le marché (dans la scène d'admission des acheteurs). De la même façon, une fois une caisse de poisson vendue, l'acheteur doit la retirer en passant par la scène des règlements des

acheteurs, pendant que les vendeurs doivent retirer leur paiement dans la scène des règlements des vendeurs une fois leur lot vendu.

La Figure 1 montre le protocole utilisé pour la vente des caisses de poisson. Le commissaire-priseur annonce le prix de vente du poisson et collecte les offres des acheteurs. L'offre d'un acheteur est une promesse d'achat au prix indiqué par le commissaire-priseur. Une fois les offres collectées pendant la période de temps allouée, le commissaire-priseur compte les offres reçues. S'il n'y en a pas, il décrémente le prix et annonce à nouveau le prix de vente du poisson. S'il y en a une, l'acheteur ayant fait l'offre remporte la vente si son crédit le permet, sinon, il doit payer une amende et le commissaire-priseur remet en vente le poisson au prix incrémenté de 25%. Si plusieurs offres sont reçues, le commissaire-priseur déclare qu'il y a eu collision et remet en vente le poisson à un prix incrémenté de 25%. Lorsque le poisson a été vendu, le crédit de l'acheteur est mis à jour, le vendeur est payé et l'acheteur peut chercher son poisson.

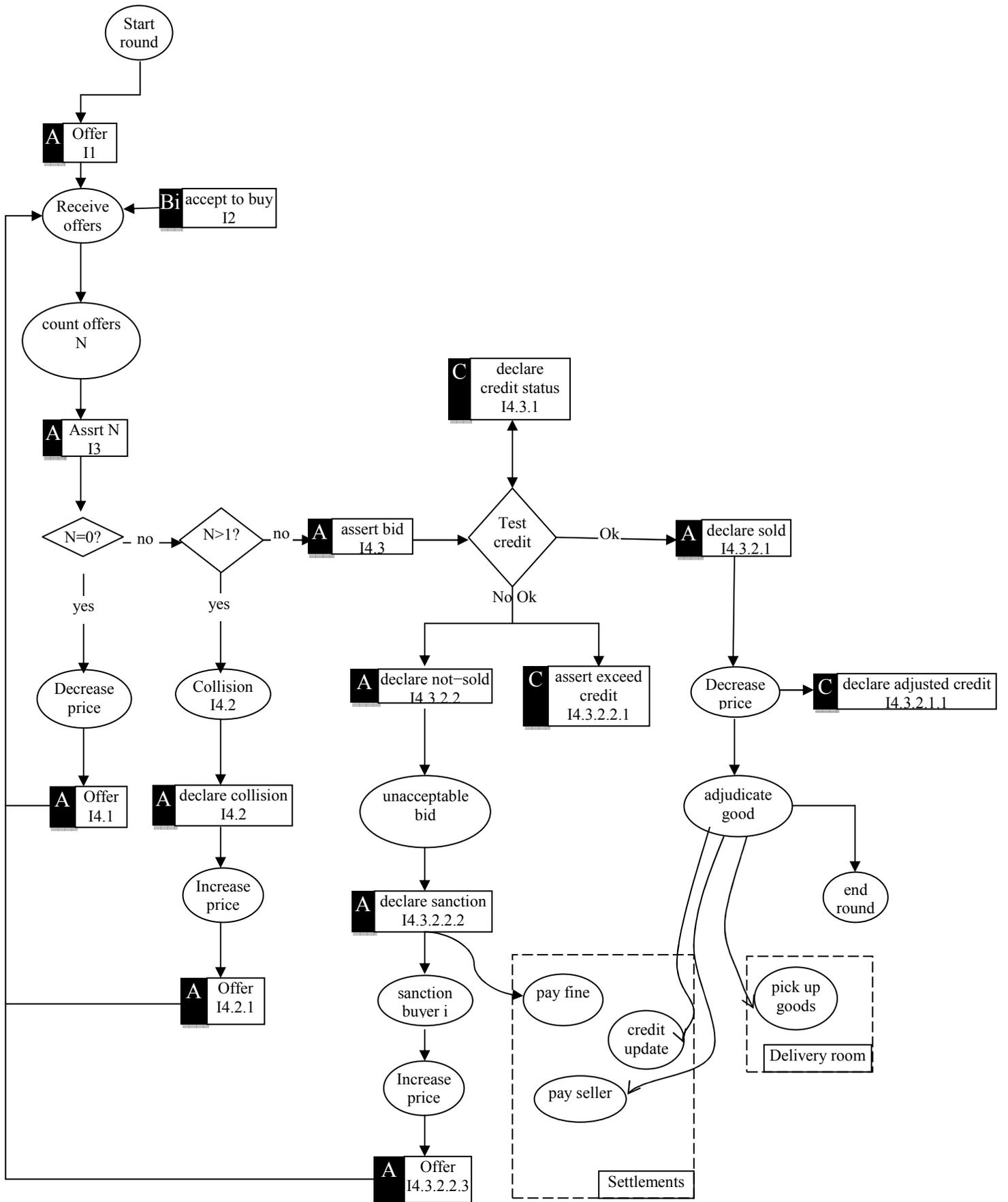


Figure 1 Protocole d'enchères descendantes de Fishmarket (Figure issue de [Rodriguez-Aguilar et al., 1997])

Un aspect important du marché au poisson, qui peut être directement transféré à la version électronique, est la présence d'intermédiaires du marché : le commissaire priseur, un patron du marché, un réceptionniste et un officier de crédit. Ces intermédiaires interagissent avec les acheteurs et les vendeurs pour le compte du marché et ont donc l'autorité de demander, reconnaître, rejeter ou accepter toutes les actions que les acheteurs et les vendeurs doivent exécuter au sein du marché. De plus, toutes ces interactions entre les intermédiaires du marché et les agents extérieurs (acheteurs ou vendeurs) peuvent en fait être associées à des allocutions standardisées, dont certaines sont probablement tacites dans le marché, mais néanmoins rendues explicites dans le modèle computationnel.

Fishmarket a été conçu pour montrer la complexité de ces interactions tout en gardant aussi fortement que possible une similarité avec l'ontologie des éléments du marché de poisson. Chaque agent logiciel de Fishmarket représente soit un intermédiaire du marché, soit un vendeur ou encore un acheteur. De même les allocutions (tacites ou explicites) utilisées dans le marché ont été reprises en allocutions toujours explicites pour les agents logiciels.

Fishmarket utilise un cadre dialectique et peut être étendu afin de convenir aux différents types d'enchères (anglaises, hollandaise, Vickrey, etc.).

## 1.4 Magnet

Magnet (Multi AGent NEgotiation Testbed) [Collins et al., 1998] est un banc de tests pour la négociation multi-agents, implémenté sous la forme d'une architecture généralisée de marché et développé à l'université du Minnesota par John Collins et al. Il fournit un support pour un large panel de transactions, du simple achat/vente de biens à la négociation complexe de contrats entre agents. Le but des auteurs est de concevoir, implémenter et analyser une architecture de marché multi-agents généralisée, qui peut fournir un support explicite et intégré pour les interactions complexes entre agents, comme dans la prise de contrats automatisée, tout comme pour d'autres types de protocoles de négociation, incluant les enchères à offres scellées et la vente et l'achat à prix public ou offres publiques. Les auteurs introduisent également un protocole de prise de contrats flexible qui peut prendre complètement avantage de l'architecture de marché proposée pour faciliter les interactions entre agents. Magnet introduit un intermédiaire explicite dans la négociation qui aide à contrôler les fraudes et décourage les contrespéculation et est organisé autour de trois composants de base : l'échange, le marché et la session.

L'échange est une collection de marchés spécifiques à un domaine dans lesquels les services et les biens sont échangés, agrémentée par des services génériques requis par tous les marchés, comme la vérification des identités des participants à une transaction.

Chaque marché au sein d'un échange est un forum pour le commerce d'une commodité particulière ou d'un secteur d'activités. Il y a des marchés voués à la publication, à la construction, au transport, etc. Chaque marché inclut un ensemble de services et d'aménagements spécifiques à son domaine et chaque marché s'appuie sur les services communs de l'échange.

Un composant important de chaque marché est l'ensemble des sessions de marché courantes dans lesquelles les interactions effectives entre agents ont lieu. Les agents participant à un marché doivent le faire en tant qu'initiateur de session, de client ou les deux à la fois. Chaque session est initialisée par un agent pour un propos particulier et en général de multiples agents peuvent rejoindre une session existante en tant que clients. Les éléments importants du marché incluent une ontologie spécifique au domaine spécifiant les termes du discours pour ce domaine, une spécification de protocole qui définit les types de négociation supportés par le marché et un registre de clients au sein du marché.

Une session (de marché) est le moyen par lequel les services du marché sont délivrés dynamiquement aux agents participant. Il sert aussi bien d'encapsulation pour une transaction au sein du marché, que de répertoire persistant pour l'état courant de la transaction. La session entoure la vie entière d'un contrat ou d'un ensemble de contrats relatifs. Le mécanisme de session assure la continuité des transactions à demi réalisées, protège contre la fraude, limite la contre-spéculation et permet aux agents participants de ne pas avoir à garder localement une trace des statuts détaillés de la négociation. Ce mécanisme de session permet aux agents de lancer un appel d'offres et de conduire d'autres travaux simultanément.

Le protocole de négociation pour la planification par contrats est constitué de trois phases : un appel d'offres, la collecte des enchères et la notification des résultats.

Un appel d'offres contient un ensemble de sous-tâches avec pour chaque sous-tâche l'intervalle de temps durant lequel le travail doit être accompli. Ce message inclut également une date limite de réception des offres, la date à laquelle le client étudiera les propositions, la date au plus tôt où les notifications

d'acceptation seront envoyées, et pour chaque sous-tâche une fonction de pénalité indiquant aux fournisseurs la pénalité en fonction du temps, dont ils seront redevables si ils n'arrivent pas à (décident de ne pas) accomplir le travail.

Ce message, une fois créé, est donné à la session de marché qui le rend disponible à tous les fournisseurs appropriés (ceux qui sont enregistrés au sein du marché et qui peuvent fournir les tâches nécessaires). En ce sens, l'appel d'offres est public, tandis que tous les autres messages sont privés. Avant de faire suivre le message, la session peut le vérifier pour être sûr qu'il est conforme à toutes les règles de marché et d'échange qui peuvent exister.

Chaque fournisseur ayant reçu l'appel d'offres l'inspecte et décide de faire ou non une offre, selon ses ressources, ses contraintes de temps et ses connaissances sur le travail à accomplir, selon le catalogue des services fourni par l'agent du marché.

Une offre est un message privé, qui peut contenir une combinaison de sous-tâches, qui peut être un sous-ensemble des sous-tâches listées dans l'appel d'offres. Dans l'offre, le fournisseur doit indiquer le coût pour le client, l'intervalle de temps et la durée estimée du travail pour la combinaison entière des sous-tâches, la date limite pour l'acceptation de l'offre et une fonction de pénalité pour le client si celui-ci attribue le travail au fournisseur puis se rétracte.

Chaque fournisseur peut envoyer plusieurs offres différentes pour un même appel d'offres mais une seule de ces offres sera attribuée à chaque fournisseur. Si un fournisseur n'envoie pas d'offre avant la date limite fixée par le client, celui-ci considère que le fournisseur a décidé de ne pas faire d'offre, le refus est donc passif.

Une fois les offres reçues, le client doit décider lesquelles accepter, en utilisant ses connaissances sur les offres, les valeurs des tâches et des sous-tâches, ses propres contraintes temporelles et le fournisseur ayant proposé une offre. Une fois ce processus accompli, le client doit choisir parmi ces trois actions pour chaque offre:

- 1 accepter l'offre entière,
- 2 accepter une partie de l'offre,
- 3 rejeter l'offre (ce qui est fait de façon passive).

Le message d'acceptation de l'offre sera envoyé à la session de marché qui le vérifiera, le validera et l'estampillera avant de le faire suivre au fournis-

seur. Les deux premiers choix sont des engagements à attribuer le travail au fournisseur et à partir de la date d'envoi du message, les fonctions de pénalité du client et du fournisseur sont effectives. Si un fournisseur ne reçoit pas d'acceptation de son offre, cela veut dire que le client l'a rejetée. Une fois les engagements pris, un agent peut déterminer qu'il ne peut pas faire les tâches pour lesquelles il s'est engagé, ou qu'il lui serait désavantageux de les faire. Dans ces situations, l'agent doit envoyer un message de désengagement indiquant quelles parties du travail il ne peut pas satisfaire. Le message de désengagement comprend également une acceptation de la pénalité à payer par l'agent du fait de ce désengagement.

## 1.5 GNP

La plate-forme de négociation générique (GNP) est une plateforme de négociation d'entreprises à entreprises orientée enchères conçue pour s'insérer dans une place de marché, elle est réalisée par Morad Benyoucef, Rudolf Keller et leurs collègues ([Benyoucef et al., 2000, Gerin-Lajoie, 2000]) au département IRO de l'université de Montréal. GNP est un système multi-tiers avec 5 tiers :

- Les participants : acheteurs, vendeurs et régisseurs de marché rejoignent les serveurs GNP avec leurs navigateurs web, soient par des pages HTML, soient par des composants Java (des Applets) activés à l'intérieur du navigateur web ou directement dans des agents Java ou JPython.

- Le portail HTTP de contenu, utilisant des Servlets Java et des pages « Java Server Pages », JSP. Ce serveur est bâti sur l'infrastructure J 2EE fournit par le serveur BEA-Weblogic;

- Le serveur de requêtes GNP, utilisant les « Enterprises Java Beans » (EJB) pour les sessions et les entités;

- Le serveur des encanteurs, des « Enterprises Java Beans » activés par des messages asynchrones « Java Message Service » (JMS);

- Le serveur de données SQL. GNP utilise la base de données PostgreSQL et Oracle 8i.

La négociation entre les agents s'effectue via le service de négociation qui s'occupe de l'appariement entre la demande des acheteurs, l'offre en bien et services des vendeurs et la fixation des prix. Cette plateforme reçoit les an-

nonces des acheteurs ou des vendeurs et les mises qui répondent à ces annonces. Les règles de marché sont définies dans des fichiers XML.

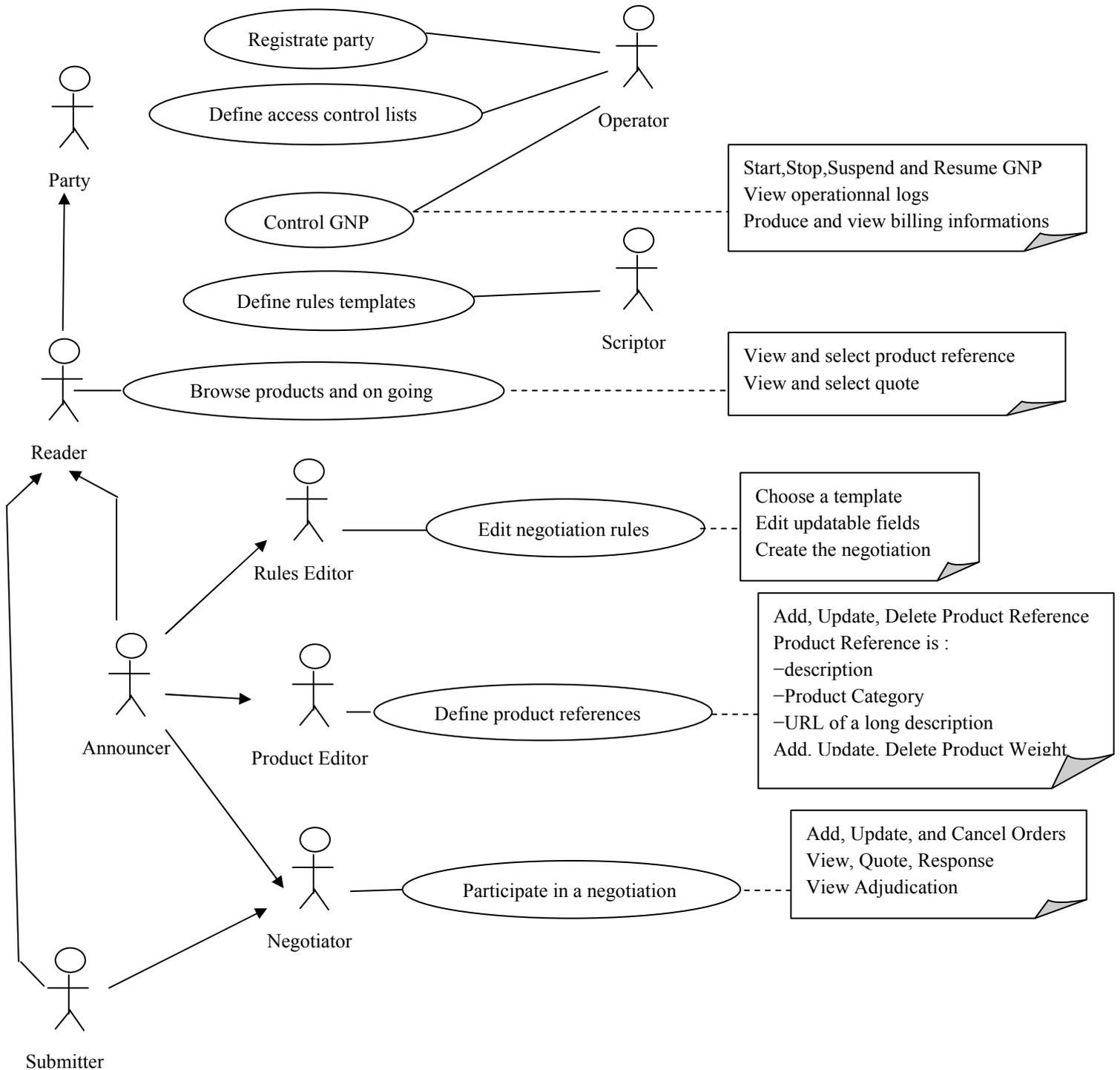


Figure 2 Scénario d'utilisation de GNP (Figure issue de [Gerin-Lajoie, 2000])

La Figure 2 illustre le scénario d'utilisation principal, les acteurs et leurs rôles dans le processus de négociation.

Lecteur - "Reader" Tous les participants peuvent a priori être des lecteurs. Ils peuvent consulter les produits, les annonces d'achats et de vente, voir les cotes publiques et finalement sélectionner celles sur lesquelles ils vont réagir et soumissionner.

Annonceur - "Announcer" Un annonceur crée une négociation. Il annonce un intérêt d'achat ou de vente d'un ou plusieurs produits ou services. Les 3 étapes de la création d'une négociation sont:

- Créer les règles de la négociation à partir de modèles existants. Fixer les paramètres éditables comme l'heure de fermeture
- Créer les références aux objets de la négociation, produits ou services
- Créer un ordre initial d'achat ou de vente, définissant le prix et la quantité de départ.

Négociateur et soumissionnaire - "Negotiator and Submitter" Un négociateur émet les ordres d'achats ou de vente. Un acheteur et un vendeur peuvent donc être un négociateur. Le négociateur émet l'ordre initial d'achat ou de vente. Le soumissionnaire est un négociateur répondant à une cote avec un ordre d'achat ou de vente.

Administrateur et opérateur - "Administrator and Operator" Les administrateurs et opérateurs gèrent les participants et leurs priorités, les marchés et les négociations. Sur les marchés et les négociations, ils peuvent non seulement les ajouter, modifier et enlever, mais aussi les démarrer, suspendre et repartir.

Déroulement d'une négociation L'initialisation d'une négociation sur GNP se fait par l'envoi d'un document XML initial : l'annonce, qui contient tous les éléments d'entrée nécessaires au démarrage d'une négociation.

Le participant désirant initialiser une négociation sur GNP dispose en premier lieu d'un choix de différents modèles de négociation (template). À un modèle correspond un formulaire d'annonce HTML unique. Le formulaire permet à l'utilisateur de définir certains paramètres du modèle et ainsi de garantir une flexibilité d'implantation à l'intérieur de chaque modèle. Le formulaire complété et envoyé donnera naissance à l'annonce, document au format XML.

Il est également possible d'envoyer directement le document d'une annonce à GNP. Dans ce cas, le document de l'annonce doit être entièrement édité manuellement dans le format XML valide et attendu, dans un fichier.

L'annonce donne à GNP tous les éléments d'entrée nécessaires au démarrage d'une négociation. L'annonce apparaît à l'utilisateur dans un format HTML. Elle contient toujours une partie règles (rules) et, selon le mécanisme choisi, une partie produit (productReference) et une partie ordre (order). La partie règles établit précisément les paramètres fixes de la négociation, telles que les heures d'ouvertures et de fermetures, l'incrément minimum, les conditions de fin de tours et les conditions d'équilibrage du marché. Si ceux-ci doivent varier, la négociation est divisée en plusieurs phases. La partie produit définit techniquement la nature du produit à négocier, renvoie à sa description précise. La partie ordre représente les données de la mise initiale faite par le participant, telles que son prix de départ, son prix de réserve, la quantité de produit mise en négociation et ses fractions permises. Lorsque l'annonce est remplie, elle est envoyée au système qui va déclencher une session d'annonce.

