

Université de Montréal

Conception d'un modèle de création
générique d'agents de médiation

par

Frédéric Loisier

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures en vue de
l'obtention du grade de maître ès sciences (M.Sc.) en informatique

Août 2003

© Frédéric Loisier, 2003



QA

76

U54

2003

V, 044

AVIS

L'auteur a autorisé l'Université de Montréal à reproduire et diffuser, en totalité ou en partie, par quelque moyen que ce soit et sur quelque support que ce soit, et exclusivement à des fins non lucratives d'enseignement et de recherche, des copies de ce mémoire ou de cette thèse.

L'auteur et les coauteurs le cas échéant conservent la propriété du droit d'auteur et des droits moraux qui protègent ce document. Ni la thèse ou le mémoire, ni des extraits substantiels de ce document, ne doivent être imprimés ou autrement reproduits sans l'autorisation de l'auteur.

Afin de se conformer à la Loi canadienne sur la protection des renseignements personnels, quelques formulaires secondaires, coordonnées ou signatures intégrées au texte ont pu être enlevés de ce document. Bien que cela ait pu affecter la pagination, il n'y a aucun contenu manquant.

NOTICE

The author of this thesis or dissertation has granted a nonexclusive license allowing Université de Montréal to reproduce and publish the document, in part or in whole, and in any format, solely for noncommercial educational and research purposes.

The author and co-authors if applicable retain copyright ownership and moral rights in this document. Neither the whole thesis or dissertation, nor substantial extracts from it, may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms, contact information or signatures may have been removed from the document. While this may affect the document page count, it does not represent any loss of content from the document.

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé :

Conception d'un modèle de création générique d'agents de médiation

présenté par :

Frédéric Loisier

a été évalué par un jury composé des personnes suivantes :

Mme Julie Vachon, présidente-rapporteur

M. Peter G. Kropf, directeur de recherche

M. Jean Vaucher, membre du jury

Mémoire accepté le 10 septembre 2003

Résumé

Le commerce électronique connaît depuis plusieurs années une croissance importante et avec lui apparaissent des marchés aussi diversifiés que les enchères et les négociations électroniques. L'automatisation de la participation à ces négociations se fait de plus en plus par le biais d'un concept informatique en plein essor, celui des agents. La problématique qui nous intéresse est celle de la création générique d'agents de négociations, afin d'éviter de faire l'implémentation spécifique d'un agent négociateur pour chaque type de marché. Dans ce travail, nous présentons dans un premier temps une revue documentaire sur le domaine des enchères, des négociations, des agents, ainsi qu'une analyse de plusieurs types de marchés nous permettant de dégager les états, les événements et les actions communs à ceux-ci. Par la suite, nous proposons un modèle de création générique d'agents, ainsi qu'une architecture pour ceux-ci, qui prenne en paramètre d'entrée les règles décrivant la négociation ciblée ainsi que les besoins émis par l'utilisateur. Enfin, nous présentons une implémentation effectuée à partir du modèle proposé, que nous soumettons à des tests afin de pouvoir valider le modèle.

Mots clés : commerce électronique, agents, enchères électroniques, négociations électroniques.

Summary

The popularity of electronic commerce has significantly increased during the past few years, and many electronic marketplaces, such as electronic auctions and negotiations have appeared. Agent technology is used more and more to automate the buying process on those markets. Our concern in this work is to create negotiation agents without having to implement them one by one according to the negotiation type.

First, we present a bibliographic review of auctions, negotiations and agent technology. We analyze several types of negotiations in order to find common states, events that can occur and actions a participant may possibly make during those negotiations. Then, we propose a generic agent creation model, and an architecture for these agents, that takes negotiation rules and user's needs as inputs. Afterwards, we present an implementation of our model and the tests that we have made in order to validate it, and discuss the results.

Key words: electronic commerce, agents, electronic negotiations, electronic auctions.

Table des matières

RÉSUMÉ	III
SUMMARY	IV
TABLE DES MATIÈRES	V
LISTE DES TABLEAUX	VIII
LISTE DES FIGURES	IX
REMERCIEMENTS	XI
1. INTRODUCTION GÉNÉRALE	1
1.1. COMMERCE ÉLECTRONIQUE ET AGENTS DE MÉDIATION	1
1.2. MISE EN CONTEXTE ET PROBLÉMATIQUE	2
1.2.1. <i>Le projet TEM</i>	2
1.2.2. <i>Le projet Consensus</i>	3
1.2.3. <i>Problématique</i>	5
1.3. OBJECTIFS DU TRAVAIL.....	5
1.4. APPROCHE ADOPTÉE	6
1.5. ORGANISATION DU MÉMOIRE.....	7
2. NÉGOCIATIONS ET ENCHÈRES	8
2.1. INTRODUCTION	8
2.2. LES NÉGOCIATIONS	8
2.2.1. <i>Définitions</i>	9
2.2.2. <i>Processus d'une négociation</i>	11
2.2.3. <i>Les négociations combinées</i>	12
2.3. LES ENCHÈRES	13
2.3.1. <i>Introduction : historique des enchères</i>	13
2.3.2. <i>Objectifs et définition de l'enchère</i>	15
2.3.3. <i>L'enchère anglaise</i>	16
2.3.4. <i>L'enchère hollandaise</i>	18
2.3.5. <i>L'enchère par l'offre secrète au premier prix (first-bid)</i>	19
2.3.6. <i>L'enchère Vickrey (ou par l'offre secrète au second prix)</i>	19
2.3.7. <i>L'enchère double (double auction)</i>	20
2.3.8. <i>Récapitulatif des enchères</i>	22
2.3.9. <i>Classification des enchères</i>	23
2.3.9.1. Les attributs relatifs au type d'enchère.....	23
2.3.9.2. Les attributs de l'enchère	25
2.4. RÉSUMÉ	26
3. MODÉLISATIONS DES NÉGOCIATIONS	27
3.1. INTRODUCTION	27
3.2. MODÉLISATION À L'AIDE DE FSM	27
3.3. ETUDE DE CINQ MARCHÉS	28

3.3.1.	<i>Enchère anglaise</i>	29
3.3.2.	<i>Enchère hollandaise à une seule unité</i>	31
3.3.3.	<i>Enchère hollandaise multi-items</i>	32
3.3.4.	<i>Enchère par l'offre secrète (sealed-bid auction)</i>	34
3.3.5.	<i>La négociation à deux pairs (Two parties négociation)</i>	35
3.4.	ANALYSE DES ÉTATS DES AUTOMATES	36
3.5.	RÉSUMÉ	38
4.	LES AGENTS	39
4.1.	INTRODUCTION	39
4.2.	LA TECHNOLOGIE AGENT	39
4.2.1.	<i>Historique de la recherche sur les agents</i>	40
4.2.2.	<i>Définitions</i>	41
4.2.3.	<i>Propriétés d'un agent</i>	42
4.2.4.	<i>Agent versus Objet</i>	43
4.3.	ARCHITECTURES D'AGENTS	44
4.3.1.	<i>Architecture délibérative</i>	45
4.3.2.	<i>Architecture réactive</i>	47
4.3.2.1.	Agents à réflexes simples.....	47
4.3.2.2.	Agents avec maintien d'états	49
4.3.3.	<i>Architecture BDI</i>	51
4.3.4.	<i>Architecture en couches ou hybride</i>	53
4.4.	RÉSUMÉ	55
5.	MODÈLE DE CRÉATION GÉNÉRIQUE D'AGENTS DE MÉDIATION	56
5.1.	INTRODUCTION	56
5.2.	CHOIX DE L'ARCHITECTURE AGENT	56
5.3.	LE MODÈLE DE CRÉATION D'AGENT.....	58
5.3.1.	<i>Les besoins du client</i>	59
5.3.2.	<i>Les paramètres des règles de la négociation</i>	60
5.4.	LES NÉGOCIATIONS CONCERNÉES.....	62
5.4.1.	<i>Analyse des événements-actions</i>	62
5.4.1.1.	Hypothèses quant aux messages échangés.....	63
5.4.1.2.	Enchère anglaise	64
5.4.1.3.	Enchère hollandaise	65
5.4.1.4.	Enchère hollandaise multi-items	66
5.4.1.5.	Enchère par l'offre secrète	66
5.4.1.6.	Négociation à deux pairs.....	67
5.4.2.	<i>Regroupement des événements-actions</i>	68
5.5.	ARCHITECTURE DE L'AGENT GÉNÉRÉ.....	70
5.5.1.	<i>L'architecture globale</i>	70
5.5.2.	<i>Description des différents modules</i>	71
5.6.	IMPLÉMENTATION	71
5.6.1.	<i>Contexte d'implémentation</i>	72
5.6.1.1.	Serveur de négociation.....	72
5.6.1.2.	Communications, langage de programmation.....	72
5.6.2.	<i>La classe « Message » et les types de messages</i>	72
5.6.2.1.	Le corps du message, <i>messageObject</i>	73
5.6.2.2.	Les autres paramètres du message	73
5.6.3.	<i>La classe « AgentFactory »</i>	74
5.6.3.1.	La méthode <i>readRules</i>	74
5.6.3.2.	La méthode <i>createAgent</i>	75
5.6.3.3.	L'interface usager <i>AgentFacPanel</i>	75
5.6.4.	<i>La classe « Agent »</i>	76
5.6.4.1.	Les attributs de la classe « Agent »	76
5.6.4.2.	La méthode <i>run</i>	76
5.6.4.3.	La méthode <i>see</i>	77
5.6.4.4.	La méthode <i>updateState</i>	77

5.6.4.5. La méthode <i>action</i>	78
5.6.5. <i>Résumé des classes et méthodes</i>	78
5.7. FONCTIONNEMENT GÉNÉRAL	80
5.8. RÉSUMÉ.....	82
6. ANALYSE DU TRAVAIL EFFECTUÉ	83
6.1. INTRODUCTION	83
6.2. ANALYSE DES COMPORTEMENTS	83
6.2.1. <i>Procédures de tests</i>	83
6.2.2. <i>Comportements observés</i>	84
6.3. LIMITES	86
6.3.1. <i>Limites au niveau conceptuel</i>	86
6.3.2. <i>Limites au niveau technologique</i>	87
6.4. RÉSUMÉ.....	88
7. CONCLUSION GÉNÉRALE	89
7.1. REVUE DU MÉMOIRE	89
7.2. RENCONTRE DES OBJECTIFS	90
7.3. TRAVAUX FUTURS	92
7.3.1. <i>Variété des marchés et stratégies</i>	92
7.3.2. <i>Vers les systèmes multiagents</i>	93
7.3.2.1. Communication entre agents.....	93
7.3.2.2. La coordination	95
7.4. CONCLUSION.....	96
BIBLIOGRAPHIE	97

Liste des tableaux

Tableau 2.1 – Mécanismes de différents types d'enchères [Anthony and Jennings, 2002].....	22
Tableau 5.1 - Liste des événements-actions : enchère anglaise	64
Tableau 5.2 - Liste des événements-actions : enchère hollandaise	65
Tableau 5.3 - Liste des événements-actions : enchère hollandaise multi-items.....	66
Tableau 5.4 - Liste des événements-actions : enchère par l'offre secrète.....	67
Tableau 5.5 - Liste des événements-actions : négociation à deux pairs	68
Tableau 5.6 - Regroupement des événements-actions	69
Tableau 7.1 - Types d'agents et leurs capacités [Huhns and Stephens, 1999].....	94

Liste des figures

Figure 1.1 - Architecture de Consensus [Benyoucef et al., 2001a].....	4
Figure 3.1 - FSM d'un marché à prix fixe [Kumar and Feldman, 1998a]	28
Figure 3.2 - Représentation de l'enchère anglaise à une unité	30
Figure 3.3 - Représentation de l'enchère hollandaise à une unité.....	31
Figure 3.4 - Représentation de l'enchère hollandaise à n unités.	33
Figure 3.5 - Représentation de l'enchère par l'offre secrète	34
Figure 3.6 - Représentation de la négociation à deux pairs.....	35
Figure 4.1 - Schéma d'un agent à réflexes simples [Russell and Norvig, 1995]	48
Figure 4.2 - Schéma d'agent avec maintien d'états [Russell and Norvig, 1995]	50
Figure 4.3 - Schéma d'agent avec maintien d'états [Wooldridge, 1999].....	50
Figure 4.4 - Architecture BDI [Wooldridge, 1999].....	51
Figure 4.5 - Architecture en couches horizontales [Wooldridge, 1999]	53
Figure 4.6 - Architecture en couches verticales [Wooldridge, 1999].....	54
Figure 5.1 - Représentation conceptuelle du modèle	58
Figure 5.2 - Architecture de l'agent généré	70
Figure 5.3 - La composition de l'objet <i>Message</i>	73
Figure 5.4 - Diagrammes des classes	79
Figure 5.5 - Interface usager lors du démarrage.....	80
Figure 5.6 - Interface usager après la lecture du fichier de règles.....	81
Figure 5.7 - Fenêtre permettant de visualiser les règles du marché.	81
Figure 5.8 - Exemple d'état de l'agent lors d'une négociation.	82
Figure 7.1 - Coordination entre agents [Huhns and Stephens, 1999].....	95

À Simone, Jean, Françoise et Dominique

Remerciements

J'aimerais remercier tous les gens qui m'ont permis de compléter ce mémoire, à commencer par mon directeur de recherche, Peter G. Kropf, pour sa confiance, ses conseils et ses commentaires toujours pertinents.

Je tiens à remercier tous les gens du CIRANO (Centre interuniversitaire de recherche en analyse des organisations) qui ont collaboré, de près ou de loin, à l'aboutissement de ce mémoire, ou qui m'ont supporté tout au long de mon travail, et plus particulièrement Robert Gérin-Lajoie pour sa générosité, ses nombreux conseils et suggestions, tous les gens du groupe Nadim et GNP, et enfin Benoît Bourbeau qui a été d'une générosité sans borne, me prodiguant ses judicieux conseils et m'encourageant à toujours être rigoureux dans ma démarche de rédaction.

Merci aux membres du groupe *Consensus*, toujours prêts à répondre à mes questions ou me fournir de l'information : Morad Benyoucef et Hakim Alj pour leur générosité et leurs précieux conseils, Rudolph K. Keller, Sarita Bassil et Kim Lévy.

Je remercie le CRSNG (Conseil de recherche en sciences naturelles et en génie du Canada) pour son soutien financier, ainsi que le CIRANO et les Laboratoires Universitaires Bell (LUB) pour leur soutien technique.

Je tiens à remercier mon père de m'avoir transmis son goût pour l'informatique alors que j'étais enfant, et de m'avoir encouragé à continuer quelques années plus tard; sur ce dernier point, merci également à Cyril Déglise.

Merci à tous les amis qui ont su me comprendre, me soutenir et m'encourager durant ma rédaction.

Enfin, je veux remercier mes parents, qui m'ont toujours appuyé avec beaucoup de générosité et d'affection dans toutes mes entreprises et sans qui ce mémoire n'aurait jamais vu le jour, ma grand-mère Simone qui m'accompagne en pensées, ainsi que ma compagne Dominique qui a été d'un très grand support, m'encourageant et faisant preuve de beaucoup de compréhension tout au long de ma rédaction.

Frédéric Loisier

1. Introduction générale

1.1. Commerce électronique et agents de médiation

La croissance rapide de l'Internet ces dernières années a vu apparaître un grand nombre de marchés électroniques offrant de bonnes opportunités d'affaires, autant pour les entreprises que pour les particuliers. Des catalogues électroniques en lignes aux centres d'achat virtuels, de nombreuses applications du commerce électronique se sont développées au fil des ans. Mais, bien que l'information sur les produits et les fournisseurs soit de plus en plus accessible, les achats électroniques restent encore largement non-automatisés, et le consommateur doit encore être présent aux différentes étapes du processus d'achat [Maes et al., 1999]. Il doit bien souvent faire lui-même la cueillette d'information et le tri, prendre des décisions quant aux produits et aux fournisseurs proposés, compléter la phase de paiement, etc.

La technologie agent peut répondre à ce besoin d'automatisation : un agent peut remplir, dans ce contexte, un rôle de médiateur à bien des niveaux dans le processus d'achat, dont les différentes étapes ont été synthétisées par Terpsidis [Terpsidis et al., 1997] et reprises par Guttman [Guttman et al., 1998] comme suit :

1. **Identification des besoins** : c'est à cette étape que le consommateur prend conscience des besoins qu'il a.
2. **Courtage de produits** (*product brokering*) : il y a recherche d'informations afin de déterminer plus précisément quoi acheter à partir de critères d'évaluation que le consommateur aura fournis. On obtient un ensemble de produits à considérer.

3. **Courtage de marchands** (*merchant brokering*): à partir des ensembles de solutions trouvées, on détermine quels sont les marchands avec qui on désire faire affaire, selon des critères de réputation, de garanties, etc.
4. **Négociation** : à cette étape on détermine les conditions (*terms*) de la transaction.
5. **Achat et livraison** : le paiement et la livraison selon les conditions de la transaction.
6. **Service après-vente et évaluation** : il s'agit du service destiné au bon fonctionnement du produit après l'achat ainsi que l'évaluation par l'acheteur de sa satisfaction pour l'ensemble du processus d'achat.

Selon Guttman [Guttman et al., 1998], les étapes de courtage de produits et de marchands, ainsi que la négociation sont autant de rôles que peuvent remplir des agents de médiation pour le commerce électronique. Nous nous pencherons, dans ce travail, sur le rôle que peuvent jouer de tels agents dans l'étape de négociation.

1.2. Mise en contexte et problématique

1.2.1. Le projet TEM

Notre travail s'inscrit dans le cadre du projet TEM (*Towards Electronic Marketplaces*), un projet conjoint entre le milieu universitaire et celui de l'industrie, notamment les Laboratoires Universitaires Bell (LUB), commencé en 1999, qui implique des chercheurs de différentes disciplines, notamment les sciences économiques, les technologies de l'information, le génie logiciel et la recherche opérationnelle. L'objectif de ce projet est d'aborder des problématiques de conception de marchés, relativement à l'allocation et au contrôle des ressources et

aux mécanismes de récompense, d'étudier des protocoles ouverts pour les places de marchés électroniques, et d'explorer des concepts et des outils pour le e-commerce. Le projet Consensus, que nous présentons dans la section suivante, s'inscrit dans le dernier point évoqué.

1.2.2. Le projet Consensus

Le travail présenté dans ce mémoire s'inscrit dans le cadre d'un projet de recherche nommé Consensus, qui fait partie intégrante du projet TEM, que nous avons présenté plus haut, et qui s'intéresse aux négociations électroniques (*e-negotiations*), un domaine de recherche du e-commerce qui prend de plus en plus d'importance. Plus particulièrement, Consensus se penche sur la problématique des négociations combinées (ou CN pour *combined negotiations*). Nous expliquerons ce qu'est une négociation combinée en section 2.2.3.

Le principal objectif du projet Consensus est de développer un outil (nommé CNSS pour *Combined Negotiation Support System*) qui permette de gérer plusieurs négociations en parallèle en vue d'acquérir des biens ou des services interdépendants et ainsi de faciliter cette tâche pour l'acheteur [Benyoucef *et al.*, 2001a].

La démarche du projet Consensus se résume comme suit :

1. détermination des besoins du client.
2. courtage de biens et de marchands correspondant aux besoins du client.
3. création d'agents capables de négocier l'achat de biens sur les serveurs de négociation des marchands concernés.

Comme nous le verrons plus en détails dans ce mémoire, notre travail se concentrera à réaliser le troisième point, soit la création d'agents de négociation.

L'architecture de *Consensus* (décrite par la Figure 1.1) s'articule autour de plusieurs points importants. Tout d'abord on retrouve le système de courtage des produits et des marchands (*product/merchant brokering*). Ce système permet à l'utilisateur de choisir les produits et aussi les vendeurs avec lesquels il voudra bien négocier. Le système retournera une liste de marchands susceptibles de satisfaire les besoins de l'utilisateur qui sera ensuite remise au module de création d'agents (appelé ici *agent factory*) responsable de créer des agents qui iront négocier l'achat de produits avec les marchands.

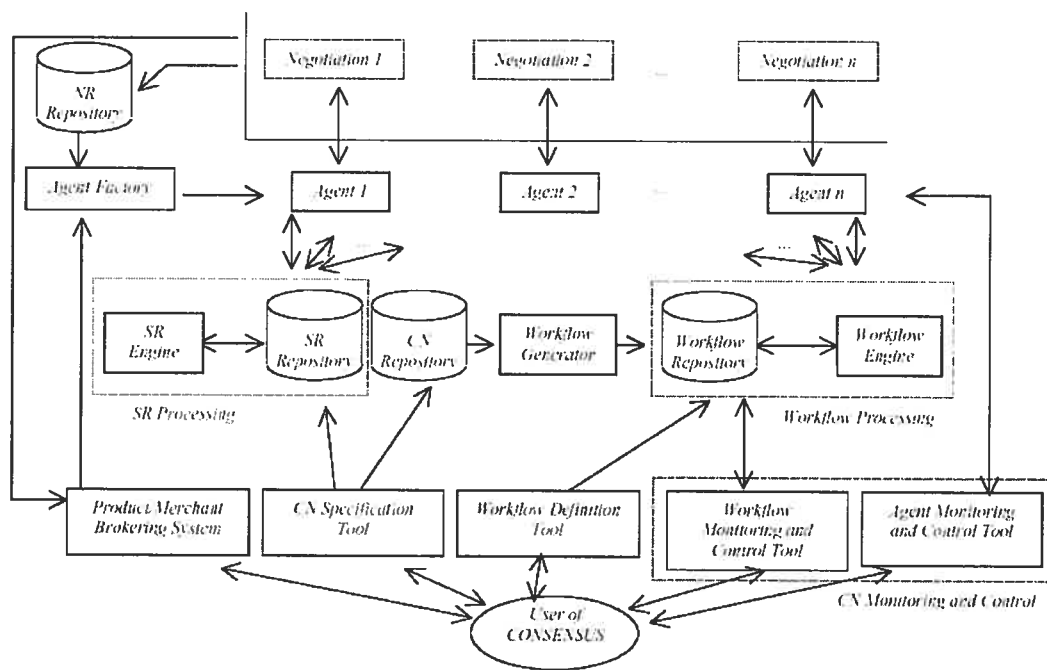


Figure 1.1 - Architecture de Consensus [Benyoucef et al., 2001a]

Un système de gestion électronique de processus (*workflow*) fait la gestion de ces agents au niveau de leur comportement individuel (stratégie à adopter) lors de chacune de leur négociation, mais également au niveau de l'ensemble des agents, afin que ceux-ci soient solidaires dans l'atteinte du but commun, soit la combinaison harmonieuse de plusieurs négociations interdépendantes.