Les utilisateurs interagissent avec GNP uniquement via un navigateur web. Aucune installation de logiciel ne doit être effectuée.

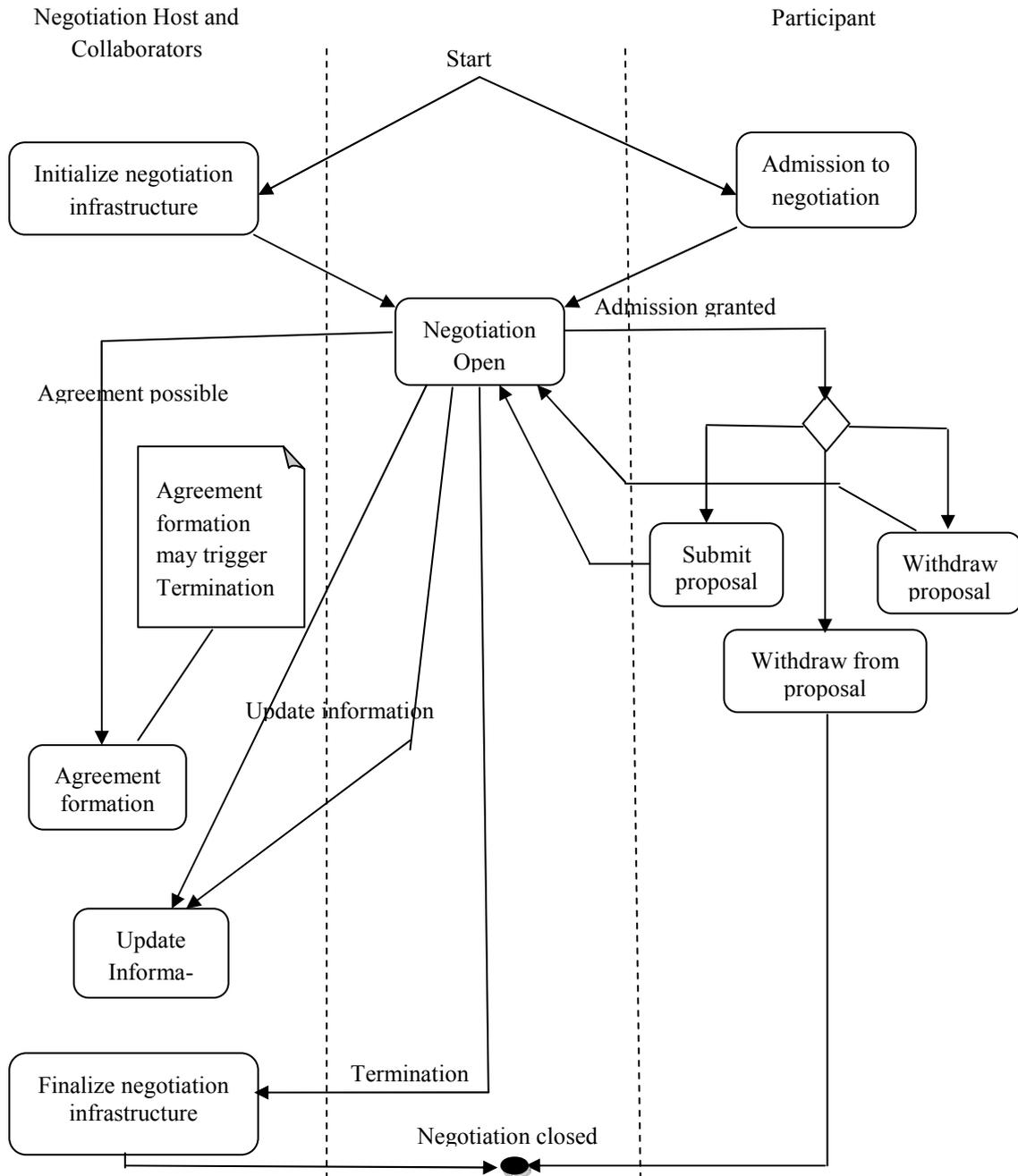
D'un point de vue formel, les auteurs clament dans [Benyoucef et al., 2000]:

"we have identified a number of operations that are common to different negotiation processes. Some of these operations are:

- define attributes and default values for the formalized concepts;
- setup the end conditions for rounds, phases and the whole negotiation;
- define the information to be displayed to or hidden from the players".

## **1.6 A generic software framework for automated negotiation**

Claudio Bartolini et ses collègues ([Bartolini and Preist, 2001, Bartolini et al., 2002b, Bartolini et al., 2002a]), des laboratoires HP, ont développé un framework générique pour la négociation automatique dédié aux mécanismes de marché. Ce framework comporte un protocole général de négociation qui se paramètre par des règles de négociation. En choisissant un ensemble de règles, différents mécanismes de négociation peuvent être réalisés. La négociation se déroule dans une scène de négociation qui est une abstraction du système de messages utilisés par les participants de la négociation pour communiquer. Après avoir été admis à la négociation, un participant peut accéder à la scène de négociation. Chaque participant peut envoyer des propositions via un message destiné à l'hôte de négociation. Ce dernier se charge de la transmission du message aux participants concernés.



**Figure 3** Diagramme d'activité de négociation (Figure issue de [Bartolini et al., 2002b])

Le processus de négociation (Figure 3) consiste à parvenir à un accord acceptable depuis le template de négociation. Au cours de la négociation, les participants échangent des propositions représentant des accords actuellement acceptables pour eux. Chaque proposition contient des contraintes sur tout ou partie des paramètres du template de négociation. Ces propositions sont envoyées à l'hôte de négociation. Cependant, avant qu'une proposition soit acceptée dans la scène, elle doit être validée. Pour être valide, une proposition doit satisfaire deux critères:

- Ce doit être une restriction valide de l'espace des paramètres défini dans le template de négociation.
- La proposition doit être soumise conformément à l'ensemble des règles qui gouvernent le déroulement de la négociation. Ces règles spécifient entre autres qui peut faire une proposition, quand et quelle proposition peut être faite selon les propositions antérieures.

Un accord est formé conformément aux règles de formation d'accord associées à la scène de négociation. Lorsque les propositions dans la scène satisfont certaines conditions, elles sont converties par ces règles en accords et retournées aux participants. La fin de la négociation est déterminée par des règles de terminaison.

Ce framework de négociation se base sur : (i) un protocole général de négociation, (ii) une taxonomie des règles de négociation, (iii) un langage pour définir les règles de négociation, et (iv) un langage pour exprimer les propositions de négociation. Il y considère deux rôles principaux dans la négociation : le participant et le hôte. Les participants sont ceux qui veulent aboutir à un accord. Un participant peut poster des propositions selon les règles fournies par l'hôte de négociation. L'hôte est chargé de créer et de faire respecter les règles gouvernant la participation, l'exécution, la résolution et la terminaison de la négociation. Il peut ou non être également un participant. Il a les sous-rôles suivants (Figure 4) :

- *Gatekeeper* : fait respecter la politique d'admission à la négociation,
- *Proposal validator* : assure qu'une proposition est conforme au template de négociation,
- *Protocol enforcer* : assure que les propositions des participants sont postées et enlevées selon les règles de négociation,
- *Agreement maker* : assure que les accords sont formés selon les règles,
- *Information updater* : notifie les participants de l'état courant de la négociation, selon les règles de visibilité et d'affichage,
- *Negotiation terminator* : déclare la fin de la négociation en se basant sur la règle de terminaison.

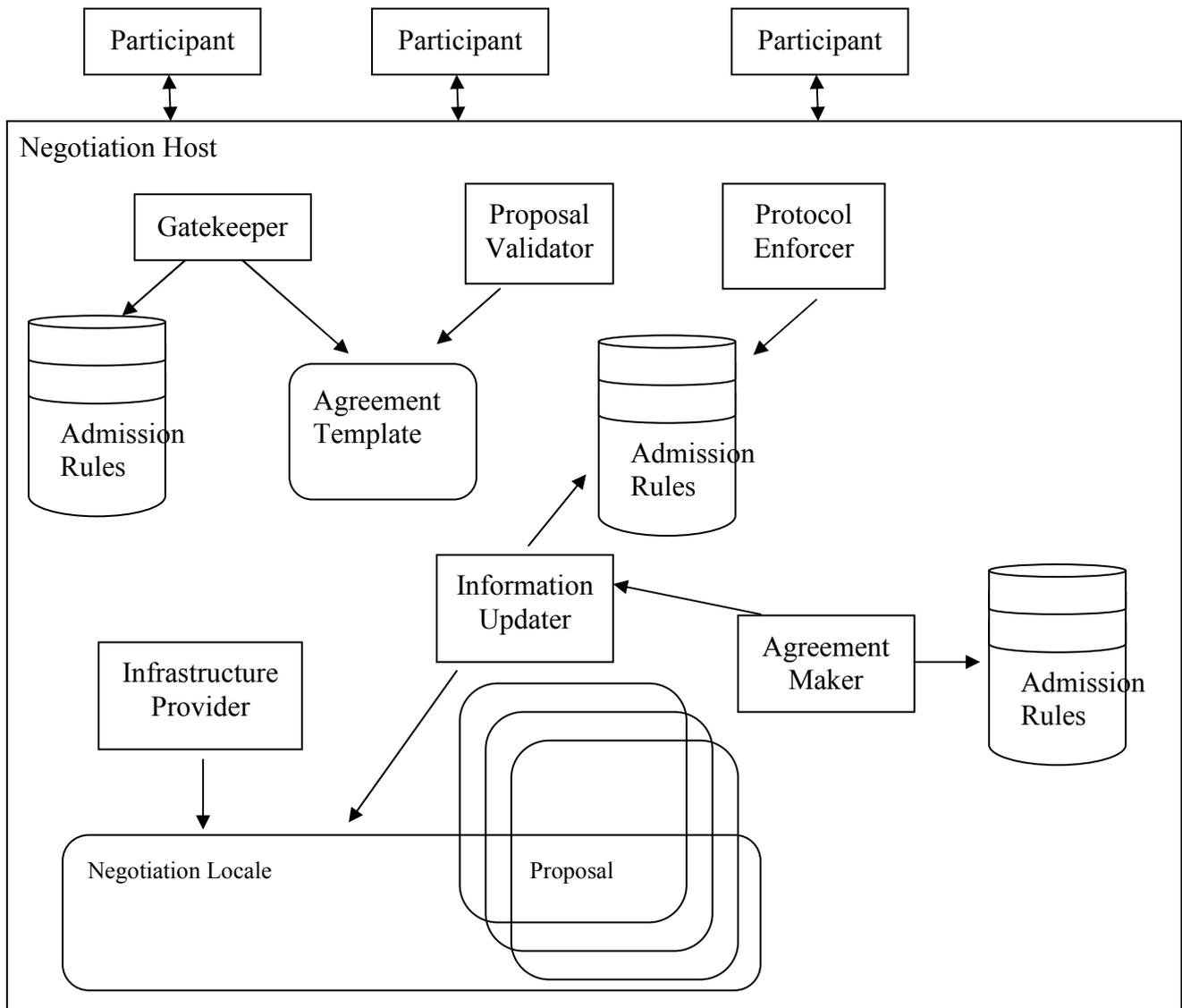


Figure 4 Architecture abstraite : sous-rôles et relations (Figure issue de [Bartolini et al., 2002b])

Ce framework a été implémenté en utilisant la plate-forme Jade, les agents communiquent par envoi de messages *FIPA-ACL* et Jess a été utilisé pour le traitement des règles. Une première critique pour ce framework est l'utilisation du Jess qui est un système assez lourd pour l'interprétation des règles. Les auteurs travaillent à utiliser DAML+OIL au lieu de Jess pour avoir une implémentation plus ouverte et indépendante de la plateforme et à incorporer d'autres formes de négociation telles que les accords multi-parties, le Contract Net et la négociation par argumentation.

## 1.7 Conclusion

Nous avons présenté dans ce chapitre les différents systèmes de négociation et des enchères réalisés par d'autres équipes de recherche, et on a constaté que la plupart des systèmes présentés dans ce chapitre sont spécialisés dans un type d'enchère bien spécifié surtout les enchères de base comme l'enchère anglaise simple dans Kasbah ou l'enchère hollandaise simple dans Fishmarket d'autres plateforme sont spécialisé dans les enchères multi-objets de même type comme dans Generic Negotiation Platform ou GNP ([Benyoucef et al., 2000]).

Dans notre système on va essayer d'implémenter une enchère combinatoire paramétrable où l'acheteur peut entrer ses propres stratégies de négociation et le vendeur peut vendre plusieurs objets déferents en même temps.

Dans le chapitre suivant on va essayer de cerner la définition d'agent et système d'agent toute en précisant leurs types et leurs caractéristiques.

# **Les Systèmes Multi-Agents**

## **Rappels et Concepts**

## 2.1 Introduction

Les Systèmes Multi-Agent (SMA) sont apparus au carrefour des recherches sur l'intelligence artificielle distribuée et sur la vie artificielle. Ces systèmes sont développés à partir de schémas de raisonnement ou d'organisations empruntées aux domaines de la vie et de la société [Ferber, 1995]. Ils empruntent à l'intelligence artificielle distribuée les modes de communication et de concertation entre agents. Ils reprennent les idées d'autonomie et d'émergence du résultat final à partir des interactions individuelles liées à la vie artificielle.

Les SMA sont particulièrement adaptés pour proposer des solutions réactives et robustes à des problèmes complexes pour lesquels il n'existe pas de contrôle centralisé [Jennings, 1999 ; Casteran et al., 2000]. Ils sont des systèmes idéaux pour représenter des problèmes possédant de multiples méthodes de résolution, de multiples perspectives et/ou de multiples solveurs (Chabdraa et al., 2001). Ces systèmes possèdent les avantages traditionnels de la résolution distribuée et concurrente de problèmes comme la modularité, la vitesse (avec le parallélisme), et la fiabilité (due à la redondance). Ils héritent aussi des bénéfices envisageables de l'Intelligence Artificielle comme le raisonnement symbolique (au niveau des connaissances), la facilité de maintenance, la réutilisation et la portabilité, mais surtout ils ont l'avantage de faire intervenir des schémas d'interaction sophistiqués.

Les types courants d'interaction incluent :

- la coopération qui implique une activité commune d'un ensemble d'agents afin d'atteindre un but commun,
- la coordination qui suppose l'organisation de l'activité de résolution de chaque agent afin d'éviter les interactions inutiles et d'exploiter celles qui sont bénéfiques,
- la négociation dont l'objet est d'arriver à un compromis acceptable entre tous les agents engagés en compétition.

Les concepts « Agent » et « Multi-agent » sont relativement récents et jusqu'à maintenant nous n'avons pas de définition formelle de ce que sont un agent et un système multi-agent, qui soit acceptée par tout le monde [Jennings et al., 1998]. Dans ce qui suit nous donnerons quelques définitions proposées autour de ces deux concepts.

## 2.2 Qu'est-ce qu'un agent ?

Dans la littérature, on trouve une multitude de définitions du concept agent. Elles présentent certaines similitudes et dépendent du type d'application pour laquelle est conçu l'agent. Dans le cadre de notre travail, nous porterons notre attention sur les définitions qui suivent :

Ferber a proposé assimiler le terme agent à une entité physique ou virtuelle [Ferber, 1995] :

- ◆ 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, d'une fonction de satisfaction ou de survie, qu'elle cherche à optimiser,
- ◆ qui possède des ressources propres,
- ◆ qui est capable de percevoir partiellement son environnement,
- ◆ qui ne dispose que d'une représentation partielle de l'environnement,
- ◆ 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.

Cette définition introduit des caractéristiques exploitables, comme la notion d'objectif et de perception limités ou locale de l'environnement, comme c'est le cas dans plusieurs applications. En effet, un robot mobile a généralement un certain rayon de perception et d'influence, pour observer et agir sur son environnement physique. Un agent mobile dans le réseau Internet ne peut pas observer tout le réseau ; il ne peut en observer qu'une partie. De même, dans un atelier de production décentralisé l'affectation des tâches des produits sur des ressources se fait de façon locale ; en conclusion, les interactions entre les agents produits et les agents ressources dépendent en grande partie de leurs propres objectifs et des contraintes dans leur environnement local.

Wooldridge et Jennings ont défini l'agent comme étant un Système Informatique, **Situé** dans un environnement, et qui agit d'une façon **Autonome** et **Flexible** pour atteindre certains objectifs pour lesquels il a été conçu [Wooldridge et Jennings, 1995 ; Jennings et al., 1998] :

1. Situé : un agent est dit situé, s'il est capable d'agir sur son environnement à partir des entrées sensorielles qu'il reçoit de ce même environnement,
2. Autonome : un agent est dit autonome s'il est capable d'agir sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne,
3. Flexible : un agent est dit flexible s'il est capable d'agir de manière réactive, proactive et sociale :
4. Réactif : être capable de percevoir son environnement et de réagir dans le temps,
5. Proactif : prendre l'initiative et être opportuniste au bon moment,
6. Social : être capable d'interagir avec les autres agents quand la situation l'exige (pour compléter ses tâches ou coopérer avec eux).

Un agent est donc une entité physique ou virtuelle en situation dans un environnement avec lequel il interagit de façon autonome et flexible. Cette notion de situation ou d'agent situé implique que les agents autonomes doivent agir dans un monde réel, en perpétuel changement, partiellement observable et intrinsèquement imprévisible. De même ils doivent réagir en temps réel parce que l'environnement change constamment, et parce qu'ils doivent éventuellement tenir compte des actions d'autres agents [Drogoul et Meyer, 1999].

### 2.3 Qu'est-ce qu'un Système Multi-agent ?

Comme pour les agents, plusieurs définitions ont été proposées, et nous allons commencer par présenter la définition formelle introduite par Ferber. Un SMA est composé des éléments suivants :

- ◆ un Environnement  $E$  c'est à dire un espace disposant généralement d'une métrique,
- ◆ un ensemble d'objets  $O$ , ces objets sont situés, c'est à dire que pour tout objet, il est possible, à un moment donné, d'associer une position dans  $E$ . ils sont passifs, ils peuvent être perçus, détruits, créés et modifiés par les agents,
- ◆ un ensemble d'Agents  $A$  qui sont les entités actives du système  $A \subset O$ ,
- ◆ un ensemble de Relations  $R$  qui unissent les objets entre eux,
- ◆ un ensemble d'Opérations  $Op$  permettant aux agents de percevoir, de détruire, de créer, de transformer et de manipuler les objets de  $O$ ,

- ◆ un ensemble d'opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification (les lois de l'univers).

Une autre définition est celle donnée par Wooldridge et Jennings, qui présente un SMA comme étant un ensemble d'agents en interaction afin de réaliser leurs buts ou d'accomplir leurs tâches [Wooldridge et Jennings, 1995]. Les interactions peuvent être directes par l'intermédiaire des communications, comme elles peuvent être indirectes via l'action et la perception de l'environnement. Les interactions peuvent être mises en œuvre dans un but de coopération entre les agents, lorsqu'ils ont des buts communs,

- ◆ coordination, c'est à dire d'organisation pour éviter les conflits et tirer le maximum de profit de leurs interactions afin de réaliser leurs buts,
- ◆ compétition, lorsque les agents ont des buts antagonistes.

Dans le mémoire nous adoptons la définition de Wooldridge et Jennings, un SMA étant un ensemble d'agents ayant des buts ou des tâches, et qui interagissent pour les accomplir en mode de coopération et/ou de compétition.

## 2.4 Agents et Objets

Un **objet** est défini par un ensemble de services offerts (ses méthodes) qu'il ne peut refuser d'exécuter si un autre objet le lui demande. Les **objets** exécutent des tâches ; ils n'ont ni but, ni recherche de satisfaction. En revanche les **agents**, bien plus élaborés, disposent d'objectifs qui leur donnent une autonomie de décision vis à vis des messages qu'ils reçoivent.

Les **objets** utilisent un mécanisme d'envoi de message qui se résume à un simple appel de méthode. Pour les **agents**, les interactions sont plus complexes et font intervenir des communications de haut niveau, où l'important est que l'agent décide par lui-même comment interagir et réagir aux messages qu'il reçoit.

Un **agent** est une entité capable de recevoir et d'émettre des messages. Ce comportement est minimaliste, et correspond d'ailleurs tout à fait à la définition d'un objet. Néanmoins, un objet est contraint de répondre aux requêtes qui lui sont soumises, contrairement à l'agent qui dispose de son propre libre arbitre. Concrètement, les objets font ce qu'on leur demande, tandis qu'avec les agents, il faut négocier. Donc, il est important que les agents contrôlent eux-mêmes leur comportement et les ressources qu'ils possèdent. C'est ce qui caractérise

leur autonomie, et d'une certaine manière leur donne conscience de leurs possibilités.