1.2.3. Problématique

L'architecture mise en place dans le projet Consensus fait intervenir des agents pour automatiser les négociations sur les places de marchés en vue d'acquérir des produits. Il peut s'agir d'une vente aux enchères de type anglaise, hollandaise, d'une vente à prix fixe, d'une négociation de type marchandage, etc. Un agent envoyé pour négocier sur l'un de ces marchés devra au préalable avoir été conçu pour avoir un comportement compatible avec les règles ayant cours sur ce marché. Il faut donc, dans le fonctionnement actuel du projet Consensus, programmer un agent différent pour chaque type de négociation. On comprend donc très vite le côté non-adaptatif des ces agents, conçus pour une et une seule application.

Notre motivation est donc de réussir à concevoir un modèle pouvant permettre d'envoyer des agents négocier sur différents marchés sans avoir à programmer leurs mécanismes internes pour chaque type de marché.

1.3. Objectifs du travail

Après avoir défini le contexte et la problématique de notre travail, nous présentons maintenant les objectifs généraux que nous nous sommes fixés:

1. étudier les négociations et les enchères afin de pouvoir en donner une classification
2. modéliser les négociations les plus répandues du point de vue du participant afin d'analyser et de dégager les états essentiels
3. comprendre la technologie des agents et en décrire les architectures principales

4. concevoir un modèle de création générique d'agents, à la lumière des concepts étudiés plus haut, et en faire l'implémentation
5. discuter les résultats empiriques obtenus afin d'évaluer le modèle proposé

Notre objectif principal est, bien sûr, le point 4. Plus précisément, nous désirons concevoir un système capable d'instancier des agents ayant un mécanisme générique, c'est-à-dire qui auront la possibilité d'effectuer des opérations essentielles, mais auxquels seront insufflés des comportements spécifiques à leur création à partir d'un nombre restreint de paramètres, ceux-ci tenant compte à la fois des besoins du participant et des règles qui définissent la négociation à laquelle l'agent devra participer.

Comme nous le verrons dans ce travail, la difficulté de concevoir un modèle générique réside dans la définition et la classification, d'une part, des besoins du client quel que soit le marché et, d'autre part, de celle des règles des différents marchés.

1.4. Approche adoptée

Avant de concevoir un modèle, nous avons commencé par définir les différents attributs qui définissent les négociations qui nous intéressent dans ce travail, à savoir celles les plus répandues sur Internet. Nous avons également étudié les mécanismes des négociations à travers des diagrammes d'états finis. Puis, nous avons étudié différentes architectures agents afin de pouvoir ensuite en retenir une qui soit appropriée à notre problématique, soit l'architecture réactive avec maintien d'états. Nous avons ensuite fait une analyse des comportements possibles et proposé un modèle de création générique d'agents et une architecture pour ces agents. Nous

avons fait l'implémentation de notre modèle et testé celui-ci afin de pouvoir en analyser les comportements et en tirer des conclusions.

1.5. Organisation du mémoire

Ce mémoire s'articule autour de cinq parties. Tout d'abord dans le chapitre 2, nous faisons une présentation des négociations et des enchères, en suggérons des définitions, donnons quelques exemples d'enchères les plus utilisées et dressons une classification de celles-ci.

Puis, dans le chapitre 3, nous modélisons à l'aide d'automates à états finis plusieurs types de négociations. Ces modélisations nous permettent de dégager les points communs, c'est à dire les états essentiels dans lesquels un participant peut se retrouver durant le déroulement des diverses négociations.

Par la suite nous présentons, au chapitre 4, la technologie des agents, et étudions plus particulièrement quatre grands types d'architectures généralement admis dans les ouvrages scientifiques.

Dans le chapitre 5, nous nous servons des éléments présentés dans les chapitres précédents afin de justifier nos choix d'architecture et présentons notre conception d'un modèle de création générique d'agents de négociation. Nous expliquons ensuite les éléments principaux de son implémentation et décrivons son fonctionnement.

Nous exposons ensuite au chapitre 6 les résultats empiriques que nous avons obtenus à partir de l'implémentation que nous avons faite de notre modèle. Nous commentons ces résultats et les analysons afin de voir si les objectifs que nous nous étions fixés ont été atteints.

2. Négociations et enchères

2.1. Introduction

Comme nous l'avons vu au chapitre précédent, notre projet s'inscrit dans le contexte du projet Consensus, qui s'intéresse aux négociations électroniques combinées. Dans ce chapitre nous tenterons d'abord de définir précisément ce qu'est une négociation et les raisons pour lesquelles elles sont utilisées. Nous verrons quel est le processus des négociations, et expliquerons ensuite ce qu'est une négociation combinée.

Par la suite nous ferons une revue des enchères et verrons qu'elles ne sont qu'un cas particulier des négociations. Nous en présenterons un historique, en donnerons des définitions et décrirons celles qui sont les plus répandues sur Internet. Nous tenterons d'en définir les caractéristiques et nous en viendrons finalement à présenter une classification, dont nous nous servirons pour développer notre modèle (plus tard au chapitre 5).

2.2. Les négociations

Quel est le besoin à l'origine d'une négociation? Pour Lewicki [Lewicki et al., 1999], une négociation survient pour l'une des deux raisons suivantes: soit pour créer quelque chose de nouveau qu'aucune des deux parties ne pourrait faire elle-même, ou encore pour résoudre un problème ou un conflit entre deux parties. Dans cette section, nous tenterons de définir ce qu'est une négociation, pour ensuite s'intéresser aux processus qui la caractérisent et finalement nous verrons comment modéliser les négociations.

2.2.1. Définitions

Il existe de nombreuses définitions relatives aux négociations. Nous tenterons dans cette section de ne retenir que les plus pertinentes au regard des objectifs de notre travail.

Tout d'abord, Zlatev et van Eck [Zlatev and van Eck, 2003] expliquent le besoin des négociations et en donnent une définition:

« We define negotiation as a process by which actors reach agreement about joint future behaviour. The need for negotiation arises when more than one entity (actor) have overlapping interests. Whatever the origin of their problem is (conflicting or cooperative interests), an interactive process of exchanging intentions and positions on the matter of the problem takes place. The entities (actors) communicate their expectations about a potential mutually acceptable agreement. The result of the negotiation process is a kind of a contract upon which actors commit themselves for a certain future course of action. Therefore, negotiation is a mechanism for coordination. »

Une négociation peut être ainsi vue comme un procédé interactif durant lequel s'élaborent les conditions d'un échange ou d'une coopération entre deux ou plusieurs parties. Elle peut également être vue comme un problème de prise de décision, car chaque partie tente de faire des concessions afin d'arriver à un compromis avec les autres parties.

Placés d'un point de vue du commerce électronique, Beam et Segev [Beam and Segev, 1997] donnent quant à eux cette définition:

« We define negotiation in electronic commerce as the process by which two or more parties multilaterally bargain resources for mutual intended gain, using the tools and techniques of electronic commerce. Under this definition, a process in which two executives use email to exchange negotiation offers would not be considered negotiation in electronic commerce, but a process in which two intelligent software agents negotiate a solution electronically and then present it to the executives would be. »

L'exemple donné par Beam et Segev dans cette définition exprime bien la différence qui existe entre une négociation traditionnelle ayant un support électronique et ce qui peut être considéré, selon eux, comme une véritable négociation électronique.

Une autre définition, cette fois tirée de Lomuscio [Lomuscio et al., 2001] rejoint la précédente et fait un lien entre les négociations et les agents (que nous verrons plus en détails au chapitre 4):

« Negotiation is the process by which group of agents communicate with one another to try and come to mutually acceptable agreement on some matter ».

On peut enfin considérer la négociation d'un point de vue plus abstrait, comme le fait Jennings [Jennings et al., 2000], pour qui une négociation est une « recherche distribuée dans un espace d'accords potentiels ».

Afin de résumer les différents points évoqués dans les définitions que nous venons de voir, nous reprenons ici les caractéristiques des négociations telles que définies en six points par Lewicki [Lewicki et al., 1999] :

1. il faut qu'il y ait **deux parties ou plus**
2. il faut qu'il y ait un **conflit à résoudre** entre les intérêts de chaque partie
3. il faut que les participants pensent pouvoir **obtenir un meilleur marché** que ce que les autres leur offriraient d'emblée sans négocier
4. il faut que les participants cherchent à **trouver une entente** plutôt que de se battre ou de laisser une instance supérieure régler le conflit pour eux
5. il faut que les participants fassent **donnant, donnant**
6. il faut que les **différents aspects**, autant concrets (le prix) que plus abstraits (la qualité) soient couverts

2.2.2. Processus d'une négociation

Dans la section 2.2.1, nous nous sommes attachés à définir les raisons inhérentes à l'utilisation des négociations et avons tenté d'en donner plusieurs définitions. Dans cette section, nous essaierons de voir de quelle manière s'organise le déroulement d'une négociation, et quels en sont les éléments essentiels.

Kumar et Feldman [Kumar and Feldman, 1998a] donnent les cinq éléments clés que l'on retrouve lors d'un processus de négociation:

1. Une **transaction** qui peut être de diverses natures: négociable, une offre finale d'un acheteur ou d'un vendeur, ou encore un échange fixé.
2. Des **participants**, qui peuvent être des acheteurs, des vendeurs, des encanteurs, des courtiers, etc.
3. Des **messages envoyés par les participants** afin de modifier la transaction. Des exemples de ces messages sont les mises ou les offres.
4. un **flot de contrôle** qui décrit comment l'état de la transaction change à la suite des messages envoyés par les participants
5. des **messages envoyés aux participants** lors de la modification de la transaction

Pour illustrer les éléments ainsi définis, prenons un exemple. Imaginons un propriétaire qui négocie la location de son appartement (la transaction) avec des locataires potentiels (les participants). Ceux-ci font des offres (messages envoyés par les participants) au propriétaire qui ne retient que la plus haute (flux de processus) et informe les locataires potentiels du prix minimum qu'ils devront maintenant lui proposer (message envoyé aux participants) s'ils veulent continuer à négocier.

Les cinq éléments clés susmentionnés seront primordiaux pour modéliser une négociation, comme nous le verrons au chapitre 3, puis plus loin au chapitre 5.

2.2.3. Les négociations combinées

Afin de bien comprendre le projet Consensus présenté en section 1.2.2, nous tenterons dans cette section d'expliquer brièvement en quoi consistent les négociations combinées.

Selon Benyoucef [Benyoucef et al., 2001a] on parle de négociations combinées lorsqu'un acheteur désire acquérir plusieurs articles de natures différentes, et donc s'engage dans plusieurs négociations en même temps.

Si les négociations servant à acquérir les biens ou services peuvent être indépendantes les unes des autres, les mêmes biens ou services à négocier sont quant à eux interdépendants.

Pour illustrer ceci, prenons l'exemple d'une négociation combinée pour une brasserie artisanale située à Montréal. Celle-ci doit, d'une part, acheter des feuilles de houblon dans le Kent, d'autre part acquérir un service de transport par camion jusqu'à Douvres et enfin obtenir un transport par bateau jusqu'à Montréal. Même si chaque négociation est indépendante (le houblon est peut-être négocié dans une enchère, le transport par camion obtenu par un *bargain*, et le transport par bateau acheté à prix fixe) il faudra faire correspondre la date d'achat du houblon avec la date de départ du transport par camion, et la date de son arrivée à Douvres avec celle du départ du bateau pour Montréal.

La nature « combinée » de la négociation prend tout son sens dans cet exemple : en effet tous les éléments négociés sont dépendants les uns des autres. Si on devient meneur dans une des négociations, il faut être sûr d'être gagnant dans les deux autres : il ne sert à rien d'avoir un transport en camion si le houblon n'est pas acheté ou si on n'a pas le transport par bateau. Dans ce cas il faut pouvoir annuler la transaction, ce qui implique souvent des coûts qu'il faut prendre en compte lorsqu'on décide de planifier une négociation combinée.

2.3. Les enchères

Selon Kumar et Feldman [Kumar and Feldman, 1998a] les enchères sont simplement une forme de négociation d'affaires. En d'autres termes, une enchère est toujours une négociation, mais une négociation n'est pas forcément une enchère.

Dans cette section nous allons étudier un cas particulier des négociation : les enchères. Nous commencerons par faire un historique des enchères, avant d'en donner une définition et de voir quelles sont celles qui sont les plus utilisées. Nous tenterons enfin de décrire ces dernières en détail.

2.3.1. Introduction : historique des enchères

On rapporte que la plus vieille enchère se serait tenue cinq siècles avant J.C. à Babylone. L'historien Hérodote, qui vécut en Asie mineure à cette époque, rapporte que des enchères se tenaient chaque année dans les villages, dans lesquelles des femmes étaient vendues à condition d'être épousées [Cassady, 1967]. Les « prix » pouvaient être très élevés pour les très jolies femmes et pouvaient également être « négatifs » pour les moins attrayantes: il s'agissait alors d'une dote accompagnant la femme...

Plus tard on retrouve les enchères d'esclaves [Shubik, 1983], dans la Grèce antique et chez les romains jusqu'aux débuts de l'ère chrétienne. L'empire romain lui-même fut mis aux enchères lorsque l'empereur Commode fut assassiné en l'an 192 ! Après la chute de Rome les enchères tombèrent plusieurs siècles dans l'oubli mais réapparurent en Europe au moyen-âge, pour la vente d'esclaves.

En 1556 en France apparaissent les Huissiers-Priseurs qui sont chargés de mettre en vente les biens des défunts ou des gens endettés. Les enchères étaient

ascendantes et se rapprochaient passablement du style d'enchères que nous connaissons aujourd'hui.

À la fin du 16^{ème} et au 17^{ème} siècle, en Hollande, les œuvres d'art sont vendues aux enchères, où le prix, fixé haut au départ, baissait jusqu'à ce qu'un acheteur se manifeste.

Au début du 17^{ème} siècle dans les monastères bouddhistes en Chine l'enchère était une des levées de fonds permises. Les effets personnels des moines décédés étaient vendus aux enchères. Par ailleurs c'est en Chine qu'a vu le jour l'enchère « poignée de mains » (*handshake auction*) [Cassady, 1967]: les acheteurs, disposés autour de l'encanteur, serraient à tour de rôle la main de celui-ci en prenant soin de cacher leurs mains sous un foulard et indiquaient le montant de leur mise par la pression du nombre de doigts correspondants.

En Angleterre à la fin du 17^{ème} siècle se tenaient des enchères *Inch Candle* ou encore *Sale by the Candle* [Cassady, 1967] pour tous les produits arrivant d'Inde. Il s'agissait d'une enchère ascendante dans laquelle on allumait une bougie d'un pouce; le dernier participant à miser avant que la bougie ne soit consumée devenait l'acquéreur. Les vaisseaux étaient également vendus de la même manière. Pendant ce temps aux États-Unis les enchères sont utilisées pour les produits agricoles, les animaux et les esclaves.

Au 18^{ème} siècle les enchères prennent de l'ampleur: on annonce dans les journaux et certains encanteurs ont pignon sur rue. Plusieurs maisons d'enchères deviennent très importantes, dont Sotheby's [Sotheby's, 2003] et Christie's [Christie's, 2003], qui existent encore de nos jours.

Puis ce fut l'arrivée d'Internet qui vint donner une évolution aux enchères en les démocratisant et les rendant accessibles à tous, pour toutes sortes de produits. En 1995 apparurent *OnSale* [OnSale, 2003], *Amazon* [Amazon, 2003] et *eBay* [eBay,

2003], ce dernier étant l'un des leaders dans le domaine des enchères en ligne. De nombreux autres sites d'enchères en lignes apparurent et il en existe aujourd'hui des centaines.

2.3.2. Objectifs et définition de l'enchère

Le mot enchère vient du latin *auctio* qui signifie « augmenter ». Bien que cette racine ne soit pas représentative de tous les types d'enchères (car comme nous le verrons plus loin, les prix peuvent augmenter ou diminuer), elle reflète le fait que beaucoup d'entre elles furent souvent basées sur des mises croissantes. À la question « pourquoi utiliser une enchère pour vendre un produit, plutôt qu'une autre méthode, telle la vente à prix fixe ? », Cassady [Cassady, 1967] donne une réponse :

« One answer is, perhaps, that some products have no standard value. For example, the price of any catch of fish (at least of fish destined for the fresh fish market) depends on the demand and supply conditions at a specific moment of time, influenced possibly by prospective market developments. For manuscripts and antiques, too, prices must be remade for each transaction. For example, how can one discover the worth of an original copy of Lincoln's Gettysburg Address except by auction method? »

On se sert des enchères pour vendre des biens dont la valeur est difficile à déterminer (une bouteille de vin, une œuvre d'art, une voiture usagée, etc.); c'est une manière pour les vendeurs d'obtenir l'assentiment général de la valeur de leurs biens. Le vendeur peut se servir de l'enchère lorsqu'il veut pouvoir tirer le maximum de sa vente lorsqu'il n'est pas sûr du prix à fixer.

Par ailleurs, vendre un bien aux enchères s'avère être plus aisé et plus rapide que de fixer un prix et d'avoir à négocier ensuite avec de potentiels acquéreurs. Dans une négociation les offres et contre-offres sont considérées séparément tandis que l'enchère permet la simultanéité des mises.

Si les vendeurs peuvent espérer tirer un bon prix de la vente de leurs biens, les acheteurs de leur côté peuvent espérer payer le moins cher possible pour ces

biens. Une particularité des enchères est le fait que ce n'est pas le vendeur qui fixe le prix de vente final, mais bien les acheteurs.

Nous retiendrons comme définition de l'enchère celle donnée par Wolfstetter [Wolfstetter, 1995] : c'est un mécanisme de mise, décrit par un ensemble de règles qui spécifient comment le gagnant est déterminé et combien il devra payer.

William Vickrey [Vickrey, 1961] a établi quatre types principaux d'enchères simples (*single-sided*, voir section 2.3.9): anglaise (voir section 2.3.3), hollandaise (voir section 2.3.4), par l'offre secrète au premier prix (*first-priced sealed-bid auction*, voir section 2.3.5) et au deuxième prix (encore appelée *Vickrey*, voir section 2.3.6). Il s'avère que ces quatre enchères sont les plus utilisées, d'après McAfee et McMillan [McAfee and McMillan, 1987]. Nous décrirons ces enchères simples, ainsi que l'enchère double, dans les sections qui suivent.

2.3.3. L'enchère anglaise

L'enchère *anglaise* (appelée également enchère *à la criée* ou encore enchère *à prix ascendant*) est sans doute l'enchère la plus connue du grand public. C'est d'ailleurs le type d'enchère que l'on retrouve le plus souvent sur Internet [Lucking-Reiley, 2000].

Nous présentons ici le déroulement de l'enchère tel qu'il est décrit par Milgrom [Milgrom, 1989]. L'enchère commence avec l'annonce par l'encanteur du prix de réserve, c'est à dire le prix le plus bas qui soit acceptable pour la vente du bien. Il appelle ensuite les acheteurs à faire des offres successivement supérieures les unes aux autres, jusqu'à ce que plus personne ne surenchérisse. Le bien est alors vendu à la personne ayant fait la mise la plus élevée, au prix misé .

L'encanteur peut décider de fixer l'incrément de la surenchère (c'est à dire la somme désirée pour surenchérir la mise en cours), d'en fixer un minimum ou encore peut laisser les acheteurs décider de la surenchère.

Une variante de cette enchère est l'*open-exit* [Lucking-Reiley, 2000] : les prix montent continuellement, mais les joueurs doivent annoncer publiquement qu'ils se retirent de l'enchère lorsque le prix devient trop élevé pour eux. À partir de ce moment ils ne peuvent plus réintégrer l'enchère et en sont définitivement exclus. Une autre variante est celle où l'encanteur propose des prix et où les offrants prêts à payer ce prix signalent leur intérêt.

En France une pratique assez particulière dans les enchères d'œuvres d'art est le pouvoir qu'a l'état de pouvoir acheter le bien après la dernière mise au prix adjugé [Rasmusen, 1989].

Dans une enchère anglaise traditionnelle tenue à la criée, on ne sait pas à l'avance combien de temps celle-ci va prendre, mais généralement elle ne dure jamais plus que quelques minutes, et quand plus personne ne fait de nouvelle mise, elle prend fin (« ...une fois, deux fois, trois fois... adjugé! »). Cependant dans les enchères anglaises tenues sur Internet les participants préfèrent avoir une date de fermeture de l'enchère fixée à l'avance [Lucking-Reiley, 2000]. Ceci pose un problème au niveau de l'incitatif à surenchérir: en effet pourquoi miser tôt quand on sait qu'il reste une semaine avant la fin de l'enchère? Une pratique, appelée *sniping* [Lucking-Reiley, 2000], [Roth and Ockenfels, 2002] s'est développée: elle consiste à attendre la toute dernière minute de l'enchère pour faire une mise légèrement supérieure à celle du meneur, qui n'a pas le temps de réagir.

Selon Lucking-Reiley [Lucking-Reiley, 2000], deux alternatives ont été développées pour encourager une participation hâtive. La première est de rajouter une courte période de prolongation. Par exemple si dans les cinq dernières minutes il y a encore de l'activité, l'enchère se prolongera cinq autres minutes. L'enchère ne

prend véritablement fin que lorsque aucune activité n'est détectée durant cinq minutes. Ceci évite que des *snipers* ne misent à une seconde de la fin. Cependant cette méthode a un désavantage, celui de contraindre un offrant à rester à l'affût jusqu'à ce que l'enchère prenne fin, et comme celle-ci peut être prolongée, on perd alors l'intérêt de la date de clôture fixe.

L'autre alternative est d'implanter un mécanisme de *proxy bidding* [Lucking-Reiley, 2000], [Roth and Ockenfels, 2002], c'est à dire un processus qui reste à l'affût des mises et qui surenchérit pour vous en ayant en mémoire le prix maximum que vous êtes prêt à payer et qu'il ne dépassera pas. Cependant, cette pratique fait ressembler l'enchère anglaise à une enchère Vickrey (voir section 2.3.6).

2.3.4. L'enchère hollandaise

L'enchère *hollandaise* (encore appelée *enchère à prix descendant* ou *enchère au rabais*) tire son nom du fait qu'aux Pays-Bas on utilise depuis longtemps cette méthode pour vendre des œuvres d'art et des fleurs [Cassady, 1967]. Nous reprenons la description de cette enchère telle que donnée par Milgrom [Milgrom, 1989] : l'encanteur fixe d'abord un prix de départ élevé pour le bien (c'est à dire généralement plus élevé qu'une évaluation que des experts pourraient en faire); il descend ensuite progressivement le prix jusqu'à ce qu'un acheteur intéressé se déclare; le bien est alors attribué à ce dernier au dernier prix formulé. Le même principe est appliqué lorsqu'un lot (plusieurs unités du même produit) est mis aux enchères: il y a plusieurs gagnants au fur et à mesure que le prix baisse, jusqu'à l'épuisement complet du stock. À noter que l'encanteur peut décider de se fixer un prix minimum (ou prix plancher) en dessous duquel il ne descendra pas; auquel cas l'enchère prendrait fin, sans qu'aucun gagnant n'ait été déclaré.

Ce type d'enchère est évidemment utilisé pour des biens périssables comme les fleurs, le poisson (comme par exemple au marché de Barcelone), et le tabac

[McAfee and McMillan, 1987]. Mais comme le soulignent Kumar et Feldman [Kumar and Feldman, 1998b], peuvent également être considérés comme biens « périssables » des légumes au même titre que des places sur un vol aérien. Bref, on a recours à l'enchère hollandaise quand on veut être sûr de se défaire d'une marchandise, et qu'on espère le faire le plus rapidement possible.

2.3.5. L'enchère par l'offre secrète au premier prix (*first-bid*)

Comme son nom l'indique, cette enchère a ceci de particulier pour les acheteurs qu'ils ne peuvent connaître les mises des autres joueurs. Le déroulement de l'enchère est décrit par McAfee et McMillan [McAfee and McMillan, 1987] comme suit: les acheteurs potentiels soumettent leurs mises de façon secrète et celui ayant formulé la plus haute mise est déclaré gagnant, et paye le bien au prix de la mise qu'il avait soumise.

Pour ce qui est des enchères en lot, il s'agit de dépouiller les mises et de ne garder que les n plus hautes (n correspondant au nombre d'unités du lot). Dans ce cas là, il y a plusieurs gagnants, mais ils ne payent pas tous le même prix. On dit alors de cette enchère qu'elle est *discriminatoire*.(voir section 2.3.9). Si par exemple 2 items sont à l'enchère et que les mises les plus hautes sont celles des joueurs A, B et C, respectivement de 50\$, 45\$ et 40\$, A et B seront déclarés gagnants, mais A payera 50\$ tandis que B ne payera que 45\$ pour le même bien.

2.3.6. L'enchère *Vickrey* (ou par l'offre secrète au second prix)

Cette enchère porte le nom de son créateur, William Vickrey, qui reçut le prix Nobel d'économie en 1996. Le principe de base est le même que celui de l'enchère au premier prix (voir section 2.3.5): tout le monde donne sa mise en ignorant celle des autres, puis on dépouille les mises. Le gagnant est le joueur ayant

misé le plus gros prix; cependant celui-ci ne paye pas le montant qu'il a misé, mais plutôt le deuxième plus haut montant misé parmi tous les joueurs [Vickrey, 1961]. Par exemple si les joueurs A et B ont soumis les deux plus hautes mises avec respectivement 50\$ et 45\$, le joueur A est déclaré gagnant et aura le bien pour 45\$.

Une différence importante avec l'enchère précédente est qu'en apparence on pourrait croire que le vendeur ferait moins de profit étant donné que le gagnant paye le bien moins cher que sa mise. Cependant, d'après Kumar et Feldman [Kumar and Feldman, 1998b] chaque joueur est porté à miser un montant un peu plus élevé que sa propre évaluation, sachant que, si jamais il gagne, il ne payera pas le bien au prix de sa mise, mais à un prix moindre, soit celui du montant de la mise de son plus proche adversaire.

Pour ce qui est de l'enchère en lot, tous les gagnants payent le même prix pour leur bien (contrairement à l'enchère au premier prix). De ce fait l'enchère est dite *uniforme* (voir section 2.3.9). Le prix auquel le bien est adjugé est alors celui misé par le premier perdant. Par exemple si 2 items sont en vente et que les joueurs A, B et C misent respectivement 50\$, 45\$ et 40\$, A et B seront les gagnants et paieront tous deux 40\$.

2.3.7. L'enchère double (*double auction*)

Les quatre enchères présentées dans les sections précédentes sont de type simple (voir section 2.3.9), c'est à dire qu'il n'y a qu'un seul vendeur pour plusieurs acheteurs (il pourrait tout aussi bien s'agir d'un seul acheteur pour plusieurs vendeurs). Dans l'enchère double, que nous décrivons dans cette section, inspirés par Friedman et Rust [Friedman and Rust, 1993], on retrouve plusieurs vendeurs et plusieurs acheteurs.

Cette enchère est utilisée depuis longtemps dans les institutions financières aux États-Unis. Il en existe de deux types: *continues* et *séquentielles*.

Lorsqu'une enchère double est *continue*, cela signifie que dès qu'un acheteur et un vendeur sont compatibles, le marché est conclu et l'enchère se poursuit. On retrouve ce genre d'enchère à la bourse, où acheteurs et vendeurs échangent des actions en quelques signes. Lorsque l'enchère double est *séquentielle*, vendeurs et acheteurs soumettent leurs offres/mises à l'intérieur d'intervalles de temps définis. À la fin de chaque intervalle on procède alors au couplage entre acheteurs et vendeurs compatibles.

Il existe plusieurs façon de faire des couplages entre vendeurs et acheteurs. On peut décider d'associer tout simplement acheteurs et vendeurs par leur prix de réserve respectivement maximal et minimal. Par exemple si deux vendeurs v_1 et v_2 soumettent des prix de réserve minimums respectivement de 40\$ et 50\$ et deux acheteurs a_1 et a_2 , quant à eux, ont des prix de réserve maximums de 40\$ et 50\$, v_1 sera couplé à a_1 , et v_2 à a_2 .

Une autre manière de coupler acheteurs et vendeurs peut être basée sur la maximisation des gains. On classe les vendeurs par ordre croissant de prix d'offre et les acheteurs par ordre décroissant de mise proposée. On fait correspondre alors le vendeur offrant le plus bas prix avec l'acheteur offrant la plus haute mise, le deuxième plus bas vendeur avec le deuxième plus haut acheteur, et ainsi de suite jusqu'à ce qu'on ne puisse plus faire de couplage, c'est à dire lorsque les mises des acheteurs sont plus basses que les prix offerts par les vendeurs. Le prix de vente est alors la moyenne entre les deux. Pour illustrer cette méthode, imaginons trois vendeurs v_1 , v_2 , et v_3 faisant des offres minimales pour leurs produits respectivement à 30\$, 40\$ et 50\$, et trois acheteurs a_1 , a_2 et a_3 proposant respectivement comme prix maximal 70\$, 50\$ et 40\$. On aurait alors deux couplages: v_1 et a_1 pour 50\$ (moyenne de 30+70), et v_2 et a_2 pour 45\$ (moyenne de 40+50). On ne peut pas coupler v_3 avec a_3 car ce dernier propose un prix inférieur à l'offre minimale qu'a faite v_3 . Par cette méthode les profits globaux des vendeurs sont maximisés (25\$ en tout : 20\$ pour v_1 et 5\$ pour v_2) et les dépenses globales des acheteurs sont minimisées (25\$ en tout : 20\$ pour a_1 et 5\$ pour a_2).

2.3.8. Récapitulatif des enchères

Dans cette section nous avons présenté différents types d'enchères les plus répandus. Le Tableau 2.1, tiré des auteurs Anthony et Jennings [Anthony and Jennings, 2002], présente un résumé des règles de mises et d'adjudication pour ces enchères.

Enchère	Règles de mise	Règles de fin et d'adjudication	Unités
Anglaise	Le vendeur annonce un prix de départ et les participants sont libres de surenchérir de manière successive.	L'expiration de la période de mise, ou lorsque plus personne ne soumet de nouvelle mise. Le gagnant est le participant qui a soumis la dernière mise.	Simple
Hollandaise	Le vendeur annonce un prix de départ très élevé, et le baisse progressivement.	Lorsqu'un participant réclame le bien au prix courant, ou lorsque le prix plancher du vendeur est atteint.	Simple ou Multiples
Offre Secrète au premier prix	Les participants soumettent leurs mises sans aucune connaissance de celles des autres.	L'expiration de la période de mise. Le gagnant est le participant ayant donnée la plus haute mise; il paye le bien à ce prix.	Simple ou Multiples
Vickrey	Les participants soumettent leurs mises sans aucune connaissance de celles des autres.	L'expiration de la période de mise. Le gagnant est le participant ayant donnée la plus haute mise; il paye le bien au prix de la deuxième plus haute mise.	Simple ou Multiples
Enchère double	Vendeurs et acheteurs soumettent leurs offres et leurs mises.	Un marché est conclu lorsqu'il y a un couplage entre une offre et une mise. Les gagnants sont les vendeurs et acheteurs impliqués dans la transaction.	Multiples

Tableau 2.1 – Mécanismes de différents types d'enchères [Anthony and Jennings, 2002]

2.3.9. Classification des enchères

Après avoir donné une définition des enchères et décrit celles qui sont les plus utilisées, nous tenterons dans cette section de classer les enchères.

Pour ce faire, nous nous sommes inspirés des classifications que proposent Kumar et Feldman [Kumar and Feldman, 1998a], [Kumar and Feldman, 1998b]. Selon eux, il existe deux grandes classes d'attributs :

- les **attributs relatifs au type d'enchère** qui permettent de définir le type de l'enchère (anglaise, hollandaise, etc.).
- les **attributs de l'enchère**, qui donnent des précisions sur le déroulement de l'enchère (la mise minimale de départ, l'incrément, etc.).

Nous présentons plus en détails ces attributs dans les deux prochaines sous-sections.

2.3.9.1. Les attributs relatifs au type d'enchère

Ces attributs permettent de définir le type d'une enchère, c'est à dire le mécanisme intrinsèque de celle-ci. On retrouve les attributs suivants :

- 1) Le **côté du marché**. L'enchère peut être de type:
 - **simple** (*single-sided*). Lorsqu'une enchère est simple, seulement un côté du marché participe à l'enchère: soit il s'agit des vendeurs (on a alors un acheteur pour plusieurs vendeurs) ou alors il s'agit des acheteurs (on a alors un vendeur pour plusieurs acheteurs).

- **double** (*double-sided*). Lorsqu'une enchère est double, on a plusieurs vendeurs et plusieurs acheteurs (mais pas forcément autant de vendeurs que d'acheteurs) qui participent à l'enchère.

2) Le **format d'interaction**. L'enchère peut être de deux types :

- **continue**, comme par exemple dans les enchères dites à la criée (*open outcry auctions*) où tout le monde connaît la mise au moment où celle-ci est faite et où chaque participant est libre de faire une contre-proposition à tout moment et où un marché peut être conclut sur-le-champ.
- **séquentielle** (*periodic*) : l'adjudication du ou des biens se fait seulement à la fin d'une certaine période, comme par exemple dans les enchères secrètes (*sealed-bid auctions*) où chaque participant doit soumettre sa mise dans un temps donné.

3) Le **contrôle des mises**. À ce niveau, une enchère peut être de deux types :

- **ouverte** (*open-bidding*) : dans ce cas, les acheteurs soumettent eux-mêmes le prix de la surenchère à l'encanteur, et le font à tout moment.
- **cadencée** (*clock-auction*) : c'est l'encanteur qui fixe la nouvelle valeur de la surenchère à sa discrétion au regard de l'état et de l'évolution de l'enchère.

4) La **progression des mises** d'une enchère peut être de deux types :

- **croissante** (*ascending*) comme l'enchère anglaise.

- **décroissante** (*descending*) comme dans l'enchère hollandaise.

5) La **politique de fixation des prix**. Une enchère peut être de deux types:

- **discriminatoire** : dans cette enchère (appelée également *Yankee Auction*), les participants qui ont été déclarés gagnants payent les biens ou services qui leur sont adjugés au prix qu'ils ont offert (comme par exemple dans l'enchère par l'offre secrète au premier prix).
- **uniforme** : dans ce cas ci, les gagnants payent tous le même prix, qui peut être moindre que le prix qu'ils avaient soumis, par exemple la plus basse des mises gagnantes ou encore la plus haute mise perdante (comme dans l'enchère Vickrey).

2.3.9.2. Les attributs de l'enchère

Les attributs d'une enchère sont des caractéristiques qui ne définissent pas le type d'enchère, mais qui paramètrent plutôt leur exécution. On retrouve les éléments suivants :

- 1) La **mise minimale de départ** (*minimum starting bid*).
- 2) L'**incrément minimum** sur les mises (*minimum bid increment*). Il peut être :
 - **fixé** au départ. Si aucun incrément minimum n'est requis, il sera alors fixé à 0.
 - **proportionnel** à la mise meneuse.
- 3) Le **nombre d'unités** d'un produit mis en vente.

4) La **règle de clôture** de l'enchère. Celle-ci prend fin :

- suite à l'**expiration d'une date de clôture**. L'enchère peut durer plusieurs heures, plusieurs jours et même plusieurs semaines, mais la date de la fin de l'enchère sera fixée et connue de tous. C'est une règle qui est généralement utilisée pour les enchères sur Internet.
- suite à une **inactivité des participants**. Dans ce cas de figure l'enchère peut durer tant que quelqu'un fait une offre supérieure à celle en cours.
- suite à l'**épuiement des stocks**
- suite à la **chute du prix en deçà d'un prix plancher** (comme par exemple dans l'enchère hollandaise)
- suite à la **combinaison** de plusieurs occurrences ci-dessus.

2.4. Résumé

Dans ce chapitre, nous avons tout d'abord défini ce qu'est une négociation et avons plus particulièrement étudié le cas des enchères. Nous avons ensuite décrit les plus répandues et enfin déterminé les caractéristiques et obtenu une classification de celles-ci. Cette classification nous permettra de modéliser de manière générique plusieurs types d'enchères et nous verrons comment, dans le chapitre suivant.

3. Modélisations des négociations

3.1. Introduction

Dans le chapitre précédent, nous avons défini ce que sont les négociations et les enchères. Nous avons vu quelques exemples d'enchères fréquemment utilisées et en avons établi une classification. Maintenant, nous désirons étudier d'un peu plus près le déroulement de plusieurs types de négociations du point de vue d'un participant, en les décomposant en une suite d'événements et de changements d'états, et chercherons à ne garder que les états essentiels.

Dans cette section, nous commençons par présenter une méthode de modélisation de négociations, puis nous nous en inspirons afin de modéliser les négociations que nous jugeons pertinentes pour notre travail d'analyse. Finalement, nous dressons une liste des états dans lesquels le participant peut se retrouver et regroupons ces états pour arriver à une liste d'états essentiels.

3.2. Modélisation à l'aide de FSM

Selon Kumar et Feldman [Kumar and Feldman, 1998a], on peut représenter une négociation par un automate à états finis (en anglais, *finite state machine*, ou FSM). Les états de l'automate représentent ceux dans lesquels la négociation peut se trouver selon les événements qui peuvent survenir au cours de celle-ci.

Un exemple est donné à la Figure 3.1 [Kumar and Feldman, 1998a] qui représente une vente à prix fixe, la forme de négociation la plus simple. Dans la figure, on retrouve les états *DT* (pour *deal template* – le modèle de la négociation), *Offer* (l'offre faite), *NA* (pour *negotiation aborted* – la négociation est abandonnée) et finalement *D* (pour *deal* - marché conclu). Quant aux événements,

on retrouve d'abord *<Seller, Offer to sell>* qui indique que le vendeur propose une offre de vente, *<Buyer, Accept>* où l'acheteur accepte l'offre proposée et finalement *<Seller, withdraw Offer>* où le vendeur se rétracte et abandonne la négociation.

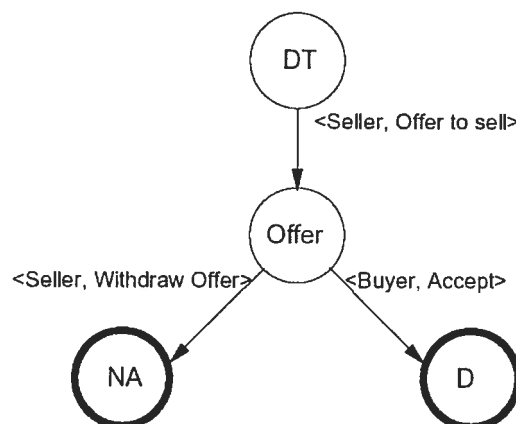


Figure 3.1 - FSM d'un marché à prix fixe [Kumar and Feldman, 1998a]

Cet exemple nous montre comment il est possible de représenter une négociation dans son ensemble, c'est à dire autant du point de vue de l'encanteur que du participant. Notre objectif, quant à lui, est de représenter la négociation uniquement du point de vue du participant. En effet, nous cherchons à retrouver les états communs pour le participant dans plusieurs négociations afin de créer un modèle générique. Dans les sections suivantes, nous nous emploierons à modéliser plusieurs types d'enchères et nous ferons une synthèse des états que nous aurons réussi à dégager.

3.3. Etude de cinq marchés

Dans cette section, nous prendrons cinq négociations différentes dont nous exprimerons le déroulement à l'aide de diagrammes d'états *du point de vue de*

l'acheteur. Cette étude nous permettra ensuite, comme nous le verrons dans la section 3.4, de dégager les états communs du participant dans tous ces marchés. C'est à partir de ces états que nous pourrons envisager la création d'un modèle générique de négociation, comme nous le verrons plus loin dans ce mémoire.

Pour notre étude, nous avons pris les cinq marchés suivants, qui sont représentatifs de la variété des négociations (prix négociable, contre-proposition, offre finale) et des enchères utilisées (voir section 2.3.9), tant au niveau de la progression des mises (croissante ou décroissante) que du format d'interaction (continu ou séquentiel) ou encore du contrôle des mises (ouvert ou cadencé) :

- enchère anglaise à une unité
- enchère hollandaise à une seule unité
- enchère hollandaise multi-items
- enchère par l'offre secrète (*sealed bid auction*)
- négociation à deux pairs (*two parties negotiation*)

Dans les prochaines sous-sections, nous expliquerons en détail nos représentations de ces marchés. Puis nous en ferons l'analyse en section 3.4.

3.3.1. Enchère anglaise

Nous avons modélisé l'enchère anglaise (voir Figure 3.2) à partir des événements possibles lors de cette négociation :

1. le participant s'enregistre.
2. le participant soumet une mise qui est acceptée par l'encanteur : il devient le meneur (*leader*).
3. le participant soumet une mise qui n'est pas retenue par l'encanteur.

4. un autre participant soumet une offre plus élevée que celle du meneur, celui-ci n'est donc plus *leader*.
5. l'enchère est close : si le participant n'était pas meneur à la fermeture, il a perdu ; si au contraire il était meneur à la fermeture, c'est le gagnant.
6. le prix de réserve du participant ne lui permet pas de surenchérir le prix du meneur, il doit donc abandonner l'enchère.
7. le participant quitte l'enchère.

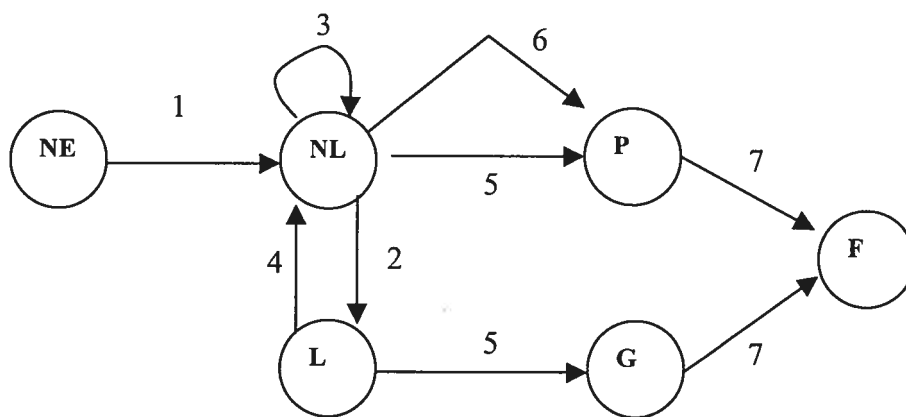


Figure 3.2 - Représentation de l'enchère anglaise à une unité

Voici les explications des symboles utilisés dans la Figure 3.2 :

- F** : fin de la participation à l'enchère
- G** : gagnant
- L** : *leader*
- NE** : non enregistré
- NL** : non *leader*
- P** : perdant

3.3.2. Enchère hollandaise à une seule unité

Nous avons modélisé l'enchère hollandaise à une seule unité (voir Figure 3.3) à partir des événements possibles lors de cette négociation :

1. le participant s'enregistre.
2. une nouvelle ronde commence à un nouveau prix si aucun participant n'a émis d'ordre et que le compteur est échu ou que l'encanteur décide de baisser le prix.
3. le participant déclare accepter le prix proposé par l'encanteur ; l'enchère est alors terminée et il reçoit une adjudication.
4. l'enchère est fermée, soit parce qu'un autre participant a accepté le prix proposé, soit parce que le prix planché a été atteint avant que quiconque ne soumette une mise.
5. on quitte l'enchère.

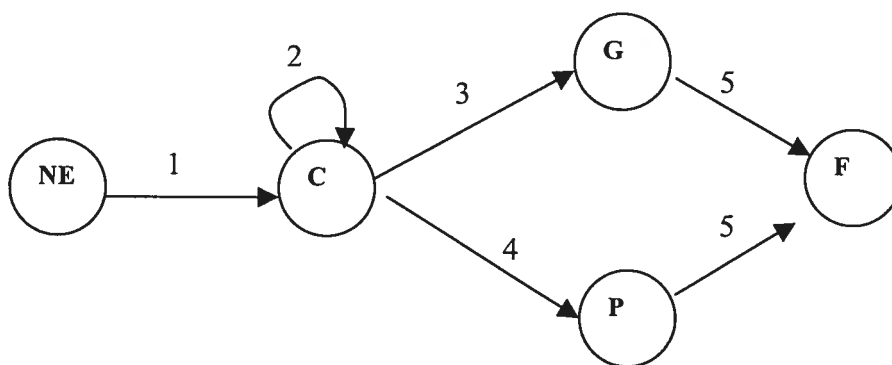


Figure 3.3 - Représentation de l'enchère hollandaise à une unité

Voici la signification des symboles utilisés dans la Figure 3.3:

- C** : choix à faire
F : fin de la participation à l'enchère
G : gagnant

NE : non enregistré

P : perdant

3.3.3. Enchère hollandaise multi-items

Ce sont essentiellement les mêmes éléments que ceux que l'on retrouve dans l'enchère hollandaise à une seule unité, à la différence que l'enchère n'est pas forcément terminée lorsqu'un participant se déclare preneur. Nous émettons les hypothèses suivantes quant au déroulement de cette enchère:

- le participant qui se déclare preneur a tout le loisir de décider combien d'unités du produit il veut se procurer, en autant qu'il en reste suffisamment en stock.
- le participant a qui l'on a adjugé un ou plusieurs items n'est pas forcé de quitter l'enchère ; il peut très bien se déclarer preneur une nouvelle fois.