En d'autre sens :

Les *objets* font les choses gratuitement du moment qu'ils ne raisonnent pas et qu'ils ne décident pas. Ça a été décidé par le concepteur.

Les *agents* font les choses parce qu'ils le veulent. En effet, ils font les choses suite à un raisonnement et/ou à une prise de décision

Dans un **SMA**, l'originalité et la source de toute la problématique **SMA** résident dans l'hypothèse d'autonomie de décision attribuée aux agents, c'est-à-dire dans la conservation d'un maximum de degrés de liberté pour chacun de ces agents. Cette perspective d'autonomie introduit une dynamique qui complexifie le maintien de la cohérence et de l'efficacité du système. Il est ici indispensable de combiner une communication plus évoluée que les objets avec une recherche de satisfaction des objectifs de chaque agent.

Néanmoins, le lien existant entre la notion d'objet et celle d'agent reste fort. Il n'y a pas de frontière nette : un objet peut être considéré comme un agent dont le langage d'expression se résume à l'emploi de mots clés correspondant à ses méthodes. Un autre aspect qui rajoute un peu plus à la confusion entre la notion d'objet et celle d'agent, est que souvent un agent est implémenté, pour des raisons pratiques, sous la forme d'objets. Par conséquent, on peut dire que les agents sont définis à un niveau conceptuel tandis que les objets sont définis à un niveau d'implémentation et d'exécution.

## 2.5 Agents et systèmes experts

Beaucoup disent que les agents sont des Systèmes Experts (SE)?

Un Systèmes Experts généralement encapsule une expertise par exemple: un système expert type diagnostic est un ensemble de règles qui disent quoi faire dans telle ou telle situation. Un dépanneur type garagiste ou autre pourrait se faire aider par un tel système.

Les principales différences un agent ne reflète pas forcément une expertise, il pourrait toutefois le faire (certains Systèmes Experts temps réel sont des agents);

Une autre différence entre un Système Expert et un agent est un Système Expert n'est pas situés forcément dans un environnement sur lequel il agit.

## 2.6 Les Agents et l'IA

Comment se situent les agents par rapport à l'IA dans son ensemble?

L'IA vise à construire des systèmes capables de : comprendre le langage; Reconnaître des scènes; Planifier et re-planifier en présence d'incertain, etc.; Reasonner selon le sens commun; Etc.

Les agents c'est juste choisir la bonne action au bon moment et ce, dans un environnement limité. Pas besoin de tout ce que l'IA veut résoudre; Un agent pourrait avoir « peu » d'intelligence.

## 2.7 Les Agents et leurs environnements

L'environnement entourant l'agent pourrait être:

- accessible ou non;
- déterministe ou non;
- statique ou dynamique;
- discret ou continu.

*Accessible ou non* : Un environnement est dit accessible si l'agent peut obtenir une information à jour, précise et complète de l'état de cet environnement; La plupart des environnements complexes (y compris le monde physique qui nous entoure) sont inaccessibles; Plus l'environnement est accessible plus il est facile pour le ou les concepteurs.

*Déterministe ou non* : Un environnement est dit déterministe si toute action a un simple effet garanti. Ainsi il n'y a pas d'incertain associé à l'état résultant de l'exécution d'une action; Le monde physique à bien des égards, peut être vu comme un monde non-déterministe; Un environnement non-déterministe présente (en général) plus de difficultés pour le ou les concepteurs.

*Statique ou dynamique* : Un environnement statique est un environnement qui reste inchangé si l'agent n'agit pas dessus; Un environnement dynamique est un environnement qui a d'autres processus opérant sur lui, et qui donc change en dehors du contrôle de l'agent; Le monde physique qui nous entoure est un monde hautement dynamique.

*Discret ou continu* : Un environnement est dit discret s'il englobe un nombre fini et fixe d'actions et de percepts; Le jeu d'échec constitue un environnement discret; La conduite automobile constitue un environnement continu.

## 2.8 Caractéristiques des agents intelligents

Les chercheurs en intelligence artificielle s'accordent sur la nécessité de l'existence de quelques caractéristiques pour qu'on puisse parler d'agents intelligents. A l'heure actuelle, aucun produit ne rassemble toutes ces caractéristiques.

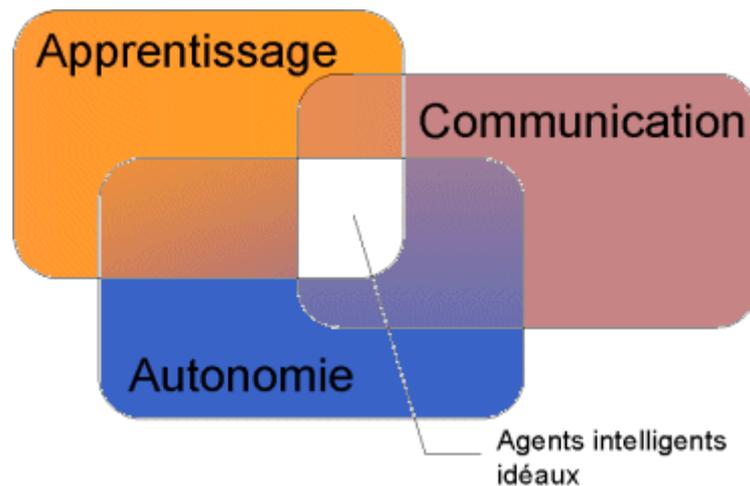


Figure 5: L'agent intelligent.

### 2.8.1. L'autonomie

L'agent doit pouvoir prendre des initiatives et agir sans intervention de l'utilisateur final. Dans le contexte du web il doit pouvoir agir alors que l'utilisateur est déconnecté. C'est le cas par exemple d'Autonomy de la société Agentware. Les agents de General Magic devraient aussi en être capable.

La plupart des éditeurs de logiciels "agents" contournent le problème de l'autonomie en programmant leurs logiciels afin qu'ils puissent, à intervalles réguliers, se connecter automatiquement au web pour y effectuer les tâches que leur ont confié les utilisateurs. Dans ce cas là, on ne peut véritablement parler d'autonomie car l'agent n'est pas actif en permanence et ne fait que reproduire comme un automate la requête programmée par l'utilisateur.

### 2.8.2. Capacité à communiquer et à coopérer

L'agent doit pouvoir échanger des informations plus ou moins complexes avec d'autres agents, avec des serveurs ou avec des humains. Le service Firefly permet, par exemple, à l'agent représentant un utilisateur, d'échanger des in-

formations avec d'autres agents représentant d'autres utilisateurs afin de rechercher ceux d'entre eux ayant les mêmes centres d'intérêts. Au fur et à mesure que le nombre d'utilisateurs augmente, le nombre des agents et la quantité des informations qu'ils échangent grandissent. Chaque agent coopère ainsi à l'accroissement des connaissances des autres. La limite de ce service est que ces échanges d'informations ne se font pas entre agents circulant sur le web mais entre agents propre au service Firefly. Les échanges d'information ne peuvent donc se faire qu'avec les abonnés au service Firefly.

### **2.8.3. Capacité à raisonner, à réagir à leur environnement**

L'agent doit être capable de s'adapter à son environnement (qui peut être composé d'autres agents, du web en général ou des utilisateurs) et aux évolutions de celui-ci. Cette adaptation doit s'appuyer sur l'analyse de l'environnement extérieur des agents.

Dans le cas de Firefly, par exemple, les connaissances d'un agent sont en partie fonction des connaissances des autres agents Firefly. A mesure que leurs connaissances s'enrichissent, les connaissances d'un agent particulier s'enrichissent aussi grâce aux échanges entre agents et aux capacités de chaque agent d'intégrer en son sein l'accroissement des connaissances dues au système d'agent Firefly qui constitue son environnement.

### **2.8.4. La mobilité**

Les agents doivent pouvoir être multi-plate-forme et multi-architecture. Ils doivent pouvoir se déplacer sur le réseau où ils accomplissent des tâches sans que l'utilisateur ait le moindre contrôle sur celles-ci.

### **2.8.5. Autres propriétés**

D'autres propriétés peuvent être discutées dans le cadre des agents;

1. *Sincérité* : un agent ne pourrait « en connaissances de cause » communiquer de fausses informations;
2. *Bénévolat* : les agents ne doivent entrer en conflit avec les autres, ils doivent toujours essayer de faire ce qu'on leur demande (au risque de délaissier leur propres tâches);
3. *Rationalité* (faible) : un agent agit pour réaliser ses buts et il ne doit en aucun cas agir en vue d'empêcher ses buts de se réaliser. Le tout conformément à ce qu'il croit (sait).

4. *Adaptation et apprentissage* : les agents améliorent leur performance à mesure que le temps passe

## 2.9 Différentes catégories et modèles d'agents

Après avoir défini les agents et les systèmes multi-agents, nous présentons dans cette partie les différents modèles d'agents, afin de comprendre leurs caractéristiques et leurs modes de fonctionnement. Nous distinguons deux grandes familles d'agents les agents réactifs et agents cognitifs.

### 2.9.1. Agents cognitifs

C'est le premier modèle d'agents qui a été proposé. Il est nommé aussi agent délibératif. Il est basé sur l'IA symbolique, et il permet de planifier les actions d'un agent au sein de son environnement [Fikes et Nilsson, 1971]. En effet, les agents cognitifs sont capables à eux seuls de réaliser des opérations relativement complexes. Généralement, ils coopèrent les uns avec les autres pour atteindre un but commun (résolution d'un problème, une tâche complexe, etc.). Ils possèdent un ensemble de représentations explicites (sur l'environnement, sur les autres agents et sur eux mêmes) décrits dans une base de connaissances sur laquelle ils peuvent raisonner. Ils réagissent en fonction de leurs connaissances, leurs buts, de leurs échanges d'informations avec les autres agents et de la perception de l'environnement (voir figure 6). Ils sont dotés de moyens et mécanismes de communication pour gérer les interactions avec d'autres agents (coopération, coordination et négociation).

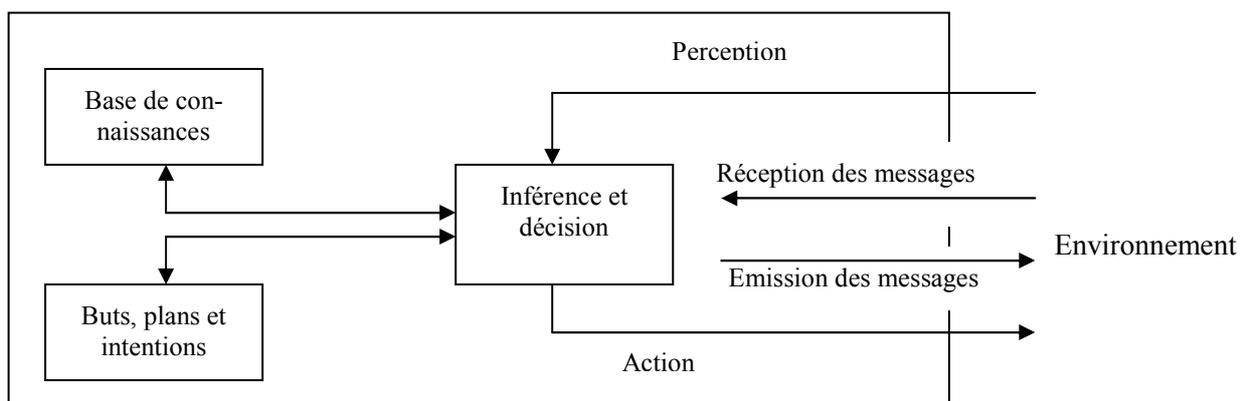


Figure 6 : Modèle d'un agent cognitif

Ce modèle d'agent est une métaphore du modèle humain et s'appuie sur la sociologie des organisations [Liu, 2002]. Nous pouvons aussi trouver son origine dans la volonté de faire coopérer des systèmes experts classiques dans le domaine d'IA.

L'agent cognitif, en terme de BDI pour Beliefs (croyances), Désires (désirs) et Intention (intentions), [Bratman, 1987 ; Bratman et al., 1988], postule que ses actions, tout comme celles d'un être humain, doivent être dictées par des représentations abstraites du monde, incluant la représentation de ses propres capacités, ses buts ainsi que ceux des autres agents. Ce modèle est fondé sur des extensions de la logique, et est basé sur les états mentaux suivants :

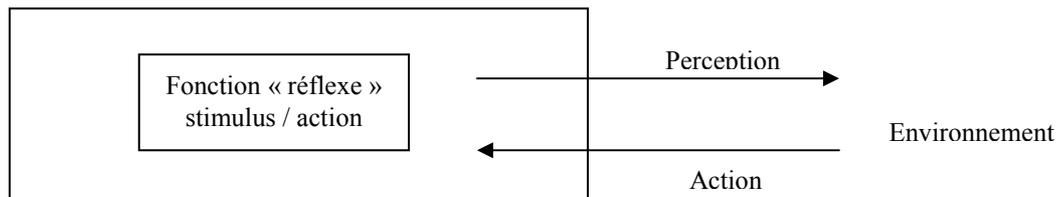
- ◆ les croyances : les connaissances de l'agent sur l'environnement,
- ◆ les désirs : les « options » de l'agent, c'est à dire les différents objectifs vers lesquels l'agent peut vouloir transiter,
- ◆ les intentions : les états vers lesquels l'agent a choisi de transiter.

Le fonctionnement d'un agent BDI consiste donc à actualiser ses croyances à partir des informations qu'il reçoit de son environnement, ensuite, à déterminer les différents choix qu'il a en face, les analyser et fixer ses états buts en fonction d'eux (intentions), et enfin, à définir ses actions en tenant en compte de ses intentions.

L'agent cognitif traite généralement des informations qualitatives tout en utilisant un raisonnement qualitatif ou symbolique. Ces traitements peuvent être établis par l'intermédiaire des outils comme les Classifieurs Génétiques/Neuronaux (CG/ CN) ou des Systèmes d'Expert (SE). Ces agents peuvent utiliser des mécanismes comme les systèmes à base de raisonnement par cas « Case Based Reasoning » (CBR), les systèmes à base de connaissances « Knowledge Base System » (KBS), les théories des jeux « Game Theory » (GT) ou à base de la logique utilisée dans les agent BDI, voir (C) et (D) dans la figure 8.

### **2.9.2. Agents réactifs**

Parmi les critiques du raisonnement symbolique figure Brooks, qui, par le biais de plusieurs papiers [Brooks, 1986 ; 1991], manifesta son opposition au modèle symbolique et proposa une approche alternative appelée aujourd'hui IA active. Selon lui, le comportement intelligent devrait émerger de l'interaction entre divers comportements plus simples.



**Figure 7 : Modèle d'un agent réactif**

Dans ce même contexte, nous considérons que les agents réactifs n'ont de connaissance explicite, ni de l'environnement, ni des autres agents, ni de leur passé, pas plus que de leurs buts (pas de planification de leurs actions). Ce sont des agents qui réagissent uniquement à leur perception de l'environnement et qui agissent en fonction de cette perception (voir figure 7).

Ce modèle d'agent est une métaphore du modèle « fourmi », il s'appuie sur les sciences de la vie et l'intelligence collective [Bonabeau et Theraulaz, 1994 ; Bonabeau et al. 1999]

Dans la même catégorie que les agents réactifs, mais avec davantage de rationalité, se trouvent les agents hédoniques. Ces agents apprennent, par auto-renforcement, à modifier leur comportement afin d'augmenter leur " plaisir ou satisfaction ". Ils sont capables d'anticipations " hédoniques" et d'adaptation lente à partir de leur expérience historique, ce qui suppose un niveau de rationalité plus élevé que l'agent purement réactif.

Les agents réactifs, traitent généralement des informations quantitatives tout en utilisant des calculs élémentaires ou d'optimisation. Ils peuvent être construits par des réseaux connexionnistes comme les Réseaux de Neurones (RN) ou en utilisant des simples algorithmes de calcul comme les Algorithmes Génétiques (AG). Leurs capacités répondent à la loi stimulus/action et apprentissage par auto-renforcement, voir (A) et (B) dans la figure 8.

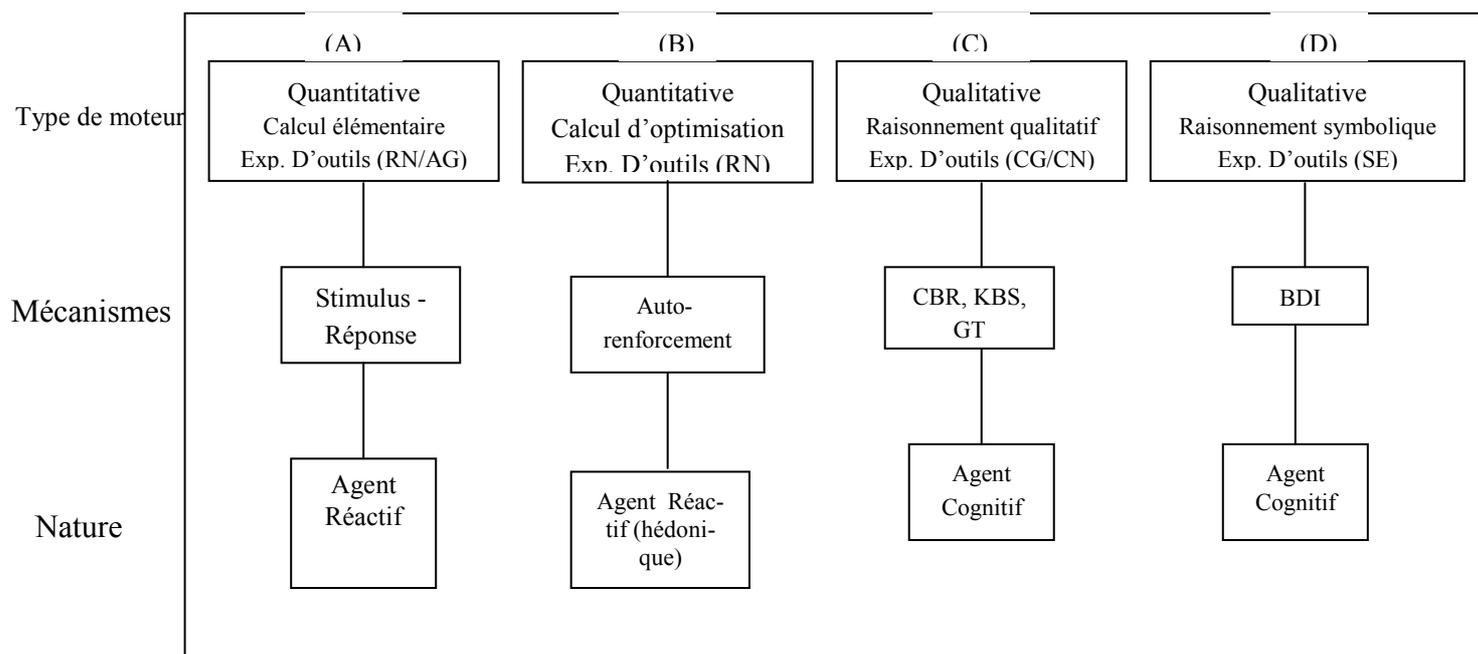


Figure 8: Structures des modèles d'agents

Les agents réactifs sont les plus simples à implémenter, et leur efficacité a été vérifiée dans plusieurs applications. Ils présentent toutefois des limites dues aux points suivants :

- ◆ l'agent n'a pas la représentation mentale de l'environnement, et doit choisir ses bonnes actions à partir des données locales uniquement,
- ◆ le comportement global de l'agent ne peut pas être facilement prévu, par conséquent, il n'est pas toujours possible de concevoir un comportement d'agent en fonction du but spécifié.

Contrairement aux agents réactifs, les agents cognitifs sont beaucoup plus complexes et plus difficiles à mettre en œuvre. Chaque agent se fonde sur ses propres compétences de façon isolée pour résoudre un problème, ce sont les principales limites de cette architecture.

Pour faire face aux inconvénients de ces deux modèles, les chercheurs ont combiné ces deux facettes, opposées mais complémentaires, de la conception des agents. Cette combinaison fait apparaître les agents hybrides.

### 2.9.3. Agents hybrides

Les agents hybrides sont conçus pour combiner des capacités réactives à des capacités cognitives, ce qui leur permet d'adapter leur comportement en temps réel à l'évolution de l'environnement [Ferguson, 1992 ; Fischer, 1999]. Dans le modèle hybride, un agent est composé de plusieurs couches, rangées

selon une hiérarchie. La plupart des architectures considèrent que trois couches suffisent amplement.