Pour simplement donner un exemple des possibilités pour l'acheteur, si celui-ci a un maximum de 400\$ à dépenser et désire se procurer 5 unités, il pourrait commencer par acheter 3 unités à 100\$ et attendre que le prix baisse pour acheter 2 autres unités à 50\$ chacune. Il pourrait tout aussi bien décider d'acheter d'un coup 5 unités à 80\$ et se retirer tout de suite après. Bien d'autres possibilités lui sont offertes, que nous n'aborderons pas ici étant donné que la problématique des stratégies n'entre pas dans le cadre de ce travail.

Voici les événements possibles relatifs à cette enchère (voir Figure 3.4) :

1. le participant s'enregistre.
2. un autre participant s'est déclaré preneur, ou encore le compteur a échu ; la ronde s'est terminée et une autre est commencée à un nouveau prix.

3. le participant se déclare preneur et l'encanteur lui a adjudgé le nombre d'items demandés.
4. le participant n'a pas rencontré la totalité de ses besoins en gagnant la ronde et retourne à l'état compétitif.
5. le participant a rencontré la totalité de ses besoins avec l'adjudication qu'il a reçue.
6. le participant quitte l'enchère.
7. l'enchère se termine avant que le participant n'ait pu rencontrer ses besoins, ou encore le participant quitte volontairement l'enchère, faute d'unités restantes suffisantes pour qu'il rencontre ses besoins.

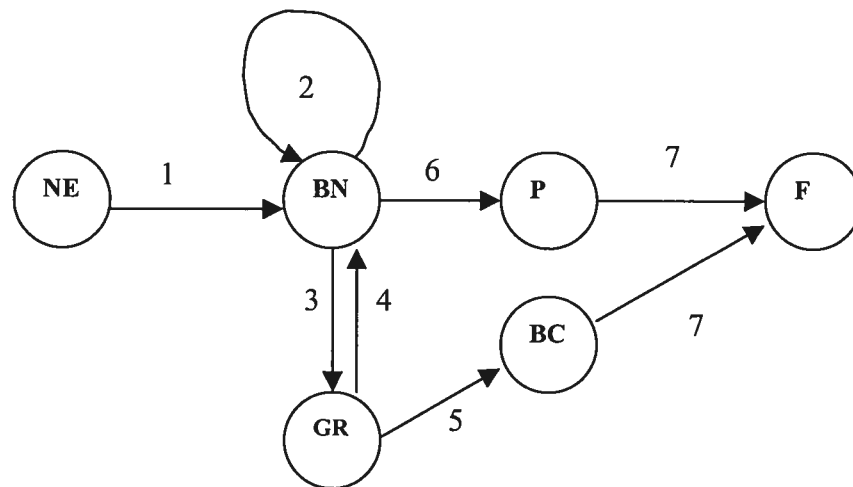


Figure 3.4 - Représentation de l'enchère hollandaise à n unités.

Voici la signification des symboles utilisés dans la Figure 3.4:

- BC** : besoins comblés
- BN** : besoins non comblés
- F** : fin de la participation à l'enchère
- GR** : gagnant de la ronde
- NE** : non enregistré
- P** : perdant

3.3.4. Enchère par l'offre secrète (*sealed-bid auction*)

Voici les événements possibles lors du déroulement de l'enchère (voir Figure 3.5) :

1. le participant s'enregistre.
2. le participant émet une mise et attend que la ronde se termine.
3. le participant a été déclaré gagnant et reçoit une adjudication.
4. le participant est déclaré perdant.
5. le participant quitte la négociation.

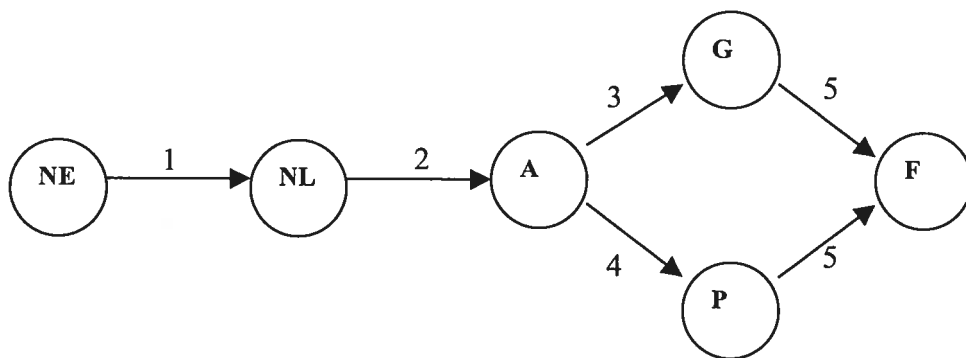


Figure 3.5 - Représentation de l'enchère par l'offre secrète

Voici la signification des symboles utilisés dans la Figure 3.5) :

- A** : en attente
F : fin de la participation à l'enchère
G : gagnant
NE : non enregistré
NL : non *leader*
P : perdant

3.3.5. La négociation à deux pairs (*Two parties negotiation*)

Cette négociation est décrite par Kumar et Feldman [Kumar and Feldman, 1998a] comme suit : un acheteur, qui propose un prix de départ, et un vendeur marchandent un produit. Ils peuvent successivement soumettre à l'autre une offre, en spécifiant si celle-ci est finale ou non. Si l'offre est finale, l'autre participant n'a d'autre choix que d'accepter ou de refuser. Si l'offre n'est pas finale, l'autre participant peut soumettre une contre-offre, elle-même finale ou négociable. La Figure 3.6 représente cette négociation du point de vue du participant acheteur.

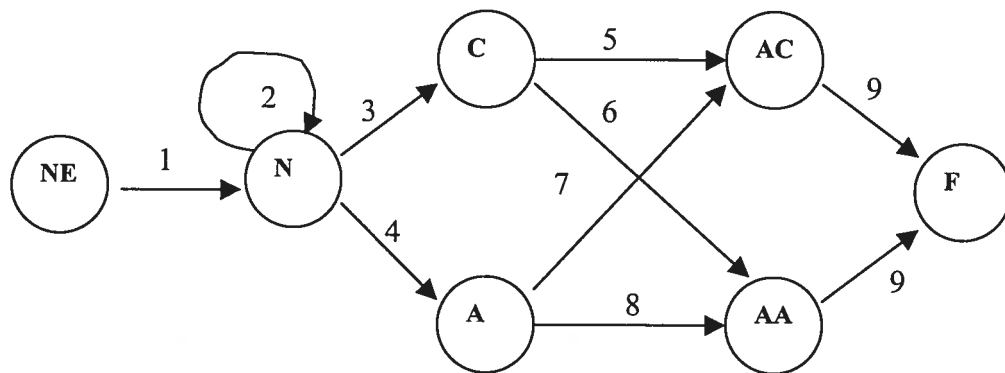


Figure 3.6 - Représentation de la négociation à deux pairs

1. le participant s'enregistre auprès du vendeur.
2. le participant et le vendeur s'échangent des offres.
3. le vendeur soumet une offre finale au participant.
4. le participant soumet une offre finale au vendeur.
5. le participant accepte l'offre finale du vendeur.
6. le participant refuse l'offre finale du vendeur.
7. le vendeur accepte l'offre finale du participant.
8. le vendeur décline l'offre finale du participant.
9. la négociation prend fin.

Voici la signification des symboles utilisés dans la Figure 3.6 :

- A** : en attente
- AA** : aucun accord
- AC** : accord conclu
- C** : choix à faire
- F** : fin de la participation à l'enchère
- NE** : non enregistré

3.4. Analyse des états des automates

Après avoir représenté cinq négociations différentes à l'aide d'automates à états finis, nous présentons, dans cette section, les états communs dans lesquels le participant peut se retrouver et que nous avons dégagés.

L'étude des diagrammes nous a permis de dégager les états que nous jugeons essentiels. Nous les énumérons ici ces états, en expliquant nos choix :

- **non-enregistré** : c'est l'état trivial de départ, l'agent devant d'abord se connecter au serveur de négociation et s'enregistrer.
- **final** : c'est aussi un état trivial, puisque il s'agit de la fin de la participation à l'enchère.
- **gagnant** : il regroupe les états *besoins comblés*, (voir section 3.3.3) puisqu'on a réussi à obtenir tous les biens que l'on voulait, *accord conclu* (voir 3.3.5) et bien sûr *gagnant*.
- **perdant** : il regroupe aussi *aucun accord* (voir section 3.3.5).

- **leader** : bien qu'on ne le retrouve que dans notre exemple de l'enchère anglaise (voir section 3.3.1) il s'agit d'un état important dans le cas d'une enchère ouverte. Le fait d'être ou de ne pas être *leader* influence le comportement du participant, qui doit être à l'affût des mises des autres participants.

- **non-leader** : il peut aussi bien englober l'état *besoins non-comblés* (voir section 3.3.3) *en négociation* (voir section 3.3.5), *choix à faire* (voir section 3.3.5) que *non-leader* (voir section 3.3.1) et *en attente* (voir section 3.3.4 et 3.3.5). Pour ce qui est de l'état *besoins non-comblés* (voir section 3.3.3) le participant retourne dans une phase de compétition, il n'est donc pas en position de meneur. Pour *en négociation*, le participant n'a pas de concurrent puisqu'il est seul à négocier avec l'encanteur. Cependant il est dans une situation dans laquelle il n'est pas meneur, car rien ne lui garantit qu'il peut gagner. Pour le cas bien particulier de *choix à faire* (voir section 3.3.5), on peut dire encore une fois que tant que rien n'est sûr le participant n'est pas dans une position de meneur. C'est après le choix qu'il aura fait seulement qu'on pourra vraiment dire dans quelle position se retrouve l'agent. Finalement, *en attente* désigne plutôt l'état non-leader en attente d'un événement (une adjudication ou un refus).

- **gagnant de ronde** : il s'agit d'un état totalement transitoire. En effet, le participant ne fait que constater qu'on lui a adjugé un certain nombre d'unités, mais il faut qu'il fasse un calcul de ses avoirs afin de savoir dans quel état il va devoir passer, à savoir *gagnant* ou *besoins non-comblés*.

3.5. Résumé

Dans ce chapitre, nous avons utilisé un modèle de représentation de négociation basé sur les automates à états finis, afin de décrire cinq négociations différentes. Notre objectif était d'en extraire les états essentiels, ce que nous avons fait après avoir analysé les différents diagrammes et en avoir cherché les points communs. Connaissant ces états essentiels, nous pourrions plus facilement concevoir un modèle générique afin de pouvoir participer à des types de négociations différentes.

Dans le prochain chapitre nous étudierons une technologie adaptée à la réalisation de l'automatisation de la participation à une négociation, soit la technologie des agents.

4. Les agents

4.1. Introduction

Depuis quelques années, de nombreux chercheurs préconisent l'utilisation des agents pour le commerce électronique. En effet, la plupart des achats ne sont pas encore automatisés et l'intervention humaine est très présente à presque chaque étape du processus d'achat, comme par exemple la recherche de produits et la découverte et l'identification de marchands potentiels. La technologie agent est de plus en plus pressentie pour automatiser des tâches complexes ou faisant perdre beaucoup de temps aux acheteurs potentiels, tout en minimisant les interactions de ceux-ci et en y alliant une rapidité de réaction. Les qualités des agents, comme par exemple l'autonomie et la persistance, en font des candidats de choix pour l'optimisation du processus d'achat. Comme nous l'avons vu en section 1.2.2, le projet Consensus est un système basé sur les agents afin d'assurer les processus de négociation automatique.

Dans ce chapitre, nous présentons la technologie agent et en étudions plusieurs architectures, ce qui nous permettra par la suite de faire un choix éclairé pour la conception de notre modèle, dont nous parlerons au chapitre suivant.

4.2. La technologie agent

Le terme *agent* est utilisé de plus en plus depuis quelques années. Mais c'est un concept qui ne date pas d'hier. Dans cette section, nous allons présenter cette technologie en la resituant dans un contexte historique, puis nous en donnerons plusieurs définitions et finalement nous nous pencherons sur ses propriétés. Nous présenterons par la suite, dans la section 4.3, les architectures les plus connues.

4.2.1. Historique de la recherche sur les agents

La recherche sur les agents provient de plusieurs champs d'étude, dont notamment l'intelligence artificielle (IA). Une des branches de l'IA, la planification (*planning research*), a été l'un des domaines de recherche qui s'est beaucoup intéressé aux agents dès le début des années soixante-dix. L'objectif de la planification est d'établir les actions et leur ordre d'application avant leur exécution, afin d'atteindre un but donné. Un des plus anciens systèmes de planification connu est le STRIPS (*Stanford Research Institute Problem Solver*) [Fikes and Nilsson, 1971], dont nous reparlerons dans la section 4.3.1.

Comme nous le rappelle Chaib-draa [Chaib-draa et al., 2001], une autre branche de l'IA, l'intelligence artificielle *distribuée*, s'intéressa également aux agents, à partir des motivations suivantes: d'abord la modularité, c'est à dire réduire la complexité, ensuite la vitesse, la fiabilité et la flexibilité que cela pouvait emmener.

L'IA donna également naissance aux systèmes experts [Wooldridge, 1999], qui sont capables de résoudre des problèmes ou de donner un avis sur un domaine de connaissance, et qui ont longtemps été comparés avec les agents.

Les domaines de la programmation orientée-objet et des systèmes objets concurrents s'intéressèrent au concept d'agent. En effet, la notion d'objet suscita un intérêt pour la notion d'autonomie, et certains pensent qu'objet et agent sont une seule et même chose (nous reviendrons sur ce point en section 4.2.4).

Finalement, un autre domaine qui accorda de l'importance à la réflexion sur les agents fut celui de la conception d'interfaces humain-machine [Jennings et al., 1998]. En effet, on voulait que certains programmes informatiques prennent des initiatives plutôt que d'attendre que l'utilisateur leur dise quoi faire. En un mot on voulait des genres de domestiques informatiques, que l'on nomme assistant-experts. Cela donna naissance à des gestionnaires de courrier, des fureteurs Web actifs, etc.

4.2.2. Définitions

Après ce bref historique, abordons la question qui nous préoccupe: qu'est-ce qu'un agent? Il existe plusieurs définitions du concept agent. Nous en avons retenus deux qui nous semblent résumer les idées les plus importantes.

La première définition que nous avons retenue est celle de Russel et Norvig [Russell and Norvig, 1995] :

« An *agent* is anything that can be viewed as *perceiving* its environment through *sensors* and *acting* upon that environment through *effectors*. »

Jennings [Jennings et al., 1998], quant à lui, explicite davantage les concepts de la précédente définition, soient la perception d'un environnement et l'action dans le but de le modifier, mais va plus loin et propose cette définition¹:

« Un *agent* est un système informatique, *situé* dans un certain environnement et capable d'agir de manière *autonome* et *flexible* afin d'atteindre ses objectifs de conception. Par *situé*, on entend par là que l'agent reçoit des données de son environnement et qu'il peut agir afin de modifier ce dernier d'une quelconque façon. Par *autonomie*, on veut simplement dire que le système doit être capable d'agir sans intervention humaine et avoir le contrôle sur ses propres actions et états internes. Enfin, par *flexible*, on entend trois choses: premièrement *réactif*, c'est à dire qui peut percevoir son environnement et réagir rapidement aux changements de celui-ci, deuxièmement *pro-actif*, c'est à dire qu'il ne se contente pas uniquement de réagir, mais prend des initiatives orientées vers l'atteinte d'un but précis, et troisièmement *sociable*, à savoir capable d'interagir avec d'autres agents ou êtres humains afin de résoudre ses problèmes ou aider les autres à résoudre les leurs. »

¹ Nous avons traduit de l'anglais cette section du texte original.

4.2.3. Propriétés d'un agent

Après avoir donné une définition de l'agent, nous tenterons maintenant d'en cerner les propriétés. En effet, un agent peut posséder plusieurs propriétés, qui peuvent être combinées; parmi toutes celles qui existent, nous en avons retenu la liste ci-après [OMG, 2000] :

Autonomie : un agent peut être capable d'agir sans intervention extérieure. Il possède un certain degré de contrôle sur son état interne.

Interaction : la faculté de pouvoir communiquer avec son environnement et/ou d'autres agents.

Adaptation : l'agent est sensible, jusqu'à un certain niveau, à son environnement et/ou d'autres agents. Dans ses niveaux les plus élevés d'adaptation, l'agent peut modifier son état interne à la lumière de son expérience.

Sociabilité : l'agent peut avoir des interactions cordiales, voire amicales, avec d'autres agents.

Mobilité : un agent peut être *mobile*, donc passer d'un environnement à un autre

Pro-activité : dans ce cas-ci, le comportement de l'agent est axé sur la réalisation d'un but et ne réagit pas simplement à son environnement

Intelligence : l'état de l'agent est basé sur la connaissance, et il interagit avec d'autres agent par le biais d'un langage symbolique.

Coordination : lorsqu'il est coordonné par des plans, des *workflows* ou d'autres systèmes de gestion de processus, l'agent est capable de réaliser des activités dans un environnement partagé avec d'autres agents.

Collaboration (ou coopération) : l'agent est capable de coopérer avec d'autres agents pour atteindre un objectif commun. Les agents collaboratifs réussissent ou échouent ensemble.

Compétition : l'agent est capable de coopérer avec d'autres agents, comme dans le cas de la collaboration, mais son succès implique l'échec des autres.

4.2.4. Agent versus Objet

En programmation orientée objet, il arrive fréquemment que les programmeurs confondent objet et agent, car en les comparant on s'aperçoit qu'il existe bien des similitudes: les objets contiennent un état, peuvent accomplir des actions ou méthodes sur ces états et également communiquer par envoi de messages.

Mais il y a des différences entre agent et objet, et en particulier les trois points suivants, selon Jennings [Jennings et al., 1998] : l'autonomie, la flexibilité et finalement la notion de *thread*.

Parlons d'abord du degré d'autonomie. Si une méthode d'un objet est disponible pour que d'autres objets puissent l'invoquer, alors on peut dire que cet objet n'a aucun contrôle sur son comportement puisqu'il ne peut pas décider de laisser la méthode s'exécuter ou non. Bien sûr on peut restreindre l'exécution de certaines méthodes, mais on devra nécessairement en laisser qui seront exécutables par d'autres, sinon il serait impossible de bâtir un système. La différence avec l'agent est que celui-ci doit avoir d'abord le consentement de l'autre agent pour que la méthode puisse s'exécuter. L'agent ne donnera l'autorisation que si cela est dans

son intérêt, si cela peut l'aider à atteindre le but pour lequel il a été investi. La distinction est imagée de la façon suivante: « *Objects do it for free; agents do it for money* » [Jennings et al., 1998].

Ensuite il y a la question de la flexibilité. Cette notion regroupe les trois éléments vus dans la définition donnée en section 4.2.2, soient la réactivité, la proactivité et la sociabilité. On constate rapidement que le concept objet ne cadre pas avec cette définition de flexibilité. On peut bien sûr essayer d'intégrer ces propriétés dans un objet mais par défaut elles ne font pas partie de l'objet.

L'autre point qui distingue les agents des objets est le fait qu'à chaque agent est associé un seul *thread*, ce qui n'est pas systématiquement le cas pour le modèle objet, même si beaucoup d'efforts ont été mis ces derniers temps pour introduire la concurrence dans la programmation objet, comme par exemple le *multi-threading* en *Java*.

Enfin, il est important de préciser que, même si un agent peut être un objet, l'inverse n'est pas forcément vrai.

4.3. Architectures d'agents

Dans la section précédente, nous avons présenté la technologie agent : nous en avons retracé l'historique, donné une définition et présenté les propriétés que ceux-ci peuvent posséder. Dans cette section, nous présenterons quatre grandes classes d'architectures d'agents, telles que définie par Wooldridge [Wooldridge, 1999], et que nous énumérons ici:

Architecture délibérative (*logic based agents*) où les prises de décision sont réalisées à partir de déductions logiques.

Architecture réactive, où les décisions sont prises en faisant une mise en correspondance entre une situation et une action.

Architecture BDI (*belief-desire-intention agents*) où les décisions se prennent en regard des croyances, des désirs et des intentions.

Architecture en couches ou hybride où la prise de décision s'effectue à travers différentes couches logicielles, qui sont des raisonnements sur l'environnement à différents niveaux d'abstraction.

Nous verrons de quelle manière un agent peut fonctionner à partir de ces différentes architectures et nous tenterons, un peu plus loin dans ce travail, de choisir parmi ces architecture celle qui sera la mieux adaptée à notre problématique.

4.3.1. Architecture délibérative

L'architecture délibérative, approche classique en matière d'agents, découle directement des concepts de l'IA symbolique. Selon Wooldridge et Jennings [Wooldridge and Jennings, 1995], un agent délibératif est un agent qui contient un modèle symbolique représentant explicitement le monde, et qui prend des décisions (par exemple sur quelle action accomplir) à partir de raisonnements logiques, ou à tout le moins pseudo-logiques.

Les deux problèmes majeurs à considérer lorsque l'on veut concevoir un tel système, toujours selon Wooldridge et Jennings [Wooldridge and Jennings, 1995], sont :

Le problème de transduction, c'est à dire de traduire le monde réel par le biais d'une description adéquate et efficace, en un temps raisonnable (par raisonnable on entend assez court pour être utile).

Le problème de représentation et de raisonnement, c'est à dire comment d'une part représenter symboliquement les informations concernant les composantes et les processus complexes du monde réel, et d'autre part comment faire raisonner l'agent à partir de ces informations en un temps raisonnable.

Comme nous l'avons dit en introduction, l'IA s'est penché sur la problématique des agents à travers le domaine de la planification. Un des systèmes de planifications les plus vieux et les mieux connus est STRIPS [Fikes and Nilsson, 1971]. Ce système est souvent pris comme exemple d'architecture délibérative, et nous en présentons brièvement les grandes lignes. Il se compose des trois éléments suivants : premièrement d'un modèle symbolique de l'environnement, représenté par une logique de prédicats; deuxièmement par une spécification symbolique des actions disponibles, c'est à dire une liste qui indique dans quelles circonstances une action doit être entreprise et les effets de cette action; troisièmement un algorithme de planification, qui prend en entrée la représentation de l'environnement, un ensemble de spécifications d'actions et une représentation de l'état final – le but – recherché et produit un plan, ou plutôt un programme, qui spécifie comment agir afin d'atteindre le but. Ce plan est issu d'une analyse moyens-fin pour essayer de trouver une séquence d'actions qui vont réaliser le but.

Le type d'algorithme utilisé par STRIPS était en apparence simple mais s'avérait inefficace pour des problèmes de complexité modérée et on s'aperçut que ce modèle ne tenait pas la route pour des scénarii réalistes. En effet le temps nécessaire à la prise de décision, selon Chapman [Chapman, 1987], était trop long pour l'environnement dans lequel se passait l'action, et celui-ci avait eu le temps de changer entre-temps: le résultat s'avérait tout à fait inutile.