Ainsi, au plus bas niveau de l'architecture, on retrouve habituellement une couche purement réactive, qui prend ses décisions en se basant sur des données brutes en provenance des senseurs. La couche intermédiaire fait abstraction des données brutes et travaille plutôt avec une vision des connaissances de l'environnement. Finalement, la couche supérieure se charge des aspects sociaux de l'environnement (communication, coopération, négociation), c'est-à-dire du raisonnement tenant compte des autres agents. Le modèle InteRRap proposé par Müller et Pischel, voir figure 9 [Müller et Pischel, 1994] figure parmi les exemples du modèle hybride. Dans ce modèle, chaque couche comporte une base de connaissance en fonction de son degré d'abstraction : modèle de l'environnement au niveau de la couche réactive, modèle mental au niveau de la couche de planification locale et modèle social pour la couche de coopération et de planification globale.

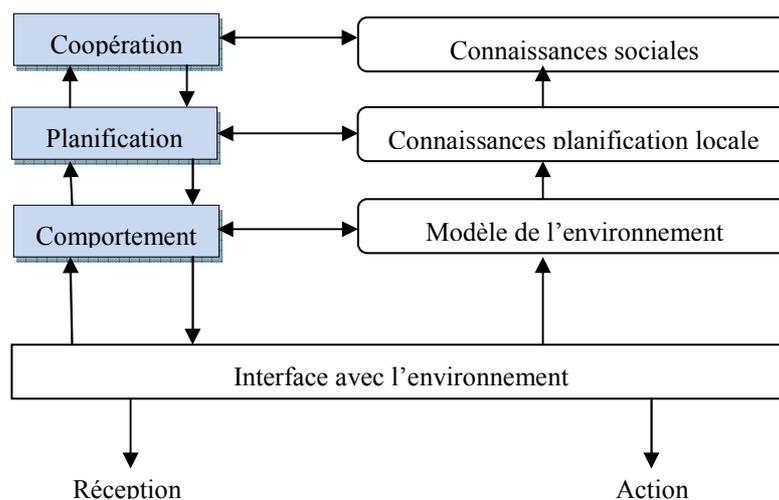


Figure 9 : Modèle d'agents InteRRap

## 2.10 Application SMA :

Il existe une vaste gamme d'application des agents et systèmes multi agents. Nous citons ici quelques exemples:

- Application industrielles : contrôle en temps réel, production, réseaux, de télécommunications, systèmes de transport, systèmes de distribution, etc....
- gestion de processus de business, support à la décision.

- l'apprentissage et le commerce et l'interaction homme machine
- systèmes d'information coopératifs : découvert des sources, recherche de l'information, fusion des informations et personnalisation
- Mondes virtuels les jeux vidéo, où, par exemple, des agents faisant partie du jeu jouent contre l'être humain.

## 2.11 La communication entre les agents :

Après la construction de l'architecture interne des agents, l'une des préoccupations les plus cruciales dans la construction d'un SMA est la communication entre les agents. Cette communication est nécessaire aussi bien pour échanger les données entre les agents que les connaissances. Pour cela, plusieurs moyens existent : au niveau le plus bas, il existe des *sockets* qui permettent aux différents agents codés en Java de communiquer entre eux ; il existe aussi d'autres technologies (en Java notamment). On peut citer l'invocation de méthodes distantes (RMI pour *Remote Method Invocation*) et la technologie CORBA (*Common Request Broker Architecture*).

Plusieurs travaux concernent la mise en œuvre d'infrastructures de communication qui dispensent le programmeur d'implémenter lui-même la communication entre les agents. Avec ce type d'outils, le fonctionnement typique consiste à fournir une adresse à chaque agent. Cette adresse lui permettra d'émettre et de recevoir des messages de la part des autres agents. L'avantage est la possibilité d'avoir des communications asynchrones et d'être dispensé de la nécessité de localiser un agent avant de lui envoyer un message. JATLite est une infrastructure de ce type, elle permet aux agents de communiquer via Internet.

## 2.12 Environnement de développement :

De nombreux environnements de développements de SMA ont été proposés par des équipes universitaires. Citons en particulier Mice par Ed Durfee de l'université du Michigan [Durfee & Montgomery 1989], Mages [Bouron et al. 1990], Ratman [Bürckert & Müller 1991] et plus récemment Desire [Brazier et al. Fv].

Mais ces environnements ne peuvent combler les contraintes industrielles, voire même pré-industrielles. C'est pour cela qu'un projet Esprit, Imagine, avait démarré en 1990, avec comme partenaire Siemens, Plessey, Intrasoft, et Steria pour la France. Il s'agissait d'offrir aux concepteurs et réalisateurs un

environnement de développement pour systèmes multi agents à granularité variable. Dans ce contexte, un agent peut être un système à base de connaissance, un capteur, un programme de supervision, voire même l'assistant informatisé d'un être humain. Deux applications avaient été envisagées. L'une concerne le trafic aérien, l'autre un système de gestion de réseau de télécommunication. Si des "maquettes" du système ont bien été réalisées en Parlog\* avec souvent de très bonnes idées, le projet global n'a pas abouti.

Plusieurs systèmes industriels se présentent maintenant comme une alternative à ces premiers projets universitaires. En particulier, AgentBuilder réalisé par IBM, environnement de réalisation d'agents "intelligents" à partir d'un ensemble de composants connectables (puggable). Chaque agent se compose d'un moteur d'inférence (dont les faits et règles sont écrits en format KIF), et d'un système de réaction à des événements http .Dans le domaine des télécoms, la société General Magic, en collaboration avec de nombreuses sociétés de télécommunications, et en particulier France Télécom, a développé un système de développement de systèmes multi agents fonctionnant sur réseau, Télé script\*\*. Bien que dans ce système les notions d'agents soient particulièrement frustes (les agents ne sont ici que de simples processus) il intègre néanmoins un certain nombre de caractéristiques qui permettent d'ores et déjà d'écrire des (petites) applications distribuées disposant d'un minimum de capacités multi agents. Gageons que les systèmes du futur s'inspireront des facilités de développement et de la technologie – très orientée objet – mise en œuvre dans ce produit.

Enfin, le langage Java de Sun commence à prendre un essor considérable et certains le voient comme un challenger particulièrement prometteur de Télé script. Bien qu'il dispose de moins de richesses quant à la migration d'objets et à la sécurité de fonctionnement, plusieurs projets commencent à promouvoir Java comme langage de développement d'agents communicants sur Internet.

## **2.13 Conclusion**

Les systèmes multi-agents sont à l'intersection de plusieurs domaines scientifiques tels que : l'information répartie, le génie logiciel, l'intelligence collective, l'intelligence artificielle et la vie artificielle [Chaib-draa, 2001]. Par conséquent, ils font appel à plusieurs autres disciplines, telles que la sociologie, la psychologie sociale, les sciences cognitives, la biologie, etc

Après une revue générale des SMA, nous nous intéressons maintenant à la partie essentielle dans les SMA, les interactions entre les différents agents dans le système. Généralement, ces interactions entre agents sont effectuées à travers des protocoles de négociation afin de satisfaire les intérêts locaux des

agents et ceux globaux du système. Dans le prochain chapitre nous détaillons les différents protocoles de négociation existant dans la littérature des SMA

**LA NEGOCIATION  
AUTOMATISEE DANS LES  
SYSTEMES MULTI-AGENTS  
ET LES ENCHERES**

### 3.1 Généralités

Comme les agents s'attachent à représenter de la manière la plus fidèle les intérêts de leur utilisateur, ils peuvent avoir des buts et des motivations différents. Pour pouvoir interagir avec succès, ils doivent en plus avoir des capacités pour coopérer, coordonner et négocier entre eux.

Dans les SMA, la négociation est le processus par lequel les agents :

- ◆ coordonnent leurs actions ;
- ◆ se partagent des ressources limitées ;
- ◆ font valoir leurs points de vue différents.

Afin de trouver une situation qui satisfasse au mieux leurs intérêts respectifs. Il faut noter que la négociation est une approche possible, d'autres alternatives pourraient être de décider de manière autoritaire (mais on n'explique pas les conflits et on peut altérer les intérêts des agents) ou de faire voter les agents.

### 3.2 Problématique dans les SMA

Dans la conception des système multi-agents, on distingue classiquement deux problèmes sous-jacents :

- ◆ le fonctionnement de l'agent : concevoir le fonctionnement interne et les capacités de l'agent, dans le but de réaliser correctement les tâches assignées ;
- ◆ le fonctionnement de la société d'agents, c'est-à-dire, concevoir les interactions entre les agents, dans le but de coopérer, coordonner et négocier.

De plus, les questions principales qui y sont rattachés sont :

- ◆ comment impulser une situation de coopération entre agents égo-centrés ?
- ◆ comment coordonner les actions des agents pour qu'ils atteignent leurs buts en coopérant les uns avec les autres ?
- ◆ comment les agents peuvent-ils reconnaître les situations de conflit et comment les résoudre ?

### 3.3 Composantes du modèle de négociation dans les SMA

Pour modéliser la négociation dans un SMA, on prend généralement en compte les aspects suivants :

- ◆ le langage de négociation (fonctionnement de la société d'agents) : composé d'un ensemble de primitives de communication, il précise la manière dont les agents communiquent ;
- ◆ le protocole de négociation (fonctionnement de la société d'agents) : c'est l'ensemble des règles qui régit la négociation (e.g propositions légales, états de la négociation) ;
- ◆ les objets de négociation : c'est sur ces objets que se porte la négociation (e.g prix, délai, temps de réponse) ;
- ◆ le processus de décision (fonctionnement de l'agent) : c'est lui qui oriente la stratégie de l'agent au cours de la négociation. C'est le coeur de fonctionnement et l'intelligence de la négociation. La stratégie peut être entièrement définie et immuable, ou alors mutable en fonction des stratégies des autres (observation, supposition).

La cardinalité des participants de la négociation peut aussi varier : négociation *un-à-un*, négociation *un-à-plusieurs*, négociation *plusieurs-à-plusieurs*.

### 3.4 Quelques modèles de négociation

#### 3.4.1. Le take it or leave it offer

Cette forme de négociation est très primaire, puisqu'elle consiste à formuler une proposition qui est à prendre ou à laisser par le ou les participants. Cette négociation se déroule sur un seul tour, sans contre-proposition ni renégociation. C'est celle que l'on rencontre tous les jours pour acheter son pain, par exemple. Le protocole est donc très simple : le vendeur propose sa ressource (bien, service, etc.) à un prix ferme à un acheteur qui soit accepte, soit refuse. S'il accepte, l'acheteur paie le prix pour obtenir la ressource. Le take it or leave it offer n'est pas, pour nous, de la négociation.

#### 3.4.2. Le vote

Ce modèle s'inspire largement du modèle de résolution des conflits humains. Transposé aux SMA, chaque agent attribue une valeur d'utilité aux pro-

chains états du système et l'état qui concentre le suffrage le plus élevé est le nouvel état du système.

Toutefois, le modèle de coopération basé sur le vote comporte des différences avec un modèle de négociation classique. La principale différence est que dans une négociation, l'état final n'est pas nécessairement un des états possibles au début du processus de négociation.

### **3.4.3. Négociation dans la théorie des jeux**

La théorie des jeux définit un modèle d'interactions très général : plusieurs agents interagissent selon des règles, ils ont des centres d'intérêts différents et des capacités de décision différentes. Cette théorie est parfois utilisée dans l'étude des problèmes politiques, militaires et économiques.

Dans la théorie des jeux, le but d'un agent est de faire le meilleur choix compte tenu des règles d'interactions.

L'utilité est le seul paramètre considéré par les agents et on suppose que les agents peuvent sélectionner la meilleure stratégie en considérant toutes les stratégies possibles.

Dans ce cas, la négociation consiste à caractériser les préférences de l'agent par rapport à tous les résultats possibles, et par conséquent l'état optimal est nécessairement atteint.

La théorie de jeux représente une discipline à part entière et demanderait des travaux plus approfondis pour évaluer complètement les modèles de négociation qui s'y rapportent.

Ce modèle de négociation présente un inconvénient majeur pour son implémentation : le calcul de la décision peut être élevé en raison des complexités calculatoires (la prise de décision doit être rapide). De plus, le domaine dans lequel évoluent ces agents est un domaine non coopératif chaque agent prend sa décision de manière à optimiser sa situation sans coordination avec les autres, parfois même à leur détriment.

### **3.4.4. Contract Net Protocol**

Le Contract Net Protocol (CNP) [Smi88, Ali03] est un protocole général qui se focalise principalement sur la coordination des agents coopératifs, ayant les mêmes buts et qui partagent et résolvent un problème plus complexe. C'est pour cette raison que le CNP est aussi vu comme un protocole d'allocation de tâches par réseau contractuel.

Ce protocole établit en fait un procédé organisationnel : les contrats permettent de répartir et de coordonner les activités de chaque agent pour résoudre le problème global. Dans CNP, on distingue le gestionnaire des contractants. Le gestionnaire décompose le problème global en sous-tâches et initie la négociation en annonçant chaque sous-tâche sur le réseau. Les agents qui ont les capacités (ressources appropriées, expertise) pour réaliser la sous-tâche participent à la négociation en envoyant au gestionnaire leurs propositions. Dès lors, le gestionnaire rassemble les propositions reçues et alloue la tâche au meilleur.

Notons, que le modèle peut être récursif : à son tour, l'agent élu peut redécomposer en sous-tâche. D'un point de vue de l'allocation des tâches, le CNP lie un gestionnaire à des contractants. Si on s'intéresse au rôles dans la négociation alors le gestionnaire est l'initiateur de la négociation et les contractants sont les participants.

La **figure 10** présente, de manière schématique, le fonctionnement du CNP standardisée par la FIPA.

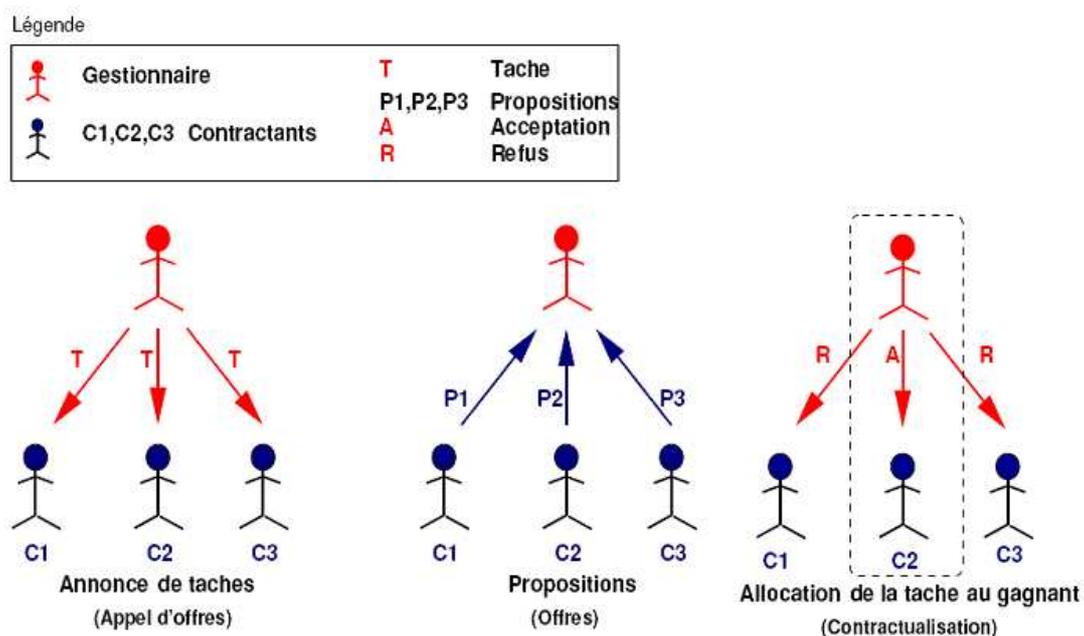


Figure 10 : Description schématique du Contract Net Protocol

Dans ce protocole, la négociation se résume à l'allocation de tâches en fonction de la qualification des agents. Chaque agent a pour but de réaliser au mieux ce qu'on lui propose de faire et n'a pas d'autres motivations.

Depuis sa version initiale et compte tenu de son caractère assez général, le CNP a connu de nombreuses extensions. On citera entre autres les compléments apportés au protocole (tours successifs, confirmation et refus de proposition) par la FIPA 1 [Com02] ainsi qu'une implémentation de CNP réalisée par T. Sandholm, qui est enrichie par un mécanisme de prix (les agents calculent localement leurs coûts marginaux pour réaliser des ensembles de tâches et le choix du contractant se base sur ces coûts) [San93]. Les **figures 11** et 12 présentent les actes de langages définis dans les versions du CNP standard et du CNP étendu

FIPA Contract Net Interaction Protocol

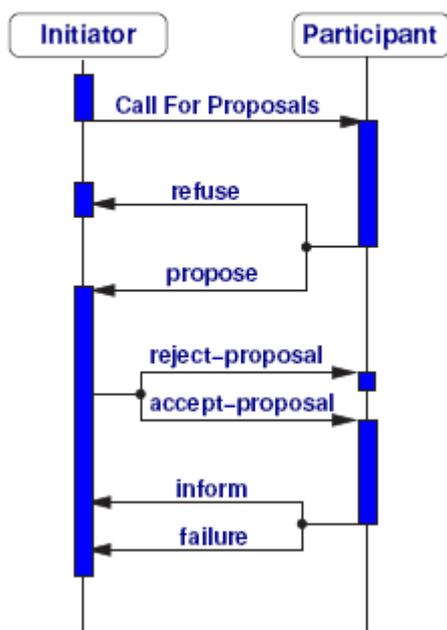


Figure 12 : Actes de langages dans le Contract Net Protocol

FIPA Iterated Contract Net Protocol

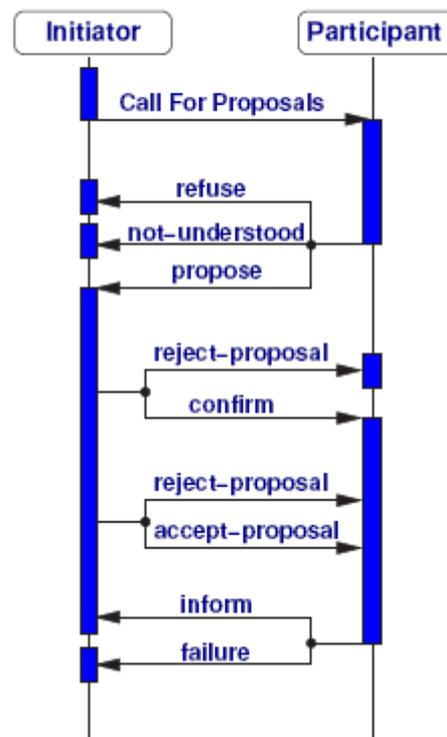


Figure 11 : Actes de langages dans le Contract Net Protocol étendu

### 3.4.5. Modèle générique de négociation Genca

Le projet de recherche Genca de l'équipe SMAC du LIFL a développé une API permettant la négociation entre agents. Ce modèle de négociation [PM02] définit trois niveaux :

1. la représentation interne de la négociation qui définit entre autres les actes de langages l
2. la communication : l'envoi des messages entre agents ;
3. la stratégie : le raisonnement des agents.

Il propose aussi une formalisation de la négociation ainsi que de l'instanciation du contrat.

Le protocole de négociation de Genca se base essentiellement sur le CNP étendu mais il définit en plus les notions de clauses minimales suffisantes pour valider un contrat et de nombre maximal de renégociation au-delà duquel le contrat est invalidé. La stratégie qui permet aux participants de générer des modifications se base sur une fonction qui est définie par l'utilisateur et qui dépend de l'application cible.

Quelques exemples ont été développés pour mettre en application le modèle de négociation de Genca. Ainsi, dans le cas des enchères, la stratégie de proposition peut se faire selon des fonctions linéaires ou quadratiques. Dans le cas de la prise de rendez-vous, une stratégie possible est de considérer une fonction d'évaluation qui attribue une note à chaque proposition (ici un créneau horaire) en prenant en compte les priorités par rapport aux participants et les créneaux préférentiels des participants.

Les caractéristiques du modèle mises en avant sont :

- la renégociation automatique : elle se produit lorsqu'un agent se rétracte d'un contrat qu'il a signé ;
- la gestion des négociations simultanées : il peut y avoir des négociations séquentielles ou en parallèle qui portent sur des ensembles de ressources communes ou disjointes (cardinalité  $N ; M$ ) ;
- la gestion de deadlocks (qui peut se produire quand 2 deux agents négocient sur des mêmes ressources et attendent la réponse de l'autre) : en définissant une réponse par défaut qui est prise en compte si la réponse attendue ne parvient pas au bout d'une durée bornée ;
- la paramétrisation XML : on y précise les propriétés de la négociation comme les participants, les ressources, le délai de réponse et la réponse

par défaut, le nombre de renégociations, le nombre de clauses minimales pour valider le contrat,... ;

- la définition de listes de priorités par rapport aux participants et aux ressources.