4.3.2. Architecture réactive

Les difficultés à régler les problèmes que l'approche délibérative imposait pour créer des agents incitèrent les chercheurs, selon Wooldridge [Wooldridge, 1999], à remettre en question le paradigme de l'IA symbolique et à travailler sur des approches alternatives. Ces nouvelles approches, telle celle de Brooks [Brooks, 1986], préconisèrent d'une part le rejet des représentations symboliques et de la prise de décision par des manipulations syntaxiques de ces dernières, et d'autre part suggérèrent l'idée selon laquelle le comportement intelligent de l'agent devrait être fortement lié à l'environnement dans lequel il se situe, et que ce comportement intelligent serait en fait le résultat d'interactions entre l'agent et cet environnement.

On fait référence à ces approches alternatives de diverses façons, selon Wooldridge [Wooldridge, 1999], tantôt « béhavioristes », puisque la caractéristique importante est le comportement de l'agent dans une situation précise, tantôt « *situated* » (ou contextualisées), faisant référence au fait que l'agent est intimement relié à l'environnement où il se situe, et tantôt « réactives » car enfin on considère que les agents réagissent à leur environnement plus qu'ils ne raisonnent sur celui-ci.

Nous présentons ici deux modèles d'architectures réactives, tels que proposés par Russel et Norvig [Russell and Norvig, 1995] et Wooldridge [Wooldridge, 1999], soient les *agents à réflexes simples*, et les *agents avec maintien d'états*.

4.3.2.1. Agents à réflexes simples

Les agents à réflexes simples (*simple reflex agents*) se servent de règles *condition-action* pour déterminer l'action à entreprendre. Pour donner un exemple d'une telle règle, Russel et Norvig [Russell and Norvig, 1995] font un parallèle avec

la conduite automobile. Un conducteur qui est attentif aux coups de freins donnés par le conducteur en avant de lui aurait comme règle condition-action :

Si la voiture en avant freine alors amorcer le processus de freinage

Un agent ayant pour tâche de freiner devrait posséder des capteurs pour connaître son environnement (ici, si la voiture en avant freine) et à la lumière de ces informations entreprendre une action à partir des règles condition-action.

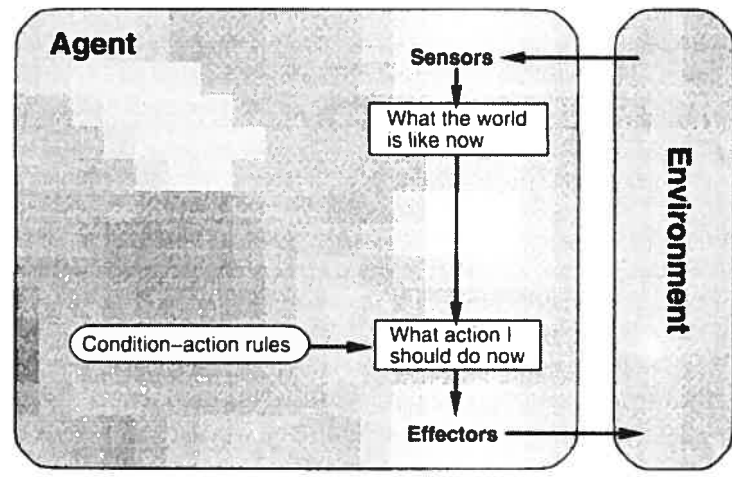


Figure 4.1 - Schéma d'un agent à réflexes simples [Russell and Norvig, 1995]

La Figure 4.1 (tirée de [Russell and Norvig, 1995]) représente schématiquement un agent à réflexe simple. L'agent, à l'aide de ses capteurs (*sensors*) perçoit son environnement et à l'aide de règles condition-action (*condition-action rules*), il prend une décision quant à l'action à entreprendre, action qu'il effectuera par le biais de ses effecteurs (*effectors*).

4.3.2.2. Agents avec maintien d'états

Les agents avec maintien d'états (*agents that keep track of the world*) ont le même comportement que les agents à réflexes simples, à la différence qu'ils conservent un état interne.

Pour en expliquer l'intérêt, reprenons l'image de la conduite automobile de Russell et Norvig: on désire maintenant faire un changement de voie, et pour cela on doit savoir s'il n'y a pas d'autres véhicules dans la voie d'à côté. Supposons également que nos seuls « capteurs » d'informations soient notre vision avant et le rétroviseur, dans lequel nous ne regardons que de temps en temps. Si nous voyons dans le rétroviseur une voiture qui s'approche et que quelques secondes après nous ne la voyons plus, nous pouvons penser qu'elle est maintenant dans notre angle mort et que nous ne pouvons pas changer de voie. Afin d'arriver à cette déduction, nous avons en mémoire le fait qu'il y avait une voiture et qu'elle se rapprochait. Nous avons conservé un état du monde et c'est grâce à cela que nous avons pu avoir la perception de la présence d'une voiture, bien que les informations fournies par nos capteurs, prises seules, nous auraient plutôt laisser croire le contraire.

Parce que les capteurs ne fournissent pas toujours une information complète de l'environnement, l'agent a besoin de maintenir de l'information sur son état interne et celui de son environnement afin de pouvoir distinguer deux états différents (dans notre exemple, l'état « il y a une voiture », et l'état « il n'y a pas de voiture ») qui peuvent générer la même perception (« il n'y a pas de voiture »). C'est donc dans ce but que les agents avec maintien d'états conservent des informations sur eux et leur environnement.

La Figure 4.2 donne une représentation de l'agent avec maintien d'états. Les informations fournies par les capteurs (*sensors*) sont combinées avec l'état interne (*state*), des informations obtenues au préalable sur l'évolution de l'environnement (*how the world evolves*) et sur la façon dont les actions prises par l'agent peuvent affecter l'environnement (*what my actions do*) afin de donner une description mise à

jour de l'état courant du monde (*what the world is like now*). Le reste du comportement de l'agent est le même que celui vu en section 4.3.2.1.

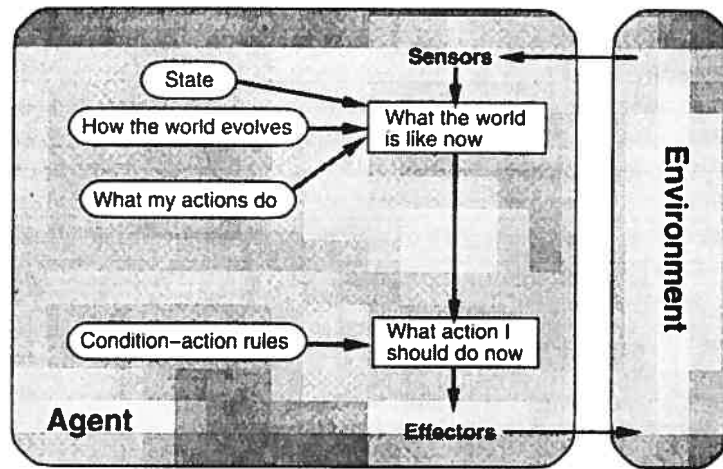


Figure 4.2 – Schéma d'agent avec maintien d'états [Russell and Norvig, 1995]

Ce type d'architecture est également présenté par Wooldridge [Wooldridge, 1999] et est représentée à la Figure 4.3.

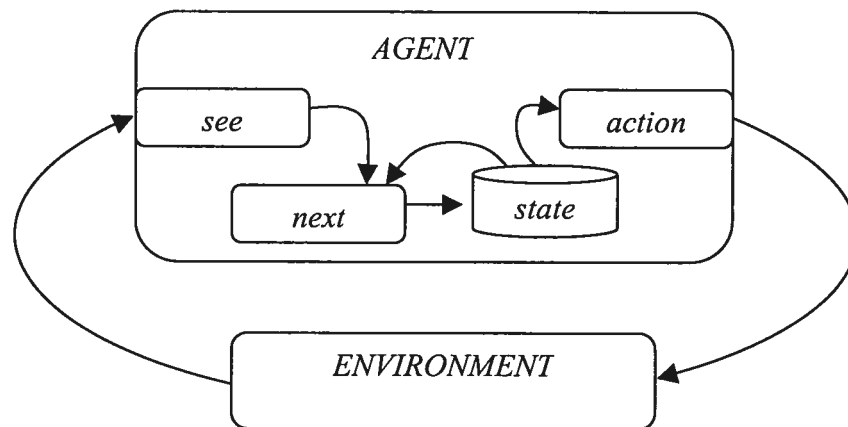


Figure 4.3 - Schéma d'agent avec maintien d'états [Wooldridge, 1999]

L'agent est à l'écoute de son environnement, et à partir des informations qu'il en a retirées, il en établit une certaine perception (*see*). Cette perception permettra de mettre à jour (*next*) l'état (*state*) dans lequel se trouve l'agent. Finalement, son nouvel état déterminera l'action à prendre (*action*).

4.3.3. Architecture BDI

L'architecture BDI (pour *belief-desire-intention*, c'est à dire croyance, désir et intention) est basée sur un procédé de décision pas à pas des actions à prendre pour atteindre un but.

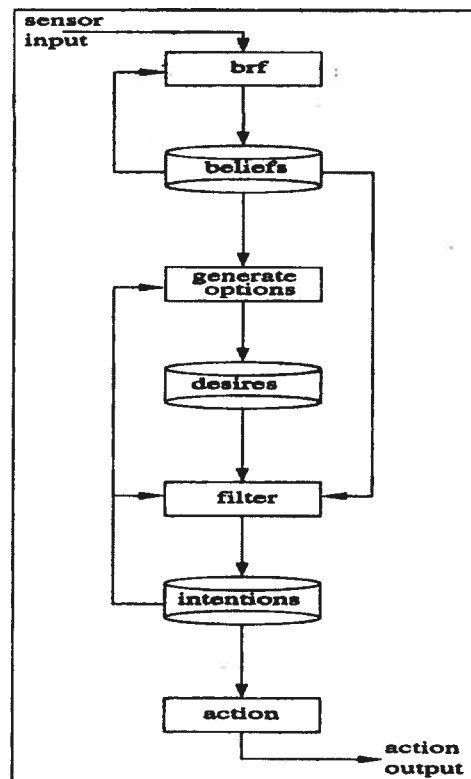


Figure 4.4 - Architecture BDI [Wooldridge, 1999]

Cette architecture se compose de sept éléments [Wooldridge, 1999] que reprend la Figure 4.4 et que nous expliquons ici:

1. **un ensemble de croyances courantes** (*beliefs*) qui sont en fait les informations que l'agent possède de son environnement.

2. **une fonction de révision** (*brf – belief revision function*), qui prend des perceptions et les croyances courantes de l'agent et retourne un nouvel ensemble de croyances.
3. **une fonction génératrice d'options** (*generate options*, sur la figure), qui détermine quelles options s'offrent à l'agent, à partir des croyances et des intentions de celui-ci, bref qui détermine ses désirs.
4. **un ensemble d'options courantes** (*desires*), à savoir des actions disponibles pour l'agent.
5. **une fonction de filtre** (*filter*), qui est en fait un procédé de délibération pour l'agent, et qui donne les intentions courantes de ce dernier à partir des croyances, des désirs et des intentions.
6. **un ensemble d'intentions courantes** (*intentions*), c'est à dire ce sur quoi l'agent se concentre actuellement, les buts qu'il souhaite atteindre et sur lesquels il est en train d'engager des ressources.
7. **une fonction d'exécution** (*action*), qui détermine l'action à prendre à partir des intentions courantes.

Pour résumer, l'agent BDI met à jour ses croyances à partir des informations reçues par ses capteurs et provenant de son environnement ; cette mise à jour lui permet d'établir les options qui lui sont disponibles ; ces options sont filtrées afin de déterminer ses nouvelles intentions et finalement l'agent peut poser des actions en ce basant sur ses intentions mises à jour.

4.3.4. Architecture en couches ou hybride

Nous avons présenté précédemment les architectures délibérative et réactive. Dans le premier cas, on désire agir après une étape de réflexion préalable, alors que dans le deuxième cas on tente de réagir le plus rapidement possible à un événement. Pourtant, il arrive que l'on ait besoin d'un agent ayant un comportement qui puisse allier à la fois les aspects réactif et délibératif : un agent hybride [Jennings et al., 1998].

Cette idée d'agent hybride implique selon Wooldridge [Wooldridge, 1999] la création de sous-systèmes séparés afin de répondre au deux types de comportements différents. Cette conception mène à une architecture dans laquelle les sous-systèmes interagissent par le biais de couches hiérarchiques. Typiquement, il devrait exister au moins deux couches, l'une répondant au besoin d'être réactif et l'autre à celui d'être délibératif. Deux types d'architectures en couches ont été identifiés :

les couches horizontales : dans ce modèle chaque couche est reliée aux récepteurs de l'agent et chacune émet des suggestions sur l'action à prendre.

les couches verticales : les données perçues sont traitées par au plus une couche, mais pas forcément par toutes, dépendamment du degré de complexité de la réflexion que l'on désire entreprendre.

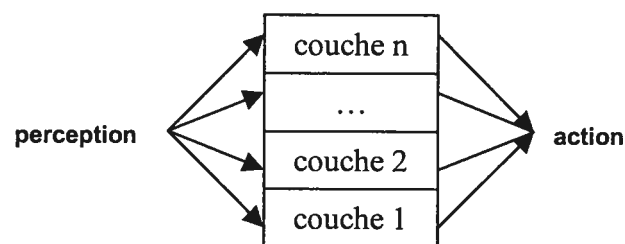


Figure 4.5 - Architecture en couches horizontales [Wooldridge, 1999]

Le grand avantage des couches horizontales (voir Figure 4.5) est leur simplicité. Afin d'attribuer à l'agent plusieurs comportements différents, il suffit d'implémenter chaque comportement désiré de manière indépendante, chacun correspondant à une couche particulière.

Dans ce modèle, rappelons le, chaque couche est directement reliée aux capteurs en entrée, et peut générer une réponse en sortie. Cependant, si plusieurs couches génèrent des actions en même temps, il pourrait y avoir un comportement incohérent de la part de l'agent. C'est pourquoi une fonction médiatrice est chargée à tout moment de prendre la décision de ne retenir qu'une seule couche pour effectuer l'action de l'agent. Un problème se pose cependant : étant donné le nombre important d'actions suggérées par toutes les couches, leur tri peut occasionner un ralentissement dans l'exécution de l'agent.

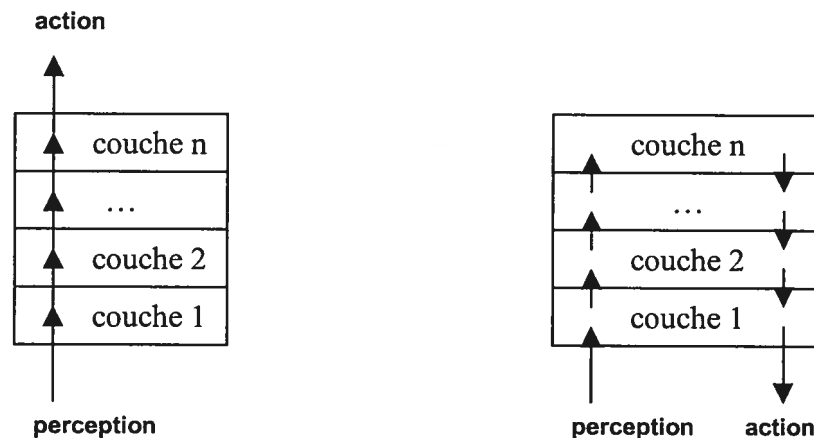


Figure 4.6 - Architecture en couches verticales [Wooldridge, 1999]

Dans le modèle en couches verticales (voir Figure 4.6), il n'y a qu'une seule couche qui reçoit les informations en entrée, et une seule qui génère des actions en sortie. Les couches sont arrangées de manière hiérarchique, les couches les plus basses étant les plus réactives et basant leurs décisions sur les données brutes provenant des capteurs, les plus élevées étant les plus délibératives, basant leurs

décisions sur une vision plus globale de l'environnement. Ce type d'architecture en couches verticales se décline en deux classes: à un seul passage, et à deux passages.

Dans le modèle à un seul passage, le contrôle du traitement se déplace à travers les n couches séquentiellement, avec en entrée de la couche la plus basse les informations provenant de la perception de l'agent, et à la sortie de la couche la plus élevée des actions possibles proposées.

Dans le cas du passage double, le principe est le même, sauf qu'une fois arrivé à la couche la plus élevée, on redescend à travers chaque couche jusqu'à la première, ceci afin de permettre une certaine cohérence dans les propositions d'action.

Le problème du goulot d'étranglement (*bottleneck*) qui peut se produire avec les couches horizontales est passablement réduit lorsqu'une architecture en couches verticales est utilisée. Cependant la nécessité de traverser chacune des couches pour décider de l'action à prendre demande non seulement une certaine flexibilité d'interaction entre les couches, mais il suffit qu'une seule d'entre elles ne fonctionne pas correctement pour que la performance de l'ensemble du système s'en trouve affectée.

4.4. Résumé

Dans ce chapitre, nous avons présenté la technologie agent et avons tenté d'en cerner les caractéristiques principales. Nous avons étudié quatre grandes classes d'architectures les plus utilisées. Comme nous le verrons dans le prochain chapitre, cette section nous permettra de choisir une architecture agent qui sera adaptée à notre contexte de travail.

5. Modèle de création générique d'agents de médiation

5.1. Introduction

Comme nous l'avons vu au début de ce mémoire, notre travail s'inscrit dans le cadre du projet Consensus, qui vise à automatiser les négociations électroniques combinées à l'aide d'agents. Notre objectif est de réussir à créer, pour chaque négociation, un agent spécifique à partir d'un modèle générique, le but étant de ne pas avoir à programmer un agent pour chaque cas.

Dans ce chapitre, nous commencerons par choisir une architecture agent adaptée au contexte dans lequel les agents que nous souhaitons créer évolueront. Puis nous présenterons conceptuellement notre modèle de création d'agent et définirons les paramètres nécessaires à son fonctionnement.

Nous présenterons ensuite notre démarche de conception, basée sur l'étude des événements et des actions qui y sont associées à partir des négociations étudiées au chapitre 3. Puis nous expliquerons l'architecture du système de création que nous avons développé. Enfin, nous présenterons les détails d'implémentation et le fonctionnement général du système ainsi généré.

5.2. Choix de l'architecture agent

Dans le chapitre 4, nous avons présenté la technologie des agents et avons vu quatre grandes classes d'architectures. Chacune de ces architectures a été développée pour répondre à des besoins et des contraintes bien particuliers. Dans cette section, nous tenterons de déterminer le type d'architecture que nous

retiendrons pour les agents de négociation que nous désirons créer, en tenant compte du contexte de notre travail. Nous expliquerons le choix que nous avons fait.

Durant l'étude, au chapitre 3, des diagrammes décrivant les mécanismes de plusieurs négociations, nous avons vu que les états dans lesquels le participant se retrouve dépendent uniquement de messages échangés avec l'encanteur : le participant envoie un message pour s'enregistrer, pour miser, reçoit des messages contenant des cotes, ou une adjudication, etc. Pour un agent que l'on désire voir reproduire le comportement d'un participant, ces messages constituent donc l'essentiel des entrées (informations reçues par les capteurs de l'agent) et sorties (les actions prises par l'agent).

Pour ce qui est de la prise de décisions, celles-ci se font à la suite de la réception d'un message mettant à jour les connaissances et l'état du participant pendant le déroulement de la négociation. Pour illustrer ceci, prenons l'exemple d'une enchère anglaise : si un participant reçoit une cote l'informant qu'il est leader, ce dernier ne fera que suivre le déroulement de l'enchère, alors que s'il reçoit un message l'informant qu'il n'est pas leader, il soumettra une mise.

En d'autres termes, la prise de décision se limite à un ensemble de règles condition-action (voir section 4.3.2). Cet argument nous fait pencher pour le choix d'une architecture réactive.

Toutefois, le comportement d'un participant n'est pas purement réactif : en effet, l'action posée par un participant ne dépend pas seulement des messages qu'il reçoit, mais aussi de l'état dans lequel il se trouve au moment de recevoir des messages. Nous avons donc fait pencher notre choix sur l'architecture réactive avec maintien d'états (voir section 4.3.2.2), qui nous semble la mieux adaptée au contexte de notre travail.

5.3. Le Modèle de création d'agent

La Figure 5.1 décrit de manière conceptuelle le modèle de création d'agents de négociation que nous proposons afin de répondre au principal objectif que nous nous sommes fixé au début de ce travail : créer un modèle d'agent générique capable de participer spécifiquement à un type de négociation donnée à partir d'un nombre restreint de paramètres. Notre modèle conceptuel de création va comme suit (voir Figure 5.1): dans un premier temps (1), l'utilisateur énonce ses besoins; quant à l'agent serveur avec lequel on souhaite pouvoir négocier, il doit pouvoir rendre disponible un fichier (2) contenant les paramètres des règles de la négociation.

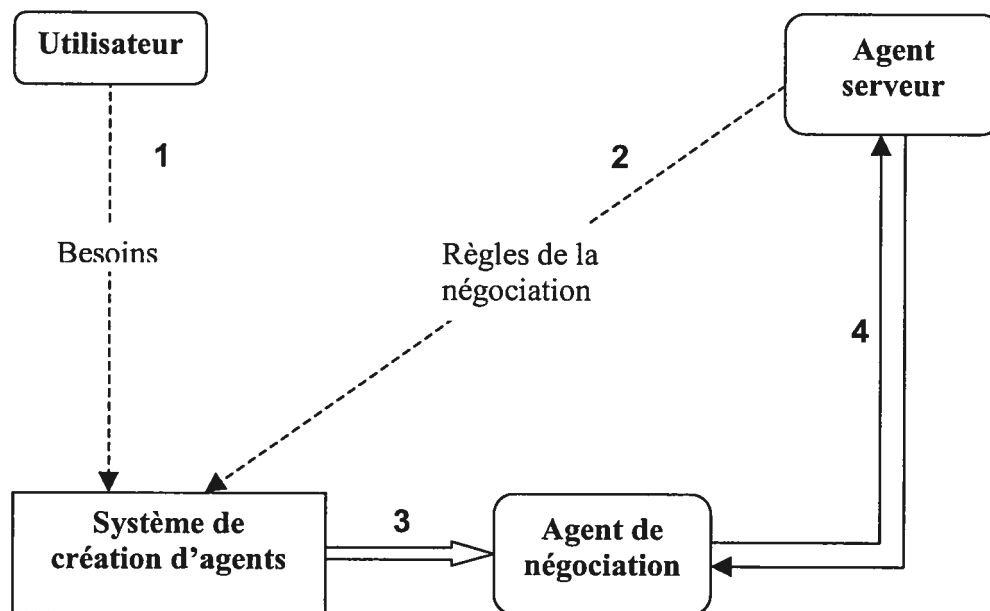


Figure 5.1 - Représentation conceptuelle du modèle

Puis, à partir de ces informations, le système instancie (3) un agent négociateur, dont le comportement sera en accord avec les règles imposées par l'agent serveur. Finalement, l'agent nouvellement créé peut s'enregistrer auprès de l'agent serveur et négocier (4) avec celui-ci en ayant comme objectif de combler les besoins de l'utilisateur.