Genca est un modèle de négociation générique qui apporte de nouvelles caractéristiques qui n'existent pas dans d'autres modèles. Plusieurs exemples d'application sont donnés (enchère, prise de rendez-vous, choix d'un restaurant) pour montrer l'applicabilité de ces modèles. Néanmoins, l'intelligence de la négociation se fait toujours par le biais de fonctions linéaire ou additives qui prennent en compte des préférences.

### 3.4.6. Négociation à base de connaissances

Dans ce type de négociation, il existe un agent omnipotent qui connaît les connaissances partagées par chaque agent. Cette connaissance partielle du système global lui permet de résoudre les conflits.

### 3.4.7. Les négociations multi-attributs

Les négociations multi-attributs, comme leur nom l'indique, sont des négociations qui impliquent différents attributs devant être négociés. Elles sont directement opposées aux enchères qui n'impliquent qu'un seul attribut : un prix. Cette forme de négociation est cependant très répandue et à la base de nombreuses variantes de négociation. Un exemple de négociation multi-attributs est la négociation d'une voiture chez un concessionnaire. Le modèle, la motorisation, la couleur et de nombreuses options comme la climatisation, la direction assistée ou encore la présence d'airbags, en plus du prix, seront négociés.

Une fonction d'utilité est utilisée pour comparer les différentes offres. Cette fonction permet de pondérer l'importance de chaque attribut et chaque attribut est associé à une autre fonction d'utilité déterminant la satisfaction pour cet attribut. Si l'on reprend l'exemple de l'achat d'une voiture, on peut utiliser la même fonction d'utilité pour les options qui vaut 1 si l'option est présente, 0 sinon. De manière générale, la fonction d'utilité d'une offre  $o_j$  est de la forme :

$$U(o_j) = \sum_{i=1}^n w_i u_i(o_j^i)$$

où la négociation comprend  $n$  attributs,  $w_i$  le poids du  $i^{\text{ème}}$  attribut,  $U_i$  la fonction d'utilité du  $i^{\text{ème}}$  attribut et  $o_j^i$  la valeur du  $i^{\text{ème}}$  attribut dans l'offre  $o_j$ . Le but est de maximiser la valeur de cette fonction d'utilité. Dans [Bichler et

al., 1999], Bichler et ses collègues proposent d'utiliser les enchères pour les négociations multi-attributs en utilisant la fonction d'utilité comme seul attribut d'évaluation de l'enchère. Dans [Jonker and Treur, 2001], Jonker et Treur proposent une architecture d'agent pour la négociation multi-attributs et décrivent le modèle de négociation qu'ils proposent et l'illustrent avec l'exemple de l'achat d'une voiture.

### **3.4.8. Les négociations combinées**

Les négociations combinées sont utilisées lorsqu'une personne a besoin d'un ensemble d'objets non disponibles auprès d'un unique vendeur. Il faut alors négocier chaque objet (ou sous-ensemble d'objets) séparément et avoir un mécanisme de liaison entre les négociations, car si l'ensemble des objets ne peut être acquis, aucun objet ne doit l'être. De plus, il ne faut obtenir chaque item qu'en un seul exemplaire. Les différentes négociations sont indépendantes les unes des autres, alors que l'ensemble des objets négociés sont typiquement interdépendants. Les négociations peuvent être de type différent pour chaque objet ou sous-ensemble d'objets. On peut donc rencontrer des enchères anglaises, des négociations de type take it or leave it, ou autres. Un exemple type de négociation combinée est la composition d'un voyage. Pour obtenir un voyage, il faut un moyen de transport (par exemple, un vol aller-retour vers la destination choisie) et les nuits d'hôtel pour la période du voyage. Le vol aller-retour se négocie avec une compagnie aérienne, tandis que les nuits d'hôtel se négocient avec une compagnie hôtelière. Cet exemple est celui retenu pour la Trading Agent Competition (TAC), une compétition où différents agents sont chargés de composer des voyages pour 8 clients ayant des préférences sur les dates de départ et d'arrivée, ainsi que sur le type d'hôtel où ils vont résider. Cette compétition, qui a lieu tous les ans depuis trois ans, utilise AuctionBot comme support logiciel pour les différentes enchères.

## **3.5 Négociation dans la vente aux enchères**

Les enchères furent utilisées depuis des milliers d'années. Le mot lui-même provient du latin *augere* qui signifie : augmenter. La pratique des enchères était fréquente dans certaines civilisations anciennes : chinoise, babylonienne, romaine, etc. Une des plus célèbres enchères de l'histoire antique s'est déroulée à Rome en l'an 193 avant notre ère. La garde prétorienne ayant assassiné l'empereur romain de l'époque et mis la couronne de l'empire aux enchères publiques. Quand Didius Julianus proposa la meilleure offre consistant en une donation de 6250 drachmas (monnaie romaine) pour chacun des gardes, ces

derniers le déclarèrent empereur. (Voir [Cassady, 1967] pour une introduction à l'histoire des enchères).

Depuis ces temps, les enchères ont largement évolué et le recours à leurs mécanismes n'a fait que se populariser. Il existe actuellement différents formats d'enchères pouvant être classifiés selon diverses caractéristiques, notamment : enchères ouvertes versus à enveloppes scellées et enchères ascendantes versus descendantes [Agorics Inc., 2001]. Depuis l'œuvre fondamentale de William Vickrey sur la théorie des enchères [Vickrey, 1961], les experts s'accordent sur le fait qu'il existe quatre types d'enchères de base : l'anglaise, la hollandaise, la premier prix à enveloppes scellées et la Vickrey.

### **3.5.1. Les enchères de base**

#### ***3.5.1.1. L'enchère anglaise***

C'est la forme la plus commune. Il s'agit d'une enchère ouverte ou publique à prix ascendant. Paul Milgrom [Milgrom, 1989], en abordant l'enchère anglaise, la définit de la manière suivante :

Ici l'encanteur commence en annonçant le prix le plus bas acceptable (prix de réserve), et procède en recevant successivement des offres de plus en plus élevées de la part des participants, jusqu'à ce qu'il n'y ai plus d'enchérisseur. L'objet est alors attribué (vendu) au meilleur offrant.

Dans certains cas toutefois, l'encanteur pourrait garder secret le prix de réserve. Les enchères commencent alors à partir de zéro. La raison de procéder de la sorte est essentiellement d'éviter la formation de coalitions entre certain enchérisseurs qui pourraient décider de ne pas entrer en compétitions les uns contre les autres afin de maintenir le prix gagnant au seuil le plus bas possible. Supposons par exemple que le prix de réserve d'un objet vendu aux enchères est de 50\$. Si les acheteurs sont au courant de ce prix de réserve, certains d'entre eux pourraient se mettre d'accord pour ne pas enchérir les uns contre les autres une fois ce prix de réserve atteint. Ils espèrent de la sorte maintenir le prix le plus proche possible des 50\$.

Par contre, s'ils ne connaissaient pas ce prix de réserve, il est plus difficile de former des coalitions. Parce que s'ils décident d'arrêter d'enchérir à un moment donné, ils ne sont pas sûrs d'avoir dépassé le prix de réserve, et prennent donc le risque de perdre l'enchère. Les acheteurs n'ont pas intérêt d'arrêter d'enchérir.

L'enchère anglaise encourage fortement la compétition. Il n'est pas rare dans ce cas de voir des participants inexpérimentés épris d'enthousiasme faire

des offres dépassant la valeur réelle de l'objet, le mécanisme de la vente concurrentielle les y exaltant. Les spécialistes utilisent un terme bien évocateur pour désigner ce phénomène : la malédiction du gagnant.

Il existe par ailleurs une autre variante de l'enchère anglaise où le prix est haussé progressivement par l'encanteur, les participants quittent l'enchère au fur et à mesure que le prix atteint devient trop haut pour eux. Le dernier restant est le gagnant.

#### **3.5.1.2. *L'enchère hollandaise***

Il s'agit d'une enchère ouverte ou publique à prix descendant. L'encanteur annonce un prix élevé qu'il diminue progressivement jusqu'à ce qu'un participant réclame l'objet à vendre au prix atteint.

L'intérêt de ce type d'enchères par rapport à l'anglaise, du point de vue du vendeur du moins, est qu'elle est supposée lui procurer plus de profits. En effet, l'acheteur avec la plus haute estimation du prix de l'objet à vendre, aura plus tendance à le réclamer dès que le prix atteint est égal ou inférieur à celui qu'il est capable d'offrir (dans le cas où il n'a pas d'informations sur les estimations des autres acheteurs). En revanche, dans une enchère anglaise, l'acheteur ayant la plus grande estimation du prix de l'objet, pourrait gagner en haussant le prix offert petit à petit. Il pourrait ainsi obtenir l'objet pour un prix très inférieur au prix maximum qu'il aurait pu atteindre.

#### **3.5.1.3. *L'enchère premier prix à enveloppes scellées***

Comme son nom l'indique, il s'agit d'une enchère à enveloppes scellées, signifiant que les offres des participants sont cachées les uns aux autres. Le gagnant de l'enchère, celui ayant fait la meilleure offre, doit payer le montant exact de son offre.

L'enchère se déroule en deux étapes : la soumissions des offres, suivie de la détermination du gagnant après examen des ces offres. Communément, sauf exceptions, l'enchère se déroule en un seul tour, les participants ne peuvent proposer qu'une seule offre, d'où l'importance de la phase de préparation de la proposition à soumettre.

#### **3.5.1.4. *L'enchère Vickrey***

A l'instar de l'enchère premier prix à enveloppes scellées, dans la Vickrey, les offres sont cachées aux différents participants. La nuance réside dans le fait que le gagnant ayant soumis la meilleure offre doit payer non pas le montant de son offre, mais le deuxième meilleur prix proposé pour l'objet à vendre au cours de l'enchère.

Vickrey [Vickrey, 1961], qui, le premier, proposa cette manière de faire, démontra que la stratégie dominante pour un acheteur est de soumettre sa vraie évaluation de l'objet.

### 3.5.1.5. Récapitulatif

Enchère	Règles	Avantages	Inconvénients
Enchère anglaise	Le vendeur annonce un prix d'ouverture. Les offres augmentent progressivement jusqu'à ce qu'il n'y ait plus d'offres. Le gagnant est le dernier offrant. Il doit payer le montant de son offre.	Simplicité Incite à la concurrence entre les acheteurs	Risque de formation de coalitions Les participants doivent suivre l'enchère Surévaluation du prix de l'objet
Enchère hollandaise	Le vendeur annonce une offre d'ouverture assez élevée. Il diminue progressivement cette offre jusqu'à ce qu'un des participants accepte d'acheter le produit au prix annoncé.	Simplicité Prix supérieur au prix du marché	Moins de compétition Risque de formation de coalitions Les participants doivent suivre l'enchère
Enchère premier prix à enveloppes scellées	Les offres sont secrètes. Elles sont soumises une seule fois au vendeur. Le gagnant doit payer le montant exact de son offre.	Simplicité Participants non présents lors de la vente	Moins de compétition Les offres des participants sont inférieures à leurs évaluations
Enchère Vickrey	Les offres sont secrètes. Elles sont soumises une seule fois au vendeur. Le gagnant doit payer le montant de la deuxième meilleure offre.	Simplicité Participants non présents lors de la vente Les participants offrent leurs vraies évaluations	Moins de compétition

### 3.5.2. Enchères multi-objets

Dans la section précédente nous avons introduit certaines notions fondamentales de la théorie des enchères. Ces concepts concernent essentiellement la vente et achat d'un objet unique. Nous présentons ici les idées sous-jacentes aux mécanismes d'enchères pour la vente et achat de plusieurs objets identiques et indépendants les uns des autres. Nous introduirons aussi le fameux exemple de vente des spectres de fréquences radio par la FCC, où les objets vendus sont complémentaires.

On distingue deux manières de faire une enchères multi-objets (voir à ce propos [Weber, 1983, Vickrey, 1961]). Soit une enchère séquentielle, où les objets sont présentés un à un. C'est le cas de la vente de collections d'objets comme des timbres, antiquités, etc. Soit une enchère simultanée où les objets sont présentés à la vente en même temps et les enchères s'effectuant alors sur tout les objets simultanément. C'est le cas de la vente des fréquences radio par la FCC. Nous nous intéressons exclusivement dans ce travail au cas d'enchères simultanées.

Dans ce qui suit nous présentons quelques exemples simples d'enchères multi-objets.

#### ***3.5.2.1. Enchère anglaise multi-objets***

Le mécanisme est similaire à celui de l'enchère anglaise simple. Dans ce cas, à la fin de l'enchère, les acheteurs ayant soumis les meilleures offres sont ceux qui obtiennent les quantités qu'ils désirent jusqu'à épuisement des objets soumis à la vente. Ils payent chacun le montant exact de leurs offres. On parle alors d'une enchère discriminatoire. Ausubel [Ausubel, 2000] présente une discussion exhaustive à propos de la généralisation de l'enchère anglaise aux cas multi-objets.

#### ***3.5.2.2. Enchère hollandaise multi-objets***

Il s'agit du même mécanisme que celui de l'enchère hollandaise pour la vente d'un seul objet. Sauf qu'ici, l'enchère ne s'arrête que si tout les objets sont vendus. Deux cas peuvent se présenter : le cas uniforme, où tout les gagnants payent le montant exact de l'offre du dernier gagnant, ou le cas discriminatoire où tout les gagnants payent chacun le montant de sa propre offre.

#### ***3.5.2.3. Enchère premier prix à enveloppes scellées multi-objets***

Il s'agit d'une enchère discriminatoire. En effet, les offres soumises au vendeur sont classées par ordre décroissant. Les objets sont alors alloués aux acheteurs ayant les plus hautes offres jusqu'à épuisement de la quantité d'objets à vendre. Les gagnants payent le montant de leurs offres respectives, d'où le caractère discriminatoire de l'enchère.

#### ***3.5.2.4. Enchère Vickrey multi-objets***

La généralisation de l'enchère Vickrey au cas multi-objets a déjà été présentée par Vickrey au début des années soixante [Vickrey, 1961]. Il s'agissait de la vente de  $k$  objets à  $n$  acheteurs, chaque acheteur désirant acquérir un seul objet. Dans ce cas, les  $k$  acheteurs ayant soumis les plus hautes offres obtiennent chacun un objet et ils payeront le montant de la  $(k+1)^{\text{ime}}$  plus haute offre. Les gagnants payent chacun donc le même prix, on parle d'une enchère uni-

forme. Il existe par ailleurs plusieurs autres façons de généraliser l'enchère Vickrey comme par exemple dans les travaux de Varian et MacKie-Mason [Varian and MacKie-Mason, 1994] qui généralisent l'enchère Vickrey aux cas multi-objets avec objets hétérogènes et où chaque agent désire acquérir un ou plusieurs objets.

En dépit de la simplicité apparente des mécanismes d'enchères multi-objets que nous venons de citer, les travaux théoriques dans le domaine restent limités. En effet, le choix d'un mécanisme d'enchères dépend essentiellement de son efficacité et des revenus qu'il doit générer. On est loin dans l'état actuel de la recherche d'avoir fondé une théorie complète et cohérente des enchères multiples.

Plusieurs tentatives néanmoins ont été faites dans ce sens, voir par exemple les travaux de Ausubel [Ausubel, 1997, Ausubel, 2000] pour la généralisation des enchères ascendantes, ou celles de Maskin [Maskin and Esö, 2000] pour une discussion générale sur le sujet.

### **3.5.3. Les enchères combinatoires**

Dans une enchère multi-objets, il est quelquefois possible de permettre la soumission d'offres sur une combinaison d'objets. Les prix offerts par un enchérisseur pour un objet dépendent alors des autres objets qu'il a déjà gagné ou qu'il pourrait éventuellement gagner. Ceci peut être dû au fait que certains objets sont complémentaires, ou qu'ils constituent les composants d'un tout (exemple : écrans d'ordinateurs et unités centrales). Ce type d'enchères s'appelle des enchères combinatoires ou enchères combinées.

Un exemple d'application des enchères combinatoires est celui d'une enchère où est vendu du matériel usagé. Un acheteur peut évaluer une télévision au prix  $x$  et un magnétoscope au prix  $y$ , mais il évalue la paire télévision et magnétoscope au prix  $z > x + y$ . Ici, les deux objets sont complémentaires. Un autre cas de figure est celui où un acheteur peut évaluer une seule copie d'un livre usagé à un prix  $x$  mais deux copies du même livre au à un prix  $z < 2x$ . Dans ce cas, les deux objets sont substitués l'un de l'autre [Leyton-Brown et al., 2000a].

Les enchères combinatoires sont intéressantes dans le sens qu'elles permettent une meilleure allocation des objets selon les besoins spécifiques des acheteurs. Seulement, le nombre exponentiel des combinaisons possibles engendre une difficulté dans la modélisation et la formalisation de ce type d'enchères. C'est la raison pour laquelle les enchères combinatoires ne sont utilisées

que rarement, dans des applications bien spécifiques [Nisan, 2000]. Les enchères de la FCC font partie de cette catégorie d'enchères combinatoires.

Les premiers travaux sur les enchères combinatoires ont été effectués par Rassenti, Smith et Bulfin en 1982 [Rassenti et al., 1982] qui ont développé un mécanisme pour l'allocation des tranches temporelles dans les aéroports entre les différentes compagnies aériennes. Il existe, par ailleurs, différentes implémentations des enchères combinatoires à l'instar du mécanisme de sélection adaptative par l'utilisateur introduit par Banks et ses collègues [Banks et al., 1989] qui est un mécanisme décentralisé, à offres continues communiquées via une sorte de babillard électronique. Cette implémentation permet la soumission d'offres sur des paquets d'objets choisis par les participants.

#### ***3.5.3.1. Le problème avec les enchères combinatoires :***

Dans une enchère combinatoire, le vendeur reçoit des offres pour des paquets d'objets différents. Le but du vendeur étant d'allouer ses objets de sorte à maximiser ses revenus. Il s'agit d'un problème d'optimisation combinatoire assez complexe [Leyton-Brown et al., 2000b]. C'est pourquoi la plupart des recherches sur les enchères combinatoires se sont concentrés sur des cas particuliers du problème qui réduisent sensiblement les champs d'applications des enchères combinatoires. D'autres chercheurs ont essayé de définir des mécanismes qui rendent les enchères combinatoires traitables du point de vue computationnel en ajoutant des contraintes comme des pénalités en cas de désistement, des règles d'activités, etc. [Rothkopf et al., 1998]. Récemment, plusieurs chercheurs ont commencé à proposer des algorithmes pour la détermination du gagnant d'une enchère combinatoire générale (voir par exemple [Sakurai, 2000, Gonen and Lehman, 2000]).

## **3.6 Conclusion**

Un des points fondamentaux des systèmes multi-agents et leurs capacités et besoins d'interagir dans un environnement donné. Cet interaction se manifeste souvent par la négociation dans un but de coopération, résolution de conflit, coordination, etc.

Nous avons présenté dans la première partie de ce chapitre la définition de la négociation que nous considérons dans ce mémoire ainsi qu'une taxonomie des négociations. Cette taxonomie n'est bien sûr pas exhaustive, mais présente les négociations automatiques les plus répandues actuellement.

Dans la deuxième partie nous avons parlé de la forme de négociation la plus répandue dans le commerce électronique c'est-à-dire les enchères, tout en

présentant les quatre types de base des enchères, en suite les enchères multi-objets et en fin les enchères combinatoires et on a constaté que les enchères combinatoires présentent un grand déficit aux chercheurs a cause de leurs explosions combinatoires. Dans le chapitre suivant on va essayer de donner un modèle mathématique pour les enchères combinatoires et simuler ce modèle a l'aide des systèmes multi-agents.

# **MODELE ET REALISATION**

Après ce que nous avons vu dans les chapitres précédents, nous allons présenter un modèle formel d'enchères combinatoire et concevoir une application apte à simuler ce modèle.

## 4.1 Fondements mathématiques d'enchère

Considérons un enchère multi-objet avec un seule vendeur, un nombre fini d'acheteurs  $A_1 \dots A_n$  et un nombre fini d'objets, chaque objet possède une quantité  $Q_j$

Chaque acheteur  $A_i$  veut acquérir une quantité  $q_i$  de l'objet  $O_j$

Nous somme dans un cas d'enchère si et seulement si :

$$\exists j, \forall i \sum_{i=1}^n q_{i,j} > Q_j$$

Chaque acheteur envoi sa demande en indiquant la quantité de chaque objet et aussi le prix total, ce prix est calculé et envoyé vers le bidder (Coordinateur) comme s'il s'agissait d'un seule Objet.

$$P_i = \sum_{j=1}^n V_{i,j} * q_{i,j}$$

Où

$P_i$  : le Prix total de l'acheteur numéro i envoyer vers le Coordinateur.

$V_{i,j}$  : est l'évaluation de l'objet numéro j estimé par l'acheteur numéro i.

$Q_{i,j}$  : est la quantité de l'objet numéro j demandé par l'acheteur numéro i.