Nous définissons dans les prochaines sous-sections de quelles natures sont les données reçues par le système.

5.3.1. Les besoins du client

Il est important de définir ce que l'on entend par les besoins du client. Il y a tout d'abord la quantité (soit en unités, en volume, en poids, etc.). Il y a ensuite le prix de réserve, c'est à dire le prix unitaire que le client est prêt à déboursier pour obtenir le produit. Les autres éléments à prendre en compte sont les dates de livraisons, les modalités de paiement, etc.

Dans la vie courante, il faut tenir compte de tous ces éléments, mais nous avons décidé de ne prendre comme variables que le prix de réserve et la quantité. Ce choix découle de notre volonté de simplifier la stratégie d'achat du participant au maximum. En effet, le domaine de la stratégie demanderait une étude approfondie qui ne fait pas partie de nos objectifs et qui dépasse le cadre de ce travail. Nous nous en tiendrons donc à l'adoption de la stratégie la plus simple pour le participant, soit de miser lorsque cela lui est possible, c'est à dire lorsque la quantité offerte est suffisante pour ses besoins et que le prix est en deçà du prix maximum qu'il est prêt à déboursier.

Nous résumons ainsi les besoins du participant dans les négociations :

- **Quantité** : nombre entier
- **Prix de réserve** : montant (dans une unité monétaire)

5.3.2. Les paramètres des règles de la négociation

Nous définissons les règles de la négociation comme étant l'ensemble des caractéristiques qui définissent une négociation.

Nous avons établi, dans la section 2.3.9, une classification des enchères. Cette classification, rappelons-le, est basée sur les deux types d'attributs suivants : les attributs du type de négociation et les attributs de la négociation.

Rappelons ces attributs, ainsi que les valeurs que ceux-ci peuvent prendre :

Les attributs du type de négociation :

- **Côté du marché** : achat | vente | double
- **Format d'interaction** : continu | séquentiel
- **Contrôle des mises** : ouvert | cadencé
- **Progression des mises** : croissante | décroissante
- **Politique de fixation des prix** : discriminatoire | uniforme

Les attributs de la négociation :

- **Mise minimale de départ** : montant (exprimé dans une unité monétaire)
- **Incrément minimum** : fixé (montant) | proportionnel (pourcentage)
- **Nombre d'unités** : nombre entier
- **Règle de clôture** : expiration d'une date de clôture | inactivité des participants | épuisement des stocks | chute du prix en deçà d'un prix plancher | combinaison de plusieurs occurrences

Nous n'avons pas conservé l'attribut « Politique de fixation des prix », et nous expliquons pourquoi : dans le cadre de notre travail, nous ne tenons compte d'aucune stratégie particulière pour notre agent (hypothèse que nous avons posée

dès le départ), et la politique de fixation des prix n'influence le comportement du participant que dans la mesure où celui-ci a une stratégie. Par exemple, dans une enchère Vickrey (voir section 2.3.6), un participant peut décider de soumettre une mise supérieure à son prix de réserve en sachant qu'il ne paiera pas le bien à ce prix. La décision de miser au-dessus du prix de réserve, et combien, constitue une stratégie qui dépasse le cadre de notre étude.

Nous avons fait l'hypothèse que la règle de clôture de la négociation était une combinaison de la date de clôture (un temps relativement court d'au plus quelques minutes), de l'épuisement des stocks et de la chute du prix en dessous d'un prix plancher. Nous avons posé cette hypothèse d'abord pour des questions de réalisme, étant donné que ce sont les trois règles de clôture les plus souvent utilisées sur Internet, puis ensuite pour des raisons pratiques en prévision des tests pour le modèle que nous avons développé, afin de pouvoir expérimenter dans des délais raisonnables le comportement des agents créés.

Nous avons ajouté à la liste un attribut de type de négociation : le prix. Nous avons vu dans la section 3.3.5 que pour les négociations de type marchandage (dont la vente à prix fixe n'est qu'un cas particulier), il existait deux possibilités quant au prix suggéré par le vendeur ou la contre-offre de l'acheteur : soit le prix est négociable (l'autre participant a le choix d'accepter le prix proposé ou de faire une contre-offre) ou alors le prix est final (ou non-négociable) et l'autre participant n'a que le choix d'accepter ou de refuser le prix proposé

Nous avons donc jugé important, pour couvrir tous les types de négociations sur lesquelles se penche Consensus, d'ajouter cet autre attribut de type de négociation :

- **Prix** : négociable | non-négociable

5.4. Les négociations concernées

Les négociations auxquelles s'intéresse le projet Consensus se résument d'abord et avant tout aux enchères que l'on retrouve le plus souvent sur Internet, à savoir les enchères anglaises, hollandaise et par l'offre secrète. S'ajoute à cela des ventes à prix fixes, et parfois des négociations de type marchandage. Tous ces modèles de négociations correspondent à ceux que nous avons étudiés dans les chapitres 2 et 3. Dans le chapitre 3, nous avons étudié plus en détail les mécanismes de cinq négociations qui sont représentatives des différents types visés dans le projet Consensus. Nous avons adopté un modèle afin de représenter ces cinq négociations. Le but de ces représentations était de réussir à extraire les états de base dans lesquels un participant peut se trouver lors de sa participation à ces négociations. Grâce à cette analyse, nous avons pu dégager les états essentiels suivants : Non Enregistré, Non Leader, Leader, Gagnant, Perdant, Final. Ces états seront très important afin de pouvoir concevoir notre modèle, car comme nous le verrons un peu plus loin, ce sont les états internes de l'agent qui l'influenceront dans ses comportements.

Nous avons vu en section 5.2 que l'architecture que nous avons retenue pour notre agent en est une réactive avec maintien d'états. Nous allons, dans cette section, dresser une liste des événements, des réactions qui y sont reliées ainsi que des changements d'états associés pour les cinq types de négociations vus au chapitre 3. Nous ferons ensuite des regroupements pour en faire ressortir des règles de condition-action en vue de la conception de l'architecture de notre système.

5.4.1. Analyse des événements-actions

Nous avons, pour chaque négociation vue dans le chapitre 3, fait un tableau exprimant les actions associées à chaque événement possible, en tenant compte de l'état et des conditions nécessaires, ainsi que le changement d'état du participant.

Comme nous l'avons dit dans la section 5.2, les événements qui surviennent sont des messages échangés entre l'agent participant et l'agent serveur. Nous commençons par expliquer les types de messages pouvant être échangés entre le vendeur et l'acheteur, avant de présenter les tableaux.

5.4.1.1. Hypothèses quant aux messages échangés

Nous tenons ici à spécifier les messages échangés et les hypothèses sur lesquelles nous nous sommes basés avant de faire la liste des événements, des mises à jour de l'état interne et des actions prises par l'agent.

La première hypothèse que nous avons posée est que l'agent serveur tient les agents participants au courant de l'état de la négociation à intervalles réguliers, que l'enchère ait évolué ou non, à moins qu'il s'agisse d'une négociation à deux pairs (voir section 3.3.5), où il n'y a qu'un envoi de messages successifs en alternance entre le vendeur et l'acheteur.

L'autre hypothèse que nous avons posée concerne le type des messages échangés entre agents clients et agents serveurs. Voici la liste des messages possibles :

- Demande d'enregistrement
- Confirmation d'enregistrement
- Rejet d'enregistrement
- Cote
- Mise (ou encore offre)
- Accepter offre
- Négociation terminée
- Quitter la négociation
- Adjudication

5.4.1.2. Enchère anglaise

Pour cette enchère, il n'y a qu'une seule unité mise en vente et les attributs du type d'enchère ont les valeurs suivantes :

Côté du marché : achat.

Format d'interaction : continu.

Contrôle des mises : ouvert.

Progression des mises : croissante.

Prix : non-négociable.

Le Tableau 5.1 présente les événements possibles pour cette enchère, les actions engendrées selon les conditions ainsi que l'état mis à jour de l'agent. On désigne par «PL» le prix leader, par «PR» le prix de réserve du participant et par «IM» l'incrément minimum de la négociation.

<i>État initial</i>	<i>Message - Événement</i>	<i>Conditions</i>	<i>Action</i>	<i>État final</i>
Non-Enregistré	Création de l'agent		Envoyer enregistrement	Non-Enregistré
	Confirmation d'enregistrement		Rien (attendre cote)	Non-Leader
	Rejet d'enregistrement		Envoyer enregistrement	Non-Enregistré
Non-Leader	Cote	Leader	Rien (attendre cote)	Leader
		$(PL + IM) \leq PR$	Miser	Non-Leader
		$(PL + IM) > PR$	Quitter	Perdant
	Adjudication			Gagnant
	Négociation terminée			Perdant
Leader	Cote	Leader	Rien (attendre cote)	Leader
		$(PL + IM) \leq PR$	Miser	Non-Leader
		$(PL + IM) > PR$	Quitter	Perdant
	Adjudication			Gagnant
	Négociation terminée			Perdant
Gagnant				Final
Perdant				Final

Tableau 5.1 - Liste des événements-actions : enchère anglaise

5.4.1.3. Enchère hollandaise

Pour cette enchère, il n'y a qu'une seule unité mise en vente, et les attributs du type d'enchère ont les valeurs suivantes :

Côté du marché : achat.

Format d'interaction : continu.

Contrôle des mises : cadencé.

Progression des mises : décroissante.

Prix : non-négociable.

<i>État initial</i>	<i>Message - Événement</i>	<i>Conditions</i>	<i>Action</i>	<i>État final</i>
Non-Enregistré	Création de l'agent		Envoyer enregistrement	Non-Enregistré
	Confirmation d'enregistrement		Rien (Attendre cote)	Non-Leader
	Rejet d'enregistrement		Envoyer enregistrement	Non-Enregistré
Non-Leader	Cote	$O > PR$	Rien (Attendre cote)	Non-Leader
		$O \leq PR$	Miser	Non-Leader
	Adjudication			Gagnant
	Négociation terminée			Perdant
Gagnant				Final
Perdant				Final

Tableau 5.2 - Liste des événements-actions : enchère hollandaise

Le Tableau 5.2 présente les événements possibles pour cette enchère, les actions engendrées selon les conditions ainsi que l'état mis à jour de l'agent. On désigne par «O» le montant de l'offre et «PR» le prix de réserve du participant.

5.4.1.4. Enchère hollandaise multi-items

Pour cette enchère, les attributs du type d'enchère ont les mêmes valeurs que pour l'enchère hollandaise à un seul item.

<i>État initial</i>	<i>Message - Événement</i>	<i>Conditions</i>	<i>Action</i>	<i>État final</i>
Non-Enregistré	Création de l'agent		Envoyer enregistrement	Non-Enregistré
	Confirmation d'enregistrement		Rien (Attendre cote)	Non-Leader
	Rejet d'enregistrement		Envoyer enregistrement	Non-Enregistré
Non-Leader	Cote	$QR < QD$	Quitter	Perdant
		$(QR \geq QD) \text{ ET } (O \leq PR)$	Miser	Non-Leader
		$(QR \geq QD) \text{ ET } (O > PR)$	Rien (Attendre cote)	Non-Leader
	Adjudication	Besoins non-comblés	Rien (Attendre cote)	Non-Leader
		Besoins comblés	Quitter	Gagnant
	Négociation terminée			Perdant
Gagnant				Final
Perdant				Final

Tableau 5.3 - Liste des événements-actions : enchère hollandaise multi-items

Le Tableau 5.3 présente les événements possibles pour cette enchère, les actions engendrées selon les conditions ainsi que l'état mis à jour de l'agent. On désigne par «O» le montant de l'offre, «PR» le prix de réserve du participant, «QR» la quantité restante de l'encanteur et «QD» la quantité désirée par le participant.

5.4.1.5. Enchère par l'offre secrète

Pour cette négociation, il n'y a qu'une seule unité mise en vente et les attributs du type d'enchère ont les valeurs suivantes :

Côté du marché : achat.

Format d'interaction : séquentiel.

Contrôle des mises : ouvert.

Progression des mises : croissante.

Prix : non-négociable.

<i>État initial</i>	<i>Message - Événement</i>	<i>Conditions</i>	<i>Action</i>	<i>État final</i>
Non-Enregistré	Création de l'agent		Envoyer enregistrement	Non Enregistré
	Confirmation d'enregistrement		Miser	Non Leader
	Rejet d'enregistrement		Envoyer enregistrement	Non Enregistré
Non-Leader	Adjudication			Gagnant
	Négociation terminée			Perdant
Gagnant				Final
Perdant				Final

Tableau 5.4 - Liste des événements-actions : enchère par l'offre secrète

Le Tableau 5.4 présente les événements possibles pour cette enchère, les actions engendrées selon les conditions ainsi que l'état mis à jour de l'agent. On désigne par «O» le montant de l'offre, «PR» le prix de réserve du participant, «QR» la quantité restante de l'encanteur et «QD» la quantité désirée par le participant.

5.4.1.6. Négociation à deux pairs

Pour cette négociation, il n'y a qu'une seule unité mise en vente et les attributs du type d'enchère ont les valeurs suivantes :

Côté du marché : achat.

Format d'interaction : séquentiel.

Contrôle des mises : ouvert.

Progression des mises : croissante.

Prix : négociable.

Le Tableau 5.5 présente les événements possibles pour cette enchère, les actions engendrées selon les conditions ainsi que l'état mis à jour de l'agent. On désigne par «O» le montant de l'offre, «ON» une offre négociable, «OF» une offre finale, «PR» le prix de réserve du participant, «QR» la quantité restante de l'encanteur et «QD» la quantité désirée par le participant.

<i>État initial</i>	<i>Message - Événement</i>	<i>Conditions</i>	<i>Action</i>	<i>État final</i>
Non-Enregistré	Création de l'agent		Envoyer enregistrement	Non-Enregistré
	Confirmation d'enregistrement		Rien (Attendre offre)	Non-Leader
	Rejet d'enregistrement		Envoyer enregistrement	Non-Enregistré
Non-Leader	Offre	ON ET (O > PR)	Envoyer une offre	Non-Leader
		ON ET (O <= PR)	Accepter offre	Non-Leader
		OF ET (O > PR)	Quitter (rejeter offre)	Perdant
		OF ET (O <= PR)	Accepter offre	Non-Leader
	Adjudication			Gagnant
	Négociation terminée			Perdant
Gagnant				Final
Perdant				Final

Tableau 5.5 - Liste des événements-actions : négociation à deux pairs

5.4.2. Regroupement des événements-actions

Après avoir dressé la liste des événements-actions pour chaque négociation, nous avons regroupé, dans le Tableau 5.6, toutes ces possibilités. Nous expliquons ici ce tableau: nous avons regroupé d'abord par état, ensuite par événement puis avons distingué les actions à prendre à partir, d'une part des règles de la négociation et d'autre part des conditions à vérifier. À noter que nous n'avons pas inclus l'attribut « côté du marché » étant donné que toutes les négociations que nous couvrons sont toutes du côté acheteur. Les attributs sont ici : PN pour spécifier si le prix est négociable ou non négociable, « Cr. » pour croissant (par opposition à décroissant), « Ca. » pour cadencé (par opposition à ouvert), « Co. » pour continu (par opposition à séquentiel) et finalement MI pour multi-items (par opposition à simple item). Un « x » dans une colonne désigne l'attribut et son absence son attribut opposé (par exemple, pas de « x » dans la colonne « Cr. » désigne une négociation à prix décroissant). Quant aux signes « - » ils indiquent qu'il n'y a pas de distinction à faire entre les types de négociations, et indiquent donc un regroupement.

État initial	Message - Événement	Règles : attributs					Conditions	Action	État final	
		PN	Cr.	Ca.	Co.	MI				
Non- Enregistré	Création de l'agent	-	-	-	-	-		Envoyer enregistrement	Non-Enregistré	
	Confirmation d'enregistrement	-	-	-	-	-		Rien	Non-Leader	
	Rejet d'enregistrement	-	-	-	-	-		Envoyer enregistrement	Non-Enregistré	
Non-Leader	Cote	x		x			Leader	Rien (Attendre cote)	Leader	
		x		x			$(PL + IM) \leq PR$	Miser	Non-Leader	
		x		x			$(PL + IM) > PR$	Quitter	Perdant	
			x		x		$O > PR$	Rien (Attendre cote)	Non-Leader	
			x		x		$O \leq PR$	Miser	Non-Leader	
			x	x		x	$QR < QD$	Quitter	Perdant	
			x	x		x	$(QR \geq QD) \text{ ET } (O \leq PR)$	Miser	Non-Leader	
			x	x		x	$(QR \geq QD) \text{ ET } (O > PR)$	Rien (Attendre cote)	Non-Leader	
			x	x		x	$ON \text{ ET } (O > PR)$	Envoi offre	Non-Leader	
			x	x		x	$ON \text{ ET } (O < PR)$	Accepter offre	Non-Leader	
			x	x		x	$OF \text{ ET } (O > PR)$	Quitter (rejeter offre)	Non-Leader	
			x	x		x	$OF \text{ ET } (O \leq PR)$	Accepter offre	Non-Leader	
			Adjudication	x		x				Gagnant
				x		x				Gagnant
	x	x			x	Besoins non comblés	Rien (Attendre cote)	Non-Leader		
	x	x			x	Besoins comblés		Gagnant		
	x							Gagnant		
	x	x			x			Gagnant		
	Négociation terminée	-	-	-	-	-		Perdant		
Leader	Cote	x		x			Leader	Rien (Attendre cote)	Leader	
		x		x			$(PL + IM) \leq PR$	Miser	Non-Leader	
		x		x			$(PL + IM) > PR$	Quitter	Perdant	
		Adjudication	x		x				Gagnant	
		Négociation terminée	x		x				Perdant	
Gagnant		-	-	-	-	-		Final		
Perdant		-	-	-	-	-		Final		

Tableau 5.6 - Regroupement des événements-actions

5.5. Architecture de l'agent généré

À la lumière des sections précédentes, nous pouvons maintenant présenter l'architecture de l'agent que nous proposons. Notre implémentation, dont nous parlerons dans la section suivante, reflètera cette architecture.

5.5.1. L'architecture globale

Comme nous l'avons vu à la section 5.2, l'architecture que nous avons adoptée est la réactive avec maintien d'états. La Figure 5.2 décrit le mécanisme de l'agent.

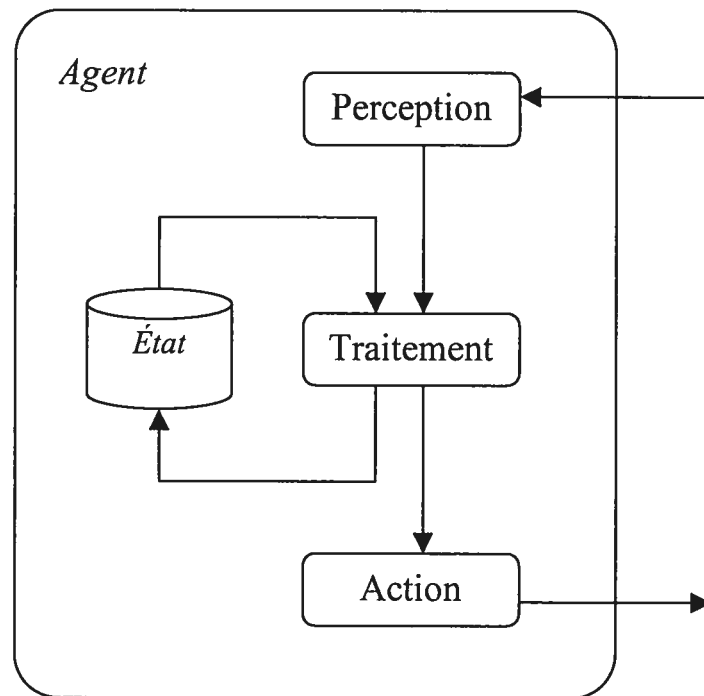


Figure 5.2 - Architecture de l'agent généré

L'agent reçoit des messages de l'environnement par le biais du module « Perception ». L'information est ensuite donnée au module « Traitement », qui

prend également en entrée l'état interne de l'agent. À partir de ces données, ce module va, d'une part, suggérer l'action à prendre et la confier au module « Action », et d'autre part remettre à jour l'état interne de l'agent. Finalement, le module « Action » exécute l'action qui a été suggérée.

5.5.2. Description des différents modules

Le module « Perception » est essentiellement le module qui reçoit les messages et qui en extrait les informations pertinentes en regard du type de message (s'il s'agit d'une adjudication, d'une cote, d'une confirmation d'enregistrement, etc.).

Le module « Traitement » associe un événement à une action à entreprendre, en tenant compte, d'une part, de l'état interne de l'agent et d'autre part des règles de la négociation et en faisant la vérification des conditions. Il transmet l'ordre d'exécuter l'action au module « Action ». Puis il met à jour l'état interne de l'agent. Typiquement, le module « Traitement » suit la liste des événements-actions (voir Tableau 5.6) que nous avons établie dans la section 5.4.

Finalement, le module « Action » s'occupe de l'exécution de l'action qui lui a été suggérée. Typiquement il s'agit de construire le type de message approprié (une mise, un enregistrement, etc.) et d'en faire l'envoi.

5.6. Implémentation

Dans cette section, nous présentons l'implémentation que nous avons faite de notre modèle de création, d'une part, et de notre architecture agent, d'autre part. Nous commençons par décrire notre contexte de programmation pour ensuite présenter plus en détails les différentes composantes de notre système.

5.6.1. Contexte d'implémentation

Nous avons dû faire des choix d'implémentation en regard du contexte du projet Consensus et des éléments déjà en place. Nous expliquons brièvement ici ce contexte.

5.6.1.1. Serveur de négociation

Afin de pouvoir permettre une compatibilité des systèmes, nous avons utilisé les serveurs de négociations qu'utilise Consensus pour ses simulations. Ceux-ci sont essentiellement des serveurs d'enchères; nous avons dû en implémenter plusieurs versions et avons adapté celles déjà existantes pour que ceux-ci puissent refléter les négociations que nous avons étudiées.

5.6.1.2. Communications, langage de programmation

Les serveurs utilisés par Consensus utilisent le langage Java, et nous avons implémenté notre système dans le même langage pour des raisons pratiques.

Les messages échangés entre l'agent serveur de négociation et l'agent client se font par le biais d'une *socket* par laquelle passe les flots de données. Nous avons conservé cette façon de faire et avons fait passer des objets sérialisés sur le canal de communication.