Généralement l'agent acheteur suit deux stratégies pour formuler sa demande, l'augmentation de prix et diminution de la quantité.

1. Augmentation de prix : si la demande de l'agent acheteur  $A_i$  à l'instant T n'est pas satisfaite et son prix  $P_i < P_{MAX}$  alors il recalculera son prix tel que :

$$P_{i,T+1} = F(P_{i,T})$$

Où  $F(P_{i,T})$  est une fonction croissante, généralement  $F(P_{i,T}) = P_{i,T} + C \ \ C \in \mathfrak{R}$

2. Diminution de la quantité : si la demande de l'acheteur  $A_i$  est toujours pas satisfaite et le prix égale au prix maximum alors il procédera à sa deuxième stratégie qui consiste à diminuer les quantités des produit un par un et recalculer les quantités des autres objets dépendants de l'objet diminué on utilisant les formules de chaque objet entré par l'acheteur, si toutes les quantités égale aux quantité minimales alors l'acheteur  $A_i$  abandonne l'enchère.

$$\exists j \ q_j > q_{\min j} \ q_{j,T+1} = F(q_{i,T})$$

Où  $F(q_{i,T})$  est une fonction décroissante, généralement

$$F(q_{i,T}) = q_{i,T} - c \ \ c \in \mathfrak{R}.$$

Les enchères multi-objets, comme leur nom l'indique, sont des négociations qui impliquent différents objets qui devant être négociés.

Une fonction d'utilité est utilisée pour comparer les différentes offres. Cette fonction permet de pondérer l'importance de chaque objet.

Cet fonction est donné par  $q_i = \sum_{j=1}^m C_j * q_{i,j}$

Où :

$P_i$  est le prix proposer par l'acheteur numéro  $i$ .

$C_j$  est le poix ou la coefficient de l'objet numéro  $j$ .

$q_{i,j}$  est la quantité de l'objet numéro  $j$  demandée par le  $i^{\text{ieme}}$  acheteur.

Soit  $T$  un taux de réduction donné par le vendeur c'est-à-dire si j'ai trois exemplaires d'un même objet, et un acheteur veut acheter un seul objet il payera «  $P = p_u$  » où  $p_u$  est le prix unitaire s'il veut acheter deux objets il payera pour chaque objet  $P = p_u - T * p_u$  .c.à.d. le prix suit une fonction linière donner par :

$$P = p_u (1 - T (q-1))$$

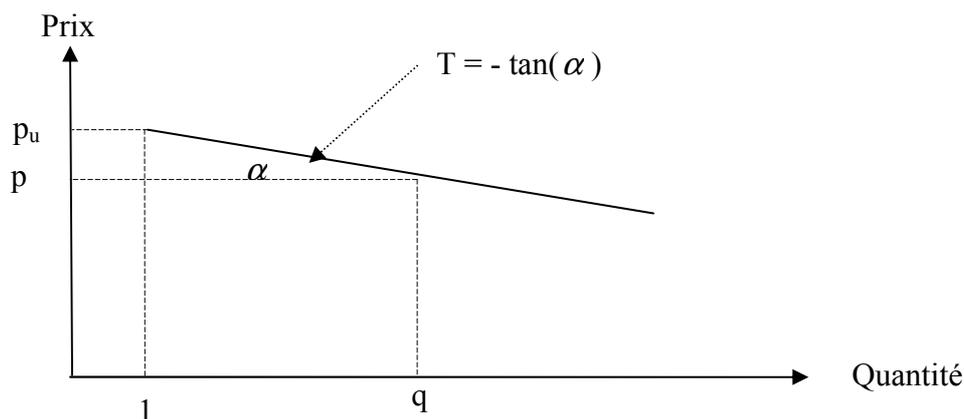


Figure 13 : équation de réduction.

La fonction d'évaluation qui détermine le ou les acheteur(s) gagnant(s) calcule la distance entre le point qui a les coordonnées  $P_i$  le prix unitaire proposé par l'acheteur  $i$  et la quantité demandée calculé par la fonction d'utilité et son image sur la ligne de l'équation :  $P = p_u (1 - T (q-1))$ .

La distance  $D_i$  est donné par :

$$D_i = \sqrt{(q_i - \bar{q})^2 + (p_i - \bar{p})^2}$$

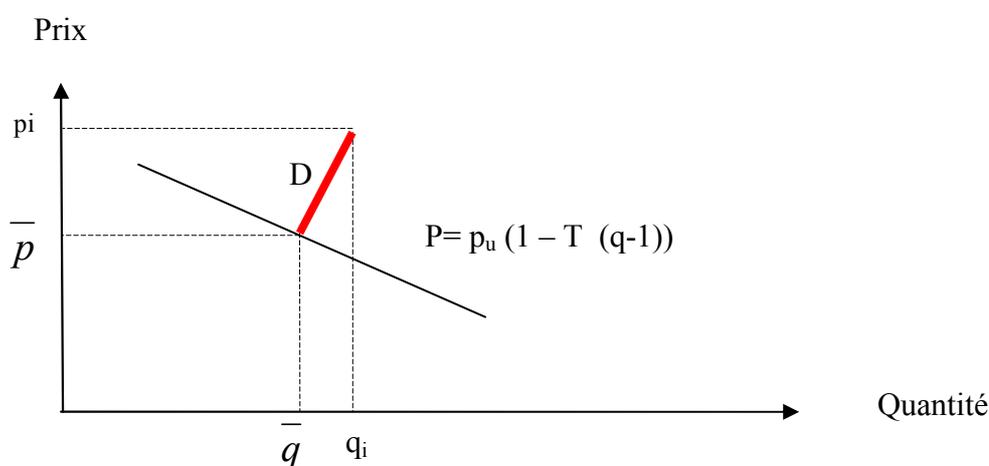


Figure 14: la fonction d'évaluation.

## 4.2 Architecture générale:

On a construit notre architecture sur un framework à base de coordinateur. Ce composant joue le rôle d'un intermédiaire entre les agents vendeurs et les agents acheteurs et établit un support de communication en centralisant les sources d'informations. Ce composant implémente le Design Pattern Singleton qui assure l'unicité de coordinateur dans le système.

Les trois pôles de coordinateur, acheteur et vendeur interagissent les uns avec les autres par l'intermédiaire des opérations de publication, de recherche et de liaison. Ces opérations sont implémentées via le langage KQML (Knowledge query manipulation language), qui est un langage de communication inter-agent.

Un vendeur informe l'agent coordinateur de son existence en utilisant l'interface de publication de l'agent coordinateur pour permettre aux agents ache-

teurs d'accéder à cette enchère. Les informations publiées décrivent l'enchère et spécifient les produits exposés. L'agent acheteur consulte l'agent coordinateur pour localiser les enchères publiées. Grâce aux informations sur les enchères

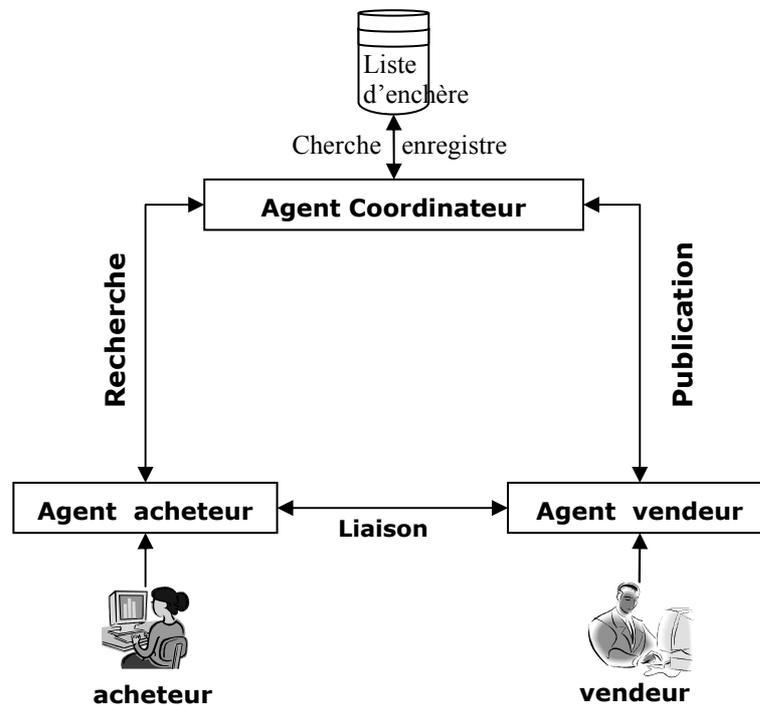


Figure 15: Architecture générale

obtenues par l'agent coordinateur, l'acheteur peut lier, ou appeler, l'agent vendeur directement

### 4.3 Conception

Nous avons utilisé l'extension de langage universel UML pour les agents comme outil de modélisation. UML est devenu un outil de modélisation objet incontournable du fait de sa facilitation à représenter et à comprendre les solutions objet :

Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solutions.

L'aspect formel de sa notation, limite les ambiguïtés et les incompréhensions.

Son indépendance par rapport aux langages de programmation, aux domaines d'application et aux processus, ont fait un langage universel.

### **4.3.1. Diagramme de classe**

Ce diagramme présente la structure statique d'enchère et les relations entre leurs classes, et aussi les interfaces utilisées par ces classes.

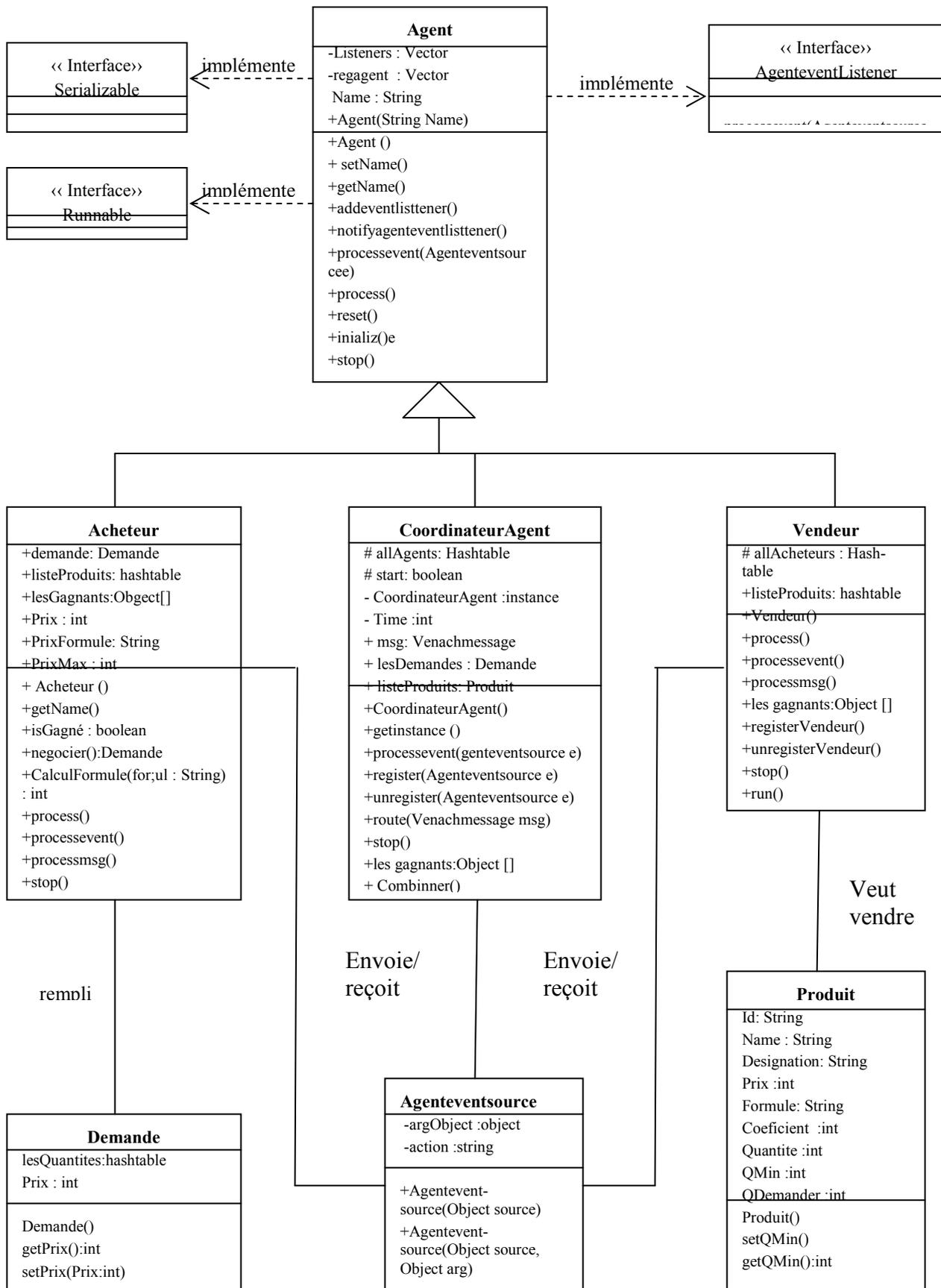


Figure 16: diagramme de classe

### 4.3.2. Diagramme de collaboration

Ce diagramme présente l'ensemble des rôles joués par les agents de la maison d'enchère( utilisateur, Agentvendeur, Agentacheteur, coordinateur ) et il montre également des interactions entre ces agents à travers des envois de messages qui définissent une action précise .

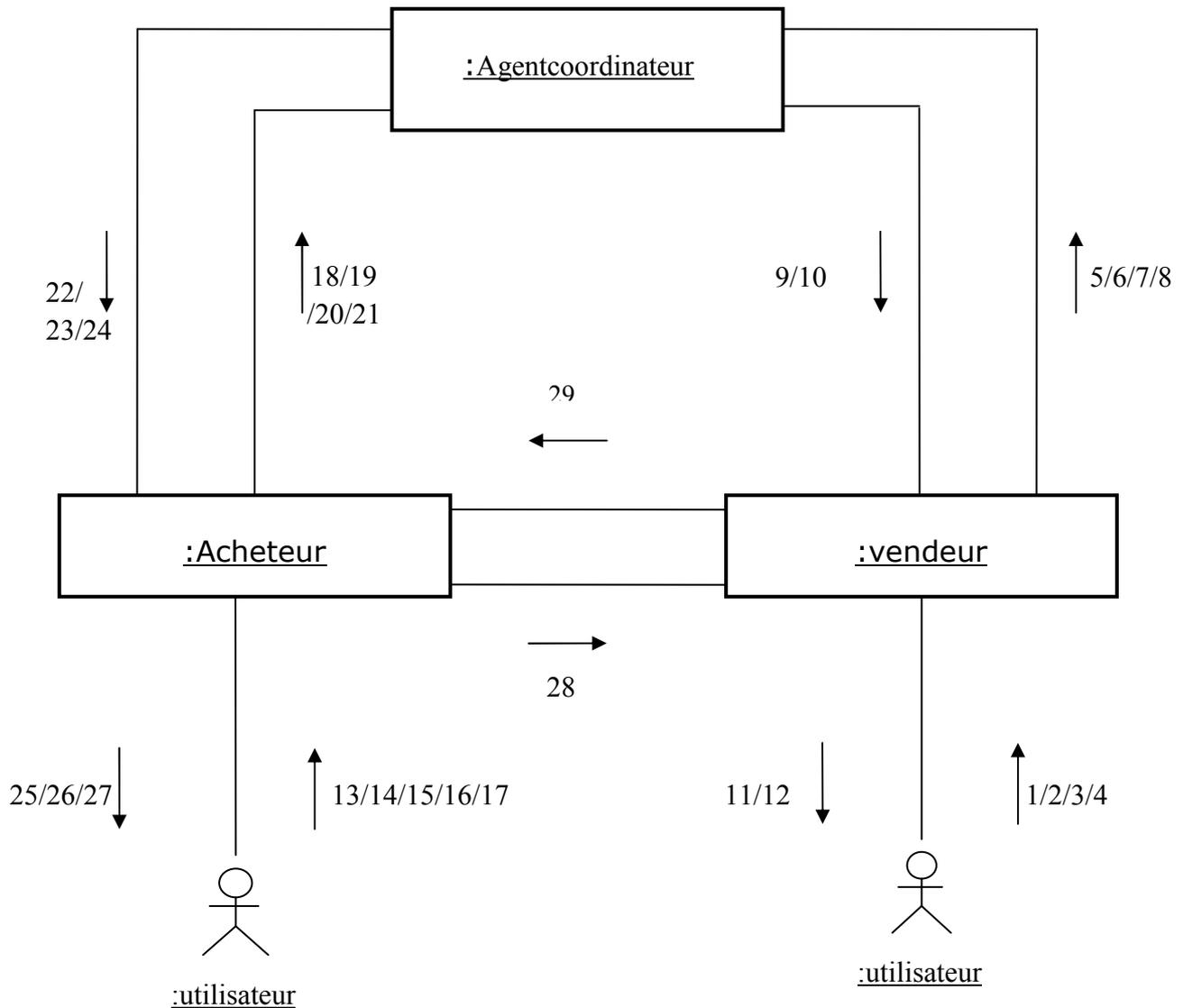


Figure 17: Diagramme de collaboration

*Utilisateur Agent Vendeur :*

- \_ (1) Demander à l'agent vendeur de s'inscrire (ou de se connecter) chez l'agent coordinateur
- \_ (2) Introduire objets à vendre + informations (prix minimum,.... )
- \_ (3) Demander à l'agent de lancer une enchère
- \_ (4) Demander à l'agent de quitter la maison d'enchères

*Agent Vendeur Agent Coordinateur*

- \_ (5) performative = registry : Demande d'enregistrement chez le coordinateur comme vendeur
- \_ (6) performative = advertise : Introduire objets à vendre + informations (prix minimum,.... )
- \_ (7) Demande d'arrêt d'une enchère
- \_ (8) Demande de quitter l'enchères

*Agent Coordinateur Agent Vendeur*

- \_ (9) Admettre (ou refuser) l'agent comme vendeur
- \_ (10) Permettre à l'agent vendeur de quitter l'enchères

*Agent Vendeur Utilisateur*

- \_ (11) Afficher le déroulement de l'enchère
- \_ (12) Afficher les résultats à la fin de l'enchère

*Utilisateur Agent Acheteur*

- \_ (13) Demander à l'agent acheteur de s'inscrire chez l'agent coordinateur
- \_ (14) Demander la liste des produits qu'ils sont en vent.
- \_ (15) Introduire les spécifications du produit à acheter
- \_ (16) Demander à l'agent de se retirer d'une enchère
- \_ (17) Demander à l'agent de quitter l'enchères

*Agent Acheteur Agent coordinateur*

- \_ (18) performative = registry : Demande d'enregistrement chez le coordinateur comme acheteur
- \_ (19) performative = recommand all : Demande la listes des produits qu'ils sont en vent .

- \_ (20) performative= offer: envoie un offre
- \_ (21) Demande de quitter la maison d'enchères

*Agent coordinateur Agent Acheteur*

- \_ (22) Admettre (ou refuser) l'agent comme acheteur
- \_ (23) performative= make-offer :demande de créer un nouvelle offre
- \_ (24) États de l'enchère (meilleure offre, fin de l'enchère, . . .)

*Agent Acheteur Utilisateur*

- \_ (25) Afficher la liste des enchères
- \_ (26) Afficher le déroulement des enchères
- \_ (27) Afficher les résultats d'une enchère

*Agent Acheteur Agent Vendeur*

- \_ (28) continu le déroulement de la transaction.

*Agent Vendeur Agent Acheteur*

- \_ (29) continu le déroulement de la transaction.

### **4.3.3. Diagramme de séquence**

Les diagrammes de séquences montrent des interactions entre les agents de la maison d'enchère comme nous avons vue dans le diagramme de collaboration mais en tient en compte le facteur de temps pour provoquer ces interactions.