5.6.2. La classe « Message » et les types de messages

Nous commençons par décrire la classe « Message », car il est important de bien expliquer de quelle façon les informations sont échangées entre agent client et agent serveur. La classe « Message » est sérialisable, ce qui permet à Java de faire

passer ces objets sur la *socket* en un flot de données. La classe « Message » est constituée des champs illustrés à la Figure 5.3, que nous décrivons plus loin :

messageType	messageObject	portNumber	expeditor	messageID
-------------	---------------	------------	-----------	-----------

Figure 5.3 - La composition de l'objet *Message*.

5.6.2.1. Le corps du message, *messageObject*

Le corps du message est *messageObject*. Ce corps de message peut être l'une des choses suivantes (avec entre parenthèses la classe correspondante):

- une demande d'enregistrement (*Registration*)
- une réponse à une demande d'enregistrement (*RegistrationConfirmation*)
- une cote (*Quote*)
- une mise/offre (*Order*)
- une réponse à une mise/offre (*OrderResponse*)
- une demande de déconnexion (*Quit*)
- un avis de fin de négociation (*End*)
- une adjudication (*Adjudication*)

5.6.2.2. Les autres paramètres du message

La variable *messageType* est en fait une chaîne décrivant le type d'objet contenu dans le message. Par exemple s'il s'agit d'une adjudication, la variable *messageType* a pour valeur « Adjudication ». L'idée est de savoir rapidement de quel type de message il s'agit à la simple lecture de *messageType*, et d'en extraire rapidement les attributs.

La variable *portNumber* indique le numéro du port de connexion, la variable *expeditor* décrit le nom de l'expéditeur et *messageID* est un numéro d'identification pour le message.

5.6.3. La classe « AgentFactory »

C'est la classe principale : en effet c'est elle qui instancie les agents à partir des règles du marché sur lequel l'agent ira négocier, et des besoins fournis par l'utilisateur.

La classe comprend essentiellement deux méthodes : la première va extraire les informations d'un fichier décrivant les règles du marché sur lequel l'agent ira négocier, et la seconde va littéralement créer l'agent.

Par ailleurs, à la création d'*AgentFactory* une interface usager est démarrée. Nous expliquerons plus loin son intérêt.

5.6.3.1. La méthode *readRules*

Cette méthode se charge de lire le fichier texte (dont le nom est passé en argument à la méthode) dans lequel se retrouvent les principaux attributs qui décrivent une négociation (tels que nous les avons vus dans la section 5.3.2) :

- **Côté du marché** : achat | vente | double
- **Format d'interaction** : continu | séquentiel
- **Contrôle des mises** : ouvert | cadencé
- **Progression des mises** : croissante | décroissante
- **Mise minimale de départ** : montant (exprimé dans une unité monétaire)
- **Incrément minimum** : fixé (montant) | proportionnel (pourcentage)
- **Nombre d'unités** : nombre entier
- **Prix** : négociable | non-négociable

Nous avons également ajouté les trois attributs suivants:

- **Durée de l'enchère** (en secondes): nombre entier.
- **Nom de l'hôte** : Chaîne de caractères.
- **Numéro de port** : nombre entier.

Ce fichier de règles est organisé dans un format similaire à celui d'un fichier XML, soit avec des étiquettes (communément appelées *tags*) qui délimitent les champs énumérés plus haut.

Lorsque le fichier est lu, on crée un objet de type *Rules*, dans lequel on retrouvera en attributs tous les champs servant à décrire les règles lues. Cet objet *Rules* servira ensuite à l'agent comme référence pour l'enchère.

5.6.3.2. La méthode *createAgent*

Il s'agit bien sûr de la méthode qui crée l'agent. Il s'agit simplement de créer une instance de la classe *Agent* (voir la description de cette classe en section 5.6.4) en lui passant en paramètres les besoins du client et les règles du marché sur lequel on désire transiger.

5.6.3.3. L'interface usager *AgentFacPanel*

Cette interface, utilisée par *AgentFactory*, permet à l'utilisateur de pouvoir spécifier le nom du fichier de règles que *AgentFactory* aura à analyser et d'activer la lecture de celui-ci. Ensuite l'utilisateur pourra spécifier ses besoins, soient le prix de réserve ainsi que la quantité, et donner son nom pour le marché sur lequel son agent ira négocier. L'interface lui permet de déclencher la création de l'agent et il peut y observer le déroulement de la négociation (voir la section 5.7).

5.6.4. La classe « Agent »

Nous ne décrivons ici que les attributs et les méthodes que nous jugeons importants pour la compréhension générale du fonctionnement interne de l'agent.

5.6.4.1. Les attributs de la classe « Agent »

Parmi les attributs principaux de la classe *Agent*, notons d'abord la variable *state*, qui définit l'état interne de l'agent.

Les variables *needs* ainsi que *negoRules* servent de références afin que l'agent puisse toujours savoir quels sont les besoins de l'utilisateur ainsi que les règles de la négociation.

Au niveau de la communication, on retrouve deux variables de type *ObjectInputStream*, l'une entrante et l'autre sortante : ces deux variables sont les points d'entrée et de sortie des messages que l'agent échange via sa *socket*. Les messages sont en fait des objets *Message* (voir la section 5.6.2) qui sont sérialisés par Java et passés par la *socket*. À l'autre bout ils sont désérialisés.

Une variable *portfolio* sert à tenir à jour l'agent sur les gains qu'il a pu faire ainsi que sur les quantités qu'il lui reste à acquérir.

5.6.4.2. La méthode *run*

La classe *Agent* est un *thread*. Après l'instanciation de l'agent, l'*AgentFactory* démarre le *thread* agent en appelant la méthode *start*. Ceci déclenche aussitôt l'exécution de la méthode *run* de ce dernier. La méthode *run* est organisée exactement selon le modèle d'architecture présentée en section 5.5.1. Dans une boucle, on appelle successivement l'écoute de l'environnement par la méthode *see* (voir la section 5.6.4.3), la mise à jour de l'état interne de l'agent à la lumière des informations recueillies, par le biais de la méthode *updateState* (voir la

description en section 5.6.4.4), et enfin la prise de décision quant à l'action à entreprendre, par l'intermédiaire de la méthode *action* (voir la description en section 5.6.4.5). On sort de la boucle lorsque l'agent se retrouve dans un état « Fin », et on se déconnecte alors du serveur.

5.6.4.3. La méthode *see*

La méthode *see*, comme son nom l'indique, est destinée à voir ou plutôt *percevoir* l'environnement. Dans le contexte dans lequel évolue l'agent, cette méthode est à l'écoute des messages qui pourraient lui parvenir. Typiquement on fait appel de manière transparente à une méthode qui va lire les messages entrant sur le *socket*. Les messages sont en fait des objets dans lesquels on retrouve tous les attributs nécessaires (voir section 5.6.2 sur la classe *Message*).

5.6.4.4. La méthode *updateState*

Toujours selon le modèle, il nous faut à présent mettre à jour l'état interne de notre agent, et la méthode *updateState* est là pour le faire. À la lumière des messages que l'agent a reçus par le biais de la méthode *see*, la méthode *updateState* va pouvoir déterminer dans quelle position l'agent se retrouve. Par exemple si l'agent a envoyé un ordre et qu'il reçoit une réponse du serveur qui confirme qu'il est leader, la méthode *updateState* met à jour l'état de l'agent d'une part, et d'autre part indique à la méthode *action* (voir section suivante) comment l'agent devra se comporter.

5.6.4.5. La méthode *action*

Cette méthode remplit exactement le rôle qui lui est conféré dans l'architecture que nous avons présentée (voir section 5.5), à savoir exécuter une action.

Si par exemple on doit envoyer une mise, la méthode va construire le message et y intégrer les différents éléments (le montant, la quantité, etc.) et le faire ensuite parvenir à l'agent serveur.

5.6.5. Résumé des classes et méthodes

La Figure 5.4, à la page suivante, fait un résumé des classes avec leurs attributs et leurs méthodes.

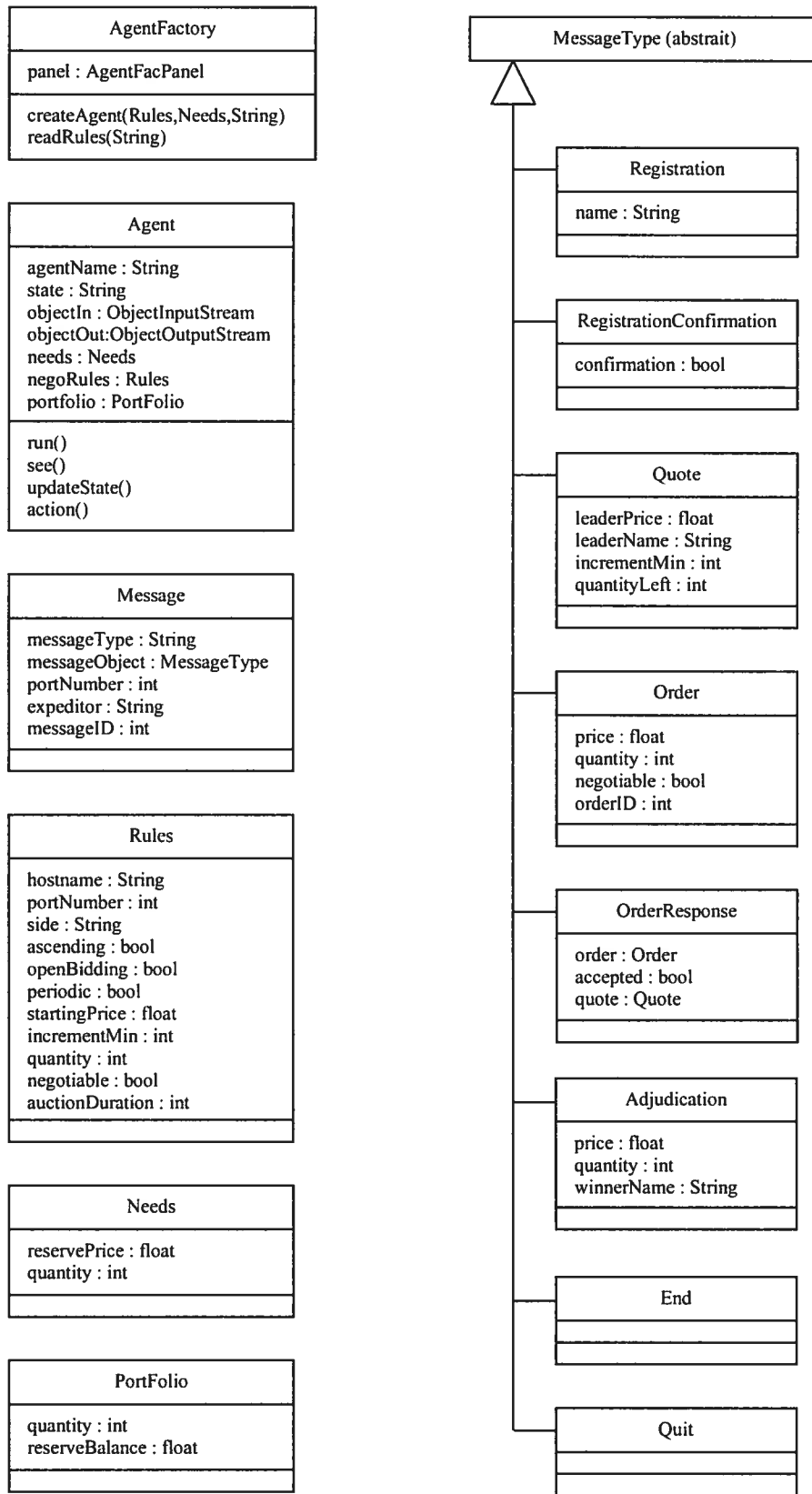


Figure 5.4 - Diagrammes des classes

5.7. Fonctionnement général

Nous présentons ici le fonctionnement général du système que nous avons implémenté. Lorsque l'on démarre le programme, l'interface nous apparaît (voir Figure 5.5). L'utilisateur doit en premier lieu rentrer au clavier le nom du fichier texte contenant les règles décrivant le marché sur lequel l'agent ira négocier.

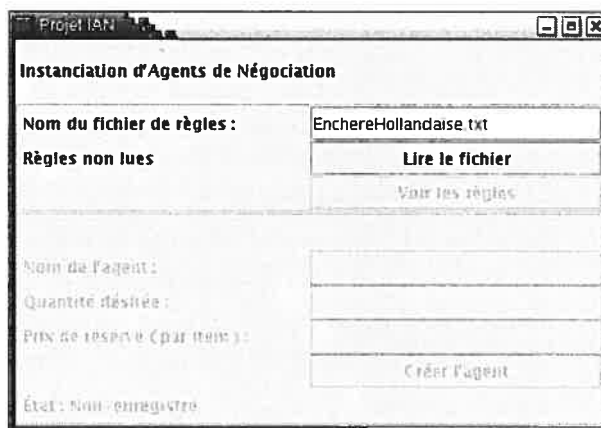


Figure 5.5 - Interface usager lors du démarrage.

Puis il faut cliquer sur le bouton « Lire le fichier ». Évidemment le fichier de règles doit être du bon format (voir la section 5.6.3.1). Une fois que le fichier a été lu, l'utilisateur est invité à remplir les champs qui s'offrent à lui, soient le nom de l'agent, la quantité désirée et le prix de réserve (voir Figure 5.6).

Projet IAN

Instanciation d'Agents de Négociation

Nom du fichier de règles : EnchèreMandat001.txt

Fichier lu

Lire le fichier

Voir les règles

Nom de l'agent : Mon_agent

Quantité désirée (max. 1) : 1

Prix de réserve (par Item) : 150

Créer l'agent

État : Non enregistré

Figure 5.6 - Interface usager après la lecture du fichier de règles.

L'utilisateur peut également visualiser les règles qui ont cours pour ce marché en cliquant sur le bouton « Voir les règles ». Une nouvelle fenêtre s'affiche alors qui permet de connaître les valeurs des différents champs définissant le marché (voir Figure 5.7).

Instanciation d'Agents de Négociation

Nom du fichier de règles : EnchèreMandat001.txt

Fichier lu

Lire le fichier

Voir les règles

Nom d

Quant

Prix de

État :

Règles

Nom du serveur	auction.serverfred.nl
Numéro de port	8998
Prix de départ	200
Incrément minimal	0
Côté (achat, vente)	buy
Ascendante ?	false
Périodique ?	false
Quantité mise en vente	1
Enchère ouverte ?	false
Durée (sec)	120
Négociable ?	false

Figure 5.7 - Fenêtre permettant de visualiser les règles du marché.

6. Analyse du travail effectué

6.1. Introduction

Dans les chapitres précédents, nous avons commencé par étudier les négociations, les enchères puis la technologie et les architectures agents, pour ensuite proposer un modèle de création d'agent qui propose une réponse à la problématique que nous avons présentée au début de ce travail; nous avons également effectué une implémentation correspondant à ce modèle.

Dans ce chapitre, nous présentons l'approche que nous avons adoptée pour tester notre implémentation afin de pouvoir valider notre modèle, nous analysons les résultats obtenus et en tirons des conclusions, que nous exposons.

6.2. Analyse des comportements

Afin de vérifier notre modèle de création d'agents, nous avons effectué des tests sur notre implémentation à partir de plusieurs cas de figure et à l'aide de différents serveurs de négociations. Nous expliquons ici la méthode que nous avons employée et présentons les comportements des agents que nous avons observés lors de ces tests.

6.2.1. Procédures de tests

Avant même de pouvoir effectuer nos tests, notre travail a consisté, comme nous en avons fait mention à la section 5.6.1.1, à créer ou modifier des serveurs de négociations (que nous appelons « agents serveurs ») existants afin que ceux-ci puissent supporter les cinq modèles d'enchères que nous avons étudiés (voir section 3.3). Les agents serveurs dont il est question ici sont au nombre de quatre : *ServerE*,

pour l'enchère anglaise à une unité, *ServerD*, pour les enchères hollandaises à une ou plusieurs unités, *ServerS* pour l'enchère à enveloppe fermée et *ServerTPN* pour la négociation à deux pairs.

Afin de tester notre implémentation, nous avons d'une part créé, pour chaque type d'agent serveur, plusieurs négociations, dont nous avons ensuite exprimé les règles dans un fichier texte (dans le format décrit en section 5.6.3.1). Par exemple, nous avons créé plusieurs enchères anglaises différentes à partir de l'agent serveur *ServerE* en changeant les paramètres tels que le prix de départ, l'incrément minimum et la durée.

Puis, pour chacune des négociations différentes ainsi créées pour chaque agent serveur, nous avons testé notre système de création d'agents en modifiant les besoins de l'utilisateur, soient le prix de réserve et la quantité, le cas échéant.

Nous avons également fait participer successivement un seul puis plusieurs agents (jusqu'à quatre) en même temps sur chacune des négociations, sauf pour le cas du serveur de négociation à deux pairs.

Le but de ces tests était de pouvoir vérifier qu'à partir des états dans lesquels se trouvaient les agents, ceux-ci agissaient de manière cohérente en tenant compte à la fois des règles de l'enchère et des besoins de l'utilisateur. Il nous fallait alors s'assurer d'avoir une bonne variété de cas afin qu'au total les agents créés se retrouvent dans tous les états et que puissent survenir tous les événements décrits en section 5.4.2.

6.2.2. Comportements observés

Les agents créés par notre système à partir des règles de négociation et des besoins de l'utilisateur ont réagi aux différents tests que nous leur avons fait subir conformément aux états et événements présentés en section 5.4.2. Nous avons noté les cas où certaines variations dans les résultats pouvaient se produire.

Dans le cas des enchères anglaises, lorsque la durée d'enchère était courte, on remarquait alors que les agents ayant les prix de réserve les plus élevés n'étaient pas forcément favorisés par rapport aux agents ayant des prix de réserve plus bas. Ce phénomène s'explique par le fait que les mises n'augmentaient pas assez vite pour atteindre les prix plafonds des agents et pour qu'ainsi ces derniers se retirent progressivement de l'enchère. Si les mises n'augmentaient pas rapidement, c'est parce que nous avons volontairement induit à l'agent serveur un certain délai dans la transmission des cotes afin que ce dernier n'envoie pas sans arrêt des messages pour tenir au courant les participants du déroulement de l'enchère. Ce délai était tout au plus de quelques secondes.

Par ailleurs, le nombre des participants n'a pas eu d'incidence sur le comportement des agents.

Dans l'enchère hollandaise à une seule unité, les agents ont réagi de la manière attendue, peu importe le nombre de participants.

Pour la même enchère à n unités, nous n'avons pas réussi à faire en sorte que la condition « besoins non-comblés » soit satisfaite, et ce pour la raison suivante : il aurait fallu inculquer aux agents une forme de stratégie d'achats par parties. Par exemple, si un prix total maximum donné pour acheter 5 produits est 100\$, un agent pourrait décider d'attendre que le prix unitaire baisse à 20\$ pour acheter d'un coup les 5 unités voulues. Mais un autre agent peut très bien décider de n'en acheter que 3 lorsque le prix atteint 20\$ l'unité, et d'en acheter 2 autres à un prix plus bas après. Un troisième pourrait également décider stratégiquement d'en acheter 2 à 30\$, pour être sûr d'en avoir au moins deux, et d'acheter les 3 derniers lorsque le prix aura baissé suffisamment.

Cet exemple nous montre qu'il nous aurait fallu établir des stratégies pour l'achat de plusieurs unités, ce que nous n'avons pas inclus dans les préférences de l'utilisateur, car comme nous l'avons fixé dès le départ, le domaine des stratégies ne faisait pas partie de notre travail.

L'enchère à enveloppe fermée, le plus simple des cinq cas de figure, n'a pas posé de problème, et les agents ont réagi de la manière attendue.

Enfin, pour ce qui est de la négociation bipartite, bien que le comportement des agents soit bien défini au niveau des actions relativement aux états et aux évènements, nous avons dû inculquer à nos agents une règle de marchandage, transparente pour l'utilisateur qui n'avait qu'à spécifier un prix de réserve. La règle de marchandage que nous avons spécifiée détermine le prix à proposer proportionnellement à la différence entre le prix de réserve de l'utilisateur et le prix proposé par l'agent serveur. Ce choix est aléatoire, mais n'influence en rien le résultats des tests, puisqu'il s'agissait de vérifier que, dans les états dans lesquels les agents pouvaient se trouver, ils effectuaient l'action qui leur était définie.

6.3. Limites

Après avoir présenté notre procédure, les tests que nous avons effectués, ainsi que les comportements obtenus, nous abordons ici les limites de notre modèle, du point de vue conceptuel autant que technologique.

6.3.1. Limites au niveau conceptuel

Dans l'étude qui a précédé notre conception, nous n'avons analysé que les marchés suivants : l'enchère anglaise, l'enchère hollandaise à un ou plusieurs items, l'enchère par l'offre secrète et la négociation à deux pairs. Par le biais des diagrammes, nous avons dégagé les principaux états dans lesquels un participant peut se retrouver. Toutefois, bien que les marchés étudiés soient représentatifs de ce que l'on retrouve le plus souvent sur Internet, nous ne pouvons affirmer que les états que nous avons dégagés soient les seuls que l'on peut retrouver dans tout type de marché. D'un autre côté, rien ne nous permet d'affirmer avec certitude que l'étude

d'un plus grand nombre de marchés nous aurait forcément conduit à trouver tous les états et évènements possibles. Cela supposerait que les chorégraphies possibles des négociations constituent un ensemble fini, et que les états possibles issus de ces chorégraphies constituent eux-mêmes un ensemble fini, ce que nous ne pouvons affirmer sans une étude poussée qui dépasse largement le cadre de ce mémoire. Notre travail se veut un premier pas vers la généralité.

Par ailleurs, l'aspect stratégique n'a pas été pris en compte dans notre travail. Nous avons limité la stratégie de nos agents au strict minimum, à savoir miser lorsque le prix de réserve est atteint. Nous avons fait le choix de ne pas aborder cette problématique car elle dépasse le cadre de notre étude. Cependant, comme nous l'avons constaté lors de l'analyse des comportements des agents dans la section 6.2.2, cet aspect est important si on désire aborder certains types de négociations, où certaines décisions d'actions dépendent en grande partie des stratégies.

6.3.2. Limites au niveau technologique

Une des limites technologiques de notre travail est le format des messages échangés entre nos agents et l'agent serveur. Nous avons pris comme exemple des agents serveurs qui échangeaient le même genre de messages avec nos agents, soient des objets semblables. Lors de la conception de notre système, nous avons pris comme hypothèse implicite que les classes utilisées par nos agents l'étaient également par les agents serveurs, ce qui est loin d'être le cas dans la réalité. Cependant, rappelons que l'objectif de notre travail ne visait pas la compatibilité des protocoles d'échanges entre n'importe quel type de serveur et d'agent, mais plutôt la généralité au niveau comportemental.