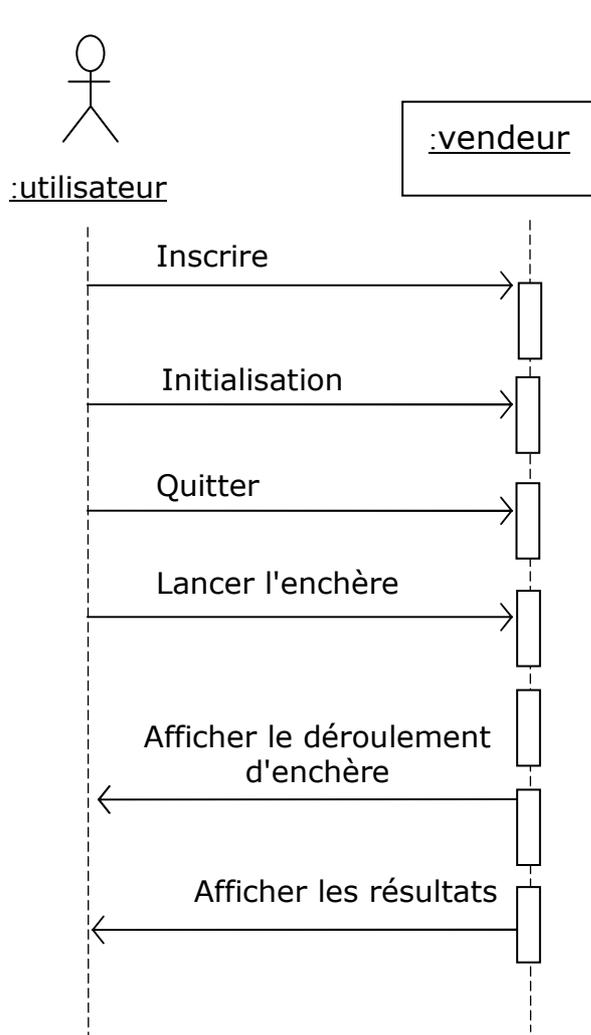


Figure 18:diagramme de séquence illustrant le cas d'utilisation de inscription d'un vendeur

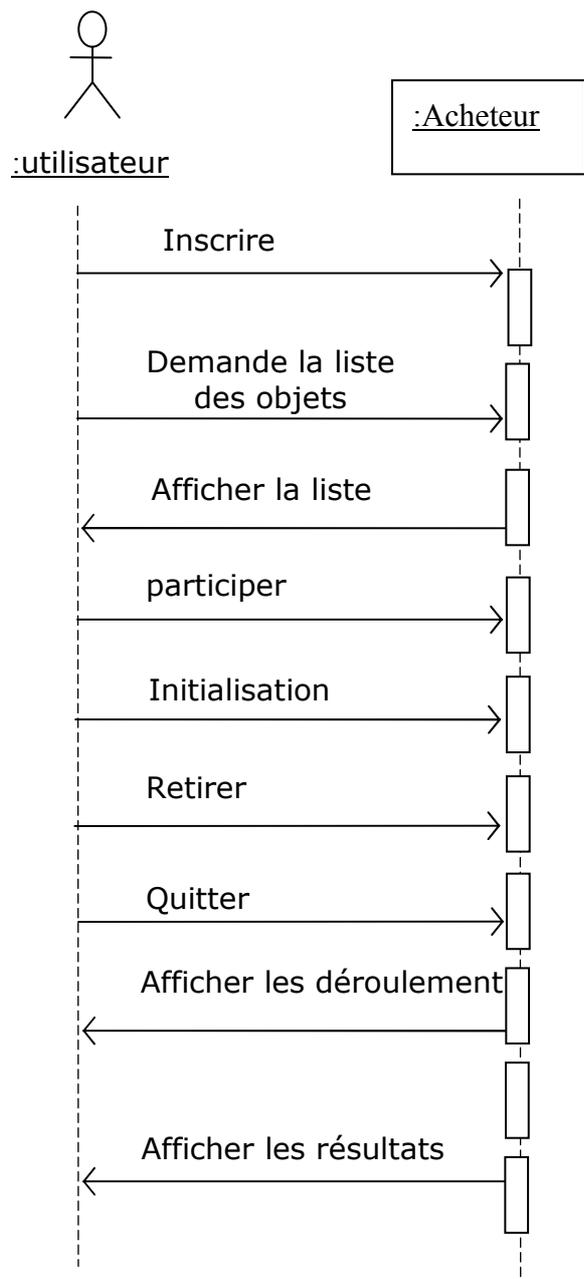


Figure 19:diagramme de séquence illustrant le cas d'utilisation de inscription d'un Acheteur

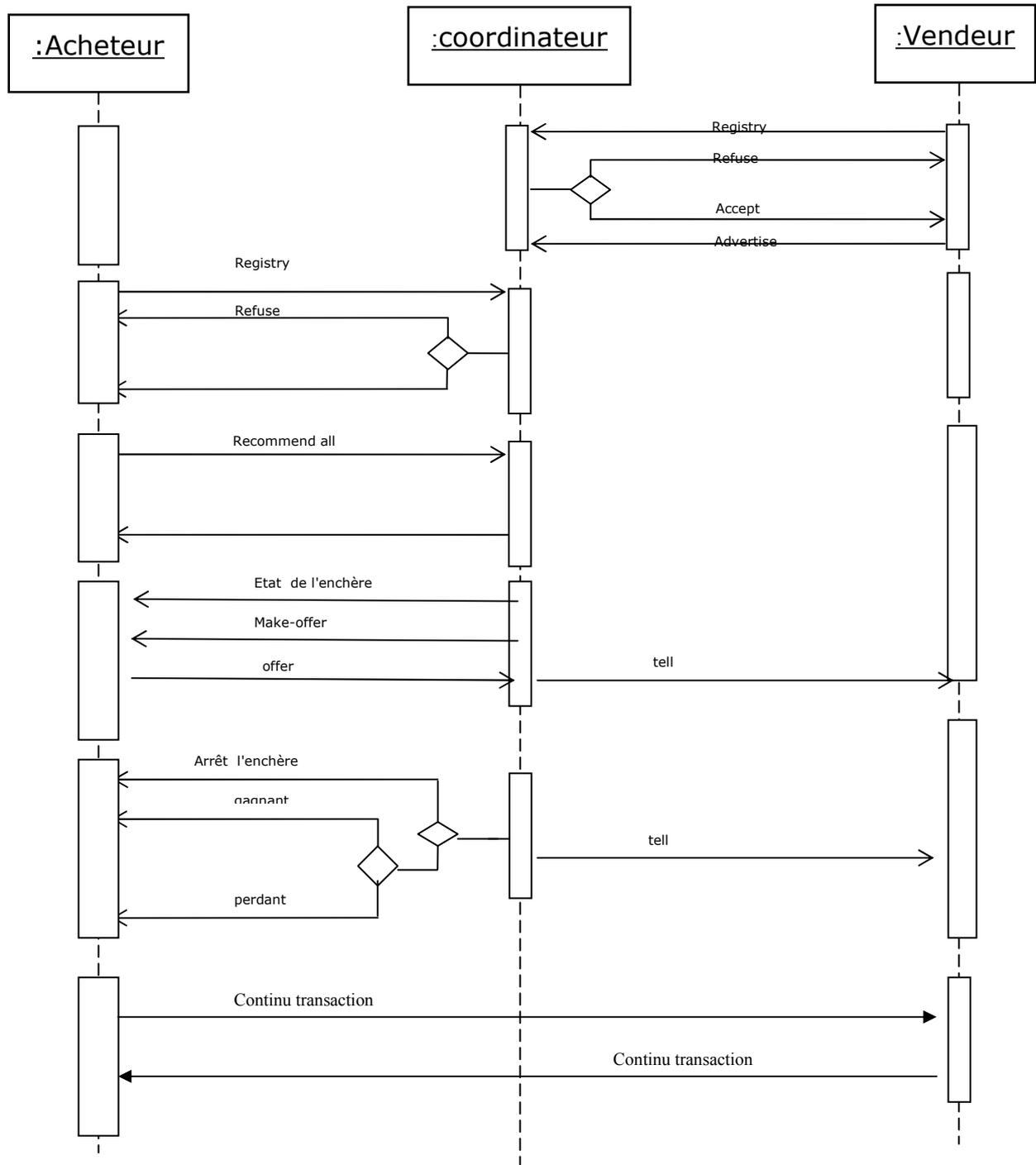


Figure 20 :diagramme de séquence illustrant le cas d'utilisation de déroulement d'un enchère

#### 4.3.4. Diagramme d'états- transitions

*Agent coordinateur :*

La figure ci-dessous montre le comportement de l'agent coordinateur durant le déroulement de l'enchère. Au début il est en état d'attente, lorsqu'il reçoit un message, il se met en état de détermination du message.

Il existe trois types de messages envoyés :

**Registry:** lorsqu'il reçoit un message il enregistre l'émetteur du message dans un hashtable qui contient tous les agents et il se met en attente.

**Recommend All :** il effectue une opération d'envoi de la liste des objets existants et revient en état d'attente

**Advertise:** il s'agit donc d'une création de l'enchère associé a l'agent émetteur et revient toujours en état d'attente.

**Abandon :** il effectue une opération de traitement de l'abandon

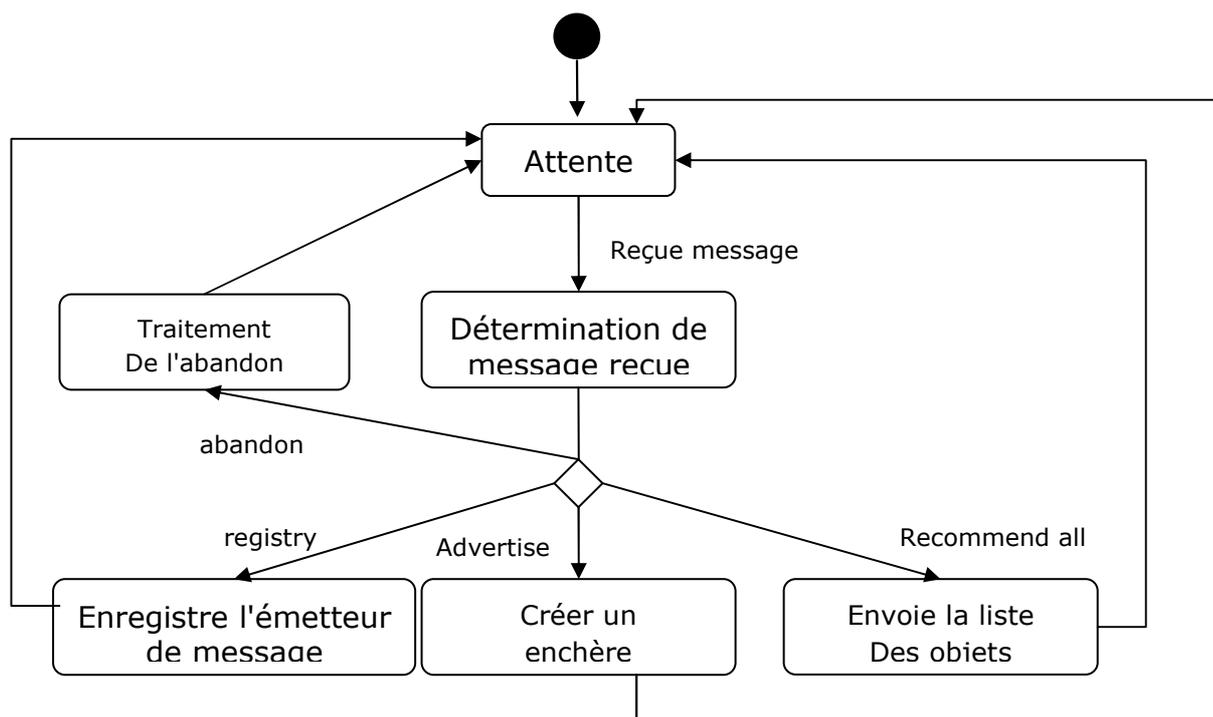


Figure 21: diagramme d'état qui illustre le comportement d'agent coordinateur

*Agent vendeur :*

Le fonctionnement de l'agent vendeur est simple comme le montre la figure ci-dessous, il se charge de contrôler le déroulement de l'enchère. Le comportement primitif du vendeur se base essentiellement sur une boucle entre un

état d'attente active, l'envoi et la réception des messages envoyés par les différents acheteurs.

L'agent vendeur lance l'enchère et annonce aux différents acheteurs son commencement. Il se met ensuite en état d'attente de réception des offres.

A chaque fois que l'agent vendeur reçoit une offre d'un agent acheteur, il déterminera la liste des agents Acheteurs gagnants et diffusera cette liste vers tous les acheteurs.

L'agent vendeur connaît le nombre des acheteurs participants à l'enchère, à chaque fois qu'il reçoit un message abandons de la part d'un participant, il décrémente le nombre d'acheteurs de un et revient en état d'attente.

Une fois que la liste des acheteurs gagnants est inférieure ou égale le nombre de tous les acheteurs, l'agent vendeur annonce la fin de l'enchère qui arrête l'enchère.

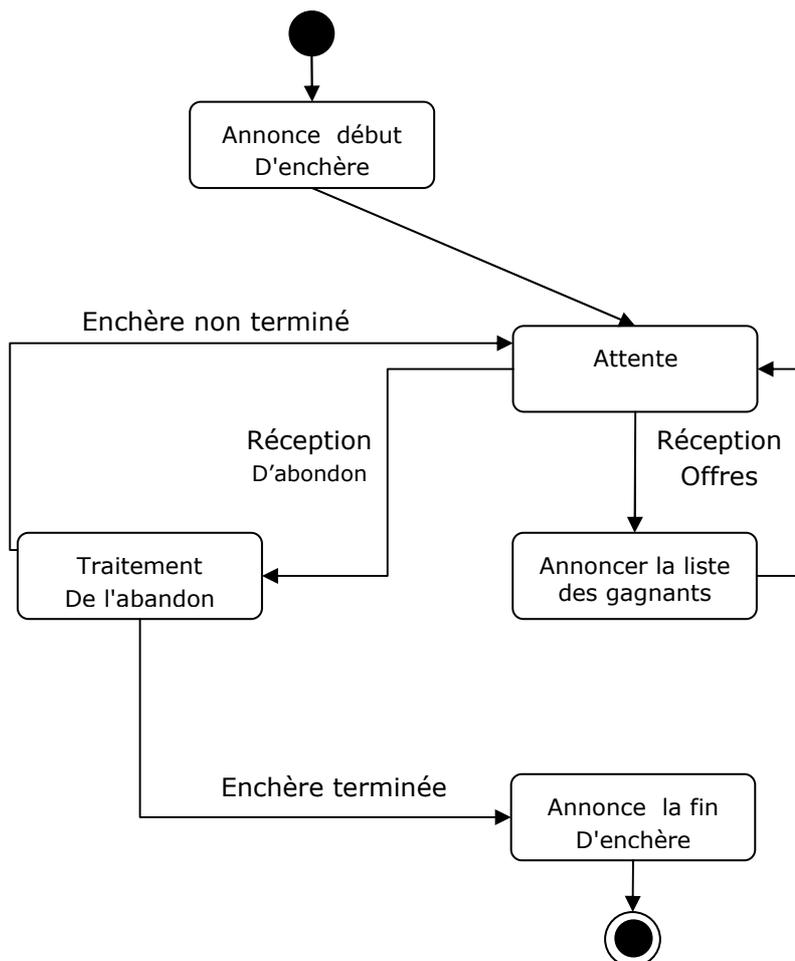


Figure 22: diagramme d'état qui illustre le comportement d'agent vendeur

### *Agent Acheteur :*

La figure ci-dessous montre les états de l'agent acheteur durant le processus d'enchérissement. Lorsque un agent acheteur entame une enchère, il se met en état d'attente d'une offre figurant en un message envoyé par le vendeur.

Lors de la réception de message (meilleure offre) l'agent effectue une vérification du prix atteint. S'il est toujours l'un des gagnants de l'enchère. Il revient alors à l'état d'attente. Sinon il procède à une détermination de l'offre à soumettre. S'il est capable de soumettre une nouvelle offre, il passe à l'état soumission de l'offre, il envoie un message à l'agent vendeur contenant sa nouvelle offre et les quantités demandées.

Dans le cas où l'acheteur ne peut pas soumettre une offre, il essaye alors de diminuer la quantité qu'il demande plus ou moins selon sa stratégie de diminution.

Tant qu'il ne peut pas soumettre une offre au vendeur, il continue à diminuer la quantité et reconsidérer la possibilité de faire une nouvelle offre, Si malgré les tentatives de diminution de la quantité, l'acheteur atteint sa quantité minimale et n'arrive toujours pas à faire une offre, il annonce son abandon et passe à l'état abandon et attend les résultats de l'enchère.

Finalement, lorsque l'enchère est arrêtée ou terminée, et l'agent acheteur était en état d'attente il reçoit un événement sous forme de message du vendeur l'informant de la fin de l'enchère, ce dernier passe alors à l'état de traitement des résultats de l'enchère où lui sera communiquée les quantités qu'il a éventuellement gagnée ainsi que le prix à payer.

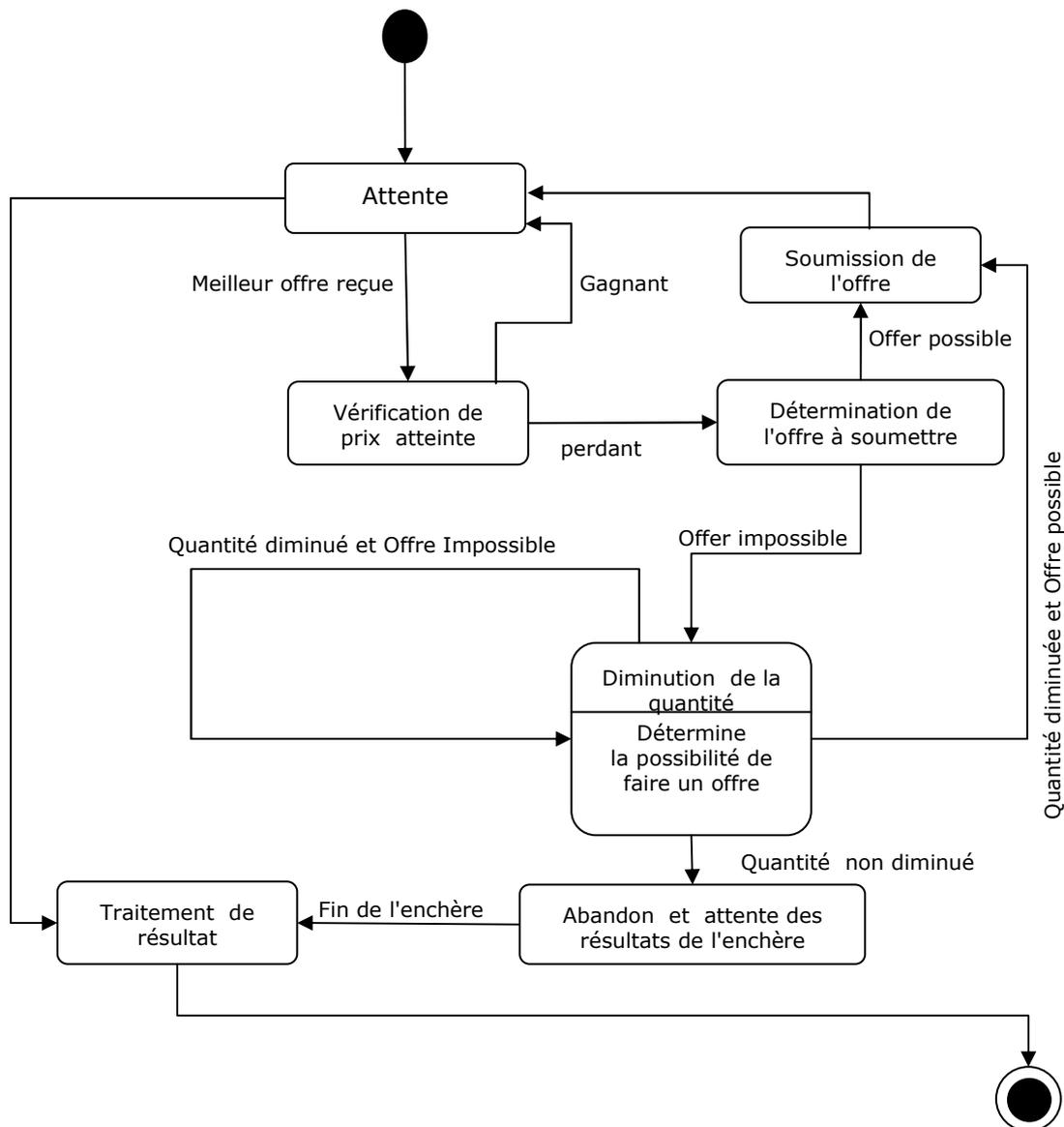


Figure 23: diagramme d'état qui illustre le comportement d'agent acheteur

#### 4.4 Réalisation :

Afin de pouvoir réaliser notre système d'enchère combinatoire basé sur la technologie agent, l'ensemble d'outils que nous avons choisis comme support technique de réalisation s'est porté sur un environnement orienté objet en utilisant le langage Java. Ce choix a été déterminé par le fait que

- l'enchère doit être disponible et intégré dans les navigateurs web existant sur le marché,

- pour sa grande souplesse dans l'implémentation des SMA,
- pour son support natif du multithreading
- son indépendance de la plateforme

## Présentation de logiciel

### *La fenêtre principale de l'application*

Cette fenêtre est utilisé pour créer le vendeur et les acheteurs, il existe aussi un jauge pour déterminer la vitesse de déroulement de l'enchère et en fin un bouton pour lancer la simulation.



Figure 24 La fenêtre principale

**Initialisation de vendeur :**

Dans cette fenêtre le vendeur initialise les caractéristiques de ses objets à vendre leurs noms, les quantités disponibles et les coefficients a la fin il envoi cette paquet vers le coordinateur.

The screenshot shows a window titled "Vendeur" with a blue title bar. Inside, there is a form for adding items. The form has the following fields and controls:

- Designation:** A text area containing "montre suisse de quartz".
- Id:** A text box with "M4251".
- Nom:** A text box with "Montre".
- Prix:** A text box with "30".
- Quantite:** A text box with "8".
- Coefficient:** A text box with "2".
- Réduction:** A text box with "5" followed by a "%" symbol.
- Buttons:** Three buttons stacked vertically: "Ajouter un Objet", "retirer l'Objet", and "Envoyer".

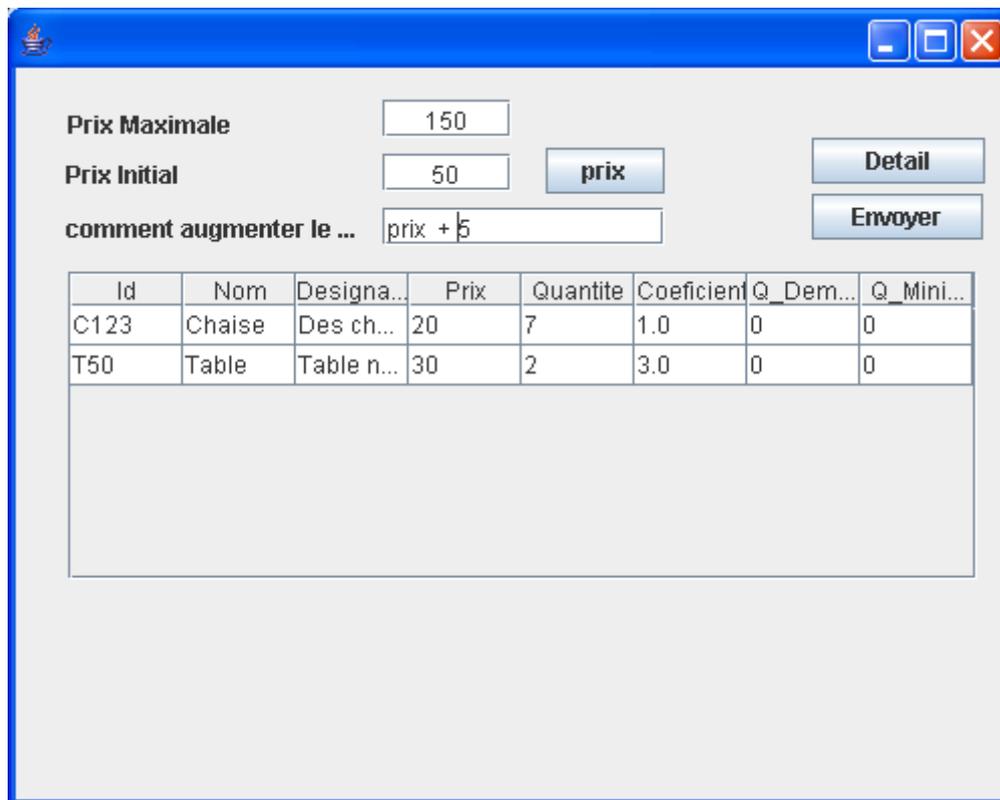
Below the form is a table with the following data:

Id	Nom	Designation	Prix	Quantite	Coefficient
T50	Table	Table noire...	30	2	3.0
C123	Chaise	Des chais...	20	7	1.0

**Figure 25 Initialisation de vendeur**

***Initialisation de la stratégie d'augmentation de prix de l'acheteur :***

Dans cette fenêtre l'acheteur initialise le prix initial et le prix maximal et la formule qui sera utilisée en cas où l'acheteur ne sera pas considéré comme gagnant pour déterminer le prix suivant.



Id	Nom	Designa...	Prix	Quantite	Coefficient	Q_Dem...	Q_Mini...
C123	Chaise	Des ch...	20	7	1.0	0	0
T50	Table	Table n...	30	2	3.0	0	0

**Figure 26 stratégie d'augmentation de prix**

***Initialisation de la stratégie de la diminution de la quantité :***

Dans cette fenêtre l'acheteur voit les détails de chaque objet et initialise la quantité demandée et la quantité minimale acceptée de chaque objet tout en précisant la formule à utiliser pour diminuer la quantité si l'utilisateur est considéré comme non gagnant.

The screenshot shows a software window with a blue title bar and standard Windows window controls (minimize, maximize, close). The window is divided into several sections:

- Designation:** A text area containing "table noire de surface de 5 mètre de 15ième siècle".
- Id:** A text input field containing "table".
- Nom:** A text input field containing "table".
- Prix:** A numeric input field containing "30".
- Coefficient:** A numeric input field containing "3.0".
- Quantite:** A numeric input field containing "2".

On the right side of the window, there are three buttons: "Valider", "Suivant", and "Preced...".

Below the main fields, there is a section for defining a formula:

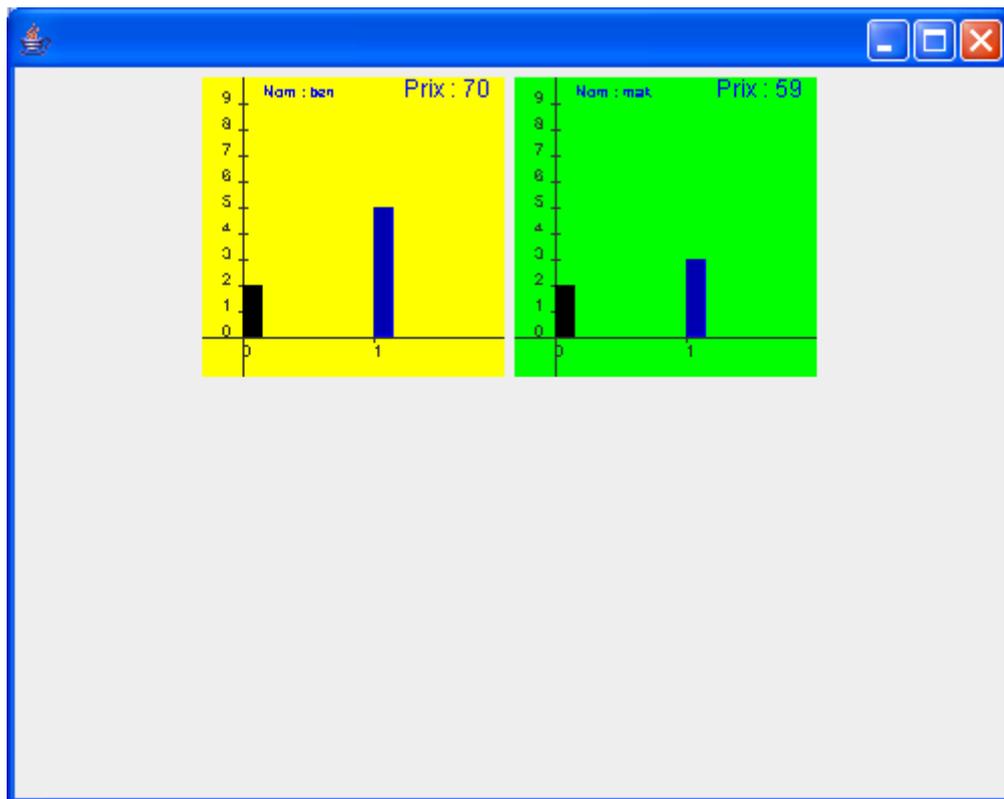
- Quantité initial:** A numeric input field containing "2".
- Quantité minim...:** A numeric input field containing "1".
- A set of arithmetic operators: "+", "-", "\*", "/", "(", and ")", each in its own button.
- Two buttons labeled "table" and "id350b", which likely represent variables or identifiers.

At the bottom, there is a large text input field containing "table-1|".

Figure 27 la stratégie de la diminution de la quantité

***Suivie de l'enchère :***

Cette fenêtre montre deux acheteurs représentés par deux diagrammes en cas de compétition . si l'arrière plan est verte cela signifie que l'agent acheteur pour le moment est gagnant.



**Figure 28** Suivie de l'enchère

## 4.5 Conclusion

Dans une enchère combinatoire, le vendeur reçoit des offres pour des paquets d'objets différents. Le but du vendeur étant d'allouer ses objets de sorte à maximiser ses revenus. Il s'agit d'un problème d'optimisation combinatoire assez complexe

Nous avons proposé un modèle mathématique d'enchères combinatoire pour la négociation automatique entre agents logiciels. et on a essayé de simuler et de concevoir ce modèle et en fin présenter quelques interfaces de notre application réalisée.

# **CONCLUSION**

Le but principal du présent travail consistait à contribuer à l'intégration du paradigme agents logiciels et systèmes multi-agents dans le cadre du commerce électronique interentreprises, on a essayé d'automatiser les tâches les plus complexes et fastidieuses du processus commercial.

Une des tâches les plus importantes dans les processus commerciaux est la négociation des prix et des services à acheter ou à vendre, pour automatiser cette tâche nous avons choisi le paradigme d'agent afin d'implémenter notre modèle.

En effet, le recours à des agents logiciels rationnels pouvant négocier automatiquement permet de rendre les places de marchés électroniques plus efficaces. Cela permet également une meilleure distribution des produits par rapport aux besoins et un gain de temps considérable. Nous nous sommes alors penché sur l'étude des mécanismes de la négociation automatique

Nous nous avons intéressé au cas des enchères combinatoires basés sur l'enchère anglaise, avec possibilité pour les agents acheteurs de diminuer leurs quantités demandées lors de l'enchère. Nous avons présenté un modèle mathématique de ce type d'enchères et des stratégies d'enchérissement des agents acheteurs. Nous avons simulé ce modèle à l'aide d'un système multi-agents afin de vérifier sa validité et de l'évaluer empiriquement.

### **Perspectives**

Nous pensons que les agents et les systèmes multi-agents sont le « futur » du développement en informatique ainsi que des nouvelles technologies. Les possibilités offertes par le paradigme agent sont immenses. Nous pensons que l'automatisation des processus de négociation sur Internet sera une sorte de révolution semblable à celle de l'automatisation des processus industriels.

Les extensions possibles à notre travail sont nombreuses. Voici quelques unes des pistes

- extension du modèle proposé aux cas de quantités d'objets vendus pouvant changer, c'est à dire que la quantité de produit disponible à la vente pourrait diminuer ou augmenter au cours de l'enchère. C'est le cas dans les bourses de valeurs par exemple ;

- approfondissement de la conception d'une place de marché électronique basée sur les enchères en étudiant les questions de : sécurité, authentification des agents, transaction et paiement, etc. ;
- l'automatisation presque totale du processus de production et d'approvisionnement au sein des entreprises. Tout le processus serait géré par des agents intelligents capables d'aller s'approvisionner sur une place de marché électronique quand c'est nécessaire. Ces agents seraient capables de négocier les meilleurs prix, de trouver de nouveaux clients ou de nouveaux fournisseurs pour l'entreprise, d'optimiser la commercialisation et l'achat de produits selon l'état des marchés, etc.

On est encore loin, aujourd'hui, de cette vue utopique que pourrait devenir le paysage économique de demain. Des obstacles de sécurité, d'efficacité et de validité des techniques utilisées, d'hétérogénéité des systèmes existants, de non-confiance des humains, etc. repousserons, pour une date inconnue, l'émergence de ce modèle économique complètement automatisé. Néanmoins, tout indique que les agents logiciels et les enchères sont là pour rester.

# Annexe A

## Les principales fonctions

Dans cette annexe en va citer les principales fonctions utilisées pour implémenter notre application. Toutes ces fonctions sont écrites en java.

La première fonction appelée Combiner est utilisée par l'agent coordinateur pour avoir toutes les combinaisons possible d'une liste d'acheteurs, c'est-à-dire si j'ai trois acheteurs A,B,C les combinaison possibles sont A, B, C, AB, AC, BC et ABC puis les enregistre sur un tableau appelle ListeCombinaison.

```
void Combiner(Object[] ListeAcheteur, Object[] laCombinaison) {
    for (int i = 0; i < ListeAcheteur.length; i++) {
        Object[] CopylaCombinaison= new Object[laCombinaison.length + 1];
        for (int k = 0; k < laCombinaison.length ; k++) {
            CopylaCombinaison [k] = laCombinaison [k];
        }
        CopylaCombinaison [laCombinaison.length] = ListeAcheteur[i];
        if (SommeDemandes(CopylaCombinaison) != null) {
            ListeCombinaison.put(CopylaCombinaison, SommeDemandes(CopylaCombinaison));
        }
        Object[] ResteLiAch = null;
        if (ListeAcheteur.length > i + 1) {
            ResteLiAch = new Object[ListeAcheteur.length - i - 1];
        }
        for (int j = 0; j < ListeAcheteur.length - i - 1; j++) {
            ResteLiAch [j] = ListeAcheteur[i + j + 1];
        }
        Combiner(ResteLiAch, CopylaCombinaison);
    }
}
```

La deuxième fonction appelée lesGagnant est utilisée aussi par l'agent coordinateur pour déterminer la combinaison momentanément gagnante

```
public Object[] lesGagnant(Hashtable ListeCombinaison) {
    Double maxEval = Double.NEGATIVE_INFINITY;
    Demande d;
    Object o, O;
    Object[] liste = null;
    for (Enumeration e = ListeCombinaison.keys(); e.hasMoreElements(); ) {
        o = e.nextElement();
        d = (Demande) lesCombinaison.get(o);
        if (d != null) {
            float Q = 0;
```

```
    for (Enumeration E = d.keys(); E.hasMoreElements(); ) {
        O = E.nextElement();
        Q += (Integer) d.get(O) * ((produit) listProduits.get(O)).getCoefficient();
    }
    double Eval = eval(reduction, Q, d.getPrix());
    if (maxEval < Eval) {
        maxEval = Eval;
        liste = (Object[]) o;
    }
}
return liste;
}
```

La troisième fonction appelée négociateur est utilisée par l'agent acheteur pour implémenter les deux stratégies utilisées par l'acheteur c'est à dire l'augmentation de prix et la démissions des quantités.

```
public Demande negocier() {
    if (demande.getPrix() == PrixMax) {
        produit p;
        for (int i = 0; i < ProdRecuratif.size(); i++) {
            index = (index + i) % ProdRecuratif.size();
            p = (produit) ProdRecuratif.get(index);
            if (p.getQDemande() != p.getQMin()) {
                p.setQDemande(Math.max(calculer(p.getFormule()), p.QMin));
                demande.put(p.getId(), p.getQDemande());
                for (int j = 0; j < ProdnonRecuratif.size(); j++) {
                    p = (produit) ProdnonRecuratif.get(i);
                    p.setQDemande(Math.max(calculer(p.getFormule()), p.QMin));
                    demande.put(p.getId(), p.getQDemande());
                }
                return demande;
            }
        }
        return null;
    } else {
        demande.setPrix(calculer(getPrixFormule()), PrixMax);
        return demande;
    }
}
```

# Bibliographie

- [Bartolini and Preist, 2001] Bartolini, C. and Preist, C. (2001). A Framework for Automated Negotiation. Technical Report HPL-2001-90, HP Laboratories Bristol.
- [Bartolini et al., 2002a] Bartolini, C., Preist, C., and Jennings, N. R. (2002a). Architecting for reuse : A software framework for automated negotiation. In Proc. 3rd IntWorkshop on Agent-Oriented Software Engineering, pages 87–98, Bologna, Italy.
- [Bartolini et al., 2002b] Bartolini, C., Preist, C., and Jennings, N. R. (2002b). A Generic software framework for automated negotiation. Technical Report HPL-2002-2, HP Laboratories Bristol.
- [Benyoucef et al., 2000] Benyoucef, M., Keller, R. K., Lamouroux, S., Robert, J., and Trusart, V. (2000). Towards a Generic E-Negotiation Platform. In Proceedings of the Sixth International Conference on Re-Technologies for Information Systems, pages 95–109, Zurich, Switzerland.
- [Bonabeau et Theraulaz, 1994] E. Bonabeau, G. Theraulaz, «Intelligence collective.»Hermès, Paris, 1994.
- [Bonabeau et al., 1999] E. Bonabeau, M. Dorigo, G. Theraulaz, «Swarm Intelligence :From Natural to artificial Systems ». Oxford University Press, 1999.
- [Bratman, 1987] M. E. Bratman. « Intention, Plans, and Practical Reason ». Harvard University Press, Cambridge, MA, 1987.
- [Bratman et al., 1988] M. E. Bratman, D. Israel, M. E. Pollack. « Plans and resource bounded practical reasoning », Computational Intelligence 4 : 349-355. 1988.
- [Brooks, 1986] R. A. Brooks. « A robust layered control system for a mobile robot.» IEEE Journal of Robotics and Automation, RA-2(1) :14-23, 1986.
- [Brooks, 1991] R. A. Brooks. « Intelligence without representation ». Artificial Intelligence , 47(1-3) :139-159, 1991.

[Casteran et al., 2000] J.C. Casteran, M.P.Gleize, P.Glize, « Des méthodologies orientées multi-agent », Actes des Journées Francophones sur les l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents, JFIADSMA'00 , Hermes, ISBN 2-7462-0176-3, p. 191-207, 2000.

[Chavez and Maes, 1996] Chavez, A. and 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, London, UK.

[Collins et al., 1998] Collins, J., Youngdahl, B., Jamison, S., Mobasher, B., and Gini, M. (1998). A Market Architecture for Multi-Agent Contracting. In 2nd Int'l Conf on Autonomous Agents, pages 285–292, Minneapolis.

[Drogoul et Meyer, 1999] A. Drogoul, J. Meyer, « Intelligence artificielle située. » Hermès Paris Science Publications, 1999.

[Ferber, 1995] J. Ferber, Les systèmes multiagents vers une intelligence collective, InterEdition, Paris, France, 1995.

[Ferguson, 1992] I. A. Ferguson. « TouringMachines : An Architecture for Dynamic Rational, Mobile Agents ». PhD thesis, Clare Hall, University of Cambridge, UK, November 1992.

[Fikes et Nilsson, 1971] R.E. Fikes et N.J. Nilsson, « STRIPS: a new approach to the application of theorem proving to problem solving ». Artificial Intelligence, 2 (3-4):189--208. 1971.

[Fischer et al., 1999] K. Fischer, B. Chaib-draa, H. J. Müller, J. P. Müller, M.Pischel. «A simulation approach based on negotiation and cooperation between agents.» IEEE Trans. on Systems, Man, and Cybernetics, 29(4), pp. 531-545, 1999.

[Gerin-Lajoie, 2000] Gerin-Lajoie, R. (2000). Architecture informatique de gnp version 1.0. Technical report, CIRANO.

[I. Brigui et al] I. Brigui M.J. Bellosta S. Kornman S. Pinson D. Vanderpooten Un mécanisme de négociation multicritère pour le commerce électronique.

[Jennings et al., 1998] N. R. Jennings, K. Sycara, M. Wooldridge. « A Roadmap of Agent Research and Development » *Int Journal of Autonomous Agents and Multi-Agent Systems* 1 (1) 7-38, 1998.

[Jennings, 1999] Jennings N. R., « Agent-Oriented Software Engineering », *Multi-Agent System Engineering*, 9<sup>o</sup> European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW'99 , F.J. Garijo and M. Boman, Springer Verlag, LNAI 1647, 1999.

[Liu, 2002] Y.J. Liu, « De la nécessité et de la façon de coopérer, de s'autoorganiser et de se reconfigurer dans des systèmes de production complexes: modélisation et gestion d'un système virtuel pour une approche multi-agents » Thèse de Doctorat en Génie Industriel, INPG, France, 2002.

[Müller et Pischel, 1994] J.P. Müller et M. Pischel. « Modelling interacting agents in dynamic environments ». In *Proceedings of the Eleventh European Conference on Artificial Intelligence (ECAI-94)*, Amsterdam, The Netherlands, pp. 709-713, 1994.

[Noriega, 1998] Noriega, P. (1998). *Agent mediated auctions : The Fishmarket Metaphor*. PhD thesis, University of Barcelona

[Rodriguez-Aguilar et al., 1997] Rodriguez-Aguilar, J. A., Noriega, P., Sierra, C., and Padget, J. (1997). FM96.5 A Java-based Electronic Auction House. In *Proceedings of the Second International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'97)*, London, UK.

[Wooldridge et Jennings, 1995] M. Wooldridge et N. R. Jennings. « Intelligent agents: Theory and practice ». *The Knowledge Engineering Review*, vol. 10(2) pp. 115–152, 1995.

[Wurman et al., 1998] Wurman, P. R., Wellman, M. P., and Walsh, W. E. (1998). *The Michigan Internet AuctionBot : A Configurable Auction Server for Human and Software Agents*. In *Proceedings of the Second International Conference on Autonomous Agents (Agents-98)*, Minneapolis, MN, USA.