Une autre hypothèse implicite que nous avons faite lors de notre travail a été de considérer que le fichier de règles de chaque négociation pouvait être fourni, et

dans un format bien précis, ce qui n'est pas forcément le cas pour la plupart des serveurs de négociation.

6.4. Résumé

Dans ce chapitre, nous avons dans un premier temps présenté et expliqué les procédures de tests que nous avons suivies, tests destinés à étudier la validité de notre modèle de création par l'entremise de l'implémentation que nous en avons faite. Nous avons rapporté les observations que nous avons effectuées lors de ces tests et avons ensuite discuté les résultats obtenus. Nous avons pu constater que l'implémentation que nous avons réalisée de notre modèle avait donné les comportements attendus, et que les agents créés par notre système répondaient de manière conforme, au regard des règles de négociations et des besoins émis.

Dans un second temps, nous avons exposé les limites de notre modèle, au niveau conceptuel et technologique, et avons vu notamment l'intérêt d'y adjoindre une analyse des stratégies à employer lors de certaines négociations.

7. Conclusion générale

7.1. Revue du mémoire

Dans cette section, nous faisons un bref retour sur les éléments auxquels nous nous sommes intéressés dans ce mémoire et les passons en revue.

Dans un premier temps, nous avons mis en contexte notre recherche en présentant le projet Consensus, qui s'intéresse aux négociations électroniques combinées. Nous avons par la suite exposé notre problématique, celle de la création d'agents adaptés à tout type de négociation. Nous nous sommes fixés un certain nombre d'objectifs, parmi lesquels la conception d'un modèle de création générique d'agents de négociation.

Nous avons présenté les négociations et les enchères, les avons définies et les avons étudiées à travers plusieurs exemples. Nous en avons cerné les caractéristiques principales et en avons donné une classification, définie en deux grands groupes : les attributs de type de négociation et les attributs de la négociation.

Par la suite, nous avons présenté une méthode de modélisation des négociations utilisant les automates à états finis, et l'avons appliquée à cinq négociations, parmi les plus représentatives dans notre contexte de travail. Ces modélisations nous permirent de dégager les états principaux possibles dans lesquels un participant peut se retrouver.

Puis nous avons abordé la technologie agent, et en avons étudié quatre architectures principales.

Ensuite, nous avons présenté notre modèle de création générique d'agents de négociation, en expliquant et en justifiant nos choix de conception à la lumière des éléments que nous avons étudiés tout au long du travail. Nous avons enfin décrit l'implémentation de notre modèle et son fonctionnement.

Finalement, nous avons présenté les procédures de tests que nous avons effectués ainsi que les résultats obtenus et avons analysé les limites de notre modèle.

7.2. Rencontre des objectifs

Dans cette section, nous reprenons un à un les objectifs que nous avons énoncés au début de ce travail et les comparons avec le travail accompli.

- 1) *étudier les négociations et les enchères afin de pouvoir en donner une classification.* Dans le chapitre 2, nous avons donné des définitions des négociations et des enchères, avons donné des exemples et avons dégagé les caractéristiques principales afin d'en arriver à une classification, telle que présentée en section 2.3.9. Cette classification définit les négociations selon deux classes d'attributs : les attributs relatifs au type de la négociation, et les attributs de la négociation.
- 2) *modéliser les négociations les plus répandues du point de vue du participant afin d'analyser et de dégager les états essentiels.* Dans le chapitre 3, nous avons présenté une méthode proposée par Kumar et Feldman [Kumar and Feldman, 1998a] qui permet de représenter des négociations à l'aide d'automates à états finis. Nous avons utilisé cette méthode pour représenter cinq grands types de négociations, largement répandues sur Internet, soient l'enchère anglaise, l'enchère hollandaise à un et à plusieurs items, l'enchère par l'offre secrète ainsi que la négociation à deux pairs. Ces représentations nous ont permis d'analyser, de regrouper et de dégager les états principaux, tels que vus en section 3.4.

- 3) *comprendre la technologie des agents et en décrire les architectures principales.* Dans le chapitre 4, nous avons commencé par définir le concept d'agent, en avons fait un bref historique avant d'en étudier quatre architectures importantes : délibérative, réactive, BDI, et hybride.
- 4) *concevoir un modèle de création générique d'agents, à la lumière des concepts étudiés plus haut, et en faire l'implémentation.* Nous avons présenté au chapitre 5 un modèle de création d'agents utilisant comme paramètres d'entrée les besoins de l'utilisateur ainsi que les règles de la négociation, règles exprimées selon la classification établie en section 2.3.9. Les états essentiels que nous avons réussi à dégager dans la section 3.4 ainsi que notre choix pour l'agent d'une architecture réactive avec maintien d'états nous ont permis d'envisager des agents génériques, dans leurs comportements atomiques, en associant chaque état-événement à une action bien définie. En tenant compte des règles de la négociation et des besoins de l'utilisateur, ces agents génériques peuvent avoir un comportement spécifique au contexte d'une négociation particulière. Nous avons par la suite présenté et expliqué notre implémentation.
- 5) *discuter les résultats empiriques obtenus afin d'évaluer le modèle proposé.* Nous avons établi, au chapitre 6, la procédure que nous avons suivie afin d'effectuer nos tests. Nous avons ensuite discuté les comportements observés lors de ces tests et à la lumière de ces résultats, il apparut, d'une part que notre modèle était adapté, et d'autre part qu'un aspect que nous n'avions pas couvert par choix, soit la problématique de la stratégie, avait une importance plus importante que prévue et méritait qu'on s'y attarde davantage.

Il apparaît, après l'analyse faite ci-dessus, que nous avons atteint les objectifs que nous nous étions fixés au début de ce travail. Quelques points n'ont pas pu être abordés en profondeur et c'est la raison pour laquelle nous parlerons, dans la

prochaine section, des travaux futurs que nous envisageons pour améliorer notre modèle.

7.3. Travaux futurs

Nous avons constaté, à travers ce travail et plus particulièrement au cours de l'analyse des résultats obtenus, que nous pouvions aller plus loin dans l'automatisation des négociations, et voyons des extensions possibles à notre modèle afin de le rendre encore plus flexible et capable de pouvoir permettre la création d'agents sur un plus grand nombre de places de marchés.

7.3.1. Variété des marchés et stratégies

Dans notre travail, nous nous sommes concentrés sur la capacité qu'ont les agents de pouvoir négocier sur plusieurs marchés différents. Comme nous l'avons vu plus tôt, notre apport constitue un premier pas dans cette voie et l'étude d'un plus grand nombre de négociations, et l'analyse des modélisations de celles-ci nous permettrait d'en tirer une gamme plus étendue d'états, d'événements et d'actions possibles.

Par ailleurs, la description plus poussée des règles d'une négociation nous permettrait de mieux cerner les éventualités possibles. Il s'agirait donc de raffiner et de compléter les attributs décrivant une négociation.

Finalement, une étude approfondie des stratégies adoptées par un participant à une négociation devrait être effectuée afin de rendre plus solide notre modèle. En effet, des stratégies dépendent les actions prises par un agent. Le fait d'établir de manière détaillée la composition d'une stratégie, c'est à dire les variables et les conditions précises dans un contexte donné, ferait en sorte de préciser les actions

atomiques et les mécanismes de vérification mis en place pour prendre une décision quant à l'action à effectuer.

7.3.2. Vers les systèmes multiagents

Dans le chapitre 4, nous nous sommes attachés à définir ce qu'est un agent et en avons étudié plusieurs architectures. Celles-ci nous ont renseigné quant au comportement qu'un agent peut avoir dans un certain contexte. La plupart du temps, cependant, un agent n'agit pas seul et est appelé à côtoyer, voire à collaborer avec d'autres agents afin de réaliser des tâches. C'est le cas dans le projet Consensus. Les systèmes dans lesquels plusieurs agents évoluent et interagissent pour atteindre un objectif sont qualifiés de systèmes multiagents (SMA). Selon Durfee et Lesser [Durfee and Lesser, 1989], un système multiagents peut être défini comme étant un réseau de solutionneurs de problèmes, en configuration dispersée, qui travaillent ensemble pour résoudre des problèmes qui sont au delà des capacités et des connaissances de chaque solutionneur pris individuellement. Ces solutionneurs de problèmes, ou agents, sont autonomes et peuvent être de natures différentes.

7.3.2.1. Communication entre agents

Les agents communiquent entre eux de manière à atteindre plus efficacement les buts individuels ou collectifs dans l'environnement dans lequel ils évoluent. La communication permet aux agents de coordonner leurs actions et leurs comportements, rendant le système plus performant.

Comme nous l'avons vu en section 1.2.2, le projet Consensus s'intéresse aux négociations combinées. Chaque agent peut, comme nous l'avons montré dans ce travail, négocier de manière indépendante, mais il pourrait également communiquer avec ses pairs pour échanger des informations concernant l'état de la négociation à laquelle il participe, par exemple s'il est leader ou s'il vient d'abandonner. Les autres agents impliqués dans les négociations qui y sont combinées peuvent alors

agir en conséquence. C'est donc l'aspect des communications que nous pourrions développer afin de permettre aux agents créés d'échanger des informations entre eux.

Dans un dialogue, chaque agent peut avoir un rôle actif, passif ou les deux, ce qui peut correspondre respectivement à maître, esclave ou pair. Selon Huhns et Stephens [Huhns and Stephens, 1999] il existe deux types de messages: les requêtes et les indications.

	Agent simple	Agent passif	Agent actif	Agent pair
Reçoit des indications	*	*	*	*
Reçoit des requêtes		*		*
Envoie des indications		*	*	*
Envoie des requêtes			*	*

Tableau 7.1 - Types d'agents et leurs capacités [Huhns and Stephens, 1999]

Le Tableau 7.1 présente les capacités requises par les agents dépendamment de leur rôle dans le dialogue. Par exemple les agents doivent avoir la capacité d'accepter de l'information, peu importe qu'ils aient un rôle actif ou passif. Ceux qui sont actifs doivent avoir la capacité d'émettre des requêtes, etc.

Les agents peuvent s'envoyer des messages directement, mais une autre manière d'échanger de l'information est celle du tableau noir (*blackboard* en anglais). Cette technique, utilisée en intelligence artificielle, permet de définir une mémoire partagée par différents systèmes. Les agents utilisant un tableau noir peuvent écrire des messages, insérer des résultats partiels de leurs calculs et obtenir de l'information [Chaib-draa et al., 2001]. Cette technique permet dans bien des cas de simplifier les échanges d'informations entre les agents, surtout lorsque ceux-ci sont nombreux dans un SMA. Par exemple, plutôt que d'envoyer de l'information à tous les agents concernés, un agent x peut se servir du tableau noir pour mettre

l'information que les autres viendront chercher au moment opportun. De même, de nombreux agents peuvent aller vérifier une information sur le tableau noir plutôt que de la demander à chaque fois à l'agent dont dépend cette information, même si celle-ci change constamment.

7.3.2.2. La coordination

Un autre aspect important à considérer dans un SMA est la *coordination*. Celle-ci peut être vue de deux façons : si elle implique deux agents concurrents ou simplement indépendants, on parle alors de *compétition*, alors que s'il s'agit d'agents travaillant ensemble pour l'atteinte d'un même but on parle de *coopération*.

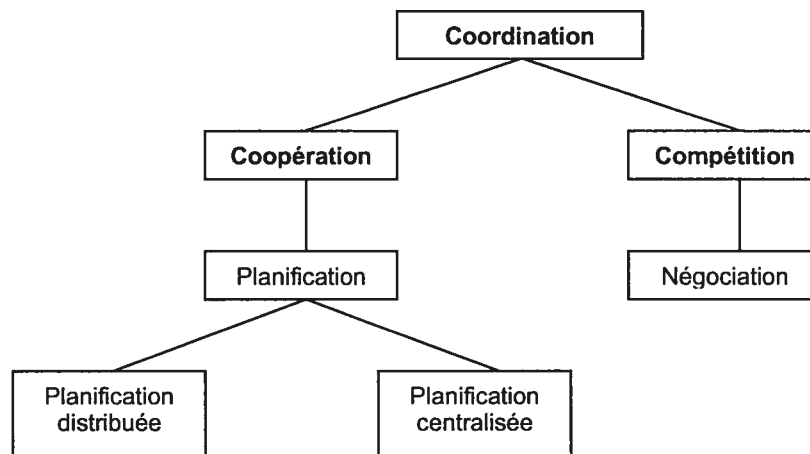


Figure 7.1 – Coordination entre agents [Huhns and Stephens, 1999]

La Figure 7.1 tirée de Huhns et Stephens [Huhns and Stephens, 1999] résume cette façon de voir les choses. On constate que si les agents sont en coopération, il faut établir une planification entre eux-ci (centralisée ou non), alors que s'ils sont en compétition il faudra avoir recours à la négociation.

7.4. Conclusion

Ce mémoire a présenté un modèle de création générique d'agents de négociation pour l'automatisation des négociations électroniques. Nous avons développé une architecture qui répond à un besoin de généricité pour un nombre limité de type de négociations; en cela notre travail constitue un premier pas vers une généricité plus importante. Nous croyons que celle-ci pourrait être obtenue par le rajout et le raffinement des attributs, donnant ainsi une description encore plus détaillée des règles de négociation ; par l'analyse d'un plus grand nombre de marchés afin de déterminer le plus d'états et événements possibles ; en tenant compte d'un plus grand nombre de variables pouvant affecter le déroulement d'une négociation, comme par exemple les stratégies possibles à adopter pour un agent mis dans un contexte particulier. Par ailleurs, nous pensons que nous pourrions développer la communication et la coordination entre les agents afin que ceux-ci coopèrent entre eux et s'intègrent ainsi dans un système multiagents.

Les agents permettent à l'heure actuelle d'automatiser les négociations ou sont en voie de le faire. Nous croyons que la création générique d'agents pourra, quant à elle, permettre d'automatiser la création de ces agents pour rendre le processus d'achat électronique par un usager le plus simple possible

Bibliographie

[Amazon, 2003] Amazon.com . <http://www.amazon.com> , 2003.

[Anthony and Jennings, 2002] P. Anthony, and N. R. Jennings. Agents in Auctions: The State of the Art. In *Proceedings of the 1st International Conference on Artificial Intelligence in Engineering and Technology (ICAIET-2002)*, pages 725-731, Sabah, Malaysia, June 2002.

[Beam and Segev, 1997] C. Beam, and A. Segev. Automated negotiations: A survey of the state of the art. *Wirtschaftsinformatik*, 39(3):-268, 1997.

[Benyoucef et al., 2001a] M. Benyoucef, H. Alj, M. Vézeau, and R. K. Keller. Combined Negotiations in E-Commerce: Concepts and Architecture. *Electronic Commerce Research Journal*, 1(3):277-299, July 2001.

[Benyoucef et al., 2001b] M. Benyoucef, H. Alj, and R. K. Keller. An Infrastructure for Rule-Driven Negotiating Software Agents. In *Proceedings of the Twelfth International Workshop on Database and Expert Systems Applications (DEXA 2001)*, pages 737-741, Munich, Germany, September 2001.

[Brooks, 1986] R.A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14-23, 1986.

[Cassady, 1967] R. Cassady. *Auctions and auctioneering*. Berkeley: University of California Press, 1967.

[Chaib-draa et al., 2001] B. Chaib-draa, I. Jarras, and B. Moulin. Systèmes multi-agents : principes généraux et applications. In J.P. Briot and Y. Demazeau editors, *Principes et architecture des systèmes multi-agents*, Hermes, Paris, France, 2001.

[Chapman, 1987] D. Chapman. Planning for conjunctive goals. *Artificial Intelligence*, 32:333-377, 1987.

[Christie's, 2003] Christie's History.
<http://www.christies.com/history/overview.asp> , 2003.

[Durfee and Lesser, 1989] E.H. Durfee and V. Lesser. Negotiating task decomposition and allocation using partial global planning. In L. Gasser and M. Huhns, editors, *Distributed Artificial Intelligence Volume II*, pages 229–244. Pitman Publishing: London and Morgan Kaufmann: San Mateo, CA, 1989.

[eBay, 2003] About Ebay: Company Overview.
<http://pages.ebay.com/community/aboutebay/overview/index.html> , 2003.

[Fikes and Nilsson, 1971] R.E. Fikes, and N.J. Nilsson. Strips: A New Approach to the Application of Theorem-Proving to Problem-Solving. *Artificial Intelligence*, 2(3):189–208, 1971.

[Friedman and Rust, 1993] D. Friedman, and J. Rust, editors. *The Double Auction Market: Institution, Theories, and Evidence*. Addison Wesley Publishing, Reading, MA, 1993.

[Guttman et al., 1998] R.H. Guttman, A.G. Moukas, and P. Maes. Agent-mediated Electronic Commerce: A Survey. *Knowledge Engineering Review*, 13(3), June 1998.

[Huhns and Singh, 1998] M.N. Huhns and M. P. Singh. Agents and Multi-agent Systems: Themes, Approaches, and Challenges. In M. N Huhns and M. P. Singh editors, *Readings in Agents*, pages 1-23, Morgan Kaufmann Publishers, San Francisco, CA, 1998.

[Huhns and Stephens, 1999] M. N. Huhns and L. M. Stephens. Multiagent Systems and Societies of Agents. In . In Gerhard Weiss editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, chapter 2, pages 79-120. The MIT Press, Cambridge, MA, 1999.

[JavaSpaces, 2003] JavaSpaces technology.
<http://java.sun.com/products/javaspaces> , 2003.

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

[Jennings et al., 2000] N. R. Jennings, S. Parsons, C. Sierra and P. Faratin. Automated Negotiation. In *Proc. 5th Int. Conf. on the Practical Application of Intelligent Agents and Multi-Agent Systems (PAAM-2000)*, pages 23-30, Manchester, UK, 2000.

[Jini, 2003] Jini Network Technology Archive. <http://java.sun.com/products/jini> , 2003.

[Klein, 1997] S. Klein. Introduction to Electronic Auctions. *EM - Electronic Markets*, 7(4): 3-6, 1997.

[Klemperer, 1999] P. Klemperer. Auction Theory: A Guide to the Literature. *Journal of Economic Surveys*, 13(3): 227-286, July 1999.

[Kumar and Feldman, 1998a] M. Kumar, and S.I. Feldman. Business negotiations on the Internet. Technical Report, IBM Institute for Advanced Commerce (IAC), March 1998. Available at <http://www.research.ibm.com/iac/tech-paper.html>

[Kumar and Feldman, 1998b] M. Kumar, and S.I. Feldman. Internet auctions. In *Third USENIX Workshop on Electronic Commerce*, pages 49-60, Boston, 1998.

[Lewicki et al., 1999] R.J. Lewicki, D. M. Saunders, and J.W. Minton. *Negotiation: Readings, Exercises, and Cases*, Third Edition, Irwin / McGraw-Hill, Boston, 1999.

[Lomuscio et al., 2001] A.R. Lomuscio, M. Wooldridge, and N.R. Jennings. A Classification Scheme for Negotiation in Electronic Commerce. In *Agent Mediated Electronic Commerce; A European AgentLink Perspective*, F.Dignum and C. Sierra editors , pages 19-33, Springer-Verlag, Berlin, 2001.

[Lucking-Reiley, 2000] D.H. Lucking-Reiley. Auctions on the Internet: What's Being Auctioned, and How? *Journal of Industrial Economics*, 48(3):227-252, September 2000.

[Maes et al., 1999] P. Maes, R.H. Guttman and A.G. Moukas. Agents that buy and Sell. *Communications of the ACM*, 42(3):81-91, 1999.

[McAfee and McMillan, 1987] R.P. McAfee, and J. McMillan. Auctions and bidding. *Journal of Economic Literature*, 25:699-738, June 1987.

[Milgrom, 1989] P. Milgrom. Auctions and Bidding: A Primer. *Journal of Economic Perspective*, 3(3):3-22, 1989.

[OMG, 2000] OMG document, *Agent Technology, Green Paper V. 1.0* , 2000.

[OnSale, 2003] OnSale.com - Company Info.
http://www.onsale.com/c2c/companyinfo/company_info.aspx , 2003.

[Rasmusen, 1989] E. Rasmusen. *Games and Information: An Introduction to Game Theory*. First edition, Oxford: Blackwell Publishers, 1989.

[Roth and Ockenfels, 2002] A. E. Roth, and A. Ockenfels. Last-Minute Bidding and the Rules for Ending Second-Price Auctions: Evidence from eBay and Amazon Auctions on the Internet. *American Economic Review*, 92(4):1093-1103, September 2002.

[Russell and Norvig, 1995] S. Russell, and P. Norvig. *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Prentice-Hall, 1995.

[Sotheby's, 2003] Sotheby's Services Corporate History.

http://search.sothebys.com/about/corporate/as_corphistory.html , 2003.

[Shubik, 1983] M. Shubik. Auctions, Bidding and Markets: An Historical Sketch. In: M. Shubik, R. Engelbrecht-Wiggins, and R. Stark editors, *Auctions, Bidding and Contracting: Uses and Theory*, NY: New York University Press, pages 33-52, 1983.

[Terpsidis et al., 1997] I. Terpsidis, A. Moukas, B. Pergioudakis, G. Doukidis, and P. Maes. The Potential of Electronic Commerce in Re-Engineering Consumer-Retail Relationships through Intelligent Agents. In J.-Y. Roger, B. Stanford-Smith, and P. Kidd. editors, *Advances in Information Technologies: The Business Challenge*, IOS Press, 1997.

[Vickrey, 1961] W. Vickrey. Counter Speculation, Auctions, and Competitive Sealed Tenders. *The Journal of Finance*, 16(1):9-37 1961.

[Wolfstetter, 1995] E. Wolfstetter. Auctions: An Introduction. *Journal of Economic Surveys*, 10(4):367-420, 1995.

[Wooldridge, 1999] M. Wooldridge. Intelligent Agents. In Gerhard Weiss editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, chapter 1, pages 27-77. The MIT Press, Cambridge, MA, 1999.

[Wooldridge and Jennings, 1995] M. Wooldridge and N.R. Jennings. Agent Theories, Architectures and Languages: A Survey. In M. Wooldridge and N.R. Jennings editors, *Intelligent Agents - Lecture Notes in Artificial Intelligence*, vol. 890, pages 1-39, Springer-Verlag, Berlin, February 1995.

[Zlatev and van Eck, 2003] Z. Zlatev, and P. van Eck. An Investigation of the Negotiation Domain for Electronic Commerce Information Systems. In: O. Camp, J. Filipe, S. Hammoudi, and M. Piattini editors, *Proceedings of the Fifth International Conference on Enterprise Information Systems, ICEIS 2003*, Angers, France, April 23-26. Volume 4, pages 386-391, 2003.

